Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Algorytmy poprawy czytelności formalnych rozumowań zapisanych w systemie naturalnej dedukcji

Rozprawa doktorska

Karol Pąk

Promotor
dr hab. Aleksy Schubert

Instytut Informatyki
Uniwersytet Warszawski

Maj 2013

Oświadczenie autora rozprawy:
oświadczam, że niniejsza rozprawa została napisana przeze mnie samodzielnie.

Maj 17, 2013 .....
data *Karol Pąk*

Oświadczenie promotora rozprawy:
niniejsza rozprawa jest gotowa do oceny przez recenzentów.

Maj 17, 2013 .....
data *dr hab. Aleksy Schubert*

The Algorithms for Improving Legibility of Natural Deduction Proofs

Abstract

In this dissertation the methods to improve legibility of existing formal reasonings written in natural deduction are presented. Computer assisted proof development frameworks can check the correctness of such reasonings, but any attempt to analyze details of the proofs scripts created in this way, according to opinion of some proof writers, is extremely difficult or even impossible. The readability of such arguments is a subjective quality which is understood by different proof writers in different ways. Still the analysis of their needs led to a distinguished set of criteria that facilitate making the formal deductions closer to the informal mathematical proofs.

First part of the dissertation describes an abstract model of mathematical proofs written in the Mizar language. This model expresses the intuitions connected with the reasonings, where the information flow in proof is regarded as a special kind of digraph. Based on this model notion and parameters associated with legibility criteria are formally defined in the second part of the dissertation. Improvement of readability has been realised by two separate approaches that are used in informal mathematical practice. The first approach is based on the finding fragments of reasoning and consists in isolation (extraction) of these fragments in the form of lemmas or encapsulation at the deeper levels of nested proof. The second approach to improvement of the readability consists in the modification of the order of independent steps written in the proof script. The methods that reorganize the order of steps focus mainly on the location of information used to justify a step. As a result of research based on the first approach, methods to extract or encapsulate reasoning fragments from existing deductions were elaborated. Also properties of reasoning fragments that determine the structure of statements which describing the information about reasoning contained in these fragments were described. In the second approach five parameters of legibility that are indicated as most important by the users users of Mizar database has been formally defined. Analysis of the proposed parameters related to improvement of proof readability revealed that four of the considered problems are NP-complete. Additionally, an auxiliary application to improve the readability of articles distributed in MML based on the most popular hierarchy of the considered parameters were created.

Keywords: Natural deduction proofs,Improving Legibility of proofs, formal languages NP–complete problems, Linear Arrangement, Acyclic partition.

Classification according to ACM: F.2.2, F.4.3, G.2.2, I.2.4.

Streszczenie

Przedmiotem badań opisanych w rozprawie doktorskiej są metody poprawy czytelności formalnych rozumowań zapisanych w systemie naturalnej dedukcji. Wykorzystanie komputerowej weryfikacji jest znanym narzędziem ułatwiającym sprawdzanie poprawności formułowanych rozumowań, aczkolwiek jakiekolwiek próby analizowania tak uszczegółowionych rozumowań są wyjątkowo trudne, a zdaniem niektórych niemożliwe. Czytelność takich wywodów jest pojęciem subiektywnym, różnie rozumianym przez poszczególnych autorów rozumowań. Analiza ich potrzeb przyczyniła się jednak do wyodrębnienia grupy kryteriów umożliwiających uczytelnienie formalnych rozumowań, poprzez upodabnienie ich postaci do takiej, która występuje w nieformalnych dowodach matematycznych.

W pierwszej części rozprawy został przedstawiony model abstrakcyjnego dowodu matematycznego odzwierciedlający rzeczywistą strukturę dowodów zapisanych w języku Mizar. Model ten umożliwia interpretowanie przepływu informacji w rozumowaniu jako szczególnego rodzaju skierowanych grafów acyklicznych. W oparciu o ten model w drugiej części rozprawy zostały formalnie opracowane pojęcia oraz wyznaczniki poprawy czytelności. Uczytelnianie formalnych rozumowań zostało zbadane pod kątem zastosowania dwóch rodzajów środków stosowanych w praktyce matematycznej. Jako pierwszy z nich, zostały zbadane metody odnajdywania lokalnych podrozumowań, a następnie ich wyizolowywania (wyodrębniania) w postaci lematów lub kapsułkowaniu na głębszych poziomach zagnieżdżenia. Drugim zaś analizowanym środkiem była reorganizacja niezależnych od siebie kroków rozumowań w sposobie ich uporządkowania w dowodzie, mająca na celu poprawę wybranych własności linearyzacji dowodu. W wyniku przeprowadzonych badań w zakresie pierwszego środka została skonstruowana metoda wyizolowywania i kapsułkowania fragmentów rozumowania przy zachowaniu poprawności modyfikowanego skryptu dowodowego oraz zostały zbadane własności fragmentów dowodu, które determinują budowę stwierdzenia opisującego rozumowanie zawarte w tych fragmentach. W zakresie zaś drugiego środka zostało opracowane pięć, najczęściej wskazywanych przez użytkowników bazy Mizar Mathematical Library wskaźników czytelności. Przeprowadzone badania nad złożonością problemu optymalizacji wartości przyjętych wskaźników wykazały, że optymalizacja czterech z nich wiąże się z rozwiązywaniem problemów NP–trudnych. Dodatkowo, zostały stworzone programy umożliwiające automatyczną poprawę czytelności skryptów dowodowych zapisanych w języku Mizar, których działanie opiera się na optymalizacji wartości opracowanych wskaźników przy zadanej przez użytkownika hierarchii ich ważności.

Słowa kluczowe: rozumowanie w systemie naturalnej dedukcji, poprawa czytelności dowodów, systemy formalne, problemy NP–zupełne, liniowe uporządkowanie, acykliczna partycja.

Klasyfikacja według ACM: F.2.2, F.4.3, G.2.2, I.2.4.

Spis treści

Wprowadzenie

1 Geneza badań

W naukach matematycznych komputerowa weryfikacja ułatwia sprawdzanie poprawności formułowanych rozumowań. Weryfikacja ta, choć wymaga szczegółowej analizy wszystkich kroków oraz możliwych przypadków, gwarantuje uniknięcie nie zawsze drobnych błędów, które czasami pojawiają się w nieformalnych dowodach matematycznych. Zapewnienie poprawności rozumowań na tak drobiazgowym poziomie wiąże się jednak z kosztem, którym jest kilkukrotne zwiększenie ich długości charakteryzowane tzw. współczynnikiem de Bruijna [16, 43, 96]. To zwiększenie długości jest jednym z czynników pogarszających czytelność rozumowań.

Problem ten istnieje od pierwszych prób formalizacji matematyki, przyczyniając się do zaniechania wielu z nich. Spowodował on m.in. porzucenie projektu grupy naukowców występujących pod pseudonimem Nicolas Bourbaki[1], który był rozwijany pomimo wykazania przez K. Gödla[2] niemożliwości realizacji w najbardziej idealistycznej postaci poglądów filozoficznych stanowiących podstawę programu D. Hilberta, zwanego współcześnie formalizmem, czy też dalszych prac nad kolejnym tomem dzieła *„Principia mathematica"* tworzonego przy ekstremalnie rygorystycznym podejściu do formułowaniu definicji i dowodów, reprezentowanego przez A. N. Whiteheada i B. Russella[3]. Zdaniem niektórych koncepcje systemów formalnych umożliwiające jasne strukturalizowanie dowodów, zapisanych w systemie dedukcji naturalnej, przedstawione przez G. Gentzena [30] lub S. Jaśkowskiego [40] utrudniają, czy wręcz uniemożliwiają, zastosowanie ich do wiernej reprezentacji dowodów twierdzeń prowadzonych w praktyce przez matematyków [80].

Rozkwit formalizacji był możliwy dopiero w wyniku wykorzystania komputerów do weryfikacji poprawności tak uszczegółowionych rozumowań. Idea komputerowo wspomaganej automatyzacji i weryfikacji formalnych rozumowań matematycznych stanowiła bowiem punkt wyjścia do zbudowania systemów prowadzenia rozumowań formalnych, tj. Coq [1], Isabelle/Izar [92], HOL Light [37], Mizar [33]. Wykorzystanie komputerów nie tylko zlikwidowało problem sprawdzania poprawności, ale umożliwiło również prowadzenie rozważań formalnych na bardziej ogólnym poziomie, który jest bliższy temu, jaki występuje w nieformalnych dowodach matematycznych. Pomimo zbudowania takich systemów zapisane w nich formalne rozumowania są nadal

[1] Grupa francuskich matematyków zawiązała się w 1935 r. i podjęła projekt m.in. opracowania kursu matematyki prezentującego ówczesny stan wiedzy najważniejszych dziedzin matematyki w nurcie formalizmu. W wyniku tych działań została opracowana seria *Elementy matematyki (fr.* Éléments de mathématique*)*.

[2] Program przedstawiony przez Davida Hilberta na Międzynarodowym Kongresie Matematyków w Paryżu w 1900 r., mający na celu „ratowanie matematyki przed nieścisłościami".

[3] Zespół brytyjskich naukowców, twórców trzytomowego dzieła *Principia mathematica* dotyczącego podstaw matematyki, reprezentujących ekstremalnie rygorystyczne podejście do formułowania definicji i dowodów. Ich prace doprowadziły do powstania najbardziej skomplikowanego uzasadnienia równości $1 + 1 = 2$, które zostało zamieszczone na ponad 300 stronach.

nieczytelne dla przeciętnego użytkownika. Implikuje to konieczność dalszych badań w zakresie metod poprawy czytelności. Zagadnieniom tym została poświęcona prezentowana rozprawa doktorska.

2 Obszar badawczy

Obszarem badawczym rozprawy jest wdrażanie metod, które są wykorzystywane w procesie poprawy czytelności nieformalnych rozumowań matematycznych, w celu uczytelniania istniejących formalnych rozumowań zapisanych w systemie naturalnej dedukcji, ze szczególnym uwzględnieniem rozumowań zgromadzonych w bibliotece Mizar Mathematical Library (MML).

Wybór biblioteki systemu Mizar został podyktowany tym, iż jest ona najbardziej rozbudowanym na świecie, intensywnie rozwijanym przez ponad dwadzieścia lat, repozytorium sformalizowanej wiedzy matematycznej. Za wyborem bazy MML przemawia również fakt, że rozwiązania wykorzystywane w języku Mizar są inspiracją do opracowywania podobnych rozwiązań mających na celu polepszanie czytelności skryptów w innych uznanych systemach [65], tj. *Declare* [82], *Mizar Mode for HOL* [36], *Isar language for Isabelle* [93], *Mizar-light for HOL-light* [97], `miz3` *for HOL-light* [7], *MMode for Coq* [32], *declarative proof language (DPL) for Coq* [17], *Formal Proof Sketches* [98].

Analiza doświadczeń związanych z rozwojem i konserwacją bazy MML potwierdziła istnienie znanego powodu nieczytelności skryptów dowodowych. Jest nim dążenie do osiągnięcia konkretnego celu (tj. udowodnienia twierdzenia) przy jednoczesnym traktowaniu pobocznych wartości, np. czytelności tworzonych skryptów dowodowych, jako kwestie drugorzędne. Oczywiście efekt ten nasila się w przypadku długich rozumowań uzasadniających coraz trudniejsze zagadnienia zawarte w MML. Rozumowania te, choć są poprawne dla systemu weryfikującego, to są jednak mało eleganckie, niejednokrotnie wręcz chaotyczne, a człowiek może je zrozumieć jedynie przy bardzo dużym nakładzie pracy i wysiłku. Czytelność takich skryptów wpływa nie tylko na czas ich analizowania, który jest potrzebny do wyodrębnienia idei sformalizowanych w ten sposób dowodów matematycznych, ale również na łatwość ich konserwacji (rewidowania). Autorzy takich skryptów mają bowiem przekonanie, iż problem znajdowania i usuwania niepotrzebnych lub możliwych do skrócenia fragmentów nie jest z ich punktu widzenia istotny, gdyż zakładają oni, że wszystkie elementy tego procesu można łatwo zautomatyzować. Warto w tym miejscu podkreślić, że istnieją narzędzia umożliwiające automatyczne odnajdywanie i usuwanie niepotrzebnych fragmentów skryptów dowodowych [57, 58]. Z tym, że stosunkowo nieliczne badania zostały przeprowadzone nad uczytelnianiem pozostałej części rozumowania po usunięciu nadmiarowych fragmentów. Dodatkowo badania korzystające z narzędzi, które dokonują wyodrębniania mniej istotnych lub powtarzających się fragmentów rozumowania w postaci lematów, są prowadzone przy całkowitym pominięciu kwestii czytelności wynikowych rozumowań [73]. Mają one na celu jedynie minimalizację długości skryptów dowodowych.

Wybór rozwiązań wykorzystywanych w procesie uczytelniania nieformalnych rozumowań matematycznych został podyktowany tym, iż wielowiekowe doświadczenia związane z formułowaniem nieformalnych dowodów matematycznych umożliwiły wypracowanie pewnego „stylu", pozwala na czytelny zapis nawet rozbudowanych rozumowań. Oczywiście czytelność dowodu jest wartością subiektywną, różnie rozumianą przez poszczególnych autorów rozumowań, aczkolwiek wśród werbalizowanych poglądów na ten temat możemy wskazać powtarzający się pewien „zestaw" metod uczy-

telniania, który różni się co najwyżej kolejnością zhierarchizowania u poszczególnych autorów własności, jakie powinien posiadać czytelny dowód. Dodatkowo metody te mają swoje uzasadnienie w badaniach psychologicznych nad procesami poznawczymi człowieka.

Za tak określonym obszarem badawczym przemawia również fakt, że metody te są przenoszone na grunt dowodów formalnych przez wielu autorów skryptów dowodowych, dla których równie ważnym priorytetem w formalizacji jak powiększanie bazy sformalizowanych twierdzeń jest jej jakość.

3 Cele rozprawy

Celem rozprawy jest zbadanie złożoności obliczeniowej metod poprawy czytelności długich formalnych skryptów dowodowych. Cel ten będzie realizowany poprzez badanie efektywności algorytmów poprawiających czytelność rozumowań zapisanych w systemie G. Gentzena lub S. Jaśkowskiego [30, 40, 54].

W związku z tak nakreślonym celem głównym rozprawy zostały wyodrębnione następujące cele szczegółowe proponowanych badań:

1. *Wyodrębnienie zbioru informacji zawartych w poszczególnych krokach rozumowania, który umożliwia interpretowanie struktury rozumowania jako grafu skierowanego reprezentującego przepływ informacji między poszczególnymi krokami rozumowania, nazywanego dalej grafem dowodu.*

2. *Stworzenie abstrakcyjnego modelu grafu dowodu, który umożliwi uniezależnienie prowadzonych rozważań od konkretnego systemu weryfikującego, co uprości adaptację oczekiwanych rozwiązań do innych systemów.*

3. *Zdefiniowanie grupy wskaźników czytelności wyrażonych w terminach abstrakcyjnego grafu dowodu oraz zbadanie złożoności problemów grafowych optymalizujących wartości tych wskaźników.*

4 Problem badawczy

Problemy w zakresie obranej tematyki można sformułować w postaci dwóch następujących pytań ogólnych:

1. Czy i w jakim stopniu możliwe jest posługiwanie się abstrakcyjnym modelem grafu dowodu do analizy metod uczytelniania istniejących rozumowań formalnych zgromadzonych w bazie MML?

2. Na ile proponowane metody uczytelniania są efektywne czasowo, a co za tym idzie, na ile są stosowalne w procesie automatycznej poprawy czytelności sformalizowanych rozumowań zapisanych w systemie naturalnej dedukcji?

Zdecydowano się na łączne potraktowanie tych dwóch problemów badawczych, ponieważ przeprowadzenie badań nad poprawą czytelności bez ich uniezależnienia od konkretnego systemu utrudniłoby, a wręcz uniemożliwiłoby adaptację oczekiwanych rozwiązań do innych systemów.

5 Hipoteza badawcza

Hipoteza badawcza została sformułowana w następujący sposób: *nie jest możliwe osiągnięcie wartości optymalnej większości proponowanych wyznaczników poprawy czytelności, stosując algorytmy dokładne o złożoności wielomianowej. Nie jest*

również możliwe równoczesne osiągnięcie wartości optymalnych wszystkich proponowanych wyznaczników. Istnieją jednak algorytmy, umożliwiające optymalizację wielokryterialną wyznaczników poprawy czytelności, które działając przy ustalonej hierarchii ważności zaproponowanych metod, dostosowanej do potrzeb określonej grupy czytelników mogą skutecznie poprawić czytelność analizowanych rozumowań formalnych.

6 Metodyka badań

Do przygotowania rozprawy wykorzystane zostały: metoda empiryczna oraz metoda teoretyczna. Badania empiryczne wykorzystane w początkowym etapie badań polegały na wyodrębnieniu zbioru informacji zawartych w poszczególnych krokach rozumowania sformułowanych w języku Mizar, niezbędnych do skonstruowania grafów poszczególnych dowodów, które to grafy w adekwatny sposób reprezentowałyby przepływ informacji w istniejących skryptach dowodowych. W dalszej części badań metody empiryczne zostały wykorzystane do analizy doświadczeń w uczytelnianiu skryptów dowodowych, jak również do jakościowej weryfikacji przyjętych wskaźników czytelności w trakcie rozmów z doświadczonymi użytkownikami systemu Mizar. Niezwykle cenny okazał się również udział w konferencjach krajowych i zagranicznych, na których dyskusja nad zaprezentowanymi cząstkowymi wynikami badań umożliwiła dopracowanie wybranych wskaźników.

Metoda teoretyczna została wykorzystana do konstrukcji abstrakcyjnego modelu grafu dowodu. Przeprowadzone badania umożliwiły bowiem rozdzielenie własności grafu dowodu wynikających z wykorzystania systemu Mizar od własności, które są zakładane w systemie naturalnej dedukcji. Na skutek tego podziału przyjęty model może być stosowany do opisu struktury każdego poprawnie zbudowanego dowodu matematycznego. Badania nad własnościami grafu dowodu wynikającymi ze składni systemu Mizar umożliwiły również ustalenie reprezentatywnej rodziny abstrakcyjnych grafów dowodów, z których każdy graf ma własności dostateczne do skonstruowania poprawnego dla systemu Mizar rozumowania o strukturze opisanej przez ten graf. Dodatkowo została przedstawiona transformacja skryptów dowodowych, umożliwiająca modyfikację każdego rozumowania do postaci, dla której graf dowodu należy do tej rodziny. W dalszej części badań metody teoretyczne zostały również wykorzystane do zbadania złożoności problemów optymalizacyjnych przyjętych wskaźników poprawy czytelności.

7 Plan rozprawy

Rozprawa składa się z czterech rozdziałów. Rozdział 1 ma charakter wprowadzający i zawiera definicje podstawowych pojęć. W rozdziale 2 rozważamy model abstrakcyjnego grafu dowodu oraz przedstawiamy niezbędne pojęcia i fakty związane z tym modelem. Rozdział 3 jest poświęcony kolejno genezie i uzasadnieniu wykorzystywanych metod poprawy czytelności, sformułowaniu problemów optymalizujących wartości wskaźników czytelności oraz wynikom eksperymentalnych badań nad poprawą czytelności rozumowań. Następnie w rozdziale 4 zostały przedstawione wyniki dotyczące złożoności postawionych problemów optymalizacyjnych. Ostatnią merytoryczną część pracy stanowią wnioski końcowe, będące odpowiedzią na pytania sformułowane w problemach badawczych oraz ustosunkowanie się do hipotezy badawczej.

Rozdział 1

Pojęcia podstawowe

Poniżej zostały zdefiniowane pojęcia z zakresu teorii grafów, które są wykorzystywane w niniejszej rozprawie.

1.1 Grafy skierowane

Definicja 1.1. Grafem prostym G *będziemy nazywać uporządkowaną parę dwóch zbiorów* $\langle \mathcal{V}(G), \mathcal{E}(G) \rangle$*, niepustego zbioru* $\mathcal{V}(G)$ *i podzbioru zbioru wszystkich dwuelementowych podzbiorów zbioru* $\mathcal{V}(G)$*,* $\mathcal{E}(G) \subseteq \{\{v, u\} : v, u \in \mathcal{V}(G) \wedge v \neq u\}$*. Elementy zbioru* $\mathcal{V}(G)$ *będziemy nazywać* wierzchołkami *grafu* G*, natomiast elementy zbioru* $\mathcal{E}(G)$ *jego* krawędziami*.*

Prowadzone w rozprawie rozważania będą skupiały się wokół zagadnień wykorzystujących grafy skierowane, dlatego wszystkie pojęcia będą wprowadzone jedynie dla tego rodzaju struktur. Wykorzystywane oznaczenia w głównej mierze są zapożyczone z [99].

Grafem skierowanym lub *digrafem* D będziemy nazywać uporządkowaną parę dwóch zbiorów $\langle \mathcal{V}(D), \mathcal{A}(D) \rangle$, niepustego zbioru $\mathcal{V}(D)$ i podzbioru $\mathcal{A}(D) \subseteq \mathcal{V}(D) \times \mathcal{V}(D)$ zbioru wszystkich par. Elementy zbioru $\mathcal{V}(D)$ będziemy nazywać *wierzchołkami* digrafu D, natomiast elementy zbioru $\mathcal{A}(D)$ jego *krawędziami skierowanymi* lub krócej *łukami*. Przyjmujemy, że łuk (v, u) jest skierowany od v do u, a więc początkiem tego łuku jest v, a końcem u. Jeżeli nie prowadzi to do niejednoznaczności, łuk (v, u) będziemy oznaczać vu. Będziemy również zakładać, że rozważane digrafy nie zawierają łuków postaci vv (równoważnie, że rozważane digrafy są *proste*).

Ustalmy dowolny digraf D oraz zbiór $A \subseteq \mathcal{A}(D)$. W poniższych oznaczeniach będziemy wyróżniać zbiór A w celu uniknięcia niejasności podczas analizowania zależności między strukturami kilku digrafów skonstruowanych nad wspólnym zbiorem wierzchołków. Dodatkowo, jeśli zbiór A będzie pokrywał się ze zbiorem $\mathcal{A}(D)$, to wskazany zbiór A będzie często zastępowany w oznaczeniach przez symbol digrafu D lub zostanie całkowicie pominięty.

Łuk vu będziemy nazywali A–łukiem, jeśli $vu \in A$. Jeśli vu jest A–łukiem, to wierzchołek $u \in \mathcal{V}(D)$ będziemy nazywać *następnikiem* wierzchołka v w rodzinie A–łuków (lub krócej w A). Relację bycia następnikiem w A będziemy oznaczać przez $\underset{A}{\rightarrow}$. Podobnie, u będziemy nazywać *poprzednikiem* v w rodzinie A–łuków (lub krócej w A) wtedy i tylko wtedy, gdy v jest *następnikiem* u w A. Dodatkowo, jeśli u jest następnikiem v w domknięciu zwrotno-przechodnim relacji $\underset{A}{\rightarrow}$, to relację tę będziemy oznaczać $v \underset{A}{\rightarrow}^* u$ i będziemy mówić, że u jest *osiągalny z* v w A.

Zbiorem następników wierzchołka $v \in \mathcal{V}(D)$ w rodzinie A–łuków będziemy nazywać zbiór $\mathcal{N}_A^+(v) := \{u \in \mathcal{V}(D) : v \underset{A}{\rightarrow} u \wedge u \neq v\}$, natomiast *zbiorem poprzedników* v w rodzinie A–łuków będziemy nazywać zbiór $\mathcal{N}_A^-(v) := \{u \in \mathcal{V}(D) : u \underset{A}{\rightarrow} v \wedge u \neq v\}$.

Ciąg elementów $\{a_i\}_{i=1}^k$ będziemy na ogół oznaczać $\langle a_1, a_2, a_3, \ldots, a_k\rangle$. Będziemy mówić, że ciąg $\mathfrak{v} := \langle v_0, e_0, v_1, e_1, v_2, e_2, \ldots, v_n, e_n, v_{n+1}\rangle$ jest *A–trasą* jeśli $e_i = (v_i, v_{i+1}) \in A \vee e_i = (v_{i+1}, v_i) \in A$, gdzie $i = 0, 1, 2, \ldots, n$. Wierzchołek v_0 będziemy wówczas nazywać *początkiem* A–trasy $\mathfrak{v}$, a v_{n+1} jej *końcem*. Zbiór A–łuków $\{e_0, e_1, e_2, \ldots, e_n\}$ będziemy oznaczać symbolem $\mathcal{A}(\mathfrak{v})$, natomiast zbiór wierzchołków $\{v_0, v_1, v_2, \ldots, v_n, v_{n+1}\}$ symbolem $\mathcal{V}(\mathfrak{v})$. Przez długość A–trasy $\mathfrak{v}$ będziemy rozumieć moc zbioru $\mathcal{A}(\mathfrak{v})$ równą $n+1$. Jeśli A–łuki $e_0, e_1, e_2, \ldots, e_n$ są parami różne, to A–trasę $\mathfrak{v}$ będziemy nazywać *A–ścieżką*. Jeśli dodatkowo wierzchołki $v_0, v_1, v_2, \ldots, v_{n+1}$ są parami różne (być może z wyjątkiem pary v_0, v_{n+1}) to $\mathfrak{v}$ będziemy nazywać *A–drogą*. W przypadku $v_0 = v_{n+1}$, A–drogę $\mathfrak{v}$ będziemy nazywać *A–półcyklem*. Każdy ciąg postaci $\langle v_i, e_i, v_{i+1}, e_{i+1} \ldots, v_j, e_j, v_{j+1}\rangle$ będziemy nazywać *segmentem* A–trasy $\mathfrak{v}$, gdzie $0 \leqslant i \leqslant j \leqslant n$.

Dwa A–łuki będziemy nazywać *sąsiednimi*, jeśli koniec jednego z nich jest początkiem drugiego. Jeśli A–trasa jest zbudowana z A–łuków sąsiednich $\mathfrak{u} := \langle u_0, (u_0, u_1), u_1, (u_1, u_2), u_2, \ldots, u_{k-1}, (u_{k-1}, u_k), u_k\rangle$, to będziemy ją nazywać *skierowaną A–trasą* i oznaczać na ogół przez $u_0 \underset{A}{\rightarrow} u_1 \underset{A}{\rightarrow} \ldots \underset{A}{\rightarrow} v_n \underset{A}{\rightarrow} u_{k+1}$. Jeśli dodatkowo $\mathfrak{u}$ jest A–drogą oraz $u_0 = u_k$, to $\mathfrak{u}$ będziemy nazywać *zamkniętą skierowaną A–drogą* lub *skierowanym A–cyklem*.

W celu uproszczenia opisu posługując się terminem A–ścieżka, A–droga będziemy zakładać, że chodzi o ich wersję skierowaną. W pozostałych przypadkach będziemy dopisywać przymiotnik „*nieskierowany*".

Definicja 1.2. *Niech D będzie digrafem, $A \subseteq \mathcal{A}(D)$. Będziemy mówić, że łuk vu jest* A–skrótem, *jeśli $vu \in A$ oraz istnieje skierowana A–droga o początku v i końcu u, której długość $\geqslant 2$.*

Definicja 1.3. *Niech D będzie digrafem, $A \subseteq \mathcal{A}(D)$, $V_1, V_2 \subseteq \mathcal{V}(D)$. Wówczas:*

$$V_1 \underset{A}{\curvearrowright} V_2 := \{vu \in A : v \in V_1 \wedge u \in V_2\}. \tag{1.1}$$

Definicja 1.4. *Niech D będzie digrafem, $A \subseteq \mathcal{A}(D)$, $V \subseteq \mathcal{V}(D)$ oraz $|V| \geqslant 2$. Wówczas* gęstością *zbioru V w domknięciu zwrotno-przechodnim zbioru łuków A będziemy nazywać liczbę*

$$\rho_A(V) := \frac{|\{\{v, u\} : v, u \in V \wedge v \neq u \wedge (v \underset{A}{\rightarrow}^* u \vee u \underset{A}{\rightarrow}^* v)\}|}{\binom{|V|}{2}}. \tag{1.2}$$

Definicja 1.5. *Niech D będzie digrafem, $V \subseteq \mathcal{V}(D)$.* Podgrafem grafu skierowanego D indukowanym wierzchołkowo *przez zbiór V będziemy nazywać digraf:*

$$D_{|V} := \langle V, \{vu \in \mathcal{A}(D) : v, u \in V\}\rangle. \tag{1.3}$$

Definicja 1.6. *Niech D będzie digrafem, $V \subseteq \mathcal{V}(D)$. Będziemy mówić, że podgraf $D_{|V}$ jest* spójny, *jeśli dla dowolnej pary wierzchołków z V istnieje nieskierowana $D_{|V}$–droga łącząca te punkty.*

Definicja 1.7. *Niech D będzie digrafem. Będziemy mówić, że digraf D jest* acykliczny *lub krócej D jest DAG-iem, jeśli nie zawiera on skierowanych D–cykli.*

Definicja 1.8. *Niech D będzie DAG-iem. Przekształcenie różnowartościowe $\tau : \mathcal{V}(D) \to \{1, 2, \ldots, |\mathcal{V}(D)|\}$ będziemy nazywać* sortowaniem topologicznym (linearyzacją) *wtedy i tylko wtedy, gdy spełniona jest zależność:*

$$\underset{vu \in \mathcal{A}(D)}{\forall} \tau(v) < \tau(u). \tag{1.4}$$

Zbiór wszystkich sortowań topologicznych acyklicznego digrafu D będziemy oznaczać przez $TS(D)$. Ustalmy dowolną linearyzację $\tau \in TS(D)$. A–drogę $\mathfrak{v} := v_0 \underset{A}{\to} v_1 \underset{A}{\to} \ldots \underset{A}{\to} v_k$ będziemy nazywać *τ_A–łańcuchem*, jeśli linearyzacja τ przypisuje kolejnym wierzchołkom drogi $\mathfrak{v}$ kolejne liczby naturalne (precyzując, $\tau(v_{i+1}) = \tau(v_i)+1$ dla wszystkich $i = 0, 1, 2, \ldots, k-1$). Dodatkowo, τ_A–łańcuch będziemy nazywali *maksymalnym*, jeśli nie jest on segmentem żadnego innego τ_A–łańcucha. Bezpośrednio z definicji maksymalnego τ_A–łańcucha uzyskujemy, że poszczególne maksymalne τ_A–łańcuchy są wierzchołkowo rozłączne. Metrykę określoną przez linearyzację τ na wierzchołkach digrafu D daną zależnością $d_\tau(v, u) = |\tau(v) - \tau(u)|$, będziemy nazywać τ–metryką, gdzie $v, u \in \mathcal{V}(D)$. Dodatkowo przez *τ–rozpiętość D–łuku* vu będziemy rozumieć liczbę $d_\tau(v, u)$. Szczególne znaczenie dla naszych rozważań będą miały A–łuki o τ–rozpiętości 1, których zbiór będziemy oznaczać

$$\mathbf{1}_\tau^A := \{vu \in A : d_\tau(v, u) = 1\}. \tag{1.5}$$

Definicja 1.9. *Niech D będzie DAG-iem, τ linearyzacją D oraz niech $A \subseteq \mathcal{A}(D)$. Partycją digrafu D względem linearyzacji τ i rodziny A–łuków, będziemy nazywać partycję $\tau_A := \{\mathcal{V}(P) : P$ jest maksymalnym τ_A–łańcuchem$\}$.*

Tak zdefiniowane τ_A jest jednoznacznie określone, ponieważ τ_A–łańcuchy muszą się składać z wierzchołków numerowanych przez τ kolejnymi liczbami naturalnymi.

Definicja 1.10. *Niech D będzie DAG-iem, τ linearyzacją D oraz niech $V \subseteq \mathcal{V}(D)$. Będziemy mówić, że zbiór V jest* τ–spoisty*, jeśli istnieje liczba naturalna i, dla której $i \leqslant \tau(v) \leqslant i + |V| - 1$, dla wszystkich $v \in V$.*

1.2 Acykliczna partycja digrafu

Wprowadźmy pojęcia oraz oznaczenia niezbędne do zdefiniowania acyklicznych partycji (ang. *acyclic $\mathcal{P}^{(k)}$ partition* [13]). Podążając za Borowiecki i Mihók, ustalmy digraf D oraz partycję $\pi := (P_1, P_2, \ldots, P_k)$ zbioru wierzchołków $\mathcal{V}(D)$. Jeśli wówczas podgraf indukowany wierzchołkowo przez zbiór P_i digrafu D posiada własność $\mathcal{Q}_i$, dla każdego $i = 1, 2, \ldots, k$ to partycję π będziemy nazywać *$\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_k$–partycją*. Jeśli dodatkowo $\mathcal{Q}_1 = \mathcal{Q}_2 = \ldots \mathcal{Q}_k = \mathcal{Q}$, to partycję π będziemy nazywać *$\mathcal{Q}^k$–partycją*, a kolejność zbiorów w partycji π może zostać pominięta. Będziemy mówić, że partycja π' zbioru wierzchołków $\mathcal{V}(D)$ jest *$\mathcal{Q}^*$–partycją*, jeśli π' jest $\mathcal{Q}^{k'}$–partycją dla pewnej liczby naturalnej k'. Naturalnie zakładamy, że każda spośród własności $\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_k, \mathcal{Q}$ jest poprawnie określona dla wszystkich podgrafów indukowanych wierzchołkowo w digrafie D.

Definicja 1.11. *Niech D będzie digrafem, π partycją zbioru wierzchołków $\mathcal{V}(D)$.* Grafem partycji π nad D *będziemy nazywać digraf:*

$$\mathcal{G}(D,\pi) := \langle \pi, \{(P_1,P_2) \in \pi^2 \,:\, P_1 \neq P_2 \wedge \underset{v_1\in P_1, v_2 \in P_2}{\exists} v_1v_2 \in \mathcal{A}(D)\}\rangle. \tag{1.6}$$

Definicja 1.12. *Niech D będzie digrafem; $\mathcal{Q}$ własnością, która jest poprawnie określona na każdym indukowanym wierzchołkowo podgrafie D; π będzie $\mathcal{Q}^n$–partycją digrafu D. Będziemy mówić, że partycja π jest* acykliczną $\mathcal{Q}^{(n)}$–partycją digrafu D *(ang. acyclic $\mathcal{Q}^{(n)}$ partition [13]), jeśli digraf $\mathcal{G}(D,\pi)$ jest acykliczny.*

Partycję π będącą $\mathcal{Q}^*$–partycją będziemy nazywać *acykliczną $\mathcal{Q}^{(*)}$–partycją*, jeśli π jest acykliczną $\mathcal{Q}^{(|\pi|)}$–partycją.

Definicja 1.13. *Niech dany będzie digraf D oraz droga $\mathfrak{v} := v_1 \underset{D}{\rightarrow} v_2 \underset{D}{\rightarrow} \ldots \underset{D}{\rightarrow} v_k$. Będziemy mówić, że $\mathfrak{v}$ jest* drogą Hamiltona *wtedy i tylko wtedy, gdy spełniona jest zależność: $k = |\mathcal{V}(D)| = |\mathcal{V}(\mathfrak{v})|$.*

Definicja 1.14. *Niech D będzie digrafem, $V \subseteq \mathcal{V}(D)$, $A \subseteq \mathcal{A}(D)$. Będziemy mówić, że indukowany wierzchołkowo podgraf $D_{|V}$ ma własność $\mathcal{H}_A$ wtedy i tylko wtedy, gdy digraf $\langle \mathcal{V}(D), A\rangle_{|V}$ zawiera drogę Hamiltona.*

Ustalmy acykliczną $\mathcal{H}_A^{(*)}$–partycję π digrafu D, gdzie $A \subseteq \mathcal{A}(D)$. Wówczas każdemu zbiorowi $P \in \pi$, możemy przyporządkować jednoznacznie skierowaną A–drogę, która jest drogą Hamiltona w $D_{|P}$. Będziemy ją oznaczać przez $\mathfrak{h}^\pi(P)$. Dodatkowo jeśli $v \in P$, to drogę Hamiltona $\mathfrak{h}^\pi(P)$ będziemy oznaczać $\mathfrak{h}^\pi(v)$.

Twierdzenie 1.15. *Niech τ będzie linearyzacją digrafu D oraz $A \subseteq \mathcal{A}(D)$. Wówczas τ_A jest acykliczną $\mathcal{H}_A^{(|\tau_A|)}$–partycją grafu D.*

Dowód. Ponieważ każdy τ_A–łańcuch jest skierowaną A–drogą, więc w szczególności digraf indukowany wierzchołkowo przez zbiór wierzchołków każdego maksymalnego τ_A–łańcucha ma własność $\mathcal{H}_A$. Pokażemy, że digraf $\mathcal{G}(D,\tau_A)$ jest acykliczny. Uzasadnimy w tym celu, że dla każdego $\mathcal{G}(D,\tau_A)$–łuku (P_1,P_2) zachodzi nierówność $\tau(v_1) < \tau(v_2)$ po wszystkich $v_1 \in P_1$, $v_2 \in P_2$, co zakończy dowód. Niech (P_1,P_2) będzie $\mathcal{G}(D,\tau_A)$–łukiem, wówczas z Def. 1.11 możemy wskazać $v_1 \in P_1$, $v_2 \in P_2$, dla których v_1v_2 jest D–łukiem. Stąd z Def. 1.8 uzyskujemy, że $\tau(v_1) < \tau(v_2)$, co w konkluzji z warunkiem $P_1 \cap P_2 = \emptyset$ oraz z faktem, że τ przypisuje w P_1 i P_2 wierzchołkom kolejne liczby naturalne uzasadnia szukaną nierówność. □

Twierdzenie 1.16. *Niech π będzie acykliczną $\mathcal{H}_A^{(*)}$–partycją digrafu D, gdzie $A \subseteq \mathcal{A}(D)$. Wówczas istnieje linearyzacja τ digrafu D, dla której $|\tau_A| \leqslant |\pi|$.*

Dowód. Ustalmy linearyzację $\tau \in TS(\mathcal{G}(D,\pi))$. Przyporządkujmy każdemu wierzchołkowi v z $\mathcal{V}(D)$, liczbę naturalną oraz element partycji π, tak aby spełniona była zależność $\mathfrak{h}^\pi(P) = v_1^P \underset{A}{\rightarrow} v_2^P \underset{A}{\rightarrow} \ldots \underset{A}{\rightarrow} v_{|P|}^P$, gdzie $P \in \pi$. Zdefiniujmy linearyzację $\sigma \in TS(D)$, daną wzorem $\sigma(v_i^Q) = i + \sum\limits_{R\in\pi:\tau(R)<\tau(Q)} |R|$, gdzie $1 \leqslant i \leqslant |Q|$, $Q \in \pi$. Wówczas każda droga Hamiltona $\mathfrak{h}^\pi(P)$, będąca z definicji skierowaną A–drogą, jest σ_A–łańcuchem, gdzie $P \in \pi$. W celu zakończenia dowodu wystarczy jedynie zauważyć, że liczba maksymalnych σ_A–łańcuchów w σ_A partycji nie przekracza liczby σ_A–łańcuchów, których wierzchołki stanowią pokrycie zbioru $\mathcal{V}(D)$. □

Definicja 1.17. *Niech π będzie partycją zbioru wierzchołków X, $X' \subseteq X$.* Obcięciem *partycji π do zbioru wierzchołków X' będziemy nazywać partycję:*

$$\pi_{|X'} := \{P \cap X' : P \in \pi\} \setminus \{\emptyset\}. \tag{1.7}$$

1.3 Ukorzenione drzewa skierowane

Definicja 1.18. *Niech T będzie acyklicznym digrafem. Będziemy mówić, że T jest* ukorzenionym drzewem skierowanym*, jeśli istnieje wierzchołek $r \in \mathcal{V}(T)$ dla którego osiągalny jest każdy wierzchołek z $\mathcal{V}(T)$ oraz dla dowolnej pary dwóch różnych wierzchołków z $\mathcal{V}(T)$ istnieje dokładnie jedna nieskierowana T–droga łącząca te wierzchołki.*

Specyfika budowy rozważanych w tej pracy dowodów formalnych wymusza posługiwanie się specjalnym rodzajem drzew skierowanych – drzew skierowanych ukorzenionych o odwróconej orientacji (ang. *arborescent directed graph*).

Definicja 1.19. *Niech T będzie acyklicznym digrafem. Będziemy mówić, że T jest* dendroidem*, jeśli istnieje wierzchołek $r \in \mathcal{V}(T)$ osiągalny z każdego wierzchołka z $\mathcal{V}(T)$ oraz dla dowolnej pary dwóch różnych wierzchołków z $\mathcal{V}(T)$, istnieje dokładnie jedna nieskierowana T–droga łącząca te punkty.*

Podobnie jak w strukturze drzewa ukorzenionego wyróżniony wierzchołek r dendroidu T będziemy nazywać *korzeniem*, zaś wierzchołki nieposiadające poprzednika w T, *liśćmi*.

Definicja 1.20. Lasem dendroidów *będziemy nazywać acykliczny digraf, w którym każdy maksymalny spójny podgraf (w sensie zawierania) jest dendroidem.*

Definicja 1.21. *Niech L będzie lasem dendroidów, wówczas n–tym poziomem zagnieżdżenia lasu dendroidów L, oznaczanym dalej L^n, będziemy nazywać zbiór wierzchołków spełniający zależność: $v \in L^n$ wtedy i tylko wtedy, gdy istnieje skierowana L–droga długości n o początku v oraz końcu będącym korzeniem maksymalnego skierowanego drzewa w L, zawierającego v.*

Rozdział 2

Reprezentacje grafowe

W niniejszej rozprawie doktorskiej skupiamy się na dotychczas słabo rozwiniętych metodach poprawy czytelności, których wdrożenie jest niezależne od dotychczas prowadzonych prac nad poprawą wizualizacji dowodów w postaci HTML [86] oraz wprowadzanych udogodnieniach w systemie Mizar [48, 49, 50, 62, 63, 64].

Mizar [12, 33, 56, 85] jest systemem z logiką pierwszego rzędu określonym za pomocą aksjomatów teorii mnogości Tarskiego-Grothendiecka [75, 83, 84], wzmocnionym o kilka narzędzi umożliwiających formalizację matematyki w sposób bardziej zrozumiały dla człowieka. Równolegle z rozwojem tego systemu budowana jest baza komputerowo zweryfikowanej wiedzy matematycznej (Mizar Mathematical Library, w skrócie MML [8, 76]), zbudowana z poszczególnych artykułów, zawierających formalizację coraz większego zakresu matematyki. Doskonalenie bazy, jak również zwiększanie wygody użytkownika wymaga jednak jej ciągłej reorganizacji.

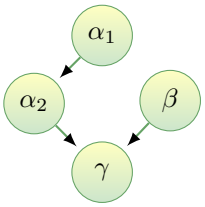
W pierwszej kolejności w celu precyzyjnego przedstawienia metod doskonalenia bazy i omówienia istoty związanych z nimi problemów, zaprezentujemy metodę opisywania zależności występujących w rozumowaniu, w oparciu o znane pojęcia grafowe. Taka interpretacja, bazująca na abstrakcyjnej definicji grafu dowodu umożliwi precyzyjne sformułowanie napotykanych problemów, zastosowanie algorytmów rozwiązujących znane problemy występujące w teorii grafów, jak również uprości zrozumienie przedstawionych rozwiązań dla nowych problemów z zakresu acyklicznych digrafów, które są wykorzystane w procesie poprawy czytelności. Uzyskane dzięki takiej abstrakcji uniezależnienie prowadzonych badań od konkretnego systemu weryfikującego, uprości również adaptację uzyskanych wyników do innych systemów umożliwiających formalizację dowodów matematycznych, np. HOL, Coq, Isabelle/Isar [1, 37, 92], jak również w dziedzinach niezwiązanych z formalizacją.

Badania nad abstrakcyjną definicją grafu dowodu, będące tematem tego rozdziału, zostały opublikowane w [70].

2.1 Zależności referencyjne

Rozumowania odpowiadające dowodom stwierdzeń stanowią główna cześć skryptów dowodowych zgromadzonych w bazie MML. Każde takie rozumowanie możemy w najprostszy sposób zilustrować jako ciąg występujących po sobie uzasadnionych faktów, między którymi przekazywane są pewne informacje. Możliwe jest też inne spojrzenie na rozumowanie, tj. interpretowanie zbioru wszystkich kroków jako zbioru wierzchołków digrafu, natomiast przekazywanych informacji jako łuków łączących te wierzchołki. Przyjmujemy, że istnieje łuk o początku w wierzchołku odpowiadającym

stwierdzeniu α oraz o końcu w wierzchołku odpowiadającym stwierdzeniu β wtedy i tylko wtedy, gdy fakt α jest bezpośrednio wykorzystywany w uzasadnieniu β. W celu uproszczenia rozważań, będziemy posługiwać się terminem *wierzchołek* α zamiast wierzchołek, związany z którym krok stwierdza α. Zauważmy, że liniowe uporządkowanie wierzchołków takiego digrafu zawiera na ogół dodatkowe zależności między wierzchołkami, które nie wynikają z przepływu informacji, a jedynie z wyboru sortowania topologicznego. Przykład digrafu, którego linearyzacja musi zawierać dodatkowe informacje, został przedstawiony na rys. 2.1. Niezależnie od wyboru linearyzacji, uporządkowanie takie musi określić porządek między wierzchołkami α_1, α_2, a β, który nie jest określony w tym digrafie. Zauważmy, że wybór porządku może mieć istotny wpływ na czytelność w przypadku nawet tak prostego grafu. Rozważmy w tym celu wszystkie możliwe linearyzacje tego grafu: β, α_1, α_2, γ; α_1, β, α_2, γ; α_1, α_2, β, γ,



Rysunek 2.1: Graf obrazujący przepływ informacji między wierzchołkami α_1, α_2, β oraz γ.

B: β ;	A1: α_1 ;	A1: α_1 ;
A1: α_1 ;	B: β ;	A2: α_2 by A1;
A2: α_2 by A1;	A2: α_2 by A1;	B: β ;
C: γ by A2, B;	C: γ by A2, B;	C: γ by A2, B;

Wydruk 2.2**:** Linearyzacje grafu przedstawionego na rys. 2.1 zapisane w języku Mizar.

przedstawione na wydruku 2.2. Pierwsza linearyzacja uwypukla wówczas spójność podrozumowania α_1, α_2, γ, ale równocześnie „rozrywa" łuk łączący wierzchołki β, γ. Owe rozerwanie wydłuża czas poszukiwania treści stwierdzenia związanego z identyfikatorem B, nazywanym dalej *etykietą*, które to stwierdzenie jest niezbędne do zrozumienia sposobu uzasadnienia γ. Poszukiwanie treści wiąże się bowiem z dodatkowym przeszukiwaniem kroków, które znajdują się w zlinearyzowanym rozumowaniu między wierzchołkami β i γ (kroki 2, 3). Dwie pozostałe linearyzacje „rozrywają" najdłuższe podrozumowanie α_1, α_2, β, aczkolwiek zmniejszają liczbę dodatkowo przeszukanych kroków w trakcie poszukiwania znaczenia etykiet.

Dodatkowo w przypadku trzeciej linearyzacji liczba ta przyjmuje najmniejszą możliwą wartość dla tego rozumowania. Jak łatwo można zauważyć, minimalizacja ta odpowiada znanemu problemowi NP-zupełnemu *Minimalnego Liniowego Uporządkowania Grafu Skierowanego* (ang. *Directed Optimal Linear Arrangement*, GT43 [2, 25]).

Znaczenie odległości między miejscem wprowadzenia przesłanki a miejscem jej użycia w rozumowaniu jest uwydatniane przez konstrukcję `then` (wydruk 2.3). Konstrukcja ta umożliwia bowiem niejawne przekazanie stwierdzenia związanego z poprzedzającym krokiem bez podania *explicite* etykiety wskazującej na to stwierdzenie. W szczególności, jeśli stwierdzenie s jest wykorzystywane jedynie w celu uzasadnienia kroku występującego w linearyzacji bezpośrednio po kroku stwierdzającym s, to

wykorzystanie konstrukcji `then` nie tylko umożliwia przekazanie przesłanki bez wykorzystania etykiety, ale skutkuje również tym, że etykieta związana ze stwierdzeniem s nie jest wykorzystywana w rozumowaniu, a więc może być usunięta z rozumowania. Stąd odpowiednie wykorzystanie konstrukcji `then` umożliwia minimalizowanie łącznej liczby wykorzystywanych etykiet. Oczywiście taka minimalizacja jest jedną z najbardziej naturalnych metod uczytelniania skryptów dowodowych. Zauważmy dodatkowo, że powstające w rozumowaniu ciągi kroków, z których każdy odwołuje się do poprzedzającego krok za pomocą `then`, wskazują czytelnikowi skryptu dowodowego w naturalny sposób liniowe fragmenty rozumowania.

```
    B: β ;                A1: α1 ;                    α1 ;
       α1 ;                  B: β ;          then A2: α2 ;
 then  α2 ;                     α2 by A1;             β ;
then C: γ by B;       then  C: γ by B;      then  C: γ by A2;
```

Wydruk 2.3: Linearyzacje grafu przedstawionego na rys. 2.1, zapisane w języku Mizar z wykorzystaniem konstrukcji `then`.

Ocena wyboru zbioru łuków, które w zlinearyzowanym rozumowaniu mają łączyć stwierdzenia występujące bezpośrednio po sobie lub dualnie – ocena wyboru zbioru łuków, które zostaną „rozerwane" w zlinearyzowanym rozumowaniu jest kwestią dyskusyjną. Niezależnie jednak od sformułowania zagadnienia, metoda optymalizacyjna poprawiająca czytelność musi wykorzystywać narzędzie, które na podstawie informacji zawartych w krokach istniejącego rozumowania – wierzchołkach grafu – zrekonstruuje rodzinę łuków tego grafu.

Kroki rozumowania, z których będą wyodrębniane informacje o rodzinie łuków, zasadniczo przyjmują postać dającą się przedstawić za pomocą następującego fragmentu gramatyki języka Mizar:

$$\begin{aligned}&\langle\textit{Label-Identifier}\rangle : \langle\textit{Formula-Expression}\rangle\\ &\qquad\qquad [\texttt{by}\ \langle\textit{Label-Identifier}\rangle\ \{\ \texttt{,}\ \langle\textit{Label-Identifier}\rangle\ \}^*]\ \texttt{;}, \end{aligned}\tag{2.1}$$

gdzie $\langle$*Formula-Expression*$\rangle$ odpowiada uzasadnianemu stwierdzeniem, a etykietom $\langle$*Label-Identifier*$\rangle$ (pełna gramatyka znajduje się w tab. D.1 str. 111). W celu zilustrowania budowy takich rozumowań rozważmy przykład z wydruku 2.4, zapożyczony z artykułu [74], gdzie etykiety `Lm3`, `Th9`, `Th14` są identyfikatorami uzasadnionych gdzie indziej w tym artykule faktów (ich sformułowania są zamieszczone w wydruku D.1).

Graf dowodu tego rozumowania chcielibyśmy budować na podstawie krotki złożonej z dwóch terminali:

$$\begin{aligned}\langle\textit{Reasoning-Step}\rangle &= \texttt{`}\langle\texttt{'}\ \langle\textit{Labels-Introduced}\rangle\ \texttt{`,'}\ \langle\textit{Labels-Used}\rangle\ \texttt{`}\rangle\texttt{'},\\ \langle\textit{Labels-Introduced}\rangle &= \texttt{`}\emptyset\texttt{'} \mid \texttt{`}\{\texttt{'}\ \langle\textit{Label-Identifier}\rangle\ \{\ \texttt{`,'}\ \langle\textit{Label-Identifier}\rangle\}^*\ \texttt{`}\}\texttt{'},\\ \langle\textit{Labels-Used}\rangle &= \texttt{`}\emptyset\texttt{'} \mid \texttt{`}\{\texttt{'}\ \langle\textit{Label-Identifier}\rangle\ \{\ \texttt{`,'}\ \langle\textit{Label-Identifier}\rangle\}^*\ \texttt{`}\}\texttt{'},\end{aligned}\tag{2.2}$$

przyporządkowanej każdemu krokowi w rozumowaniu. Jednak w istniejących skryptach nie jest wykluczone istnienie powielonych kroków oraz nadpisywanych etykiet, dlatego w celu odróżnienia poszczególnych kroków w rozumowaniu na podstawie jedynie krotek $\langle$*Reasoning-Step*$\rangle$ rozszerzmy tę krotkę o trzeci terminal $\langle$*Id*$\rangle$ wskazujący na unikalny identyfikator kroku, który w analizowanych przykładach będzie wyrażony za pomocą małych liter greckich.

```
α:  A1: 13 = 2 * 6 + 1;
β:  A2: not 2 divides 13 by A1, Th9;
γ:  A3: 13 = 3 * 4 + 1 ;
δ:  A4: not 3 divides 13 by A3, Th9;
ϵ:  A5: for n be Nat st 1 < n & n * n <= 13 & n is prime
        holds not n divides 13 by A2, A4, Lm3;
ζ:  A6: 13 is prime by A5, Th14;
```

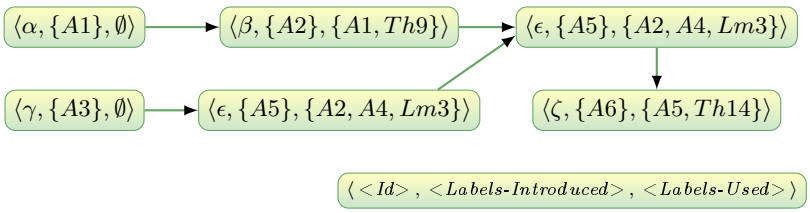
Wydruk 2.4: Dowód faktu, że 13 jest liczbą pierwszą.

Grafem dowodu D przy takiej interpretacji jest wówczas digraf, którego wierzchołkami są krotki:

$$<\textit{Reasoning-Step}> = `\langle` <\textit{Id}> `,` <\textit{Labels-Introduced}> `,` <\textit{Labels-Used}> `\rangle`, \quad (2.3)$$

zaś łukami pary $(\langle \alpha_1, L_1^I, L_1^U \rangle, \langle \alpha_2, L_2^I, L_2^U \rangle)$ takie, że $L_1^I \cap L_2^U \neq \emptyset$, które będziemy nazywać *łukami referencyjnymi* (zob. rys. 2.5).

W analizowanym przykładzie z wydruku 2.4, z krokiem δ związana jest krotka $\langle \delta, \{A4\}, \{A3, Th9\} \rangle$, z krokiem ϵ krotka $\langle \epsilon, \{A5\}, \{A2, A4, Lm3\} \rangle$, które to krotki są połączone łukiem referencyjnym, ponieważ $A4 \in \{A4\} \cap \{A2, A4, Lm3\}$ (zob. rys. 2.5).



Rysunek 2.5: Graf dowodu dla rozumowania przedstawionego na wydruku 2.4.

Digraf, w którym zbiór łuków pokrywa się ze zbiorem łuków referencyjnych jest często nazywany *grafem referencyjnym* [59]. Naturalnie graf referencji jest digrafem acyklicznym. Istnienie cyklu oznaczałoby, że uzasadnienie jakiegoś stwierdzenia s wykorzystuje przesłankę, która jest wywnioskowana przy użyciu uzasadnianego stwierdzenia s, co jest sprzeczne z założeniami dowodu matematycznego. Dopuszczalne są jednak półcykle – zamknięte D–drogi, które uniemożliwiają rozpatrywanie grafu referencji jako drzewa skierowanego (dodanie łuku $(\langle \alpha, \{A1\}, \emptyset \rangle, \langle \gamma, \{A3\}, \emptyset \rangle)$ do grafu przedstawionego na rys. 2.5 generuje półcykl, zob. również rys. 2.7).

W celu uproszczenia opisu grafu dowodu, będziemy posługiwać się terminem *krok* α zamiast *krok, którego terminal* <Id> *ma wartość* α oraz *wierzchołek* α, zamiast *wierzchołek, z którym związany terminal* <Id> *w grafie dowodu ma wartość* α. Dodatkowo, wykorzystując jednoznaczność wartości terminalu $<Id>$, będziemy oznaczać łuk łączący wierzchołki α_1, α_2 za pomocą pary uporządkowanej (α_1, α_2).

W dalszej części rozprawy, krotka $<\textit{Reasoning-Step}>$ reprezentująca informację zawartą w poszczególnych krokach rozumowania będzie powiększana o kolejne terminale. Modyfikacje te będą jednak zachowywać kolejność istniejących terminali, co umożliwia wykorzystanie pojęcia łuku referencyjnego mimo zmian w opisie krotki. Ostateczna postać krotki $<\textit{Reasoning-Step}>$ została sformułowana w Def. 2.1.

2.2 Łuki pierwotnie porządkujące

Przedstawione do tej pory zależności referencyjne nie opisują wszystkich możliwych zależności między krokami rozumowania. Konieczne jest bowiem uwzględnienie struktury formuł (tab. 2.1), co wykażemy w tym podrozdziale rozprawy. Formuły zapisane

$\bot$	`contradiction`
$\neg\alpha$	`not` α
$\alpha \wedge \beta$	α `&` β
$\alpha \vee \beta$	α `or` β
$\alpha \Longrightarrow \beta$	α `implies` β
$\alpha \Longleftrightarrow \beta$	α `iff` β
$\exists_x \alpha$	`ex` x `st` α
$\forall_x \alpha$	`for` x `holds` α
$\forall_{x\,:\,\alpha} \beta$	`for` x `st` α `holds` β

Tablica 2.1: Zapis spójników logicznych oraz kwantyfikatorów w języku Mizar.

w języku Mizar, mogą zawierać identyfikatory zmiennych, które wiążą się z kolejnym rodzajem zależności występującej między wierzchołkami grafu dowodu. Zależności te są następstwem faktu, że jeśli rozważana w pewnym kroku formuła ma zmienne wolne, to muszą ją poprzedzać w rozumowaniu kroki wprowadzające te zmienne. W celu uproszczenia rozumowania zakładamy, że rozważane skrypty dowodowe nie posiadają zarezerwowanych identyfikatorów zmiennych [33]. Zmienne, które zostały wprowadzone do rozumowania, a więc zmienne odpowiadające za wprowadzenie stałej, jak również kwantyfikatora uniwersalnego oraz egzystencjalnego do rozumowania będziemy nazywać *zmiennymi ustalonymi* lub *stałymi* (ang. *dummy variable* [39]).

Zauważmy, że pominięcie informacji opisującej porządek wyznaczony przez zależności wynikające z wykorzystania zmiennych ustalonych może prowadzić do powstania wierzchołków izolowanych w grafie dowodu. Natrafiamy na taką sytuację podczas analizy kroków rozumowania, które odpowiadają wyłącznie za wprowadzanie do rozumowania nowych identyfikatorów zmiennych ustalonych.

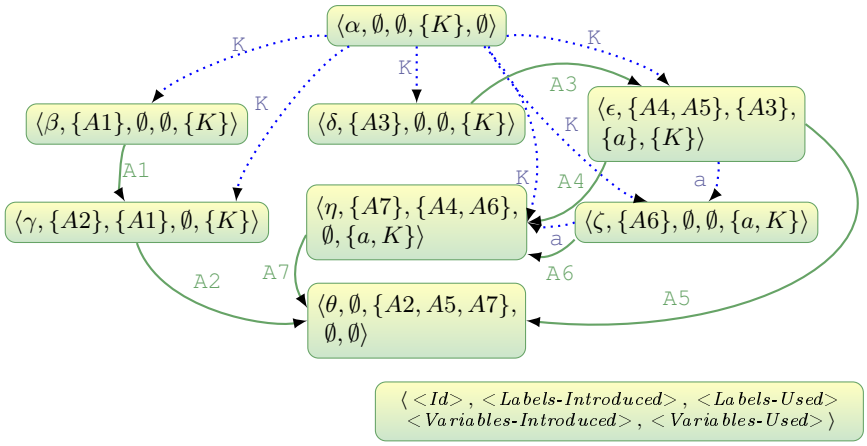
Narzucającym się rozwiązaniem tego problemu jest wzbogacenie opisu wierzchołka w grafie dowodu o terminale umożliwiające opis zależności, które wynikają z wprowadzania lub używania identyfikatorów zmiennych ustalonych (<*Variable-Identifier*>). Rozszerzmy w tym celu opis kroku rozumowania o dwa terminale: listę wprowadzonych w tym kroku do rozumowania nowych identyfikatorów zmiennych ustalonych (<*Variables-Introduced*>) oraz listę identyfikatorów zmiennych ustalonych, które są zmiennymi wolnymi wykorzystywanych w stwierdzeniu sformułowanym w tym kroku, ale nie zostały ustalone w tym kroku (<*Variables-Used*>).

$$\begin{aligned}
&<\textit{Reasoning-Step}> = \text{`}\langle\text{'} <\textit{Id}> \text{`,'} <\textit{Labels-Introduced}> \text{`,'} <\textit{Labels-Used}> \text{`,'} \\
&\qquad\qquad <\textit{Variables-Introduced}> \text{`,'} <\textit{Variables-Used}> \text{`}\rangle\text{'}, \\
&<\textit{Variables-Introduced}> = \text{`}\emptyset\text{'} \mid \text{`}\{\text{'} <\textit{Variable-Identifier}> \{\text{`,'} <\textit{Variable-Identifier}>\}^* \text{`}\}\text{'}, \\
&<\textit{Variables-Used}> = \text{`}\emptyset\text{'} \mid \text{`}\{\text{'} <\textit{Variable-Identifier}> \{\text{`,'} <\textit{Variable-Identifier}>\}^* \text{`}\}\text{'}.
\end{aligned} \tag{2.4}$$

Niech $\langle \alpha_1, L_1^I, L_1^U, V_1^I, V_1^U \rangle$, $\langle \alpha_2, L_2^I, L_2^U, V_2^I, V_2^U \rangle$ będą wierzchołkami grafu dowodu z interpretacją informacji zawartej w krokach rozumowania w postaci (2.4). Wówczas kroki te są połączone łukiem nowego rodzaju, jeśli $V_1^I \cap L_2^V \neq \emptyset$. Takie łuki będziemy nazwać *łukami pierwotnie porządkującymi*.

```
    for K be Field st 1.K <> -1.K holds K is Fanoian
      proof
α:      let K be Field;
β:      assume A1: 1.K <> -1.K;
γ:      A2: 1.K + 1.K <> 0.K by A1, VECTSP_1: 63;
δ:      assume A3: not K is Fanoian;
ϵ:      consider a be Element of K such that
          A4: a + a = 0.K and A5: a <> 0.K by A3, VECTSP_1: def 30;
ζ:      A6: a = a * 1.K by VECTSP_1: def 13;
η:      A7: 0.K = a * (1.K+1.K) by A4, A6, VECTSP_1: def 18;
θ:      thus contradiction by A2, A5, A7, VECTSP_1: 44;
      end;
```

Wydruk 2.6: Dowód faktu, że dowolne ciało posiadające dwa różne elementy $\mathbf{1}$, $-\mathbf{1}$, nie jest charakterystyki 2.



Rysunek 2.7: Graf dowodu dla rozumowania przedstawionego na wydruku 2.6.

W celu zilustrowania rodzin łuków referencyjnych oraz pierwotnie porządkujących w grafie dowodu, rozważmy rozumowanie przedstawione na wydruku 2.6, pochodzące z artykułu [67], którego struktura przy interpretacji informacji zawartej w wierzchołku w postaci (2.4) jest przedstawiona na rys. 2.7 (wykorzystujemy tam oznaczenia `1.K`, `-1.K`, `0.K` na elementy $\mathbf{1}$, $-\mathbf{1}$, $\mathbf{0}$ ciała `K` odpowiednio). Naturalnie graf dowodu nie zawiera łuków wielokrotnych, a łuki nie są etykietowane. Użyte opisy mają na celu jedynie ułatwienie identyfikacji łuków oraz wyróżnienie tych spośród nich, które są jednocześnie referencyjne i pierwotnie porządkujące. Opisy nad łukami przedstawiają odpowiednio wspólne elementy odpowiednich list <*Labels-Introduced*>, <*Labels-Used*>, w przypadku łuków referencyjnych oznaczanych ciągłą strzałką oraz list <*Variables-Introduced*>, <*Variables-Used*>, w przypadku łuków pierwotnie porządkujących oznaczanych wykropkowaną strzałką.

2.3 Szkielet rozumowania jako szczególny rodzaj informacji porządkujących

Niektóre dowody w systemie naturalnej dedukcji mają pewną ustaloną strukturę, np. żeby udowodnić twierdzenie uniwersalne zwykle najpierw wprowadza się zmienną reprezentującą dowolny element, a potem coś na temat tego elementu się dowodzi. Charakterystyczną cechą takiej struktury jest to, że pewne kroki muszą nastąpić przed innymi. Kolejność ta często nie jest powiązana z dotychczas wprowadzonymi rodzajami łuków. Dlatego potrzebne jest wprowadzenie tzw. łuków szkieletowych do naszego modelu grafu dowodu. *Szkielet* jest zatem częścią rozumowania, która ustala strukturę dowodu (szkielet rozumowania przedstawionego na wydruku 2.6 jest wyznaczony przez kroki α, β, δ, θ). Poszczególne jego elementy, nazywane dalej <*Skeleton-Step*>, odpowiadają za wprowadzenie do rozumowania uniwersalnego oraz egzystencjalnego kwantyfikatora, eliminację implikacji, w tym rozpoczęcie rozumowania nie wprost, oraz stwierdzanie tezy. Dodatkowo, do opisanych powyżej czterech rodzajów <*Skeleton-Step*> (por. tab. D.1) należy dodać kroki odpowiadające za rozpoczęcie rozumowania prowadzonego przez przypadki oraz kroki rozpoczynające poszczególne przypadki. Ponieważ każde rozumowanie prowadzone przez przypadki może zostać sformułowane w postaci ciągu zagnieżdżonych podrozumowań uzasadniających poszczególne przypadki (tak jak zostało to zrobione w przypadku rozumowania przedstawionego na wydruku 2.11). Będziemy zakładać, że rozumowania rozważane w tej rozprawie nie wykorzystują kroków umożliwiających prowadzenie rozumowań przez przypadki.

Rozważmy kilka możliwych konstrukcji szkieletu rozumowania uzasadniającego zawieranie się dwóch relacji R, S, gdzie predykat c= opisuje relację zawierania $\subseteq$, natomiast wyrażenie [x,y] odpowiada parze uporządkowanej (x, y). Przytoczone

a) R c= S
```
proof
  let x be set;
  assume x in R;
  <Reasoning>
  thus x in S;
end;
```

b) R c= S
```
proof
  let x be set;
  assume x in R;
  assume not x in S;
  <Reasoning>
  thus contradiction;
end;
```

c) R c= S
```
proof
  let x, y be set;
  assume [x,y] in R;
  <Reasoning>
  thus [x,y] in S;
end;
```

d) R c= S
```
proof
  let x be set;
  assume ex y be set st
    [x,y] in R & not [x,y] in S;
  <Reasoning>
  thus contradiction;
end ;
```

e) R c= S
```
proof
  assume ex x, y be set st
    [x,y] in R & not [x,y] in S;
  <Reasoning>
  thus contradiction;
end;
```

Rysunek 2.8: Wybrane konstrukcje szkieletu rozumowania uzasadniającego zawieranie R$\subseteq$S.

warianty szkieletów rozumowania (rys. 2.8) prezentują wówczas wybrane poprawnie zbudowane szkielety dowodów uzasadniających formułę R$\subseteq$S. Dwa pierwsze szkielety *a*), *b*) odpowiadają strukturom dowodu wykorzystującym jedynie definicję predykatu zawierania zbiorów. Aby wykorzystać własności, że R jest relacją (S nie musi być

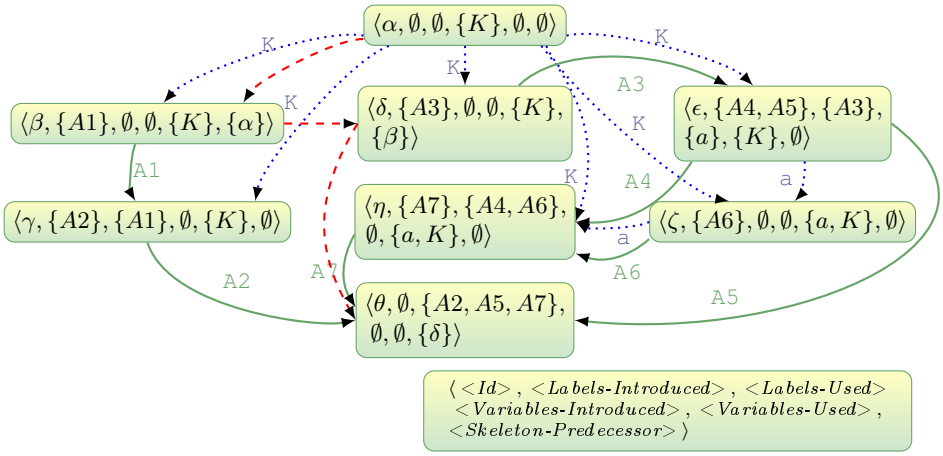
relacją), niezbędne jest umieszczenie w środowisku redefinicji predykatu zawierania, opisującej przypadek zawierania się zbioru będącego relacją w dowolnym zbiorze, wraz z niezbędnymi notacjami, umożliwiającymi sformułowanie szkieletu w trzech pozostałych postaciach (c), d), e)). Wprowadzenie tej redefinicji oraz odpowiednich notacji do środowiska skryptu dowodowego uniemożliwia jednak dostęp do rozwinięcia definicyjnego predykatu zawierania sformułowanego dla zbiorów, odkąd weryfikator systemu Mizar na podstawie informacji zawartych w środowisku oraz skrypcie dowodowym może wywnioskować, że `R` jest relacją. Tym samym szkielety zaproponowane w przypadkach a), b) stają się niepoprawne dla weryfikatora tego systemu.

Szczególnym zbiorem informacji zawartym w środowisku, które weryfikator systemu Mizar wykorzystuje w procesie wnioskowania są *rejestracje* [33]. Jest to lista stwierdzeń, które w danym środowisku są uznane za oczywiste i są domyślnie wykorzystywane m.in. do uzasadnienia każdego kroku. Taką rejestracją może być np. stwierdzenie, że zbiór pusty jest relacją, która wpływa na szkielet uzasadnienia zawierania `0` `c=` `S`.

Wyznaczenie kolejności kroków rozumowania <*Skeleton-Step*> w ogólnym przypadku wymaga więc bardzo szczegółowej analizy zasobów środowiska skryptów dowodowych lub wykorzystania modułu *Reasoner*, który weryfikuje poprawność zbudowanego szkieletu. Moduł ten musiałby jednak sprawdzić poprawność wszystkich możliwych linearyzacji podgrafu dowodu indukowanego przez kroki rodzaju <*Skeleton-Step*>, których liczba może być bliska liczbie permutacji zbioru tych kroków. Rozsądnym wydaje się zatem przechowywanie informacji o kolejności kroków <*Skeleton-Step*> w opisie rozumowania, za pomocą dodatkowego terminalu:

$$<\textit{Skeleton-Predecessor}> \;=\; \text{`}\emptyset\text{`} \;\mid\; \text{`}\{\text{`} <\textit{Id}> \text{`}\}\text{`}. \tag{2.5}$$

Terminal <*Skeleton-Predecessor*> w wierzchołku α_2 ma wartość $\{\alpha_1\}$ wtedy i tylko wtedy, gdy oba kroki α_1, α_2 są rodzaju <*Skeleton-Step*>, krok α_1 jest poprzednikiem kroku α_2 w rozumowaniu oraz nie istnieje krok w zlinearyzowanym rozumowaniu rodzaju <*Skeleton-Step*> znajdujący się między α_1 oraz α_2.



Rysunek 2.9: Graf dowodu przedstawiony na rys. 2.7 wzbogacony o łuki szkieletowe oznaczone strzałkami kreskowanymi.

Łukiem szkieletowym będziemy nazywać parę (α_1, α_2) wtedy i tylko wtedy, gdy α_1 jest elementem zbioru, który jest wartością terminalu <*Skeleton-Predecessor*> w krotce opisującej krok α_2 (w przykładzie rys. 2.7, są to trzy łuki (α, β), (β, δ), (δ, θ), zob. rys. 2.9). Dodatkowo wierzchołki, które odpowiadają krokom rodzaju <*Skeleton-Step*>, będziemy często nazywać *szkieletowymi*.

Łuki szkieletowe, podobnie jak łuki pierwotnie porządkujące, określają warunki narzucone na porządek linearyzacji grafu dowodu. Ze względu na podobny charakter informacji związany z tymi łukami, definiujemy nową rodzinę łuków w grafie dowodu, powstałą z połączenia dwóch rodzin: rodziny łuków szkieletowych oraz rodziny łuków pierwotnie porządkujących, której elementy będziemy nazywać *łukami porządkującymi*. Podobnie definiujemy rodzinę łuków w grafie dowodu powstałą z połączenia dwóch rodzin: rodziny łuków referencyjnych oraz rodziny łuków porządkujących, której elementy będziemy nazywać *łukami argumentującymi*.

2.4 Pozostałe łuki porządkujące

Ostatnim rodzajem informacji, który można opisać za pomocą łuków porządkujących, są zależności wynikające z możliwości nadpisywania wprowadzonych identyfikatorów zmiennych ustalonych oraz etykiet. Naturalnie wystarczającym rozwiązaniem tego problemu jest rozszerzenie rodziny łuków porządkujących o łuki opisujące zależność między każdym wierzchołkiem, w którym został wykorzystany identyfikator przed nadpisaniem, a wierzchołkiem w którym ten identyfikator został nadpisany. Takie rozwiązanie nie generuje skierowanych cykli w powstałym grafie dowodu, aczkolwiek ze względu na istnienie w systemie Mizar programów pomocniczych dokonujących α–konwersji [58], które eliminują problemy nadpisywania, będziemy zawsze zakładać, że w rozważanych rozumowaniach nie występują problemy związane z nadpisywaniem identyfikatorów.

2.5 Metakrawędzie

Przedstawiane powyżej przykłady służące opisaniu zależności występujących w grafie dowodu, posiadały zawsze „płaskie", czy też „jednopoziome" rozumowanie. W celu zilustrowania struktury dowodów, w których rozumowanie jest prowadzone na kliku poziomach zagnieżdżenia, rozważmy jeden z możliwych dowodów „reguły pijącego" [70, 97] zapisany w języku Mizar (wydruk 2.11). W strukturze tego rozumowania możemy odnaleźć trzy poziomy zagnieżdżenia: *zerowy*, zbudowany wyłącznie z kroku α; *pierwszy*, z kroków β, ι, ν oraz *drugi*, zawierający pozostałe kroki (rys. 2.12).

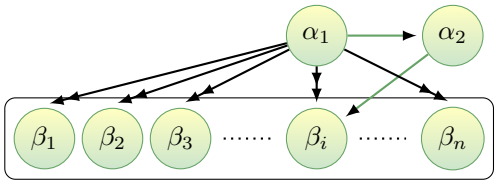
Zauważmy, że żadna ze zgromadzonych dotąd informacji w grafie dowodu nie charakteryzuje związku między wierzchołkami, które znajdują się na tym samym poziomie zagnieżdżenia w rozumowaniu. Dodatkowo, nie odnajdziemy żadnych zależności miedzy wierzchołkiem, którego uzasadnienie opiera się o zagnieżdżone podrozumowanie, a wierzchołkami odpowiadającymi krokom tego podrozumowania (np. między krokiem β, a którymkolwiek z pośród γ, δ, ϵ, ζ, η, θ).

Wykorzystując pojęcia związane z drzewem skierowanym, możemy uniknąć utraty tego rodzaju informacji zawartych w rozumowaniu, wzbogacając krotkę opisującą kroki dowodu. Naturalnym wydaje się przechowywanie informacji o tym aspekcie struktury dowodu w postaci drzewa skierowanego, w którym uzasadnione stwierdzenie odpowiada korzeniowi drzewa, natomiast kolejne poziomy zagnieżdżenia odpowiadają „warstwom" drzewa, zbudowanym z wierzchołków o tej samej odległości od korzenia. Precyzując, zbiór następników wierzchołka v odpowiada zbiorowi kro-

ków podrozumowania, które stanowi zagnieżdżone podrozumowanie uzasadniające poprawność stwierdzenia sformułowanego w kroku v.

Skonstruowana w ten sposób nowa rodzina łuków ma zupełnie inne znaczenie w strukturze grafu dowodu niż rodziny łuków referencyjnych oraz łuków porządkujących. Opisuje ona wyłącznie zależności między kolejnymi poziomami zagnieżdżenia. Dla odróżnienia tej relacji w strukturze grafu dowodu, łuki tego drzewa będziemy nazywali *metakrawędziami*.

Analizując własności, którymi odznacza się powstała struktura łuków oraz metakrawędzi, możemy zauważyć, że wybór orientacji metakrawędzi od korzenia do liści prowadzi do komplikacji w opisie interesujących nas własności. W celu zilustrowania tego problemu, rozważmy sytuację niedozwoloną w dowodzie matematycznym (rys. 2.10), w której stwierdzenie α_2 wynikające z α_1 jest wykorzystywane w uzasadnieniu np. formuły β_i należącej do podrozumowania, zbudowanego ze stwierdzeń $\beta_1, \beta_2, \ldots, \beta_n$, stanowiącego zagnieżdżone uzasadnienie formuły α_1. Chcielibyśmy,



Rysunek 2.10: Przykład niepoprawnego rozumowania, które nie generuje cyklu skierowanego, jeżeli metakrawędzie prowadzą od korzenia drzewa do jego liści. Metakrawędzie są tutaj reprezentowane przez strzałki postaci $\twoheadrightarrow$.

aby taka sytuacja, w której wniosek z kroku α_1 został wykorzystany w uzasadnieniu tego kroku, wiązała się z powstaniem cyklu skierowanego, tak jak miało to miejsce w grafie referencji. Możemy uzyskać ten warunek, zmieniając orientację metakrawędzi, tak aby prowadziły one od liści do korzeni. Nowa rodzina metakrawędzi wraz ze zbiorem wierzchołków tworzy wówczas skierowane drzewo o odwróconej orientacji (ang. *arborescent directed graph*).

Informacje o rodzinie metakrawędzi będziemy przechowywać w opisie kroku rozumowania <*Reasonings-Step*> w oparciu o dodatkowy terminal

$$<\textit{Justified-Statement}> \;=\; `\emptyset` \;\mid\; `\{` <\textit{Id}> `\}`. \tag{2.6}$$

Terminal <*Justified-Statement*> w wierzchołku α_2 przyjmuje wartość $\{\alpha_1\}$ wtedy i tylko wtedy, gdy uzasadnienie kroku α_1 opiera się o zagnieżdżone podrozumowanie zawierające krok α_2. W rozważanym przykładzie (wydruk 2.11), wartość terminala <*Justified-Statement*> związanego z wierzchołkiem α jest zbiorem pustym, w wierzchołkach β, ι, ν ma wartość $\{\alpha\}$, a w wierzchołkach γ, δ, ϵ, ζ, η, θ oraz κ, λ, μ ma odpowiednio wartość $\{\beta\}$ oraz $\{\iota\}$.

Opisując strukturę dowodów formalnych, niejednokrotnie rozważamy strukturę konkretnego rozumowania z pominięciem zawartych w nim zagnieżdżonych podrozumowań. Wynika to z faktu, że proces odtworzenia koncepcji analizowanego dowodu możemy naturalnie podzielić ze względu na kolejne „jednopoziomowe" rozumowania, które stanowią zagnieżdżone uzasadnienia pojedynczych kroków dowodu lub „zerowy poziom" rozumowania. Możemy bowiem najpierw rozważyć zerowy poziom dowodu, z pominięciem zagnieżdżonych podrozumowań, które stanowią uzasadnienia wybranych kroków tego dowodu. W drugim „kroku" możemy wówczas analizować wyłącznie podrozumowania zawarte w pominiętych uzasadnieniach. Naturalnie, jeżeli w którymś

podrozumowaniu istniały kroki, których uzasadnienie opierało się o dowód, to ich analiza zostanie wykonana w kolejnym „kroku", itd. Jednopoziomowe rozumowania odpowiadają więc rodzinom kroków dowodu, które posiadają wspólną wartość terminalu <*Justified-Statement*>. W dalszej części rozprawy, jednopoziomowe rozumowanie będziemy nazywać *rozumowaniami pierwotnymi*. W rodzinie rozumowań pierwotnych określamy relację bycia rozumowaniem nadrzędnym. Rozumowanie pierwotne R_1 będziemy nazywać *bezpośrednio nadrzędnym* względem rozumowania pierwotnego R_2, jeśli R_2 jest zagnieżdżonym podrozumowaniem uzasadniającym wybrany krok z R_1 (rozumowanie pierwotne zbudowane z kroku α, przedstawione na rys. 2.12 jest bezpośrednio nadrzędne względem rozumowania pierwotnego zbudowanego z wierzchołków β, ι, ν). Relacją nadrzędności jest wówczas domknięcie zwrotno–przechodnie relacji bezpośredniej nadrzędności. Podobnie rozumowanie pierwotne R_2 będziemy nazywać *podrzędnym* względem rozumowania pierwotnego R_1, jeśli R_1 jest nadrzędne względem R_2.

```
α:  ex x st (P[x] implies for y holds P[y])
      proof
β:      A1: (ex x st not P[x]) implies thesis
          proof
γ:          assume A2: ex x st not P[x] ;
δ:          consider a such that A3: not P[a] by A2;
ϵ:          take a;
ζ:          assume A4: P[a];
η:          A5: contradiction by A3, A4;
θ:          thus for y holds P[y] by A5;
          end;
ι:      A6: (for x holds P[x]) implies thesis
          proof
κ:          assume A7: for x holds P[x];
λ:          take b=the set;
μ:          thus P[b] implies for y holds P[y] by A7;
          end;
ν:      thus thesis by A1, A6;
      end;
```

Wydruk 2.11: Dowód „reguły pijącego", która stwierdza, że w każdej grupie ludzi można wskazać jedną osobę taką, że jeżeli ta osoba pije, to wszyscy piją.

2.6 Graf dowodu

Przedstawiona powyżej konstrukcja opisu informacji zawartej w poszczególnych krokach rozumowania dostarcza dostatecznej liczby zależności występujących między krokami rozumowania, które umożliwiają odtworzenie grafu dowodu (badania wykorzystujące przedstawiony opis informacji zostały przedstawione w sekcji 3.3.3). Ostateczna forma opisu informacji zawartej w krokach rozumowania ma postać:

$$\begin{aligned} <\textit{Reasoning-Step}> = \text{`}\langle\text{'} <\textit{Id}> \text{`,'} <\textit{Labels-Introduced}> \text{`,'} <\textit{Labels-Used}> \text{`,'} \\ <\textit{Variables-Introduced}> \text{`,'} <\textit{Variables-Used}> \text{`,'} \\ <\textit{Skeleton-Predecessor}> \text{`,'} <\textit{Justified-Statement}> \text{`}\rangle\text{'}. \end{aligned} \tag{2.7}$$

Naturalnie krotkę tę możemy zastosować do opisu informacji zawartej w każdym poprawnie zbudowanym kroku dowodu w systemie Mizar (tab. D.1). Na jej podstawie formułujemy definicję grafu dowodu.

Definicja 2.1. *Niech D będzie rozumowaniem zapisanym w uproszczonej składni systemu Mizar (tab. D.1), poprawnym dla weryfikatora tego systemu, V zbiorem krotek* <Reasoning-Step> *opisujących kroki rozumowania D, zaś $A, M \subseteq V \times V$ podzbiorami zbioru wszystkich par krotek. Uporządkowaną trójkę $\mathfrak{P} = \langle V, A, M\rangle$ będziemy nazywać* grafem dowodu D *wtedy i tylko wtedy, gdy dla dowolnych dwóch krotek $v_1 = \langle \alpha_1, L_1^I, L_1^U, V_1^I, V_1^U, SP_1, JS_1\rangle$, $v_2 = \langle \alpha_2, L_2^I, L_2^U, V_2^I, V_2^U, SP_2, JS_2\rangle$ ze zbioru V, spełnione są warunki:*

1. $(v_1, v_2) \in A \iff \big(L_1^I \cap L_2^U \neq \emptyset \vee V_1^I \cap V_2^U \neq \emptyset \vee \alpha_1 \in SP_2\big)$,
2. $(v_1, v_2) \in M \iff \alpha_1 \in JS_2$.

W grafie dowodu $\mathfrak{P} = \langle V, A, M\rangle$ łuki ze zbioru A odpowiadają łukom argumentującym, zaś łuki ze zbioru M metakrawędziom.

Definicja 2.2. *Uporządkowaną trójkę $\mathfrak{P} = \langle V, A, M\rangle$ będziemy nazywać* konstruktywnym *grafem dowodu, jeśli istnieje poprawne dla weryfikatora systemu Mizar rozumowanie D zapisane w uproszczonej składni tego systemu (tab. D.1), dla którego $\mathfrak{P}$ jest grafem dowodu D.*

Wprowadźmy oznaczenia pomocnicze $\mathfrak{D}_\mathfrak{P} = \langle V, A\rangle$, $\mathfrak{Meta}_\mathfrak{P} = \langle V, M\rangle$. Z konstrukcji grafu dowodu, uzyskujemy dwa poniższe stwierdzenia.

Stwierdzenie 2.3. *Niech $\mathfrak{P} = \langle V, A, M\rangle$ będzie konstruktywnym grafem dowodu. Para $\langle V, A \cup M\rangle$ jest acyklicznym digrafem, w którym dopuszczalne są półcykle, natomiast para $\langle V, M\rangle$ jest lasem dendroidów.*

Stwierdzenie 2.4. *Jeżeli (v, u) jest $\mathfrak{D}_\mathfrak{P}$–łukiem konstruktywnego grafu dowodu $\mathfrak{P}$, to każdy następnik wierzchołka v w $\mathfrak{Meta}_\mathfrak{P}$, jest osiągalny z u w $\mathfrak{Meta}_\mathfrak{P}$ i jest różny od u.*

Dowód. Niech (u, v) będzie $\mathfrak{D}_\mathfrak{P}$–łukiem konstruktywnego grafu dowodu $\mathfrak{P}$. Oznaczmy przez p_u, p_v poziomy zagnieżdżenia w lesie $\mathfrak{Meta}_\mathfrak{P}$, dla których $u \in \mathfrak{Meta}_\mathfrak{P}^{p_u}$, $v \in \mathfrak{Meta}_\mathfrak{P}^{p_v}$ oraz niech R_u, R_v będą rozumowaniami pierwotnymi zawierającymi kroki u, v odpowiednio. Zauważmy, że niezależnie, czy w u zostało uzasadnione stwierdzenie, czy też zostały wprowadzone nowe zmienne ustalone do rozumowania, żadna z tych informacji nie może być wykorzystana w rozumowaniu pierwotnym, które jest jednocześnie nadrzędne względem R_u i różnym od R_u. Stąd $p_v \geqslant p_u$. Przypadek $p_u = 0$ jest oczywisty, ponieważ u jako korzeń nie posiada żadnego następnika w $\mathfrak{Meta}_\mathfrak{P}$. Załóżmy więc, że $p_u > 0$ oraz niech $u \underset{\mathfrak{Meta}_\mathfrak{P}}{\rightarrow} w$. Rozumowanie R_w jest wówczas nadrzędne względem R_v oraz $R_w \neq R_v$, gdzie R_w oznacza rozumowanie pierwotne zawierające wierzchołek w. Stąd ostatecznie $v \underset{\mathfrak{Meta}_\mathfrak{P}}{\rightarrow}{}^* w$ oraz $v \neq w$. □

Własności konstruktywnego grafu dowodu sformułowane w Stw. 2.3, 2.4, charakteryzują podstawowe oczekiwania, stawiane przed poprawnie zbudowanym dowodem w systemie naturalnej dedukcji G. Gentzena, S. Jaśkowskiego [30, 40, 66]. Własności te umożliwiają zdefiniowanie struktury abstrakcyjnego grafu dowodu, wyrażonej wyłącznie w terminach digrafów. Pokażemy w twierdzeniu 2.17, że każdy abstrakcyjny graf dowodu $\mathfrak{P}$, w którym rodzina wierzchołków szkieletowych spełnia dodatkową zależność wynikającą z syntaktyki systemu Mizar, jest konstruktywny.

Definicja 2.5. Abstrakcyjnym grafem dowodu $\mathfrak{P} = \langle \mathcal{V}(\mathfrak{P}), \mathcal{A}(\mathfrak{P}), \mathcal{M}(\mathfrak{P}) \rangle$ *nazywamy uporządkowaną trójkę zbiorów, niepustego zbioru* $\mathcal{V}(\mathfrak{P})$ *oraz dwóch podzbiorów* $\mathcal{A}(\mathfrak{P}), \mathcal{M}(\mathfrak{P}) \subseteq \mathcal{V}(\mathfrak{P}) \times \mathcal{V}(\mathfrak{P})$ *zbioru wszystkich uporządkowanych par, spełniającą zależności:*

1. *digraf* $\mathfrak{Meta}_{\mathfrak{P}} := \langle \mathcal{V}(\mathfrak{P}), \mathcal{M}(\mathfrak{P}) \rangle$ *jest lasem dendroidów,*
2. *każdy łuk* (u, v) *w digrafie dowodu* $\mathfrak{D}_{\mathfrak{P}} := \langle \mathcal{V}(\mathfrak{P}), \mathcal{A}(\mathfrak{P}) \rangle$ *spełnia warunek: każdy następnik* u *w lesie* $\mathfrak{Meta}_{\mathfrak{P}}$ *jest osiągalny z* v *w lesie* $\mathfrak{Meta}_{\mathfrak{P}}$ *i jednocześnie różny od* v*,*
3. *digraf* $\mathfrak{G}_{\mathfrak{P}} := \langle \mathcal{V}(\mathfrak{P}), \mathcal{A}(\mathfrak{P}) \cup \mathcal{M}(\mathfrak{P}) \rangle$ *jest acykliczny.*

Z warunku Def. 2.5.2° uzyskujemy natychmiast, że zbiory $\mathcal{A}(\mathfrak{P})$, $\mathcal{M}(\mathfrak{P})$ są rozłączne. Dodatkowo ze stwierdzeń 2.3, 2.4, uzyskujemy następujący lemat.

Lemat 2.6. *Konstruktywny graf dowodu jest abstrakcyjnym grafem dowodu.*

Niech $v_1 = \langle \alpha_1, L_1^I, L_1^U, V_1^I, V_1^U, SP_1, JS_1 \rangle$, $v_2 = \langle \alpha_2, L_2^I, L_2^U, V_2^I, V_2^U, SP_2, JS_2 \rangle$ oraz $v_1, v_2 \in V$. W rodzinie $\mathcal{A}(\mathfrak{P})$–łuków konstruktywnego grafu dowodu $\mathfrak{P}$ zostały wyróżnione podczas konstrukcji cztery rodzaje łuków o szczególnym znaczeniu:

- (v_1, v_2) jest łukiem referencyjnym wtedy i tylko wtedy, gdy $L_1^I \cap L_2^U \neq \emptyset$,
- (v_1, v_2) jest łukiem pierwotnie porządkującym wtedy i tylko wtedy, gdy $V_1^I \cap V_2^U \neq \emptyset$,
- (v_1, v_2) jest łukiem szkieletowym wtedy i tylko wtedy, gdy $\alpha_1 \in JS_2$,
- (v_1, v_2) jest łukiem porządkującym wtedy i tylko wtedy, gdy (v_1, v_2) jest pierwotnie porządkującym lub szkieletowym,

Zdefiniujmy teraz odpowiedniki tych rodzin w digrafie $\mathfrak{D}_{\mathfrak{P}}$ abstrakcyjnego grafu dowodu $\mathfrak{P}$, nadając im analogiczne nazwy.

Definicja 2.7. *Niech* $\mathfrak{P}$ *będzie abstrakcyjnym grafem dowodu. W rodzinie* $\mathcal{A}(\mathfrak{P})$*–łuków wyróżniamy podrodziny:* $\mathcal{R}ef_{\mathfrak{P}}$*,* $\mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}}$*,* $\mathcal{S}kel_{\mathfrak{P}}$*,* $\mathcal{O}rd_{\mathfrak{P}}$ *spełniające warunki* $\mathcal{R}ef_{\mathfrak{P}} \cup \mathcal{O}rd_{\mathfrak{P}} = \mathcal{A}(\mathfrak{P})$*,* $\mathcal{O}rd_{\mathfrak{P}} = \mathcal{S}kel_{\mathfrak{P}} \cup \mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}}$*. Elementy rodzin* $\mathcal{R}ef_{\mathfrak{P}}$*,* $\mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}}$*,* $\mathcal{S}kel_{\mathfrak{P}}$*,* $\mathcal{O}rd_{\mathfrak{P}}$ *będziemy nazywać odpowiednio łukami* referencyjnymi*,* pierwotnie porządkującymi*,* szkieletowymi*,* porządkującymi*.*

Wprowadźmy oznaczenia dwóch grafów: grafu referencyjnego $\mathfrak{R}_{\mathfrak{P}} = \langle \mathcal{V}(\mathfrak{P}), \mathcal{R}ef_{\mathfrak{P}} \rangle$ i grafu pierwotnie porządkującego $\mathfrak{O}_{\mathfrak{P}} = \langle \mathcal{V}(\mathfrak{P}), \mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}} \rangle$, a także dla zbioru wierzchołków szkieletowych $\mathfrak{S}_{\mathfrak{P}} = \{v \in \mathcal{V}(\mathfrak{P}) : \exists_{w \in \mathcal{V}(\mathfrak{P})} (v, w) \in \mathcal{S}kel_{\mathfrak{P}} \vee (w, v) \in \mathcal{S}kel_{\mathfrak{P}}\}$.

Wykorzystanie konstrukcji `then` w procesie poprawy czytelności skutkuje powstaniem jeszcze jednej ważnej podrodziny $\mathcal{R}ef_{\mathfrak{P}}$–łuków. Z syntaktyki języka Mizar wynika bowiem, że nie każda informacja związana $\mathcal{R}ef_{\mathfrak{P}}$–łukiem (α_1, α_2) może zostać przekazana między wierzchołkami α_1, a α_2 za pomocą konstrukcji `then`, nawet jeśli kroki te występują bezpośrednio po sobie w zlinearyzowanym dowodzie. W rozumowaniu mogą bowiem istnieć kroki, które nie mogą wykorzystywać konstrukcji `then` jako metody pobrania informacji z poprzedzającego kroku w żadnej linearyzacji (np. krok uzasadniony w oparciu o schemat) oraz kroki, z którymi związana informacja nie może zostać przekazana do kolejnego kroku w żadnej linearyzacji za pomocą konstrukcji `then` (np. krok wprowadzający zmienne do rozumowania). Wybrane łuki referencyjne abstrakcyjnego grafu dowodu $\mathfrak{P}$, które mogą być zastąpione konstrukcją `then` będziemy nazywali `then`*–łukami*, a przez $\texttt{then}_{\mathfrak{P}}$ będziemy oznaczać rodzinę wszystkich `then`–łuków.

W dalszych rozważaniach w celu uniezależnienia badań nad strukturą abstrakcyjnych grafów dowodu od kontekstu systemu Mizar, rozważając rodzinę $\texttt{then}_{\mathfrak{P}}$–łuków, będziemy o niej zakładać jedynie, że $\texttt{then}_{\mathfrak{P}} \subseteq \mathcal{R}ef_{\mathfrak{P}}$. Wpływ kontekstu systemu Mizar będzie rozważany jedynie w podrozdziale 4.3. Rozważanie ogólnego przypadku, jak również ograniczonej instancji, w której $\texttt{then}_{\mathfrak{P}}$ jest zbiorem jedynie tych łuków referencyjnych, których początki nie są początkami żadnych łuków porządkujących:

$$\texttt{then}_{\mathfrak{P}} \subseteq \{vu : vu \in \mathcal{R}ef_{\mathfrak{P}} \wedge \forall_{w \in \mathcal{V}(\mathfrak{P})} vw \notin \mathcal{O}rd_{\mathfrak{P}}\} \tag{2.8}$$

umożliwi bowiem przedstawienie wpływu syntaktyki systemu Mizar na złożoność problemu optymalizacji jednego ze wskaźników czytelności.

Wprowadzimy teraz odpowiedniki pojęć poziomu zagnieżdżenia oraz rozumowania pierwotnego, wywodzące się z konstruktywnego grafu dowodu, zaadoptowane do abstrakcyjnego grafu dowodu.

Definicja 2.8. *Niech $\mathfrak{P}$ będzie abstrakcyjnym grafem dowodu, wówczas n–tym* poziomem zagnieżdżenia *w $\mathfrak{P}$ będziemy nazywać podgraf digrafu $\mathfrak{D}_{\mathfrak{P}}$ indukowany wierzchołkowo przez n-tego poziomu zagnieżdżenia lasu dendroidów $\mathfrak{M}eta_{\mathfrak{P}}$, $\mathfrak{D}_{\mathfrak{P}|\mathfrak{M}eta^n_{\mathfrak{P}}}$.*

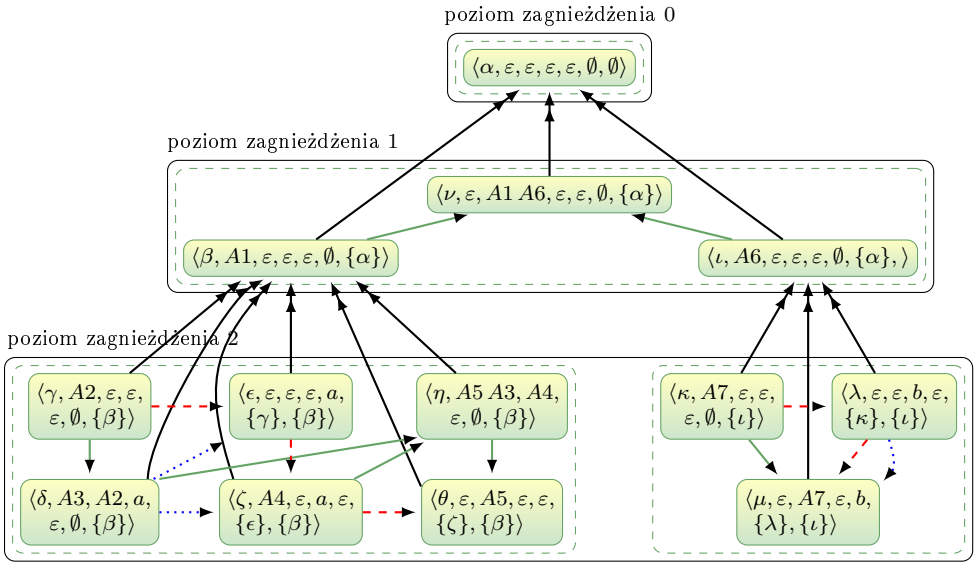
Definicja 2.9. *Niech $\mathfrak{P}$ będzie abstrakcyjnym grafem dowodu.* Rozumowaniem pierwotnym *w $\mathfrak{P}$, będziemy nazywać każdy podgraf G digrafu $\mathfrak{D}_{\mathfrak{P}}$ spełniający jeden z dwóch warunków:*

- *G jest zerowym poziomem zagnieżdżenia,*
- *istnieje wierzchołek $v \in \mathcal{V}(\mathfrak{D}_{\mathfrak{P}})$, dla którego $G = \mathfrak{D}_{\mathfrak{P}|\mathcal{N}^-_{\mathfrak{M}eta_{\mathfrak{P}}}(v)}$.*

Dodatkowo, symbolem $RP(\mathfrak{P})$ będziemy oznaczać rodzinę wszystkich rozumowań pierwotnych w $\mathfrak{P}$.

W celu zilustrowania wprowadzonych pojęć rozważmy dowód „reguły pijącego" (wydruk 2.11) oraz graf dowodu wygenerowany na jego podstawie (rys. 2.12), gdzie łuki ciągłe odpowiadają łukom referencyjnym, wykropkowane – pierwotnie porządkującym, kreskowane – szkieletowym, $\twoheadrightarrow$ – metakrawędziom, a rodziny wierzchołków otoczone kreską przerywaną – rozumowaniom pierwotnym.

Analizując położenie rozumowań pierwotnych względem poziomów zagnieżdżenia, stwierdzamy że każdy poziom zagnieżdżenia zawiera co najmniej jedno rozumowanie pierwotne. W szczególności drugi poziom zagnieżdżenia (rys. 2.12) zawiera dwa rozumowania pierwotne w postaci dwóch spójnych podgrafów. W ogólnym przypadku, nieopłacalnym jest jednak interpretowanie pojęcia rozumowania pierwotnego w terminach maksymalnych (w sensie zawierania) spójnych podgrafów. Istnienie referencji między różnymi poziomami zagnieżdżenia umożliwia bowiem „niewidoczny" przepływ informacji między wierzchołkami. Niech bowiem α_1, α_2 będą dwoma wierzchołkami z ustalonego rozumowania pierwotnego $\mathfrak{A}$. Wówczas, jeżeli istnieje podrozumowanie B uzasadniające stwierdzenie związane z wierzchołkiem α_2 oraz istnieje wierzchołek β należący do B, który wykorzystuje w uzasadnieniu stwierdzenie związane z wierzchołkiem α_1 lub identyfikator wprowadzony do rozumowania w tym wierzchołku, to informacja o tej zależności (między α_1 a α_2) nie będzie przechowywana w digrafie $\mathfrak{A}$. Ponadto, składnia systemu Mizar nie dopuszcza istnienia łuków referencyjnych łączących wierzchołki tego rodzaju. Wzbogacenie struktury abstrakcyjnego grafu dowodu o ten rodzaj informacji, umożliwia jednak analizowanie wynikłego digrafu w oparciu o zawarte w nim rozumowania pierwotne. Wprowadźmy więc operator wzbogacający strukturę o omówiony rodzaj informacji.



Rysunek 2.12: Abstrakcyjny graf dowodu opisujący rozumowanie zawarte w dowodzie „reguły pijącego” (wydruk 2.11).

Definicja 2.10. *Niech $\mathfrak{P}$ będzie abstrakcyjnym grafem dowodu. Mówimy, że $\widetilde{\mathfrak{P}} = \langle \mathcal{V}(\mathfrak{P}), \widetilde{\mathcal{A}(\mathfrak{P})}, \mathcal{M}(\mathfrak{P}) \rangle$ jest* domknięciem *abstrakcyjnego dowodu $\mathfrak{P}$ wtedy i tylko wtedy, gdy:*

$$(u,w) \in \widetilde{\mathcal{A}(\mathfrak{P})} \iff (u,w) \in \mathcal{A}(\mathfrak{P}) \vee \underset{\mathfrak{A} \in RP(\mathfrak{P}),\, v \in \mathcal{V}(\mathfrak{P})}{\exists} (u, w \in \mathcal{V}(\mathfrak{A}) \wedge u \underset{\mathfrak{D}_{\mathfrak{P}}}{\rightarrow} v \wedge v \underset{\mathfrak{Meta}_{\mathfrak{P}}}{\rightarrow}{}^{*} w). \tag{2.9}$$

W tym momencie oczywistym powinien być następujący fakt.

Lemat 2.11. *Domknięcie abstrakcyjnego grafu dowodu jest abstrakcyjnym grafem dowodu.*

W powstałym abstrakcyjnym grafie dowodu $\widetilde{\mathfrak{P}}$ definiujemy rodziny $\mathcal{R}ef_{\widetilde{\mathfrak{P}}}$, $\mathcal{O}rd_{\widetilde{\mathfrak{P}}}$, $\mathcal{O}rd^{\mathcal{P}}_{\widetilde{\mathfrak{P}}}$, $\mathcal{S}kel_{\widetilde{\mathfrak{P}}}$, $\mathtt{then}_{\widetilde{\mathfrak{P}}}$.

Definicja 2.12. *Niech $\mathfrak{P}$ będzie abstrakcyjnym grafem dowodu. Wówczas:*

$$\begin{aligned}
(u,w) \in \mathcal{R}ef_{\widetilde{\mathfrak{P}}} &\iff (u,w) \in \mathcal{R}ef_{\mathfrak{P}} \vee ((\underset{\mathfrak{A} \in RP(\mathfrak{P})}{\exists} u, w \in \mathfrak{A}) \wedge \\
&\qquad (\underset{v \in \mathcal{V}(\mathfrak{P})}{\exists} (u,v) \in \mathcal{R}ef_{\mathfrak{P}} \wedge u \underset{\mathfrak{Meta}_{\mathfrak{P}}}{\rightarrow}{}^{*} w)), \\
(u,w) \in \mathcal{O}rd_{\widetilde{\mathfrak{P}}} &\iff (u,w) \in \mathcal{O}rd_{\mathfrak{P}} \vee ((\underset{\mathfrak{A} \in RP(\mathfrak{P})}{\exists} u, w \in \mathfrak{A}) \wedge \\
&\qquad (\underset{v \in \mathcal{V}(\mathfrak{P})}{\exists} (u,v) \in \mathcal{O}rd_{\mathfrak{P}} \wedge u \underset{\mathfrak{Meta}_{\mathfrak{P}}}{\rightarrow}{}^{*} w)), \\
(u,w) \in \mathcal{O}rd^{\mathcal{P}}_{\widetilde{\mathfrak{P}}} &\iff (u,w) \in \mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}} \vee ((\underset{\mathfrak{A} \in RP(\mathfrak{P})}{\exists} u, w \in \mathfrak{A}) \wedge \\
&\qquad (\underset{v \in \mathcal{V}(\mathfrak{P})}{\exists} (u,v) \in \mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}} \wedge u \underset{\mathfrak{Meta}_{\mathfrak{P}}}{\rightarrow}{}^{*} w)), \\
\mathcal{S}kel_{\widetilde{\mathfrak{P}}} &= \mathcal{S}kel_{\mathfrak{P}}, \\
\mathtt{then}_{\widetilde{\mathfrak{P}}} &= \mathtt{then}_{\mathfrak{P}}.
\end{aligned} \tag{2.10}$$

Własności domknięcia abstrakcyjnego grafu dowodu $\widetilde{\mathfrak{P}}$ umożliwiają sformułowanie dodatkowych własności $\mathfrak{P}$. Pomijając przypadki rozumowań, które są prowadzone przez przypadki, możemy stwierdzić, że w rozumowaniu pierwotnym z $\widetilde{\mathfrak{P}}$, dla każdego kroku rozumowania, który jest wykorzystywany przy uzasadnieniu dowodzonego faktu, możemy wskazać co najmniej jeden krok szkieletowy w $\widetilde{\mathfrak{P}}$, który jest osiągalny z tego kroku. Jeśli więc istnieje krok, z którego żaden krok szkieletowy nie jest osiągalny, to jego usuniecie nie ma wpływu na poprawność rozumowania. Uwzględniając dodatkowo fakt, że łuki szkieletowe (jeżeli istnieją) tworzą skierowaną drogę łączącą wszystkie wierzchołki szkieletowe występujące w obrębie ustalonego rozumowania pierwotnego uzyskujemy poniższą obserwacje.

Obserwacja 2.13. *Niech $\mathfrak{P}$ będzie abstrakcyjnym grafem dowodu opisującym strukturę poprawnego dowodu w systemie Mizar. Jeżeli domknięcie $\widetilde{\mathfrak{P}}$ posiada rozumowanie pierwotne zawierające co najmniej dwa ujścia, tzn. wierzchołki nie posiadające następnika, to dowód ten posiada krok, którego usuniecie nie wpływa na poprawność rozumowania. Ponadto każde rozumowanie pierwotne w $\widetilde{\mathfrak{P}}$ zawiera dokładnie jeden maksymalny (w sensie zawierania) spójny podgraf, który jest konieczny do uzasadnienia dowodzonego stwierdzenia.*

2.7 Konstruktywność abstrakcyjnego grafu dowodu

W przedstawionych dotąd rozważaniach, omawiając strukturę abstrakcyjnego grafu dowodu ograniczaliśmy się do tych grafów, które odpowiadają konstruktywnym grafom dowodu. W obecnym podrozdziale wskażemy podrodzinę abstrakcyjnych grafów dowodu, dla której możliwe jest podanie konstrukcji dowodów akceptowalnych przez weryfikator systemu Mizar. Dodatkowo wskażemy sposób modyfikacji skryptów dowodowych, tak aby uzyskane rozumowania posiadały strukturę grafu dowodu należącą do tej rodziny, przy zachowaniu ich poprawności względem weryfikatora systemu Mizar.

Wprowadźmy oznaczenie $\mathfrak{L}_{\mathfrak{P}} = \{v \in \mathcal{V}(\mathfrak{P}) : \underset{w \in \mathcal{V}(\mathfrak{P})}{\forall} (v \neq w \implies \neg(w \underset{\mathfrak{Meta}_{\mathfrak{P}}}{\rightarrow}{}^{*} v))\}$ na zbiór liści lasu $\mathfrak{Meta}_{\mathfrak{P}}$ (przypomnijmy, że $\mathfrak{Meta}_{\mathfrak{P}}$ jest lasem drzew skierowanych o odwróconej orientacji). Zanim zaczniemy podawać warunki dostateczne dla konstruowalności rozumowania, zauważmy najpierw, że składnia systemu Mizar narzuca warunki na rodziny $\mathcal{R}ef_{\mathfrak{P}}$, $\mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}}$–łuków.

Obserwacja 2.14. *Niech $\mathfrak{P}$ będzie konstruktywnym grafem dowodu. Wówczas:*

1. *każdy krok wprowadzający zmienną ustaloną do rozumowania nie może być uzasadniany za pomocą zagnieżdżonego dowodu ($\underset{(v_1,v_2)\in\mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}}}{\forall} v_1 \in \mathfrak{L}_{\mathfrak{P}}$),*

2. *uzasadnienie każdego kroku w rozumowaniu wykorzystuje albo listę referencji* <Justified-Statement>*, albo zagnieżdżony dowód ($\underset{(v_1,v_2)\in\mathcal{R}ef_{\mathfrak{P}}}{\forall} v_2 \in \mathfrak{L}_{\mathfrak{P}}$).*

Warunki te narzucają również zależności między digrafem $\mathfrak{D}_{\mathfrak{P}}$ a lasem $\mathfrak{Meta}_{\mathfrak{P}}$ w konstruktywnym grafie dowodu $\mathfrak{P}$. Ustalmy bowiem $\mathcal{A}(\mathfrak{P})$–łuk (v_1, v_2) oraz załóżmy, że $v_2 \notin \mathfrak{L}_{\mathfrak{P}}$. Wówczas na mocy Obs. 2.14.2° wnioskujemy, że $(v_1, v_2) \in \mathcal{O}rd_{\mathfrak{P}}$. Stąd jeżeli dodatkowo założymy, że $v_1 \notin \mathfrak{L}_{\mathfrak{P}}$, to wykorzystując Obs. 2.14.1° stwierdzamy, że $(v_1, v_2) \in \mathcal{S}kel_{\mathfrak{P}} \setminus \mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}}$. W konsekwencji $\mathcal{A}(\mathfrak{P})$–łuki łączące wierzchołki niebędące liśćmi lasu $\mathfrak{Meta}_{\mathfrak{P}}$, nazywane dalej łukami *bezwzględnie szkieletowymi*, są łukami szkieletowymi w grafie dowodu każdego rozumowania zapisanego w języku Mizar, którego struktura jest opisywana za pomocą $\mathfrak{P}$.

Definicja 2.15. *Niech $\mathfrak{P}$ będzie abstrakcyjnym grafem dowodu.* Łukiem bezwzględnie szkieletowym *w $\mathfrak{P}$, będziemy nazywali każdy $\mathcal{A}(\mathfrak{P})$–łuk (v_1, v_2), dla którego wierzchołki v_1, v_2 nie należą do $\mathfrak{L}_{\mathfrak{P}}$.*

Zauważmy również, że łuki bezwzględnie szkieletowe muszą posiadać własności, jakie spełnia szkielet rozumowania, np. każdy łuk szkieletowy łączy wierzchołki w obrębie jednego rozumowania pierwotnego. Zauważmy również, że konsekwencją Obs. 2.14 jest istnienie wierzchołków, które w każdym dowodzie opisanym za pomocą struktury $\mathfrak{P}$ odpowiadają krokom wprowadzającym zmienne ustalone do rozumowania. Stąd w szczególności dopuszczenie łuków bezwzględnie szkieletowych w abstrakcyjnym grafie dowodu $\mathfrak{P}$ wymusza sformułowanie warunków gwarantujących kontrolę nad wykorzystaniem zmiennych w krokach szkieletowych (składnia systemu Mizar wymusza, iż zmienna ustalona v wprowadzona w nieszkieletowym kroku rozumowania pierwotnego $\mathfrak{A}$ może być wykorzystana w szkielecie tego rozumowania tylko w sytuacji, gdy pierwsze jej wystąpienie w tym szkielecie jest w kroku <*Exemplification*>).

Naturalnie, możliwe jest sformułowanie kilkunastu warunków gwarantujących konstruktywność $\mathfrak{P}$ w przypadku dopuszczającym istnienie łuków bezwzględnie szkieletowych w strukturze grafu dowodu. Sprawdzanie jednak, czy te warunki są zachowywane przez transformacje grafu dowodu, którym celem jest poprawa czytelności nie prowadzi do lepszego zrozumienia charakteru tych modyfikacji.

Omówione powyżej problemy znikną, jeśli zastosujemy transformacje skryptów dowodowych polegającą na modyfikacji kroków szkieletowych rodzaju <*Conclusion*> w rozumowaniu, które są uzasadniane za pomocą zagnieżdżonego rozumowania. Transformacja ta wiąże się z dodaniem nowego kroku rozumowania o powtórzonym polu <*Statement*> według schematu, gdzie podrozumowanie:

$$\begin{array}{l}\texttt{thus}\,[\,\texttt{then}\,]\,\texttt{<}\mathit{Statement}\texttt{>}\\ \quad\texttt{proof}\,\texttt{<}\mathit{Reasoning}\texttt{>}\,\texttt{end;}\end{array}\ , \tag{2.11}$$

zostaje zastąpione podrozumowaniem:

$$\begin{array}{l}[\,\texttt{then}\,]\,\texttt{<}\mathit{Statement}\texttt{>}\\ \quad\texttt{proof}\,\texttt{<}\mathit{Reasoning}\texttt{>}\,\texttt{end;}\\ \texttt{thus then}\,\texttt{<}\mathit{Statement}\texttt{>}\,\texttt{;}\end{array}\ . \tag{2.12}$$

Zastosowanie tej reguły nie wpływa na poprawność zmodyfikowanego skryptu i jednocześnie eliminuje łuki bezwzględnie szkieletowe z grafu dowodu.

Obserwacja 2.16. *Niech $\mathfrak{P}$ będzie konstruktywnym grafem dowodu opisującym strukturę poprawnego rozumowania w systemie Mizar, które zostało zmodyfikowane zgodnie z regułą (2.11-2.12). Wówczas w każdym rozumowaniu pierwotnym $\mathfrak{A} \in RP(\mathfrak{P})$, nie pokrywającym sie z zerowym poziomem zagnieżdżenia, istnieje wierzchołek należący do $\mathfrak{L}_{\mathfrak{P}}$, którego każdy następnik należący do $\mathcal{V}(\mathfrak{P})$ w grafie $\mathfrak{D}_{\mathfrak{P}}$ jest elementem zbioru $\mathfrak{L}_{\mathfrak{P}}$. Precyzując:*

$$\underset{\mathfrak{A}\in RP(\mathfrak{P})}{\forall}\ \mathcal{V}(\mathfrak{A}) \neq \mathfrak{M}eta^0_{\mathfrak{P}} \Longrightarrow \underset{c\in\mathcal{V}(\mathfrak{A})}{\exists}\,(c \in \mathfrak{L}_{\mathfrak{P}} \wedge \underset{w\in\mathcal{V}(\mathfrak{P})}{\forall}\, w \notin \mathfrak{L}_{\mathfrak{P}} \Longrightarrow \neg(c \underset{\mathfrak{D}_{\mathfrak{P}}}{\rightarrow} w)). \tag{2.13}$$

Dowód. Z konstruowalności $\mathfrak{P}$ uzyskujemy, że każde rozumowanie pierwotne $\mathfrak{A} \in \mathrm{RP}(\mathfrak{P})$, z pominięciem zerowego poziomu zagnieżdżenia, zawiera wierzchołek c rodzaju <*Conclusion*>. Przypuśćmy, że $c \notin \mathfrak{L}_{\mathfrak{P}}$. Wówczas krok związany z wierzchołkiem c ma postać 2.11, skąd w wyniku transformacji został wygenerowany nowy krok, który jest liściem w grafie dowodu zmodyfikowanego skryptu. W celu zakończenia dowodu wystarczy więc jedynie zauważyć, że każdy wierzchołek c w zmodyfikowanym

skrypcie rodzaju <*Conclusion*> spełnia własność $c \in \mathfrak{L}_{\mathfrak{P}} \wedge \underset{w\in\mathcal{V}(\mathfrak{P})}{\forall}(w \notin \mathfrak{L}_{\mathfrak{P}} \Longrightarrow \neg(c \underset{\mathfrak{D}_{\mathfrak{P}}}{\rightarrow} w))$. □

Twierdzenie 2.17. *Niech $\mathfrak{P}$ będzie abstrakcyjnym grafem dowodu nie zawierającym łuków bezwzględnie szkieletowych oraz spełniającym własność* (2.13). *Wówczas $\mathfrak{P}$ jest grafem konstruktywnym oraz skonstruowane rozumowanie nie zawiera łuków bezwzględnie szkieletowych, a zbiór kroków rozumowania wprowadzających zmienne do rozumowania pokrywa się ze zbiorem $\{v \in \mathcal{V}(\mathfrak{P}) : \underset{w\in\mathcal{V}(\mathfrak{P})}{\exists} v \underset{\mathfrak{D}_{\mathfrak{P}}}{\rightarrow} w \wedge w \notin \mathfrak{L}_{\mathfrak{P}}\}$.*

Dowód. Niech $\mathfrak{P}$ spełnia założenia twierdzenia. Oznaczmy przez $\mathfrak{A}_0$ rozumowanie pierwotne pokrywające się wierzchołkowo z zerowym poziomem zagnieżdżenia oraz niech $\mathcal{C}on\colon \mathrm{RP}(\mathfrak{P})\setminus\{\mathfrak{A}_0\} \to \mathfrak{L}_{\mathfrak{P}}$ przyporządkowuje każdemu rozumowaniu pierwotnemu $\mathfrak{A} \in \mathrm{RP}(\mathfrak{P}) \setminus \{\mathfrak{A}_0\}$ wierzchołek $c \in \mathcal{V}(\mathfrak{A}) \cap \mathfrak{L}_{\mathfrak{P}}$ spełniający zależność:

$$\underset{w\in\mathcal{V}(\mathfrak{P})}{\forall} w \notin \mathfrak{L}_{\mathfrak{P}} \Longrightarrow \neg(c \underset{\mathfrak{D}_{\mathfrak{P}}}{\rightarrow} w). \tag{2.14}$$

Wprowadźmy oznaczenie $\mathcal{N}^{-}_{\mathfrak{D}_{\mathfrak{P}}}(v, V_1)$ na zbiór $\mathcal{N}^{-}_{\mathfrak{D}_{\mathfrak{P}}}(v) \cap V_1$, gdzie $v \in \mathcal{V}(\mathfrak{P})$, $V_1 \subseteq \mathcal{V}(\mathfrak{P})$. Ustalmy również linearyzację $\tau \in TS(\mathfrak{G}_{\mathfrak{P}})$ oraz oznaczmy dwa zbiory

$$\mathcal{V}er := \{v \in \mathcal{V}(\mathfrak{P}) : \underset{w\in\mathcal{V}(\mathfrak{P})}{\exists} v \underset{\mathfrak{D}_{\mathfrak{P}}}{\rightarrow} w \wedge w \notin \mathfrak{L}_{\mathfrak{P}}\},\ \ \mathcal{C} := \mathcal{C}on(\mathrm{RP}(\mathfrak{P}) \setminus \{\mathfrak{A}_0\}). \tag{2.15}$$

Jak łatwo można wykazać, zbiory $\mathcal{V}er$, $\mathcal{C}$ są rozłączne. Zdefiniujmy następnie dwa funktory:

$$\begin{array}{rcl} \mathit{Just}(V_1) & := & \texttt{by}\ A\tau(v_1), A\tau(v_2), A\tau(v_{|V_1|}), \\ \mathit{Just}(\emptyset) & := & \text{' '}, \\ \mathit{Expr}(V_1) & := & x\tau(v_1) = x\tau(v_1)\ \texttt{\&}\ \ldots\ \texttt{\&}\ x\tau(v_{|V_1|}) = x\tau(v_{|V_1|}), \\ \mathit{Expr}(\emptyset) & := & \texttt{not contradiction}, \end{array} \tag{2.16}$$

gdzie $\emptyset \neq V_1 \subseteq \mathcal{V}(\mathfrak{P})$, $V_1 = \{v_1, v_2, \ldots, v_{|V_1|}\}$, $\tau(v_1) < \tau(v_2) < \ldots < \tau(v_{|V_1|})$. Funktor $\mathit{Just}(V_1)$ określa frazę wskazującą, że krok, w którym zostanie ona użyta, korzysta w swoim uzasadnieniu odwołuje się do stwierdzeń sformułowanych w krokach z V_1. Z kolei $\mathit{Expr}(V_1)$ definiuje formułę używającą zmienne ustalone wprowadzone w krokach z V_1. Rozważmy konstrukcję rozumowania podzieloną na trzy etapy:

(1°) Przyporządkowujmy każdemu wierzchołkowi $v \in \mathfrak{L}_{\mathfrak{P}}$ krok $S(v)$ zgodnie z regułą:

$$S(v) := \begin{cases} \texttt{consider}\ x\tau(v)\ \texttt{be}\ \textit{set}\ \texttt{such that} & \\ \quad A\tau(v)\texttt{:}\ \mathit{Expr}(\emptyset)\ \texttt{by}\ \mathit{Just}(\mathcal{N}^{-}_{\mathfrak{D}_{\mathfrak{P}}}(v, \mathcal{V}(\mathfrak{P})))\texttt{;} & \text{gdy } v \in \mathcal{V}er, \\ \texttt{thus}\ A\tau(v)\texttt{:}\ \texttt{thesis by}\ \mathit{Just}(\mathcal{N}^{-}_{\mathfrak{D}_{\mathfrak{P}}}(v, \mathcal{V}(\mathfrak{P})))\texttt{;} & \text{gdy } v \in \mathcal{C}, \\ A\tau(v)\texttt{:}\ \mathit{Expr}(\mathcal{N}^{-}_{\mathfrak{D}_{\mathfrak{P}}}(v, \mathcal{V}er))\ \texttt{by}\ \mathit{Just}(\mathcal{N}^{-}_{\mathfrak{D}_{\mathfrak{P}}}(v, \mathcal{V}(\mathfrak{P})))\texttt{;} & \text{w.p.p.} \end{cases} \tag{2.17}$$

Krok $S(v)$ w przypadku $v \in \mathcal{V}er$ odpowiada wówczas za wprowadzenie do rozumowania zmiennej ustalonej $x\tau(v)$ oraz etykiety $A\tau(v)$. Zauważmy, że wprowadzenie do rozumowania zmiennej ustalonej oraz etykiety umożliwia zarówno odwołanie się do tego kroku z innego kroku s łukiem porządkującym jak i referencyjnym. Pierwszy rodzaj łuków odpowiada sytuacji, w której w sformułowaniu kroku s została wykorzystana zmienna $x\tau(v)$. Natomiast drugi rodzaj łuku odpowiada sytuacji, w której w uzasadnieniu kroku s istnieje odwołanie do kroku $S(v)$ przez wskazanie etykiety $A\tau(v)$. Umożliwienie prowadzenia obu

rodzajów łuków argumentujących z wierzchołka v jest jednak uzasadnione tylko w przypadku $v \in \mathcal{V}er$. Z Obs.2.13.2° oraz z konstrukcji zbioru $\mathcal{V}er$ wynika bowiem, że istnieje co najmniej jeden wierzchołek $u \notin \mathcal{V}(\mathfrak{P}) \setminus \mathfrak{L}_{\mathfrak{P}}$, dla którego argumentujący łuk vu nie może być łukiem referencyjnym.

Z kolei $S(v)$ w przypadku $v \in \mathcal{C}$ odpowiada za wprowadzenie do rozumowania kroku rodzaju $<$*Conclusion*$>$. W poprawnie zbudowane zagnieżdżonym rozumowaniu zapisanym w języku Mizar musi bowiem istnieć co najmniej jeden krok rodzaju $<$*Skeleton-Step*$>$. Dodatkowo ostatni krok szkieletowy w każdym takim rozumowaniu musi być rodzaju $<$*Conclusion*$>$.

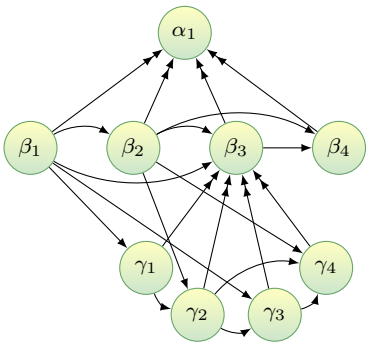
W ostatnim przypadku, $v \notin \mathcal{V}er \cup \mathcal{C}$, krok $S(v)$ odpowiada jedynie za wprowadzenie do rozumowania etykiety $A\tau(v)$, umożliwiającej odwoływanie się do tego kroku łukiem referencyjnym.

(2°) Oznaczmy następnie przez $\mathfrak{R}(v)$ rozumowanie pierwotne $\mathfrak{D}_{\mathfrak{P}|\mathcal{N}^-_{\mathfrak{Meta}_{\mathfrak{P}}}(v)}$ należące do $\mathrm{RP}(\mathfrak{P}) \setminus \{\mathfrak{A}_0\}$, gdzie $v \in \mathcal{V}(\mathfrak{P}) \setminus \mathfrak{L}_{\mathfrak{P}}$ (w konstruowanym dowodzie, rozumowanie $\mathfrak{R}(v)$ będzie opisywać strukturę zagnieżdżonego podrozumowania uzasadniającego krok związany z wierzchołkiem v). Ustawmy w ciąg $\{u_i\}_{i=1}^{|\mathcal{V}(\mathfrak{P})\setminus\mathfrak{L}_{\mathfrak{P}}|}$ elementy zbioru $\mathcal{V}(\mathfrak{P}) \setminus \mathfrak{L}_{\mathfrak{P}}$, tak aby spełniona była zależność: jeśli $1 \leqslant i \leqslant j \leqslant |\mathcal{V}(\mathfrak{P}) \setminus \mathfrak{L}_{\mathfrak{P}}|$, zaś p_i, p_j są poziomami zagnieżdżenia w $\mathfrak{Meta}_{\mathfrak{P}}$ do których należą odpowiednio wierzchołki u_i, u_j, to $p_i \geqslant p_j$.

(3°) Wyznaczymy teraz rekurencyjnie ciąg wyrażeń $S(u_i)$. Ustalmy w tym celu liczbę i spełniającą zależność $1 \leqslant i \leqslant |\mathcal{V}(\mathfrak{P}) \setminus \mathfrak{L}_{\mathfrak{P}}|$ oraz załóżmy, że zostały już wyznaczone wyrażenia $S(u_j)$ dla wszystkich $1 \leqslant j < i$ (przypadek $i = 1$ nie wymaga odrębnej analizy). Z założenia, z każdym wierzchołkiem $v \in \mathcal{V}(\mathfrak{R}(u_i))$ zostało związane wyrażenie $S(v)$, skąd poprawnie określone jest wyrażenie $S(u_i)$:

$$S(u_i) = A\tau(u_i)\colon \mathit{Expr}(\mathcal{N}^-_{\mathfrak{D}_{\mathfrak{P}}}(v, \mathcal{V}er))\ \texttt{proof}\ S(v_1)\ S(v_2)\ \ldots\ S(v_{|\mathcal{V}(\mathfrak{R}(u_i))|})\ \texttt{end;}, \tag{2.18}$$

gdzie $\{v_1, v_2, \ldots, v_{|\mathcal{V}(\mathfrak{R}(u_i))|}\} = \mathcal{V}(\mathfrak{R}(u_i))$ oraz $\tau(v_1) < \tau(v_2) < \ldots < \tau(v_{|\mathcal{V}(\mathfrak{R}(u_i))|})$.




```
α1: AV: not contradiction
      proof
β1:     consider x1 be set such that
          A1: not contradiction ;
β2:     consider x2 be set such that
          A2: not contradiction by A1;
β3:     A7: x1=x1 & x2=x2
          proof
γ1:         A3: x1=x1 by A1;
γ2:         A4: x2=x2 by A2, A3;
γ3:         A5: x1=x1 by A1, A4;
γ4:         thus A6: thesis by A2, A4, A5;
          end;
β4:     thus A8: thesis by A2, A7;
      end;
```

Rysunek 2.13: Linearyzacja grafu dowodu, ilustrująca konstrukcję zawartą w dowodzie Tw. 2.17, gdzie linearyzacja τ jest określona uporządkowaniem: β_1, β_2, γ_1, γ_2, γ_3, γ_4, β_3, β_4, α_1; $\mathcal{V}er = \{\beta_1, \beta_2\}$, $\mathcal{C} = \{\beta_4, \gamma_4\}$.

Ciąg stwierdzeń $S(v_1)\ S(v_2)\ \ldots\ S(v_{|\mathcal{V}(\mathfrak{A}_0)|})$ jest wówczas szukanym dowodem zapisanym w języku Mizar, gdzie $\{v_1, v_2, \ldots, v_{|\mathcal{V}(\mathfrak{A}_0)|}\} = \mathcal{V}(\mathfrak{A}_0)$ oraz $\tau(v_1) < \tau(v_2) < \ldots < \tau(v_{|\mathcal{V}(\mathfrak{A}_0)|})$. W celu zakończenia dowodu wystarczy jedynie zauważyć, że formuły $Expr(\mathcal{N}^{-}_{\mathfrak{D}_\mathfrak{P}}(v, \mathcal{V}er))$ są oczywiste dla weryfikatora systemu Mizar oraz, że każde wystąpienie identyfikatora etykiety $\mathtt{A}\tau(v)$ i zmiennej ustalonej $\mathtt{x}\tau(u)$ w zlinearyzowanym rozumowaniu jest poprzedzone krokiem, w którym te identyfikatory zostały wprowadzone, gdyż $\tau \in TS(\mathfrak{G}_\mathfrak{P})$, gdzie $v \in \mathcal{V}(\mathfrak{P})$, $u \in \mathcal{V}ar$. W konsekwencji, skonstruowane rozumowanie jest poprawne dla weryfikatora systemu Mizar oraz, jak łatwo można to uzasadnić, graf dowodu tego rozumowania pokrywa się z $\mathfrak{P}$. □

Twierdzenie 2.18. *Niech D będzie acyklicznym digrafem, wówczas istnieje konstruktywny abstrakcyjny graf dowodu zawierający rozumowanie pierwotne $\mathfrak{A}$, którego struktura pokrywa się z A. Dodatkowo każdy D–łuk odpowiada łukowi referencyjnemu w $\mathfrak{A}$.*

Dowód. Niech D będzie acyklicznym digrafem. Wybierzmy element $r \notin \mathcal{V}(D)$, który będzie korzeniem w lesie dendroidów konstruowanego grafu dowodu. Łatwo można wykazać, że struktura

$$\mathfrak{P}(D) := \langle \mathcal{V}(D) \cup \{r\}, \mathcal{A}(D), \{vr : v \in \mathcal{V}(D)\}\rangle \tag{2.19}$$

jest abstrakcyjnym grafem dowodu, posiadającym dokładnie jedno rozumowanie pierwotne $\mathfrak{A}_D$ zawarte w pierwszym poziomie zagnieżdżenia, tożsamościowo równe D. Pokażemy, że struktura $\mathfrak{P}(D)$ spełnia założenia Tw. 2.17. Na mocy równości $\mathfrak{L}_{\mathfrak{P}(D)} = \mathcal{V}(D)$ stwierdzamy, że w $\mathfrak{P}(D)$ istnieje dokładnie jeden wierzchołek r, który nie jest liściem. Stąd na mocy Def. 2.15, $\mathfrak{P}(D)$ nie zawiera łuków bezwzględnie szkieletowych. Z równości $\mathfrak{L}_{\mathfrak{P}(D)} = \mathcal{V}(D)$ uzyskujemy również, że dla dowolnego wierzchołka v z $\mathfrak{A}_D$, każdy następnik wierzchołka v w $\mathfrak{D}_{\mathfrak{P}(D)}$, należy do $\mathfrak{L}_{\mathfrak{P}(D)}$. W celu zakończenia dowodu, wystarczy więc jedynie zauważyć, że

$$\{v \in \mathcal{V}(D) \cup \{r\} : \underset{w \in \mathcal{V}(D) \cup \{r\}}{\exists} v \underset{\mathfrak{D}_{\mathfrak{P}(D)}}{\rightarrow} w \wedge w \notin \mathfrak{L}_{\mathfrak{P}(D)}\} = \emptyset, \tag{2.20}$$

skąd graf dowodu $\mathfrak{P}(D)$ nie posiada łuków porządkujących. Tym samym, każdy $\mathfrak{A}_D$–łuk jest referencyjnym $\mathcal{A}(D)$–łukiem. □

Rozdział 3

Kryteria czytelności dowodów

Czytelność dowodów zapisanych w deklaratywnym systemie formalnym jest własnością dyskusyjną, różnie rozumianą przez autorów poszczególnych artykułów. Naturalnie chcielibyśmy, aby teksty formalnych rozumowań były bardziej zbliżone do tych, które występują w nieformalnych dowodach matematycznych. Analizując długie rozumowania coraz trudniejszych zagadnień zawartych w bibliotece systemu Mizar (MML), możemy jednak zauważyć proces odwrotny. Niejednokrotnie przeprowadzane rozumowania, choć poprawne dla systemu weryfikującego, są mało eleganckie, niejednokrotnie wręcz chaotyczne, a człowiek może je zrozumieć jedynie przy bardzo dużym nakładzie pracy. Autorzy takich dowodów mają bowiem przekonanie, że kwestia odnajdywania i usuwania fragmentów, które są niepotrzebne lub możliwe do skrócenia jest drugorzędna, ponieważ można ją zautomatyzować.

Warto podkreślić w tym miejscu, że odnajdywanie kroków możliwych do usunięcia z rozumowania (Obs. 2.13), jest problemem nieporównywalnie łatwiejszym od reorganizacji struktury rozumowania w celu poprawy jej czytelności. Uzasadnimy bowiem w dalszych rozważaniach, że poprawa czytelności wiąże się na ogół z rozwiązywaniem problemów NP-trudnych.

Do rozwiązywania problemów poprawy czytelności są wykorzystywane dwa rodzaje środków (metod), reprezentujące niezależne podejścia do tego zagadnienia. Pierwsza grupa metod, opisana w podrozdziale 3.1, służy wyróżnieniu idei dowodu w oparciu o kapsułkowanie lokalnych podrozumowań oraz wprowadzanie cięć w rozumowaniu. Wyniki generowane przez narzędzia stworzone dla tej grupy metod skracają długie rozumowania pierwotne, ukrywając podrozumowania uzasadniające główne kroki dowodu na głębszych poziomach zagnieżdżenia dowodu albo izolując je poza obszar dowodu w postaci lematów. Natomiast druga grupa metod, omawiana w podrozdziale 3.2, opiera się na modyfikacji uporządkowania (linearyzacji) niezależnych od siebie kroków rozumowania w celu uwypuklenia spójności lokalnych podrozumowań oraz poprawy wybranych własności linearyzacji rozumowania.

Przedstawione kategorie różnią się również pod względem aparatury pojęciowej niezbędnej do opisu napotykanych problemów. Realizacja pierwszej grupy metod poprawy czytelności wiąże się w głównej mierze z modyfikacją abstrakcyjnego grafu dowodu oraz z wdrożeniem uzyskanych zmian w rozważanej linearyzacji rozumowania przy możliwie minimalnej jej modyfikacji. Natomiast druga grupa metod jest realizowana jedynie w oparciu o przekształcenie linearyzacji poszczególnych rozumowań pierwotnych.

Część wyników przedstawionych w tym rozdziale została opublikowana w *Journal of Automated Reasoning* [72]. Na potrzebę rozwijania i wdrażania w innych systemach metod przedstawionych w tej publikacji zwrócono uwagę w [42].

3.1 Metody poprawy czytelności oparte o wyodrębnianie podrozumowań

Porównując rozwój komputerowej weryfikacji formalnych rozumowań matematycznych oraz języków programowania możemy dostrzec wiele podobieństw. Należy wskazać dwie główne przyczyny tych zależności:

(*i*) w obu przypadkach źródło jest tworzone w sztucznym języku, który ma jednoznacznie zdefiniowane znaczenie dla systemu komputerowego,

(*ii*) autorzy kodu w obu przypadkach dążą do osiągnięcia konkretnego celu, pomniejszając rolę czytelności tworzonego źródła.

Naturalnie czytelność źródła ma istotny wpływ na jego konserwację [3], dlatego stosunkowo wcześnie zostały rozpoczęte badania nad sposobami poprawy ich czytelności, które miały na celu opracowanie narzędzi ułatwiających proces analizowania źródła. Ze względu jednak na wcześniejszy rozwój języków programowania i ich powszechne stosowanie, uzyskane rozwiązania w tej dziedzinie były inspiracją dla wielu rozwiązań wykorzystywanych we wspomaganej komputerowo formalizacji.

W obu tych dziedzinach wdrożony bowiem został cukier syntaktyczny (ang. *syntactic sugar*), który umożliwia upodobnienie sztucznego języka do naturalnego, dzięki czemu staje się on bardziej zrozumiały dla użytkownika. Nie będzie on jednak przedmiotem dalszych rozważań. Innym powszechnie wykorzystywanym rozwiązaniem w językach programowania jest poprawa wizualizacyjna źródła w dedykowanych edytorach tekstowych, opierająca się na automatycznym rozpoznawaniu składni języka oraz rozbudowanym systemie podpowiedzi (ang. *infotip*). Badania przeprowadzone przez zespół B. A. Myersa [47] wykazały, że brak takich narzędzi istotnie wydłuża czas potrzebny na modyfikację kodu. Wyniki uzyskane wśród eksperymentalnej grupy programistów dowiodły bowiem, że aż 35% czasu potrzebnego na modyfikację kodu jest zużywane na przeszukiwanie informacji o jego strukturze.

Prace prowadzone przez J. Urbana nad wdrożeniem takich narzędzi dostarczyły użytkownikom systemu Mizar specjalnego trybu w dystrybucji edytora GNU Emacs [79, 81, 87, 88]. Tryb ten oprócz kolorowania składni i wskazywania stwierdzeń, do których odwołują się referencje, umożliwił „zwijanie" istniejących zagnieżdżonych rozumowań (bloków kodu ograniczonych przez określone słowa kluczowe m.in `proof`, `end`), jak również automatyczne generowanie szkieletu dowodów wskazanych formuł. Uprościł on również dostęp do narzędzi dystrybuowanych z systemem Mizar [90] oraz dostarczył moduł, który generuje uzasadnienie poszczególnych kroków rozumowania, wykorzystując przy tym rozwiązania stosowane w automatycznym dowodzeniu twierdzeń [5, 89]. Niezależne prace J. Urbana nad wizualizacją istniejących rozumowań doprowadziły również do stworzenia narzędzia przekształcającego skrypt dowodowy do zlinkowanej postaci HTML ([4, 6, 86], zob. również http://mizar.org/version/current/html/), zawierającej dodatkowe informacje o strukturze dowodu, które nie są zawarte *explicite* w skrypcie dowodowym m.in.:

(*i*) wskazanie definicji funktorów, które zostały wykorzystane w sformułowaniach stwierdzeń,

(*ii*) wskazanie definicji, które zostały rozwinięte w szkielecie dowodu,

(*iii*) wskazanie formuły, która pozostała do udowodnienia przy każdym kroku szkieletowym występującym w rozumowaniu.

Informacje te mają szczególne znaczenie ze względu na liczbę redefinicji funktorów oraz nadpisywaniem (ang. *overloading*) symboli (w bazie MML symbol "+" jest definiowany lub redefiniowany dla różnych typów argumentów ponad 130 razy).

Wszystkie te narzędzia w naturalny sposób upraszczają proces poszukiwania stwierdzeń wykorzystywanych w uzasadnieniach, jak również rekonstrukcję idei konkretnego rozumowania, odwracając uwagę czytelnika od nieanalizowanych w danym momencie fragmentów tekstu dowodu. Badając jednak dowody zgromadzone w bazie MML, bez trudu odnajdujemy jednopoziomowe rozumowania zbudowane z ponad 50 krokami, dla których rekonstrukcja idei nawet przy zastosowaniu ww. narzędzi wymaga ogromnego nakładu pracy.

Poszukując rozwiązania tego problemu wśród metod poprawy czytelności długich rozumowań zawartych w nieformalnych dowodach matematycznych, możemy zauważyć, że autorzy tych rozumowań niejednokrotnie wyodrębniają z nich mniej istotne fragmenty w postaci lematów. Metoda ta jest szczególnie stosowana w sytuacji, kiedy ustalony fragment rozumowania jest wykorzystywany kilkakrotnie w dowodzie i został on wyodrębniony w postaci lematu w celu uniknięcia powtórzeń. Wyodrębnianie takich fragmentów umożliwia również odwrócenie uwagi czytelnika od mniej ważnych części rozumowania, ułatwiając tym samym odnalezienie i zrozumienie idei dowodu. Czytelnik tak zmodyfikowanego rozumowania ma oczywiście możliwość szczegółowego przeanalizowania sposobu uzasadnienia konkretnego kroku poprzez:

(*i*) odesłanie do wskazanych wcześniej uzasadnionych faktów

(*ii*) lub wskazane zagnieżdżonego rozumowania, występującego bezpośrednio po tym kroku jako jego uzasadnienie.

Wybór jednej z tych dwóch metod wyodrębnienia fragmentu rozumowania, jak również sam dobór fragmentu, zależy od indywidualnych poglądów autora. Wybierane fragmenty, które powinny stanowić zamknięte i spójne części dowodu z punktu widzenia opisywanych zagadnień, będziemy nazywać *paczkami*.

Przedstawione sposoby uwypuklania idei rozumowania są przenoszone przez wielu autorów na grunt dowodów formalnych w systemie Mizar. Autorzy ci maja bowiem przekonanie, że równie ważnym priorytetem w formalizacji jak powiększanie bazy sformalizowanych twierdzeń jest jej jakość. Przekonanie to uwidacznia się również wśród użytkowników innych systemów (Isabelle [11]), którzy podejmują prace nad poprawą czytelności skryptów dowodowych, w tym również nad wyodrębnianiem lematów [73]. Poszukiwanie sposobów, umożliwiających automatyczne wyszukiwanie, jak i wyodrębnianie paczek z rozumowań wydaje się więc pożądanym kierunkiem rozwoju metod poprawy czytelności dowodów formalnych, wychodzącym naprzeciw oczekiwaniom wielu użytkowników.

3.1.1 Metody wyodrębniania paczek z rozumowania

Przedstawione powyżej własności rozumowań nieformalnych, możemy wdrażać w dowodach formalnych, stosując jedną z dwóch metod.

Pierwsza metoda: „ukrywając" (kapsułkując) mniej ważne, często techniczne fragmenty rozumowania na bardziej zagłębionych poziomach dowodu w postaci zagnieżdżonych podrozumowań, których dowody zostały wygenerowane na podstawie kroków rozumowania z wyodrębnionych paczek. Każda spośród wyodrębnionych paczek jest wówczas zastępowana na ogół pojedynczym nowym krokiem, którego zadaniem jest opisanie informacji uzyskanych na podstawie rozumowania zawartego w tej paczce.

Druga metoda: „usuwając" (izolując) paczki poza obszar dowodu, zastępując je na ogół pojedynczymi krokami, z których każdy odwołuje się do zewnętrznego lematu, wygenerowanego na podstawie tych paczek.

Szczegółowa analiza przedstawionych metod, będąca tematem tego podrozdziału, została opublikowana w [72].

W obu przypadkach stwierdzenie związane z nowym krokiem w zmodyfikowanym rozumowaniu powinno być koniunkcją stwierdzeń uzasadnionych wewnątrz paczki, do których odwołują się inne kroki z poza obszaru paczki. Konieczność wprowadzenia nowych kroków do rozumowania jest bowiem następstwem faktu, że podczas wyodrębniania paczki z rozumowania wszystkie łuki referencyjne prowadzące z wewnątrz na zewnątrz obszaru paczki muszą być usunięte. W celu zapewnienia poprawności w ten sposób zmodyfikowanego rozumowania wszystkie usuwane referencje odwołujące się do wnętrza ustalonej paczki, są zastępowane referencjami do nowego kroku zastępującego tę paczkę.

W dalszej części sekcji, każdy krok należący do wnętrza paczki, z którym związane stwierdzenie jest wykorzystywane jako przesłanka poza obszarem tej paczki będziemy nazywać *konkluzją paczki*. Podobnie, każdy krok nie należący do wnętrza paczki, który jest wykorzystywany jako przesłanka jakiegokolwiek kroku należącego do wnętrza paczki będziemy nazywać *przesłanką paczki*.

Definicja 3.1. *Niech $\mathfrak{P}$ będzie abstrakcyjnym grafem dowodu.* Paczką $\mathcal{P}$ *będziemy nazywać każdy niepusty indukowany wierzchołkowo podgraf digrafu $\mathfrak{D}_{\mathfrak{P}}$, dla którego istnieje rozumowanie pierwotne $\mathfrak{A} \in RP(\mathfrak{P})$ zawierające jako podgraf $\mathcal{P}$.*

Wykorzystując fakt, że zbiory wierzchołków rozumowań pierwotnych w $\mathfrak{P}$ stanowią partycję zbioru $\mathcal{V}(\mathfrak{P})$, uzyskujemy poniższą obserwację.

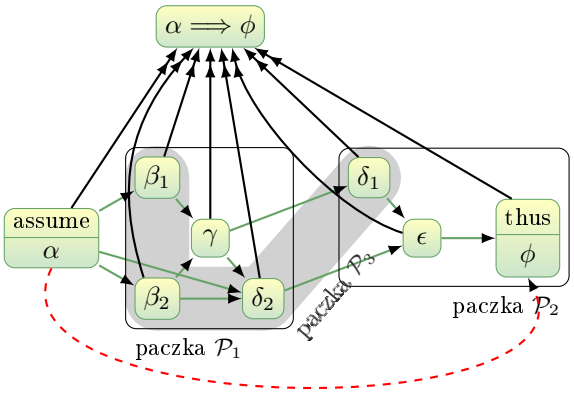
Obserwacja 3.2. *Rozumowanie pierwotne $\mathfrak{A}$ związane z paczką $\mathcal{P}$ abstrakcyjnego grafu dowodu $\mathfrak{P}$ jest wyznaczone jednoznacznie.*

Definicja 3.3. *Niech $\mathcal{P}$ będzie paczką w abstrakcyjnym grafie dowodu $\mathfrak{P}$.* Obszarem paczki *będziemy nazywać zbiór wierzchołków $\mathcal{O}bsz(\mathcal{P}) \subseteq \mathcal{V}(\mathfrak{P})$ spełniający zależność* $v \in \mathcal{O}bsz(\mathcal{P}) \iff \underset{u \in \mathcal{V}(\mathcal{P})}{\exists} v \underset{\mathfrak{Meta}_{\mathfrak{P}}}{\rightarrow} {}^{*}u$.

Definicja 3.4. *Niech $\mathcal{P}$ będzie paczką w abstrakcyjnym grafie dowodu $\mathfrak{P}$.* Konkluzją paczki *będziemy nazywać każdy wierzchołek $v \in \mathcal{V}(\mathcal{P})$, dla którego możemy wskazać $\mathcal{R}ef_{\mathfrak{P}}$–łuk o początku w tym wierzchołku oraz końcu nie należącym do $\mathcal{O}bsz(\mathcal{P})$. Zbiór wszystkich konkluzji paczki $\mathcal{P}$ będziemy oznaczać $\mathcal{C}on(\mathcal{P})$.*

Definicja 3.5. *Niech $\mathcal{P}$ będzie paczką w abstrakcyjnym grafie dowodu $\mathfrak{P}$.* Przesłanką paczki *będziemy nazywać każdy wierzchołek z $\mathfrak{P}$ nie należący do $\mathcal{P}$, dla którego możemy wskazać $\mathcal{R}ef_{\mathfrak{P}}$–łuk o początku w tym wierzchołku oraz końcu należącym do $\mathcal{O}bsz(\mathcal{P})$. Zbiór wszystkich przesłanek paczki $\mathcal{P}$ będziemy oznaczać $\mathcal{P}re(\mathcal{P})$.*

W celu zilustrowania przedstawionych metod wyizolowywania paczek z rozumowania, rozważmy abstrakcyjny graf dowodu (rys. 3.1) posiadający dwie wyróżnione paczki $\mathcal{P}_1$, $\mathcal{P}_2$, który to graf reprezentuje m.in. uzasadnienie formuły $\alpha \Longrightarrow \phi$ przedstawione na wydruku 3.2. Dodatkowa paczka $\mathcal{P}_3$ będzie analizowana w dalszej części podrozdziału. Uwaga, wierzchołki <*Skeleton-Step*> posiadają dodatkową informacje o rodzaju wierzchołka, `assume` w przypadku wierzchołka rodzaju <*Assumption*> oraz `thus` w przypadku <*Conclusion*>. Aplikacja pierwszej metody wyodrębniania do paczek $\mathcal{P}_1$, $\mathcal{P}_2$ wiąże się wówczas z dodaniem do rozumowania dwóch nowych kroków $\gamma \wedge \delta_2$, ϕ (rys. 3.5), z którymi związane stwierdzenia są koniunkcją konkluzji odpowiednich paczek (wyodrębnienie paczek z rozumowania przedstawionego na wydruku 3.2



Rysunek 3.1: Abstrakcyjny graf dowodu oraz jego linearyzacja zapisana w języku Mizar, reprezentujące przykładowe uzasadnienie implikacji $\alpha \Longrightarrow \phi$.

```
α ⟹ φ
proof
  assume A: α;
  B1: β1 by A;
  B2: β2 by A;
  C: γ by B1, B2;
  D1: δ1 by C;
  D2: δ2 by A, B2, C;
  E: ϵ by D1,D2;
  thus F: φ by E;
end;
```

Wydruk 3.2: Przykład rozumowania zapisanego w języku Mizar, którego struktura jest opisana przez graf dowodu przedstawionym na rys. 3.1.

za pomocą pierwszej metody zostało przedstawione na wydruku 3.3). Dodatkowo łuki referencyjne, które odwoływały się do konkluzji paczek (np. (γ, δ_1)), w zmodyfikowanym rozumowaniu zostały zastąpione łukami odwołującymi się do nowych kroków (np. $(\gamma \wedge \delta_2, \delta_1)$). Analizując graf dowodu rozumowania, w którym paczki zostały wyodrębnione za pomocą drugiej metody (rys. 3.5), możemy zauważyć, że aplikacja tej metody wiąże się z dodatkowym częściowym „oderwaniem" paczki od kontekstu rozumowania zawierającego tę paczkę, w szczególności od wybranych jej przesłanek (rys. 3.5, wyodrębnienie paczek z rozumowania przedstawionego na wydruku 3.2 za pomocą drugiej metody zostało przedstawione na wydruku 3.4). Oderwanie paczki od jej przesłanek wiąże się jednak z:

(*i*) przytoczeniem *explicite* wszystkich jej przesłanek w postaci założenia implikacji abstrachującej rozumowanie zawarte w paczce w sformułowaniu lematu (np. przesłanki α w implikacji $\alpha \Longrightarrow (\gamma \wedge \delta_2)$ na rys. 3.5)

(*ii*) oraz utworzeniem dodatkowego kroku rodzaju $<$*Assumption*$>$, który odpowiada za eliminację implikacji w uzasadnieniu lematu (np. kroki α', $(\gamma \wedge \delta_2)'$ na rys. 3.5).

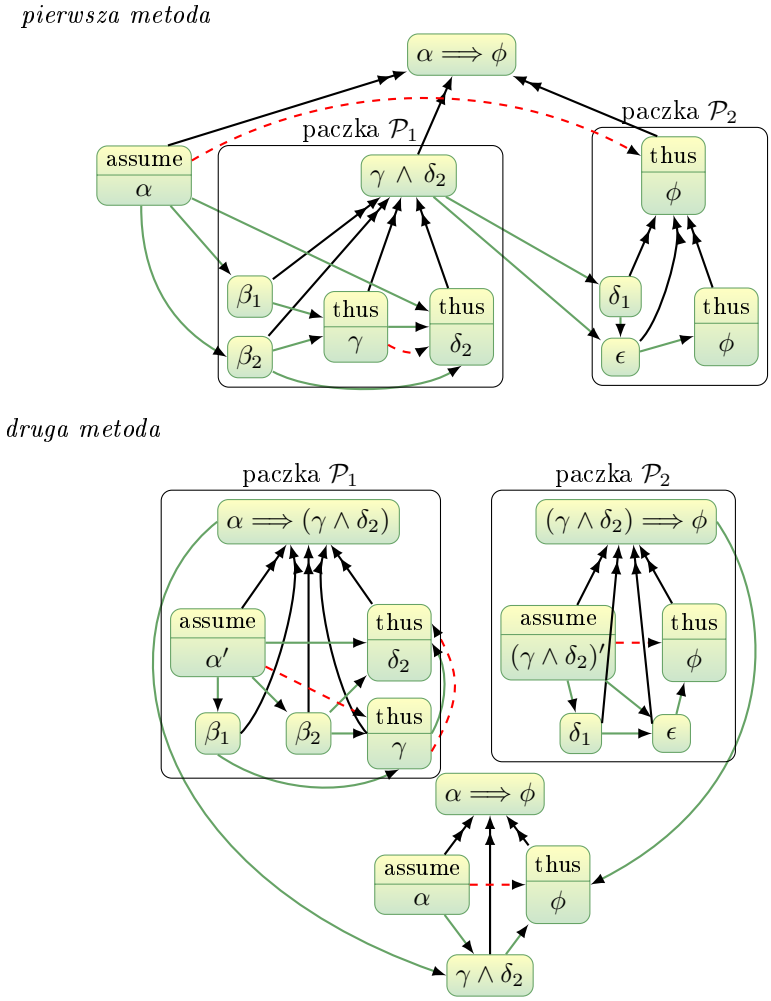
```
α ⟹ φ
proof
  assume A: α;
  New1: γ ∧ δ2
  proof
    B1: β1 by A;
    B2: β2 by A;
    thus C: γ by B1, B2;
    thus D2: δ2 by A, B2, C;
  end;
  thus F: φ
  proof
    D1: δ1 by New1;
    E: ϵ by D1, New1;
    thus F′: φ by E;
  end;
end;
```

Wydruk 3.3: Modyfikacja rozumowania z wydruku 3.2 będąca następstwem wyodrębnienia paczek $\mathcal{P}_1$, $\mathcal{P}_2$ za pomocą pierwszej metody wyodrębniania w grafie dowodu rys. 3.1, którego struktura została przedstawiona na rys. 3.5.

```
Lemma1: α ⟹ (γ ∧ δ2)
proof
  assume A′: α;
  B1: β1 by A′;
  B2: β2 by A′;
  thus C: γ by B1, B2;
  thus D2: δ2 by A, B2, C;
end;

Lemma2: (γ ∧ δ2) ⟹ φ
proof
  assume C∧D2: γ ∧ δ2;
  D1: δ1 by C∧D2;
  E: ϵ by D1, C∧D2;
  thus F′: φ by E;
end;

α ⟹ φ
proof
  assume A: α;
  New1: γ ∧ δ2 by Lemma1;
  thus F: φ by New1, Lemma2;
end;
```

Wydruk 3.4: Modyfikacja rozumowania z wydruku 3.2 będąca następstwem wyodrębnienia paczek $\mathcal{P}_1$, $\mathcal{P}_2$ za pomocą pierwszej metody wyodrębniania w grafie dowodu rys. 3.1, którego struktura została przedstawiona na rys. 3.5.

Wówczas łuki referencyjne, które odwoływały się uprzednio do przesłanek paczki, po zmodyfikowaniu odwołują się do nowo utworzonych kroków rodzaju $<Assumption>$. Podobnie jak w przypadku pierwszej metody, paczki wyodrębnione z rozumowania, zostały zastąpione nowymi krokami $\gamma \wedge \delta_2$, ϕ stwierdzającymi konkluzje lematów wygenerowanych na podstawie paczek, z tą różnicą, że każdy nowy krok rozumowania odwołuje się zarówno do wygenerowanego lematu, jak i przesłanek paczki.



Rysunek 3.5: Modyfikacje abstrakcyjnego grafu dowodu, które reprezentują dwie metody wyodrębniania paczek w grafie przedstawionym na rys. 3.1.

Krótka analiza zmodyfikowanych grafów dowodu prowadzi więc do wniosku, że pierwsza metoda wyodrębniania dostarcza czytelnikowi informacji o konkluzjach uzyskanych wewnątrz paczki. Ukrywa ona jednak informacje o przesłankach paczki, które czytelnik może odnaleźć samodzielnie, badając wszystkie łuki referencyjne prowadzące poza obszar rozumowań wygenerowanych na postawie tej paczki. Naturalnie, jeśli liczba wykorzystywanych przesłanek jest mała, rozsądnym wydaje się przytoczenie ich *explicite* w stwierdzeniu opisującym rozumowanie zawarte w paczce, tak jak ma to miejsce w przypadku drugiej metody. Aplikacja drugiej metody generuje bowiem

lemat, z którym związane sformułowanie ma postać implikacji, w której założeniu pojawia się koniunkcja stwierdzeń związanych z przesłankami paczki, natomiast tezie – koniunkcja stwierdzeń związanych z konkluzjami paczki. Przytoczenie założeń wiąże się jednak z wydłużeniem rozumowania, wynikającym z wprowadzenia dodatkowego wierzchołka odpowiadającego za eliminację implikacji w uzasadnieniu lematu.

Względny wzrost liczby wierzchołków w przypadku wyizolowywania dużych paczek jest niewielki w stosunku do korzyści, jakich dostarcza kapsułkowanie rozbudowanych szczegółów dowodu w procesie poprawy czytelności. W przypadku krótkich, kilkuelementowych paczek koszt ten może jednak wydawać się zbyt duży w stosunku do uzyskanych rezultatów i stawia pod znakiem zapytania sens wyodrębniania tego rodzaju lematów–paczek. Nawet tak krótkie lematy odgrywają jednak istotną rolę w procesie wyszukiwania powtarzających się prostych podrozumowań, które niejednokrotnie występują w długich skryptach dowodowych. Warto w tym miejscu podkreślić, że takie wydzielanie umożliwia nie tylko eliminację powtarzających się krótkich podrozumowań, różniących się co najwyżej identyfikatorami zmiennych, ale również podrozumowań, które są opisywane przez stwierdzenia równoważne. Dodatkowo eliminacja powtarzających się rozumowań w bazie MML, jest jednym z pożądanych kierunków dalszych badań nad poprawą jakości tej bazy [34, 35].

3.1.2 Metody wyodrębniania paczek nie domkniętych na prowadzenie dróg skierowanych

W przeprowadzonej dotąd analizie, wybrane do wyodrębniania paczki pokrywały się ze spójnymi (w pewnym sensie) fragmentami rozumowania. Oczywiście wybrane paczki były wierzchołkowo rozłączne, ale posiadały one również własność *domknięcia ze względu na prowadzenie dróg skierowanych*, tj. każda $\mathcal{R}ef_{\mathfrak{P}}$–droga skierowana łącząca dowolne dwa wierzchołki z obszaru paczki była wierzchołkowo zawarta w obszarze tej paczki. Ujmujemy tę kluczową własność w precyzyjnej definicji.

Definicja 3.6. *Niech $\mathcal{P}$ będzie paczką w abstrakcyjnym grafie dowodu $\mathfrak{P}$. Wówczas paczkę $\mathcal{P}$ będziemy nazywać* domkniętą ze względu na prowadzenie dróg skierowanych *lub krócej* domkniętą *wtedy i tylko wtedy, gdy*

$$\forall_{u,v\in\mathcal{V}(\mathcal{P}),\, w\in\mathcal{V}(\mathfrak{P})} (u \underset{\mathfrak{D}^{\sim}_{\mathfrak{P}}}{\rightarrow}{}^{*} w \underset{\mathfrak{D}^{\sim}_{\mathfrak{P}}}{\rightarrow}{}^{*} v \Longrightarrow w \in \mathcal{V}(\mathcal{P})). \tag{3.1}$$

W dalszym ciągu tej sekcji skupimy uwagę na wpływie własności domkniętości paczki na postać stwierdzenia dowodzonego przez rozumowanie zawarte w paczce. Rozważmy w tym celu paczkę $\mathcal{P}_3$ (rys. 3.1) zbudowaną z wierzchołków: β_1, β_2, δ_1, δ_2, dla której możemy wskazać drogę skierowaną $\beta_1 \rightarrow \gamma \rightarrow \delta_2$ „wychodzącą" poza obszar $\mathcal{P}_3$ ($\gamma \notin \mathcal{O}bsz(\mathcal{P}_3)$). Zauważmy, że aplikując którąkolwiek z metod wyodrębnienia do paczki $\mathcal{P}_3$, uzyskujemy cykl skierowany w grafie dowodu, odkąd co najmniej jedna z przesłanek ($\gamma \in \mathcal{P}re(\mathcal{P}_3)$) jest osiągalna z jakiejś konkluzji ($\beta_1 \in \mathcal{C}on(\mathcal{P}_3)$). Zastąpienie paczki $\mathcal{P}_3$ krokiem stwierdzającym formułę $\beta_1 \wedge \beta_2 \wedge \delta_1 \wedge \delta_2$ powoduje bowiem, że każdy bezpośredni następnik konkluzji paczki (np. γ będąca bezpośrednim następnikiem β_1) jest osiągalny w domkniętym grafie dowodu z każdej przesłanki paczki (np. γ).

Dysponując jednak opisem rozumowania zawartym w paczce $\mathcal{P}_3$ w postaci $(\alpha \Longrightarrow \beta_1) \wedge (\alpha \Longrightarrow \beta_2) \wedge ((\alpha \wedge \gamma) \Longrightarrow \delta_2) \wedge (\gamma \Longrightarrow \delta_1)$ możemy, jak zostanie pokazane w dalszej części podrozdziału 3.1, wyodrębnić tę paczkę zachowując poprawność rozumowania (wyodrębnienie paczki $\mathcal{P}_3$ w grafie dowodu rys. 3.1 oraz modyfikacja skryptu dowodowego z wydruku 3.2, zostało przedstawione na rys. 3.7,

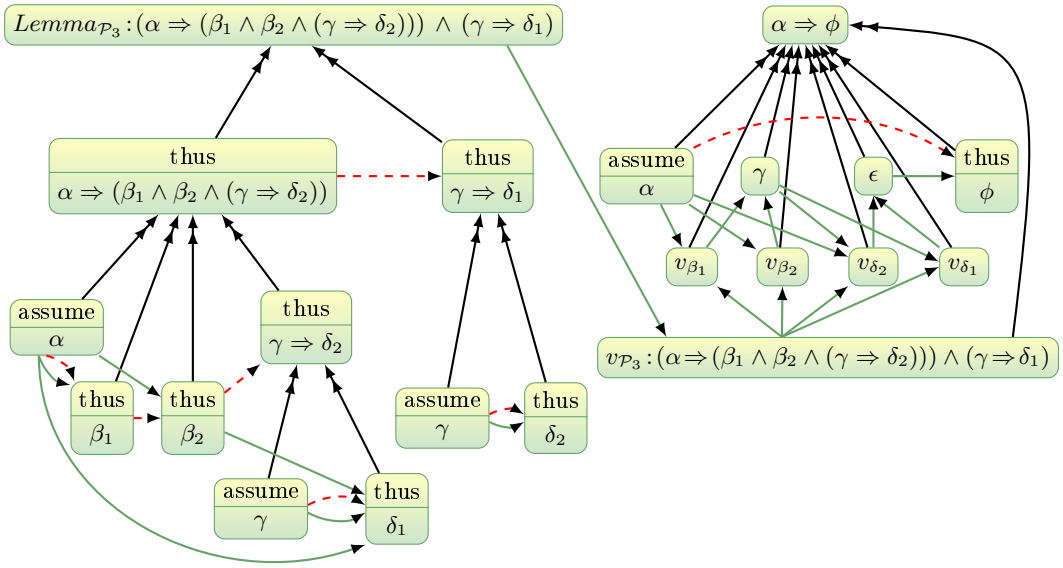
```
Lemma_{P_3}: (α ⟹ (β_1 ∧ β_2 ∧ (γ ⟹ δ_2))) ∧ (γ ⟹ δ_1)
proof
  thus α ⟹ (β_1 ∧ β_2 ∧ (γ ⟹ δ_2))
  proof
    assume A': α;
    thus B1: β_1 by A';
    thus B2: β_2 by A';
    thus γ ⟹ δ_1
    proof
      assume C':γ;
      thus δ_1 by C';
    end;
  end;
end;

α ⟹ φ
proof
  assume A: α;
  New_{P_3}: (α ⟹ (β_1 ∧ β_2 ∧ (γ ⟹ δ_2))) ∧ (γ ⟹ δ_1) by Lemma_{P_3};
  New_{β_1}: β_1 by New_{P_3}, A;
  New_{β_2}: β_2 by New_{P_3}, A;
  C: γ by New_{β_1}, New_{β_2};
  New_{δ_1}: δ_1 by New_{P_3}, A, C;
  New_{δ_2}: δ_2 by New_{P_3}, A, C;
  E: ϵ by New_{δ_1}, New_{δ_2};
  thus F: φ by E;
end;
```

Wydruk 3.6: Modyfikacja rozumowania z wydruku 3.2 będąca następstwem wyodrębnienia niedomkniętej paczki $\mathcal{P}_3$ na prowadzenie dróg skierowanych w grafie dowodu rys. 3.1, którego struktura została przedstawiona na rys. 3.7.

wydruk 3.6 odpowiednio). Zauważmy w tym celu, że rozumowanie zawarte w dowolnej paczce $\mathcal{P}$ może zostać opisane przez koniunkcję implikacji, gdzie następnik każdej spośród implikacji odpowiada stwierdzeniu związanemu z ustaloną konkluzją $c \in \mathcal{C}on(\mathcal{P})$, a poprzednik jest koniunkcją stwierdzeń najmniejszego zbioru (w sensie zawierania) przesłanek paczki $\mathcal{P}$, które zostały wykorzystane do uzasadnienia tej konkluzji. W szczególności, jeśli najmniejszy zbiór przesłanek jest pusty, będziemy przyjmować, że przesłanką implikacji jest verum, które w języku Mizar jest zapisywane jako `not contradiction`. Każdą taką implikację będziemy nazywać *bazową*, a koniunkcję wszystkich implikacji bazowych ustalonej paczki $\mathcal{P}$, jej *stwierdzeniem bazowym*. Naturalnie wybierając formułę opisującą rozumowanie zawarte w paczce, możemy wybrać każdą formułę, która jest równoważna ze stwierdzeniem bazowym (np. $(\alpha \Longrightarrow (\beta_1 \wedge \beta_2 \wedge (\gamma \Longrightarrow \delta_2))) \wedge (\gamma \Longrightarrow \delta_1)$). Formuły, które są równoważne ze stwierdzeniem bazowym, będziemy nazywać *zmodyfikowanymi stwierdzeniami bazowymi*.

Definicja 3.7. *Niech c będzie konkluzją paczki $\mathcal{P}$ w abstrakcyjnym grafie dowodu $\mathfrak{P}$. Zbiorem przesłanek konkluzji c w paczce $\mathcal{P}$ będziemy nazywać wyrażenie*

$$\mathcal{P}(c) := \{v \in \mathcal{P}re(\mathcal{P}) : \underset{u \in \mathcal{V}(\mathcal{P})}{\exists} v \underset{\mathfrak{D}^{\sim}_{\mathfrak{P}}}{\rightarrow} u \xrightarrow[\mathfrak{D}^{\sim}_{\mathfrak{P}|\mathcal{V}(\mathcal{P})}]{}{}^{*} c\}. \tag{3.2}$$



Rysunek 3.7: Modyfikacja abstrakcyjnego grafu dowodu (rys. 3.1), wynikająca z wyodrębnienia paczki $\mathcal{P}_3$ przy użyciu modyfikacji drugiej metody wyodrębniania.

Implikacją bazową *konkluzji c w paczce $\mathcal{P}$ będziemy wówczas nazywać implikację, w której konkluzją jest stwierdzenie związane z c, a przesłanką jest koniunkcja stwierdzeń związanych z wierzchołkami należącymi do zbioru $\mathcal{P}(c)$ lub* `not contradiction` *w przypadku $\mathcal{P}(c) = \emptyset$.*

Można innymi słowy określić zbiór $\mathcal{P}(c)$ jako zbiór tych kroków, z których bezpośrednio korzysta paczka, ale tak, że prowadzą one w końcu do konkluzji c.

Definicja 3.8. *Niech $\mathcal{P}$ będzie paczką w abstrakcyjnym grafie dowodu.* Stwierdzeniem bazowym *paczki $\mathcal{P}$ nazywamy koniunkcję zbudowaną z wszystkich implikacji bazowych, odpowiadających poszczególnym konkluzjom paczki $\mathcal{P}$.*

Wyodrębnienie paczek nie mających własności domkniętości na prowadzenie dróg skierowanych w abstrakcyjnym grafie dowodu wpływa nie tylko na rozbudowanie stwierdzenia opisującego rozumowanie zawarte w paczce, ale wymusza również bardziej zaawansowaną modyfikację grafu dowodu. Zauważmy, że rozumowanie uzasadniające stwierdzenie związane z taką paczką, musi być zbudowane z serii odrębnych dowodów uzasadniających poszczególne implikacje bazowe (np. $\gamma \Longrightarrow \delta_1$ rys. 3.7). Takiemu rozwiązaniu towarzyszy na ogół powielanie części wierzchołków zawartych w paczce, ponieważ dowody poszczególnych implikacji bazowych nie muszą być rozłączne, w sensie podzbiorów zbioru wszystkich kroków paczki, które zostały wykorzystane do uzasadnienia poszczególnych implikacji. Zauważmy bowiem, że dowód implikacji bazowej konkluzji $c \in \mathcal{C}on(\mathcal{P})$, zawiera wierzchołek v jeśli istnieje wierzchołek u, który jest osiągalny z jakiegoś wierzchołka należącego do $\mathcal{P}(c)$, c jest osiągalne z u w $\mathfrak{D}_{\widetilde{\mathfrak{P}}}$ oraz $v \in \mathcal{O}bsz(u)$ $(\{v \in \mathcal{O}bsz(\mathcal{P}) : \underset{p \in \mathcal{P}(c), w \in \mathcal{O}bsz(\mathcal{P})}{\exists} p \underset{\mathfrak{D}_{\widetilde{\mathfrak{P}}}}{\rightarrow}{}^* w \underset{\mathfrak{D}_{\widetilde{\mathfrak{P}}}}{\rightarrow}{}^* c \wedge v \underset{\mathfrak{Meta}_{\widetilde{\mathfrak{P}}}}{\rightarrow}{}^* w\})$ oraz możliwe inne wierzchołki, które należą do $\mathcal{O}bsz(\mathcal{P})$ i są wykorzystywane w uzasadnieniu u, ale nie są osiągalne z żadnej przesłanki paczki $\mathcal{P}$, tzw. *wierzchołki*

swobodne, których uzasadnienie wynika z własności termów oraz predykatów występujących w stwierdzeniach związanych z tymi wierzchołkami ($\{v \in \mathcal{O}bsz(\mathcal{P}) : (\underset{w\in\mathcal{O}bsz(\mathcal{P})}{\exists} v \underset{\mathfrak{M}eta_{\mathfrak{P}}^{\sim}}{\rightarrow}{}^{*} w \underset{\mathfrak{D}_{\mathfrak{P}}^{\sim}}{\rightarrow}{}^{*} c) \wedge (\underset{p\in\mathcal{P}re(\mathcal{P})}{\forall} \neg p \underset{\mathfrak{D}_{\mathfrak{P}}^{\sim}}{\rightarrow}{}^{*} v)\}$). Odpowiedni wybór zmodyfikowanego stwierdzenia bazowego umożliwia jednak zmniejszenie liczby powielonych wierzchołków. Staje się to możliwe dzięki połączeniu odpowiednich implikacji o identycznych założeniach (m.in. w oparciu o prawo kompozycji następników koniunkcji) oraz dzięki wyodrębnianiu powtarzających się przesłanek (w oparciu o prawo eksportacji, np. wybór formuły $\alpha \Longrightarrow (\beta_1 \wedge \beta_2 \wedge (\gamma \Longrightarrow \delta_2))$ równoważnej z $(\alpha \Longrightarrow \beta_1) \wedge (\alpha \Longrightarrow \beta_2) \wedge ((\alpha \wedge \gamma) \Longrightarrow \delta_2)$ (rys. 3.7), umożliwił zastąpienie trzech wierzchołków szkieletowych „`assume` α" jednym).

Innym następstwem wyodrębniania niedomkniętych paczek, jest bardziej zaawansowana modyfikacja pozostałej po wyodrębnianiu części rozumowania pierwotnego. W celu przedstawienia tych modyfikacji ustalmy dowolną paczkę $\mathcal{P}$ oraz załóżmy, że $\mathcal{C}on(\mathcal{P}) = \{c_1, c_2, \ldots, c_n\}$, gdzie $n = |\mathcal{C}on(\mathcal{P})|$. Oznaczmy zbiór

$$\bigcap \mathcal{P} := \bigcap_{1\leqslant i\leqslant n} \mathcal{P}(c_i) \tag{3.3}$$

oraz wprowadźmy oznaczenie $\mathfrak{sta}(v)$ na stwierdzenie sformułowane w wierzchołku $v \in \mathcal{V}(\mathfrak{P})$ oraz $\mathfrak{lab}(v)$ na etykietę umożliwiającą odwołanie się do stwierdzenia $\mathfrak{sta}(v)$. Dodatkowo przez $\mathfrak{sta}(V)$ będziemy oznaczać koniunkcję wszystkich stwierdzeń sformułowanych w wierzchołkach ze zbioru V, a przez $\mathfrak{lab}(V)$ ciąg wszystkich etykiet stwierdzeń sformułowanych w wierzchołkach ze zbioru V oddzielanych przecinkami, gdzie $\emptyset \neq V \subseteq \mathcal{V}(\mathfrak{P})$. Dodatkowo, w przypadku $\emptyset = V$ będziemy przyjmować: $\mathfrak{sta}(V) =$ `not contradiction`, a za $\mathfrak{lab}(V)$ etykietę dodatkowego kroku stwierdzającego `not contradiction`.

Ze względu na częściowe wykorzystanie rozwiązań dotyczących drugiej metody wyodrębniania w pierwszej, metody te będą rozważane w kolejności odwrotnej.

Druga metoda W przypadku drugiej zmodyfikowanej metody wyodrębniania nowy krok związany z wierzchołkiem $v_{\mathcal{P}}$, który zastępuje paczkę $\mathcal{P}$, ma postać:

$$\mathfrak{lab}(v_{\mathcal{P}})\texttt{:} \bigwedge_{1\leqslant i\leqslant n} \texttt{(}\, \mathfrak{sta}(\mathcal{P}(c_i) \setminus \bigcap \mathcal{P}) \,\texttt{implies}\, \mathfrak{sta}(c_i) \,\texttt{)}\, \texttt{by}\, \mathfrak{lab}(\bigcap \mathcal{P})\texttt{,}\, \mathfrak{lab}(Lemma_{\mathcal{P}})\texttt{;} \tag{3.4}$$

gdzie $v_{Lemma\mathcal{P}}$ oznacza wierzchołek odpowiadający krokowi rozumowania postaci (3.5), w którym został sformułowany lemat wygenerowany na podstawie paczki $\mathcal{P}$, a $\mathfrak{lab}(v_{\mathcal{P}})$ oraz $\mathfrak{lab}(v_{Lemma\mathcal{P}})$ są nie wykorzystywanymi dotąd etykietami w dowodzie.

$$\mathfrak{lab}(v_{Lemma\mathcal{P}})\texttt{:}\, \mathfrak{sta}(\bigcap \mathcal{P})\, \texttt{implies} \left(\bigwedge_{1\leqslant i\leqslant n} \texttt{(}\, \mathfrak{sta}(\mathcal{P}(c_i) \setminus \bigcap \mathcal{P}) \,\texttt{implies}\, \mathfrak{sta}(c_i) \,\texttt{)} \right) \texttt{proof ... end;}, \tag{3.5}$$

Zauważmy, że w pierwotnej wersji drugiej metody, zastąpienie poszczególnych odwołań do różnych konkluzji paczki, odwołaniem do nowego wspólnego kroku zastępującego paczkę umożliwiało zachowanie poprawności modyfikowanych w ten sposób uzasadnień. Sytuacja ta ulega jednak istotnej zmianie w przypadku wierzchołka $v_{\mathcal{P}}$. Zastąpienie odwołania do konkluzji c_i przez odwołanie do $v_{\mathcal{P}}$ jest bowiem uzasadnione jedynie w przypadku $\mathcal{P}(c_i) \setminus \bigcap \mathcal{P} = \emptyset$, gdzie $1 \leqslant i \leqslant n$. W celu rozwiązania tego problemu wprowadźmy do rozumowania krok związany z nowym wierzchołkiem v_{c_i}, odpowiadający konkluzji c_i postaci:

$$\mathfrak{lab}(c_i)\texttt{:}\, \mathfrak{sta}(c_i)\, \texttt{by}\, \mathfrak{lab}(\mathcal{P}(c_i) \setminus \bigcap \mathcal{P})\texttt{,}\, \mathfrak{lab}(v_{\mathcal{P}})\texttt{;}, \tag{3.6}$$

gdzie $\mathfrak{lab}(c_i)$ jest nie wykorzystywaną dotąd etykietą w dowodzie. Oczywiście zastąpienie każdego wystąpienia etykiety $\mathfrak{lab}(c_i)$ przez etykietę $\mathfrak{lab}(v_{c_i})$, nie powoduje błędów we w ten sposób zmodyfikowanych uzasadnieniach. Dodatkowo, przeprowadzone empiryczne badania nad uzyskanymi modyfikacjami wykazały, że wykorzystując istniejące w systemie Mizar programy, tj. RelPrem, RelInfer, Inacc [57, 59, 78], RenInfer, MergeItems [70], które szerzej są opisane w sekcji 3.3.1, możliwe jest „usunięcie" części wierzchołków postaci (3.6) lub ich „pogrupowanie" przy zachowaniu poprawność rozumowania.

Przedstawiony sposób modyfikacji uzasadnień poszczególnych kroków rozumowania pierwotnego po wyodrębnieniu paczki $\mathcal{P}$ nie budzi wątpliwości w zakresie poprawności tak zmodyfikowanych uzasadnień poszczególnych kroków. Równie intuicyjne jest uzasadnienie stwierdzenia bazowego w oparciu o ciąg kroków stwierdzających kolejne implikacje bazowe, które zostały uzasadnione na podstawie odpowiednio wybranych fragmentów paczki.

W celu uzasadnienia zachowywania poprawności rozumowania przez zmodyfikowaną drugą metodę pokażemy jedynie, że modyfikacja ta zachowuje własność acykliczności.

Obserwacja 3.9. *Modyfikacja rozumowania pierwotnego, będąca następstwem wyodrębnienia paczki za pomocą zmodyfikowanej drugiej metody, nie generuje cykli skierowanych w uzyskanej strukturze grafu dowodu.*

Dowód. Ustalmy abstrakcyjny graf skierowany $\mathfrak{P}$ oraz jego modyfikację $\mathfrak{P}_1$ związaną z wyodrębnieniem paczki $\mathcal{P}$. Załóżmy nie wprost, że $\mathfrak{a}$ jest skierowanym cyklem w $\mathfrak{G}_{\mathfrak{P}_1}$ (zob. Def. 2.5). Z Def. 2.5 digraf $\mathfrak{G}_{\mathfrak{P}}$ jest acykliczny, skąd cykl $\mathfrak{a}$ musi przechodzić przez „nowe" wierzchołki związane z krokami rozumowania postaci: (3.4), (3.6), $Lemma_{\mathcal{P}}$. W celu uzyskania sprzeczności pokażemy, że każdy maksymalny w sensie zawierania segment cyklu $\mathfrak{a}$ zbudowany wyłącznie z „nowych" wierzchołków może zostać zastąpiony drogą zbudowaną z wierzchołków należących do $\mathcal{O}bsz(\mathcal{P})$, w taki sposób, że po iteracyjnym zastąpieniu poszczególnych maksymalnych segmentów określonego rodzaju, uzyskany ciąg będzie cyklem, co doprowadzi do sprzeczności z wynikającą z definicji acyklicznością $\mathfrak{G}_{\mathfrak{P}}$.

Ustalmy więc segment $\mathfrak{a}' := \langle a_0, a_1, a_2, \ldots, a_k, a_{k+1}\rangle$ drogi $\mathfrak{a}$, gdzie a_i są „nowymi" wierzchołkami dla $i = 1, 2, \ldots, k$, a $a_0, a_{k+1} \in \mathcal{V}(\mathfrak{P})$. Wówczas możliwe są tylko dwa przypadki, gdyż $v_{Lemma_{\mathcal{P}}}$ jako źródło w $\mathfrak{D}_{\widetilde{\mathfrak{P}_1}}$ nie może należeć do $\mathfrak{a}'$ (rys. 3.7):

1. segment $\mathfrak{a}' = \langle a_0, a_1, a_2, a_3\rangle$, gdzie
   - $a_0 \in \bigcap\limits_{d \in \mathcal{C}on(\mathcal{P})} \mathcal{P}(d)$,
   - $a_1 = v_{\mathcal{P}}$,
   - $a_2 = v_c$ dla pewnej konkluzji $c \in \mathcal{C}on(\mathcal{P})$,
   - a_3 odwołuje się do c,
2. segment $\mathfrak{a}' = \langle a_0, a_1, a_2, a_2\rangle$, gdzie
   - $a_0 \in \mathcal{P}(c)$ dla $c \in \mathcal{C}on(\mathcal{P})$, ale $a_0 \notin \bigcap\limits_{d \in \mathcal{C}on(\mathcal{P})} \mathcal{P}(d)$,
   - $a_1 = v_c$,
   - a_2 odwołuje się do c.

Jednak w obu przypadkach $a_0 \in \mathcal{P}(c)$, a zatem z Def. 3.7 konkluzja c jest osiągalna z przesłanki a_0 w $\mathfrak{D}_{\widetilde{\mathfrak{P}}_{|\mathcal{V}(\mathcal{P})}}$. Możemy zatem segment miedzy a_0 i a_{k+1} zastąpić wierzchołkami z pierwotnego grafu, zachowując osiągalność a_{k+1} z a_0. □

Pierwsza metoda Zauważmy, że w przypadku drugiej metody wyodrębniania generowanie lematu o możliwie najogólniejszej postaci kosztem powielania wierzchołków w uzasadnieniu było uzasadnione tym, iż „położenie" lematu w skrypcie dowodowym umożliwiało jego zastosowanie nie tylko w uzasadnieniu kroku zastępującego paczkę, ale również potencjalnie w uzasadnieniu innych wierzchołków. Koszt ten jest jednak nieuzasadniony w przypadku pierwszej metody, gdzie wygenerowane rozumowanie jest wykorzystane tylko do uzasadnienia tego kroku. Oczywiście, prowadząc analogiczne rozważania jak w przypadku drugiej metody,

(*i*) zastępując krok związany z wierzchołkiem $v_{\mathcal{P}}$ krokiem $v'_{\mathcal{P}}$ postaci:

$$\mathfrak{lab}(v'_{\mathcal{P}})\texttt{:}\bigwedge_{1\leqslant i\leqslant n}\texttt{(}\,\mathfrak{sta}(\mathcal{P}(c_i)\setminus\bigcap\mathcal{P})\ \texttt{implies}\ \mathfrak{sta}(c_i)\,\texttt{)}\ \texttt{proof}\ \ldots\ \texttt{end;}, \tag{3.7}$$

gdzie `proof ... end;` jest rozumowaniem wykorzystywanym w uzasadnieniu kroku $v_{Lemma\mathcal{P}}$ o postaci (3.5),

(*ii*) usuwając z tego rozumowania krok rodzaju $<Assumption>$, który odpowiada za eliminację implikacji o założeniem $\mathfrak{sta}(\bigcap\mathcal{P})$ – implikacja ta nie występuje w $\mathfrak{sta}(v'_{\mathcal{P}})$,

(*iii*) zastępując odwołania do usuniętego kroku bezpośrednimi odwołaniami do przesłanek ze zbioru $\bigcap\mathcal{P}$ w uzasadnieniu kroku $v'_{\mathcal{P}}$,

uzyskujemy zachowanie poprawności zmodyfikowanego rozumowania.

W sekcji tej skupimy uwagę nad sposobem modyfikacji zbiorów przesłanek $\mathcal{P}re(c)$, dzięki któremu wygenerowane rozumowanie na podstawie paczki nie będzie posiadać powielonych wierzchołków, przy jednoczesnym zachowaniu poprawności zmodyfikowanego w ten sposób rozumowania, gdzie $c\in\mathcal{C}on(c)$. Cel ten zostanie osiągnięty poprzez odpowiednie osłabienie stwierdzenia bazowego.

Definicja 3.10. *Niech $\mathfrak{P}$ będzie abstrakcyjnym grafem dowodu, V podzbiorem zbioru wierzchołków w $\mathfrak{P}$, zaś A podzbiorem zbioru łuków $\mathcal{A}(\mathfrak{D}_{\widetilde{\mathfrak{P}}})$ oraz v,u parą wierzchołków w $\mathfrak{P}$. Będziemy mówić, że wierzchołek u jest* osiągalny *z v* względem zbiorów *A, V, jeśli spełniona jest następująca zależność:*

$$\left(\exists_{w_1,w_2,\ldots,w_k\in V} v\underset{A}{\rightarrow}w_1\underset{A}{\rightarrow}w_2\underset{A}{\rightarrow}\ldots\underset{A}{\rightarrow}w_k\underset{A}{\rightarrow}u\right)\wedge$$
$$\left(\forall_{s_1,s_2,\ldots,s_l\in\mathcal{V}(\mathfrak{P})} v\underset{A}{\rightarrow}s_1\underset{A}{\rightarrow}s_2\underset{A}{\rightarrow}\ldots\underset{A}{\rightarrow}s_k\underset{A}{\rightarrow}u\Longrightarrow s_1,s_2,\ldots,s_k\in V\right) \tag{3.8}$$

i będziemy oznaczać $v\overset{V}{\underset{A}{\longmapsto}}u$.

Ustalmy paczkę $\mathcal{P}$. Wierzchołki zbioru $\mathcal{PC}:=\mathcal{P}re(\mathcal{P})\cup\mathcal{C}on(\mathcal{P})$ możemy wówczas przedstawić w postaci dwóch ciągów zbiorów: $\{\mathcal{P}re(\mathcal{P})^i\}_{i=1}^{\infty}$, $\{\mathcal{C}on(\mathcal{P})^i\}_{i=1}^{\infty}$ danych zależnościami:

$$\begin{aligned}
\mathcal{P}re(\mathcal{P})^0 &= \{p\in\mathcal{P}re(\mathcal{P}):\neg\exists_{u\in\mathcal{C}on(\mathcal{P})}\,u\underset{\mathfrak{D}_{\widetilde{\mathfrak{P}}}}{\rightarrow}{}^{*}p\},\\
\mathcal{C}on(\mathcal{P})^i &= \{c\in\mathcal{C}on(\mathcal{P}):\exists_{p\in\mathcal{P}re(\mathcal{P})^i}\,p\xmapsto[\mathcal{A}(\mathfrak{D}_{\widetilde{\mathfrak{P}}})]{\mathcal{O}bsz(\mathcal{P})}c\},\\
\mathcal{P}re(\mathcal{P})^{i+1} &= \{p\in\mathcal{P}re(\mathcal{P}):\exists_{c\in\mathcal{C}on(\mathcal{P})^i}\,c\xmapsto[\mathcal{A}(\mathfrak{D}_{\widetilde{\mathfrak{P}}})]{\mathcal{V}(\mathfrak{P})\setminus\mathcal{O}bsz(\mathcal{P})}p\},
\end{aligned} \tag{3.9}$$

gdzie i jest dowolną liczbą naturalną. Oczywiście, elementy ciągów $\mathcal{P}re(\mathcal{P})^0, \mathcal{C}on(\mathcal{P})^0$, $\mathcal{P}re(\mathcal{P})^1, \mathcal{C}on(\mathcal{P})^1, \mathcal{P}re(\mathcal{P})^2, \mathcal{C}on(\mathcal{P})^2, \ldots$ stanowią partycję zbioru $\mathcal{PC}$. Dodatkowo, tylko skończona liczba elementów może być niepusta, gdyż $\mathcal{PC}$ jest zbiorem skończonym. Ponadto istnieje liczba naturalna m, dla której $\mathcal{C}on(\mathcal{P})^0 \neq \emptyset$, $\mathcal{P}re(\mathcal{P})^i \neq \emptyset$, $\mathcal{C}on(\mathcal{P})^i \neq \emptyset$ dla $i = 1, \ldots, m$ oraz $\mathcal{P}re(\mathcal{P})^i = \mathcal{C}on(\mathcal{P})^i = \emptyset$ dla $i > m$, ponieważ dla każdej przesłanki istnieje co najmniej jedna konkluzja osiągalna w $\mathfrak{D}_{\widetilde{\mathfrak{P}}}$ (uwaga, w poniższych rozważaniach w przypadku $\mathcal{P}re(\mathcal{P})^0 = \emptyset$ będziemy przyjmować, że zbiór $\mathcal{P}re(\mathcal{P})^0$ jest zbudowany z wierzchołka stwierdzającego `not contradiction`).

Korzystając następnie bezpośrednio z konstrukcji ciągów (3.9) stwierdzamy, że nie zachodzi $c \underset{\mathfrak{D}_{\widetilde{\mathfrak{P}}}}{\rightarrow}{}^* p$ dla każdej konkluzji $c \in \mathcal{C}on(\mathcal{P})^i$ oraz przesłanki $p \in \mathcal{P}re(\mathcal{P})^j$, gdzie $j \leqslant i$. Stąd partycja zbioru wierzchołków rozumowania pierwotnego zawierającego jako podgraf $\mathcal{P}$, która została zbudowana ze zbiorów $\mathcal{P}re(\mathcal{P})^0, \mathcal{C}on(\mathcal{P})^0, \mathcal{P}re(\mathcal{P})^1$, $\mathcal{C}on(\mathcal{P})^1, \ldots, \mathcal{P}re(\mathcal{P})^m, \mathcal{C}on(\mathcal{P})^m$ oraz jednoelementowych zbiorów zawierających wierzchołki nie należących do $\mathcal{PC}$, wyznacza acykliczną partycję tego grafu dowodu. Możemy więc osłabić stwierdzenie bazowe paczki $\mathcal{P}$ do postaci:

$$\begin{array}{l} \mathfrak{sta}(\mathcal{P}re(\mathcal{P})^0)\ \texttt{implies}\ (\ \mathfrak{sta}(\mathcal{C}on(\mathcal{P})^0)\ \texttt{\&}\ (\\ \quad \mathfrak{sta}(\mathcal{P}re(\mathcal{P})^1)\ \texttt{implies}\ (\ \mathfrak{sta}(\mathcal{C}on(\mathcal{P})^1)\ \texttt{\&}\ (\\ \quad\quad \ldots \\ \quad\quad\quad (\ \mathfrak{sta}(\mathcal{P}re(\mathcal{P})^m)\ \texttt{implies}\ \mathfrak{sta}(\mathcal{C}on(\mathcal{P})^m)\ )\ldots)))), \end{array} \tag{3.10}$$

które jest uzasadnione za pomocą rozumowania:

```
proof
   assume sta(Pre(P)^0);
      „zbiór wierzchołków nie należących do Con(P)^0, dla których istnieją
      konkluzje ze zbioru Con(P)^0, osiągalne z tych wierzchołków w D_P~"
   thus sta(Con(P)^0);
   assume sta(Pre(P)^1);
      „zbiór wierzchołków nie należących do Con(P)^1, dla których istnieją
      konkluzje ze zbioru Con(P)^1, osiągalne z tych wierzchołków w D_P~
      oraz które nie zostały jeszcze wymienione"
   thus sta(Con(P)^1);
   ...
   assume sta(Pre(P)^m);
      „zbiór wierzchołków nie należących do Con(P)^m, dla których istnieją
      konkluzje ze zbioru Con(P)^m, osiągalne z tych wierzchołków w D_P~
      oraz które nie zostały jeszcze wymienione"
   thus sta(Con(P)^m);
end;,
```

(3.11)

w którym odwołania do stwierdzeń sformułowanych w krokach związanych z przesłankami paczki $\mathcal{P}$ zostały odpowiednio zastąpione odwołaniami do kroków rodzaju $<$*Assumption*$>$. Oczywiście konstrukcja tego rozumowania nie wymaga powielania wierzchołków należących do paczki $\mathcal{P}$.

Rozważmy teraz krok postaci (3.7), w którym stwierdzenie bazowe zostało zastąpione jedynie następnikiem implikacji o przesłance $\mathfrak{sta}(\mathcal{P}re(\mathcal{P})^0)$ sformułowanej w (3.10). Uzasadnieniem tego kroku jest wówczas rozumowanie (3.11), w którym został pominięty krok „`assume` $\mathfrak{sta}(\mathcal{P}re(\mathcal{P})^0)$`;`" oraz wszystkie odwołania do tego kroku zostały zastąpione poprzez odwołania do odpowiednich stwierdzeń związanych z krokami, będącymi przesłankami ze zbioru $\mathcal{P}re(\mathcal{P})^0$.

Poprawność zmodyfikowanego rozumowania przy takiej postaci kroku związanego z wierzchołkiem $v_{\mathcal{P}}$ jest osiągana dzięki zastąpieniu każdego wierzchołka v_c postaci (3.6), przez wierzchołek $v_{\mathcal{C}on(\mathcal{P})^i}$ postaci:

$$\mathfrak{lab}(v_{\mathcal{C}on(\mathcal{P})^i})\texttt{:}\ \mathfrak{sta}(c_i)\ \texttt{by}\ \mathfrak{lab}(v_{\mathcal{P}}), \mathfrak{lab}(\mathcal{P}re(\mathcal{P})^1), \mathfrak{lab}(\mathcal{P}re(\mathcal{P})^2), \ldots, \mathfrak{lab}(\mathcal{P}re(\mathcal{P})^{i-1})\texttt{;}, \tag{3.12}$$

gdzie $0 \leqslant i \leqslant n$, $\mathfrak{lab}(c_i)$ jest nie wykorzystywaną dotąd etykietą w dowodzie; oraz zastąpieniu każdego odwołania do stwierdzenia $\mathfrak{sta}(c)$ konkluzji $c \in \mathcal{C}on(\mathcal{P})$ przez odwołanie do stwierdzenia $\mathfrak{sta}(v_{\mathcal{C}on(\mathcal{P})^i})$, jeśli $c \in \mathcal{C}on(\mathcal{P})^i$.

Zauważmy również, że zaproponowane uogólnienie pierwszej metody wyodrębniania paczek przy pomocy osłabionego stwierdzenia bazowego w naturalny sposób uogólnia metodę zaproponowaną dla domkniętych paczek, gdyż w przypadku domkniętej paczki $\mathcal{P}$ ciągi określone zależnością (3.9), posiadają co najwyżej dwa niepuste elementy, mianowicie $\mathcal{P}re(\mathcal{P})^0$, $\mathcal{C}on(\mathcal{P})^0$ oraz $v_{\mathcal{C}on(\mathcal{P})^0} = v_{\mathcal{P}}$.

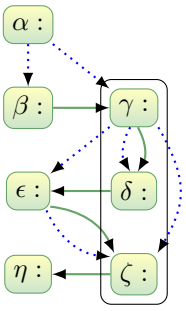
3.1.3 Metody wyodrębniania paczek uwzględniające zmienne w rozumowaniu

W przedstawionych dotąd metodach wyodrębniania paczek, pomijane były informacje opisujące sposób wykorzystania zmiennych w paczkach. Intuicyjnie, wykorzystanie zmiennych, które zostały wprowadzone do rozumowania poza obszarem paczki, ale są wykorzystywane wewnątrz tej paczki powinno wpływać co najwyżej na drugą metodę, która wyodrębnia paczkę w postaci lematu. Oderwanie paczki od kontekstu rozumowania w przypadku drugiej metody wymusza bowiem poprzedzenie stwierdzenia bazowego kwantyfikatorami ogólnymi, umożliwiającymi związanie zmiennych wolnych występujących w tym stwierdzeniu. Sytuacja taka ma jednak miejsce tylko w przypadku, jeśli zmienne ustalone wprowadzone do rozumowania wewnątrz paczki są wykorzystywane tylko w jej obszarze. Dostateczną modyfikacją drugiej metody jest wówczas poprzedzenie stwierdzenia bazowego kwantyfikatorami ogólnymi, a następnie poprzedzeniem wszystkich kroków w uzasadnieniu krokiem rodzaju <*Generalization*>, który odpowiada za wprowadzenie tych zmiennych ustalonych do rozumowania wygenerowanego lematu.

Zbiór zmiennych występujących w stwierdzeniu bazowym, nie musi się jednak pokrywać ze zbiorem zmiennych wykorzystywanych w paczce. Rozważmy bowiem przypadek, w którym zmienna ustalona x typu Θ jest wykorzystywana w rozumowaniu zawartym w paczce, ale nie jest wykorzystana w stwierdzeniu bazowym. Wówczas wykorzystywane są w paczce co najwyżej własności wynikające z istnienia zmiennej określonego typu, skąd możliwe jest zastąpienie wszystkich wystąpień tej zmiennej przez reprezentanta określonego typu (ang. the global choice [38, 51, 100]), który w języku Mizar ma postać $\texttt{the}\,\Theta$. Wykorzystanie reprezentanta, jest jednak możliwe wyłącznie dla typów niepustych – typów dla których wcześniej został udowodniony klaster egzystencjalny (szerzej na ten temat w [33]), aczkolwiek ten sam klaster jest konieczny do wprowadzenia do rozumowania zmiennej ustalonej określonego typu. Tym samym zastąpienie zmiennej x reprezentantem $\texttt{the}\,\Theta$ nie generuje błędów w rozumowaniu, a nawet może je uprościć w przypadku kilku zmiennych ustalonych tego samego typu, które na wskutek zastąpienia tych zmiennych wspólnym reprezentantem zostały utożsamione.

Rozważmy przykład z rys. 3.8. Przedstawiona została tam sytuacja, w której zmienna ustalona wprowadzona do rozumowania wewnątrz paczki, jest wykorzystywana poza jej obszarem, gdzie symbole `P`, `Q`, `R`, `S`, `T` reprezentują predykaty jednoargumentowe, a `F` reprezentuje funktor jednoargumentowy. Zauważmy najpierw, że

```
α :  let x be set;
β :  A1: P[x];
γ :  consider y be set such that
       A2:y=F(x) by A1;
δ :  A3: Q[y] by A2;
ϵ :  consider z be Subset of y such that
       A4: R[z] by A3;
ζ :  consider r be Relation of y,z such that
       A5: S[r] by A4;
η :  A6: T[r] by A5;
```



paczka $\mathcal{P}_4$

Rysunek 3.8: Fragment rozumowania zapisanego w języku Mizar przedstawiający niedomkniętą paczkę $\mathcal{P}_4$. Paczka $\mathcal{P}_4$ jest niedomknięta, gdyż ścieżka $\delta \to \epsilon \to \zeta$ złożona z łuków referencyjnych wychodzi poza jej zakres. Gdyby ten warunek nie zachodził, to niedomknięcie wynikałoby też z obecności ścieżki $\gamma \to \epsilon \to \zeta$ złożonej z łuków porządkujących.

stwierdzenie bazowe paczki $\mathcal{P}_4$ ma postać:

$$(\ P[x]\ \texttt{implies}\ Q[y]\ )\ \texttt{\&}\ (\ R[z]\ \texttt{implies}\ S[r]\ ), \tag{3.13}$$

w którym występują cztery zmienne wolne x, y, z, r odpowiadające zbiorom, przy czym zmienne x i z są zmiennymi ustalonymi wprowadzonymi do rozumowania poza obszarem paczki $\mathcal{P}_4$, natomiast pozostałe zmienne odpowiadają zmiennym ustalonym wprowadzonym wewnątrz obszaru paczki $\mathcal{P}_4$. Stąd, w celu zapewnienia poprawności modyfikowanego rozumowania zmienne x i z w stwierdzeniu opisującym rozumowanie zawarte w paczce, muszą być związane kwantyfikatorami ogólnymi, natomiast y i r egzystencjalnymi. Dodatkowo kolejność wiązania zmiennych w przypadku paczki $\mathcal{P}_4$ jest następstwem dwóch obserwacji:

(i) konstrukcja zmiennej ustalonej y w kroku γ wykorzystuje zmienną ustaloną x, skąd kwantyfikator $\underset{x}{\forall}$ musi poprzedzać kwantyfikator $\underset{y}{\exists}$,

(ii) zmienna ustalona z jest podzbiorem zbioru y oraz zmienna ustalona z jest podzbiorem iloczynu kartezjańskiego $y \times z$, skąd kwantyfikator $\underset{y}{\exists}$ musi poprzedzać kwantyfikator $\underset{z:z\subseteq y}{\forall}$, który to następnie musi poprzedzać $\underset{r:r\subseteq y\times z}{\exists}$.

Generalizując powyższe stwierdzenie możemy zauważyć, że kolejność kwantyfikatorów w sformułowaniu lematu jest jedyną możliwą dla tej paczki. Jest ona określona relacją osiągalności w grafie $\mathfrak{O}_{\mathfrak{P}}$, która w tym przypadku jednoznacznie determinuje kolejność kwantyfikatorów. Relacją osiągalności w grafie $\mathfrak{O}_{\mathfrak{P}}$ determinuje bowiem kolejność kwantyfikatorów z dokładnością do wyboru sortowania topologicznego digrafu rozpiętego na wierzchołkach odpowiadających krokom wprowadzającym zmienne ustalone do rozumowania, które to zmienne są wprowadzone lub wykorzystywane wewnątrz paczki; natomiast łuki łączą pary różnych wierzchołków będących w relacji osiągalności w grafie $\mathfrak{O}_{\mathfrak{P}}$. Stąd jako stwierdzenie opisującą rozumowanie zawarte w paczce $\mathcal{P}_4$, możemy przyjąć formułę:

$$\underset{x}{\forall}\,\underset{y}{\exists}\,\underset{z:z\subseteq y}{\forall}\,\underset{r:r\subseteq y\times z}{\exists}\ ((P(x) \Longrightarrow Q(y)) \wedge (R(z) \Longrightarrow S(r)))\,, \tag{3.14}$$

która to odpowiada stwierdzeniu bazowemu paczki, w którym zostały związane występujące zmienne wolne. Uzasadnienie tej formuły (wydruk 3.9), jak również mody-

```
α0 :   Lemma_P4: for x be set ex y be set st
          for z be Subset of y ex r be Relation of y,z st
             (P[x] implies Q[y]) & (R[z] implies S[r])
       proof
α1 :      let x be set;
α2 :      per cases;
α3 :         suppose A1: P[x];
γ' :            consider y be set such that
                   A2:y=F(x) by A1;
δ' :            A3: Q[y] by A2;
α4 :            take y;
α5 :            let z be Subset of y;
α6 :            per cases;
α7 :               suppose A4: R[z];
ζ' :                  consider r be Relation of y,z such that
                         A5: S[r] by A4;
α8 :                  take r;
α9 :                  thus (P[x] implies Q[y]) & (R[z] implies S[r])
                         by A3, A5;
                   end;
α10 :              suppose A6: not R[z];
α11 :                 take y= the Relation of y,z;
α12 :                 thus (P[x] implies Q[y]) & (R[z] implies S[r])
                         by A3, A6;
                   end;
            end;
α13 :       suppose A7: not P[x];
α14 :          take y = the set;
α15 :          let z be Subset of y;
α16 :          per cases;
α17 :             suppose A8: R[z];
ζ'' :                 consider r be Relation of y,z such that
                         A9: S[r] by A8;
α18 :                 take r;
α19 :                 thus (P[x] implies Q[y]) & (R[z] implies S[r])
                         by A7, A9;
                   end;
α20 :              suppose A10: not R[z];
α21 :                 take y= the Relation of y,z;
α22 :                 thus (P[x] implies Q[y]) & (R[z] implies S[r])
                         by A7, A10;
                   end;
            end;
       end;
```

Wydruk 3.9: Dowód formuły zapisany w systemie Mizar, z wykorzystaniem konstrukcji per cases, powstałej ze stwierdzenia bazowego paczki $\mathcal{P}_4$ (rys. 3.8), w której zmienne wolne zostały związane czterema kwantyfikatorami poprzedzającymi to stwierdzenie. Nowe kroki w dowodzie, które nie posiadają swoich odpowiedników w paczce $\mathcal{P}_4$, zostały sformułowane w wierszach: $\alpha_0, \alpha_1, \ldots, \alpha_{22}$.

```
α :  let x be set;
β :  A1: P[x];
θ1 : consider y be set such that
       B1: for z be Subset of y ex r be Relation of y,z st
         (P[x] implies Q[y]) & (R[z] implies S[r]) by A1, Lemma_P4;
θ3 : consider r1 be Relation of y,the Subset of y such that
       B2: (P[x] implies Q[y]) & (R[the Subset of y] implies S[r1])
         by B1;
δ :  A3: Q[y] by B2, A1;
ϵ :  consider z be Subset of y such that
       A4: R[z] by A3;
θ2 : consider r be Relation of y,z such that
       B3: (P[x] implies Q[y]) & (R[z] implies S[r]) by B1;
η :  A6: T[r] by A4, B3;
```

Wydruk 3.10: Modyfikacja pozostałej części rozumowania (rys. 3.8) po wyodrębnieniu paczki $\mathcal{P}_4$, gdzie etykieta $Lemma_{\mathcal{P}_4}$ jest identyfikatorem stwierdzenia uzasadnionego na wydruku 3.9. Nowe kroki w dowodzie, które zostały dodane do pozostałej części rozumowania po wyodrębnieniu paczki, zostały sformułowane w wierszach: $\theta_1, \theta_2, \theta_3$.

fikacja pozostałej części rozumowania po wyodrębnieniu paczki (wydruk 3.10), jest jednak dość skomplikowana i wymaga wprowadzenie do rozumowania dużej liczby nowych kroków w celu zagwarantowania poprawności modyfikowanego skryptu dowodowego.

Jedną z metod, umożliwiającą zmniejszenie liczby dodatkowych kroków jest osłabienie formuły (3.14), poprzez m.in. „poprzedzenie" każdego kwantyfikatora egzystencjalnego wiążącego pewną zmienną związaną ze stałą wprowadzoną do rozumowania w paczce implikacją, której poprzednikiem jest koniunkcja przesłanek wykorzystywanych do skonstruowania tej zmiennej ustalonej. Wówczas stwierdzenie, które opisuje rozumowanie zawarte w paczce $\mathcal{P}_4$, przyjmuje m.in. poniższą postać:

$$\underset{x}{\forall}(P(x) \Longrightarrow \underset{y}{\exists}(Q(y) \wedge \underset{z:z\subseteq y}{\forall}(R(z) \Longrightarrow \underset{r:r\subseteq y\times z}{\exists} S(r)))), \tag{3.15}$$

odkąd uzasadnienie kroku γ, w którym została skonstruowana stała y odwołuje się do stwierdzenia P[x], sformułowanego w przesłance paczki $\mathcal{P}_4$ oraz analogicznie uzasadnienie kroku ζ, w którym została skonstruowana zmienna ustalona r odwołuje się do stwierdzenia R[z], sformułowanego w przesłance paczki $\mathcal{P}_4$. Modyfikacja skryptu dowodowego rys. 3.8 związana z wyodrębnieniem paczki w postaci lematu stwierdzającego formułę (3.15) została przedstawiona na wydruku 3.11.

Zauważmy, że istnienie zmiennych ustalonych wprowadzonych do rozumowania wewnątrz paczki, które są wykorzystywane poza jej obszarem, wiąże się nieodzownie z ponownym wprowadzeniem tych zmiennych do zmodyfikowanego rozumowania po wyodrębnieniu paczki (zob. kroki θ_1, θ_2 przedstawione na wydrukach 3.10, 3.11). Dodatkowo, stwierdzenie formuły Q[y] na podstawie przesłanki P[x] oraz stwierdzenia uzasadnionego w kroku θ_1 (wydruk 3.10) wymaga wprowadzenia do rozumowania zmiennej ustalonej r1 (krok θ_3) w celu umożliwienia bezpośredniego odwołania się do implikacji bazowej P[x] implies Q[y]. Ponadto liczba dodatkowych kroków wprowadzonych do rozumowania w celu umożliwienia jedynie „bezpośredniego" odwołania się do odpowiednich implikacji bazowych, może być proporcjonalna nawet do kwadratu liczby kwantyfikatorów egzystencjalnych. Rozwiązaniem tego problemu jest generalizacja metody, wykorzystanej w przypadku stwierdzenia (3.15).

```
α0   Lemma_P4: for x be set st P[x] holds
        ex y be set st Q[y] & for z be Subset of y st R[z] holds
           ex r be Relation of y,z st S[r]
     proof
α1 :    let x be set;
α2 :    assume A1: P[x];
γ' :    consider y be set such that
           A2:y=F(x) by A1;
α3 :    take y;
δ' :    thus Q[y] by A2;
α4 :    let z be Subset of y;
α5 :    assume A3: R[z];
ζ' :    consider r be Relation of y,z such that
           A4: S[r] by A3;
α6 :    take y;
α8 :    thus S[r] by A4;
     end;

α :  let x be set;
β :  A1: P[x];
θ1 : consider y be set such that
        B1: Q[y] & for z be Subset of y st R[z] holds
           ex r be Relation of y,z st S[r] by A1, Lemma_P4;
δ :  A3: Q[y] by B1;
ϵ :  consider z be Subset of y such that
        A4: R[z] by A3;
θ2 : consider r be Relation of y,z such that
        B2: S[r] by B1;
η :  A6: T[r] by B2;
```

Wydruk 3.11: Modyfikacja skryptu dowodowego rys. 3.8 związana z wyodrębnieniem paczki $\mathcal{P}_4$ w postaci lematu stwierdzającego formułę (3.15).

Ze względu jednak na uzależnienie konstrukcji od składni systemu Mizar w poniższych rozważaniach, przedstawiona zostanie jedynie metoda generowania stwierdzenia opisującego rozumowanie zawarte w paczce, uwzględniająca wykorzystywane zmienne oraz szkic metody modyfikacji pozostałej części rozumowania po wyodrębnieniu paczki rozważanego tutaj rodzaju.

Ustalmy zatem paczkę $\mathcal{P}$ w abstrakcyjnym grafie dowodu $\mathfrak{P}$. Dla uproszczenia będziemy zakładać, że w jednym kroku rozumowania jest wprowadzana co najwyżej jedna zmienna. Oznaczmy dwa zbiory wierzchołków w $\mathfrak{P}$:

$$\begin{aligned}\mathcal{A}_{\mathcal{P}}:=&\{v\in\mathcal{V}(\mathfrak{P}) : v\notin\mathcal{O}bsz(\mathcal{P})\wedge \underset{u\in\mathcal{O}bsz(\mathcal{P})}{\exists} v \xmapsto[\mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}}]{\mathcal{V}(\mathfrak{P})\setminus\mathcal{O}bsz(\mathcal{P})} u\},\\ \mathcal{E}_{\mathcal{P}}:=&\{v\in\mathcal{O}bsz(\mathcal{P}) : \underset{u\in\mathcal{V}(\mathfrak{P})\setminus\mathcal{O}bsz(\mathcal{P})}{\exists} v \xmapsto[\mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}}]{\mathcal{O}bsz(\mathcal{P})} u\}.\end{aligned} \tag{3.16}$$

Skonstruujmy następnie dwa ciągi $\{\mathcal{A}^i_{\mathcal{P}}\}_{i=1}^{\infty}$, $\{\mathcal{E}^i_{\mathcal{P}}\}_{i=1}^{\infty}$, będące odpowiednikami ciągów

danych zależnością (3.9):

$$\begin{aligned}
\mathcal{A}^0_{\mathcal{P}} &= \{a \in \mathcal{A}_{\mathcal{P}} : \underset{u \in \mathcal{A}_{\mathcal{P}} \cup \mathcal{E}_{\mathcal{P}}}{\forall} u \underset{\mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}}}{\rightarrow}{}^* a \implies u \in \mathcal{A}_{\mathcal{P}}\},\\
\mathcal{E}^i_{\mathcal{P}} &= \{e \in \mathcal{E}_{\mathcal{P}} : \underset{a \in \mathcal{A}^i_{\mathcal{P}}}{\exists} a \xmapsto[\mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}}]{\mathcal{O}bsz(\mathcal{P})} e\},\\
\mathcal{A}^{i+1}_{\mathcal{P}} &= \{a \in \mathcal{A}_{\mathcal{P}} : \underset{e \in \mathcal{E}^i_{\mathcal{P}}}{\exists} e \xmapsto[\mathcal{O}rd^{\mathcal{P}}_{\mathfrak{P}}]{\mathcal{V}(\mathfrak{P}) \setminus \mathcal{O}bsz(\mathcal{P})} a\},
\end{aligned} \tag{3.17}$$

gdzie i jest dowolną liczbą naturalną. Oczywiście elementy ciągu $\mathcal{A}^0_{\mathcal{P}}, \mathcal{E}^0_{\mathcal{P}}, \mathcal{A}^1_{\mathcal{P}}, \mathcal{E}^1_{\mathcal{P}}, \ldots$ stanowią partycję zbioru $\mathcal{A}_{\mathcal{P}} \cup \mathcal{E}_{\mathcal{P}}$ oraz istnieje liczba naturalna m, dla której m pierwszych elementów ciągu jest niepuste z wyjątkiem co najwyżej pierwszego, a kolejne wyrazy począwszy od $m+1$ są puste.

Każdemu zbiorowi wierzchołków $\mathcal{A}^i_{\mathcal{P}}$ oraz $\mathcal{E}^i_{\mathcal{P}}$ przyporządkujmy sortowanie topologiczne digrafu $\mathfrak{D}_{\widetilde{\mathfrak{P}}_{|\mathcal{A}^i_{\mathcal{P}}}}$ oraz $\mathfrak{D}_{\widetilde{\mathfrak{P}}_{|\mathcal{E}^i_{\mathcal{P}}}}$ oznaczane $\mathfrak{a}^i_{\mathcal{P}}$ oraz $\mathfrak{e}^i_{\mathcal{P}}$. Zdefiniujmy również zbiór przesłanek paczki, które zostały wykorzystane przy uzasadnieniu istnienia zmiennych ustalonych ze zbioru $\mathcal{E}^i_{\mathcal{P}}$ dane zależnością:

$$\mathcal{P}(\mathcal{E}^i_{\mathcal{P}}) := \{v \in \mathcal{P}re(\mathcal{P}) : \underset{e \in \mathcal{E}^i_{\mathcal{P}}}{\exists} v \underset{\mathfrak{D}_{\widetilde{\mathfrak{P}}}}{\rightarrow}{}^* e\}. \tag{3.18}$$

Rozumowanie zawarte wewnątrz paczki, jest wówczas opisywane przez ciąg fraz $\mu(1)$, $\mu(2), \ldots, \mu(m)$ danych zależnością:

$\mu(1) :=$ `for` *„uporządkowane zmienne dom*$(\mathfrak{a}^0_{\mathcal{P}})$*”* `holds` *„koniunkcja implikacji bazowych, które w sformułowaniu wykorzystują jedynie zmienne ustalone ze zbioru* $\mathcal{A}^0_{\mathcal{P}}$*”*

$\mu(2) :=$ `&` $\mathcal{P}(\mathcal{E}^0_{\mathcal{P}})$ `implies ex` *„uporządkowane zmienne dom*$(\mathfrak{e}^0_{\mathcal{P}})$*”* `st` *„koniunkcja implikacji bazowych, które nie zostały wymienione we frazie* $\mu(1)$*, w których są wykorzystywane co najwyżej zmienne ustalone ze zbioru* $\mathcal{A}^0_{\mathcal{P}} \cup \mathcal{E}^0_{\mathcal{P}}$*. Dodatkowo, w każdej takiej implikacji zostały pominięte przesłanki ze zbioru* $\mathcal{P}(\mathcal{E}^0_{\mathcal{P}})$*”*

$\vdots$

$\mu(2i+1) :=$ `for` *„uporządkowane zmienne dom*$(\mathfrak{a}^{i+1}_{\mathcal{P}})$*”* `holds` *„koniunkcja implikacji bazowych, które nie zostały wymienione we frazach* $\mu(1), \mu(2), \ldots, \mu(2i)$*, w których są wykorzystywane co najwyżej zmienne ustalone ze zbioru* $\mathcal{A}^{i+1}_{\mathcal{P}} \cup \bigcup\limits_{1 \leqslant j \leqslant i} (\mathcal{A}^j_{\mathcal{P}} \cup \mathcal{E}^j_{\mathcal{P}})$*. Dodatkowo, w każdej takiej implikacji zostały pominięte przesłanki ze zbioru* $\bigcup\limits_{1 \leqslant j \leqslant i} \mathcal{P}(\mathcal{E}^j_{\mathcal{P}})$*”*

$\mu(2i+2) :=$ `&` $\mathcal{P}(\mathcal{E}^i_{\mathcal{P}}) \setminus \bigcup\limits_{1 \leqslant j < i} \mathcal{P}(\mathcal{E}^j_{\mathcal{P}})$ `implies ex` *„uporządkowane zmienne dom*$(\mathfrak{e}^i_{\mathcal{P}})$*”* `st` *„koniunkcja implikacji bazowych, które nie zostały wymienione we frazach* $\mu(1), \mu(2), \ldots, \mu(2i+1)$*, w których są wykorzystywane co najwyżej zmienne ustalone ze zbioru* $\bigcup\limits_{1 \leqslant j \leqslant i} (\mathcal{A}^j_{\mathcal{P}} \cup \mathcal{E}^j_{\mathcal{P}})$*. Dodatkowo, w każdej takiej implikacji zostały pominięte przesłanki ze zbioru* $\bigcup\limits_{1 \leqslant j \leqslant i} \mathcal{P}(\mathcal{E}^j_{\mathcal{P}})$*”*

$\vdots$

Wykorzystanie powyższego stwierdzenia w kroku związanym z wierzchołkiem $v_{\mathcal{P}}$, który zastępuje wyodrębnioną paczkę $\mathcal{P}$, wymusza wprowadzenie dodatkowych wierzchołków v_j w modyfikowanym grafie dowodu, z których każdy odpowiada za wprowadzenie do rozumowania zmiennych ustalonych ze zbioru $\mathcal{E}_{\mathcal{P}}$ i przyjmuje poniższą postać:

```
consider „uporządkowane zmienne dom(e^j_P)” such that
    ⌈ab(v_j): „implikacje bazowe wymienione we frazie μ(2j) oraz ciąg fraz
        μ(2j+1), μ(2j+2), ..., μ(m)”
    by ⌈ab(P(E^j_P)), (⌈ab(v_{j-1}) jeśli j > 1, ⌈ab(v_P) jeśli j = 1);
```
(3.19)

gdzie $1 \leqslant j \leqslant \frac{m}{2}$. Dodatkowo, w celu zapewnienia poprawności zmodyfikowanego rozumowania, każde odwołanie się do konkluzji c paczki $\mathcal{P}$ musi zostać zastąpione odwołaniem do kroku v_j, dla którego implikacja bazowa związana z konkluzją c została wymieniona we frazie $\mu(2j)$ lub $\mu(2j+1)$. Również każdy łuk porządkujący, który łączył wierzchołek $e \in \mathcal{E}_{\mathcal{P}}$ z wierzchołkiem u nie należącym do obszaru $\mathcal{O}bsz(\mathcal{P})$, musi zostać zastąpiony łukiem porządkującym (v_j, u), gdzie j jest liczbą naturalną dla której $e \in \mathcal{E}^j_{\mathcal{P}}$.

3.2 Metody poprawy czytelności oparte o reorganizację kolejności kroków rozumowania

Śledząc opinie różnych użytkowników bazy danych systemu Mizar, natrafiamy na rozmaite poglądy dotyczące sposobu poprawy czytelności skryptów dowodowych wykorzystujących reorganizację kolejności kroków rozumowania. Możemy jednak wśród nich odnaleźć powtarzającą się część, pewien kanoniczny zestaw sposobów uszlachetniania nieformalnych dowodów matematycznych. Wśród nich szczególnie istotne wydają się te polegające na dążeniu do osiągnięcia wskazanych poniżej pięciu optymalnych sytuacji. Wyrazimy je, stosując wprowadzone w rozdziale 1 pojęcia dotyczące linearyzacji τ digrafu $\mathfrak{D}_{\mathfrak{P}}$.

1. Maksymalne $\tau_{\mathcal{R}ef_{\mathfrak{P}}}$–łańcuchy powinny mieć możliwie największą długość.
2. Dla każdego maksymalnego $\tau_{\mathcal{R}ef_{\mathfrak{P}}}$–łańcucha jego wierzchołki powinny wykorzystywać możliwie małą liczbę stwierdzeń, które są związane z wierzchołkami nienależącymi do tego łańcucha.
3. Liczba kroków rozumowania, z których każdy jest początkiem co najmniej jednego łuku referencyjnego o τ–rozpiętości większej od jeden powinna być minimalna.
4. Łączna τ–rozpiętość wszystkich łuków referencyjnych powinna być minimalna.
5. Zbiory kroków rozumowania, których gęstość w domknięciu zwrotno-przechodnim zbioru łuków $\mathcal{R}ef_{\mathfrak{P}}$ jest dostatecznie duża, w zlinearyzowanym rozumowaniu powinny stanowić τ–spoiste fragmenty.

Przedstawione metody poprawy czytelności wykorzystujące optymalizację wybranych wskaźników w zlinearyzowanym rozumowaniu, które będziemy nazywać *wskaźnikami czytelności*, wydają się w istocie naturalne i zgodne z intuicją, nie budziły również kontrowersji podczas prezentacji na konferencjach oraz innych forach dyskusji naukowych [69, 71]. Duży problem stanowi natomiast określenie hierarchii ważności tak sformułowanych metod optymalizacji. Niejednokrotnie napotykamy opinię, że metody

optymalizujące własności określone w punktach 1, 3, 5 są nieporównywalne, a dobór kolejności zależy od konkretnego rozumowania. Taka sytuacja sugeruje więc wykorzystanie optymalizacji wielokryterialnej lub użycie aplikacji, która umożliwi użytkownikowi wybór hierarchii. Innym rodzajem napotykanych trudności, są problemy związane z formalnym sformułowaniem przedstawionych metod oraz parametryzacją stwierdzeń „możliwie największą" oraz „dostatecznie duża".

3.2.1 Uzasadnienie wyboru metod poprawy czytelności

Użyteczność przedstawionych metod potwierdzają nie tylko użytkownicy bazy MML. Odnajdujemy ich uzasadnienie również w modelu procesów poznawczych występujących w czytaniu tekstów. Lokalność odwołań jest bowiem postrzegana jako ważny czynnik w procesie rozumienia czytanego tekstu. Był on analizowany już przez O. Behaghela [9]. Sformułowane przez niego pierwsze prawo stwierdza, że elementy, które są blisko siebie znaczeniowo powinny znajdować się blisko siebie (z ang. *Behaghel's First Law: 'elements that belong close together intellectually will also be placed close together'*). Liczne odwołania do tego prawa znajdujemy we współczesnej literaturze przedmiotu [31, 52].

Prawo to może również zostać zastosowane na gruncie rozważań związanych z czytelnością rozumowań formalnych. Zauważmy, że przesłanki wykorzystane do uzasadnienia konkretnego kroku, muszą zawsze poprzedzać ten krok w zlinearyzowanym rozumowaniu. Informacje te mogą jednak znajdować się daleko lub w bliskim sąsiedztwie tego kroku. Naturalnie nie wszystkie przesłanki mogą znajdować się w bezpośrednim sąsiedztwie tego kroku. Sytuacja taka występuje m.in. w przypadku, gdy jakaś przesłanka jest wielokrotnie wykorzystywana w rozumowaniu. Znaczna jednak część przesłanek jest wykorzystywana jedynie kilkakrotnie w rozumowaniu lub nawet tylko jeden raz. Naturalnie, przy odpowiednim doborze sposobu linearyzacji, istotna część spośród tych przesłanek może być zlokalizowana w bezpośrednim otoczeniu kroków, które się do nich odwołują lub nawet być sformułowana w kroku bezpośrednio poprzedzającym krok, w którym następuje odwołanie. Z prawa Behaghela możemy w szczególności więc wnioskować, że w poprawnie zbudowanych tekstach liczba kroków, które odwołują się do przesłanek sformułowanych w bezpośrednio poprzedzających krokach powinna być maksymalna. Stąd równoważnie, liczba łuków wyznaczających maksymalne $\tau_{\mathcal{R}ef_{\mathfrak{P}}}$–łańcuchy, powinna być maksymalna – na tym opiera się nasza *pierwsza metoda optymalizacyjna*.

Maksymalne $\tau_{\mathcal{R}ef_{\mathfrak{P}}}$–łańcuchy pełnią jeszcze jedną ważną rolę w procesie poszukiwania idei rozumowania. Zauważmy, że każdy taki $\tau_{\mathcal{R}ef_{\mathfrak{P}}}$–łańcuch jest liniowym fragmentem rozumowania, którego uwypuklenie wskazuje przyszłemu czytelnikowi, że fragment ten stanowi „lokalnie spójną" część rozumowania. Wybór podrozumowań, które miałyby odzwierciedlać lokalną spójność liniowych fragmentów, wymaga jednak szczegółowej analizy stwierdzeń formułowanych w krokach rozumowania, a ta musiałaby być wykonana przez człowieka. W celu uniknięcia tej konieczności możemy jednak przyjąć, że spójne fragmenty charakteryzują się dużą liczbą wewnętrznych odwołań. Precyzując, kroki należące do maksymalnego $\tau_{\mathcal{R}ef_{\mathfrak{P}}}$–łańcucha powinny odwoływać się do maksymalnej liczby faktów należących do tego łańcucha lub równoważnie, liczba odwołań między różnymi maksymalnymi $\tau_{\mathcal{R}ef_{\mathfrak{P}}}$–łańcuchami powinna być minimalna – na tym opiera się nasza *druga metoda optymalizacyjna*.

W celu przedstawienia genezy *trzeciej metody optymalizacyjnej* zauważmy, że kryterium numer 3 wiąże się ze zliczaniem kroków, które muszą posiadać etykietę. Przypomnijmy, że odwołanie się do wcześniej sformułowanego faktu w uzasadnieniu kroku dowodu jest realizowane w systemie Mizar przez przytoczenie w tym uzasadnieniu unikalnej etykiety przyporządkowanej do tego faktu. Jednym z cukrów syntaktycz-

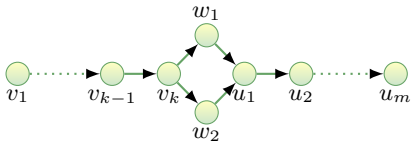
nych wdrożonym w systemie Mizar jest możliwość zastąpienia referencji do bezpośrednio poprzedzającego kroku, dodając na początku sformułowania kroku wyrażenie `then`. Naturalnie odpowiedni dobór kolejności kroków w zlinearyzowanym rozumowaniu, umożliwiający podkreślenie spójnych fragmentów rozumowania za pomocą konstrukcji `then` jest rozważany w dwóch pierwszych metodach optymalizacyjnych. Trzecia metoda optymalizacyjna skupia się wokół jeszcze jednego aspektu związanego z konstrukcją `then`. Opiera się ona bowiem na fakcie, że krok rozumowania, który jest przesłanką w co najwyżej jednym uzasadnieniu oraz jeśli jest wykorzystywany jako przesłanka, to tylko w kroku bezpośrednio występującym po nim, nie musi mieć przyporządkowanej etykiety. Minimalizacja liczby etykiet realizowana przez trzecią metodę optymalizacji umożliwia więc maksymalizacje liczby faktów, które będą pomijane w trakcie poszukiwań „daleko położonych" przesłanek w rozumowaniu, jako fakty, które jeśli są wykorzystywane, to tylko lokalnie.

Bardziej narzucającą się interpretacją trzeciej metody optymalizacyjnej jest zmniejszenie liczby przesłanek, których zapamiętanie przez czytelnika umożliwi przyśpieszenie odnajdywania stwierdzeń wskazywanych przez etykiety. Badania nad pojemnością pamięci krótkotrwałej człowieka, prowadzone przez Millera i Cowana [19, 61], wskazują jednak, że zapamiętanie dużej liczby przesłanek jest mało prawdopodobne. Pojemność ta jest bowiem oszacowana na 7 ± 2 „jednostki" (*kęsy* albo *elementy informacji*), a dostępność do zapamiętywanych jednostek spada szybko wraz z upływem czasu. Uwzględniając dodatkowo zjawisko inferencji informacji, wynikającej w tym przypadku z podobieństwa zapamiętywanych informacji uzyskujemy, że liczba informacji w swobodnym odtwarzaniu jednostek, jest oszacowana na jedynie 2-3 [91]. Niezależne badania przeprowadzone przez D. D. Wicknes, D. G. Born i C. K. Allen [95] wykazały bowiem, że poprawność odtworzenia czterech informacji z różnych kategorii semantycznych jest dwukrotnie większa niż w przypadku semantycznie jednolitych. Zależność ta została również potwierdzona w badaniach przeprowadzonych przez Loessa [53]. Badania te nie uwzględniały jednak wpływu treningu na możliwość zapamiętywania [14, 15, 18]. Wykorzystując model pamięci roboczej K. A. Ericsson i W. Kintsch [23, 24] wykazali bowiem, że długotrwały trening umożliwia zapamiętywanie nawet 80 cyfr, przy czym efekt ten był ograniczony wyłącznie do wybranego rodzaju materiału, jakim mogą być formuły. Obciążenie pamięci roboczej poprzez przechowywanie większego zbioru informacji prowadzi jednak do spowolnienia procesu wykonywania złożonych zadań poznawczych, jakim jest m.in. analizowanie sposobu uzasadnienia poszczególnych kroków rozumowania. Tym samym zapamiętywanie nadmiernej liczby przesłanek może przyśpieszyć proces wyszukiwania stwierdzeń wskazywanych przez referencje, aczkolwiek łączny czas analizowania całego rozumowania może ulec wydłużeniu. Badania nad pamięcią krótkotrwałą oraz pamięcią roboczą wskazują więc, że ograniczenie na liczbę zapamiętywanych informacji jest cechą indywidualną, ale stałą dla poszczególnych badanych. Stąd trzecia metoda optymalizacyjna minimalizująca liczbę stwierdzeń posiadających etykiety, analizowana w kontekście znanych modeli zapamiętywania, może być rozpatrywana w równoważnej postaci jako maksymalizacja procentu zapamiętywanych przesłanek spośród wszystkich przesłanek posiadających etykietę.

Wykorzystanie τ–rozpiętości łuków referencyjnych w sformułowaniu kryterium *czwartej metody optymalizacyjnej* związane jest z obciążeniem czytelnika wynikającym z konieczności nawracania do odległych części rozumowania celem przypomnienia sobie ich wyników. Ze względu na ograniczenie pojemności pamięci roboczej, czytelnik jest zmuszony do częstego poszukiwania większości stwierdzeń związanych z etykietami. Czas poszukiwania znaczenia tych etykiet w długich skryptach jest więc na ogół proporcjonalny do τ-rozpiętości odpowiednich łuków referencyjnych, gdyż poszukiwa-

nie etykiet przeważnie związane jest z liniowym przeskanowaniem dowodu wstecz od obecnego miejsca do miejsca wystąpienia etykiety. Dodatkowo czytelnik rozumowania zmuszony jest w trakcie poszukiwania do wielokrotnego przełączania się między zadaniami, co dodatkowo spowalnia ten proces [41]. Proces poszukiwania znaczenia dwóch kolejnych przesłanek jest bowiem często rozdzielany przez czytelników próbą „częściowego" uzasadnienia analizowanego kroku rozumowania w oparciu o odnalezioną przesłankę. Następstwem tego rozdzielania jest sumowanie τ-rozpiętości wszystkich łuków referencyjnych, a nie tylko największych τ-rozpiętości łuków o wspólnym końcu po wszystkich wierzchołkach.

W celu przedstawienia genezy kryterium *piątej metody optymalizacyjnej* zauważmy, że pierwsza metoda uwypukla w postaci $\tau_{\mathcal{R}ef_{\mathfrak{P}}}$–łańcuchów jedynie liniowe fragmenty rozumowania. Analizując rozumowania zgromadzone w bazie MML możemy jednak zauważyć, że zawarte w niej rozumowania są często „prawie" liniowe, ale nie dają się zoptymalizować przez pierwszą metodę. Rozważmy fragment rozumowania przedstawiony na rys. 3.12. Naturalnie fragment ten nie jest liniowy, ale dobrze odzwierciedla sens określenia „prawie liniowy" dla dostatecznie dużych k, m. Gęstość tego zbioru w domknięciu zwrotno-przechodnim jest bowiem bliska 1 (dokładnie wynosi ona $1 - \frac{2}{(k+m+2)\cdot(k+m+1)}$). Piąta metoda optymalizacyjna ma więc na celu wybór linearyzacji, dzięki której liczba zbiorów wierzchołków o dostatecznie dużej gęstości p, zapisanych w spójnych fragmentach jest maksymalna. Z punktu widzenia psychologi poznawczej kryterium to jest jedynie niewielkim rozluźnieniem wcześniejszych kryteriów i podobnie jak one opiera się na zasadzie lokalności odwołań.

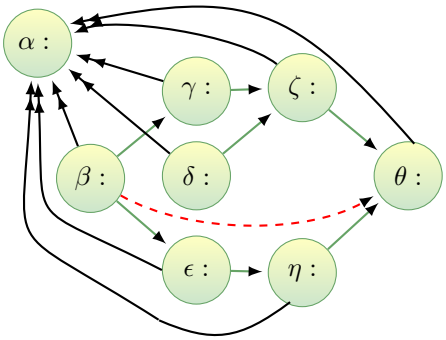


Rysunek 3.12: Przykład nieliniowego rozumowania, które zostanie uznane za dostatecznie gęste przez piątą metodę optymalizacyjną dla $p < 1$ oraz odpowiednio dużych k, m.

3.2.2 Hierarchia metod optymalizacji

Wśród hierarchii ważności przedstawionych kryteriów optymalizacji postaci liniowej rozumowania na szczególną uwagę zasługują dwie najbardziej popularne gradacje 1, 2, 3, 4, 5 oraz 3, 5, 1, 2, 4. Priorytetem pierwszej gradacji w procesie poprawy czytelności jest maksymalizowanie długości poszczególnych $\tau_{\mathcal{R}ef_{\mathfrak{P}}}$–łańcuchów, których wybór powinien odzwierciedlać „zwięzłe" liniowe fragmenty rozumowania. Przez zwięzłość rozumiemy liczbę odwołań w obszarze poszczególnych $\tau_{\mathcal{R}ef_{\mathfrak{P}}}$–łańcuchów. Priorytetem drugiej gradacji jest minimalizacja liczby wprowadzonych etykiet oraz spoisty zapis nie tylko $\tau_{\mathcal{R}ef_{\mathfrak{P}}}$–łańcuchów, ale również wszystkich fragmentów rozumowania odznaczających się dostatecznie dużą gęstością sieci informacji wewnątrz tych fragmentów.

Rozważmy dwie linearyzacje grafu dowodu uzasadnienia stwierdzenia $i \in \mathrm{Seg}(n) \implies i + m \in \mathrm{Seg}(n + m)$, gdzie $\mathrm{Seg}(k) := \{1, 2, \ldots, k\}$. Są one przedstawione na rys. 3.13. Na ich przykładzie przyjrzymy się dokładniej różnicom, jakie wynikają z reorganizacji rozumowania przy użyciu tych dwóch gradacji.




```
      theorem                                        theorem
α :    i in Seg n implies i+m in Seg(n+m)   α :    i in Seg n implies i+m in Seg(n+m)
      proof                                          proof
δ:     A1: i<=i+m by NAT_1:11;              β:     assume A1: i in Seg n;
β:     assume A2: i in Seg n;               γ:     then A2: 1<=i by FINSEQ_1:3;
ϵ:     then i<=n by FINSEQ_1:3 ;            ϵ:     i<=n by A1, FINSEQ_1:3;
η:     then A3: i+m<=n+m by XREAL_1:9;      η:     then A3: i+m<=n+m by XREAL_1:9;
γ:     1<=i by A2, FINSEQ_1:3;              δ:     i<=i+m by NAT_1:11;
ζ:     then 1<=i+m by A1, XXREAL_0:2;       ζ:     then 1<=i+m by A2, XXREAL_0:2;
θ:     hence thesis by A3, FINSEQ_1:3;      θ:     hence thesis by A3, FINSEQ_1:3;
      end;                                           end;
```

Rysunek 3.13: Dwie linearyzacje abstrakcyjnego grafu dowodu rozumowania uzasadniającego stwierdzenie $i \in \mathrm{Seg}(n) \Longrightarrow i + m \in \mathrm{Seg}(n + m)$ zapisane w języku Mizar.

Zauważmy, że w obu linearyzacjach zostały użyte dokładnie trzy etykiety, a więc obie linearyzacje w równym stopniu minimalizują wartość wskaźnika trzeciej metody. Również w przypadku wskaźnika pierwszej metody, obie linearyzacje w równym stopniu minimalizują liczbę maksymalnych łańcuchów. W obu linearyzacjach istnieją bowiem dokładnie cztery maksymalne łańcuchy α; δ; $\beta \rightarrow \epsilon \rightarrow \eta$; $\gamma \rightarrow \zeta \rightarrow \theta$ oraz α; $\beta \rightarrow \gamma$; $\epsilon \rightarrow \eta$; $\delta \rightarrow \zeta \rightarrow \theta$. Pierwsza metoda optymalizacji nie precyzuje jednak, czy większy wpływ na poprawę czytelności ma wybór linearyzacji, która generuje maksymalne łańcuchy o możliwie największej długości (dwa łańcuchy długości 3 oraz dwa długości 1 w pierwszej linearyzacji), kosztem generowania wielu krótkich, często jednoelementowych łańcuchów, czy też generowanie kilku „średniej" długości maksymalnych łańcuchów (łańcuch długości 3, dwa długości 2 oraz jeden długości 1 w drugiej linearyzacji).

Analizując wskazane maksymalne łańcuchy pod kątem drugiej metody optymalizacyjnej, możemy zauważyć, że dokładnie jeden maksymalny łańcuch w pierwszej analizowanej linearyzacji ($\gamma \rightarrow \zeta \rightarrow \theta$) ma odwołanie do przesłanki nie należącej do tego łańcucha. Natomiast w przypadku drugiej linearyzacji, możemy wskazać dwa takie łańcuchy ($\epsilon \rightarrow \eta$, $\delta \rightarrow \zeta \rightarrow \theta$) . Wykorzystują one odpowiednio 3 oraz 1, 2 zewnętrzne przesłanki. Sformułowanie drugiej metody optymalizacyjnej, podobnie jak sformułowanie pierwszej metody, nie precyzuje jednak, czy minimalizacja dotyczy łącznej liczby zewnętrznych odwołań, czy też górnego ograniczenia na liczbę zewnętrznych odwołań w poszczególnych maksymalnych łańcuchach.

Porównanie jakości realizacji czwartej metody optymalizacji dla tych dwóch linearyzacji sprowadza się do porównania wartości wskaźnika będącego sumą rozpiętości wszystkich łuków referencyjnych. Suma ta w przypadku pierwszej linearyzacji ma wartość 15, zaś w przypadku drugiej – 13. Tym samym, druga linearyzacja lepiej

minimalizuje wskaźnik, który jest optymalizowany przez czwartą metodę.

Porównując jakość realizacji piątej metody optymalizacyjnej w dwóch analizowanych linearyzacjach, możemy zauważyć, że spośród 14 zbiorów wierzchołków co najmniej dwu elementowych miary jeden, 6 z pośród nich tworzy spójne fragmenty w przypadku pierwszej linearyzacji, zaś 5 z nich w drugim przypadku (zbiory te odpowiadają łańcuchom rozumowania o długości 2). Tym samym pierwsza linearyzacja lepiej maksymalizuje liczbę spójnych fragmentów o mierze równej 1. Dodatkowo szczegółowa analiza miar wszystkich spójnych fragmentów w obu linearyzacjach wskazuje, że pierwsza linearyzacja zawiera większą liczbę spójnych fragmentów przy każdym dolnym ograniczeniu na wartość miary, co potwierdza stwierdzenie, że pierwsza linearyzacja lepiej realizuje piątą metodę optymalizacyjną.

3.2.3 Formalizacja kryteriów

Podejmując próbę formalnego sformułowania przedstawionych kryteriów optymalizacji struktury dowodu widzimy, że opisują one warunki narzucające własności na linearyzacje poszczególnych rozumowań pierwotnych w abstrakcyjnym grafie dowodu $\mathfrak{P}$, a nie całą linearyzację grafu. Wynika to z faktu, że przedstawione metody odzwierciedlają pewien zbiór własności nieformalnych dowodów matematycznych, w których rzadko prowadzi się rozumowanie na kilku poziomach zagnieżdżenia. Jeśli nawet ma to miejsce, często są one zamknięte w pojedynczych zdaniach bądź akapitach i tworzą zamknięte fragmenty rozumowania. Dodatkowo odwołania do tego rodzaju zagnieżdżonych podrozumowań nigdy nie wskazują na poszczególne jego kroki, ale na stwierdzenia uzasadnione na jego podstawie. Dopuszczalne jest natomiast odwołanie się do idei rozumowania uzasadniającego konkretny krok, ale tylko w przypadku mającym na celu zasugerowania czytelnikowi, że adaptując fragment tego uzasadnienia możliwe jest uzasadnienie innego kroku dowodu, które to uzasadnienie ze względu na analogię zostało pozostawione czytelnikowi. W celu uniknięcia wielopoziomowych rozumowań, wykorzystuje się również serię „oczywistych zdań" w rozumowaniu lub serię pomocniczych lematów, poprzedzających uzasadniane twierdzenie, wykorzystywanych na ogół wyłącznie w tym twierdzeniu.

Geneza przedstawionych wskaźników sugeruje również, że warunki narzucone na linearyzację ustalonego rozumowania pierwotnego $\mathfrak{A} \in \mathrm{RP}(\mathfrak{P})$ opisują w rzeczywistości własności domknięcia tego rozumowania (rozumowania pierwotnego z $\widetilde{\mathfrak{P}}$, które wierzchołkowo pokrywa się z $\mathfrak{A}$), oznaczane dalej $\widetilde{\mathfrak{A}}$.

Za takim ograniczeniem przemawia również fakt, że konkatenacja dwóch $\tau_{\mathcal{R}ef_{\mathfrak{A}}}$–łańcuchów $\mathfrak{u}_1 := u_1^1 \underset{\mathcal{R}ef_{\mathfrak{A}}}{\rightarrow} u_2^1 \underset{\mathcal{R}ef_{\mathfrak{A}}}{\rightarrow} \ldots \underset{\mathcal{R}ef_{\mathfrak{A}}}{\rightarrow} u_{n_1}^1$, $\mathfrak{u}_2 := u_1^2 \underset{\mathcal{R}ef_{\mathfrak{A}}}{\rightarrow} u_2^2 \underset{\mathcal{R}ef_{\mathfrak{A}}}{\rightarrow} \ldots \underset{\mathcal{R}ef_{\mathfrak{A}}}{\rightarrow} u_{n_2}^2$, zawartych na różnych poziomach zagnieżdżenia, intuicyjnie nie jest „łańcuchem rozumowania", jeśli nawet $u_1^1 \underset{\mathcal{R}ef_{\mathfrak{A}}}{\rightarrow} \ldots \underset{\mathcal{R}ef_{\mathfrak{A}}}{\rightarrow} u_{n_1}^1 \underset{\mathcal{R}ef_{\mathfrak{A}}}{\rightarrow} u_1^2 \underset{\mathcal{R}ef_{\mathfrak{A}}}{\rightarrow} \ldots \underset{\mathcal{R}ef_{\mathfrak{A}}}{\rightarrow} u_{n_2}^2$ jest $\tau_{\mathcal{R}ef_{\mathfrak{A}}}$–łańcuchem. Przypuśćmy dodatkowo, że $\mathfrak{u}_2$ stanowi fragment zagnieżdżonego uzasadnienia kroku v, który należy do rozumowania pierwotnego $\mathfrak{A}_1$ zawierającego wierzchołki z $\mathfrak{u}_1$. Wówczas ciąg $u_1^1 \underset{\mathcal{R}ef_{\mathfrak{A}}}{\rightarrow} \ldots \underset{\mathcal{R}ef_{\mathfrak{A}}}{\rightarrow} u_{n_1}^1 \underset{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}{\rightarrow} v$ jest bliższy intuicyjnie „łańcuchowi rozumowania", pomimo że nie jest on $\tau_{\mathcal{R}ef_{\mathfrak{A}}}$–łańcuchem. Jest on jednak $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchem.

W poniższych rozważaniach będziemy zawsze przyjmować, że $\widetilde{\mathfrak{A}}$ jest domknięciem rozumowania pierwotnego $\mathfrak{A}$, rozumianym jako acykliczny digraf, a $\tau \in TS(\widetilde{\mathfrak{A}})$. W $\mathcal{A}(\widetilde{\mathfrak{A}})$ będą wyróżnione dwie podrodziny łuków $\mathcal{R}ef_{\widetilde{\mathfrak{A}}}$, $\mathcal{O}rd_{\widetilde{\mathfrak{A}}}$, dla których $\mathcal{R}ef_{\widetilde{\mathfrak{A}}} \cup \mathcal{O}rd_{\widetilde{\mathfrak{A}}} = \mathcal{A}(\widetilde{\mathfrak{A}})$. Będziemy również wykorzystywać podzbiór zbioru wszystkich łuków referencyjnych w $\mathfrak{A}$, $\mathcal{R}ef_{\mathfrak{A}}$ oraz zbiór `then`–łuków, $\mathtt{then}_{\mathfrak{A}}$ zakładając o tych zbiorach

jedynie, że $\texttt{then}_{\mathfrak{A}} \subseteq \mathcal{R}ef_{\mathfrak{A}} \subseteq \mathcal{R}ef_{\widetilde{\mathfrak{A}}}$.

Wprowadźmy dodatkowe pojecie pomocnicze, umożliwiające bardziej precyzyjne sformułowanie metod optymalizacyjnych.

Definicja 3.11. Łańcuchem rozumowania $\mathfrak{A}$ *będziemy nazywać każdą skierowaną* $\mathcal{R}ef_{\widetilde{\mathfrak{A}}}$*–drogę* $\mathfrak{c}$*, dla której istnieje linearyzacja* $\sigma \in TS(\widetilde{\mathfrak{A}})$ *taka, że* $\mathfrak{c}$ *jest* $\sigma_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$*–łańcuchem.*

Jak łatwo można zauważyć, nie każda skierowana $\mathcal{R}ef_{\widetilde{\mathfrak{A}}}$–droga może być łańcuchem rozumowania $\mathfrak{A}$. W analizowanym przykładzie z rys. 3.13 rozumowanie pierwotne $\mathfrak{A}'$ zbudowane z wierzchołków β, γ, δ, ϵ, ζ, η, θ zawiera $\mathcal{R}ef_{\mathfrak{A}'}$–łańcuch $\beta \underset{\mathcal{R}ef_{\mathfrak{A}'}}{\rightarrow} \gamma \underset{\mathcal{R}ef_{\mathfrak{A}'}}{\rightarrow} \zeta \underset{\mathcal{R}ef_{\mathfrak{A}'}}{\rightarrow} \theta$, który nie jest łańcuchem rozumowania w $\mathfrak{A}'$, ponieważ w każdej linearyzacji $\tau \in TS(\mathfrak{A}')$ miedzy wierzchołkiem β i θ muszą się znajdować wierzchołki ϵ, η.

Stwierdzenie 3.12. *Niech* $\mathfrak{u} := u_1 \underset{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}{\rightarrow} u_2 \underset{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}{\rightarrow} \ldots \underset{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}{\rightarrow} u_n$ *będzie skierowaną* $\mathcal{R}ef_{\widetilde{\mathfrak{A}}}$*–drogą. Wówczas* $\mathfrak{u}$ *jest łańcuchem rozumowania wtedy i tylko wtedy, gdy istnieje dokładnie jedna skierowana* $\widetilde{\mathfrak{A}}$*–droga o początku* u_1 *i końcu* u_n*.*

3.2.4 Pierwsza metoda optymalizacyjna

Jak zostało zauważone w analizowanym przykładzie na rys. 3.13, wyrażenie „maksymalizacja" nie precyzuje dokładnie, w jaki sposób $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchy mają być maksymalizowane przez wybór linearyzacji $\tau \in TS(\widetilde{\mathfrak{A}})$. Najbardziej narzucającą się interpretacją tej metody jest stwierdzenie, że *najdłuższy łańcuch rozumowania z* $\mathfrak{A}$ *jest* $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$*–łańcuchem.* Pierwsza metoda optymalizacyjna poszukuje wówczas sortowania topologicznego spełniającego poniższą zależność:

$$\tau \in TS(\widetilde{\mathfrak{A}}) : \max_{P \in \tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}} |P| = \max_{\sigma \in TS(\widetilde{\mathfrak{A}})} \left(\max_{P \in \sigma_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}} |P| \right) \tag{3.20}$$

(przypomnijmy, że zgodnie z Def. 1.9 τ_A jest partycją digrafu D względem τ i zbioru $A \subseteq \mathcal{A}(D)$). Sformułowanie pierwszej metody optymalizacyjnej w postaci (3.20) nie określa jednak długości pozostałych maksymalnych $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchów w wybranej linearyzacji. Jednym z możliwych sposobów rozwiązania tego problemu jest sprowadzenie tej metody do poszukiwania największego elementu w kracie $\langle \{\sigma_{\overset{\searrow}{\mathcal{R}ef}_{\widetilde{\mathfrak{A}}}} : \sigma \in TS(\widetilde{\mathfrak{A}})\}, \preccurlyeq \rangle$,

$$\tau \in TS(\widetilde{\mathfrak{A}}) : \underset{\sigma \in TS(\widetilde{\mathfrak{A}})}{\forall} \sigma_{\overset{\searrow}{\mathcal{R}ef}_{\widetilde{\mathfrak{A}}}} \preccurlyeq \tau_{\overset{\searrow}{\mathcal{R}ef}_{\widetilde{\mathfrak{A}}}}. \tag{3.21}$$

gdzie $\tau_{\overset{\searrow}{\mathcal{R}ef}_{\widetilde{\mathfrak{A}}}} = \langle |P_1|, |P_2|, \ldots, |P_{|\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}|}| \rangle$, jeśli $|P_1| \geqslant |P_2| \geqslant \ldots \geqslant |P_{|\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}|}|$ oraz $\{P_1, P_2, \ldots, P_{|\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}|}\} = \tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$ dla $\tau \in TS(\mathfrak{A})$, a $\preccurlyeq$ oznacza porządek leksykograficzny. Takie ujęcie tej metody dopuszcza jednak, by linearyzacja τ zawierała maksymalne jednoelementowe $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchy. Odwrócenie kolejności wyrazów w ciągach $\sigma_{\overset{\searrow}{\mathcal{R}ef}_{\widetilde{\mathfrak{A}}}}$ oraz poszukiwanie elementu największego względem porządku $\preccurlyeq$ minimalizuje liczbę krótkich maksymalnych $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchów, ale uzyskane rozwiązanie nie musi spełniać warunku (3.20).

Przedstawione dotąd interpretacje pierwszej metody optymalizacyjnej posiadają charakter lokalny. Jest to szczególnie widoczne w rozumowaniach zawierających dokładnie jeden maksymalny łańcuch rozumowania o maksymalnej długości. Globalny charakter sformułowania pierwszej metody możemy uzyskać, wykorzystując związek między długością poszczególnych maksymalnych $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchów a mocą partycji $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$. Minimalizowanie mocy partycji $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$ wpływa bowiem na wzrost długości składających się na nią $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchów, umożliwiając tym samym sformułowanie pierwszej metody w postaci:

$$\tau \in TS(\widetilde{\mathfrak{A}}) : |\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}| = \min_{\sigma \in TS(\widetilde{\mathfrak{A}})} |\sigma_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}|, \tag{3.22}$$

gdzie wskaźnikiem optymalizowanym przez tę metodę jest liczność partycji. Za taką interpretacją przemawia również fakt, że $|\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}|$ ma ścisły związek z liczbą $\mathbf{1}_\tau^{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łuków, tzn. $\mathcal{R}ef_{\widetilde{\mathfrak{A}}}$–łuków referencyjnych, które łączą kroki występujące bezpośrednio po sobie w zlinearyzowanym rozumowaniu τ (zob. definicję na str. 13).

Stwierdzenie 3.13. *Niech D będzie acyklicznym digrafem, $\tau \in TS(D)$, $A \subseteq \mathcal{A}(D)$. Wówczas $|\tau_A| + |\mathbf{1}_\tau^A| = |\mathcal{V}(D)|$.*

Dowód. Pokażemy w istocie, że $|\mathbf{1}_\tau^A| = |\mathcal{V}(D)| - |\tau_A|$. Partycja τ_A dzieli $\mathcal{V}(D)$ na rozłączne części. W ramach $P \in \tau_A$ wierzchołki sa połączone $\mathbf{1}_\tau^A$–łukami i jest ich $|P| - 1$. Zauważmy, że nie ma $\mathbf{1}_\tau^A$–łuków poza elementami partycji τ_A, bo inaczej obecne w niej τ_A–łańcuchy nie byłyby maksymalne. □

Warunek (3.22) jest więc równoważny z maksymalizacją liczby $\mathbf{1}_\tau^{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łuków w linearyzacji τ rozumowania $\widetilde{\mathfrak{A}}$. Tym samym (3.22), dobrze oddaje intuicje związane z pierwszą metodą i umożliwia jasne sformułowanie problemu grafowego charakteryzującego tę metodę:

$\mathcal{K}.1$: Instancja: DAG D, zbiór $A \subseteq \mathcal{A}(D)$, liczba naturalna $0 \leqslant M \leqslant |\mathcal{V}(D)|$.
Pytanie: Czy istnieje sortowanie topologiczne $\tau \in TS(D)$ spełniające zależność: $|\mathbf{1}_\tau^A| \geqslant M$?

Wykorzystując Tw. 1.15, 1.16 oraz Stw. 3.13, możemy sformułować postawiony problem $\mathcal{K}.1$ w postaci równoważnej, wykorzystując przy tym acykliczną partycję digrafu na drogi Hamiltona:

$\mathcal{K}.1'$: Instancja: DAG D, zbiór $A \subseteq \mathcal{A}(D)$, liczba naturalna $0 \leqslant M \leqslant |\mathcal{V}(D)|$.
Pytanie: Czy istnieje acykliczna $\mathcal{H}_A^{(*)}$–partycja π digrafu D spełniająca zależność: $|\pi| \leqslant M$?

3.2.5 Druga metoda optymalizacyjna

W drugiej metodzie optymalizacji poszukiwana jest linearyzacja τ, która minimalizuje wskaźnik K_τ będący ilością informacji przekazywanych między różnymi maksymalnymi $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchami. Wskaźnik K_τ możemy wówczas zdefiniować na co najmniej dwa niezależne sposoby:

(*i*) jako górne ograniczenie na liczbę przekazywanych informacji między dowolnymi dwoma różnymi maksymalnymi $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchami:

$$K_\tau := \max_{\substack{P_i, P_j \in \tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}} \\ P_i \neq P_j}} |P_i \underset{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}{\curvearrowright} P_j|, \tag{3.23}$$

(*ii*) lub jako górne ograniczenie na liczbę wszystkich referencji między różnymi maksymalnymi $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchami:

$$K_\tau := \sum_{\substack{P_i, P_j \in \tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}} \\ P_1 \neq P_j}} |P_i \underset{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}{\curvearrowright} P_j|. \tag{3.24}$$

Minimalizowanie wskaźnika K_τ sformułowanego w postaci (3.23) charakteryzuje pewnego rodzaju zbalansowanie, wskazujący górne ograniczenie na liczbę przekazywanych informacji między poszczególnymi maksymalnymi $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchami. Tak sformułowane kryterium nie narzuca jednak żadnych ograniczeń na liczbę maksymalnych $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchów. Ponadto, jeśli istnieje linearyzacja $\tau' \in TS(\widetilde{\mathfrak{A}})$, w której każdy maksymalny ${\tau'}_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuch zawiera dokładnie jeden wierzchołek, to $K_{\tau'} = 1$, stąd jeśli $\widetilde{\mathfrak{A}}$ nie posiada $\mathcal{R}ef_{\widetilde{\mathfrak{A}}}$–drogi Hamiltona, to linearyzacja τ' minimalizuje tak zdefiniowany wskaźnik K_τ.

Zdefiniowanie wskaźnika K_τ w postaci (3.24) nadaje drugiej metodzie optymalizacyjnej bardziej globalny charakter. Dodatkowo takie sformułowanie tej metody upodabnia ją do pierwszej metody optymalizacyjnej. Zauważmy bowiem, że jeśli abstrakcyjny graf dowodu nie zawiera $\mathcal{R}ef_{\widetilde{\mathfrak{A}}}$–skrótów, to dla każdego $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcucha P, zachodzi tożsamość $|\mathcal{V}(P) \underset{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}{\curvearrowright} \mathcal{V}(P)| = |\mathcal{V}(P)| - 1$ (zob. Def. 1.3). Stąd w szczególności uzyskujemy, że $K_\tau = |A| - |\mathcal{V}(\mathfrak{A})| + |\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}|$ (Lem. 3.14).

Lemat 3.14. *Niech D będzie acyklicznym digrafem, $A \subseteq \mathcal{A}(D)$ oraz $\tau \in TS(D)$. Wówczas jeśli D nie zawiera A–skrótów, to prawdziwa jest tożsamość*

$$|A| - \sum_{\substack{P_1, P_2 \in \tau_A \\ P_1 \neq P_2}} |P_1 \underset{A}{\curvearrowright} P_2| = |\mathcal{V}(D)| - |\tau_A|. \tag{3.25}$$

Dowód. Niech D, A oraz τ spełniają założenia lematu. Na mocy Tw. 1.15 τ_A jest acykliczną $\mathcal{H}_A^{(|\tau_A|)}$–partycją digrafu D. Wówczas zbiór łuków $\mathcal{A}(\mathfrak{h}^{\tau_A}(P)) \subseteq A$ oraz $|\mathcal{A}(\mathfrak{h}^{\tau_A}(P))| = |\mathcal{V}(\mathfrak{h}^{\tau_A}(P))| - 1$ dla wszystkich $P \in \tau_A$. Stąd ostatecznie

$$\sum_{\substack{P_1, P_2 \in \tau_A \\ P_1 \neq P_2}} |P_1 \underset{A}{\curvearrowright} P_2| = |A| - \sum_{P \in \tau_A} |P \underset{A}{\curvearrowright} P| =$$

$$|A| - \sum_{P \in \tau_A} (|\mathcal{V}(\mathfrak{h}^{\tau_A}(P))| - 1) = |A| - |\mathcal{V}(D)| + |\tau_A|. \tag{3.26}$$

□

Istnienie $\mathcal{R}ef_{\widetilde{\mathfrak{A}}}$–skrótów rozróżnia więc dwie pierwsze metody optymalizacyjne. Jeśli bowiem istnieją $\mathcal{R}ef_{\widetilde{\mathfrak{A}}}$–skróty, to liczba referencji odwołujących się do kroków w obrębie jednego $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcucha P może przewyższać liczbę $|\mathcal{V}(P)| - 1$. Zauważmy również, że jeśli $|\mathcal{V}(P) \underset{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}{\curvearrowright} \mathcal{V}(P)| \geqslant |\mathcal{V}(P)|$ to co najmniej jeden $\mathcal{R}ef_{\widetilde{\mathfrak{A}}}$–łuk łączący dwa wierzchołki z $\mathcal{V}(P)$ ma τ–rozpiętości $\geqslant 2$. Tym samym kroki w maksymalnych $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchach wykorzystują bezpośrednio poprzedzający krok w linearyzacji τ, ale również inne kroki należące do tego łańcucha. Dodatkowo łączna liczba odwołań o τ–rozpiętości $\geqslant 2$ w obrębie poszczególnych maksymalnych $\tau_{\mathcal{R}ef_{\widetilde{\mathfrak{A}}}}$–łańcuchów jest maksymalizowana przez drugą metodę optymalizacyjną, jeśli ograniczymy przestrzeń

poszukiwań tej metody do sortowań topologicznych $\tau' \in TS(\widetilde{\mathfrak{A}})$ o ustalonej liczebności partycji $\tau'_{\mathcal{R}ef^{\sim}_{\mathfrak{A}}}$ (Lem. 3.15). Linearyzacja wyznaczona przez drugą metodę optymalizacyjną zwiększa więc „spójność" lokalnych rozumowań zawartych w maksymalnych $\tau_{\mathcal{R}ef^{\sim}_{\mathfrak{A}}}$–łańcuchach, jeśli za miarę spójności przyjąć liczbę wewnętrznych odwołań.

Lemat 3.15. *Niech D będzie acyklicznym digrafem, $A \subseteq \mathcal{A}(D)$, $\tau \in TS(D)$. Wówczas*

$$|\bigcup_{P\in\tau_A}\{vu \in A : d_\tau(v,u) \geqslant 2 \wedge v,u \in P\}| = |A| - \mathcal{V}(D) + |\tau_A| - \sum_{\substack{P_1,P_2\in\tau_A\\ P_1\neq P_2}} |P_1 \underset{A}{\curvearrowright} P_2|. \tag{3.27}$$

Dowód. Wystarczy zauważyć, że $\bigcup_{P\in\tau_A}\big((P \underset{A}{\curvearrowright} P) \setminus \{vu \in A : d_\tau(v,u) \geqslant 2 \wedge v,u \in P\}\big) = \mathbf{1}^A_\tau$, co w konkluzji ze Stw. 3.13, dowodzi uzasadnianą równość. □

Zdefiniowanie wskaźnika K_τ w postaci (3.24) oddaje więc w większym stopniu intuicje związane z drugą metodą optymalizacyjną. Ostateczna postać problemu grafowego charakteryzującego tę metodę przyjmuje zatem następującą postać:

$\mathcal{K}.2$: INSTANCJA: DAG D, zbiór $A \subseteq \mathcal{A}(D)$, liczba naturalna $0 \leqslant M \leqslant |A|$.
PYTANIE: Czy istnieje sortowanie topologiczne $\tau \in TS(D)$ spełniające zależność: $\sum_{\substack{P_1,P_2\in\tau_A\\ P_1\neq P_2}} |P_1 \underset{A}{\curvearrowright} P_2| \leqslant M$?

Prowadząc rozumowanie analogiczne do rozważań zawartych w uzasadnieniu Tw. 1.16, formułujemy postawiony problem $\mathcal{K}.2$ w równoważnej postaci:

$\mathcal{K}.2'$: INSTANCJA: DAG D, zbiór $A \subseteq \mathcal{A}(D)$, liczba naturalna $0 \leqslant M \leqslant |A|$.
PYTANIE: Czy istniej acykliczna $\mathcal{H}^{(*)}_A$–partycja π digrafu D spełniająca zależność: $\sum_{\substack{P_1,P_2\in\pi\\ P_1\neq P_2}} |P_1 \underset{A}{\curvearrowright} P_2| \leqslant M$?

3.2.6 Trzecia metoda optymalizacyjna

Z charakteryzacji trzeciej metody optymalizacji przeprowadzonej w ramach analizy przykładu rys. 3.13 uzyskujemy, że metoda ta minimalizuje liczbę etykiet L_τ, które należy wprowadzić do zlinearyzowanego rozumowania, aby możliwe było odwołanie się do wcześniej uzasadnionych faktów, które nie mogą zostać przekazane do odpowiednich uzasadnień przy pomocy konstrukcji `then`. Przypomnijmy, że ze wstępnej analizy składni systemu Mizar (tab. D.1) wynika, że za pomocą konstrukcji `then` może zostać przekazana tylko taka przesłanka, która jest sformułowana w bezpośrednio poprzedzającym kroku oraz jeśli łuk referencyjny opisujący odwołanie do tej przesłanki jest `then`–łukiem. Tym samym z wierzchołkiem $v \in \mathcal{V}(\mathfrak{A})$ musi być związana etykieta wtedy i tylko wtedy, gdy:

1. $vu \in \mathcal{R}ef^{\sim}_{\mathfrak{A}} \wedge d_\tau(v,u) > 1$ lub
2. $vu \in \mathcal{R}ef^{\sim}_{\mathfrak{A}} \setminus \mathtt{then}_{\mathfrak{A}}$.

Stąd wskaźnik L_τ dany jest zależnością:

$$L_\tau = |\{v \in \mathcal{V}(\mathfrak{A}) : \underset{u\in\mathcal{V}(\mathfrak{A})}{\exists} vu \in \mathcal{R}ef^{\sim}_{\mathfrak{A}} \wedge (d_\tau(v,u) > 1 \vee vu \notin \mathtt{then}_{\mathfrak{A}})\}|. \tag{3.28}$$

Trzecia metoda optymalizacyjna rozwiązuje więc następujący problem grafowy:

$\mathcal{K}.3$: INSTANCJA: DAG D, zbiory $A_1 \subseteq A_2 \subseteq \mathcal{A}(D)$, liczba naturalna $0 \leqslant M \leqslant |\mathcal{V}(D)|$.
PYTANIE: Czy istnieje sortowanie topologiczne $\tau \in TS(D)$ spełniające zależność:

$$|\{v \in \mathcal{V}(D) : \underset{u \in \mathcal{V}(D)}{\exists} vu \in A_2 \wedge (\, d_\tau(v,u) > 1 \vee vu \notin A_1\,)\}| \leqslant M? \tag{3.29}$$

Analiza zależności występujących między zbiorami $\texttt{then}_{\mathfrak{A}}$, $\mathcal{R}ef_{\mathfrak{A}}$, $\mathcal{O}rd_{\mathfrak{P}}$ sformułowana w postaci (2.8) na str. 30, wskazuje na dodatkową zależność występującą między zbiorami A_1, A_2, $\mathcal{A}(D)$, które są zdefiniowane w sformułowaniu problemu $\mathcal{K}.3$. Składnia systemu Mizar wymusza bowiem przyporządkowanie etykiety każdemu krokowi rozumowania ze zbioru $\{v \in \mathcal{V}(D) : \underset{u,w \in \mathcal{V}(D)}{\exists} vu \in \mathcal{R}ef_{\mathfrak{P}} \wedge vw \in \mathcal{O}rd_{\mathfrak{P}}\}$. Stąd sformułowanie problemu $\mathcal{K}.3$ uwzględniające składnię systemu Mizar przyjmuje następującą postać:

$\mathcal{K}.3_{MIZ}$: INSTANCJA: DAG D, zbiory $A_1 \subseteq A_2 \subseteq \mathcal{A}(D)$, liczba naturalna $0 \leqslant M \leqslant |\mathcal{V}(D)|$.
PYTANIE: Czy istnieje sortowanie topologiczne $\tau \in TS(D)$ spełniające zależność:

$$|\{v \in \mathcal{V}(D) : \underset{u \in \mathcal{V}(D)}{\exists} vu \in A_2 \wedge (\, d_\tau(v,u) > 1 \vee vu \notin A_1 \vee \underset{w \in \mathcal{V}(D)}{\exists} vw \in \mathcal{A}(D) \setminus A_2\,)\}| \leqslant M? \tag{3.30}$$

3.2.7 Czwarta metoda optymalizacyjna

Sformułowanie czwartej metody optymalizacyjnej jednoznacznie określa minimalizowany wskaźnik B_τ uzależniony od linearyzacji $\tau \in TS(\widetilde{\mathfrak{A}})$, dany zależnością:

$$B_\tau := \sum_{vu \in \mathcal{R}ef_{\widetilde{\mathfrak{A}}}} d_\tau(v,u). \tag{3.31}$$

Wartość wskaźnika B_τ danego zależnością (3.31) jest uzależniona od łuków referencyjnych. Zauważmy jednak, że czytelnik może poszukiwać w rozumowaniu nie tylko informacji związanej z referencjami, ale na przykład związanej z wykorzystaniem identyfikatorów zmiennych ustalonych lub chcieć korzystać intensywnie z dowolnej innej klasy krawędzi. W związku z tym naturalne jest uogólnienie formuły 3.31 w następujący sposób. Ustalmy pomocniczą funkcję wagi $w_{\mathfrak{A}} : \mathcal{A}(\widetilde{\mathfrak{A}}) \to \mathbb{N} \cup \{0\}$. Wówczas wskaźnik B_τ dany jest zależnością:

$$B_\tau = \sum_{vu \in \mathcal{A}(\widetilde{\mathfrak{A}})} w_{\mathfrak{A}}(vu) \cdot d_\tau(v,u). \tag{3.32}$$

Stąd ostateczna postać problemu grafowego rozwiązywanego przez tę metodę przyjmuje następującą postać:

$\mathcal{K}.4$: INSTANCJA: DAG D, funkcja $w : \mathcal{A}(D) \to \mathbb{N} \cup \{0\}$, liczba naturalna $0 \leqslant M \leqslant \binom{|\mathcal{V}(D)|+1}{3} \cdot \max_{vu \in \mathcal{A}(D)} w(vu)$.
PYTANIE: Czy istnieje sortowanie topologiczne $\tau \in TS(D)$ spełniające zależność:

$$\sum_{vu \in \mathcal{A}(D)} w(vu) \cdot d_\tau(v,u) \leqslant M? \tag{3.33}$$

Dzięki takiemu postawieniu problemu czytelnik może indywidualnie określić, które krawędzie są dla niego istotne i przez nadanie im dużej wagi sprawić, że sortowanie topologiczne umieści ich końce niedaleko od siebie w wynikowym skrypcie.

Na uwagę zasługuje również wyrażenie $\binom{|\mathcal{V}(D)|+1}{3}$. Zauważmy, że sortując topologicznie za pomocą τ' digraf o n wierzchołkach, możemy uzyskać co najwyżej $n-1$ łuków o τ'–rozpiętości 1, $n-2$ łuków o τ'–rozpiętości 2 itd. Stąd ograniczenie na sumę τ'–rozpiętości ma wartość $\sum_{i=1}^{n-1} i \cdot (n-i) = \binom{n+1}{3}$.

3.2.8 Piąta metoda optymalizacyjna

Sformułowanie piątej metody optymalizacyjnej, podobnie jak czwartej, jednoznacznie określa maksymalizowany wskaźnik $S_{\tau,p}$, uzależniony od linearyzacji $\tau \in TS(\widetilde{\mathfrak{A}})$ oraz liczby dodatniej p nie większej od 1, dany zależnością:

$$S_{\tau,p} := |\{V \subseteq \mathcal{V}(\mathfrak{A}) : |V| \geqslant 2 \wedge \rho_{\mathcal{R}ef^{\sim}_{\mathfrak{A}}}(V) \geqslant p \wedge V \text{ jest } \tau\text{–spoisty}\}|, \tag{3.34}$$

gdzie $\rho_{\mathcal{R}ef^{\sim}_{\mathfrak{A}}}(V)$ oznacza gęstość zbioru $V \subseteq \mathcal{V}(D)$ w domknięciu zwrotno-przechodnim zbioru łuków $\mathcal{R}ef^{\sim}_{\mathfrak{A}}$ (zob. Def. 1.4 na str. 12), natomiast zbiór V jest τ–spoisty, jeśli istnieje liczba naturalna i, dla której $i \leqslant \tau(v) \leqslant i + |V| - 1$ (zob. Def. 1.10 na str. 13). Stąd ostateczna postać problemu grafowego charakteryzującego tę metodę optymalizacyjną ma postać:

$\mathcal{K}.5$: Instancja: DAG D, zbiór $A \subseteq \mathcal{A}(D)$, liczba rzeczywista $0 < p \leqslant 1$, liczba naturalna $0 \leqslant M \leqslant 2^{|\mathcal{V}(D)|}$.
Pytanie: Czy istnieje sortowanie topologiczne $\tau \in TS(D)$ spełniające zależność:

$$|\{V \subseteq \mathcal{V}(D) : |V| \geqslant 2 \wedge \rho_A(V) \geqslant p \wedge V \text{ jest } \tau\text{–spoisty}\}| \geqslant M? \tag{3.35}$$

3.3 Metody poprawy czytelności wykorzystujące modyfikację formuł w krokach rozumowania

Przeprowadzone dotąd rozważania dotyczące sposobu poprawy czytelności rozumowań skupiały się jedynie wokół modyfikacji struktury dowodu lub sposobu jego linearyzacji przy zachowaniu formuł stwierdzonych w poszczególnych krokach rozumowania. Zakładały one bowiem m.in., że modyfikowane rozumowanie nie zawiera niepotrzebnych lub możliwych do usunięcia fragmentów, a wszystkie przesłanki przytoczone w poszczególnych krokach są konieczne do uzasadnienia sformułowanych w nich faktów. Rozważania zawarte w tym podrozdziale będą dotyczyły metod „wstępnej" poprawy budowy rozumowań, mających na celu zagwarantowanie powyższych założeń. Będą one skupiały się głównie na dwóch działaniach: upraszczaniu struktury grafu dowodu oraz modyfikowaniu budowy formuł stwierdzanych w krokach rozumowania.

3.3.1 Narzędzia dystrybuowane z systemem Mizar

Od początków rozwoju systemu Mizar podejmowane były próby tworzenia nowych narzędzi, których głównym zadaniem było poprawienie jakości artykułów zgromadzonych w bazie MML. Niektóre z nich nie zapewniały założonych skutków lub okazywały

się z czasem bezużyteczne na skutek modyfikacji systemu. Istnieje również znaczna grupa programów, które są intensywnie wykorzystywane i uaktualniane do nowych wersji systemu Mizar, a nawet dystrybuowane jako dodatkowe narzędzia systemowe. Niektóre z nich zostały wręcz wbudowane na stałe do systemu jako dodatkowe moduły [48, 49, 60, 64]. Możemy tu wyróżnić pięć programów [57, 58, 77, 78], działających na poprawnych artykułach, które realizują wybrane aspekty wstępnej poprawy budowy struktur rozumowań:

RelPrem – wskazuje przesłanki, których usunięcie z uzasadnień poszczególnych kroków, nie wpływa na ich akceptowalność przez weryfikator systemu Mizar,

RelInfer – wskazuje te przesłanki w uzasadnieniu, które mogą zostać zastąpione przez listę przesłanek użytych do ich uzasadnienia,

TrivDemo – wskazuje stwierdzenia, których uzasadnienie w postaci zagnieżdżonego podrozumowania może zostać zastąpione listą przesłanek, które jednocześnie są wykorzystywane w tym podrozumowaniu oraz zostały stwierdzone poza obszarem tego podrozumowania,

Inacc – wskazuje niewykorzystywane fragmenty rozumowania, których usunięcie nie wpływa na poprawność zmodyfikowanego w ten sposób rozumowania,

ChkLab– wskazuje etykiety, które nie zostały wykorzystane w żadnym uzasadnieniu znajdującym się w skrypcie dowodowym.

Programy te mają również istotne znaczenie w procesie przyjmowania nowych artykułów do bazy MML. Jednym z podstawowych wymogów edytorskich stawianych przed artykułami nadsyłanymi do bazy MML jest bowiem eliminacja wszystkich usterek wskazywanych przez te programy.

Modyfikacja fragmentów rozumowania wskazywanych przez wyżej wymienione programy wymaga szczegółowej analizy wskazań tylko w przypadku dwóch pierwszych z nich. Wskazanie przez program RelPrem dwóch przesłanek w uzasadnieniu jednego kroku nie oznacza bowiem w ogólnym przypadku, że możliwe jest usunięcie ich obu (np. w sytuacji, gdy przesłanki te są równoważne). Analogiczny problem dotyczy programu RelInfer. Z faktu, że weryfikator systemu Mizar może uzasadnić krok rozumowania pomimo zastąpienia jednej referencji przez odpowiednią listę przesłanek, nie wynika, że po zastąpieniu wskazanych referencji w pierwotnym uzasadnieniu odpowiednimi listami, zmodyfikowane uzasadnienie będzie w dalszym ciągu dostateczne do uzasadnienia związanego z tym krokiem stwierdzenia. Zmodyfikowane uzasadnienie może bowiem mieć liczbę referencji przekraczającą górne ograniczenie narzucone na liczbę przesłanek w pojedynczym uzasadnieniu (w dystrybuowanej wersji systemu Mizar ograniczenie to ma wartość 25). Dodatkowo odpowiedni dobór zbioru przesłanek wybranych wśród tych, które zostały wskazane przez RelInfer, może utrudnić zrozumienie idei tak zmodyfikowanych uzasadnień. Sytuacja ta jest następstwem m.in. wzmocnienia systemu Mizar o oczywistość w sensie M. Davisa [20], która jest intensywnie wykorzystywana przez RelInfer. Naturalnie, oczywistość ta umożliwia tworzenie krótszych rozumowań, ale równocześnie zmusza niejednokrotnie czytelnika do analizowania skomplikowanych uzasadnień. W szczególności, wykorzystanie oczywistości w sensie M. Davisa w weryfikatorze systemu Mizar umożliwia stwierdzenie poprawności reguły pijącego z wydruku 2.11 bez podania jego uzasadnienia.

Regularne przebudowywanie bazy MML oraz wzmacnianie systemu Mizar przyczyniają się jednak do powstawania nowych wskazań, generowanych przez wyżej wymienione programy w „oczyszczonych" przez autora artykułach. Szczegółowa analiza każdego takiego przypadku ze względu na liczbę artykułów w bazie wiązałaby się

jednak ze zbyt dużym nakładem pracy, co w konsekwencji uniemożliwiałoby przeprowadzenie modernizacji bazy. W celu uniknięcia tego problemu zostały stworzone dwa niżej przedstawione programy, rozwiązujące problem eliminacji wskazań generowanych przez RelPrem oraz RelInfer.

ChkRPrem – program ten jest modyfikacją programu RelPrem, umożliwiającą wskazanie minimalnego, choć nie zawsze najmniejszego, podzbioru przesłanek, który to zbiór jest wystarczający do zaakceptowania uzasadnienia przez weryfikator systemu Mizar. Jest ona jednym spośród pierwszych narzędzi stworzonych dla potrzeb poprawy bazy MML i jest dystrybuowana z systemem Mizar do dnia dzisiejszego [55]. Analogicznie jak w przypadku RelPrem algorytm wykorzystywany w tym programie przebiega listę przesłanek w każdym kroku rozumowania (począwszy od ostatniej przesłanki) w poszukiwaniu tych przesłanek, których odrzucenie nie wpływa na dostateczność uzasadnienia tego kroku. Jedyna różnica między tymi dwoma algorytmami występuje w przypadku odnalezienia niepotrzebnej przesłanki. Jest ona bowiem usuwana z listy, a do dalszych poszukiwań nadmiarowych przesłanek jest wykorzystywana jedynie zmodyfikowana lista. Oczywiście odnajdywane w ten sposób zbiory przesłanek nie są w ogólnym przypadku minimalne w sensie liczebności.

RenInfer – program ten zastępuje wybrane referencje wskazane przez RelInfer odpowiednimi listami referencji, modyfikując w uzasadnieniu każdego kroku rozumowania co najwyżej jedną referencję [70]. W intencji program ten ma służyć temu, aby upraszczać strukturę dowodu, a w skrajnym przypadku nawet ujawniać pewne niezbyt potrzebne kroki. Stosowana jest tutaj taka pragmatyczna zasada, że w uzasadnieniach chcemy w jak największym stopniu wykorzystywać tylko te elementy przesłanki, które są istotnie potrzebne – jeśli w uzasadnieniu danego kroku używana jest przesłanka o skomplikowanej treści, ale tylko część tej treści jest potrzebna w uzasadnieniu, to o ile to możliwe, warto odwołać się bezpośrednio do tej istotnej części. Obecność kroków zbierających wiedzę, z której potem należy odfiltrować nieistotną część, wpływa w istotny sposób ujemnie na czytelność dowodu. Stąd podjęty został wysiłek, aby tego rodzaju sytuacje wyeliminować.

Warto zwrócić uwagę, że nie jest do końca jasne, czym jest wspomniana „istotna część treści". W związku z tym w bibliotece MML wprowadzono pewne pragmatyczne rozwiązanie, które jedynie przybliża opisany tutaj proces. Mianowicie uznano, że istotna część treści da się wyodrębnić z danej przesłanki α, jeśli w danym kroku β można zastąpić odwołanie do α przez pewien podzbiór przesłanek uzasadniających α. Przy tym w skrajnym przypadku – gdy α służy wyłącznie zebraniu kilku przesłanek we wspólną, oczywistą konkluzję z nich wynikającą – każde odwołanie do α da się zastąpić przez odwołania do (części) jej bezpośredniego uzasadnienia. W ten sposób odwołania do α znikną i możliwe stanie się usunięcie tego kroku za pomocą programu Inacc.

Tego rodzaju upraszczanie bazy MML zostało podjęte przy pełnej świadomości faktu, iż może ono prowadzić do zmniejszenia czytelności skryptów w wyniku pojawienia się długich list referencji w uzasadnieniach. Zagadnienie systematycznego skracania takich list jest otwartym problemem badawczym.

Dokonywany przez RenInfer wybór podzbioru zbioru referencji wskazanych przez RelInfer, które następnie są zastępowane odpowiednimi listami, jest dokonywany rekurencyjnie w oparciu o trzy poniższe kryteria przy ustalonej ich gradacji: 1, 2, 3.

1. Wybierane referencje odwołują się do przesłanki (każda swojej własnej), do której łączna liczba odwołań w skrypcie dowodowym nie wskazanych przez RelInfer jest minimalna.

2. Gdy zbiór referencji z punktu 1. ma moc > 1, wybierane referencje wskazują na przesłankę (każda swoją własną), która jest wykorzystywana jako przesłanka w uzasadnieniach różnych kroków minimalną liczbę razy.

3. Gdy zbiór referencji z punktu 2. ma moc > 1, wybierane referencje wskazują na przesłankę (każda swoją własną), która w swoim uzasadnieniu wykorzystuje minimalną liczbę przesłanek wskazanych przez RelInfer.

Kryterium 1. wyodrębnia te kroki (przesłanki), których pełne sformułowanie jest wykorzystywane możliwie rzadko. W wyniku jego zastosowania w szczególności ujawniane są takie kroki, których sformułowanie zawsze jest wykorzystywane jedynie częściowo, co jak żeśmy powyżej opisali, prowadzi do ich eliminacji. Dodatkowo przetwarzanie w pierwszej kolejności kroków rzadko w pełni wykorzystywanych prowadzi do wykrycia niepotrzebnych kroków przy zastosowaniu RenInfer.

Jeśli kryterium 1. wskazuje wiele referencji, to staramy się dokonać wyboru spełniającego kryterium 2. Kryterium 2. wymusza wybór tych spośród referencji, które odwołują się do rzadko wykorzystywanych przesłanek.

W przypadku, w którym oba przedstawione kryteria nie dają jednoznacznego wskazania, w celu uzyskania jeszcze lepszego efektu dodatkowo stosujemy kryterium 3. Zastosowanie kryterium 3. ma na celu zmniejszenie liczby referencji wskazanych przez program RelInfer, które pozostają niezmodyfikowane po jednokrotnym zastosowaniu programu RenInfer. Skrypty dowodowe, w których zostały zmodyfikowane wszystkie referencje wskazywane przez RelInfer, mogą okazać się niemożliwe do zaakceptowania przez weryfikator systemu Mizar. Przypuśćmy, że kroki s_1, s_2 mają w uzasadnieniu referencje R_1, R_2 odpowiednio, wskazane przez RelInfer. Dodatkowo, niech referencja R_2 odwołuje się do kroku s_1. Wówczas uzasadnienie kroku s_2, w którym referencja R_2 została zastąpiona listą przesłanek wykorzystywanych do uzasadnienia kroku s_1, gdzie dodatkowo występująca referencja R_1 została zastąpiona odpowiednią listą, może okazać się nieakceptowalne przez weryfikator systemu Mizar. W celu zagwarantowania poprawności zmodyfikowanego rozumowania, referencja R_2 jest zastępowana oryginalną listą przesłanek wykorzystywanych do uzasadnienia kroku s_1. Sprawdzenie, czy możliwe jest zmodyfikowanie referencji R_1 w uzyskanym uzasadnieniu kroku s_2, wymaga ponownego wykorzystania programu RelInfer, a następnie ponownej analizy wskazanych uchybień w programie RenInfer.

3.3.2 Narzędzia umożliwiające rozbijanie koniunkcji w formułach

Działanie narzędzi analizowanych w poprzedniej sekcji skupiało się głównie wokół odnajdywania niepotrzebnych przesłanek lub modyfikacji wybranych referencji w celu odnalezienia bardziej elementarnych uzasadnień. Nie uwzględniały one jednak faktu, że stwierdzenia, do których odwołuje się uzasadnienie konkretnego kroku, mogą być koniunkcją kliku formuł, z których nie wszystkie muszą być wykorzystywane w tym uzasadnieniu. Łączenie kilku formuł za pomocą koniunkcji w pojedynczych krokach jest dozwolone w systemie Mizar, ale uzasadnione jedynie w przypadku, gdy wybrana grupa formuł opisuje fakty, które zdaniem autora rozumowania wspólnie opisują jedną sytuację oraz jako przesłanki występują na ogół razem w uzasadnieniach. Chaotyczny dobór takich grup formuł nie tylko zwiększa czas weryfikacji przez maszynę, ale na skutek istnienia niepotrzebnych przesłanek w uzasadnieniu, istotnie obniża jego czytelność. W skrajnym przypadku taki rodzaj połączeń umożliwia nawet powstanie twierdzeń, które posiadają założenia w rzeczywistości niewykorzystywane w ich dowodzie. Problem ten zostanie szerzej opisany w dalszej części sekcji. Z badań przeprowadzonych na bazie MML [70], które zostały omówione w sekcji 3.3.3, wynika, że

problem ten dotyczył 541 spośród ponad 30 tys. zgromadzonych twierdzeń, a ręczne poprawienie tego byłoby niezwykle pracochłonne.

W celu przedstawienia najczęściej występujących rodzajów chaotycznych połączeń grup faktów za pomocą koniunkcji oraz narzędzi umożliwiających eliminowanie tych połączeń rozważmy dowód zapisany w systemie Mizar zawierający dwa „równoległe" rozumowania przedstawiony na wydruku 3.14. Zakładamy, że system Mizar na

α_1 `implies` α_n	β_1 `implies` β_n	α_1 `&` β_1 `implies` α_n `&` β_n
`proof`	`proof`	`proof`
`assume A1:` α_1`;`	`assume B1:` β_1`;`	`assume AB1:` α_1 `&` β_1`;`
`A2:` α_2 `by A1;`	`B2:` β_2 `by B1;`	`AB2:` α_2 `&` β_2 `by AB1;`
`A3:` α_3 `by A2;`	`B3:` β_3 `by B3;`	`AB3:` α_3 `&` β_3 `by AB2;`
$\vdots$	$\vdots$	$\vdots$
`An:` α_n`by An-1;`	`Bn:` β_n`by Bn-1;`	`ABn:` α_n `&` β_n `by ABn-1;`
`thus thesis by An;`	`thus thesis by Bn;`	`thus thesis by ABn;`
`end;`	`end;`	`end;`

Wydruk 3.14: Przykład dowodu powstałego w wyniku „równoległego" połączenia dwóch rozumowań.

podstawie formuły α_i nie może uzasadnić β_j, oraz formuły α_i na podstawie β_j, gdzie $1 \leqslant i, j \leqslant n$. Naturalnie, jeśli oba rozumowania były akceptowalne przez weryfikator systemu Mizar, to skonstruowany dowód jest również akceptowalny. Zauważmy, że powielenie wybranych kroków w rozumowaniach przed ich połączenie, w celu uzyskania równej długości obu rozumowań, również nie generuje błędów w rozumowaniu. Dodatkowo odpowiedni sposób powielenia wybranych kroków uniemożliwia ich wykrycie nawet przy pomocy omówionych dotychczas programów tj. RelInfer. Precyzując, referencja `AB`i zostanie wskazana przez RelInfer tylko w przypadku jeśli, przed połączeniem rozumowań obie referencje `A`i, `B`i były wskazane przez ten program. W szczególności, jeśli przedstawione dotąd narzędzia dystrybuowane z systemem Mizar nie wskazują żadnych usterek w skrypcie dowodowym, to dopisanie frazy `&(0=0)` lub innego równie oczywistego faktu poprzedzonego `&` do stwierdzania sformułowanego w dowolnym kroku rodzaju <*Compact-Statement*> nie zostanie wykryte jako usterka przez wspomniane programy.

Innym rodzajem wady napotykanym w dowodach zawierających równoległe fragmenty rozumowania jest sytuacja, gdy autor skryptu dowodowego w trakcie jego pisania usunął wybrane konkluzje z dowodzonego faktu (np. β_n), nie modyfikując równocześnie odpowiednich list założeń i dowodu. Oczywiście usunięte konkluzje mogą znajdować się na każdym poziomie zagnieżdżenia (wydruk 3.15), co dodatkowo utrudnia proces odnajdywania nieuzasadnionych koniunkcyjnych połączeń faktów w poszczególnych krokach rozumowania.

Przypadkowe równoległe połączenie kilku rozumowań lub ich fragmentów wpływa nie tylko na czas weryfikacji poprawności skryptów dowodowych, ale również istotnie obniża ich czytelność. Zauważmy, że odwołanie się do przesłanki, która jest sformułowana w postaci koniunkcji kilku faktów, zmusza czytelnika do samodzielnego zbadania, które spośród tych faktów są rzeczywiście wykorzystywane w tym uzasadnieniu. Analogiczna sytuacja występuje również w przypadku, kiedy samo uzasadniane stwierdzenie ma postać koniunkcji faktów. Przypadek, w którym zarówno wykorzystywana przesłanka, jak i uzasadniane stwierdzenie jest koniunkcją faktów został przedstawiony w kroku $\eta \wedge \zeta$ (wydruk 3.16).

Jedną z najbardziej intuicyjnych metod rozwiązywania problemów przedstawionych w tej sekcji jest „rozbicie" w rozumowaniach wszystkich kroków, które stwier-

```
α1 & β1 implies α4
proof
  assume A1: α1 & β1;
  A2: α4 & β4
  proof
    B1: α2 & β2 by A1;
    B2: α3 & β3 by B1;
    B3: α4 & β4 by B2;
    thus thesis by B3;
  end;
  thus thesis by A2;
end ;
```

Wydruk 3.15: Przykład dowodu, w którym została usunięta konkluzja jednego z równoległych rozumowań, bez modyfikacji pozostałej części skryptu.

```
          theorem
     α:    i in Seg n implies i+m in Seg(n+m)
          proof
     β:     assume i in Seg n ;
γ∧δ∧ϵ:     then 1<=i & i<=i+m & i<=n by FINSEQ_1:3, NAT_1:11;
  η∧ζ:     then i+m<=n+m & 1<=i+m by XREAL_1:9, XXREAL_0:2;
     θ:     hence thesis by FINSEQ_1:3;
          end ;
```

Wydruk 3.16: Modyfikacja rozumowania przedstawionego na rys. 3.13, zawierająca równoległe fragmenty rozumowania.

dzają koniunkcję faktów, a następnie połączenie za pomocą koniunkcji tylko tych stwierdzeń, które spełniają ściśle określone kryteria. Metoda ta jest realizowana za pomocą pięciu poniższych programów, które zostały stworzone dla przeprowadzenia wstępnych badań nad poprawą czytelności rozumowań w systemie Mizar [70].

BreakBinaryAnd – program ten „rozbija" koniunkcje występujące w krokach rozumowania, które zostały uzasadnione przez wskazanie listy referencji (<*Simple-Justification*>). Rozbijanie koniunkcji jest realizowane w oparciu o zastąpienie każdego kroku stwierdzającego koniunkcję formuł ciągiem kroków, z których każdy stwierdza pojedynczą formułę składową pierwotnej koniunkcji. Dodatkowo wszystkie uzasadnienia w każdym ciągu kroków są identyczne z uzasadnieniem odpowiedniego „rozbijanego" kroku, natomiast każde odwołanie do „rozbitego" kroku jest zastępowane ciągiem referencji do wszystkich kroków z odpowiedniego ciągu.

DelBlock – program ten „rozbija" koniunkcje występujące w każdym kroku uzasadnionym za pomocą zagnieżdżonego rozumowania, jeśli w rozumowaniu tym każdy krok rodzaju <*Skeleton-Step*> jest również rodzaju <*Conclusion*>. W ogólnym przy-

padku, modyfikacja wykonywana przez ten program ma postać:

```
<Reasoning>_0                                  <Reasoning>_0
<Proposition>_0
  proof
    <Reasoning>_1                              <Reasoning>_1
    thus <Proposition>_1 <Justification>_1     T1: <Proposition>_1 <Justification>_1
    <Reasoning>_2                              <Reasoning>_2
    thus <Proposition>_2 <Justification>_2     T2: <Proposition>_2 <Justification>_2
    ⋮                                          ⋮
    <Reasoning>_n                              <Reasoning>_n
    thus <Proposition>_n <Justification>_n     Tn: <Proposition>_n <Justification>_n
  end;                                         <Proposition>_0 by T1,T2,...,Tn;
<Reasoning>_{n+1}                              <Reasoning>_{n+1}
```

(3.36)

Rozbicie zagnieżdżonego uzasadnienia kroku $<Proposition>_0$ umożliwia usunięcie nie tylko „nadmiarowych" koniunktów mogących występować w sformułowaniu tego kroku, ale również fragmentów zagnieżdżonego rozumowania odpowiedzialnych za uzasadnienie tych koniunktów.

TrivConsider – program ten eliminuje zmienne ustalone wprowadzone do rozumowania za pomocą konstrukcji `consider`, dla których:

(*i*) identyfikator tej zmiennej ustalonej nie jest wykorzystywany w rozumowaniu poza krokiem, w którym został wprowadzony,

(*ii*) stwierdzenie sformułowane w kroku wprowadzającym tę zmienną do rozumowania jest wykorzystywane co najmniej raz jako przesłanka w rozumowaniu.

Działanie tego programu opiera się na zastąpieniu każdego kroku wprowadzającego niewykorzystywaną zmienną ustaloną do rozumowania:

`consider` <*Qualified-Variable*> `such that`
[<*Label-Identifier*>`:`] <*Formula-Expression*> <*Simple-Justification*>`;`, (3.37)

przez krok stwierdzający formułę egzystencjalną:

[<*Label-Identifier*>`:`] `ex` <*Qualified-Variable*> `st`
<*Formula-Expression*> <*Simple-Justification*>`;`. (3.38)

Wykorzystanie tego programu umożliwiło eliminację 6186 użyć konstrukcji `consider` w bazie MML, co stanowiło 11,83% spośród wszystkich istniejących użyć tej konstrukcji.

MergeItems – program ten w pierwszym etapie odnajduje w ramach rozumowań pierwotnych pary kroków, które:

(*i*) jako przesłanki występują zawsze razem w uzasadnieniach

(*ii*) takie, dla których żaden element nie jest osiągalny z drugiego elementu pary w domknięciu grafu dowodu.

W drugim etapie część spośród odnalezionych par, która spełnia dodatkowe własności, jest zastępowana nowymi krokami. Każdy taki nowy krok stwierdza wówczas koniunkcję faktów sformułowanych w modyfikowanej parze, przy czym jego uzasadnienie powstaje w wyniku połączenia zbiorów przesłanek obu kroków z pary. Dodatkowe własności, jakie muszą spełniać pary, są przekazywane do programu za pomocą

parametrów wywołania umożliwiających uniknięcie generowania rozbudowanych uzasadnień. Określają one poziom „zgodności" uzasadnień wybieranych par kroków. Precyzując, para kroków u_1, u_2 należąca do rozumowania pierwotnego $\mathfrak{A}$ jest zastępowana nowym krokiem, jeśli dla każdej przesłanki p, należącej do różnicy symetrycznej zbiorów przesłanek uzasadniających u_1 oraz u_2, zachodzi:

`-Thm`: przesłanka została sformułowana poza obszarem twierdzenia, którego dowód zawiera rozumowanie pierwotne $\mathfrak{A}$,

`-Block`: przesłanka została sformułowana poza obszarem rozumowania pierwotnego $\mathfrak{A}$,

`-L`: jeśli p jest przesłanką kroku u_1 (u_2) oraz nie istnieje linearyzacja τ, dla której p i u_1 (u_2) należą do tego samego τ–łańcucha,

`-D[liczba]`: jeśli p jest przesłanką kroku u_1 (u_2) oraz nie istnieje linearyzacja τ, dla której τ–rozpiętość łuku (p, u_1) (lub (p, u_2)) jest mniejsza od zadanej liczby.

Kroki, które są wybierane do „łączenia", domyślnie nie mogą stwierdzać koniunkcji formuł. Ograniczenie to może jednak zostać pominięte po wykorzystaniu dodatkowego parametru wywołania `-MoreBC`.

SortItem – program ten reorganizuje kolejność kroków w linearyzacji rozumowania, wykorzystując algorytm zachłanny do optymalizacji wyznaczników poprawy czytelności zaproponowanych w podrozdziale 3.2. Program optymalizuje wartość czterech pierwszych wskaźników w domyślnej hierarchii ważności 1, 3, 2, 4 (lub 4, 1, 3, 2 przy zastosowaniu parametru wywołania `-MinDist`). Uzyskiwana linearyzacja rozumowania jest generowana w oparciu o rekurencyjną konkatenację par zlinearyzowanych segmentów rozumowania w każdym rozumowaniu pierwotnym. Naturalnie wybór pary zlinearyzowanych segmentów dokonywany na każdym etapie rekurencji w ustalonym rozumowaniu pierwotnym $\mathfrak{A}$ nie generuje konfliktów w grafie dowodu. Precyzując:

(i) pierwszy element pary, traktowany jako zbiór wierzchołków rozumowania, nie jest osiągalny z drugiego elementu pary, traktowanego również jako zbiór wierzchołków rozumowania w grafie partycji domkniętego rozumowania pierwotnego $\mathfrak{A}$, wyznaczonego przez zbiory wierzchołków zlinearyzowanych segmentów,

(ii) partycja ta jest acykliczna pomimo konkatenacji wybranej pary zlinearyzowanych segmentów.

Wybór pary zlinearyzowanych segmentów $\langle F^1, F^2\rangle$, gdzie $F^1 = \langle f^1_1, f^1_2, \ldots, f^1_{n_1}\rangle$, $F^2 = \langle f^2_1, f^2_2, \ldots, f^2_{n_2}\rangle$, będącej lokalnym optimum przy ustalonej hierarchii ważności (wśród wszystkich par spełniających warunki (i), (ii)) jest realizowany w oparciu o cztery poniższe warunki, które są odpowiednikami poszczególnych kryteriów poprawy czytelności, wyszczególnionych na str. 57:

1) Krok $f^1_{n_1}$ jest przesłanką kroku f^2_1 oraz przesłanka $f^1_{n_1}$ może zostać przekazana do uzasadnienia kroku f^2_1 za pomocą konstrukcji `then` ($(f^1_{n_1}, f^2_1) \in \mathtt{then}_{\mathfrak{P}}$).

2) Liczba referencji łącząca zbiór $\mathcal{V}(F^1) \cup \mathcal{V}(F^2)$ z pozostałymi wierzchołkami w grafie dowodu jest minimalna. Precyzując, wyrażenie:

$$|\left(\mathcal{V}(\mathfrak{P}) \setminus (\mathcal{V}(F^1) \cup \mathcal{V}(F^2))\right) \underset{\mathcal{R}ef_{\mathfrak{P}}}{\curvearrowright} (\mathcal{V}(F^1) \cup \mathcal{V}(F^2))| + |(\mathcal{V}(F^1) \cup \mathcal{V}(F^2)) \underset{\mathcal{R}ef_{\mathfrak{P}}}{\curvearrowright} \left(\mathcal{V}(\mathfrak{P}) \setminus (\mathcal{V}(F^1) \cup \mathcal{V}(F^2))\right)| \quad (3.39)$$

ma wartość minimalną.

3) Krawędź $(f^1_{n_1}, f^2_1) \in \mathcal{R}ef_{\mathfrak{P}}$ oraz zbiór $|\mathcal{N}^-_{\mathcal{R}ef_{\mathfrak{P}}}(f^1_{n_1})|$ ma minimalną liczność.

4) Wzrost sumy τ–rozpiętości łuków referencyjnych, związany z powstaniem nowych wierzchołków v, dla których w zlinearyzowanym rozumowaniu będą istniały referencje $\langle p, u\rangle \in \mathcal{R}ef_{\mathfrak{P}}$, takie że $\neg(p \underset{\mathfrak{D}\widetilde{\mathfrak{P}}}{\rightarrow}{}^* v)$ lub $\neg(v \underset{\mathfrak{D}\widetilde{\mathfrak{P}}}{\rightarrow}{}^* u)$, jest minimalny. Precyzując, wyrażenie:

$$n_1 \cdot \left(\sum_{i=1}^{n_2} |\mathcal{N}^+_{\mathcal{R}ef_{\mathfrak{P}}}(f^2_i)| \right) + n_2 \cdot \left(\sum_{i=1}^{n_1} |\mathcal{N}^-_{\mathcal{R}ef_{\mathfrak{P}}}(f^1_i)| \right) - (n_1 + n_2) \cdot |(\mathcal{V}(F^1) \cup \mathcal{V}(F^2)) \underset{\mathcal{R}ef_{\mathfrak{P}}}{\curvearrowright} (\mathcal{V}(F^1) \cup \mathcal{V}(F^2))| \quad (3.40)$$

ma wartość minimalną.

3.3.3 Rewizja bazy MML wykorzystująca rozbijanie koniunkcji w formułach – wyniki statystyczne

Grupa programów, które zostały przedstawione w sekcji 3.3.2, umożliwiła przeprowadzenie badanie sensowności modyfikacji struktur rozumowań zgromadzonych w bazie MML, w celu wyeliminowania niewykorzystywanych przesłanek z uzasadnień. Badania te zostały przeprowadzone na wersji MML 4.121.1054, a uzyskane dzięki niej modyfikacje struktur po pozytywnej ocenie zostały wdrożone w wersji bazy MML 4.127.1060 [70].

Przeprowadzona poprawa struktur rozumowań dostarczyła również danych o liczbie niewykorzystywanych przesłanek w uzasadnieniach. Zastąpienie wszystkich kroków w rozumowaniach, które stwierdzały koniunkcję formuł odpowiednimi ciągami formuł, umożliwiło bowiem odnalezienie 755196 nadmiarowych referencji (28,6% z pośród wszystkich referencji występujących w skryptach dowodowych zgromadzonych w MML). Dodatkowo 2% wśród pozostałych referencji zostało wskazane przez program RelInfer. Modyfikacja poszczególnych referencji w celu usunięcia tych wskazań umożliwiła odnalezienie 38944 kroków w rozumowaniach (2,8% spośród wszystkich kroków w bazie), które przestały być wykorzystywane w rozumowaniu. Dodatkowym następstwem modyfikacji sposobu uzasadnień kroków w rozumowaniach było odnalezienie 461 twierdzeń, które posiadały założenia niewykorzystywane w toku rozumowania. Naturalnie autorzy artykułów, którzy odwoływali się do tych twierdzeń, musieli zakładać lub dowodzić te niewykorzystane założenia. Stąd, eliminacja tych założeń pozwoliła na odnalezienie i usunięcie fragmentów rozumowań, które były wykorzystywane jedynie do uzasadnienia tych nadmiarowych założeń, jak również pozwoliła na odnalezienie kolejnych twierdzeń posiadających niewykorzystywane założenia w toku rozumowania.

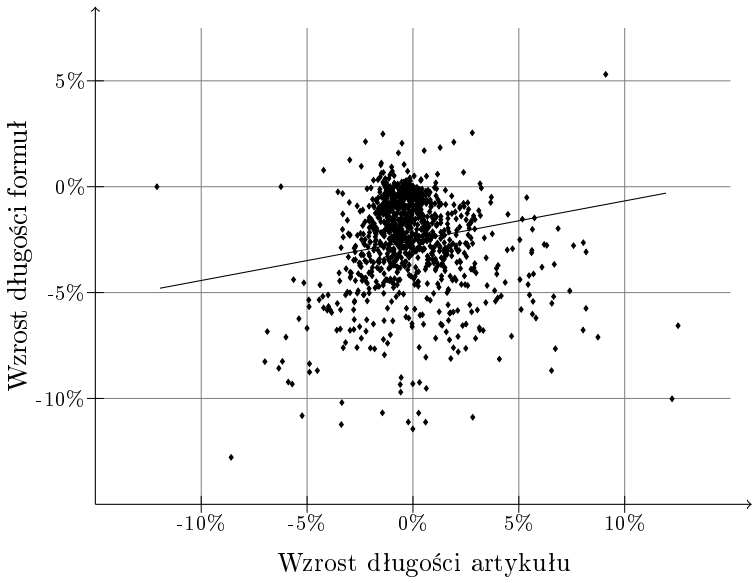
Etap	Liczba usuniętych referencji	Liczba wskazań programu RelInfer	Liczba kroków usuniętych z rozumowania	Liczba zmodyfikowanych artykułów	Liczba zmodyfikowanych twierdzeń
1	755196	37304	38944	1017	461
2	160	23	633	118	64
3	596	2	168	55	16

Istnienie nadmiarowych założeń w toku rozumowania nie zawsze jest jednak następstwem usunięcia przez autora wybranych konkluzji w sformułowaniu twierdzenia. Analiza sformułowań zmodyfikowanych twierdzeń, które zostały wzmocnione dzięki

eliminacji niewykorzystywanych założeń, wykazała, że istnienie nadmiarowych założeń mogło wynikać z różnic między definicjami pojęć w bazie MML oraz literaturą, na podstawie której była dokonywana formalizacja. Jednym z takich twierdzeń jest lemat wykorzystywany w uzasadnieniu twierdzenia Nagata-Smirnov (wydruk 3.17), sformalizowany w artykule [68] na podstawie dowodu zawartego w książce [22]. Jego dowód po sformalizowaniu zawierał 77 kroków, znajdujących się na pięciu poziomach zagnieżdżenia. W konwencji przyjętej przez R. Engelkinga [22], przestrzeń topologiczna jest regularna lub krócej T_3, jeśli jest T_1 i spełnia warunek regularności. Natomiast w bazie MML, przestrzenią regularną jest nazywana przestrzeń topologiczna spełniająca jedynie warunek regularności, a T_3 jest regularną przestrzenią T_1. Interpretacja własności regularności przestrzeni topologicznej, wyrażona w języku Mizar w postaci "T is regular & T is T_1" oraz wykorzystanie narzędzi do rozbijania koniunkcji umożliwiły automatyczne wywnioskowanie, że przesłanka "T is T_1" nie jest wykorzystywana w dowodzie tego lematu.

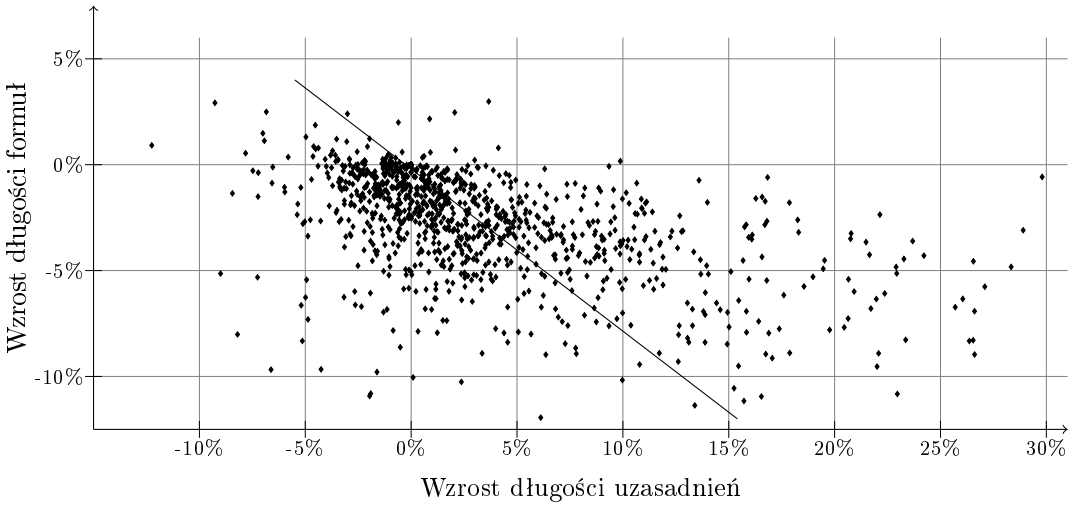
```
theorem :: NAGATA_1:20
  for T being non empty TopSpace st T is regular & T is T_1 &
    ex Bn being FamilySequence of T st
      Bn is Basis_sigma_locally_finite
  holds T is normal;
```

Wydruk 3.17: Lemat wykorzystywany w dowodzie twierdzeniu Nagata-Smirnov, sformułowany w języku Mizar.

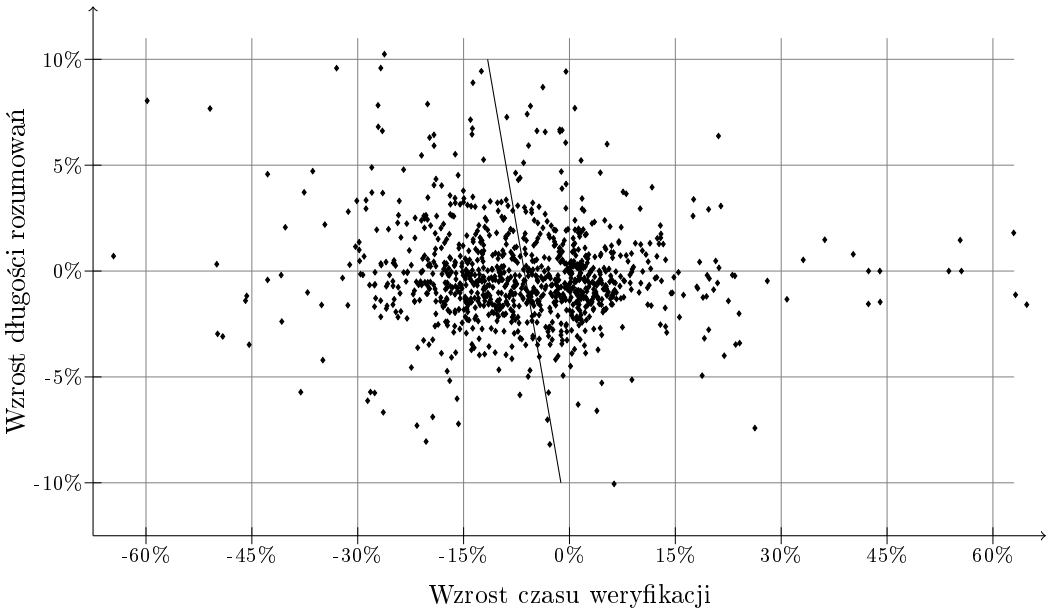


Rysunek 3.18: Modyfikacja długości skryptów dowodowych w zależności od długości uzasadnianych formuł jako następstwo rozbicia koniunkcji w formułach. Przedstawiona prosta regresji przedstawia współzależność zmiennej objaśniającej – wzrostu długości artykułu, względem zmiennej objaśnianej – wzrostu długości formuł.

Usunięcie łącznie 39745 kroków z rozumowań zgromadzonych w bazie MML nie wpłynęło jednak istotnie na długość skryptów dowodowych mierzoną w jednostkach leksykalnych. Średnia długość skryptu zmniejszyła się bowiem o 0,21% (rys. 3.18).



Rysunek 3.19: Zmiana długości uzasadnień kroków rozumowań w zależności od długości uzasadnianych formuł jako następstwo rozbicia koniunkcji w formułach. Przedstawiona prosta regresji przedstawia współzależność zmiennej objaśniającej – wzrostu długości uzasadnień, względem zmiennej objaśnianej – wzrostu długości formuł.



Rysunek 3.20: Modyfikacja długości skryptu dowodowego w zależności od czasu weryfikacji jego poprawności systemem Mizar jako następstwo rozbicia koniunkcji w formułach. Przedstawiona prosta regresji przedstawia współzależność zmiennej objaśniającej – wzrostu czasu weryfikacji, względem zmiennej objaśnianej – wzrostu długości rozumowań.

Główną przyczyną tak małego spadku długości skryptów jest fakt, że „rozbijanie" koniunkcji i usuwanie nadmiarowych kroków zmniejszyło łączną długość formuł w krokach rozumowania średnio o 3,02%, ale równocześnie zwiększyło łączną długość uzasadnień średnio o 3,79% (rys. 3.19), co jest następstwem powielania uzasadnień w trakcie „rozbijania" koniunkcji.

Rozbicie stwierdzeń sformułowanych w postaci koniunkcji faktów wpłynęło również na czas weryfikacji poprawności skryptów dowodowych zgromadzonych w bazie MML. Średni czas weryfikacji uległ bowiem skróceniu średnio o 5,13% (rys. 3.20).

3.3.4 Metody eliminacji konstrukcji `reconsider`

W prowadzonych dotąd rozważaniach było zakładane, że przepływ informacji o wykorzystywanych przesłankach między dowolnymi dwoma krokami w rozumowaniu jest obrazowany jedynie za pomocą łuków referencyjnych. Przy tym założeniu łuki porządkujące dostarczały jedynie informacji o wykorzystanych identyfikatorach zmiennych ustalonych oraz budowie szkieletu rozumowania. Założenie to nie jest jednak prawdziwe, jeśli w skrypcie dowodowym zapisanym w języku Mizar wykorzystywana była konstrukcja `reconsider`. Konstrukcja ta umożliwia bowiem jednoznaczne przyporządkowanie typu do termu, jeśli na podstawie zgromadzonych informacji w rozumowaniu możliwe jest uzasadnienie, że rozważany term ma m.in. dany typ. Przyporządkowanie to jest realizowane poprzez wprowadzenie do rozumowania nowej zmiennej ustalonej określonego typu, która jest utożsamiana z tym termem przez weryfikator systemu Mizar. Informacja o tym utożsamieniu nie jest jednak przechowywana w grafie dowodu.

W celu przedstawienia ukrytych zależności w grafie dowodu, rozważmy fragment rozumowania z wydruku 3.21. Rozumowanie to jest akceptowalne przez system Mi-

```
α:  let X be non empty set;
β:  let f be Function of X,X;
γ:  assume A1: f is one-to-one onto;
δ:  reconsider P=f as Permutation of X by A1, FUNCT_2: def 4;
ϵ:  A2: P is Function-like;
ζ:  A3: f is one-to-one & f is onto by A2;
```

Wydruk 3.21: Przykład rozumowania w języku Mizar, zawierającego ukryty łuk referencyjny (γ, ζ) odwołujący się do równości `P=f`.

zar. Zauważmy, że uzasadnienie kroku ζ, które odwołuje się tylko do stwierdzenia `P is Function-like`, wydaje się niedostateczne do udowodnienia stwierdzenia `f is one-to-one & f is onto`. Poprawność uzasadnienia tego kroku nie wynika bowiem z informacji zawartej w tej przesłance, a jedynie z faktu, że przesłanka ta wykorzystuje w swym sformułowaniu identyfikator zmiennej ustalonej `P`. Zmienna ustalona `P` jest bowiem domyślnie utożsamiana ze zmienną ustaloną `f`, skąd przy odwołaniu do kroku ϵ, lista przesłanek w ζ jest niejawnie powiększana przez weryfikator o równość `P=f`. Zauważmy również, że stwierdzenie `P is Function-like` może zostać zastąpione przez dowolny oczywisty fakt, który wykorzystuje identyfikator `P` w swym sformułowaniu, np. `P = P`.

Jak łatwo można zauważyć, każdy „ukryty" łuk referencyjny (v, u) w abstrakcyjnym grafie dowodu $\mathfrak{P}$ jest następstwem istnienia ścieżki skierowanej $v \underset{\mathcal{O}rd_{\mathfrak{P}}}{\rightarrow} w \underset{\mathcal{O}rd_{\mathfrak{P}}}{\rightarrow} u$, gdzie $w \in \mathcal{V}(\mathfrak{P})$ oraz wykorzystania konstrukcji `reconsider` w kroku v. Istnienie jednak takiej ścieżki nie jest warunkiem dostatecznym istnienia ukrytego łuku referencyjnego – identyfikacja termu ze zmienną ustaloną nie musi być wykorzystywana

w uzasadnieniu. Odnajdywanie ukrytych łuków referencyjnych w strukturze grafu dowodu jest więc niemożliwe, nawet w przypadku, jeśli struktura tego grafu zawierałaby informację o zbiorze wierzchołków, w których została wykorzystana konstrukcja `reconsider`. Tym samym nie jest więc możliwe dokładne wyznaczenie zbioru przesłanek dowolnej paczki, jak również zbadanie, czy dana paczka jest domknięta na prowadzenie dróg skierowanych.

Zadowalającym jednak rozwiązaniem przedstawionego problemu, jest wyeliminowanie ukrytych łuków referencyjnych z grafu dowodu poprzez usunięcie konstrukcji `reconsider` ze skryptów dowodowych. Eliminacja ta może zostać zrealizowana w dwóch opisanych poniżej etapach.

Etap 1. – eliminacja kroków rozumowania wykorzystujących konstrukcję `reconsider`, w których został nadpisany identyfikator zmiennej (zob. wydruk 3.22). W etapie tym jest zastępowany rekurencyjnie każdy krok rozumowania postaci:

```
reconsider {<Variable-Identifier> = <Term-Expression>,}*
             <Variable-Identifier>
           {,{<Variable-Identifier> |<Variable-Identifier> = <Term-Expression> }}*
        as <Type-Expression> <Simple-Justification> ;
```
(3.41)

przez

```
reconsider {<Variable-Identifier> = <Term-Expression>,}*
             <New-Variable-Identifier> = <Variable-Identifier>
           {,{<Variable-Identifier> |<Variable-Identifier> = <Term-Expression> }}*
        as <Type-Expression> <Simple-Justification> ;
```
(3.42)

oraz wszystkie wystąpienia *<Variable-Identifier>* przez *<New-Variable-Identifier>* w sformułowaniach kroków, które w skrypcie dowodowym znajdują się po kroku (3.41), gdzie *<New-Variable-Identifier>* jest niewykorzystywanym dotąd w skrypcie dowodowym identyfikatorem zmiennej ustalonej.

```
α:  let x be set such that A1: x in NAT;
β:  reconsider x as Nat by A1;
```

Wydruk 3.22**:** Fragment rozumowania zapisanego w języku Mizar, zawierający nadpisanie typu zmiennej ustalonej `x` w kroku wykorzystującym konstrukcję `reconsider`.

Etap 2. – zastąpienie kroków wykorzystujących konstrukcję `reconsider` przez wygenerowane na ich podstawie kroki wykorzystujące konstrukcję `consider`. W etapie tym zastępowany jest każdy krok postaci:

```
reconsider <Variable-Identifier>_1 = <Term-Expression>_1,
           <Variable-Identifier>_2 = <Term-Expression>_2,
           ⋮
           <Variable-Identifier>_n = <Term-Expression>_n
        as <Type-Expression> <Simple-Justification> ;
```
(3.43)

przez

```
consider <Variable-Identifier>_1, <Variable-Identifier>_2, ..., <Variable-Identifier>_n
        be <Type-Expression> such that
<New-Label-Identifier>: <Variable-Identifier>_1 = <Term-Expression>_1 &
                        <Variable-Identifier>_2 = <Term-Expression>_2 &
                        ⋮
                        <Variable-Identifier>_n = <Term-Expression>_n
        <Simple-Justification>;
```

(3.44)

oraz etykieta <*New-Label-Identifier*> jest dopisywana do każdego uzasadnienia kroku, który w sformułowaniu wykorzystuje identyfikator <*Variable-Identifier*> lub odwołują się do przesłanek wykorzystujących ten identyfikator w sformułowaniu, gdzie <*New-Label-Identifier*> jest nie wykorzystywaną dotąd etykietą w skrypcie dowodowym. Naturalnie, nie wszystkie dopisane etykiety <*New-Label-Identifier*> w uzasadnieniach są konieczne do zachowania dostateczności uzasadnień w zmodyfikowanym rozumowaniu. Problem odnajdywania i usuwania z rozumowania niepotrzebnych etykiet został omówiony w sekcji 3.3.1 i zostanie pominięty w tym miejscu.

Rozdział 4

Złożoność problemów reorganizujących kolejność kroków w rozumowaniu

Analiza metod poprawy czytelności opierających się o modyfikację kolejności kroków rozumowania, przeprowadzona w podrozdziale 3.2, umożliwiła sformułowanie pięciu głównych problemów grafowych. W niniejszym rozdziale zostanie zbadana ich złożoność.

4.1 NP-zupełność problemów $\mathcal{K}.1'$, $\mathcal{K}.2'$

W trakcie formalizacji dwóch pierwszych metod optymalizacyjnych, został określony związek między tymi metodami w szczególnej rodzinie acyklicznych digrafów. Korzystając ze Stw. 3.13 oraz Lem. 3.14 możemy bowiem stwierdzić, że problem $\mathcal{K}.1$ ma rozwiązanie wtedy i tylko wtedy, gdy ma rozwiązanie problem $\mathcal{K}.2$, przy ograniczeniu zbioru instancji tych problemów do rodziny acyklicznych digrafów z wyróżnionym zbiorem łuków nie zawierającym skrótów oraz przy odpowiedniej zależności między parametrami K_1, K_2, gdzie K_1 oraz K_2 są liczbami naturalnymi występującymi odpowiednio w instancji problemów $\mathcal{K}.1'$ oraz $\mathcal{K}.2'$. Obserwacja ta jest oczywistym następstwem równości:

$$\sum_{\substack{P_1,P_2\in\tau_A\\ P_1\neq P_2}} |P_1 \underset{A}{\curvearrowright} P_2| + |\mathbf{1}_\tau^A| = |A|, \tag{4.1}$$

gdzie τ jest sortowaniem topologicznym digrafu D z wyróżnionym zbiorem $A \subseteq \mathcal{A}(D)$, dla którego D nie zawiera A–skrótów, a zależność między parametrami K_1, K_2 dana jest równaniem: $K_1 + K_2 = |A|$.

Zależność (4.1) wyrażona w terminach acyklicznej $\mathcal{H}_A^{(*)}$–partycji digrafu D określa analogiczną zależność między problemami $\mathcal{K}.1'$, $\mathcal{K}.2'$, gdzie $A \subseteq \mathcal{A}(D)$.

Twierdzenie 4.1. *Niech D będzie acyklicznym digrafem bez A–skrótów, M liczbą naturalną $1 \leqslant M \leqslant |\mathcal{V}(D)|$, gdzie $A \subseteq \mathcal{A}(D)$. Wówczas istnieje acykliczna $\mathcal{H}_A^{(*)}$–partycja π digrafu D o liczebności co najwyżej M wtedy i tylko wtedy, gdy istnieje*

acykliczna $\mathcal{H}_A^{(*)}$–partycja π digrafu D spełniająca zależność:

$$\sum_{\substack{P_1,P_2\in\pi\\P_1\neq P_2}} |P_1 \underset{A}{\curvearrowright} P_2| \leqslant |A| - |\mathcal{V}(D)| + M. \tag{4.2}$$

Dowód. Niech D, A, M spełniają założenia twierdzenia. Ustalmy dowolną acykliczną $\mathcal{H}_A^{(*)}$–partycję π digrafu D. Zauważmy, że $|P \underset{A}{\curvearrowright} P| = |P| - 1$ w każdym digrafie nie zawierającym A–skrótów dla dowolnego P będącego elementem acyklicznej $\mathcal{H}_A^{(*)}$–partycji. Stąd

$$|A| = \sum_{\substack{P_1,P_2\in\pi\\P_1\neq P_2}} |P_1 \underset{A}{\curvearrowright} P_2| + \sum_{P\in\pi} (|P| - 1) = \sum_{\neq P_1,P_2\in\pi} |P_1 \underset{A}{\curvearrowright} P_2| + |\mathcal{V}(D)| - |\pi|, \tag{4.3}$$

a zatem $\sum_{\substack{P_1,P_2\in\pi\\P_1\neq P_2}} |P_1 \underset{A}{\curvearrowright} P_2| = |A| - |\mathcal{V}(D)| + |\pi|$ co ostatecznie kończy dowód. □

W szczególności z Tw. 4.1 wynika fakt, że jeśli problem $\mathcal{K}.1'$ jest NP–zupełny w rodzinie acyklicznych digrafów nie zawierających skrótów w wyróżnionych rodzinach łuków, to tym samym uzyskamy, że problem $\mathcal{K}.2'$ jest również problemem NP–zupełnym. Dodatkowo, jak zostało wykazane w Tw. 2.18, rozważając acykliczny digraf D z wyróżnionym zbiorem $\mathcal{A}(D)$–łuków A, możemy bez straty ogólności zakładać równość $A = \mathcal{A}(D)$. Stąd, w szczególności w celu uzasadnienia NP–zupełności problemów $\mathcal{K}.1'$, $\mathcal{K}.2'$, wystarczy pokazać, że NP–zupełny jest problem *Acyklicznej Partycji Hamiltona*.

Acykliczna Partycja Hamiltona (APH)
INSTANCJA: DAG D bez $\mathcal{A}(D)$–skrótów, liczba naturalna $0 \leqslant K \leqslant |\mathcal{V}(D)|$.
PYTANIE: Czy istnieje acykliczna $\mathcal{H}_{\mathcal{A}(D)}^{(*)}$–partycja digrafu D o mocy nie przekraczającej K.

Warto w tym miejscu zauważyć, że problem odnajdywania minimalnej liczby wierzchołkowo–rozłącznych skierowanych ścieżek, stanowiących partycję acyklicznego digrafu, bez ograniczenia w postaci acykliczności uzyskiwanej partycji, jest znanym problemem grafowym rozwiązywalnym w czasie wielomianowym [10].

4.1.1 Redukcja FAS $\propto$ APH

Jak łatwo można zauważyć, rozważany problem APH należy do klasy NP problemów. W bieżącej sekcji skonstruujemy redukcję wielomianową, transformującą szczególny podproblem problemu *Minimalnego Zbioru Sprzężonego* (ang. *Minimum Feedback Arc Set*, zob. GT8 [25, 28, 44]), który zachowuje NP–zupełność do APH.

Wprowadźmy niezbędne pojęcia w celu sformułowania dwóch znanych problemów grafowych.

Definicja 4.2. *Niech G będzie grafem (nieskierowanym). Zbiór wierzchołków $V \subseteq \mathcal{V}(G)$ będziemy nazywać* pokryciem wierzchołkowym *wtedy i tylko wtedy, gdy zbiór $\{u, v\} \cap V$ jest niepusty dla każdej $\mathcal{E}(G)$–krawędzi $\{u, v\}$.*

Definicja 4.3. *Niech D będzie digrafem. Zbiór $A \subseteq \mathcal{A}(D)$ będziemy nazywać* sprzężonym *wtedy i tylko wtedy, gdy do A należy co najmniej jeden $\mathcal{A}(D)$–łuk z dowolnego skierowanego $\mathcal{A}(D)$–cyklu.*

Pokrycie Wierzchołkowe (ang. *Vertex Cover*, **VC**, GT1 [21, 25, 26, 27])
INSTANCJA: Graf nieskierowany G, liczba naturalna $0 \leqslant K \leqslant |\mathcal{V}(G)|$.
PYTANIE: Czy istnieje pokrycie wierzchołkowe $V \subseteq \mathcal{V}(G)$ o mocy nie przekraczającej K.

Minimalny Zbiór Sprzężony (**FAS**)
INSTANCJA: Digraf D, liczba naturalna $0 \leqslant K \leqslant |\mathcal{A}(D)|$.
PYTANIE: Czy istnieje sprzężony podzbiór $\mathcal{A}(D)$–łuków o mocy co najwyżej K.

Analizując redukcję VC $\propto$ FAS zaproponowaną przez R. M. Karpa w pracy [45], możemy zaobserwować, że problem FAS jest NP–zupełny jeśli nawet założymy, że każdy wierzchołek v w rozpatrywanym digrafie D spełnia co najmniej jeden z dwóch warunków: $|\mathcal{N}_D^-(v)| = 1$ lub $|\mathcal{N}_D^+(v)| = 1$.

Obserwacja 4.4. *Problem FAS zachowuje NP–zupełność w rodzinie digrafów, spełniających zależność:* $|\mathcal{N}_D^-(v)| = 1$ *lub* $|\mathcal{N}_D^+(v)| = 1$ *dla każdego* $v \in \mathcal{V}(D)$.

Dowód. Niech G będzie grafem nieskierowanym, K liczbą naturalną spełniającą zależność $1 \leqslant K \leqslant |\mathcal{E}(G)|$, zaś V pokryciem wierzchołkowym G o mocy nie przekraczającej K. Powtarzając konstrukcję zaproponowaną przez R. M. Karpa, rozważmy digraf $\mathfrak{D}(G)$ określony równościami:

$$\begin{aligned} \mathcal{V}(\mathfrak{D}(G)) &= \mathcal{V}(G) \times \{0,1\}, \\ \mathcal{A}(\mathfrak{D}(G)) &= \{(\langle v,0\rangle, \langle v,1\rangle) : v \in \mathcal{V}(G)\} \cup \{(\langle u,1\rangle, \langle v,0\rangle) : \{u,v\} \in \mathcal{E}(G)\} \end{aligned} \tag{4.4}$$

oraz zbiór $\mathcal{A}(\mathfrak{D}(G))$–łuków $\mathfrak{D}(V) := \{(\langle v,0\rangle, \langle v,1\rangle) : v \in V\}$. Naturalnie, digraf $\mathfrak{D}(G)$ spełnia warunek sformułowany w obserwacji. W celu zakończenia uzasadniania, powtarzając rozumowanie za R. M. Karpem stwierdzamy, że $\mathfrak{D}(V)$ jest sprzężonym zbiorem $\mathcal{A}(\mathfrak{D}(G))$–łuków w digrafie $\mathfrak{D}(G)$ o mocy $|\mathfrak{D}(V)| \leqslant |V|$. Ustalmy następnie dowolny sprzężony zbiór $\mathcal{A}(\mathfrak{D}(G))$–łuków $\mathcal{F}$ w digrafie $\mathfrak{D}(G)$ o mocy co najwyżej K. Wówczas, zastępując każdy $\mathcal{F}$–łuk postaci $(\langle u,1\rangle, \langle v,0\rangle)$, przez $\mathcal{A}(\mathfrak{D}(G))$–łuk $(\langle v,0\rangle, \langle v,1\rangle)$, uzyskujemy sprzężony zbiór $\mathcal{A}(\mathfrak{D}(G))$–łuków $\mathcal{F}'$, dla którego $|\mathcal{F}'| \leqslant |\mathcal{F}|$, taki iż zbiór $\{v : (\langle v,0\rangle, \langle v,1\rangle) \in \mathcal{F}'\}$ jest pokryciem wierzchołkowym grafu G o mocy $|\mathcal{F}'|$. □

Zdefiniowany w Obs. 4.4 szczególny przypadek podproblemu FAS pełni ważną rolę w uzasadnieniu NP–zupełności problemu APH. Skonstruujemy bowiem redukcję z tego podproblemu FAS do APH, wykorzystując technikę gadżetów. Ustalmy dowolny digraf $\mathcal{G}$ spełniający zależność: $|\mathcal{N}_G^-(v)| = 1$ lub $|\mathcal{N}_G^+(v)| = 1$ dla każdego $v \in \mathcal{V}(\mathcal{G})$; oraz różnowartościową funkcję $e : \mathcal{A}(\mathcal{G}) \longrightarrow \{1, 2, \ldots, |\mathcal{A}(\mathcal{G})|\}$ numerującą zbiór łuków $\mathcal{A}(\mathcal{G})$. Przy wykorzystaniu struktury digrafu $\mathcal{G}$ zostanie skonstruowany acykliczny digraf $\mathbb{G}_{\mathcal{G},e}$ nie zawierający $\mathcal{A}(\mathbb{G}_{\mathcal{G},e})$–skrótów, który dodatkowo będzie posiadał zbiór wierzchołków pogrupowany na dwie rodziny: „gadżety"– zbiory wierzchołków w $\mathbb{G}_{\mathcal{G},e}$ odpowiadające pojedynczym wierzchołkom w $\mathcal{G}$ oraz „mosty" – pojedyncze wierzchołki w $\mathbb{G}_{\mathcal{G},e}$ odpowiadające pojedynczym łukom w $\mathcal{G}$. Konstrukcja tego digrafu oraz jego własności zostaną przedstawione w trzech kolejnych sekcjach. W sekcji 4.1.2 zdefiniowana zostanie struktura „gadżetu" (Def. 4.5) oraz udowodnione zostaną podstawowe jej własności. Następnie, w sekcji 4.1.3 dokończona zostanie konstrukcja digrafu $\mathbb{G}_{\mathcal{G},e}$ oraz skonstruowana zostanie acykliczna $\mathcal{H}^{(k)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycja $\pi_{\mathcal{F}}$ digrafu $\mathbb{G}_{\mathcal{G},e}$, na podstawie sprzężonego zbioru $\mathcal{A}(\mathcal{G})$–łuków $\mathcal{F}$ (Tw. 4.11), gdzie $k = |\mathcal{V}(\mathcal{G})| \cdot (|\mathcal{A}(\mathcal{G})| + 1) + |\mathcal{F}|$. W ostatniej sekcji zostanie skonstruowany sprzężony zbiór $\mathcal{A}(\mathcal{G})$–łuków $\mathcal{F}$ o mocy co najwyżej M, w oparciu o acykliczną $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycję π digrafu $\mathbb{G}_{\mathcal{G},e}$ (Tw. 4.16), gdzie $|\pi| \leqslant k$, $0 \leqslant M \leqslant |\mathcal{A}(\mathcal{G})|$.

4.1.2 Gadżety

Definicja 4.5. *Niech r będzie wierzchołkiem w digrafie $\mathcal{G}$.* Gadżetem *związanym z wierzchołkiem r będziemy nazywać digraf $\mathcal{N}_r$ (zob. rys. 4.1), zdefiniowany równościami:*

$$\begin{aligned}\mathcal{V}(\mathcal{N}_r) &= \{r_{i,j} : 0 \leqslant i,j \leqslant |\mathcal{A}(\mathcal{G})|\},\\ \mathcal{A}(\mathcal{N}_r) &= \{(r_{i,j}, r_{i,j+1}), (r_{i,j}, r_{i+1,j}) : 0 \leqslant i,j < |\mathcal{A}(\mathcal{G})|\} \cup\\ &\qquad \{(r_{n,i}, r_{n,i+1}), (r_{i,n}, r_{i+1,n}) : 0 \leqslant i < |\mathcal{A}(\mathcal{G})|\}.\end{aligned} \tag{4.5}$$

Wykorzystując Def. 4.5 uzyskujemy, że dla każdego $\mathcal{A}(\mathcal{N}_r)$–łuku $(r_{i_1,i_2}, r_{j_1,j_2})$, zachodzi tożsamość $i_1{+}i_2{+}1 = j_1{+}j_2$. Stąd łatwo można wykazać, że $\mathcal{N}_r$ jest acyklicznym digrafem niezawierającym $\mathcal{A}(\mathcal{N}_r)$–skrótów.

Wyróżnijmy w gadżecie $\mathcal{N}_r$ dwie rodziny łuków:

$$\begin{aligned}\swarrow_r &:= \{(r_{i,j}, r_{i,j+1}) : 0 \leqslant i \leqslant n \wedge 0 \leqslant j < |\mathcal{A}(\mathcal{G})|\},\\ \searrow_r &:= \{(r_{i,j}, r_{i+1,j}) : 0 \leqslant i < n \wedge 0 \leqslant j \leqslant |\mathcal{A}(\mathcal{G})|\},\end{aligned} \tag{4.6}$$

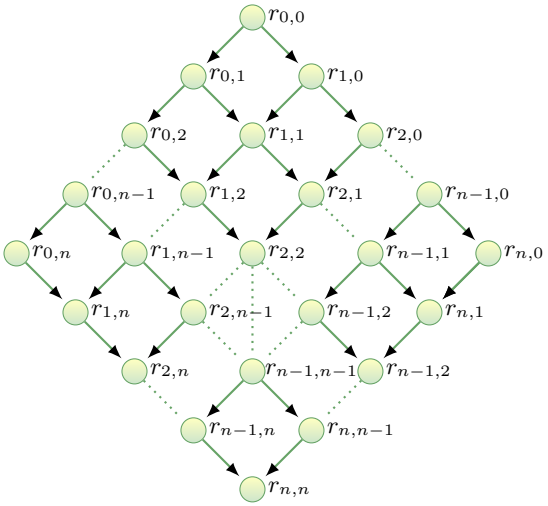
dwa rodzaje skierowanych $\mathcal{A}(\mathcal{N}_r)$–dróg:

$$\begin{aligned}\swarrow_r^i &:= r_{i,0} \underset{\mathcal{A}(\mathcal{N}_r)}{\rightarrow} r_{i,1} \underset{\mathcal{A}(\mathcal{N}_r)}{\rightarrow} \cdots \underset{\mathcal{A}(\mathcal{N}_r)}{\rightarrow} r_{i,n},\\ \searrow_r^i &:= r_{0,i} \underset{\mathcal{A}(\mathcal{N}_r)}{\rightarrow} r_{1,i} \underset{\mathcal{A}(\mathcal{N}_r)}{\rightarrow} \cdots \underset{\mathcal{A}(\mathcal{N}_r)}{\rightarrow} r_{n,i},\end{aligned} \tag{4.7}$$

gdzie $0 \leqslant i \leqslant |\mathcal{A}(\mathcal{G})|$; oraz dwie $\mathcal{H}^{(*)}_{\mathcal{A}(\mathcal{N}_r)}$–partycje gadżetu $\mathcal{N}_r$:

$$\mathcal{L}_r := \{\mathcal{V}(\swarrow_r^i) : 0 \leqslant i \leqslant |\mathcal{A}(\mathcal{G})|\}, \qquad \mathcal{R}_r := \{\mathcal{V}(\searrow_r^i) : 0 \leqslant i \leqslant |\mathcal{A}(\mathcal{G})|\}. \tag{4.8}$$

Będziemy mówić, że $\mathcal{A}(\mathcal{N}_r)$–droga s jest $\swarrow_r^*$–ukosem, jeśli $s = \swarrow_r^j$ dla pewnej liczby $0 \leqslant j \leqslant |\mathcal{A}(\mathcal{G})|$. Analogicznie, będziemy mówić, że $\mathcal{A}(\mathcal{N}_r)$–droga s jest $\searrow_r^*$–ukosem, jeśli $s = \searrow_r^j$ dla pewnej liczby $0 \leqslant j \leqslant |\mathcal{A}(\mathcal{G})|$.



Rysunek 4.1: Gadżet $\mathcal{N}_r$.

Lemat 4.6. *Niech π będzie acykliczną $\mathcal{H}^{(*)}_{\mathcal{A}(\mathcal{N}_r)}$–partycją gadżetu $\mathcal{N}_r$. Wówczas dla każdego $P \in \pi$, $\mathfrak{h}^\pi(P)$ jest skierowaną $\swarrow_r$ lub $\searrow_r$–drogą.*

Dowód. Przypuśćmy nie wprost, że $\mathcal{A}(\mathcal{N}_r)$–droga $\mathfrak{h}^\pi(P)$ zawiera co najmniej jeden $\swarrow_r$–łuk oraz co najmniej jeden $\searrow_r$–łuk. Bez straty ogólności możemy przyjąć, że $r_{i,j} \underset{\swarrow_r}{\rightarrow} r_{i,j+1} \underset{\searrow_r}{\rightarrow} r_{i+1,j+1}$ jest segmentem $\mathfrak{h}^\pi(P)$ dla pewnych $0 \leqslant i,j < |\mathcal{A}(\mathcal{G})|$ (dla drugiej możliwości, kiedy $r_{i,j} \underset{\searrow_r}{\rightarrow} r_{i,j+1} \underset{\swarrow_r}{\rightarrow} r_{i+1,j+1}$ dowód jest analogiczny). Zauważmy najpierw, że dla każdych dwóch kolejnych wierzchołków r_{i_1,j_1}, r_{i_2,j_2} z P, zachodzi $i_1+j_1 < i_2+j_2$. Wówczas $i+1+j < i+1+j+1 < k+l$ dla każdego wierzchołka $r_{k,l} \in P$ występującego po $r_{i+1,j+1}$ na drodze $\mathfrak{h}^\pi(P)$ oraz $i+1+j > i+j > k+l$ dla każdego $r_{k,l} \in P$ występującego przed $r_{i,j}$ skąd $\mathfrak{h}^\pi(P) \neq \mathfrak{h}^\pi(r_{i+1,j})$. Następnie z zależności $P \ni r_{i,j} \underset{\searrow_r}{\rightarrow} r_{i+1,j} \in \mathfrak{h}^\pi(r_{i+1,j})$, $\mathfrak{h}^\pi(r_{i+1,j}) \ni r_{i+1,j} \underset{\swarrow_r}{\rightarrow} r_{i+1,j+1} \in P$ stwierdzamy, że $(P, \mathcal{V}(\mathfrak{h}^\pi(r_{i+1,j}))), (\mathcal{V}(\mathfrak{h}^\pi(r_{i+1,j})), P)$ są $\mathcal{A}(\mathcal{G}(\mathcal{N}_r,\pi))$–łukami, które generują $\mathcal{A}(\mathcal{G}(\mathcal{N}_r,\pi))$–cykl w acyklicznym digrafie $\mathcal{G}(\mathcal{N}_r,\pi)$. Uzyskana sprzeczność kończy dowód. □

Własność acyklicznej $\mathcal{H}^{(*)}_{\mathcal{A}(\mathcal{N}_r)}$–partycji π gadżetu $\mathcal{N}_p$ sformułowana w Lem. 4.6 determinuje więc postać każdej skierowanej drogi Hamiltona $\mathfrak{h}^\pi(P)$, gdzie $P \in \pi$. W poniższych dwóch lematach pokażemy, że jeśli moc partycji π ma wartość minimalną, to każda droga $\mathfrak{h}^\pi(P)$ jest $\swarrow_r^*$ lub $\searrow_r^*$ ukosem.

Lemat 4.7. *Niech π będzie acykliczną $\mathcal{H}^{(*)}_{\mathcal{A}(\mathcal{N}_r)}$–partycją gadżetu $\mathcal{N}_r$. Wówczas dla każdej liczby $i = 0,1,\ldots,|\mathcal{A}(\mathcal{G})|$ istnieje zbiór $L_i \in \pi$, dla którego $\mathfrak{h}^\pi(L_i)$ jest segmentem $\swarrow_r^i$ drogi, lub dla każdej liczby $i = 0,1,\ldots,|\mathcal{A}(\mathcal{G})|$ istnieje zbiór $R_i \in \pi$ dla którego $\mathfrak{h}^\pi(R_i)$ jest segmentem $\searrow_r^i$ drogi.*

Dowód. Przypuśćmy, że istnieje liczba $0 \leqslant i \leqslant |\mathcal{A}(\mathcal{G})|$, dla której $\mathfrak{h}^\pi(L)$ nie jest segmentem $\swarrow_r^i$ dla dowolnego $L \in \pi$. Wówczas wykorzystując Lem. 4.6 uzyskujemy, że drogi $\mathfrak{h}^\pi(r_{i,0}), \mathfrak{h}^\pi(r_{i,1}), \ldots, \mathfrak{h}^\pi(r_{i,|\mathcal{A}(\mathcal{G})|})$ są parami wierzchołkowo rozłączne oraz każda droga $\mathfrak{h}^\pi(r_{i,j})$ jest segmentem $\searrow_r^j$, gdzie $0 \leqslant j \leqslant |\mathcal{A}(\mathcal{G})|$, co ostatecznie kończy dowód. □

Wniosek 4.8. *Niech π będzie acykliczną $\mathcal{H}^{(*)}_{\mathcal{A}(\mathcal{N}_r)}$–partycją gadżetu $\mathcal{N}_r$, wówczas $|\pi| \geqslant |\mathcal{A}(\mathcal{G})| + 1$.*

Lemat 4.9. *$\mathcal{L}_r$ oraz $\mathcal{R}_r$ są jedynymi acyklicznymi $\mathcal{H}^{(*)}_{\mathcal{A}(\mathcal{N}_r)}$–partycjami gadżetu $\mathcal{N}_r$ o minimalnej liczebności, różnymi jeśli dodatkowo $|\mathcal{A}(\mathcal{G})| > 0$.*

Dowód. Wykorzystując Wn. 4.8, uzyskujemy natychmiast, że partycje $\mathcal{L}_r$, $\mathcal{R}_r$ posiadają minimalną liczebność spośród wszystkich acyklicznych $\mathcal{H}^{(*)}_{\mathcal{A}(\mathcal{N}_r)}$–partycji gadżetu $\mathcal{N}_r$. Załóżmy, że istnieje acykliczna $\mathcal{H}^{(|\mathcal{A}(\mathcal{G})|+1)}_{\mathcal{A}(\mathcal{N}_r)}$–partycja π' gadżetu $\mathcal{N}_r$, różna od $\mathcal{L}_r$ i $\mathcal{R}_r$. Korzystając wówczas z Lem. 4.7, przyjmijmy bez straty ogólności, że dla każdego $0 \leqslant i \leqslant |\mathcal{A}(\mathcal{G})|$ istnieje droga Hamiltona $\mathfrak{h}^{\pi'}(L_i)$ będąca w całości segmentem $\swarrow_r^i$, gdzie $L_i \in \pi'$ (dowód przypadku, w którym dla każdego $0 \leqslant i \leqslant |\mathcal{A}(\mathcal{G})|$ istnieje droga Hamiltona $\mathfrak{h}^{\pi'}(R_i)$ będąca w całości segmentem $\swarrow_r^i$ jest analogiczny). Wówczas $\swarrow_r$–drogi $\mathfrak{h}^{\pi'}(L_0), \mathfrak{h}^{\pi'}(L_1), \ldots, \mathfrak{h}^{\pi'}(L_{|\mathcal{A}(\mathcal{G})|})$ są wierzchołkowo rozłączne na mocy Lem. 4.6. Dodatkowo $|\mathcal{A}(\mathcal{G})| + 1 = |\{L_i : 0 \leqslant i \leqslant |\mathcal{A}(\mathcal{G})|\}| \leqslant |\pi| = |\mathcal{A}(\mathcal{G})| + 1$, skąd $\bigcup_{0 \leqslant i \leqslant |\mathcal{A}(\mathcal{G})|} L_i = \mathcal{V}(\mathcal{N}_r)$, odkąd oraz $\pi' = \mathcal{L}_p$ wbrew założeniu, że $\pi' \neq \mathcal{L}_r, \mathcal{R}_r$. Uzyskana sprzeczność kończy dowód. □

4.1.3 DAG $\mathbb{G}_{\mathcal{G},e}$

W konstrukcji digrafu $\mathbb{G}_{\mathcal{G},e}$ będziemy wykorzystywać ustaloną różnowartościową funkcję e, aczkolwiek jej wybór nie będzie miał wpływu na rozważane własności tego digrafu.

Definicja 4.10. *Niech $e : \mathcal{A}(\mathcal{G}) \longrightarrow \{1, 2, \ldots, |\mathcal{A}(\mathcal{G})|\}$ będzie równowartościową funkcją numerującą zbiór łuków $\mathcal{A}(\mathcal{G})$.* Digrafem $\mathbb{G}_{\mathcal{G},e}$ *wyznaczonym przez digraf $\mathcal{G}$ oraz funkcję numerującą e będziemy nazywać strukturę zdefiniowaną równościami:*

$$\begin{aligned}\mathcal{V}(\mathbb{G}_{\mathcal{G},e}) &= \bigcup_{v\in\mathcal{V}(\mathcal{G})} \mathcal{V}(\mathcal{N}_v) \cup \mathcal{A}(\mathcal{G}),\\ \mathcal{A}(\mathbb{G}_{\mathcal{G},e}) &= \bigcup_{v\in\mathcal{V}(\mathcal{G})} \mathcal{A}(\mathcal{N}_v) \cup \{(vu, v_{e(vu),0}) : vu \in \mathcal{A}(\mathcal{G})\} \cup \\ &\qquad\qquad \{(vu, u_{0,e(vu)}) : vu \in \mathcal{A}(\mathcal{G})\}.\end{aligned} \tag{4.9}$$

Zatem zbiór wierzchołków składa się tutaj ze zbioru wierzchołków gadżetów $\mathcal{N}_v$ dla wszystkich $v \in \mathcal{V}(\mathcal{G})$ oraz z łuków digrafu $\mathcal{G}$. Z kolei zbiór łuków składa się tutaj ze zbioru łuków gadżetów $\mathcal{N}_v$, a także z łuków łączących łuki $\mathcal{G}$ z gadżetami $\mathcal{N}_v$. Definicja ta zilustrowana jest przykładem na rys. 4.2.

Bezpośrednio z konstrukcji digrafu $\mathbb{G}_{\mathcal{G},e}$ uzyskujemy, że jest to struktura acykliczna spełniająca tożsamość $\mathbb{G}_{\mathcal{G},e|\mathcal{V}(\mathcal{N}_v)} = \mathcal{N}_v$ dla każdego $v \in \mathcal{V}(\mathcal{G})$.

W poniższym twierdzeniu skonstruujemy acykliczną $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycję $\pi_{\mathcal{F}}$ digrafu $\mathbb{G}_{\mathcal{G},e}$ dla dowolnego ustalonego zbioru sprzężonego $\mathcal{F}$ w $\mathcal{G}$, dla których będzie spełniona równość $|\pi_{\mathcal{F}}| = |\mathcal{V}(\mathcal{G})|\cdot(|\mathcal{A}(\mathcal{G})|+1)+|\mathcal{F}|$ (zob. rys. 4.2). Podkreślmy w tym miejscu, że poniższa konstrukcja nie wykorzystuje ograniczenia na moce zbiorów $\mathcal{N}^-_{\mathcal{G}}(v)$, $\mathcal{N}^+_{\mathcal{G}}(v)$ w digrafie $\mathcal{G}$, gdzie $v \in \mathcal{V}(\mathcal{G})$, które to ograniczenie jest wykorzystywane jedynie w uzasadnieniu Tw. 4.15.

Twierdzenie 4.11. *Niech $\mathcal{F}$ będzie zbiorem sprzężonym $\mathcal{A}(\mathcal{G})$–łuków w digrafie $\mathcal{G}$. Wówczas istnieje acykliczna $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycja digrafu $\mathbb{G}_{\mathcal{G},e}$ o liczebności $|\mathcal{V}(\mathcal{G})| \cdot (|\mathcal{A}(\mathcal{G})| + 1) + |\mathcal{F}|$.*

Dowód. Obierzmy partycję zbioru wierzchołków $\mathcal{V}(\mathbb{G}_{\mathcal{G},e})$ wyznaczoną przez sprzężony zbiór $\mathcal{A}(\mathcal{G})$–łuków $\mathcal{F}$, zdefiniowany równością:

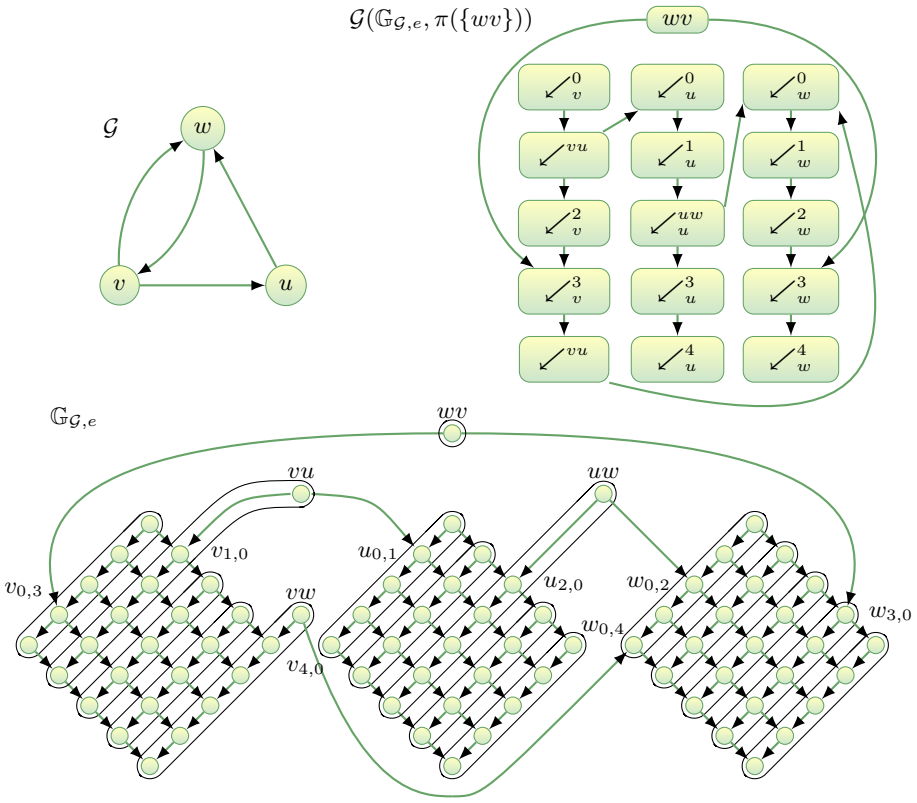
$$\begin{aligned}\pi(\mathcal{F}) = \{&\{e^{-1}(i), v_{i,0}, v_{i,1}, \ldots, v_{i,|\mathcal{A}(\mathcal{G})|}\} : v \in \mathcal{V}(\mathcal{G}) \wedge 0 \leqslant i \leqslant |\mathcal{A}(\mathcal{G})| \wedge \\ &\exists_{u\in\mathcal{V}(\mathcal{G})} (e^{-1}(i) = vu \wedge vu \in \mathcal{A}(\mathcal{G}) \setminus \mathcal{F})\} \cup \\ &\{\{v_{i,0}, v_{i,1}, \ldots, v_{i,|\mathcal{A}(\mathcal{G})|}\} : v \in \mathcal{V}(\mathcal{G}) \wedge 0 \leqslant i \leqslant |\mathcal{A}(\mathcal{G})| \wedge \\ &\neg \exists_{u\in\mathcal{V}(\mathcal{G})} (e^{-1}(i) = vu \wedge vu \in \mathcal{A}(\mathcal{G}) \setminus \mathcal{F})\} \cup \\ &\{\{vu\} : vu \in \mathcal{F}\}.\end{aligned} \tag{4.10}$$

Elementy partycji $\pi(\mathcal{F})$ odpowiadają wówczas trzem rodzajom skierowanych $\mathcal{A}(\mathbb{G}_{\mathcal{G},e})$–dróg w $\mathbb{G}_{\mathcal{G},e}$:

1. $vu \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\rightarrow} \underbrace{v_{e(vu),0} \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\rightarrow} v_{e(vu),1} \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\rightarrow} \cdots \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\rightarrow} v_{e(vu),|\mathcal{A}(\mathcal{G})|}}_{\swarrow_v^{e(vu)}}$, jeśli $vu \in \mathcal{A}(\mathcal{G})\setminus\mathcal{F}$, oznaczanych dalej symbolem $\swarrow_{vu}$,
2. $\swarrow_v^i$, jeśli $\neg \exists_{u\in\mathcal{V}(\mathcal{G})} (e^{-1}(i) = vu \wedge vu \in \mathcal{A}(\mathcal{G}) \setminus \mathcal{F})$,
3. vu, jeśli $vu \in \mathcal{F}$.

Jak łatwo można sprawdzić, $\pi(\mathcal{F})$ jest $\mathcal{H}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycją digrafu $\mathbb{G}_{\mathcal{G},e}$ o mocy $|\mathcal{V}(\mathcal{G})| \cdot (|\mathcal{A}(\mathcal{G})| + 1) + |\mathcal{F}|$. W celu zakończenia dowodu pokażemy jedynie, że digraf $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi(\mathcal{F}))$ jest acykliczny.

Załóżmy nie wprost, że $\mathfrak{a} := a_1 \to a_2 \to \ldots \to a_k \to a_1$ jest $\mathcal{A}(\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi(\mathcal{F})))$–cyklem, gdzie $k > 1$. Ponieważ każdy wierzchołek postaci vu jest źródłem w $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi(\mathcal{F}))$, to cykl $\mathfrak{a}$ jest zbudowany wyłącznie z wierzchołków odpowiadających $\mathcal{A}(\mathbb{G}_{\mathcal{G},e})$–drogom postaci *1*, *2*. Stąd, dla każdego wierzchołka a_i istnieje dokładnie jeden wierzchołek $v_i \in \mathcal{V}(\mathcal{G})$ spełniający zależność $a_i \cap \mathcal{V}(\mathcal{N}_{v_i}) \neq \emptyset$, gdzie $1 \leqslant i \leqslant k$. Naturalnie, wierzchołki występujące w ciągu $\mathfrak{v} := \langle v_1, v_2, \ldots, v_k \rangle$ nie muszą być parami różne, aczkolwiek uzasadnimy, że zastępując występujące bezpośrednio po sobie powtórzenia tego samego wierzchołka jego jednym wystąpieniem, uzyskamy $\mathcal{A}(\mathcal{G})\backslash\mathcal{F}$–cykl, którego obecność będzie sprzeczna z własnością sprzężoności zbioru $\mathcal{F}$ (zakładamy dodatkowo, że v_1 występuje bezpośrednio po v_k). Ustalmy w tym celu dowolne



Rysunek 4.2: Acykliczna $\mathcal{H}^{(16)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycja digrafu $\mathbb{G}_{\mathcal{G},e}$, ilustrująca konstrukcję zawartą w dowodzie Tw. 4.11, gdzie $\mathcal{F} = \{wv\}$, $e(vu) = 1$, $e(uw) = 2$, $e(wv) = 3$, $e(vw) = 4$.

dwa wierzchołki $a_i, a_{i+1} \in \mathcal{V}(\mathfrak{a})$, dla których $v_i \neq v_{i+1}$. Oba wierzchołki odpowiadają $\mathcal{A}(\mathbb{G}_{\mathcal{G},e})$–drogom postaci *1*, *2*, co w konkluzji z warunkiem $a_i \underset{\mathcal{A}(\mathcal{G}(\mathbb{G}_{\mathcal{G},e},\pi(\mathcal{F})))}{\rightarrow} a_{i+1}$ implikuje dwie równości: $\mathfrak{h}^{\pi(\mathcal{F})}(a_i) = \swarrow^{v_i v_{i+1}}$, $\mathfrak{h}^{\pi(\mathcal{F})}(a_{i+1}) = \swarrow^{0}_{v_{i+1}}$. Stąd w szczególności $v_i v_{i+1}$ jest $\mathcal{A}(\mathcal{G}) \setminus \mathcal{F}$–łukiem, co ostatecznie kończy dowód. □

Wykorzystując fakt, że zbiór wszystkich $\mathcal{A}(\mathcal{G})$–łuków w digrafie $\mathcal{G}$ jest zbiorem sprzężonym, w konkluzji z Tw. 4.11 uzyskujemy następujące stwierdzenie.

Stwierdzenie 4.12. *Istnieje acykliczna $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycja digrafu $\mathbb{G}_{\mathcal{G},e}$ o liczebności $|\mathcal{V}(\mathcal{G})| \cdot (|\mathcal{A}(\mathcal{G})| + 1) + |\mathcal{A}(\mathcal{G})|$.*

4.1.4 Zbiór sprzężony w digrafie $\mathcal{G}$ wyznaczony przez acykliczną $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycję digrafu $\mathbb{G}_{\mathcal{G},e}$

Ustalmy dowolną acykliczną $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycję π digrafu $\mathbb{G}_{\mathcal{G},e}$. W sekcji tej przedstawimy konstrukcję zbioru sprzężonego, rozbijając ją na dwa etapy. W pierwszym kroku (Tw. 4.14) skonstruujemy acykliczną $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycję π', dla której $|\pi'| \leqslant |\pi|$ oraz partycja π' obcięta do zbioru wierzchołków gadżetu $\mathcal{N}_v$ jest tożsamościowo równa $\mathcal{L}_v$ lub $\mathcal{R}_v$, gdzie $v \in \mathcal{V}(\mathcal{G})$. Posiadając partycję π', w której każdy gadżet odpowiada $\mathcal{L}_v$ lub $\mathcal{R}_v$–partycji, w Tw. 4.15 skonstruujemy sprzężony zbiór $\mathcal{A}(\mathcal{G})$–łuków w $\mathcal{G}$ o mocy ograniczonej przez liczbę $|\pi'| - |\mathcal{V}(\mathcal{G})| \cdot (|\mathcal{A}(\mathcal{G})| + 1) \leqslant |\pi| - |\mathcal{V}(\mathcal{G})| \cdot (|\mathcal{A}(\mathcal{G})| + 1)$.

Ustalmy wierzchołek $v \in \mathcal{V}(\mathcal{G})$. W poniższych rozważaniach będziemy wykorzystywać następujące notacje:

$$\pi_{|v} := \{\mathcal{V}(\mathfrak{h}^{\pi}(w)) : w \in \mathcal{V}(\mathcal{N}_v)\}, \qquad \mathbb{G}_{\mathcal{G},e_{|\pi,v}} := \mathbb{G}_{\mathcal{G},e_{|\bigcup \pi_{|v}}}. \tag{4.11}$$

Definicja 4.13. *Niech π będzie acykliczną $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycją digrafu $\mathbb{G}_{\mathcal{G},e}$ oraz $v \in \mathcal{V}(\mathcal{G})$. Będziemy Mówić, że gadżet $\mathcal{N}_v$ jest* zorientowany *w π wtedy i tylko wtedy, gdy $\pi_{|\mathcal{V}(\mathcal{N}_v)} = \mathcal{L}_v$ lub $\pi_{|\mathcal{V}(\mathcal{N}_v)} = \mathcal{R}_v$. W przypadku $\pi_{|\mathcal{V}(\mathcal{N}_v)} = \mathcal{L}_v$, gadżet $\mathcal{N}_v$ będziemy nazywać* $\swarrow$–zorientowanym, *natomiast w przeciwnym przypadku* $\searrow$–zorientowanym.

Twierdzenie 4.14. *Niech π będzie acykliczną $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycją digrafu $\mathbb{G}_{\mathcal{G},e}$ w której gadżet $\mathcal{N}_{\mathfrak{u}}$ nie jest zorientowany, dla jakiegoś $\mathfrak{u} \in \mathcal{V}(\mathcal{G})$. Wówczas istnieje acykliczna $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycja π' digrafu $\mathbb{G}_{\mathcal{G},e}$, w której gadżet $\mathcal{N}_{\mathfrak{u}}$ jest zorientowany oraz $|\pi'| \leqslant |\pi|$. Dodatkowo każdy gadżet, który był zorientowany w π zachowuje swoją orientację w π'.*

Dowód. Niech π, $\mathfrak{u}$ spełniają założenia twierdzenia (wierzchołek $\mathfrak{u}$ zapisujemy tutaj alfabetem gotyckim dla zwiększenia czytelności wywodu). Zauważmy najpierw, że w digrafie $\mathbb{G}_{\mathcal{G},e_{|\pi,\mathfrak{u}}}$ możemy wyróżnić trzy rodzaje wierzchołków:

(*i*) zbiór $\mathcal{V}(\mathcal{N}_{\mathfrak{u}})$,

(*ii*) wierzchołki, które można przedstawić w postaci $\mathfrak{u}l$, gdzie $l \in \mathcal{N}^{+}_{\mathcal{G}}(\mathfrak{u})$,

(*iii*) wierzchołki, które można przedstawić w postaci $r\mathfrak{u}$, gdzie $r \in \mathcal{N}^{-}_{\mathcal{G}}(\mathfrak{u})$.

Wprowadźmy oznaczenia dla wierzchołków rodzaju (*ii*) oraz (*iii*): $L := \{\mathfrak{u}l \in \mathcal{A}(\mathcal{G}) : \mathfrak{u}l \in \mathcal{V}(\mathbb{G}_{\mathcal{G},e_{|\pi,\mathfrak{u}}})\}$, $R := \{r\mathfrak{u} \in \mathcal{A}(\mathcal{G}) : r\mathfrak{u} \in \mathcal{V}(\mathbb{G}_{\mathcal{G},e_{|\pi,\mathfrak{u}}})\}$. Zauważmy, że L i R są równocześnie zbiorami $\mathcal{G}$-łuków oraz wierzchołków w $\mathbb{G}_{\mathcal{G},e}$. Przyjmijmy, że $L = \{\mathfrak{u}l_1, \mathfrak{u}l_2, \ldots, \mathfrak{u}l_i\}$, $R = \{r_1\mathfrak{u}, r_2\mathfrak{u}, \ldots, r_j\mathfrak{u}\}$, gdzie $i = |L|$, $j = |R|$, $e(\mathfrak{u}l_1) < e(\mathfrak{u}l_2) <$

$\ldots < e(\mathfrak{u}l_i)$, $e(\mathfrak{u}r_1) < e(\mathfrak{u}r_2) < \ldots < e(\mathfrak{u}r_j)$. W związku z tym że, digraf $\mathcal{G}$ nie zawiera łuków postaci vv, mamy $\{l_1, l_2, \ldots, l_i\} \cap \{r_1, r_2, \ldots, r_i\} = \emptyset$. Wprowadźmy teraz oznaczenie $\mathfrak{h}^{\pi}_{|\mathfrak{u}}(v)$ na segment $\mathcal{A}(\mathbb{G}_{\mathcal{G},e})$–drogi $\mathfrak{h}^{\pi}(v)$, wyznaczony przez zbiór $\mathcal{V}(\mathfrak{h}^{\pi}(v)) \cap \mathcal{V}(\mathcal{N}_{\mathfrak{u}})$, gdzie v jest dowolnym wierzchołkiem w $\mathbb{G}_{\mathcal{G},e}$. Wiemy, że jeśli $\mathcal{V}(\mathfrak{h}^{\pi}(v)) \cap \mathcal{V}(\mathcal{N}_{\mathfrak{u}}) \neq \emptyset$, to zbiór $\mathcal{V}(\mathfrak{h}^{\pi}(v)) \setminus \mathcal{V}(\mathcal{N}_{\mathfrak{u}})$ zawiera co najwyżej wierzchołki ze zbioru $\mathcal{A}(\mathcal{G})$ będące źródłami w $\mathbb{G}_{\mathcal{G},e}$. Z tego zaś wynika, że segment $\mathfrak{h}^{\pi}_{|\mathfrak{u}}(v)$ jest $\mathcal{A}(\mathbb{G}_{\mathcal{G},e})$–drogą.

Krok 1. Pokażemy najpierw, że $|\pi_{|\mathfrak{u}}| \geqslant |\mathcal{A}(\mathcal{G})| + 1 + \min\{i, j\}$. Dowód przez przypadki:

(1°) Załóżmy, że dla każdej liczby naturalnej k spełniającej zależność $1 \leqslant k \leqslant |\mathcal{A}(\mathcal{G})|$, istnieje zbiór $P \in \pi_{|\mathfrak{u}}$, dla którego $\mathfrak{h}^{\pi}_{|\mathfrak{u}}(P)$ jest niepustym segmentem $\swarrow^{k}_{\mathfrak{u}}$. Stąd, wykorzystując Lem. 4.6, stwierdzamy, że partycja π zawiera co najmniej $|\mathcal{A}(\mathcal{G})|$ elementów, z których każdy jest wyznaczony jednoznacznie przez odpowiednią $\mathcal{A}(\mathcal{N}_{\mathfrak{u}})$–drogę będącą $\swarrow^{*}_{p}$–ukosem. Dodatkowo, $\mathcal{V}(\swarrow^{k}_{\mathfrak{u}}) \cap (R \cup \{\mathfrak{u}_{0,0}\}) = \emptyset$ dla każdego $1 \leqslant k \leqslant |\mathcal{A}(\mathcal{G})|$, a zatem każda z tych dróg jest wierzchołkowo rozłączna z $\mathfrak{h}^{\pi}(v)$ dla każdego $v \in R \cup \{\mathfrak{u}_{0,0}\}$. Ponadto dla każdych $v_1, v_2 \in R \cup \{\mathfrak{u}_{0,0}\}$, wierzchołki v_1, v_2 są źródłami w $\mathbb{G}_{\mathcal{G},e}$, skąd $\mathfrak{h}^{\pi}(v_1) \neq \mathfrak{h}^{\pi}(v_2)$, jeśli tylko $v_1 \neq v_2$. W związku z tym ostatecznie $|\pi| \geqslant |\mathcal{A}(\mathcal{G})| + |R \cup \{\mathfrak{u}_{0,0}\}| = |\mathcal{A}(\mathcal{G})| + j + 1 \geqslant |\mathcal{A}(\mathcal{G})| + 1 + \min\{i, j\}$.

(2°) Niech k będzie liczbą naturalną $1 \leqslant k \leqslant |\mathcal{A}(\mathcal{G})|$, dla której $\mathfrak{h}^{\pi}_{|\mathfrak{u}}(P)$ nie jest segmentem $\swarrow^{k}_{\mathfrak{u}}$, gdzie P jest dowolnym elementem zbioru $\pi_{|\mathfrak{u}}$. Wówczas, jak łatwo można wykazać korzystając z Lem. 4.6, droga $\mathfrak{h}^{\pi}(\mathfrak{u}_{k,i})$ jest niepustym segmentem $\searrow^{i}_{\mathfrak{u}}$, gdzie $1 \leqslant i \leqslant |E|$. Stąd, $\mathfrak{h}^{\pi}(\mathfrak{u}_{k,1}), \mathfrak{h}^{\pi}(\mathfrak{u}_{k,2}), \ldots, \mathfrak{h}^{\pi}(\mathfrak{u}_{k,|\mathcal{A}(\mathcal{G})|})$ są wierzchołkowo rozłączne (uwaga: droga $\mathfrak{h}^{\pi}(\mathfrak{u}_{k,0})$ jest wierzchołkowo rozłączna z wymienionymi $\mathcal{A}(\mathbb{G}_{\mathcal{G},e})$–drogami, ale może pokrywać się z $\mathfrak{h}^{\pi}(v)$, dla pewnego $v \in L \cup \{\mathfrak{u}_{0,0}\}$). Dodatkowo, $\mathfrak{h}^{\pi}(\mathfrak{u}_{k,d}) \neq \mathfrak{h}^{\pi}(v)$ dla $1 \leqslant d \leqslant |\mathcal{A}(\mathcal{G})|$, $v \in \{l_1, l_2, \ldots, l_i\} \cup \{\mathfrak{u}_{0,0}\}$ oraz z obserwacji, iż $\mathfrak{h}^{\pi}(v_1) \neq \mathfrak{h}^{\pi}(v_2)$ dla $\neq v_1, v_2 \in \{l_1, l_2, \ldots, l_i\} \cup \{\mathfrak{u}_{0,0}\}$ wnioskujemy, że $|\pi| \geqslant |\mathcal{A}(\mathcal{G})| + |\{l_1, l_2, \ldots, l_i\} \cup \{\mathfrak{u}_{0,0}\}| = |\mathcal{A}(\mathcal{G})| + i + 1 \geqslant |\mathcal{A}(\mathcal{G})| + 1 + \min\{i, j\}$.

Krok 2. Opiszemy teraz własności łuków w $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi)$, które łączą wierzchołki mające niepuste przekroje ze zbiorami wierzchołków różnych gadżetów. Ustalmy w tym celu dowolny łuk (P_1, P_2) w $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi)$, dla którego $P_1 \cap \mathcal{V}(\mathcal{N}_v) \neq \emptyset$, $P_2 \cap \mathcal{V}(\mathcal{N}_w) \neq \emptyset$, $v \neq w$, gdzie $v, w \in \mathcal{V}(\mathcal{G})$. Pokażemy, że są możliwe tylko dwa przypadki:

(*i*) vw, $v_{e(vw),0} \in P_1$, $w_{0,e(vw)} \in P_2$,

(*ii*) wv, $v_{0,e(wv)} \in P_1$, $w_{e(wv),0} \in P_2$

(uwaga, rys. 4.3 ilustruje poniższe rozumowanie nie wprost, w którym $v_{e(vw),0} \in P_2$ zamiast $v_{e(vw),0} \in P_2$ oraz $v_{0,e(wv)} \in P_2$ zamiast $v_{0,e(wv)} \in P_1$). Z założenia $(P_1, P_2) \in \mathcal{A}(\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi))$, więc $P_1 \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\curvearrowright} P_2 \neq \emptyset$. Dodatkowo korzystając z definicji Def. 4.9, uzyskujemy, że $\mathcal{V}(\mathcal{N}_v) \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\curvearrowright} \mathcal{V}(\mathcal{N}_w) = \emptyset$. Stąd początek lub koniec każdego łuku ze zbioru $P_1 \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\curvearrowright} P_2$ nie należy do $\mathcal{V}(\mathcal{N}_v) \cup \mathcal{V}(\mathcal{N}_w)$. Oczywiście wierzchołek taki jest częścią skierowanej drogi $\mathfrak{h}^{\pi}(P_1)$ lub $\mathfrak{h}^{\pi}(P_2)$. Stąd, jak łatwo można wykazać, wierzchołek ten ma jedną z czterech postaci: vx, wx, xv, xw. Dodatkowo każda z tych postaci odpowiada wierzchołkowi, który jest źródłem w $\mathbb{G}_{\mathcal{G},e}$, a zatem wierzchołek ten musi być połączony z co najmniej jednym wierzchołkiem z $\mathcal{V}(\mathcal{N}_v)$ oraz $\mathcal{V}(\mathcal{N}_w)$. Ostatecznie możliwe więc są dwa przypadki: $vw \in P_1 \cup P_2$, $wv \in P_1 \cup P_2$. Pokażemy,

że (i) zachodzi dla $vw \in P_1 \cup P_2$, a (ii) w drugim przypadku. Ze względu na podobieństwo prowadzonych rozumowań, uzasadnimy tylko pierwszy z nich. Załóżmy więc, że $vw \in P_1 \cup P_2$ oraz nie wprost, że $vw \notin P_1$ (np. rys. 4.3.(a)). Oczywiście vw jest jedynym wspólnym poprzednikiem dwóch wierzchołków $v_{e(vw),0}$, $w_{0,e(vw)}$, które nie mogą równocześnie należeć do P_2, gdyż P_2 jest zbiorem wierzchołków drogi $\mathfrak{h}^{\pi}(P_2)$ w $\mathbb{G}_{\mathcal{G},e}$. Dodatkowo, $P_2 \setminus \{vw\} \subseteq \mathcal{V}(\mathcal{N}_w)$, stąd vw jest jedynym wierzchołkiem z P_2, który jest połączalny z jakimkolwiek wierzchołkiem z $\mathcal{V}(\mathcal{N}_v)$. Stąd $(P_2, P_1) \in \mathcal{A}(\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi))$ oraz $P_1 \underset{\mathcal{A}(\mathcal{G}(\mathbb{G}_{\mathcal{G},e},\pi))}{\rightarrow} P_2 \underset{\mathcal{A}(\mathcal{G}(\mathbb{G}_{\mathcal{G},e},\pi))}{\rightarrow} P_1$, co przeczy acykliczności partycji $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi)$. Uzyskana sprzeczność dowodzi więc, że $vw \in P_1$, skąd już łatwo można pokazać, że $v_{e(vw),0} \in P_1$, $w_{0,e(vw)} \in P_2$.

Krok 3. Rozważmy dwie acykliczne $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e_{|\pi,\mathfrak{u}}})}$–partycje:

$$\begin{aligned}
\overline{\mathcal{L}}_{\mathfrak{u}} :=& \{\mathcal{V}(\swarrow^{k}_{\mathfrak{u}}) : 0 \leqslant k \leqslant |\mathcal{A}(\mathcal{G})| \wedge k \notin e(L)\} \cup \\
& \{\{\mathfrak{u}l_k\} \cup \mathcal{V}(\swarrow^{k}_{\mathfrak{u}}) : 0 \leqslant k \leqslant i\} \cup \{\{r_k\mathfrak{u}\} : 1 \leqslant k \leqslant j\}, \\
\overline{\mathcal{R}}_{\mathfrak{u}} :=& \{\mathcal{V}(\searrow^{k}_{\mathfrak{u}}) : 0 \leqslant k \leqslant |\mathcal{A}(\mathcal{G})| \wedge k \notin e(R)\} \cup \\
& \{\{r_k\mathfrak{u}\} \cup \mathcal{V}(\searrow^{k}_{\mathfrak{u}}) : 0 \leqslant k \leqslant i\} \cup \{\{\mathfrak{u}l_k\} : 1 \leqslant k \leqslant i\}.
\end{aligned} \tag{4.12}$$

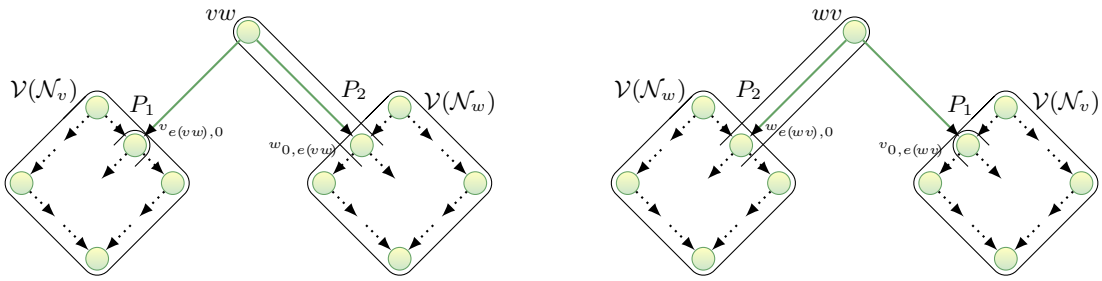
Oczywiście $|\overline{\mathcal{L}}_{\mathfrak{u}}| = |\mathcal{A}(\mathcal{G})|+1+i$, $|\overline{\mathcal{R}}_{\mathfrak{u}}| = |\mathcal{A}(\mathcal{G})|+1+j$. Z założenia przyjętego o digrafie $\mathcal{G}$ uzyskujemy, że $|\mathcal{N}^{-}_{\mathcal{G}}(\mathfrak{u})| = 1$ lub $|\mathcal{N}^{+}_{\mathcal{G}}(\mathfrak{u})| = 1$. Bez straty ogólności przyjmijmy, że $|\mathcal{N}^{-}_{\mathcal{G}}(\mathfrak{u})| = 1$ (przypadek $|\mathcal{N}^{+}_{\mathcal{G}}(\mathfrak{u})| = 1$ jest analogiczny). Istnieje wówczas wierzchołek $\mathfrak{r} \in \mathcal{V}(\mathcal{G})$, dla którego $\mathcal{N}^{-}_{\mathcal{G}}(\mathfrak{u}) = \{\mathfrak{r}\}$. Dodatkowo, $j \leqslant 1$ (j może mieć wartość 0, gdyż przypadek $\mathfrak{ru} \notin \mathcal{V}(\mathbb{G}_{\mathcal{G},e_{|\pi,\mathfrak{u}}})$ nie jest wykluczony).

Rozważmy dowód przez przypadki:

(1°) *Załóżmy, że* $j = 1, i \geqslant 1$. Korzystając z *Kroku* 1 uzyskujemy, że $|\pi_{|\mathfrak{u}}| \geqslant |\mathcal{A}(\mathcal{G})|+2 = |\overline{\mathcal{L}}_{\mathfrak{u}}|$, skąd $\pi' = (\pi \setminus \pi_{|\mathfrak{u}}) \cup \overline{\mathcal{L}}_{\mathfrak{u}}$ jest $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycją $\mathbb{G}_{\mathcal{G},e_{|\pi,\mathfrak{u}}}$, dla której $|\pi'| \leqslant |\pi|$. Dodatkowo $\mathcal{N}_{\mathfrak{u}}$ ma $\swarrow$–orientację w π' oraz pozostałe gadżety, które były zorientowane w π zachowały swoją orientację w π'. W celu zakończenia dowodu w tym przypadku pokażemy jedynie, że $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$ jest digrafem acyklicznym.

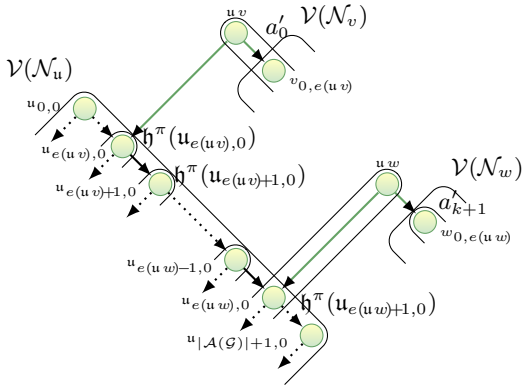
Załóżmy nie wprost, że istnieje $\mathfrak{s}$, które jest $\mathcal{A}(\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi'))$–cyklem w digrafie $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$. Ponieważ digraf $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi)$ jest strukturą acykliczną, więc zbiór wierzchołków $\mathcal{V}(\mathfrak{s})$ musi mieć niepusty przekrój z $\overline{\mathcal{L}}_{\mathfrak{u}}$. Wierzchołek $\{\mathfrak{ru}\}$ jako źródło w $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$ nie może należeć do $\mathfrak{s}$, więc cykl ten, jeśli zawiera segment zbudowany z wierzchołków należących do $\pi'_{|\mathfrak{u}}$, to każdy z tych wierzchołków ma niepusty przekrój z $\mathcal{V}(\mathcal{N}_{\mathfrak{u}})$. W celu uzyskania sprzeczności pokażemy teraz, że każdy maksymalny w sensie zawierania segment cyklu $\mathfrak{s}$ zbudowany wyłącznie z wierzchołków należących do $\pi'_{|\mathfrak{u}}$ może zostać zastąpiony drogą zbudowaną z wierzchołków należących do $\pi_{|\mathfrak{u}}$, w taki sposób, że po iteracyjnym zastąpieniu poszczególnych maksymalnych segmentów określonego rodzaju, uzyskany ciąg będzie cyklem w acyklicznym digrafie $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi)$.

Ustalmy więc segment $\mathfrak{s}' = \langle a_1, a_2, \ldots, a_k \rangle$ drogi $\mathfrak{s}$ zbudowany z wierzchołków należących do $\overline{\mathcal{L}}_{\mathfrak{u}}$ oraz oznaczmy dwa wierzchołki $a_0, a_{k+1} \in \mathcal{V}(\mathfrak{s}) \setminus \overline{\mathcal{L}}_{\mathfrak{u}}$, dla których a_0a_1, $a_ka_{k+1} \in \mathcal{A}(\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi'))$ (zob. rys. 4.4). Dodatkowo $\mathfrak{s}' \neq \mathfrak{s}$ oraz $a_0, a_{k+1} \notin \{a_1, a_2, \ldots, a_k\}$ gdyż $\overline{\mathcal{L}}_{\mathfrak{u}}$ jest acykliczną $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e_{|\pi,\mathfrak{u}}})}$–partycją. Wykorzystując fakt, że $\{\mathfrak{ru}\}$ jest źródłem w $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$, stwierdzamy, iż $a_i \cap \mathcal{V}(\mathcal{N}_{\mathfrak{u}}) \neq \emptyset$ dla każdego $i = 1, 2, \ldots, n$. Dodatkowo istnieją wierzchołki $v, w \in \mathcal{V}(\mathcal{G})$, dla których $\mathfrak{u}v \in \mathcal{A}(\mathcal{G})$, $a_0 \cap \mathcal{V}(\mathcal{N}_v) \neq \emptyset$ oraz $\mathfrak{u}w \in \mathcal{A}(\mathcal{G})$, $a_{k+1} \cap \mathcal{V}(\mathcal{N}_w) \neq \emptyset$. Korzystając wówczas z *Kroku* 2, uzyskujemy, że $\mathfrak{u}v \in a_0$, $\mathfrak{u}_{e(\mathfrak{u}v),0} \in a_1$, $\mathfrak{u}w \in a_k$,



(*a*) przypadek $vw \in P_1 \cup P_2$ (*b*) przypadek $wv \in P_1 \cup P_2$

Rysunek 4.3: Fragment digrafu $\mathbb{G}_{\mathcal{G},e}$, ilustrujący rozumowanie nie wprost zawarte w uzasadnieniu drugiego kroku w twierdzeniu Tw. 4.14.



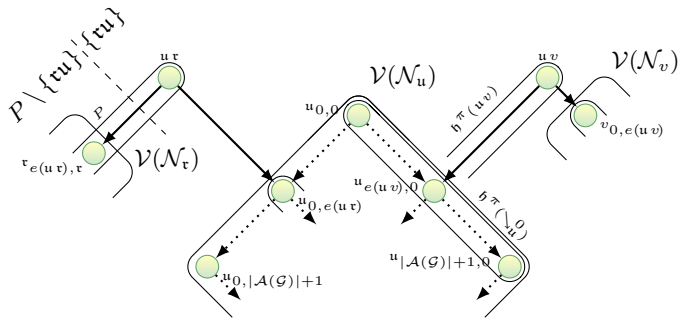
Rysunek 4.4: Fragment digrafu $\mathbb{G}_{\mathcal{G},e}$, ilustrujący konstrukcję zawartą w uzasadnieniu przypadku (1°), kroku 3, twierdzenia Tw. 4.14.

$\mathfrak{u}_{e(\mathfrak{u}w),0} \in a_k$, $w_{0,e(\mathfrak{u}w)} \in a_{k+1}$, skąd w szczególności $v, w \in \mathcal{N}^+_{\mathcal{G}}(\mathfrak{u})$. Dodatkowo, jak łatwo można uzasadnić, $e(\mathfrak{u}v) < e(\mathfrak{u}w)$. Rozważmy ciąg:

$$\mathfrak{t} := \langle \mathcal{V}(\mathfrak{h}^\pi(\mathfrak{u}_{e(\mathfrak{u}v),0})), \mathcal{V}(\mathfrak{h}^\pi(\mathfrak{u}_{e(\mathfrak{u}v)+1,0})), \mathcal{V}(\mathfrak{h}^\pi(\mathfrak{u}_{e(\mathfrak{u}v)+2,0})), \ldots, \mathcal{V}(\mathfrak{h}^\pi(\mathfrak{u}_{e(\mathfrak{u}w),0}))\rangle, \tag{4.13}$$

wierzchołków z $\pi_{|\mathfrak{u}}$. Oczywiście kolejne wyrazy w tym ciągu nie muszą być parami różne. Rozważmy zatem maksymalny podciąg $\mathfrak{t}' = \langle t_1, t_2, \ldots, t_{k'}\rangle$ ciągu $\mathfrak{t}$ taki, że $t_i \neq t_{i+1}$ dla wszystkich $i = 1, 2, \ldots, k'-1$. Z konstrukcji podciągu $\mathfrak{t}'$ uzyskujemy, że dla każdego i istnieje j, dla którego $e(\mathfrak{u}v) \leqslant j \leqslant e(\mathfrak{u}w)$, $t_i = \mathcal{V}(\mathfrak{h}^\pi(\mathfrak{u}_{j,0}))$, $t_{i+1} = \mathcal{V}(\mathfrak{h}^\pi(\mathfrak{u}_{j+1,0}))$. Z definicji $\mathbb{G}_{\mathcal{G},e}$ stwierdzamy, że $\mathfrak{u}_{j,0}\mathfrak{u}_{j+1,0} \in \mathcal{A}(\mathbb{G}_{\mathcal{G},e})$, skąd t_it_{i+1} jest łukiem w $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi)$. Zatem $\mathfrak{t}'$ jest skierowaną
$\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi)$–drogą, dla której a_0t_1, $t_{k'}a_{k+1}$ są $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi)$–łukami, co ostatecznie kończy dowód tego przypadku.

(2°) *Załóżmy, że* $j = i = 0$. Oznaczmy $\pi' := (\pi \setminus \pi_{|\mathfrak{u}}) \cup \overline{\mathcal{L}}_{\mathfrak{u}}$. Korzystając z *Kroku* 1, stwierdzamy, że $|\pi_{|\mathfrak{u}}| \geqslant |\mathcal{A}(\mathcal{G})| + 1 = |\overline{\mathcal{L}}_{\mathfrak{u}}|$, skąd łatwo można uzasadnić, że π' jest $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycją $\mathbb{G}_{\mathcal{G},e}$ oraz $|\pi'| \leqslant |\pi|$. Zauważmy również, że każdy $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$–łuk, łączący wierzchołek należący do zbioru $\pi \setminus \pi_{|\mathfrak{u}}$ z wierzchołkiem



Rysunek 4.5: Fragment digrafu $\mathbb{G}_{\mathcal{G},e}$, ilustrujący konstrukcję zawartą w uzasadnieniu przypadku (3°), kroku 3, twierdzenia Tw. 4.14.

należącym do $\overline{\mathcal{L}}_\mathfrak{u}$ jest zawsze skierowany do $\overline{\mathcal{L}}_\mathfrak{u}$. Stąd uzyskujemy, że digraf $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$ jest strukturą acykliczną, ostatecznie kończy to dowód tego przypadku.

(3°) *Załóżmy, że* $j = 0$, $i > 0$. Przyjmijmy, że $\pi' := (\pi \setminus \pi_{|\mathfrak{u}}) \cup \overline{\mathcal{L}}_\mathfrak{u}$. Oczywiście $|\pi_{|\mathfrak{u}}| \geqslant |\mathcal{A}(\mathcal{G})|+1 = |\overline{\mathcal{L}}_\mathfrak{u}|$ na mocy *Kroku* 1, skąd $|\pi'| \leqslant |\pi|$ oraz π' jest $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycją $\mathbb{G}_{\mathcal{G},e}$. Zauważmy, że jeśli digraf $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$ jest acykliczny, to partycja π' spełnia warunki sformułowane w twierdzeniu. Załóżmy więc nie wprost, że $\mathfrak{s}$ jest skierowanym $\mathcal{A}(\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi'))$–cyklem w digrafie $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$. Podobnie jak w uzasadnieniu przypadku (1°) stwierdzamy, że $\mathfrak{s}$ musi przechodzić przez wierzchołki z $\overline{\mathcal{L}}_\mathfrak{u}$, Dodatkowo $j = 0$, więc $P := \mathcal{V}(\mathfrak{h}^{\pi'}(\mathfrak{ru}))$ nie musi być źródłem w $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$. Stąd $\mathfrak{s}$ może przechodzić przez wierzchołki z $\overline{\mathcal{L}}_\mathfrak{u}$, wchodząc i wychodząc z „prawej strony" gadżetu $\mathcal{N}_\mathfrak{u}$ (zob. rys. 4.4), ale może również wchodzić z „lewej strony" gadżetu $\mathcal{N}_\mathfrak{u}$ przez punkt P (zob. rys. 4.5). W przypadku, gdy istnieje skierowany $\mathcal{A}(\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi'))$–cykl wchodzący do $\overline{\mathcal{L}}_\mathfrak{u}$ przez wierzchołek P można pokazać, że $|\pi'| < |\pi|$. Stąd $\pi'' := (\pi' \setminus \{P\}) \cup \{\{\mathfrak{ru}\}, P \setminus \{\mathfrak{ru}\}\}$ będąca $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycją $\mathbb{G}_{\mathcal{G},e}$ powstałą z podzielenia P w π' na dwie części $\{\mathfrak{ru}\}$, $P \setminus \{\mathfrak{ru}\}$, będzie miała co najwyżej $|\pi|$ elementów, zaś gadżety różne od $\mathcal{N}_\mathfrak{u}$, które były zorientowane w π, zachowają swoją orientację w π''. Dodatkowo, $\mathcal{V}(\mathfrak{h}^{\pi''}(\mathfrak{ru})) = \{\mathfrak{ru}\}$ jest źródłem w $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi'')$, a więc każdy cykl w $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi'')$ musi przechodzić przez wierzchołki z $\overline{\mathcal{L}}_\mathfrak{u}$ wchodząc i wychodząc z „prawej strony" gadżetu $\mathcal{N}_\mathfrak{u}$, tak jak miało to miejsce w przypadku (1°). Powtarzając rozumowanie zawarte w tym przypadku, uzyskujemy tezę.

Ustalmy skierowany $\mathcal{A}(\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi'))$–cykl w digrafie $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$ wchodzący z „lewej strony" gadżetu $\mathcal{N}_\mathfrak{u}$ przez punkt P. Pokażemy, iż $|\pi'| < |\pi|$, uzasadniając w tym celu, że $|\mathcal{A}(\mathcal{G})| + 1 < |\pi_{|\mathfrak{u}}|$. Załóżmy nie wprost, że $|\mathcal{A}(\mathcal{G})| + 1 \geqslant |\pi_{|\mathfrak{u}}|$. Korzystając wówczas z Wn. 4.8, stwierdzamy, że $|\mathcal{A}(\mathcal{G})| + 1 \leqslant |(\pi_{|\mathfrak{u}})_{|\mathcal{V}(\mathcal{N}_\mathfrak{u})}| \leqslant |\pi_{|\mathfrak{u}}| \leqslant |\mathcal{A}(\mathcal{G})| + 1$, skąd $(\pi_{|\mathfrak{u}})_{|\mathcal{V}(\mathcal{N}_\mathfrak{u})} = \mathcal{L}_\mathfrak{u}$ lub $(\pi_{|\mathfrak{u}})_{|\mathcal{V}(\mathcal{N}_\mathfrak{u})} = \mathcal{R}_\mathfrak{u}$ na mocy Lem. 4.9 oraz $|(\pi_{|\mathfrak{u}})_{|\mathcal{V}(\mathcal{N}_\mathfrak{u})}| = |\pi_{|\mathfrak{u}}|$. Wówczas w przypadku $(\pi_{|\mathfrak{u}})_{|\mathcal{V}(\mathcal{N}_\mathfrak{u})} = \mathcal{L}_\mathfrak{u}$ otrzymujemy równości $\pi_{|\mathfrak{u}} = \overline{\mathcal{L}}_\mathfrak{u}$, $\pi' = \pi$, która przeczy założeniu, że digraf $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi)$ jest acykliczny. W konsekwencji $(\pi_{|\mathfrak{u}})_{|\mathcal{V}(\mathcal{N}_\mathfrak{u})} = \mathcal{R}_\mathfrak{u}$. Rozważmy wierzchołek $\mathfrak{u}l_1 \in \mathcal{V}(\mathbb{G}_{\mathcal{G},e_{|\pi,v}})$, który jest połączalny z dokładnie jednym wierzchołkiem z $\mathcal{V}(\mathbb{G}_{\mathcal{G},e_{|\pi,v}})$, mianowicie $\mathfrak{u}_{e(\mathfrak{u}l_1),0}$. Oczywiście wierzchołek $\mathfrak{u}_{e(\mathfrak{u}l_1),0}$

należy do zbioru $\mathcal{V}(\searrow_{\mathfrak{u}}^{0})$ będącego elementem $\mathcal{R}_{\mathfrak{u}}$. Wykorzystując więc równość $|\pi_{|\mathfrak{u}}| = |\mathcal{R}_{\mathfrak{u}}|$, stwierdzamy, że $\{\mathfrak{u}l_1\} \cup \mathcal{V}(\searrow_{\mathfrak{u}}^{0}) \subseteq \mathcal{V}(\mathfrak{h}^{\pi}(\mathfrak{u}_{e(\mathfrak{u}l_1),0}))$, ale zbiór $\{\mathfrak{u}l_1\} \cup \mathcal{V}(\searrow_{\mathfrak{u}}^{0})$ posiada dwa źródła z $\mathbb{G}_{\mathcal{G},e}$, mianowicie $\mathfrak{u}l_1$, $\mathfrak{u}_{0,0}$, co przeczy istnieniu skierowanej drogi Hamiltona $\mathfrak{h}^{\pi}(\mathfrak{u}l_1)$.

(4°) *Załóżmy, że* $j = 1, i = 0$. Przyjmijmy $\pi' := (\pi \setminus \pi_{|\mathfrak{u}}) \cup \overline{\mathcal{R}}_{\mathfrak{u}}$. Zauważmy, że jeśli digraf $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$ jest acykliczny, to partycja π' spełnia warunki sformułowane w twierdzeniu, gdyż $|\pi_{|\mathfrak{u}}| \geqslant |\mathcal{A}(\mathcal{G})| + 1 = |\overline{\mathcal{R}}_{\mathfrak{u}}|$ na mocy *Kroku* 1. Przypuśćmy nie wprost, że digraf $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$ nie jest acykliczny. Istnieje wówczas skierowany $\mathcal{A}(\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi'))$–cykl $\mathfrak{s}$ w digrafie $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$, który musi przechodzić przez $\mathcal{R}_{\mathfrak{u}}$, wchodząc do tej partycji z „prawej strony", ponieważ $i = 0$, a wychodząc przez pewien wierzchołek z wierzchołka $P := \mathcal{V}(\mathfrak{h}^{\pi'}(\mathfrak{r}\mathfrak{u}))$. Prowadząc rozważania podobne do przypadku (3°), da się pokazać, że $|\mathcal{A}(\mathcal{G})| + 1 > |\pi_{|\mathfrak{u}}|$. Określmy partycję $\pi'' := (\pi' \setminus \{P\}) \cup \{\{\mathfrak{r}\mathfrak{u}\}, P \setminus \{\mathfrak{r}\mathfrak{u}\}\}$. Zauważmy, iż każdy $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi'')$–łuk łączący, łączący wierzchołek należący do zbioru $\pi \setminus \pi_{|\mathfrak{u}}$ z wierzchołkiem należącym do $\pi'' \setminus (\pi \setminus \pi_{|\mathfrak{u}})$ jest skierowany do $\pi'' \setminus (\pi \setminus \pi_{|\mathfrak{u}})$. W związku z tym dostajemy, że π'' jest acykliczną $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycją $\mathbb{G}_{\mathcal{G},e}$ spełniającą warunki określone w twierdzeniu.

Ustalmy skierowany $\mathcal{A}(\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi'))$–cykl w digrafie $\mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi')$ wychodzący z „prawej strony" gadżetu $\mathcal{N}_{\mathfrak{u}}$ przez punkt P. Pokażemy, że $|\mathcal{A}(\mathcal{G})| + 1 < |\pi_{|\mathfrak{u}}|$. Powtarzając rozumowanie zawarte w przypadku (3°), stwierdzamy, że następstwem założenia nie wprost $|\mathcal{A}(\mathcal{G})| + 1 \geqslant |\pi_{|\mathfrak{u}}|$ jest równość $|(\pi_{|\mathfrak{u}})_{|\mathcal{V}(\mathcal{N}_{\mathfrak{u}})}| = |\pi_{|\mathfrak{u}}|$, a w konsekwencji również $(\pi_{|\mathfrak{u}})_{|\mathcal{V}(\mathcal{N}_{\mathfrak{u}})} = \mathcal{L}_{\mathfrak{u}}$ lub $(\pi_{|\mathfrak{u}})_{|\mathcal{V}(\mathcal{N}_{\mathfrak{u}})} = \mathcal{R}_{\mathfrak{u}}$. W analogiczny sposób eliminowany jest również przypadek $(\pi_{|\mathfrak{u}})_{|\mathcal{V}(\mathcal{N}_{\mathfrak{u}})} = \mathcal{R}_{\mathfrak{u}}$, którego następstwem jest równość $\pi = \pi'$ prowadząca do sprzeczności. Mamy zatem $(\pi_{|\mathfrak{u}})_{|\mathcal{V}(\mathcal{N}_{\mathfrak{u}})} = \mathcal{L}_{\mathfrak{u}}$. Posługując się w tym momencie wierzchołkiem $\mathfrak{r}\mathfrak{u}$ zamiast $\mathfrak{u}l_1$ oraz drogą $\searrow_{\mathfrak{u}}^{0}$ zamiast $\swarrow_{\mathfrak{u}}^{0}$, uzyskujemy analogicznie sprzeczność kończącą dowód.

□

Twierdzenie 4.15. *Niech* π *będzie acykliczną* $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$*–partycją digrafu* $\mathbb{G}_{\mathcal{G},e}$*, w której każdy gadżet jest zorientowany. Wówczas zbiór* $\mathcal{A}(\mathcal{G})$*–łuków* $\{vu : \{vu\} \in \pi\}$ *jest zbiorem sprzężonym o mocy ograniczonej liczbą* $|\pi| - |\mathcal{V}(\mathcal{G})| \cdot (|\mathcal{A}(\mathcal{G})| + 1)$.

Dowód. Niech π spełnia założenia twierdzenia. Zauważmy najpierw, że w przypadku $\mathcal{A}(\mathcal{G}) = \emptyset$ twierdzenie jest spełnione w sposób trywialny, gdyż $\{vu : \{vu\} \in \pi\} \subseteq \mathcal{A}(\mathcal{G})$. Przyjmijmy więc, że $\mathcal{A}(\mathcal{G})$ jest zbiorem niepustym. Wprowadźmy dwa pomocnicze oznaczenia $\mathcal{F} := \{vu : \{vu\} \in \pi\}$, $\mathcal{O} := \{\{vu\} \in \pi\}$. Pokażemy, że $\mathcal{F}$ jest zbiorem sprzężonym w $\mathcal{G}$. Załóżmy nie wprost, że istnieje trasa $\mathfrak{c} := c_1 \underset{\mathcal{A}(\mathcal{G})\setminus\mathcal{F}}{\rightarrow} c_2 \underset{\mathcal{A}(\mathcal{G})\setminus\mathcal{F}}{\rightarrow} \ldots \underset{\mathcal{A}(\mathcal{G})\setminus\mathcal{F}}{\rightarrow} c_k \underset{\mathcal{A}(\mathcal{G})\setminus\mathcal{F}}{\rightarrow} c_1$ będąca $\mathcal{A}(\mathcal{G})\setminus\mathcal{F}$–cyklem w $\mathcal{G}$, gdzie $k > 1$. Oznaczmy $c_{k+1} := c_1$ oraz niech

$$\begin{aligned} \swarrow^{uv} &:= uv \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\rightarrow} \underbrace{u_{e(uv),0} \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\rightarrow} u_{e(uv),1} \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\rightarrow} \cdots \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\rightarrow} u_{e(uv),|\mathcal{A}(\mathcal{G})|}}_{\swarrow_u^{e(uv)}}, \\ {}^{uv}\searrow &:= uv \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\rightarrow} \underbrace{v_{0,e(uv)} \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\rightarrow} v_{1,e(uv)} \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\rightarrow} \cdots \underset{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}{\rightarrow} v_{|\mathcal{A}(\mathcal{G})|,e(uv)}}_{\searrow_v^{e(uv)}}, \end{aligned} \tag{4.14}$$

gdzie $uv \in \mathcal{A}(\mathcal{G})$. Jak łatwo można uzasadnić, dla każdego $\mathcal{A}(\mathcal{G})\setminus\mathcal{F}$–łuku c_ic_{i+1} zachodzi dokładnie jedna tożsamość: albo $\mathfrak{h}^{\pi}(c_ic_{i+1}) = \swarrow^{c_ic_{i+1}}$ albo $\mathfrak{h}^{\pi}(c_ic_{i+1}) = {}^{c_ic_{i+1}}\searrow$,

gdzie $1 \leqslant i \leqslant k$. Dodatkowo następstwem równości $\mathfrak{h}^\pi(c_i c_{i+1}) = \swarrow^{c_i c_{i+1}}$ jest $\swarrow$–orientacja gadżetu $\mathcal{N}_{c_i}$, a następstwem równości $\mathfrak{h}^\pi(c_i c_{i+1}) = {}^{c_i c_{i+1}}\searrow$ jest $\searrow$–orientacja gadżetu $\mathcal{N}_{c_{i+1}}$. Ponieważ przy $|\mathcal{A}(\mathcal{G})| > 0$ żaden gadżet nie może mieć jednocześnie obu orientacji w π, uzyskujemy, że albo każdy gadżet $\mathcal{N}_{c_i}$ ma $\swarrow$–orientację w π dla $i = 1, 2, \ldots k$, albo każdy gadżet $\mathcal{N}_{c_i}$ ma $\searrow$–orientację w π dla $i = 1, 2, \ldots k$. Wówczas w przypadku $\swarrow$–orientacji,

$$
\begin{aligned}
&\mathcal{V}(\swarrow^0_{c_1}) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\quad \mathcal{V}(\mathfrak{h}^\pi((c_1)_{1,0})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\mathfrak{h}^\pi((c_1)_{2,0})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \cdots \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\mathfrak{h}^\pi((c_1)_{e(c_1c_2)-1,0})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\qquad\qquad \mathcal{V}(\swarrow^{c_1c_2}) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\swarrow^0_{c_2}) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\quad \mathcal{V}(\mathfrak{h}^\pi((c_2)_{1,0})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\mathfrak{h}^\pi((c_2)_{2,0})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \cdots \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\mathfrak{h}^\pi((c_2)_{e(c_2c_3)-1,0})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\qquad\qquad \mathcal{V}(\swarrow^{c_2c_3}) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\swarrow^0_{c_2}) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\qquad\qquad \cdots \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\quad \mathcal{V}(\mathfrak{h}^\pi((c_k)_{1,0})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\mathfrak{h}^\pi((c_k)_{2,0})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \cdots \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\mathfrak{h}^\pi((c_k)_{e(c_kc_{k+1})-1,0})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \mathcal{V}(\swarrow^{c_kc_{k+1}}) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\swarrow^0_{c_{k+1}})
\end{aligned}
\tag{4.15}
$$

jest $\mathcal{A}(\mathfrak{G})$–cyklem, a w przypadku $\searrow$–orientacji,

$$
\begin{aligned}
&\mathcal{V}(\searrow^0_{c_k}) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\quad \mathcal{V}(\mathfrak{h}^\pi((c_k)_{0,1})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\mathfrak{h}^\pi((c_k)_{0,2})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \cdots \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\mathfrak{h}^\pi((c_k)_{0,e(c_{k-1}c_k)-1})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\qquad\qquad \mathcal{V}({}^{c_{k-1}c_k}\searrow) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\searrow^0_{c_{k-1}}) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\quad \mathcal{V}(\mathfrak{h}^\pi((c_{k-1})_{0,1})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\mathfrak{h}^\pi((c_{k-1})_{0,2})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \cdots \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\mathfrak{h}^\pi((c_{k-1})_{0,e(c_{k-2}c_{k-1})-1})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\qquad\qquad \mathcal{V}({}^{c_{k-2}c_{k-1}}\searrow) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\searrow^0_{c_{k-2}}) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\qquad\qquad \cdots \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\quad \mathcal{V}(\mathfrak{h}^\pi((c_1)_{0,1})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\mathfrak{h}^\pi((c_1)_{0,2})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \cdots \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\mathfrak{h}^\pi((c_1)_{0,e(c_kc_1)-1})) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \mathcal{V}({}^{c_kc_1}\searrow) \underset{\mathcal{A}(\mathfrak{G})}{\rightarrow} \mathcal{V}(\searrow^0_{c_k})
\end{aligned}
\tag{4.16}
$$

jest $\mathcal{A}(\mathfrak{G})$–cyklem, którego istnienie jest sprzeczne z acyklicznością digrafu $\mathfrak{G}$, gdzie $\mathfrak{G} := \mathcal{G}(\mathbb{G}_{\mathcal{G},e}, \pi)$.

W celu zakończenia dowodu wystarczy jedynie zauważyć, że $\pi \setminus \mathcal{O} = \bigcup_{v \in \mathcal{V}(\mathcal{G})} \pi_{|v}$, $|\pi_{|v}| = |\mathcal{A}(\mathcal{G})| + 1$, dla każdego $v \in \mathcal{V}(\mathcal{G})$ oraz $\pi_{|v_1} \cap \pi_{|v_2} = \emptyset$ dla różnych wierzchołków $v_1, v_2 \in \mathcal{V}(\mathcal{G})$. □

Twierdzenie 4.16. *Problem APH jest NP-zupełny.*

Dowód. Dla każdego digrafu $\mathcal{G}$ oraz liczby naturalnej K będących instancją problemu FAS określoną w Obs. 4.4 możemy ustalić funkcję $e : \mathcal{A}(\mathcal{G}) \longrightarrow \{1, 2, \ldots, |\mathcal{A}(\mathcal{G})|\}$ i skonstruować digraf $\mathbb{G}_{\mathcal{G},e}$ oraz przyjąć liczbę $K' := K + |\mathcal{V}(\mathcal{G})| \cdot (|\mathcal{A}(\mathcal{G})| + 1)$. W ten sposób zostaje określona instancja problemu APH. Wykorzystując Tw. 4.11, jeśli $\mathcal{G}$

posiada zbiór sprzężony łuków o liczebności nie przekraczającej K, to istnieje acykliczna $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycja digrafu $\mathbb{G}_{\mathcal{G},e}$ o liczebności nie większej niż K'. Wykorzystując następnie Tw. 4.14 oraz 4.15 stwierdzamy, że jeśli istnieje acykliczna $\mathcal{H}^{(*)}_{\mathcal{A}(\mathbb{G}_{\mathcal{G},e})}$–partycja π digrafu $\mathbb{G}_{\mathcal{G},e}$ o liczebności nie większej niż K' to istnieje zbiór sprzężony łuków o mocy co najwyżej K. Stąd ostatecznie translacja $\mathcal{G}$, K do $\mathbb{G}_{\mathcal{G},e}$, K' jest redukcją. □

Wykorzystywany w uzasadnieniu NP-zupełności problemu APH digraf $\mathbb{G}_{\mathcal{G},e}$ posiada ograniczoną do dwóch liczebność zbioru następników oraz poprzedników każdego wierzchołka. Fakt ten umożliwia sformułowanie twierdzenia Tw. 4.16 w poniższej postaci.

Twierdzenie 4.17. *Problem APH zachowuje NP-zupełność w rodzinie digrafów, której każdy element D spełnia zależność: $|\mathcal{N}^-_D(v)| \leqslant 2$ oraz $|\mathcal{N}^+_D(v)| \leqslant 2$ dla każdego $v \in \mathcal{V}(D)$.*

4.2 Złożoność problemu $\mathcal{K}.3_{MIZ}$

W podrozdziale tym skupimy uwagę na złożoności problemu $\mathcal{K}.3_{MIZ}$. Wykażemy, że problem ten jest rozwiązywalny w czasie wielomianowym i opiszemy algorytm rozwiązujący ten problem w czasie $O(|\mathcal{V}(D)| + |\mathcal{A}(D)|)$, gdzie digraf D jest instancją problemu $\mathcal{K}.3_{MIZ}$.

Twierdzenie 4.18. *Problem $\mathcal{K}.3_{MIZ}$ jest rozwiązywalny w czasie wielomianowym.*

Dowód. Ustalmy acykliczny digraf D oraz dwa zbiory łuków $A_1 \subseteq A_2 \subseteq \mathcal{A}(D)$ będące instancją problemu $\mathcal{K}.3_{MIZ}$. Pokażemy, że wyznaczenie najmniejszej liczby naturalnej K, dla której istnieje sortowanie topologiczne $\tau \in TS(D)$ spełniające zależność $|\mathfrak{L}_\tau| \leqslant K$ jest problemem rozwiązywalnym w czasie wielomianowym, gdzie

$$\mathfrak{L}_\tau = \{v \in \mathcal{V}(D) : \underset{u \in \mathcal{V}(D)}{\exists} vu \in A_2 \wedge (\, d_\tau(v,u) > 1 \vee vu \notin A_1 \vee \underset{w \in \mathcal{V}(D)}{\exists} vw \in \mathcal{A}(D) \setminus A_2\,)\}. \tag{4.17}$$

Wprowadźmy oznaczenia:

$$\begin{aligned} L_1 &:= \{v \in \mathcal{V}(D) : \underset{u \in \mathcal{V}(D)}{\exists} vu \in A_2 \setminus A_1\}, \\ L_2 &:= \{v \in \mathcal{V}(D) : \underset{u,w \in \mathcal{V}(D)}{\exists} vu \in A_2 \wedge vw \in \mathcal{A}(D) \setminus A_2\}, \\ L_3 &:= \{v \in \mathcal{V}(D) : |\mathcal{N}^+_{\langle \mathcal{V}(D), A_1 \rangle}(v)| > 1\}. \end{aligned} \tag{4.18}$$

Jak widać, zbiory te nie nie zależą od konkretnego wyboru τ. Zauważmy teraz, że zawierania $L_1, L_2 \subseteq \mathfrak{L}_\tau$ wynikają bezpośrednio z definicji zbioru $\mathfrak{L}_\tau$. W celu wykazania, że $L_3 \subseteq \mathfrak{L}_\tau$ ustalmy wierzchołek $v \in L_3$, dla którego istnieją różne $u_1, u_2 \in \mathcal{N}^+_{\langle \mathcal{V}(D), A_1 \rangle}(v)$ w digrafie $\langle \mathcal{V}(D), A_1 \rangle$. Wówczas $vu_1 \in A_1$, $vu_2 \in A_1$, ale nie mogą być spełnione jednocześnie obie równości $d_\tau(v,u_1) = 1$, $d_\tau(v,u_2) = 1$ jeśli $\tau(v) < \tau(u_1), \tau(u_2)$ oraz $u_1 \neq u_2$. Stąd, $d_\tau(v,u_1) > 1$ lub $d_\tau(v,u_2) > 1$, a zatem ostatecznie $v \in \mathfrak{L}_\tau$.

Wprowadźmy następnie oznaczenia $L_0 := L_1 \cup L_2 \cup L_3$, $R_0 := \{vu \in A_1 : v \in L_0\}$. Naturalnie zbiory L_0, R_0 mogą być wyznaczone w czasie wielomianowym względem

rozmiaru D (zob. wydruk 4.6). Dodatkowo, jak łatwo można wykazać:

$$\begin{aligned}
\mathfrak{L}_\tau \setminus L_0 &= (\mathfrak{L}_\tau \setminus L_1) \setminus (L_2 \cup L_3) \\
&= (\{v \in \mathcal{V}(D) : \underset{u \in \mathcal{V}(D)}{\exists} vu \in A_1 \cap A_2 \wedge \\
&\quad (\, d_\tau(v,u) > 1 \vee \underset{w \in \mathcal{V}(D)}{\exists} vw \in \mathcal{A}(D) \setminus A_2 \,)\}) \setminus (L_2 \cup L_3) \\
&= (\{v \in \mathcal{V}(D) : \underset{u \in \mathcal{V}(D)}{\exists} vu \in A_1 \wedge d_\tau(v,u) > 1\} \cup \\
&\quad \{v \in \mathcal{V}(D) : \underset{u \in \mathcal{V}(D)}{\exists} vu \in A_1 \wedge d_\tau(v,u) = 1 \wedge \underset{w \in \mathcal{V}(D)}{\exists} vw \in \mathcal{A}(D) \setminus A_2\}) \setminus \\
&\quad (L_2 \cup L_3) \\
&= \{v \in \mathcal{V}(D) : \underset{u \in \mathcal{V}(D)}{\exists} vu \in A_1 \wedge d_\tau(v,u) > 1\} \setminus (L_2 \cup L_3) \\
&= (\{v \in \mathcal{V}(D) : \underset{u \in \mathcal{V}(D)}{\exists} vu \in A_1 \wedge d_\tau(v,u) > 1\} \setminus L_0 \\
&= \{v \in \mathcal{V}(D) : \underset{u \in \mathcal{V}(D)}{\exists} vu \in A_1 \setminus R_0 \ \wedge \ d_\tau(v,u) > 1\}.
\end{aligned} \tag{4.19}$$

Stąd, problem wyznaczenia minimalnej liczebności zbioru $\mathfrak{L}_\tau$ po wszystkich $\tau \in TS(D)$, może być sprowadzony do wyznaczenia minimalnej liczebności zbioru $\mathfrak{L}_\tau \setminus L_0$.

Zauważmy, że konsekwencją odrzucenia zbioru L_3 od $\mathfrak{L}_\tau$ jest ograniczenie na liczebność zbioru następników w digrafie $\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle$. Precyzując:

$$|\mathcal{N}^+_{\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle}(v)| \leqslant 1, \tag{4.20}$$

dla każdego $v \in \mathcal{V}(D)$, skąd $\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle$ jest lasem dendroidów. Dodatkowo $\tau(w) < \tau(v)$ dla każdego $w \in \mathcal{N}^-_{\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle}(v)$ skąd co najwyżej jeden wierzchołek ze zbioru $\mathcal{N}^-_{\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle}(v)$ może spełniać tożsamość $d_\tau(w,v) = 1$. Tym samym, jeśli zbiór $\mathcal{N}^-_{\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle}(v)$ jest niepusty, to co najmniej $|\mathcal{N}^-_{\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle}(v)| - 1$ elementów musi należeć do zbioru $\mathfrak{L}_\tau \setminus L_0$. Dodatkowo zbiory poprzedników różnych wierzchołków w każdym lesie dendroidów są rozłączne, w związku z czym możliwe jest dolne oszacowanie na liczebność zbioru $\mathfrak{L}_\tau \setminus L_0$ w postaci:

$$|\mathfrak{L}_\tau \setminus L_0| \geqslant \sum_{v \in \mathcal{V}(D)} \max\{0, |\mathcal{N}^-_{\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle}(v)| - 1\}. \tag{4.21}$$

Pokażemy, że istnieje sortowanie topologiczne $\tau' \in TS(D)$, dla którego dowolny niepusty zbiór poprzedników wierzchołka v w digrafie $\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle$, ma dokładnie $|\mathcal{N}^-_{\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle}(v)| - 1$ elementów wspólnych ze zbiorem $\mathfrak{L}_{\tau'} \setminus L_0$. Rozważmy w tym celu funkcję wyboru $\mathcal{C} : V \to \mathcal{V}(D)$ spełniającą zależność:

$$\mathcal{C}(v) \in \mathcal{N}^-_{\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle}(v), \tag{4.22}$$

dla każdego $v \in V$, wybierającą po jednym wierzchołku z każdego niepustego zbioru $\mathcal{N}^-_{\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle}(v)$, gdzie $V = \{v \in \mathcal{V}(D) : \mathcal{N}^-_{\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle}(v) \neq \emptyset\}$. Wprowadźmy oznaczenie $L_\mathcal{C} := \bigcup\limits_{u \in V} (\mathcal{N}^-_{\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle}(u) \setminus \{\mathcal{C}(u)\})$. Korzystając wówczas z (4.21) stwierdzamy, że $|\mathfrak{L}_\tau| \geqslant |L_0| + |L_\mathcal{C}|$ dla każdego $\tau \in TS(D)$. W związku z tym, że liczebność L_0, $L_\mathcal{C}$ nie zależą od τ, dostajemy wniosek, iż ograniczenie K nie może być mniejsze od $|L_0| + |L_\mathcal{C}|$.

W celu zakończenia dowodu skonstruujemy sortowanie topologiczne $\tau_\mathcal{C} \in TS(D)$, dla którego $|\mathfrak{L}_{\tau_\mathcal{C}}| = |L_0| + |L_\mathcal{C}|$. Będzie ono świadczyło o tym, że liczba $|L_0| + |L_\mathcal{C}|$ może być ograniczeniem K, a w związku z uprzednim wnioskiem jest najmniejszym takim ograniczeniem. Rozważmy acykliczny digraf $D_\mathcal{C} = \langle \mathcal{V}(D), A_1 \setminus (R_0 \cup R_\mathcal{C}) \rangle$,

gdzie $R_{\mathcal{C}} = \{vu \in A_1 : v \in L_{\mathcal{C}}\}$. Zauważmy, że konsekwencją odrzucenia zbioru $R_{\mathcal{C}}$ ze zbioru łuków $A_1 \setminus R_0$, jest usunięcie wszystkich łuków w digrafie $\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle$ prowadzących do wierzchołka v za wyjątkiem krawędzi $(\mathcal{C}(v), v)$, jeśli zbiór $\mathcal{N}^-_{D_{\mathcal{C}}}(v)$ był niepusty (równoważnie $v \in V$). Zatem $|\mathcal{N}^-_{D_{\mathcal{C}}}(v)| \leqslant 1$ dla dowolnego $v \in \mathcal{V}(D)$. Analogicznie $|\mathcal{N}^+_{D_{\mathcal{C}}}(v)| \leqslant 1$, gdyż $L_3 \subseteq L_0$. Zatem dowolne dwie maksymalne $D_{\mathcal{C}}$–drogi są równe albo wierzchołkowo rozłączne.

Pokażemy teraz, że $\pi(\mathcal{C}) := \{\mathcal{V}(P) : P \text{ jest maksymalną } D_{\mathcal{C}}\text{–drogą}\}$ jest acykliczną $\mathcal{H}^{(*)}$–partycją digrafu D, czego konsekwencją jest istnieje sortowania topologicznego $\sigma \in \mathcal{G}(D, \pi(\mathcal{C}))$. Załóżmy nie wprost, że istnieje $\mathfrak{s}$, które jest $\mathcal{A}(\mathcal{G}(D, \pi(\mathcal{C})))$–cyklem w digrafire $\mathcal{A}(\mathcal{G}(D, \pi(\mathcal{C})))$. W celu uzyskania sprzeczności pokażemy teraz, że każdy wierzchołek z $\mathfrak{s}$ może zostać zastąpiony odpowiednią $\mathcal{A}(D)$–drogą w taki sposób, że po iteracyjnym zastąpieniu wszystkich wierzchołków, uzyskany ciąg $\mathcal{A}(D)$–dróg będzie cyklem, co stanie w sprzeczności z acyklicznością digrafu D. Ustalmy więc wierzchołek $P \in \mathfrak{s}$ oraz niech

$$P_1 \underset{\mathcal{A}(\mathcal{G}(D,\pi(\mathcal{C})))}{\rightarrow} P \underset{\mathcal{A}(\mathcal{G}(D,\pi(\mathcal{C})))}{\rightarrow} P_2 \tag{4.23}$$

będzie segmentem $\mathfrak{s}$. Z Def. 1.11 istnieją wówczas wierzchołki $v \in P_1$, $u, w \in P$, $r \in P_2$, dla których vu, $wr \in \mathcal{A}(D)$. W celu zakończenia tego podrozumowania wystarczy pokazać, że wierzchołki są v, w są odpowiednio końcami $\mathcal{A}(D)$–dróg $\mathfrak{h}^{\pi(\mathcal{C})}(P_1)$, $\mathfrak{h}^{\pi(\mathcal{C})}(P)$, skąd w szczególności uzyskamy również, że $u \underset{\mathcal{A}(D)}{\rightarrow} {}^*w$ oraz $\mathcal{A}(D)$–droga łącząca u i w jest odpowiednią drogą zastępującą wierzchołek P. Przypuśćmy, że wierzchołek w nie jest ostatnim wierzchołkiem w $\mathfrak{h}^{\pi(\mathcal{C})}(P)$ (dowód w przypadku wierzchołka v jest analogiczny). Oznaczmy przez w_1 wierzchołek występującym bezpośrednio po w w $\mathfrak{h}^{\pi(\mathcal{C})}(P)$. Wykorzystując następnie fakt, że $\mathcal{N}^+_{D_{\mathcal{C}}}(w) \subseteq \mathcal{N}^+_D(w)$ stwierdzamy, że $w_1, r \in \mathcal{N}^+_D(w)$, skąd $w \in L_3$, a zatem $ww_1 \in R_0$. Oznacza to, że wierzchołki w, w_1 nie mogą występować bezpośrednio po sobie w $\mathfrak{h}^{\pi(\mathcal{C})}(P)$ gdyż $ww_1 \notin \mathcal{A}(D_{\mathcal{C}})$ $(= A_1 \setminus (R_0 \cup R_{\mathcal{C}}))$. Uzyskana sprzeczność ostatecznie kończy dowód podrozumowania.

Wybierzmy sortowanie topologiczne $\sigma \in \mathcal{G}(D, \pi(\mathcal{C}))$. Zdefiniujmy funkcję $\tau_{\mathcal{C}} : \mathcal{V}(D) \to |\{1, 2, \ldots, \mathcal{V}(D)\}|$ sortującą wierzchołki w każdej maksymalnej $D_{\mathcal{C}}$–drodze, a następnie scalającą posortowane fragmenty według σ, daną zależnością:

$$\tau_{\mathcal{C}}(v) = i + \sum_{P \in \pi(\mathcal{C}) : \sigma(P) < \sigma(\mathcal{V}(\mathfrak{h}^{\pi(\mathcal{C})}(v)))} |P|, \tag{4.24}$$

gdzie $v = v_i$ oraz $\mathfrak{h}^{\pi(\mathcal{C})}(v) = v_1 \underset{D_{\mathcal{C}}}{\rightarrow} v_2 \underset{D_{\mathcal{C}}}{\rightarrow} \ldots \underset{D_{\mathcal{C}}}{\rightarrow} v_{|\mathcal{V}(\mathfrak{h}^{\pi(\mathcal{C})}(v))|}$. Oczywiście takie uporządkowanie jest sortowaniem topologicznym digrafu D. W celu zakończenia dowodu wystarczy więc jedynie pokazać równość $\mathfrak{L}_{\tau_{\mathcal{C}}} \setminus L_0 = L_{\mathcal{C}}$.

Dla dowodu inkluzji $\mathfrak{L}_{\tau_{\mathcal{C}}} \setminus L_0 \subseteq L_{\mathcal{C}}$ rozważmy dowolny wierzchołek v ze zbioru $\mathfrak{L}_{\tau_{\mathcal{C}}} \setminus L_0$. Załóżmy nie wprost, że $v \notin L_{\mathcal{C}}$. Z $v \in \mathfrak{L}_{\tau_{\mathcal{C}}} \setminus L_0$ i (4.19) wynika, że istnieje wierzchołek $u \in \mathcal{V}(D)$, dla którego $(v, u) \in A_1 \setminus R_0$ oraz $d_{\tau_{\mathcal{C}}}(v, u) > 1$. Stąd z definicji zbioru $R_{\mathcal{C}}$ i faktu, że $v \notin L_{\mathcal{C}}$ stwierdzamy, że $(v, u) \notin R_{\mathcal{C}}$ oraz $(v, u) \in \mathcal{A}(D_{\mathcal{C}})$. Dodatkowo, $D_{\mathcal{C}}$–łuk (v, u) łączy dwa kolejne wierzchołki w $\mathfrak{h}^{\pi(\mathcal{C})}(v)$, skąd $\tau_{\mathcal{C}}(u) - \tau_{\mathcal{C}}(v) = 1$, co jest sprzeczne z warunkiem $d_{\tau_{\mathcal{C}}}(v, u) > 1$.

Dla dowodu inkluzji $L_{\mathcal{C}} \subseteq \mathfrak{L}_{\tau_{\mathcal{C}}} \setminus L_0$ rozważmy v ze zbioru $L_{\mathcal{C}}$. Istnieje wówczas $u \in V$, dla którego $v \in \mathcal{N}^-_{\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle}(u) \setminus \mathcal{C}(u)$, skąd $vu \in A_1 \setminus R_0$. Korzystając wówczas z (4.19), o ile pokażemy, że $d_{\tau_{\mathcal{C}}}(v, u) > 1$, odstajemy szukaną przynależność $v \in \mathfrak{L}_{\tau_{\mathcal{C}}} \setminus L_0$. Załóżmy więc nie wprost, że $d_{\tau_{\mathcal{C}}}(v, u) \leqslant 1$. Wówczas $d_{\tau_{\mathcal{C}}}(v, u) = 1$ oraz $\tau_{\mathcal{C}}(u) = \tau_{\mathcal{C}}(v) + 1$. Korzystając następnie z (4.22) uzyskujemy, że $(\mathcal{C}(u), u) \in A_1 \setminus R_0$, a zatem $(\mathcal{C}(u), u)$ jest $D_{\mathcal{C}}$–łukiem oraz $\mathcal{C}(u) \notin L_{\mathcal{C}}$ z definicji zbioru $L_{\mathcal{C}}$. Wówczas

$D_{\mathcal{C}}$–łuk $(\mathcal{C}(u), u)$ łączy dwa kolejne wierzchołki w $\mathfrak{h}^{\pi(\mathcal{C})}(u)$. Stąd $\tau_{\mathcal{C}}(\mathcal{C}(u))+1 = \tau_{\mathcal{C}}(u)$ oraz $\tau_{\mathcal{C}}(\mathcal{C}(u)) = \tau_{\mathcal{C}}(v)$, co przeczy warunkowi $\mathcal{C}(u) \neq v$. Uzyskana sprzeczność kończy ostatecznie dowód. □

Wykorzystując konstrukcję sortowania topologicznego $\tau_{\mathcal{C}}$ zawartą w dowodzie Tw. 4.18, przedstawimy teraz algorytm wyznaczający $\tau_{\mathcal{C}}$. Dodatkowo, analizując różne scenariusze działania tego algorytmu uzasadnimy, że jego złożoność czasowa to $O(|\mathcal{V}(D)| + |\mathcal{A}(D)|)$. Dla uproszczenia będziemy przyjmować, że wierzchołkom digrafu D zostały przyporządkowane jednoznacznie liczby ze zbioru $\{1, 2, \ldots, |\mathcal{V}(D)|\}$.

Definicja 4.19. *Niech D będzie digrafem, wówczas* listą następników *digrafu D będziemy nazywać tablicę list $L_D[1..|\mathcal{V}(D)|]$, gdzie $L_D[i]$ jest listą następników wierzchołka, do którego została przyporządkowana liczba i.*

Dodatkowo liczbę elementów listy $L_D[i]$ będziemy oznaczać $|L_D[i]|$, a pierwszy element listy $L_D[i]$ symbolem $First(L_D[i])$.

Ustalmy acykliczny digraf D oraz dwa zbiory łuków $A_1 \subseteq A_2 \subseteq \mathcal{A}(D)$ będące instancją problemu $\mathcal{K}.3_{MIZ}$. Dla digrafów $\langle \mathcal{V}(D), \mathcal{A}(D)\rangle$, $\langle \mathcal{V}(D), A_1\rangle$, $\langle \mathcal{V}(D), A_2\rangle$ zakładamy, że zbiory łuków, odpowiednio $\mathcal{A}(D)$, A_1, A_2 są reprezentowane przez listy następników $L_{\mathcal{A}(D)}$, L_{A_1}, L_{A_2}.

Rozważmy najpierw inicjalizację digrafu $\langle \mathcal{V}(D), A_1 \setminus R_0\rangle$ z dowodu Tw. 4.18 (wydruk 4.6). Oczywiście koszt wykonania pętli `for` w wierszu 1. wynosi $O(|\mathcal{V}(D)|)$, na-

```
1  for each v ∈ V(D) do
2     if (|L_{A_2}[v]| > |L_{A_1}[v]|)                          \\ v ∈ L_1
3           or ((|L_{A(D)}[v]| > |L_{A_2}[v]|) and (|L_{A_2}[v]| > 0))   \\ v ∈ L_2
4           or (|L_{A_1}[v]| > 1)                               \\ v ∈ L_3
5     then L_{A_1\R_0}[v]:= nill else L_{A_1\R_0}[v] := L_{A_1}[v];
```

Wydruk 4.6**:** Fragment kodu odpowiadający za inicjalizację listy następników w digrafie $\langle \mathcal{V}(D), A_1 \setminus R_0\rangle$.

tomiast koszt wykonania kopiowania list w wierszu 5. jest ograniczony przez $O(|A_1|)$. Warunek sformułowany w instrukcji warunkowej `if` w wierszu 4. wskazuje jednak, że kopiowanie to jest wykonywane tylko w przypadku $|L_{A_1}[v]| \leqslant 1$, skąd kopiowanie list w wierszu 5. wiąże się z wygenerowaniem list długości co najwyżej 1. Stąd łączna złożoność obliczeniowa kodu z wydruku 4.6 wynosi $O(|\mathcal{V}(D)|)$.

Rozważmy następnie kod z wydruku 4.7 inicjalizujący tablicę $\mathcal{C}[1..|\mathcal{V}(D)|]$, w której wierzchołkom nie należącym do V przyporządkowana została wartość domyślna `nill`. Naturalnie koszt wykonania obu pętli `for` w wierszach 6. i 7. jest równy

```
6  for each v ∈ V(D) do C[v] := nill ;
7  for each v ∈ V(D) do
8     if L_{A_1\R_0}[v] ≠ nill then C[First(L_{A_1\R_0}[v])] := v ;
```

Wydruk 4.7**:** Fragment kodu inicjalizujący tablicę $\mathcal{C}$.

$O(|\mathcal{V}(D)|)$, skąd łączna złożoność obliczeniowa kodu z wydruku 4.7 wynosi $O(|\mathcal{V}(D)|)$. Ograniczenie się jedynie do pierwszego elementu listy $L_{A_1\setminus R_0}[v]$ w wierszu 8. jest uzasadnione tym, że $|\mathcal{N}^{+}_{\langle \mathcal{V}(D), A_1\setminus R_0\rangle}(v)| \leqslant 1$ dla każdego $v \in \mathcal{V}(D)$, gdyż digraf $\langle \mathcal{V}(D), A_1 \setminus R_0\rangle$ jest lasem dendroidów.

Przejdziemy teraz do analizy działania i w dalszej kolejności złożoności procedury z wydruku 4.8. Konstrukcja sortowania topologicznego $\tau_{\mathcal{C}}$ zawarta w dowodzie Tw. 4.18 wykorzystuje graf partycji $\mathcal{G}(D, \pi(\mathcal{C}))$ zależny od funkcji $\mathcal{C}$ oraz sortowania topologicznego σ. W celu uprzystępnienia prezentacji $\tau_{\mathcal{C}}$ rozważymy uproszczony

wariant tej konstrukcji (wydruk 4.8) nie wykorzystujący digrafu $\mathcal{G}(D, \pi(\mathcal{C}))$. Zamiast tego używać będziemy wyłącznie funkcję $\mathcal{C}$, zaś digraf $\mathcal{G}(D, \pi(\mathcal{C}))$ będzie określony implicite.

Algorytm ten będzie modyfikacją rekurencyjnego algorytmu przeszukującego digraf D w głąb przy wykorzystaniu pomocniczej tablicy logicznej $visited[1..|\mathcal{V}(D)|]$, przechowującej informacje o odwiedzonych wierzchołkach.

```
 9  \\ Inicjacja
10  for each v ∈ V(D) do visited[v] :=FALSE ;
11  \\ Właściwe obliczenia
12  procedure TopSort(v);
13  begin
14     if visited[v] =FALSE then
15        if (C[v] ≠nill)and(visited[C[v]] =FALSE) then
16        begin
17           u := v;
18           while C[u] ≠nill do u := C[u];
19           TopSort(u)
20        end
21        else
22        begin
23           visited[v] :=TRUE;
24           if L_{A1\R0}[v] ≠nill then
25           begin
26              if (visited[First(L_{A1\R0}[v]) = FALSE)
27                 and(C[First(L_{A1\R0}[v])] ≠ v) then C[First(L_{A1\R0}[v])] := v;
28              TopSort(First(L_{A1\R0}[v]))
29           end
33           else
31           for each u ∈ L_{A(D)}[v] do TopSort(u);
32           dopisz v na początek τ_C
33        end;
34  end;
35  \\ Wywołanie
36  for each v ∈ V(D) do TopSort(v);
```

Wydruk 4.8: Fragment kodu odpowiadający za wyznaczenie sortowania topologicznego $\tau_{\mathcal{C}}$.

Zauważmy najpierw, że wiersz 14. wraz z fragmentami 22–23, 31–33 oraz pętla `for` w wierszu 36. stanowią „klasyczną część" algorytmu wyznaczającego sortowanie topologiczne digrafu D.

Modyfikacja fragmentu 22–33 związana z istnieniem instrukcji warunkowej `if` w wierszu 24. narzuca jedynie wybór pierwszego wierzchołka do przeszukiwania w głąb zbioru $L_{\mathcal{A}(D)}[v]$. Przypuśćmy bowiem, że $L_{A_1 \setminus R_0}[v] \neq nill$, wówczas wierzchołek $First(L_{A_1 \setminus R_0}[v])$ jest wybierany jako pierwszy ze zbioru $L_{\mathcal{A}(D)}[v]$ do wywołania procedury `TopSort`. Dodatkowo, jeśli $L_{A_1 \setminus R_0}[v] \neq nill$, to z instrukcji warunkowej `if` w wierszu 2. (wydruk 4.6) wynika, że

$$|L_{A_1 \setminus R_0}[v]| = |L_{A_1}[v]| = |L_{A_2}[v]| = |L_{\mathcal{A}(D)}[v]| = 1. \tag{4.25}$$

Stąd $First(L_{A_1 \setminus R_0}[v])$ jest jedynym elementem zbioru $L_{\mathcal{A}(D)}[v]$, a więc przeszukiwanie listy $L_{A_1 \setminus R_0}$ może ograniczać się jedynie do pierwszego elementu tej listy.

Rozważmy teraz scenariusz działania instrukcji w wierszu 15., która ma zagwarantować spoisty zapis drogi Hamiltona $\mathfrak{h}^{\pi(\mathcal{C})}(v)$ dla ostatecznej postaci funkcji wyboru

$\mathcal{C}$. Przyjmujemy, że droga jest zapisana spoiście, jeśli jest $\tau_{\mathcal{C}}$–łańcuchem. Korzystając z (4.25), możemy jedynie uzasadnić, że wywołanie procedury `TopSort(`v`)` w wierszu 36. bez fragmentu 15–21, zapewnia jedynie spoisty zapis drogi $\mathfrak{h}^{\pi(\mathcal{C})}(v)$ począwszy od v, gdzie $v \in \mathcal{V}(D)$. Pokażemy teraz, że fragment 15–21 odpowiada za spoisty zapis całej drogi $\mathfrak{h}^{\pi(\mathcal{C})}(v)$. Ustalmy w tym celu drogę $\mathfrak{h}^{\pi(\mathcal{C})}(v) = v_1 \underset{D_{\mathcal{C}}}{\rightarrow} v_2 \underset{D_{\mathcal{C}}}{\rightarrow} \ldots \underset{D_{\mathcal{C}}}{\rightarrow} v_n$ oraz niech $v = v_i$, dla $1 < i \leqslant n$. Załóżmy dodatkowo, że $visited[\mathcal{C}[v_i]] = FALSE$. Wówczas następstwem wywołania procedury `TopSort(`v_i`)` jest odnalezienie wierzchołka v_1 będącego początkiem drogi $\mathfrak{h}^{\pi(\mathcal{C})}(v)$ przez pętlę `while` w wierszu 18., a następnie rekurencyjne wywołanie `TopSort(`v_1`)` w wierszu 19. Pomijając w tym momencie spełnialność warunku w wierszu 14., działanie procedury `TopSort` sprowadza się wówczas do rekurencyjnego wywoływania tej procedury dla kolejnych wierzchołków $v_2, v_3, \ldots, v_n$ przy jednoczesnym oznaczaniu kolejnych wierzchołków drogi $\mathfrak{h}^{\pi(\mathcal{C})}(v)$ jako odwiedzonych, w szczególności wierzchołka v_i. Zauważmy, że jeśli wierzchołek v_n nie jest ujściem w digrafie $\langle \mathcal{V}(D), A_1 \setminus R_0 \rangle$ oraz jego następnik $First(L_{A_1 \setminus R_0}[v])$ nie został jeszcze odwiedzony, to instrukcja warunkowa `if` w wierszu 26. dokonuje modyfikacji funkcji $\mathcal{C}$, której następstwem jest „wydłużenie" drogi $\mathfrak{h}^{\pi(\mathcal{C})}(v)$ o wierzchołek $First(L_{A_1 \setminus R_0}[v])$, itd. dopóki $L_{A_1 \setminus R_0}[v] \neq nill$ oraz $visited[First(L_{A_1 \setminus R_0}[v])] = FALSE$.

Analizując powyższy scenariusz, wnioskujemy, że następstwem wywołania procedury `TopSort` w wierszu 36. dla nieodwiedzonego wierzchołka $v \in \mathcal{V}(D)$ jest oznaczenie w pierwszym etapie wszystkich wierzchołków drogi $\mathfrak{h}^{\pi(\mathcal{C})}(v)$ jako odwiedzonych. Stąd łatwo można pokazać, stosując rozumowanie indukcyjne, że przy każdym wywołaniu procedury `TopSort` w wierszu 31. lub 36., jeśli został już odwiedzony którykolwiek wierzchołek należący do elementu P partycji $\pi(\mathcal{C})$, to odwiedzony został również każdy wierzchołek z P. Tym samym poczynione założenia w powyższym scenariuszu są uzasadnione.

Analizując kod z wydruku 4.8, dostrzegamy, że „klasyczna część" algorytmu poszukującego sortowania topologicznego digrafu D ma złożoność czasową $O(|\mathcal{V}(D)| + |\mathcal{A}(D)|)$ (odwiedzenie wszystkich wierzchołków w pętli `for` – wiersz 36. oraz ich następników – wiersze 28, 31). W pozostałej części jedynie pętla `while` w wierszu 18. mogłaby zwiększyć tę złożoność, aczkolwiek pętla ta jest wywoływana tylko w trakcie poszukiwania początków poszczególnych dróg postaci $\mathfrak{h}^{\pi(\mathcal{C})}(v)$, przy czym dla każdej takiej drogi jest wywoływana dokładnie raz. Stąd wnioskujemy, że ostateczna złożoność czasowa kodu z wydruku 4.8 wynosi $O(|\mathcal{V}(D)| + |\mathcal{A}(D)|)$.

Podsumowując analizę złożoności poszczególnych listingów, uzyskujemy poniższe twierdzenie.

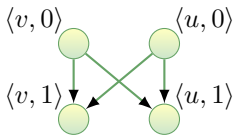
Twierdzenie 4.20. *Problem $\mathcal{K}.3_{MIZ}$ jest rozwiązywalny w czasie $O(|\mathcal{V}(D)|+|\mathcal{A}(D)|)$.*

4.3 Złożoność problemu $\mathcal{K}.3$

W podrozdziale tym udowodnimy NP–zupełność problemu $\mathcal{K}.3$. Dowód tego faktu będzie oczywistą konsekwencją Tw. 4.21 poniżej, które mówi, że podproblem $\mathcal{K}.3$ powstały przez ograniczenie instancji do takich, gdzie $A_2 = A_1 = A$, jest problemem NP-zupełnym

$\mathcal{K}.3'$: Instancja: DAG D, zbiór $A \subseteq \mathcal{A}(D)$, liczba naturalna $0 \leqslant M \leqslant |\mathcal{V}(D)|$.
Pytanie: Czy istnieje sortowanie topologiczne $\tau \in TS(D)$ spełniające zależność:

$$|\{v \in \mathcal{V}(D) : \underset{u \in \mathcal{V}(D)}{\exists} vu \in A \wedge d_\tau(v,u) > 1\}| \leqslant M. \tag{4.26}$$



Rysunek 4.9: Podgraf digrafu G' odpowiadający krawędzi $\{u, v\} \in \mathcal{E}(G)$, który ilustruje konstrukcję z Tw. 4.21.

Twierdzenie 4.21. *$\mathcal{K}.3'$ jest problemem NP–zupełnym.*

Dowód. Pokażemy NP-zupełność problemu $\mathcal{K}.3'$ wskazując redukcję z VC do $\mathcal{K}.3'$. Ustalmy w tym celu instancją problemu VC. Niech więc G będzie nieskierowanym grafem prostym, a K liczbą naturalną, taką że $K \leqslant |\mathcal{V}(G)|$. Pokażemy, że istnieje acykliczny digraf G' oraz zbiór łuków $A' \subseteq \mathcal{A}(G')$, dla których istnieje pokrycie wierzchołkowe grafu G o mocy co najwyżej K wtedy i tylko wtedy, gdy istnieje sortowanie topologiczne $\tau \in TS(G')$, dla którego $|\mathfrak{L}(\tau)| \leqslant K$, gdzie

$$\mathfrak{L}(\tau) := \{v \in \mathcal{V}(G') : \underset{u \in \mathcal{V}(G')}{\exists} vu \in A' \wedge \tau(u) - \tau(v) > 1\}. \tag{4.27}$$

Rozważmy digraf G' zdefiniowany równościami:

$$\begin{array}{rcl} \mathcal{V}(G') & = & \mathcal{V}(G) \times \{0, 1\}, \\ \mathcal{A}(G') & = & \{(\langle v, 0\rangle, \langle v, 1\rangle) : v \in \mathcal{V}(G)\} \cup \{(\langle v, 0\rangle, \langle u, 1\rangle) : \{v, u\} \in \mathcal{E}(G)\} \end{array} \tag{4.28}$$

oraz podzbiór zbioru łuków digrafu G', $A' := \{(\langle v, 0\rangle, \langle v, 1\rangle) : v \in \mathcal{V}(G)\}$. Dla ilustracji tej konstrukcji na np. rys. 4.9 został przedstawiony podgraf digrafu G' odpowiadający krawędzi $\{u, v\} \in \mathcal{E}(G)$. Naturalnie digraf G' oraz zbiór łuków A' może być skonstruowany przy użyciu logarytmicznej ilości pamięci. Dodatkowo $|\mathcal{N}^-_{G'}(\langle v, 0\rangle)| = 0$ oraz $|\mathcal{N}^+_{G'}(\langle v, 1\rangle)| = 0$ dla każdego wierzchołka $v \in \mathcal{V}(G)$, a zatem digraf G jest acykliczny.

Główna idea poniższego dowodu będzie opierać się na obserwacji, że dla dowolnego $\tau \in TS(G')$ i krawędzi $\{u, v\} \in \mathcal{E}(G)$ może zachodzić co najwyżej jedna spośród dwóch równości: $\tau(\langle v, 0\rangle) + 1 = \tau(\langle v, 1\rangle)$, $\tau(\langle u, 0\rangle) + 1 = \tau(\langle u, 1\rangle)$. Mamy z różnowartościowości τ, że $\tau(\langle v, 0\rangle) > \tau(\langle u, 0\rangle)$ lub $\tau(\langle v, 0\rangle) < \tau(\langle u, 0\rangle)$. Ze względu na symetrię sytuacji możemy bez straty ogólności założyć, że $\tau(\langle v, 0\rangle) > \tau(\langle u, 0\rangle)$. Skoro $\tau(\langle v, 0\rangle) + 1 = \tau(\langle v, 1\rangle)$ i $\langle v, 1\rangle \neq \langle u, 0\rangle$, to dostajemy też $\tau(\langle v, 1\rangle) > \tau(\langle u, 0\rangle)$. To jest sprzeczne ze zgodnością τ ze zbiorem łuków, implikującą $\tau(\langle u, 0\rangle) > \tau(\langle v, 1\rangle)$. Stąd wybór co najmniej jednego wierzchołka z dowolnej krawędzi $\{v, u\} \in \mathcal{E}(G)$ do pokrycia wierzchołkowego grafu G może być wyrażony poprzez wybór co najmniej jednego spośród wierzchołków $\langle v, 0\rangle$, $\langle u, 0\rangle$ do zbioru $\mathfrak{L}(\tau)$.

Pokażemy teraz, że dla zbioru V będącego pokryciem wierzchołkowym G o mocy co najmniej K, istnieje $\sigma_v \in TS(G')$, takie że $|\mathfrak{L}(\sigma_V)| \leqslant |V| \leqslant K$. Wprowadźmy oznaczenie na partycję zbioru wierzchołków $\mathcal{V}(G')$ wyznaczoną przez zbiór V:

$$\pi(V) = \{\{\langle v, 0\rangle\} : v \in V\} \cup \{\{\langle v, 1\rangle\} : v \in V\} \cup \{\{\langle v, 0\rangle, \langle v, 1\rangle\} : v \in \mathcal{V}(G) \setminus V\}. \tag{4.29}$$

Zauważmy, że jedynymi zbiorami w $\pi(\mathcal{V})$, które jako wierzchołki digrafu $\mathcal{G}(G', \pi(\mathcal{V}))$ mogą posiadać równocześnie niepusty zbiór następników i poprzedników, są zbiory dwuelementowe Rzeczywiście, zbiory jednoelementowe postaci $\{\langle v, 0\rangle\}$ nie posiadają poprzedników, gdyż nie posiadają ich wierzchołki postaci $\langle v, 0\rangle$, gdzie $v \in V$. Analogicznie zbiory postaci $\{\langle v, 1\rangle\}$ nie posiadają następników, gdyż nie posiadają ich

wierzchołki postaci $\langle v, 1\rangle$. Dodatkowo z faktu, że V jest pokryciem wierzchołkowym wynika, że wierzchołki będące zbiorami dwuelementowymi są niepołączalne w $\mathcal{G}(G', \pi(\mathcal{V}))$, skąd $\mathcal{G}(G', \pi(\mathcal{V}))$ jest digrafem acyklicznym. Wybierzmy więc $\tau_V \in TS(\mathcal{G}(G', \pi(V)))$ oraz określmy funkcję $\sigma_V : \mathcal{V}(G') \to |\{1, 2, \ldots, \mathcal{V}(G')\}|$ daną wzorem:

$$\sigma_V(\langle v, i\rangle) = \begin{cases} 1 + \sum\limits_{R \in \pi(V) : \tau_V(R) < \tau_V(P)} |R| & \text{dla} \quad |P| = 1, \\ 1 + i + \sum\limits_{R \in \pi(V) : \tau_V(R) < \tau_V(P)} |R| & \text{dla} \quad |P| = 2, \end{cases} \tag{4.30}$$

gdzie P jest jedynym elementem z $\pi(V)$, do którego należy $\langle v, i\rangle$. Zauważmy, że funkcja σ_V sortuje wierzchołki występujące w elementach partycji $\pi(V)$, a następnie scala posortowane fragmenty według τ. Oczywiście takie uporządkowanie jest sortowaniem topologicznym digrafu G'. W celu zakończenia dowodu pierwszej implikacji wystarczy więc zauważyć, że $\mathfrak{L}(\sigma_V) \subseteq \{\langle v, 0\rangle : v \in V\}$, skąd ostatecznie $|\mathfrak{L}(\sigma_V)| \leqslant |V| \leqslant K$. Rozważmy w tym celu wierzchołek $w \in \mathfrak{L}(\sigma_V)$ oraz załóżmy nie wprost, że $w \notin \{\langle v, 0\rangle : v \in V\}$. Z $w \in \mathfrak{L}(\sigma_V)$ i (4.27) wynika, że istnieje $r \in \mathcal{V}(G')$, dla którego $(w, r) \in A'$ oraz $\sigma_V(r) - \sigma_V(w) > 1$. Z określenia zbioru A' wynika, że istnieje wierzchołek $u \in \mathcal{V}(G)$, dla którego $w = \langle u, 0\rangle$, $r = \langle u, 1\rangle$. Stąd $u \in \mathcal{V}(G) \setminus V$, gdyż $w \notin \{\langle v, 0\rangle : v \in V\}$. Z określenia σ_V stwierdzamy wówczas, że $\sigma_V(\langle u, 1\rangle) + 1 = \sigma_V(\langle u, 0\rangle)$, co przeczy nierówności $\sigma_V(r) - \sigma_V(w) > 1$, ostatecznie kończąc dowód.

Pokażemy teraz drugą implikację konieczną dla stwierdzenia redukcji. Weźmy dowolne sortowanie topologiczne $\sigma \in TS(G')$, dla którego $|\mathfrak{L}(\sigma)| \leqslant K$ oraz wprowadźmy oznaczenie $V_\sigma := \{v \in V : \langle v, 0\rangle \in \mathfrak{L}(\sigma)\}$. Oczywiście $|V_\sigma| \leqslant |\mathfrak{L}(\sigma)| \leqslant K$, skąd w celu zakończenia dowodu twierdzenia wystarczy pokazać, że V_σ jest pokryciem wierzchołkowym G. Załóżmy nie wprost, że istnieje krawędź $\{v, u\} \in \mathcal{E}(G)$, dla której $\{v, u\} \cap V_\sigma = \emptyset$. Zauważmy, że nierówność $\sigma(\langle v, 1\rangle) - \sigma(\langle v, 0\rangle) > 1$ w konkluzji z przynależnością $(\langle v, 0\rangle, \langle v, 1\rangle) \in A'$, wynikającą z określenia zbioru A', implikuje, że $\langle v, 0\rangle \in \mathfrak{L}(\sigma)$, a to z kolei daje $v \in V_\sigma$, co jest sprzeczne z założeniem $\{v, u\} \cap V_\sigma = \emptyset$. Stąd $\sigma(\langle v, 1\rangle) - \sigma(\langle v, 0\rangle) \leqslant 1$. Analogicznie $\sigma(\langle u, 1\rangle) - \sigma(\langle u, 0\rangle) \leqslant 1$. Wykorzystując następnie fakt, że $\sigma \in TS(G')$ uzyskujemy, że $\sigma(\langle v, 0\rangle), \sigma(\langle u, 0\rangle) < \sigma(\langle v, 1\rangle), \sigma(\langle u, 1\rangle)$. Dodatkowo nierówności te są określone miedzy liczbami naturalnymi, skąd $\sigma(\langle v, 0\rangle) = \sigma(\langle u, 0\rangle)$, $\sigma(\langle v, 1\rangle) = \sigma(\langle v, 1\rangle)$ oraz $v = u$, co przeczy założeniu, że graf G jest prosty. □

Analizując redukcję skonstruowaną w dowodzie twierdzenia Tw. 4.21 uzyskujemy poniższy wniosek.

Wniosek 4.22. *Problem $\mathcal{K}.3'$ zachowuje NP–zupełność w rodzinie acyklicznych digrafów, w których wyróżnione podzbiory zbioru łuków nie zawierają łuków sąsiednich.*

Wykorzystując fakt, że $\mathcal{K}.3$ jest podproblemem $\mathcal{K}.3'$, uzyskujemy poniższe twierdzenie.

Twierdzenie 4.23. *$\mathcal{K}.3$ jest problemem NP–zupełnym.*

4.4 Złożoność problemu $\mathcal{K}.4$

W trakcie formalizacji czwartej metody optymalizacyjnej w sekcji 3.2.7 rozważana była funkcja wagi, która umożliwiała m.in. nadanie różnych wag dla poszczególnych rodzajów łuków w grafie dowodu. Analiza struktury konstruktywnych grafów dowodu w ogólnym przypadku wymaga bowiem posługiwania się kilkoma rodzajami łuków.

Jednak w wyniku Tw. 2.18 możliwe jest ograniczenie rozważań do konstruktywnych grafów dowodów nieposiadających łuków porządkujących. Wykazaliśmy bowiem, że dla każdego acyklicznego digrafu D istnieje konstruktywny abstrakcyjny graf dowodu zawierający rozumowanie pierwotne $\mathfrak{D}$, dla którego $D = \mathfrak{D}$. Ograniczając następnie instancję problemu $\mathcal{K}.4$ do przypadku, w którym każdy łuk jest łukiem referencyjnym z wagą równą 1, uzyskujemy sformułowanie $\mathcal{K}.4$ w poniższej postaci.

$\mathcal{K}.4'$: Instancja: DAG D, liczba naturalna $0 \leqslant M \leqslant \binom{|\mathcal{V}(D)|+1}{3}$.

Pytanie: Czy istnieje sortowanie topologiczne $\tau \in TS(D)$ spełniające zależność:

$$\sum_{vu \in \mathcal{A}(D)} d_\tau(v,u) \leqslant M. \tag{4.31}$$

Problem ten jest jednak znanym jako NP–zupełny problem grafowy: *Minimalne Liniowe Uporządkowanie Grafu Skierowanego* (ang. *Directed Optimal Linear Arrangement*, GT43 [2, 25]), skąd problem $\mathcal{K}.4$, jako uogólnienie problemu GT43, jest również problemem NP–zupełnym.

4.5 Złożoność problemu $\mathcal{K}.5$

W podrozdziale tym udowodnimy NP–zupełność problemu $\mathcal{K}.5$, wykorzystując wniosek uzyskany w sekcji 4.3. Pokażemy bowiem, że w przypadku rodziny digrafów określonej we Wn. 4.22 problemy $\mathcal{K}.3'$ oraz $\mathcal{K}.5$ są równoważne, jeśli przyjąć $M_5 = |A| - M_3$, a za p dowolną liczbę rzeczywistą taką, że $\frac{1}{3} < p \leqslant 1$, gdzie M_3 oraz M_5 są liczbami naturalnymi występującymi odpowiednio w instancji problemów $\mathcal{K}.3'$ oraz $\mathcal{K}.5$.

Niech D będzie acyklicznym digrafem z wyróżnionym zbiorem łuków $A \subseteq \mathcal{A}(D)$ Przypomnijmy, że przez gęstość zbioru $V \subseteq \mathcal{V}(D)$ w domknięciu zwrotno-przechodnim zbioru łuków A (zob. Def. 1.4 na str. 12) będziemy rozumieć liczbę

$$\rho_A(V) := \frac{|\{\{v,u\} : v,u \in V \wedge v \neq u \wedge (v \underset{A}{\rightarrow}^* u \vee u \underset{A}{\rightarrow}^* v)\}|}{\binom{|V|}{2}}. \tag{4.32}$$

Natomiast zbiór V jest τ–spoisty (zob. Def. 1.10 na str. 13), jeśli

$$\underset{i \in \mathbb{N}}{\exists} \underset{v \in V}{\forall} i \leqslant \tau(v) \leqslant i + |V| - 1, \tag{4.33}$$

gdzie $\tau \in TS(D)$.

Twierdzenie 4.24. *Niech D będzie acyklicznym digrafem, A zbiorem łuków nie zawierającym łuków sąsiednich (zob. definicję na str. 12), M liczbą naturalną $0 \leqslant M \leqslant |\mathcal{V}(D)|$, p liczbą rzeczywistą $\frac{1}{3} < p \leqslant 1$, gdzie $A \subseteq \mathcal{A}(D)$. Wówczas sortowanie topologiczne $\tau \in TS(D)$ spełnia zależność:*

$$|\{v \in \mathcal{V}(D) : \underset{u \in \mathcal{V}(D)}{\exists} vu \in A \wedge d_\tau(v,u) > 1\}| \leqslant M \tag{4.34}$$

wtedy i tylko wtedy, gdy

$$|\{V \subseteq \mathcal{V}(D) : |V| \geqslant 2 \wedge \rho_A(V) \geqslant p \wedge V \textit{ jest } \tau\textit{–spoisty}\}| \geqslant |A| - M. \tag{4.35}$$

Dowód. Niech D, A, M, p spełniają założenia twierdzenia. Ustalmy dowolne sortowanie topologiczne $\tau \in TS(D)$. Naturalnie zbiór A można przedstawić w postaci unii $A = \{vu \in A : d_\tau(v,u) = 1\} \cup \{vu \in A : d_\tau(v,u) > 1\}$. Z założenia zbiór A nie zawiera łuków sąsiednich, a więc $|\mathcal{N}_A^+(v)| \leqslant 1$, co oznacza m.in., że każdy A–łuk jest jednoznacznie wyznaczony przez swój początek. Stąd zbiory $\{vu \in A : d_\tau(v,u) > 1\}$, $\{v \in \mathcal{V}(D) : \underset{u \in \mathcal{V}(D)}{\exists} vu \in A \wedge d_\tau(v,u) > 1\}$ są równoliczne, a więc warunek (4.34) możemy przedstawić w równoważnej postaci:

$$|\{vu \in A : d_\tau(v,u) = 1\}| \geqslant |A| - M. \tag{4.36}$$

W celu zakończenia dowodu wystarczy więc jedynie pokazać, że zbiory $A' := \{vu \in A : d_\tau(v,u) = 1\}$, $V' := \{V \subseteq \mathcal{V}(D) : |V| \geqslant 2 \wedge \rho_A(V) \geqslant p \wedge V \text{ jest } \tau\text{–spoisty}\}$ są równoliczne. Rozważmy w tym celu przekształcenie $h : \mathcal{A}(D) \to 2^{\mathcal{V}(D)}$, które każdemu łukowi vu przyporządkowuje zbiór $\{v,u\}$. Oczywiście h jest funkcją różnowartościową, gdyż digraf D jest acykliczny. Zauważmy następnie, że dla każdego łuku $v_1u_1 \in A'$ zachodzi równość $\rho_A(h(v_1u_1)) = 1$ oraz $h(vu)$ jest zbiorem τ–spoistym. Stąd $h(A') \subseteq V'$. W celu uzasadnienie inkluzji odwrotnej ustalmy dowolny zbiór $W \in V'$. Ponieważ zbiór A nie zawiera łuków sąsiadujących, więc w szczególności $\rho_A(W) \leqslant \frac{1}{2n-1}$, jeśli $|W| = 2n$, oraz $\rho_A(W) \leqslant \frac{1}{2n+1}$, jeśli $|W| = 2n+1$. Z założenia $\frac{1}{3} < p \leqslant \rho_A(W)$ oraz $|W| \geqslant 2$. Mamy tożsamości $\frac{1}{3} < p \leqslant \frac{1}{|W|-1}$ dla $|W|$ parzystego, $\frac{1}{3} < p \leqslant \frac{1}{|W|}$ dla $|W|$ nieparzystego. Zatem $|W| < 4$ dla $|W|$ parzystego oraz $|W| < 3$ dla $|W|$ nieparzystego. Co w związku z $|W| \geqslant 2$ daje, że $|W| = 2$. To zaś prowadzi do $\rho_A(W) = 1$ oraz istnienia A-łuku $v''u''$, dla którego $W = \{v'', u''\}$. Z definicji τ-spoistości (Def. 1.10) istnieje liczba i taka, że $i \leqslant \tau(v''), \tau(u'') \leqslant i+1$. Stąd $\tau(u'') = \tau(v'') + 1$, co oznacza, że $v''u'' \in A'$ i $h(v''u'') = W$, co ostatecznie kończy dowód. □

Z Tw. 4.24 oraz Wn. 4.22 uzyskujemy natychmiast poniższe twierdzenie.

Twierdzenie 4.25. *$\mathcal{K}.5$ jest problemem NP–zupełnym.*

Wnioski końcowe

Pierwszym problemem badawczym rozprawy było pytanie, *czy i w jakim stopniu możliwe jest posługiwanie się abstrakcyjnym modelem grafu dowodu do analizy metod uczytelniania istniejących rozumowań formalnych zgromadzonych w bazie MML.*

Odnosząc się do pytania sformułowanego w pierwszym problemie badawczym, należy rozważyć dwa jego aspekty. Po pierwsze, należy zweryfikować, czy linearyzacja zaproponowanego grafu dowodu nie prowadzi do powstania błędów w skrypcie dowodowym. Po drugie, należy sprawdzić, czy obrane pojęcie grafu konstruktywne. Przy czym przyjmujemy, że graf dowodu jest konstruktywny jeśli istnieje dowód w języku Mizar o strukturze opisanej tym grafem.

W związku z pierwszym aspektem naszego problemu przeprowadzona została w ramach niniejszej rozprawy szczegółowa analiza składni systemu Mizar, w wyniku której wykazaliśmy poprawność i pełność przyjętego modelu grafowego. Dodatkowo testy empiryczne przeprowadzone nad wstępną poprawą budowy ponad 30 tys. rozumowań z bazy MML nie wygenerowały błędów w trakcie modyfikacji skryptów dowodowych. Stąd możemy wnioskować, że modyfikacja sposobu linearyzacji poszczególnych rozumowań pierwotnych przy uwzględnieniu jedynie informacji zgromadzonej w grafie dowodu, nie prowadzi do powstania błędów w skryptach dowodowych.

Odnosząc się do drugiego aspektu problemu badawczego, stwierdzamy, że nie wszystkie grafy dowodów dadzą sie zrealizować konstruktywnie. Jednak można wskazać podrodzinę *konstruktywnych abstrakcyjnych grafów dowodu.* Rodzina ta jest na tyle bogata, że zawiera wszystkie grafy spośród tych, które były wykorzystywane przy badaniu złożoności rozważanych problemów optymalizacyjnych. Dodatkowo każdy skrypt po niewielkiej modyfikacji prowadzi do powstania grafu ze wspomnianej podrodziny.

Konkludując, w odniesieniu do pierwszego problemu badawczego stwierdzamy, że graf dowodu jest dobrym narzędziem umożliwiającym wierne odzwierciedlenie struktury rozumowań zapisanych w języku Mizar. Jednocześnie wprowadzenie tego pojęcia umożliwia uniezależnienie prowadzonych rozważań od tego systemu.

Drugi problem badawczy został sformułowany w formie pytania, *na ile proponowane metody uczytelniania są efektywne czasowo, a co za tym idzie, na ile są stosowalne w procesie automatycznej poprawy czytelności sformalizowanych rozumowań zapisanych w systemie naturalnej dedukcji.*

Odnosząc się do pytania sformułowanego w drugim problemie badawczym, należy stwierdzić, że została zbadana złożoność problemu optymalizacji pięciu metod poprawy czytelności. Związane są one z następującymi własnościami dowodów:

- $\mathcal{K}.1$ Liczba kroków, z których każdy w swoim uzasadnieniu odwołuje się m.in. do przesłanki sformułowanej w bezpośrednio poprzedzającym kroku dowodu powinna być maksymalna.
- $\mathcal{K}.2$ Liczba odwołań do przesłanek w obrębie poszczególnych liniowych fragmentów dowodu powinna być maksymalna.

$\mathcal{K}.3$ Liczba etykiet, które należy wprowadzić w dowodzie, w celu umożliwienia odwoływania się do daleko położonych przesłanek, jakie nie mogą być przekazane do uzasadnienia za pomocą konstrukcji `then`, powinna być minimalna.

$\mathcal{K}.4$ Suma odległości po wszystkich odwołaniach między krokami, które odwołują się do przesłanek, a krokami, w których te przesłanki zostały uzasadnione, powinna być minimalna.

$\mathcal{K}.5$ Liczba fragmentów rozumowania zapisanych spoiście w dowodzie, w których przepływ informacji jest dostatecznie gęsty, powinna być maksymalna.

W wyniku przeprowadzonych badań została udowodniona NP–zupełność czterech spośród pięciu problemów decyzyjnych odpowiadających ww. własnościom dowodu. Natomiast wielomianowa złożoność problemu $\mathcal{K}.3$ jest jedynie następstwem syntaktycznych ograniczeń na stosowanie konstrukcji `then` w systemie Mizar. Problem ten przy pominięciu tych ograniczeń, staje się bowiem problemem NP–zupełnym, co zostało udowodnione w rozprawie.

Przedstawione wyniki badań utwierdzają zatem w przekonaniu, że poprawa czytelności rozumowań zapisanych w języku Mizar wiąże się w większości przypadków z rozwiązywaniem NP-zupełnych optymalizacyjnych problemów grafowych. Stąd wnioskujemy, że programy, które mogłyby realizować poprawę czytelności w zadowalającym czasie mogą co najwyżej aproksymować optymalne wartości wskaźników. Dodatkowo przeprowadzone wstępne badania empiryczne pokazały również, że zbiory sortowań topologicznych, w których zaproponowane w rozprawie wskaźniki posiadają wartości optymalne, w ogólnym przypadku nie mają elementów wspólnych. Stąd wnioskujemy, że programy uczytelniające skrypty dowodowe mogą działać jedynie przy ustalonej hierarchii wartości optymalizowanych wskaźników.

Wstępne badania nad metodami poprawy czytelności wykorzystującymi wyodrębnianie fragmentów rozumowania w postaci lematów umożliwiły uzupełnienie odpowiedzi na pytanie sformułowane w drugim problemie badawczym. Wykazały one bowiem, że stosując algorytm o złożoności wielomianowej, możliwe jest wyodrębnianie z rozumowania dowolnych paczek przy jednoczesnym zachowaniu poprawności modyfikowanych skryptów dowodowych [72]. Przeprowadzone badania pozwoliły na uzyskanie istotnego wyniku, który wskazuje kierunek dalszych badań nad narzędziami służącymi do automatycznego odnajdywania i wyodrębniania fragmentów rozumowania. Wstępne wyniki badań sugerowały bowiem, iż stwierdzenie opisujące rozumowanie zawarte w paczce powinno mieć postać implikacji, której

(*i*) poprzednikiem jest koniunkcja przesłanek, które jednocześnie są zlokalizowane poza obszarem paczki oraz są wykorzystane w krokach paczki,

(*ii*) a następnikiem jest koniunkcja stwierdzeń uzasadnionych w paczce, które są wykorzystywane w dalszej części rozumowania poza paczką.

Jednak w przypadku wyodrębniana paczki niedomkniętej na prowadzenie dróg skierowanych, następstwem takiego uproszczenia są cykle skierowane, które powstają w zmodyfikowanym grafie dowodu. Natomiast ograniczenie możliwości wyodrębniania jedynie do przypadku paczek domkniętych na prowadzenie dróg skierowanych zawężałoby przestrzeń poszukiwań do acyklicznych partycji rozumowania. Ograniczenie to sprowadzało więc problem podziału rozumowania do rozwiązywania znanego NP-zupełnego problemu *Partycji Acyklicznej* (ang. *Acyclic Partition*, zob. ND15 [25, 46]). Stworzenie metody wyodrębniania nawet niedomkniętych paczek stanowi zatem istotny krok w badaniach nad poprawą czytelności wykorzystujących wyodrębnianie paczek. Niestety wyodrębnienie takich paczek jest bardziej skomplikowane

niż w przypadku paczek domkniętych oraz wiąże się z koniecznością dodawania dodatkowych kroków i powielania fragmentów rozumowania. Stąd możemy wnioskować, że algorytm poszukujący najbardziej optymalnej partycji rozumowania na paczki nie będzie zawężał swoich poszukiwań jedynie do partycji acyklicznych. Będzie on jednak jednocześnie musiał uwzględniać liczbę „wychodzących" dróg skierowanych w paczkach, które odpowiadają za rozbudowywanie rozumowania.

Sformułowana odpowiedź na pytanie postawione w drugim problemie badawczym potwierdza więc pierwszą część sformułowanej na wstępie hipotezy badawczej. Natomiast druga część hipotezy została potwierdzona za pomocą empirycznych badań przeprowadzonych w ramach „wstępnej" poprawy budowy rozumowań. Zaproponowany tam algorytm poszukujący linearyzacji rozumowania przy ustalonej hierarchii ważności wyznaczników czytelności umożliwił zestandaryzowanie rozumowań zgromadzonych w bazie MML. Wykorzystywany przy tym algorytm zachłanny nie prowadzi jednak w ogólnym przypadku do odnajdywania rozwiązań optymalnych. Analiza zmodyfikowanych skryptów dowodowych wykazała jednak, że zastosowanie algorytmu dokonującego lokalnej optymalizacji okazało się przydatnym narzędziem. Umożliwia ono bowiem odnajdywanie i spójny zapis w zlinearyzowanym rozumowaniu gęstych zbiorów wierzchołków występujących w grafie dowodu. Do efektywnej poprawy czytelności i jasności zgromadzonych rozważań niezbędne jest jednak prowadzenie dalszych badań nad algorytmami dokonującymi optymalizacji wielokryterialnej oraz standardami czytelności dowodów.

Dodatek

```
<Theorem> = [<Label-Identifier>:] <Sentence> <Justification> ;,
  <Justification> =
    <Simple-Justification> | {@proof|proof} <Reasoning> end,
  <Simple-Justification> =
    by {<Reference> | <Reference>{,<Reference>}*} |
    from {<Label-Identifier> | <File-Name>: sch<Numeral>}
      [<Label-Identifier> | <Label-Identifier>{,<Label-Identifier>}*],
  <Reference> =
    <Label-Identifier> | <File-Name>: {<Numeral> |def <Numeral>},
  <Reasoning> =
    {<Reasoning-Step>}*
      [per cases <Simple-Justification> ;
        { suppose {<Proposition> | <Conditions>} ; <Reasoning> end ;}],
  <Reasoning-Step> = <AuxiliaryStep> | <Skeleton-Step>,
  <AuxiliaryStep> = [then] <Statement> | <Private-Definition>,
  <Statement> =
    <Compact-Statement> |
    consider <Qualified-Variables> such <Conditions> <Simple-Justification> ;|
    reconsider {<Variable-Identifier> = <Term-Expression> | <Variable-Identifier>}
      as <Type-Expression> <Simple-Justification> ;,
  <Compact-Statement> = <Proposition> <Justification> ;,
  <Private-Definition> = <Variable-Identifier> = <Term-Expression> ; |
    deffunc <Variable-Identifier> ( [<Type-Expression-List>] ) = <Term-Expression> |
    defpred <Variable-Identifier> [ [<Type-Expression-List>] ]
      means <Formula-Expression>,
  <Skeleton-Step> =
    <Generalization> | <Assumption> | <Conclusion> | <Exemplification>,
  <Generalization> = let <Qualified-Variables> [ such <Conditions> ] ;,
  <Assumption> = assume { <Proposition> | <Conditions> } ;,
  <Conclusion> = thus [ then ] <Proposition> <Justification> ;,
  <Exemplification> = take {<Term-Expression> |
    <Variable-Identifier> = <Term-Expression>} ;,
  <Conditions> = that { <Proposition> | <Proposition> { and <Proposition> }* } ,
  <Proposition> = [<Label-Identifier>:] <Formula-Expression>,
  <Qualified-Variables> = <Qualified-Variable> |
    <Qualified-Variable> {, <Qualified-Variable> }*,
  <Qualified-Variable> = <Variable-Identifier-List> { be|being } <Type-Expression>.
```

Tablica D.1: Uproszczona składnia kroków rozumowania w systemie Mizar.

```
Lm3: for k being Element of NAT st k<25 holds
       for n being Element of NAT st 1<n & n*n<=k & n is prime holds
          n=2 or n=3;

theorem Th9:
  for i, j, k, l being Nat st i=j*k+l & l<j & 0<l holds
     not j divides i;

theorem Th14:
  for p being Nat holds
    p is prime
  iff
    p>1 & for n being Element of NAT holds
      1<n & n*n<=p & n is prime implies not n divides p;
```

Wydruk D.1: Stwierdzenia związane z etykietami wykorzystanymi na wydruku 2.4

Bibliografia

[1] The Coq Development Team, *The Coq Proof Assistant Reference Manual – Version V8.1*, 2007, dostępne z: http://coq.inria.fr/V8.1beta/refman/.

[2] D. Adolphson, T. C. Hu, *Optimal Linear Ordering*, SIAM Journal on Applied Mathematics, t. 25, nr 3, s. 403–423, 1975.

[3] K. K. Aggarwal, Y. Singh, J. K. Chhabra, *An Integrated Measure of Software Maintainability*, w: Reliability and Maintainability Symposium, s. 235–241, 2002, doi: 10.1109/RAMS.2002.981648.

[4] J. Alama, *Metadata for a Wiki of Formalized Mathematics*, w: *MathWikis-2011-Proc. of the ITP 2011 Workshop on Mathematical Wikis*, s. 2–5, 2011, dostępne z: http://ceur-ws.org/Vol-767/paper-02.pdf.

[5] J. Alama, Escape to Mizar for ATPs, *The Computing Research Repository*, 2012, dostępne z: http://www.newton.ac.uk/preprints/NI12025.pdf.

[6] J. Alama, K. Brink, L. Mamane, J. Urban, *Large Formal Wikis: Issues and Solutions*, w: J. H. Davenport i in. (red.), Proc. of 18th Symposium, Calculemus 2011, and 10th International Conference Mathematical Knowledge Management 2011, LNCS, t. 6824, s. 133–148, Springer-Verlag, 2011, doi: 10.1007/978-3-642-22673-1_10.

[7] G. Bancerek, P. Carlson, *A Synthesis of the Procedural and Declarative Styles of Interactive Theorem Proving*, Logic in Computer Science, t. 8, nr 1, s. 1–26, 2012, doi: 10.2168/LMCS-8(1:30)2012.

[8] G. Bancerek, P. Rudnicki, Information Retrieval in MML, w: A. Asperti i in. (red.), Proc. of Second International Conference Mathematical Knowledge Management 2003, LNCS, t. 2594, s. 119–131, Springer, Heidelberg, 2003, doi: 10.1007/3-540-36469-2_10.

[9] O. Behaghel, *Beziehungen zwischen Umfang und Reihenfolge von Satzgliedern*, Indogermanische Forschungen, t. 25, s. 110–142, 1909.

[10] N. Beldiceanu, X. Lorca, *Necessary Condition for Path Partitioning Constraints*, w: P. V. Hentenryck i in. (red.), Proc. of CP-AI-OR'07, LNCS, t. 4510, s. 141–154, Springer-Verlag, 2007, doi: 10.1007/978-3-540-72397-4_11.

[11] J. C. Blanchette, L. Bulwahn, T. Nipkow, *Automatic Proof and Disproof in Isabelle/HOL*, w: Proc. of the 8th International Conference on Frontiers of Combining Systems, FroCoS'11, LNCS, t. 6989, s. 12–27, Springer-Verlag, 2011, doi: 10.1007/978-3-642-24364-6_2.

[12] E. Bonarska, *An Introduction to PC Mizar*, Fondation Philippe le Hodey, 1990.

[13] M. Borowiecki, P. Mihók, *Hereditary Properties of Graphs*, w: V. R. Kulli (red.), Advances in Graph Theory, s. 42–69, Vishwa International Publishers, 1991.

[14] D. E. Broadbend, *Perception and Communication*, Pergamon Press, 1958.

[15] D. E. Broadbend, *Decision and Stress*, London Academic Press, 1971.

[16] N. G. de Bruijn, *A Survey of the Project Automath*, w: J. P. Seldin i in. (edt.), To H.B. Curry: Essays in Combinatory Logic, Lambda Calculus, and Formalism, s. 589–606, Academic Press, 1980.

[17] P. Corbineau, *A Declarative Language for the Coq Proof Assistant*, w: Proc. of the 2007 International Conference on Types for Proofs and Programs, LNCS, t. 4941, s. 69–84, 2007, doi: 10.1007/978-3-540-68103-8_5.

[18] N. Cowan, *Attention and Memory: An Integrated Framework*, Oxford University Press, 1998.

[19] N. Cowan, *The Magical Number 4 in Short-term Memory: A Reconsideration of Mental Storage Capacity*, Behavioral and Brain Sciences, t. 24, nr 1, s. 87–114, 2001, doi: 10.1017/S0140525X01003922.

[20] M. Davis, *Obvious Logical Inferences*, w: Proc. of the 20th International Joint Conference on Artificial Intelligence, s. 530–531, 1981.

[21] I. Dinur, S. Safra, *On the Hardness of Approximating Minimum Vertex Cover*, Annals of Mathematics, t. 162, nr 1, s. 439–485, 2005, doi: 10.4007/annals.2005.162.439.

[22] R. Engelking, *General Topology*, PWN, 1977.

[23] K. A. Ericsson, *Memory skill*, Canadian Journal of Psychology, 39(2):188–231, 1985.

[24] K. A. Ericsson, *Analysis of Memory Performance in Terms of Memory Skill*, Advances in the psychology of human intelligence, t. 4, Hillsdale, NJ: Lawrence Erlbaum Associates Ins., 1988.

[25] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, A Series of Books in the Mathematical Science, 1979.

[26] M. R. Garey, D. S. Johnson, *The Rectilinear Steiner Tree Problem is NP-Complete*, SIAM Journal on Applied Mathematics, t. 32, nr 4, s. 826–834, 1977.

[27] M. R. Garey, D. S. Johnson, L. Stockmeyer, *Some Simplified NP-Complete Problems*, w: Proc. of the 6th annual ACM Symposium on Theory of computing, s. 47–63, 1974.

[28] F. Gavril, *Some NP-complete Problems on Graphs*, w: Proc. 11th Conference on Information Sciences and System, Johns Hopkins University, 1977.

[29] K. Gödel, *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme*, I. Monatshefte für Mathematik und Physik, t. 38, s. 173–198, 1931.

[30] G. Gentzen, *Untersuchungen über das logische Schließen*, Mathematische Zeitschrift, t. 35, nr 1, s. 176–210, 1935.

[31] E. Gibson, *Linguistic complexity: locality of syntactic dependencies*, Cognition, t. 68, s. 1–76, 1998.

[32] M. Giero, F. Wiedijk, *MMode, A Mizar Mode for the proof assistant Coq*, ICIS, Radboud Universiteit Nijmegen, 2004.

[33] A. Grabowski, A. Korniłowicz, A. Naumowicz, *Mizar in a Nutshell*, Journal of Formalized Reasoning, t. 3, nr. 2, s. 153–245, 2010, dostępne z: http://jfr.unibo.it/article/view/1980/1356.

[34] A. Grabowski, C. Schwarzweller, *On Duplication in Mathematical Repositories*, w: D. Hutchison i in. (red.), Intelligent Computer Mathematics, LNCS, t. 6167, s. 300–314, Springer-Verlag, 2010, doi: 10.1007/978-3-642-14128-7_26.

[35] A. Grabowski, Ch. Schwarzweller, *Improving Representation of Knowledge within the Mizar Library*, Studies in Logic, Grammar and Rhetoric, t. 18, nr 31, s. 35–50, 2009, dostępne z: http://logika.uwb.edu.pl/studies/download.php?volid=31&artid=ag.

[36] J. Harrison, *A Mizar Mode for HOL*, w: Proc. of the 9th International Conference on Theorem Proving in Higher Order Logics, LNCS, t. 1125, s. 203–220, Springer-Verlag, 1996, doi: 10.1007/BFb0105406.

[37] J. Harrison, *The HOL Light manual (1.1)*, 2000, dostępne z: http://www.cl.cam.ac.uk/∼jrh13/hol-light/manual-1.1.pdf.

[38] D. Hilbert, P. Bernays, *Die Grundlagen der Mathematik*, Springer, 1968/70.

[39] M. Huth, M. Ryan, *Logic in Computer Science, Modelling and Reasoning about Systems*, Cambridge University Press, 2004.

[40] S. Jaśkowski, *On the Rules of Supposition in Formal Logic*, Studia Logica, 1934.

[41] A. T. Jersild, *Mental set and mental shift*, Archives of Psychology, t. 14, s. 669–679, 1927.

[42] C. Kaliszyk, J. Urban, *PRocH: Proof reconstruction for HOL Light (System description)*, Praca przyjęta na konferencję the 24th International Conference on Automated Deduction, 2013, dostępne z: http://cl-informatik.uibk.ac.at/users/cek/docs/kaliszyk-cade13.pdf.

[43] F. Kamareddine, R. Nederpelt, *A Refinement of de Bruijn's Formal Language of Mathematics*, Journal of Logic, Language and Information, t. 13, nr 3, s. 287–340, 2004, doi: 10.1023/B:JLLI.0000028393.47593.b8.

[44] V. Kann, *On the Approximability of NP-complete Optimization Problems*, rozprawa doktorska, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, 1992, dostępne z: http://www.nada.kth.se/∼viggo/papers/phdthesis.pdf.

[45] R. M. Karp, *Reducibility Among Combinatorial Problems*, Plenum Press New York, Complexity of Computer Computations, s. 85–103, 1972.

[46] B. W. Kernighan, *Optimal Sequential Partitions of Graphs*, Journal of the ACM, t. 18, nr 1, s. 24–40, 1971.

[47] A. J. Ko, B. A. Myers, M. J. Coblenz, Htet Htet Aung, *An Exploratory Study of How Developers Seek, Relate, and Collect Relevant Information during Software Maintenance Tasks*, IEEE Transactions On Software Engineering, t. 32, nr 12, s. 971–988, 2006.

[48] A. Kornilowicz, *Tentative Experiments with Ellipsis in Mizar*, w: J. Jeuring i in. (edt.), Intelligent Computer Mathematics, LNCS, t. 7362, s. 453–457, Springer-Verlag, 2012, doi: 10.1007/978-3-642-31374-5_35.

[49] A. Kornilowicz, *On Rewriting Rules in Mizar*, Journal of Automated Reasoning, t. 50, nr 2, s. 203–201, 2013, doi: 10.1007/s10817-012-9261-6.

[50] K. Korniłowicz, *How to Define Terms in Mizar Effectively*, Studies in Logic, Grammar and Rhetoric, t. 18, nr 31, s. 67–78, 2009, dostępne z: http://logika.uwb.edu.pl/studies/download.php?volid=31&artid=ak.

[51] A. C. Leisenring, *Mathematical Logic and Hilbert's ε-Symbol*, Gordon and Breach, 1969.

[52] R. Levy, *Expectation-based syntactic comprehension*, Cognition, t. 106(2008), s. 1126–1177, 2007, dostępne z: http://idiom.ucsd.edu/∼rlevy/papers/levy-2008-cognition.pdf

[53] H. Loess, *Short-Term Memory and Item Similarity*, Journal of Verbal Learning and Verbal Behavior, t. 7, s. 87–92, 1968.

[54] W. Marciszewski, *A Jaśkowski-Style System of Computer-Assisted Reasoning*, Philosophical Logic in Poland, Kluwer, 1993.

[55] R. Matuszewski, *Preface*, Formalized Mathematics, Université Catholique de Louvain, t. 1, nr 4,s. 623–624, 1990.

[56] R. Matuszewski, P. Rudnicki, *MIZAR: the first 30 years*, Mechanized Mathematics and Its Applications, t. 4, nr 1, s. 3–24, 2005, dostępne z: http://mizar.org/people/romat/MatRud2005.pdf.

[57] R. Milewski, *New Auxiliary Software for MML Database Management*, Mechanized Mathematics and Its Applications, t. 5, nr 2, s 1–10, 2006, dostępne z: http://markun.cs.shinshu-u.ac.jp/mizar/mma.dir/2006/MMA_2006_paper_7_for_web.pdf.

[58] R. Milewski, *Transformations of MML Database's Elements*, w: Proc. of Mathematical Knowledge Management 2006, LNCS, t. 3863, s. 376–388, Springer-Verlag, 2006, doi: 10.1007/11618027_25.

[59] R. Milewski, *Algorithms for Analysis of a System Supporting Formal Deduction*, rozprawa doktorska, Politechnika Białostocka, Białystok, 2008.

[60] R. Milewski, *The Influence of Delocalization on the Results of Eliminating Repetitions of Semantically Equivalent Sentences in the MML Database*, Studies in Logic, Grammar and Rhetoric, t. 18, nr 31, s. 79–88, 2009, dostępne z: http://logika.uwb.edu.pl/studies/download.php?volid=31&artid=rm.

[61] G. A. Miller, *The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information*, Psychological Review, t. 63, s. 81–97, 1956.

[62] A. Naumowicz, *Teaching How to Write a Proof*, w: Proc. of the ETAPS 2008 satellite workshop Formal Methods in Computer Science Education, s. 91–100, 2008.

[63] A. Naumowicz, *Enhanced Processing of Adjectives in Mizar*, Studies in Logic, Grammar and Rhetoric, t. 18, nr 31, s. 89–101, 2009, dostępne z: http://logika.uwb.edu.pl/studies/download.php?volid=31&artid=an.

[64] A. Naumowicz, C. Byliński, *Improving MIZAR Texts with Properties and Requirements*, w: A. Asperti i in. (red.), Proc. of Mathematical Knowledge Management 2004, LNCS, t. 3119, s. 290–301, Springer-Verlag, 2004, doi: 10.1007/978-3-540-27818-4_21.

[65] A. Naumowicz, A. Korniłowicz, *A Brief Overview of Mizar*, w: S. Berghofer i in. (red.), Theorem Proving in Higher Order Logics, LNCS, t. 5674, s. 67–72, Springer-Verlag, 2009, doi: 10.1007/978-3-642-03359-9_5.

[66] F. J. Pelletier, *A Brief History of Natural Deduction*, History and Philosophy of Logic, t. 20, nr 1, s. 1–31, 1999, dostępne z: http://www.sfu.ca/∼jeffpell/papers/NDHistory.pdf.

[67] K. Pąk, *Basic Properties of Determinants of Square Matrices over a Field*, Formalized Mathematics, t. 15, nr 1, s. 17–25, 2007, doi: 10.2478/v10037-007-0003-x.

[68] K. Pąk, *The Nagata-Smirnov Theorem. Part I*, Formalized Mathematics, t. 12, nr 3, s. 341–346, 2004, dostępne z: http://fm.mizar.org/2004-12/pdf12-3/nagata_1.pdf.

[69] K. Pąk, *Algorithms to Improving and Reorganizing Natural Deduction Proofs*, 5th Conference on Technologies of Knowledge Exploration and Representation, Hołny Mejera, 2010.

[70] K. Pąk, *The Algorithms for Improving and Reorganizing Natural Deduction Proofs*, Studies in Logic, Grammar and Rhetoric, t. 22, nr 35, s 95–112, 2010, dostępne z: http://logika.uwb.edu.pl/studies/download.php?volid=35&artid=kp.

[71] K. Pąk, *The Methods of Improving and Reorganizing Natural Deduction Proofs*, Mathematical User-Interfaces, 2010.

[72] K. Pąk, *Methods of Lemma Extraction in Natural Deduction Proofs*, Journal of Automated Reasoning, t. 50, nr 2, s. 217–228, 2013, doi: 10.1007/s10817-012-9267-0.

[73] S. P. Rahul, G. C. Necula, *Proof Optimization Using Lemma Extraction*, UCB/CSD-01-1143, Computer Science Division (EECS), University of California, 2001, dostępne z: http://techreports.lib.berkeley.edu/accessPages/CSD-01-1143.html.

[74] M. Riccardi, *Pocklington's Theorem and Bertrand's Postulate*, Formalized Mathematics, t. 14, nr 2, s. 47–52, 2006, doi: 10.2478/v10037-006-0007-y.

[75] P. Rudnicki, A. Trybulec, *On Equivalents of Well-foundedness – An Experiment in Mizar*, Journal of Automated Reasoning, t. 23, nr 3, s. 197–234, 1999.

[76] P. Rudnicki, A. Trybulec, *Mathematical Knowledge Management in Mizar*, w: Proc. of Third International Conference Mathematical Knowledge Management 2001, Electronic Proc. of MKM 2001, 2001.

[77] P. Rudnicki, *Obvious Inferences*, Journal of Formalized Reasoning, t. 3, nr 4, s. 383–393, 1987.

[78] P. Rudnicki, A. Trybulec, *On the Integrity of a Repository of Formalized Mathematics*, w: A. Asperti i in. (red.), Proc. of Second International Conference Mathematical Knowledge Management 2003, LNCS, t. 2594, s. 132–146, Springer-Verlag, 2003, doi: 10.1007/3-540-36469-2_13.

[79] P. Rudnicki, J. Urban, *Escape to ATP for Mizar*, w: P. Fontaine i in. (red.), PxTP 2011: First International Workshop on Proof eXchange for Theorem Proving, s. 46–59, 2011, dostępne z: http://pxtp2011.loria.fr/paper1.pdf.

[80] E. Snapper, *The Three Crises in Mathematics: Logicism, Intuitionism and Formalism*, Mathematics Magazine, t. 52, nr 4, s. 207–216, 1979.

[81] R. M. Stallman, *EMACS: The Extensible, Customizable Self-Documenting Display Editor*, SIGPLAN Not., t. 16, nr 6, s. 147–156, 1981.

[82] D. Syme, *Three Tactic Theorem Proving*, w: Theorem Proving in Higher Order Logics, LNCS, t. 1690, s. 203–220, Springer-Verlag, 1999, doi: 10.1007/3-540-48256-3_14.

[83] A. Tarski, *Über Unerreaichbare Kardinalzahlen*, Fundamenta Mathematicae, t. 30, s. 167–183, 1938.

[84] A. Trybulec, *Tarski Grothendieck Set Theory*, Formalized Mathematics, t. 1, s. 9–11, 1990.

[85] A. Trybulec, *Some Features of the Mizar Language*, w: Proc. of ESPRIT Workshop, Torino, 1993.

[86] J. Urban, *XML-izing Mizar: Making Semantic Processing and Presentation of MML Easy*, w: Proc. of Mathematical Knowledge Management 2005, LNCS, t. 3863, s. 346–360, Springer-Verlag, 2006, doi: 10.1007/11618027_23.

[87] J. Urban, *MizarMode - An Integrated Proof Assistance Tool for the Mizar Way of Formalizing Mathematics*, Journal of Applied Logic, t. 4, nr 4, s. 414–427, 2006, dostępne z: http://ktiml.mff.cuni.cz/∼Eurban/mizmode.ps.

[88] J. Urban, *MoMM - Fast Interreduction and Retrieval in Large Libraries of Formalized Mathematics*, International Journal on Artificial Intelligence Tools, t. 15, nr 1, s. 109–130, 2006, dostępne z: http://ktiml.mff.cuni.cz/∼urban/MoMM/momm.ps.

[89] J. Urban, *Combining Mizar and TPTP Semantic Presentation and Verification Tools*, Studies in Logic, Grammar and Rhetoric, t. 18, nr 31, s. 121–136, 2009, dostępne z: http://logika.uwb.edu.pl/studies/download.php?volid=31&artid=ju.

[90] J. Urban, G. Sutcliffe, *On the Integrity of a Repository of Formalized Mathematics*, w: S. Autexier i in. (red.), Intelligent Computer Mathematics, LNCS, t. 6167, s. 132–146. Springer-Verlag, 2010, doi: 10.1007/3-540-36469-2_13.

[91] N. C. Waugh, D. A. Norman, *Primary Memory*, Psychological Review, t. 72, s. 89–104, 1965.

[92] M. Wenzel, *The Isabelle/Isar Reference Manual*, University of Cambridge, 2011, dostępne z: http://www.ki.informatik.uni-frankfurt.de/doc/nonhtml/ Isabelle99/doc/ref.pdf.

[93] M. Wenzel, F. Wiedijk, *A Comparison of Mizar and Isar*, Journal of Automated Reasoning, t. 29, nr 3–4, s. 389–411, 2002.

[94] A. N. Whitehead, B. Russell, *Principia Mathematica*, Cambridge Mathematical Library, Cambridge University Press, 1910.

[95] D. D. Wicknes, D. G. Born, C. K. Allen, *Proactive Inhibition and Item Similarity in Short-Term Memory*, Journal of Verbal Learning and Verbal Behavior, t. 2, s. 440–445, 1963.

[96] F. Wiedijk, *The De Bruijn Factor*, dostępne z: http://www.cs.ru.nl/~freek/factor/factor.pdf.

[97] F. Wiedijk, *Mizar Light for HOL Light*, w: J. B. Richard i in. (red.), Proc. of the 14th International Conference on Theorem Proving in Higher Order Logics, LNCS, t. 2152, s. 378–393, Springer-Verlag, 2001, doi: 10.1007/3-540-44755-5_26.

[98] F. Wiedijk, *Formal Proof Sketches*, w: S. Berardi i in. (red.), Theorem Proving in Higher Order Logics, LNCS, t. 3085, s. 378–393, Springer-Verlag, 2003.

[99] R. J. Wilson *Wprowadzenie do teorii grafów*, PWN, 2007.

[100] C. P. Wirth, *Hilbert's Epsilon as an Operator of Indefinite Committed Choice*, Journal of Applied Logic, t. 6, nr 3, s. 287–317, 2008, doi: 10.1016/j.jal.2007.07.009.

Wykaz oznaczeń

$\langle a_1, a_2, \ldots, a_k \rangle$	ciąg $\{a_i\}_{i=1}^k$, 12
L^n	n–ty poziom zagnieżdżenia w lesie dendroidów L, 15
$\mathcal{E}(G)$	zbiór krawędzi grafu G, 11
$\mathcal{V}(G)$	zbiór wierzchołków grafu G, 11
$\mathcal{A}(D)$	zbiór łuków digrafu D, 11
$\mathcal{V}(D)$	zbiór wierzchołków digrafu D, 11
(v, u)	łuk łączący wierzchołki v i u, 11
vu	łuk łączący wierzchołki v i u, 11
$d_\tau(v, u)$	τ–rozpiętość łuku vu, 13
$\underset{A}{\longrightarrow}$	relacja bycia następnikiem w digrafie $\langle \mathcal{V}(D), A \rangle$, 11
$\underset{A}{\longrightarrow}^*$	relacja osiągalności w digrafie $\langle \mathcal{V}(D), A \rangle$, 11
$\underset{A}{\overset{V}{\longmapsto}}$	relacja osiągalności względem A, V, 49
$u_1 \underset{A}{\longrightarrow} \ldots \underset{A}{\longrightarrow} u_n$	skierowana A–trasa, 12
$\mathcal{A}(\mathfrak{u})$	zbiór łuków trasy $\mathfrak{u}$, 12
$\mathcal{V}(\mathfrak{u})$	zbiór wierzchołków trasy $\mathfrak{u}$, 12
$\mathcal{N}_A^-(v)$	zbiór poprzedników wierzchołka v w digrafie $\langle \mathcal{V}(D), A \rangle$, 12
$\mathcal{N}_A^+(v)$	zbiór następników wierzchołka v w digrafie $\langle \mathcal{V}(D), A \rangle$, 12
$V_1 \underset{A}{\curvearrowright} V_2$	zbiór A–łuków łączących wierzchołki ze zbioru V_1 z wierzchołkami ze zbioru V_2, 12
$\rho_A(V)$	gęstość zbioru wierzchołków V w domknięciu zwrotno przechodnim zbioru łuków A, 12
$D_{\|V}$	podgraf digrafu D indukowany wierzchołkowo przez zbiór V, 12
$\mathbf{1}_\tau^A$	zbiór A–łuków o τ–rozpiętości 1, 13
$TS(D)$	zbiór sortowań topologicznych digrafu D, 13
τ_A	partycja digrafu D względem linearyzacji τ i zbioru A, 13
$\mathcal{H}_A$	własność $\mathcal{H}_A$, 14
$\mathcal{V}(\mathfrak{P})$	zbiór wierzchołków w grafie dowodu $\mathfrak{P}$, 29
$\mathcal{A}(\mathfrak{P})$	zbiór łuków argumentujących w grafie dowodu $\mathfrak{P}$, 29
$\mathcal{M}(\mathfrak{P})$	zbiór metakrawędzi w grafie dowodu $\mathfrak{P}$, 29
$\mathcal{O}rd_{\mathfrak{P}}$	zbiór łuków porządkujących w grafie dowodu $\mathfrak{P}$, 29
$\mathcal{O}rd_{\mathfrak{P}}^{\mathcal{P}}$	zbiór łuków pierwotnie porządkujących w grafie dowodu $\mathfrak{P}$, 29
$\mathcal{R}ef_{\mathfrak{P}}$	zbiór łuków referencyjnych w grafie dowodu $\mathfrak{P}$, 29
$\mathfrak{S}_{\mathfrak{P}}$	zbiór wierzchołków szkieletowych w grafie dowodu $\mathfrak{P}$, 29
$\mathcal{S}kel_{\mathfrak{P}}$	zbiór łuków szkieletowych w grafie dowodu $\mathfrak{P}$, 29
$\mathfrak{D}_{\mathfrak{P}}$	digraf $\langle \mathcal{V}(\mathfrak{P}), \mathcal{A}(\mathfrak{P}) \rangle$, 28, 29
$\mathfrak{G}_{\mathfrak{P}}$	digraf $\langle \mathcal{V}(\mathfrak{P}), \mathcal{A}(\mathfrak{P}) \cup \mathcal{M}(\mathfrak{P}) \rangle$, 29
$\mathfrak{M}eta_{\mathfrak{P}}$	digraf $\langle \mathcal{V}(\mathfrak{P}), \mathcal{M}(\mathfrak{P}) \rangle$, 28, 29
$\mathfrak{O}_{\mathfrak{P}}^{\mathcal{P}}$	graf pierwotnie porządkujący $\langle V, \mathcal{O}rd_{\mathfrak{P}}^{\mathcal{P}} \rangle$, 29
$\mathfrak{R}_{\mathfrak{P}}$	graf referencyjny $\langle V, \mathcal{R}ef_{\mathfrak{P}} \rangle$, 29
$\mathcal{O}bsz(\mathcal{P})$	obszar paczki $\mathcal{P}$, 40
$\mathcal{P}re(\mathcal{P})$	zbiór przesłanek paczki $\mathcal{P}$, 40
$\mathcal{C}on(\mathcal{P})$	zbiór konkluzji paczki $\mathcal{P}$, 40
$\mathcal{P}(c)$	zbiór przesłanek konkluzji c w paczce $\mathcal{P}$, 46
$\mathcal{G}(D, \pi)$	graf partycji π digrafu D, 14
$\pi_{\|X}$	obcięcie partycji π do zbioru X, 14
$\mathfrak{h}^\pi(P)$	droga Hamiltona w digrafie $D_{\|P}$, 14
$\mathfrak{h}^\pi(v)$	droga Hamiltona $\mathfrak{h}^\pi(P)$ zawierająca wierzchołek v jeśli $v \in P$, 14

Skorowidz