Metody algebraiczne w teorii języków drzew nieskończonych

autoreferat rozprawy doktorskiej

Tomasz Idziaszek

Wydział Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego

20 lutego 2013

Celem niniejszej rozprawy jest zbadanie algebraicznego podejścia do regularnych języków drzew nieskończonych. Realizacja tego celu przebiegła dwutorowo, co znalazło swoje odbicie w strukturze pracy:

(1) Zaproponowanie algebraicznej struktury dla drzew nieskończonych – temu zagadnieniu poświęcona jest pierwsza część rozprawy.

(2) Użycie tej struktury w celu otrzymania algorytmów, które rozstrzygają przynależność danego języka regularnego drzew nieskończonych do ustalonej klasy języków – o tym traktuje druga część rozprawy.

Metody i techniki wykorzystywane w pracy leżą na pograniczu algebry, logiki i teorii automatów.

Rozważane w rozprawie drzewa nieskończone są nieurangowane (każdy węzeł posiada skończoną liczbę dzieci, ale ich liczba nie jest z góry ograniczona), uporządkowane (zadany jest porządek na dzieciach węzła) i mogą posiadać liście (w szczególności do klasy drzew nieskończonych zaliczamy wszystkie drzewa skończone). Ponadto drzewa są etykietowane (każdy węzeł ma przypisaną etykietę z pewnego skończonego alfabetu).

Podczas badań okazało się, że na szczególną uwagę zasługuje pewna podklasa drzew nieskończonych, a mianowicie klasa drzew *cienkich*. Drzewo nieskończone nazwiemy cienkim, jeśli zawiera ono przeliczalnie wiele gałęzi (lub, równoważnie, nie zawiera ono pełnego nieskończonego drzewa binarnego jako minora). Cienkie drzewa mogą być traktowane jako struktura pośrednia pomiędzy słowami nieskończonymi a drzewami nieskończonymi. Z jednej strony uogólniają zarówno drzewa skończone jak i słowa nieskończone, z drugiej strony są prostsze w badaniu niż ogólne drzewa nieskończone i stanowią interesującą klasę o dobrych własnościach. Wierzę, że badanie drzew cienkich może nam pomóc lepiej zrozumieć ogólne drzewa nieskończone. Z tego też powodu część wyników rozprawy dotyczy metod algebraicznych i algorytmów dla drzew cienkich.

Wyniki rozprawy dotyczące logiki EF i algebr dla drzew dowolnych zostały opublikowane w pracy [4], zaś dotyczące drzew cienkich – w pracy [5].

Dziedzina badań

Języki słów skończonych, słów nieskończonych, drzew skończonych oraz drzew nieskończonych można opisywać za pomocą formuł monadycznej logiki drugiego rzędu (MSO). Ponadto, dla każdego z tych obiektów istnieje równoważny (tzn. rozpoznający tę samą klasę języków) model automatu skończonego. Stąd też języki definiowalne formułami MSO są nazywane *językami regularnymi* [9, 13, 25]. Jednakże wiele języków może być opisanych w prostszych logikach, np. logika pierwszego rzędu lub logiki temporalne. Możemy więc zadać pytanie, czy dany język regularny może być wyrażony w takiej logice. W ogólności takie pytanie możemy zadać dla dowolnej klasy języków X. Powiemy, że klasa X posiada *efektywną charakteryzację*, jeśli następujący problem decyzyjny jest rozstrzygalny: „mając dany na wejściu język regularny, stwierdzić, czy należy on do klasy X".

Dla języków słów skończonych Schützenberger [20] oraz McNaughton i Papert [15] udowodnili twierdzenie, które mówi, że następujące cztery warunki są równoważne dla regularnego języka słów skończonych L:

(a) L może być opisany formułą logiki pierwszego rzędu (z relacją porządku i predykatami dla liter),

(b) L może być opisany za pomocą wyrażenia bezgwiazdkowego (podzbioru wyrażeń regularnych, który nie dopuszcza gwiazdki Kleene'ego, ale dopuszcza użycie dopełnienia),

(c) minimalny automat deterministyczny języka L jest *counter-free* (nie zawiera pewnego rodzaju pętli),

(d) monoid syntaktyczny języka L (tzn. minimalny monoid, który rozpoznaje L) nie zawiera nietrywialnej podgrupy.

Powyższe twierdzenie daje nam charakteryzację pewnej klasy języków w terminach (a) logiki, (b) wyrażeń regularnych, (c) automatów i (d) algebry. Warunki (c) i (d) mogą być sprawdzone efektywnie, zatem twierdzenie daje nam efektywną charakteryzację logiki pierwszego rzędu, jak również efektywną charakteryzację wyrażeń bezgwiazdkowych.

Efektywne charakteryzacje są żywym i ważnym tematem w teorii języków regularnych. W przypadku słów skończonych jest tu wiele wyników [23, 24, 10, 14, 22, 27, 21], z których część przenosi się również na przypadek słów nieskończonych [26, 18]. Mniej wiadomo dla drzew skończonych, ale tutaj też uzyskano ciekawe wyniki [7, 6, 19, 1, 3]. Efektywne charakteryzacje są zwykle trudnymi problemami, które wymagają głębokiego zrozumienia struktury badanej klasy języków. Zwykle jest ono osiągnięte dzięki zastosowaniu odpowiednich struktur algebraicznych opisujących języki, tj. półgrupy dla słów skończonych, algebry Wilkego dla słów nieskończonych oraz algebry leśne dla drzew skończonych. Co więcej, metody algebraiczne zwykle powodują, że sformułowania twierdzeń są elegantsze (sprowadzają się do sprawdzania równości w algebrze), a dowody prostsze niż w przypadku użycia innych technik.

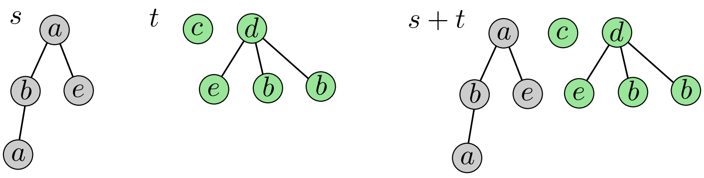
Dla drzew nieskończonych satysfakcjonująca struktura algebraiczna nie została jeszcze zaproponowana. Są powody by podejrzewać, że jest to trudne zadanie. W przypadku obiektów skończonych struktury algebraiczne dla języków są ściśle związane z automatami deterministycznymi rozpoznającymi te języki, zaś te w przypadku obiektów nieskończonych mają mniejszą siłę wyrazu niż automaty niedeterministyczne. Z drugiej strony, w przeciwieństwie do słów, nie znamy naturalnego sposobu zwartej reprezentacji licznych sposobów budowy drzew z mniejszych drzew. Struktura algebraiczna dla drzew nieskończonych musi poradzić sobie z oboma tymi problemami. Dotychczasowe propozycje [4, 2] albo nie dają skończonej reprezentacji, albo nie wykazały jeszcze swojej użyteczności, prowadząc do efektywnej charakteryzacji nietrywialnej klasy języków.

Część pierwsza: algebry

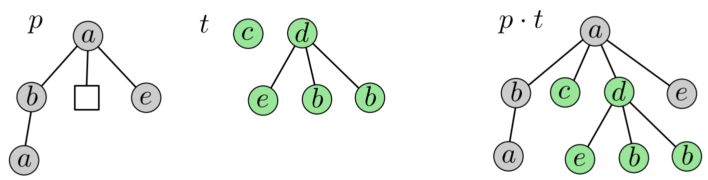
W pierwszej części rozprawy przedstawiam trzy algebry dla drzew nieskończonych. Dwie z nich będą dotyczyć drzew cienkich, trzecia – ogólnych drzew nieskończonych.

Ponieważ jesteśmy zainteresowani algebraicznym podejściem do drzew, będziemy potrzebowali operacji, które pozwolą nam budować drzewa z podstawowych cegiełek (pojedynczych liter). W przypadku słów skończonych taką operacją jest konkatenacja słów, która umożliwia sklejenie dwóch słów w jedno. W przypadku drzew nie mamy tutaj naturalnego kandydata. W szczególności konkatenacja dwóch drzew (dla jednego ze sposobów uogólnienia konkatenacji) powoduje powstanie obiektu o dwóch korzeniach. Z tego też powodu w algebrze leśnej dla drzew skończonych [8] postanowiono pracować z *lasami* (uporządkowanymi ciągami drzew), a nie z drzewami. Z tego samego powodu pracujemy z lasami nieurangowanymi, podczas gdy w literaturze bardziej popularne są obiekty urangowane (głównie binarne, zob. [11]), gdyż dla takich obiektów łatwiej jest zdefiniować automat skończony.

Pierwszą operacją jest *konkatenacja lasów*, która umożliwia poziome połączenie dwóch lasów. Oznaczamy tę operację symbolem $+$. Jeśli s i t są dwoma lasami, to $s+t$ jest lasem, który ma tyle samo korzeni, co s i t łącznie. Operacja ta niekoniecznie jest przemienna. Las pusty jest jej elementem neutralnym.



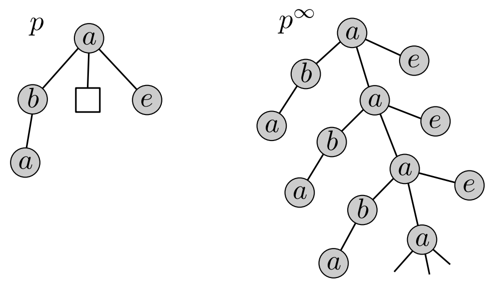
Potrzebujemy również sposobu na pionowe łączenie lasów. Z tego powodu wprowadzamy nowy obiekt – las z jednym wyróżnionym liściem. Ten nowy obiekt nazwiemy *kontekstem*, zaś jego wyróżniony liść, który oznacza miejsce, gdzie możemy doczepić drugi las – *portem*. Możemy *złączyć* kontekst p z lasem t, uzyskując las pt, który powstaje przez zastąpienie portu w kontekście p przez las t. Zauważmy, że liczba następników rodzica portu zwiększa się o liczbę korzeni w t mniej jeden.



Możemy również złączyć kontekst p z innym kontekstem q, a uzyskany w ten sposób kontekst pq spełnia równanie $(pq)t = p(qt)$ dla dowolnego lasu t. W końcu, możemy skonkatenować las t z kontekstem p, co da kontekst $t+p$. Symetrycznie możemy stworzyć kontekst $p+t$.

Powyższe operacje są wystarczające do wygenerowania wszystkich lasów skończonych (zawierających skończenie wiele węzłów). W pracy [8] opisano *algebrę leśną*, strukturę o dwóch sortach (dla lasów i dla kontekstów), z powyższymi operacjami i odpowiednimi aksjomatami, która spełnia dwa warunki: (a) zbiór lasów skończonych i kontekstów skończonych z powyższymi operacjami jest wolną algebrą w klasie wszystkich algebr leśnych, (b) język lasów skończonych jest regularny wtedy i tylko wtedy, gdy jest rozpoznawany przez homomorfizm w pewną skończoną algebrę leśną, tzn. język ten jest przeciwobrazem homomorficznym pewnego podzbioru elementów algebry. Minimalną algebrę rozpoznającą język nazywamy *algebrą syntaktyczną*.

W celu wygenerowania lasów nieskończonych, rozszerzamy algebrę leśną o nową operację. Kontekst nazwiemy *strzeżonym*, jeśli jego port nie znajduje się w korzeniu. Możemy złączyć strzeżony kontekst p ze sobą nieskończenie wiele razy, co da nam nieskończony las t, który jest jedynym lasem spełniającym równanie $pt = t$. Oznaczamy tę operację przez p^∞.



Przypominamy, że las nazywamy cienkim, jeśli ma przeliczalnie wiele gałęzi. Las nazywamy *regularnym*, jeśli zawiera tylko skończenie wiele nieizomorficznych poddrzew. Przykładowo, pełne drzewo binarne jest regularne (każde jego poddrzewo jest izomorficzne z całym drzewem), natomiast żadne drzewo, w którym liczba następników węzła nie jest ograniczona, nie jest regularne.

W **rozdziale 4** pokazuję, że zdefiniowane powyżej operacje są wystarczające, aby wygenerować wszystkie cienkie lasy regularne. Definiuję *algebrę cienką*, trzysortową (lasy, konteksty strzeżone i pozostałe konteksty) strukturę i dowodzę następujących twierdzeń:

Twierdzenie 1 *Zbiór cienkich lasów regularnych i cienkich kontekstów regularnych z powyższymi operacjami jest wolną algebrą w klasie wszystkich algebr cienkich.*

Twierdzenie 2 *Język cienkich lasów regularnych jest regularny wtedy i tylko wtedy, gdy jest rozpoznawany przez homomorfizm w pewną skończoną algebrę cienką.*

Zauważmy, że do wygenerowania wszystkich lasów (których jest nieprzeliczalnie wiele) nie wystarczy nam skończona liczba operacji. Możemy jednak ograniczyć się do regularnych lasów, korzystając z faktu, że każdy język regularny lasów jest jednoznacznie wyznaczony przez regularne lasy do niego należące.

Niemniej jednak, definiuję również drugi rodzaj algebry cienkiej, dla której obiektem wolnym jest zbiór wszystkich cienkich lasów i kontekstów. Ma ona, oczywiście, nieskończenie wiele operacji. W pewnym sensie można na algebrę cienką patrzeć jako na uogólnienie zarówno algebry leśnej jak i algebry Wilkego dla słów nieskończonych [26], natomiast drugi rodzaj algebry cienkiej uogólnia algebrę leśną jak również ω-półgrupy [17].

Ostatecznie pokazuję jak efektywnie przechodzić pomiędzy reprezentacją języka lasów cienkich w postaci automatu skończonego a homomorfizmem w skończoną algebrę cienką:

Twierdzenie 3 *Istnieje algorytm, który dla danego języka regularnego lasów cienkich zadanego przez automat skończony oblicza jego syntaktyczną algebrę cienką i homomorfizm w tę algebrę, który rozpoznaje dany język. Istnieje algorytm, który, mając dany homomorfizm w algebrę cienką, oblicza automat skończony dla języka rozpoznawanego przez ten homomorfizm.*

W **rozdziale 5** proponuję algebrę dla lasów dowolnych. Nie ma ona tak dobrych własności, jak algebra cienka, w szczególności zawiera ona nieskończenie wiele operacji i aksjomatów. Jednakże pokazuję, że każdy język regularny regularnych lasów nieskończonych jest rozpoznawany przez homomorfizm w skończoną algebrę, oraz przedstawiam algorytm, który oblicza syntaktyczną algebrę dla języka zadanego przez automat skończony. Jak pokazuję w drugiej części rozprawy, to wystarczy, aby zastosować tę algebrę w praktyce.

Ponieważ w pracy definiuję kilka struktur algebraicznych, zebrałem w **rozdziale 3** podstawowe definicje i twierdzenia, używając terminologii algebry abstrakcyjnej dla algebr wielosortowych. Ponadto definiuję odpowiedniki rozmaitości algebr dla przypadku wielosortowego i dowodzę wielosortowego odpowiednika twierdzenia Eilenberga [12]:

Twierdzenie 4 *Istnieje wzajemna jednoznaczność pomiędzy rozmaitościami algebr wielosortowych a rozmaitościami języków.*

Część druga: efektywne charakteryzacje

W drugiej części rozprawy przedstawiam algorytmy, które rozstrzygają przynależność regularnych języków lasów nieskończonych do wybranych klas języków. W większości przypadków sprowadzają się one do sprawdzenia odpowiedniego równania w algebrze syntaktycznej rozpoznającej dany język lasów. Elementy sortu leśnego oznaczam literami h, g, dla sortów kontekstowych używam liter v, u, w.

W **rozdziale 6** przedstawiam algorytmy badające, czy język regularny lasów cienkich spełnia pewne proste własności: przemienność, zamkniętość na bisymulację oraz otwartość.

Język lasów skończonych nazwiemy *przemiennym*, jeśli jest zamknięty na zmianę kolejności braci (węzłów drzewa, które mają tego samego rodzica). Łatwo pokazać, że język lasów skończonych jest przemienny, wtedy i tylko wtedy gdy w jego syntaktycznej algebrze leśnej prawdziwe jest równanie $h+g=g+h$. Pokazuję, że równanie to w przypadku lasów nieskończonych opisuje słabszą przemienność, tzn. język je spełniający jest zamknięty na taką zmianę kolejności braci, w której na każdej ścieżce mamy jedynie skończenie wiele zmian. Pokazuję również, że ogólna przemienność dla lasów cienkich jest równoważna prawdziwości równania $h + v = v + h$.

Dwa lasy t_0 i t_1 są *bisymulacyjnie równoważne*, jeśli gracz II wygrywa następującą grę, rozgrywaną przez dwóch graczy [16]. Gracz I rozpoczyna grę, wybierając $i \in \{0, 1\}$ oraz korzeń x_i lasu t_i (gracz I przegrywa, jeśli lasy są puste). Następnie gracz II odpowiada, wybierając korzeń x_{1-i} lasu t_{1-i}, o tej samej etykiecie co x_i (jeśli takiego węzła nie ma – gra się kończy i wygrywa gracz I). Następnie gra jest kontynuowana na lasach s_0, s_1, gdzie s_i jest lasem powstałym po wzięciu poddrzewa zaczepionego w x_i i usunięciu z niego korzenia. Gracz II wygrywa, jeśli wytrwa nieskończenie wiele rund.

Język regularny lasów L jest *zamknięty na bisymulację*, jeśli dla każdych dwóch bisymulacyjnie równoważnych lasów do L należą oba lub żaden z nich. Dla języków lasów skończonych ta własność jest równoważna równaniom $h + g = g + h$ i $h + h = h$ [8]. Pokazuję, że język regularny lasów cienkich jest zamknięty na bisymulację, jeśli w jego syntaktycznej algebrze cienkiej prawdziwe są równania

$$h + v = v + h, \quad h + h = h, \quad (v^\infty + v)^\infty = v^\infty.$$

Niektóre z własności języków nie mają swoich odpowiedników dla obiektów skończonych. Zbiór lasów może być traktowany jako przestrzeń metryczna. Definiujemy odległość pomiędzy dwoma lasami jako 2^{-n}, gdzie n jest najmniejszą głębokością, na której te dwa lasy się różnią. Dla tak zdefiniowanej topologii możemy zadać pytanie, czy dany język lasów jest *otwarty*, tzn. czy jest zbiorem otwartym w tej topologii. Rozstrzygalność tego problemu była znana, wkład rozprawy polega na przedstawieniu charakteryzacji algebraicznej. Pokazuję, że język regularny lasów cienkich jest otwarty, jeśli jego syntaktyczna algebra cienka spełnia warunek, że jeśli element v^∞ należy do obrazu języka przy homomorfizmie, to dla dowolnego h element $v^\omega h$ również należy. Co ciekawe, warunek ten jest analogiczny do warunku, który musi spełniać otwarty język słów nieskończonych [18].

Twierdzenie 5 *Dla każdej z trzech własności języków (przemienność, zamkniętość na bisymulację, otwartość w standardowej topologii) istnieje algorytm stwierdzający, czy dany język regularny cienkich lasów posiada tę własność.*

Rozdział 7 poświęcony jest badaniu definiowalności języków lasów nieskończonych w logice temporalnej EF (równoważnej XPath z operacją *descendant*). Pokazuję ją dla lasów cienkich i ogólnych.

Formuły w logice EF są budowane następująco: (a) dla dowolnej etykiety istnieje formuła, która sprawdza, czy w korzeniu drzewa jest ta etykieta, (b) formuły są zamknięte na operacje Boole'owskie, (c) jeśli φ jest formułą, to EFφ jest formułą, która jest prawdziwa w drzewach, w których φ jest prawdziwa w pewnym właściwym poddrzewie.

Z logiką EF związane jest pojęcie EF-bisymulacji. Powiemy, że język regularny lasów L jest *zamknięty na EF-bisymulację*, jeśli dla każdych dwóch EF-bisymulacyjnie równoważnych lasów do L należą oba lub żaden z nich. Definicja EF-bisimulacyjnej równoważności jest analogiczna do bisymulacyjnej równoważności, z tą różnicą, że gracz I jako węzeł x_i może wybrać dowolny węzeł lasu t_i (niekoniecznie korzeń) i taką samą swobodę ma gracz II. Lasy EF-bisymulacyjnie równoważne są modelami dla tych samych formuł EF.

W [8] udowodniono, że język regularny lasów skończonych jest definiowalny w logice EF, gdy jest zamknięty na EF-bisymulację. W przypadku lasów nieskończonych tak nie jest, o czym świadczy język „wszystkie lasy skończone", który spełnia tylko tę drugą własność. Pokazuję jednak, że język regularny lasów nieskończonych jest definiowalny w logice EF, jeśli jest zamknięty na EF-bisymulację oraz jego algebra syntaktyczna spełnia równanie $v^\omega h = (v + v^\omega h)^\infty$.

Jak widać, nadal kluczową rolę odgrywa tu zamkniętość na EF-bisymulację. W przypadku lasów ogólnych podaję algorytm, który ją rozstrzyga. W przypadku regularnych lasów cienkich podaję również charakteryzację algebraiczną za pomocą równań

$$h + v = v + h, \quad vh = vh + h, \quad (v + (vw)^\infty)^\infty = (vw)^\infty, \quad (vwu)^\infty = (wvu)^\infty.$$

Twierdzenie 6 *Dla każdej z dwóch własności języków (definiowalność w logice temporalnej EF, zamkniętość na EF-bisymulację) istnieje algorytm stwierdzający, czy dany język regularny lasów nieskończonych posiada tę własność.*

Literatura

[1] M. Benedikt and L. Segoufin. Regular languages definable in FO. In *Symposium on Theoretical Aspects of Computer Science*, volume 3404 of *Lecture Notes in Computer Science*, pages 327–339, 2005. A revised version, correcting an error from the conference paper, is available at `www.lsv.ens-cachan.fr/ segoufin/Papers/`.

[2] A. Blumensath. Recognisability for algebras of infinite trees. *Theor. Comput. Sci.*, 412(29):3463–3486, 2011.

[3] M. Bojańczyk. Effective characterizations of tree logics. In *PODS*, pages 53–66, 2008.

[4] M. Bojańczyk and T. Idziaszek. Algebra for infinite forests with an application to the temporal logic EF. In M. Bravetti and G. Zavattaro, editors, *CONCUR*, volume 5710 of *Lecture Notes in Computer Science*, pages 131–145. Springer, 2009.

[5] M. Bojańczyk, T. Idziaszek, and M. Skrzypczak. Regular languages of thin trees. Accepted for *International Symposium on Theoretical Aspects of Computer Science, STACS*, 2013.

[6] M. Bojańczyk and L. Segoufin. Tree languages definable with one quantifier alternation. In *International Colloquium on Automata, Languages and Programming*, volume 5126 of *Lecture Notes in Computer Science*, pages 233–245, 2008.

[7] M. Bojańczyk, L. Segoufin, and H. Straubing. Piecewise testable tree languages. In *Logic in Computer Science*, pages 442–451, 2008.

[8] M. Bojańczyk and I. Walukiewicz. Forest algebras. In *Automata and Logic: History and Perspectives*, pages 107–132. Amsterdam University Press, 2007.

[9] J.R. Büchi. Weak second-order arithmetic and finite automata. *Z. Math. Logik Grundl. Math.*, 6:66–92, 1960.

[10] J. Cohen, D. Perrin, and J.-É. Pin. On the expressive power of temporal logic. *Journal of Computer and System Sciences*, 46(3):271–294, 1993.

[11] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata`, 2007. release October, 12th 2007.

[12] S. Eilenberg. *Automata, Languages and Machines*, volume B. Academic Press, New York, 1976.

[13] C.C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the AMS*, 98:21–52, 1961.

[14] K. Etessami and T. Wilke. An UNTIL hierarchy for temporal logic. In *Logic in Computer Science*, pages 108–117, 1996.

[15] R. McNaughton and S. Papert. *Counter-Free Automata.* MIT Press, 1971.

[16] R. Milner. *Communication and concurrency.* PHI Series in computer science. Prentice Hall, 1989.

[17] D. Perrin and J.-É. Pin. Semigroups and automata on infinite words. *Semigroups, formal languages and groups (York, 1993)*, pages 49–72, 1995.

[18] D. Perrin and J.-É. Pin. *Infinite Words*. Elsevier, 2004.

[19] T. Place and L. Segoufin. Deciding definability in $FO_2(<)$ (or XPath) on trees. In *LICS*, pages 253–262. IEEE Computer Society, 2010.

[20] M.P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8:190–194, 1965.

[21] I. Simon. Piecewise testable events. In *Automata Theory and Formal Languages*, pages 214–222, 1975.

[22] D. Thérien and T. Wilke. Temporal logic and semidirect products: An effective characterization of the Until hierarchy. In *Foundations of Computer Science*, pages 256–263, 1996.

[23] D. Thérien and T. Wilke. Over words, two variables are as powerful as one quantifier alternation. In *ACM Symposium on the Theory of Computing*, pages 256–263, 1998.

[24] W. Thomas. Classifying regular events in symbolic logic. *Journal of Computer and System Sciences*, 25:360–375, 1982.

[25] B.A. Trakthenbrot. Finite automata and the logic of monadic second order predicates (Russian). *Doklady Academii Nauk SSSR*, 140:326–329, 1961.

[26] T. Wilke. An algebraic theory for languages of finite and infinite words. *Inf. J. Alg. Comput.*, 3:447–489, 1993.

[27] T. Wilke. Classifying discrete temporal properties. In *Symposium on Theoretical Aspects of Computer Science*, volume 1563 of *Lecture Notes in Computer Science*, pages 32–46, 1999.