

Autoreferat rozprawy doktorskiej:

Implementacja systemu wnioskującego za pomocą DNA

Łukasz Rogowski

Wydział Matematyki i Informatyki Uniwersytetu Łódzkiego

Obliczenia DNA (z ang. DNA computing) są nowatorską dziedziną nauki, leżącą na pograniczu informatyki i biologii molekularnej. Przedmiotem jej badań są komputery biomolekularne, które w przyszłości mogą stanowić alternatywę dla tradycyjnych systemów komputerowych. Mają one ogromny potencjał ze względu na masowość i równoległość obliczeń. Narzędziem, wykorzystywanym do rozwiązywania problemów algorytmicznych, matematycznych czy logicznych, są tu łańcuchy DNA, a same obliczenia przeprowadzane są w odpowiednich warunkach laboratoryjnych za pomocą operacji biochemicznych. Siła tych obliczeń nie tkwi w ich prędkości a w masowej równoległości operacji, dzięki czemu problemy o naprawdę dużej złożoności mogą być rozwiązywane przy sensownych ograniczeniach czasowych [15], [20], [32]. Komputer DNA (rozumiany jako odpowiednio dobrany zbiór molekuł i enzymów reagujących w określonych warunkach laboratoryjnych) jest też najmniejszym modelem komputera jaki udało się dotąd zaprojektować [41]. Jednym z kierunków w tej dziedzinie jest implementacja systemów wnioskujących znanych z logiki matematycznej. Prezentowana praca dotyczy tego tematu.

Najprostszym systemem formalnym logiki matematycznej jest klasyczny rachunek zdań. Bada on zbiór zdań elementarnych (zwanych atomowymi), które mogą przyjmować tylko dwie wartości: *prawdy* lub *falszu*, oraz powstający z nich (za pomocą spójników logicznych) zbiór zdań złożonych. W otaczającym nas świecie często spotykamy się z następującym problemem: dany jest układ skończony zdań złożonych A (mogą to być dowolne zdania, niekoniecznie tautologie). Traktujemy je jako specyficzne aksjomaty czyli zdania, które uznajemy w danym układzie za prawdziwe. Wówczas dla dowolnego zdania p pytamy czy wynika ono z A za pomocą reguł wnioskowania, czyli czy p jest konsekwencją zdań z A [7]. Realizacja rozwiązania tego problemu za pomocą łańcuchów DNA i operacji na nich jest głównym celem tej pracy.

DNA (kwas deoksyrybonukleinowy) to polimer czyli substancja chemiczna składająca się z wielokrotnie powtórzonych jednostek, którego elementami są deoksyrybonukleotydy, w skrócie nazywane nukleotydami. Struktura DNA została odkryta w 1953 roku przez Jamesa Watsona i Francisca Cricka [47]. Zaproponowany model budowy został potwierdzony wieloma późniejszymi badaniami w tym również związanymi bezpośrednio z obliczeniami wykorzystującymi DNA [11], [20]. Każdy nukleotyd zbudowany jest z reszty cukrowej (deoksyrybozy), grupy fosforanowej i jednej z czterech zasad azotowych: adeniny, guaniny, cytozyny i tyminy. Ponieważ głównym nośnikiem informacji są właśnie zasady azotowe, nazwy czterech możliwych do uzyskania nukleotydów pochodzą od pierwszych liter nazw zasad: A, G, C i T. W skład cząsteczki DNA zwykle wchodzi dwa łańcuchy, w których

początek pierwszego łączy się z końcem drugiego. Owijają się one wokół wspólnej osi i tworzą prawoskrętną podwójną helisę. Zasady skierowane są do wnętrza i tworzą odpowiednie pary komplementarne – adenina łączy się zawsze z tyminą (dwa wiązania wodorowe), a cytozyna z guaniną (trzy wiązania wodorowe). Nie istnieją inne możliwości połączenia się nukleotydów. Łańcuchy biegną antyrównolegle, dzięki czemu znając jeden łańcuch możemy bez problemu odtworzyć drugi – działa tu zasada komplementarności Watsona-Cricka [47]. Łańcuch DNA wykazuje polarność (czyli nierównomierne rozłożenie cząstkowych ładunków elektrycznych, dzięki czemu molekula jest dipolem elektrycznym). Jeden z końców ma grupę 5'-OH, a drugi 3'-OH. Przyjmuje się, że niezwiązana grupa 5'-OH jest ulokowana po lewej stronie zapisu (oznaczana skrótowo 5'), a grupa 3'-OH po prawej (oznaczana 3'). Wewnątrz pojedynczego łańcucha nukleotydy łączą się ze sobą za pomocą silnych wiązań fosfodiesterowych. Wyróżniamy następujące podstawowe operacje laboratoryjnego działania na DNA: łączenie się łańcuchów komplementarnych (ligacja), cięcie za pomocą odpowiednich enzymów restrykcyjnych (restryktacja), kopiowanie łańcuchów (replikacja), rozdzielanie komplementarnych nici (denaturacja), laboratoryjne uzyskiwanie łańcucha o dowolnej sekwencji zasad (synteza), sortowanie molekuł według ich długości (elektroforeza). W rozprawie operacje te zostały omówione szczegółowo wraz z ilustrującymi ich działaniem rysunkami i przykładami.

Historia obliczeń DNA sięga roku 1994, kiedy to Leonard Adleman (jeden ze współtwórców systemu kryptograficznego RSA) jako pierwszy zademonstrował możliwość rozwiązania problemu drogi Hamiltona w grafie (dla siedmiu wierzchołków) za pomocą łańcuchów DNA [1]. Zagadnienie to polega na rozstrzygnięciu czy w grafie skierowanym istnieje droga przechodząca przez wszystkie jego wierzchołki (przez każdy z nich tylko raz). Problem ten jest problemem NP-zupełnym, o którym nie wiadomo czy jego rozwiązanie można znaleźć w czasie wielomianowym. Wzorem Adlemana pojawiło się wiele kolejnych koncepcji rozwiązania innych problemów informatycznych, zarówno w formie teoretycznych rozważań jak i implementacji laboratoryjnych, między innymi [14], [54]. Trochę inną gałęzią obliczeń DNA stały się implementacje teoretycznych modeli maszyn obliczeniowych, w szczególności implementacje konkretnych modeli automatów, zapoczątkowane w 2001 roku przez grupę naukowców z Instytutu Weizmanna w Izraelu (Yaakov Benenson, Ehud Shapiro i inni). W pierwszej ich pracy został przedstawiony model dwustanowego, niedeterministycznego, programowalnego automatu skończonego o dwuelementowym alfabetcie, zbudowanego z łańcuchów DNA oraz jednego enzymu restrykcyjnego [2]. Po raz pierwszy przedstawiona została idea splatania czyli naprzemiennego cięcia i łączenia się łańcuchów DNA, reprezentujących poszczególne elementy automatu. Koncepcja była rozszerzana wielokrotnie, zarówno przez samych autorów [3], [4], [5] jak i innych [40], [44], gdzie przedstawiono ideę rozszerzenia do maszyny 3-stanowej. Ważna praca w tej dziedzinie, przedstawiająca laboratoryjną implementację takiej maszyny wykorzystującej jednocześnie dwa enzymy restrykcyjne i rozszerzającą jej możliwości do 6 stanów została przedstawiona w [12] oraz dalej rozwijana w [35], [36], [37]. Pojawiły się także koncepcje przedstawienia za pomocą DNA automatu ze stosem [6], [13] czy nawet maszyny Turinga [31] (w tym ostatnim przypadku system nie jest autonomiczny czyli wymaga ciągłej ingerencji laboranta podczas przeprowadzania obliczeń, co niestety przeczy zaleceniom szybkości i masowej równoległości).

Zatem temat molekularnej implementacji pojedynczego modelu o mocy obliczeniowej analogicznej do klasycznych komputerów jest wciąż otwarty.

Kolejną gałęzią obliczeń DNA są systemy oparte o zagadnienia związane z matematyką, szczególnie logiką matematyczną. Pierwsza praca, dotycząca implementacji bramek logicznych za pomocą DNA, została przedstawiona przez Mitsunori Ogiharę i Animesha Raya w 1997 roku [18]. Autorzy udowodnili, że za pomocą DNA można przedstawić standardowy model przetwarzania informacji w informatyce, czyli model sieci logicznych. Wiele prac (między innymi [43], [44]), dotyczyło implementacji prostych systemów wnioskujących (zgodnych z przedstawionym wcześniej problemem „czy dowolne zdanie p jest konsekwencją skończonego zbioru zdań złożonych A , których prawdziwość przyjmujemy?”), które stanowią temat przewodni mojej pracy naukowej. Pierwsze kroki w tej dziedzinie wykonał Olgierd Unold i Maciej Troć w roku 2001 [42], publikując teoretyczny model systemu splatania, który wykorzystuje wcześniejsze koncepcje przedstawione w pracach Shapiro i Benensona [2], [3]. Podobny model, ale tym razem poparty doświadczeniami laboratoryjnymi, został zaproponowany przez zespół badawczy Ehuda Shapiro w roku 2009 [22]. Żaden ze wspomnianych powyżej systemów nie zawiera jednak kompletnej implementacji wnioskowania z użyciem praw klasycznego rachunku zdań. Większość z tych systemów używa wyłącznie zdań atomowych np. p , q , r , nie przewidując możliwości kodowania ich zaprzeczeń np. $\neg p$, $\neg q$, $\neg r$ [22], [42]. Dlatego głównym celem opracowania nowego systemu wnioskującego było zebranie zalet wszystkich dotychczas istniejących modeli oraz wyeliminowanie wad, które zawierają.

Jedną z głównych inspiracji, na bazie której powstała nowa koncepcja molekularnego systemu wnioskującego, przedstawiona została w pracy Olgierda Unolda i Macieja Trocia „Biomolekularne modele obliczeniowe” wydanej w 2005 roku na łamach czasopisma Inżynieria Komputerowa [42]. W pracy tej zawarto trzy różne modele obliczeń: rozszerzenie automatu skończenie-stanowego, ideę obliczania wartości funkcji logicznych w postaci kanonicznej oraz prosty model wnioskowania. Drugą, bardzo ważną inspiracją jest system wnioskowania zaprezentowany w pracy Ehuda Shapiro, Shai Kaplana i Toma Rana „Molecular implementation of simple logic program”, wydanej w 2009 przez Nature Nanotechnologies [22]. W pracy tej zawarto dwie koncepcje systemu wnioskującego, różniące się od siebie kolejnością przeprowadzania obliczeń. Obie opierają się na modelu splatania, czyli naprzemiennych łączeniach i cięciach molekuł. Badania zostały poparte doświadczeniami laboratoryjnymi. Obie wymienione prace [22], [42], stanowiące bardzo dobre odniesienie do nowego systemu wnioskującego, zostały również szczegółowo omówione w rozprawie doktorskiej.

Podstawowe modele obliczeń za pomocą DNA to: model filtrowania (wykorzystujący wyłącznie łączenie się komplementarnych molekuł ze sobą, zapoczątkowany pracą Leonarda Adlemana [1]); model splatania (wykorzystujący naprzemienne działanie enzymu ligazy oraz enzymu restrykcyjnego, opisany po raz pierwszy w roku 1987 przez Toma Heada [8], a następnie rozwinięty przez zespół badawczy Yaakova Benensona w roku 2001 [2]). Zostały one szczegółowo omówione, włącznie z przykładami. Oprócz wspomnianych już prac z dziedziny obliczeń biomolekularnych oraz prac mających największy wpływ na powstanie nowej koncepcji systemu wnioskującego, w rozprawie zostało przedstawionych też wiele

innych artykułów i koncepcji wiążących obliczenia DNA z logiką matematyczną. Za istotne należy uznać prace grupy badawczej prof. dr hab. inż. Jana Mulawki [16], [17], [48], [49], [50], [51], [52], [53], prace dotyczące wykorzystania modelu DSD czyli przemieszczenia nici DNA (z ang. *DNA strand displacement*) [10], [19], [21], [23], [24], [25], [33], [45], [46], [55] i inne [9], [34], [38].

Omawiany w rozprawie model systemu wnioskującego, który jest najważniejszym jej elementem, oparty jest o wspomniany już model komputera DNA – system splatania, wykorzystujący naprzemienne cięcie i łączenie się ze sobą komplementarnych łańcuchów. Używa on enzymu ligazy jako katalizatora reakcji łączenia się oraz enzymu restrykcyjnego *BbvI*, który rozpoznaje konkretny ciąg nukleotydów 5'-GCAGC-3' i rozcina molekułę w odległości 8 (na dolnej nici) i 12 (na górnej nici) nukleotydów od wspomnianej sekwencji rozpoznawalnej, pozostawiając zawsze lepkie końce o długości 4 na nici 3'-5' (patrzac w kierunku cięcia).

Jako podstawowy zbiór aksjomatów przyjmujemy dowolny zbiór zdań logicznych, zapisanych w koniunkcyjnej postaci normalnej. Oznacza to, że każde takie zdanie A jest koniunkcją dowolnej skończonej liczby klauzul B_1, B_2, \dots, B_n , czyli:

$$A \equiv B_1 \wedge \dots \wedge B_n,$$

gdzie dla każdego $k \in \langle 1, \dots, n \rangle$, każda z nich jest alternatywą dowolnej skończonej liczby literałów, czyli:

$$\forall k \in \langle 1..n \rangle \quad B_k \equiv b_{1,k} \vee \dots \vee b_{m,k},$$

gdzie literał jest pojedynczą zmienną zdaniową lub jej negacją [39]. W szczególnych przypadkach indeksy m oraz n mogą wynosić 1. Jeżeli dla danego k , wartość m wynosi 1, to taką klauzulę B_k nazywamy klauzulą faktu (jest to w rzeczywistości pojedynczy literał).

W omawianym systemie wnioskowania każdy literał reprezentowany jest przez unikalną sekwencję czterech nukleotydów (a jego negacja przez sekwencję nukleotydów komplementarnych do tej czwórki), a każdą z klauzul B_i będziemy kodować za pomocą pojedynczej molekuly, której częścią będą czwórki reprezentujące odpowiednie literały. Szczególnym przypadkiem są tutaj klauzule faktu, których kodowanie różni się od kodowania pozostałych klauzul. Każdej z nich odpowiada dokładnie jedna molekula zawierająca sekwencję rozpoznawalną przez enzym restrykcyjny *BbvI*, zatem to właśnie one są odpowiedzialne za właściwy proces wnioskowania. Samo wnioskowanie odbywa się na podstawie zapisanych w koniunkcyjnej postaci normalnej aksjomatów początkowych, zakodowanych molekularnie według określonego schematu (przedstawionego i szczegółowo omówionego w rozprawie). Aby wnioskowanie przebiegało poprawnie, oprócz zakodowanych odpowiednio klauzul, niezbędne jest umieszczenie w próbówce także molekuly terminalnej, która uczestniczy w większości reakcji. Następnym elementem wnioskowania są zapytania, również zapisane w postaci molekuł DNA. Możemy zadawać pytania o istnienie w systemie dowolnej klauzuli faktu, zarówno takiej która znajdowała się w pierwotnym zbiorze aksjomatów jak i takiej, która mogła zostać wyprowadzona podczas wnioskowania. Pojedyncze zapytanie może dotyczyć też koniunkcji kilku faktów bądź ich alternatywy, jeżeli użyjemy do jego zapisania więcej molekuł (jedna molekula reprezentuje

jeden składnik alternatywy), a nawet dowolnego zdania, zapisanego w dysjunkcyjnej postaci normalnej (odwrotnej względem koniunkcyjnej postaci normalnej czyli składającej się z koniunkcji rozdzielonych alternatywami). Podczas reakcji wnioskowania w jednej próbówce możemy zadać większą liczbę zapytań i otrzymać osobne odpowiedzi dla każdego z nich.

Wprowadzenie do systemu spójnika negacji (co jest zasadniczą zmianą względem systemów [22], [42]) jest dużą zaletą, ale niesie też ze sobą konieczność rozpatrzenia sytuacji pojawienia się w tej samej próbówce przeciwnych sobie faktów (np. p oraz $\neg p$). Omawiany system rozpoznaje taką sytuację i w odpowiedni sposób taką ewentualność sygnalizuje. Ze względu na komplementarność unikalnych czwórek reprezentujących literały o przeciwnych wartościach tego samego zdania atomowego, w przypadku pojawienia się ich w tej samej próbówce nastąpi między nimi reakcja, którą nazywam reakcją wykrycia sprzeczności. Katalizatorem tej reakcji jest cząsteczka terminalna, ta sama która bierze udział w reakcjach wnioskowania. Z matematycznego punktu widzenia, jeżeli w danym systemie logicznym pojawią się zaprzeczające sobie fakty, to możemy z niego wywnioskować dowolne zdanie logiczne – w takiej sytuacji odpowiedź o pojawieniu się sprzeczności, możemy potraktować jako pozytywną na wszystkie zadane pytania.

Przedstawiony w pracy system wnioskowania jest w pełni autonomiczny (wymaga udziału laboranta wyłącznie w celu przygotowania molekuł i odczytania odpowiedzi), a jego działanie (po sprowadzeniu aksjomatów wejściowych do koniunkcyjnej postaci normalnej, a zapytań do dysjunkcyjnej postaci normalnej) składa się z trzech podstawowych kroków:

1. Kodowanie aksjomatów wejściowych oraz zapytań za pomocą odpowiednich molekuł DNA według ustalonego i przedstawionego w pracy schematu.
2. Przeprowadzenie reakcji w próbówce w sposób autonomiczny: poszukiwanie sprzeczności, poszerzanie bazy wiedzy zgodnie z regułami wnioskowania *modus ponens* i *modus tollens*, znajdowanie odpowiedzi na zadane pytania.
3. Odczytanie wyników za pomocą elektroforezy w żelu, co kończy proces.

Odpowiedź twierdzącą rozpoznajemy poprzez pojawienie się w próbówce molekuly o konkretnej długości, odpowiadającej danemu zapytaniu i bez lepkich końców. Brak odpowiedzi systemu oznacza informację, że podanego wniosku nie udało się wyprowadzić na podstawie danych aksjomatów początkowych.

Taki system uwzględnia większość praw klasycznego rachunku zdań w sposób automatyczny i dokonuje zgodnego z nimi wnioskowania. Szczegółowo omówione zostały: podwójne zaprzeczenie, reguła kontrapozycji, prawa de Morgana oraz przechodność implikacji. Istnieją jednak pewne ograniczenia dotyczące niektórych sytuacji, które wymagają specjalnego przygotowania – zostały one przedstawione w pracy wraz z ich możliwymi rozwiązaniami. Oprócz dokładnej analizy ograniczeń, opisane zostały również modyfikacje, rozszerzenia i dodatkowe zastosowania systemu, a mianowicie: obliczanie wartości funkcji logicznych, częściowe rozwiązanie problemu SAT, wykorzystanie metody dowodu nie-wprost jako alternatywnego sposobu dowodzenia, a także szczegółowa analiza zwiększenia możliwości systemu w związku z potencjalnym odkryciem nowych, nieznanych dotąd enzymów restrykcyjnych.

W porównaniu z zaprezentowanymi przez inne grupy badawcze wcześniejszymi koncepcjami, można powiedzieć, że ta zdecydowanie rozszerza ich możliwości, a przede wszystkim:

- wprowadza pojęcie negacji,
- aksjomaty początkowe, wyciągane wnioski czy zapytania do systemu mogą być również w formie koniunkcji, alternatyw czy dowolnego zdania złożonego,
- w trakcie wnioskowania korzysta z praw klasycznego rachunku zdań, które są naturalną konsekwencją wprowadzenia negacji, między innymi regułę kontrapozycji czy prawa de Morgana,
- sygnalizuje o ewentualnym pojawieniu się sprzeczności (czyli równoczesnym wystąpieniu dwóch przeciwnych wartości tego samego zdania atomowego) niezależnie czy wynikła ona z błędnie dobranych aksjomatów początkowych czy powstała w trakcie reakcji w wyniku wnioskowania.

Zastosowanych zostało też kilka innowacji związanych z budową molekuł:

- zastosowanie cząsteczek symetrycznych, które pozwalają na wykonanie (zależnie od warunków początkowych) różnych przekształceń tej samej molekuly (np. na podstawie p , $(\neg p \vee q)$ wnioskujemy q (*modus ponens*), podczas gdy na podstawie $\neg q$, $(\neg p \vee q)$ wnioskujemy $\neg p$ (*modus tollens*),
- fragment rozpoznawalny przez enzym restrykcyjny umieszczony został na przeciwległych końcach tej samej molekuly, zatem cięcie w obrębie takiej cząsteczki może pojawić się z obu jej stron (a w niektórych przypadkach jednocześnie),
- cząsteczki mogą posiadać lepkie końce dobrane w taki sposób, że są komplementarne z samymi sobą i w wyniku połączenia się dwóch takich samych molekuł (oczywiście w obrębie jednej próbki wszystkie cząsteczki są zwielokrotnione) „domykają się” tworząc cząsteczkę nieposiadającą lepkich końców, która oznacza konkretną odpowiedź na zadane pytanie.

Pierwsze koncepcje i pomysły rozbudowy istniejących systemów wnioskujących pojawiły się już na pierwszym roku moich studiów doktoranckich, a po raz pierwszy zostały zaprezentowane w roku 2011 jako preprint Uniwersytetu Łódzkiego [26], a następnie wraz z pojawianiem się kolejnych ulepszeń i dalszym rozwojem systemu, w kolejnych publikacjach zarówno polskich [27], [29] jak i mających zasięg międzynarodowy [28], [30]. Ostatnie lata swojej pracy naukowej spędziłem głównie na analizowaniu eksperymentów i staraniem się by udało się je fizycznie przeprowadzić, co zaowocowało najbardziej praktycznym rozdziałem rozprawy, szczegółowo prezentującym trzy wybrane eksperymenty, a także dołączeniem do omawianego systemu i omówieniem koncepcji „biblioteki łańcuchów DNA”, która umożliwia wykorzystywanie takich samych łańcuchów DNA dla różnych obliczeń, a co za tym idzie znacznie ogranicza koszty pojedynczego doświadczenia laboratoryjnego. Faktyczne eksperymenty, mam nadzieję, uda mi się przeprowadzić w przyszłości. W tej pracy natomiast znalazła się szczegółowa analiza ich przygotowania i przeprowadzenia każdego z etapów. Wszystkie wyszczególnione kroki doświadczeń laboratoryjnych zostały skonsultowane i potwierdzone przez specjalistów-genetyków, z którymi współpracuję.

Bibliografia

- [1] ADLEMAN L.: Molecular computation of solutions to combinatorial problems. *Science* 1994; 226: 1021-1024.
- [2] BENENSON Y, PAZ-ELIZUR T, ADAR R, KEINAN E, LIVNEH Z, SHAPIRO E.: Programmable and autonomous computing machine made of biomolecules. *Nature* 2001; 414: 430-434.
- [3] BENENSON Y, ADAR R, PAZ-ELIZUR T, LIVNEH Z, SHAPIRO E.: DNA molecule provides a computing machine with both data and fuel. *PNAS* 2003; 100: 2191-2196.
- [4] BENENSON Y, GIL B, BEN-DOR U, ADAR R, SHAPIRO E.: An autonomous molecular computer for logical control of gene expression. *Nature* 2004; 429: 423-429.
- [5] BENENSON Y.: Biomolecular computing systems: principles, progress and potential. *Nature Reviews Genetics*, 2012; 13: 455-468.
- [6] CAVALIERE M., JONOSKA N., YOGEV S., PIRAN R., KEINAN E., SEEMAN N.: Biomolecular implementation of computing devices with unbounded memory. *Lecture Notes in Computer Science* 2005; 3384: 35-49.
- [7] GRZEGORCZYK A.: Zarys logiki matematycznej. PWN, 1975.
- [8] HEAD T.: Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors. *Bull. Math. Biology*, 1987; 49: 737-759.
- [9] IN-HEE L., JI-YOON P., HAE-MAN J., YOUNG-GYU C., BYOUNG-TAK Z.: DNA Implementation of Theorem Proving with Resolution Refutation in Propositional Logic. *Proceeding DNA8 Revised Papers from the 8th International Workshop on DNA Based Computers: DNA Computing* 2002; 1: 156-167.
- [10] GUO Y., WEI B., XIAO S., YAO D., LI H., XU H., SONG T., LI X., LIANG H.: Recent advances in molecular machines based on toehold-mediated strand displacement reaction. *Quant Biol*; 2017; 5: 25-41.
- [11] KRASIŃSKI T, SAKOWSKI S.: Przegląd modeli i praktycznych implementacji DNA obliczeń. *Studia Informatica*, 2008; 29: 5-31.
- [12] KRASIŃSKI T, SAKOWSKI S.: Extended Shapiro Finite State Automaton. *Foundations of Computing and Decision Science*, 2008; 33: 241-255.
- [13] KRASIŃSKI T, SAKOWSKI S. POPLAWSKI T. Autonomous push-down automaton built on DNA. *Informatica* 2012; 36: 263-276.
- [14] LIPTON R.: DNA solution of hard computational problems. *Science* 1995; 268: 542-545.
- [15] MARTYN A.: DNA Computing. The Encyclopedia of Complexity and System Science. Springer 2008.
- [16] MULA WKA J, WĘGLEŃSKI P, BORSUK P.: Sposób dowodzenia dedukcyjnego za pomocą wnioskowania regresywnego. *Opis Patentowy Rzeczpospolitej Polskiej*, 1997.
- [17] MULA WKA J, WĘGLEŃSKI P, BORSUK P.: Sposób wnioskowania dedukcyjnego z wykorzystaniem drzewa decyzyjnego. *Opis Patentowy Rzeczpospolitej Polskiej*, 1997.
- [18] OGIHARA M, RAY A.: Simulating Boolean circuits on a DNA computer. *RECOMB '97 Proceedings of the first annual international conference on Computational molecular biology*, 1997; 1: 226-231
- [19] ORDÓÑEZ-GUILLÉN N., MARTÍNEZ-PÉREZ I.: Catalytic DNA Strand Displacement Cascades Applied to Logic Programming, in *IEEE Access*, 2019; 7: 100428-100441.
- [20] PAUN G, ROZENBERG G, SALOMAA A.: DNA Computing. *New Computing Paradigms*. Springer-Verlag, 1999.
- [21] QIAN L., WINFREE E.: Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades. *Science* 2011; 6034, 1196-1201.
- [22] RAN T, KAPLAN S, SHAPIRO E.: Molecular implementation of simple logic program. *Nature Nanotechnology*, 2009; 203(2): 1-7.

- [23] RAY K., MONDAL M.: Logical Inference by DNA Strand Algebra, *New Mathematics and Natural Computation*, 2016; 12: 29-44.
- [24] RODRÍGUEZ-PATÓN A, LARREA J, SAINZ DE MURIETA I.: Inference with DNA molecules. In: CALUDE C, HAGIYA M, MORITA K, ROZENBERG G, TIMMIS J.: *UC 2010. LNCS, Springer Heidelberg*, 2010; 6079: 192.
- [25] RODRÍGUEZ-PATÓN A, SAINZ DE MURIETA I, SOSÍK P.: DNA strand displacement system running logic programs. *Biosystems*, 2014; 115: 5–12.
- [26] ROGOWSKI Ł.: Implementacje systemów logicznych za pomocą DNA – koncepcje i pomysły rozbudowy. *WMiI Preprint (Wydział Matematyki i Informatyki Uniwersytetu Łódzkiego)*, 2011.
- [27] ROGOWSKI Ł.: Wnioskowanie logiczne za pomocą DNA. Reguła kontrapozycji, *Studia i Materiały Informatyki Stosowanej*, 2013; 13: 11-18.
- [28] ROGOWSKI Ł.: DNA computer-based search for unsatisfiable formulas, *PhD Interdisciplinary Journal, Special Issue: BioTech Conference*, 2013; 1: 105-110.
- [29] ROGOWSKI Ł, KRASIŃSKI T, SAKOWSKI S, BŁASIAK J, POPLAWSKI T.: Wnioskowanie logiczne przy użyciu DNA, *Postępy Biologii Komórki*, 2013; 4(40): 645-658
- [30] ROGOWSKI Ł, SOSÍK P.: The laws of natural deduction in inference by DNA computer, *The Scientific World Journal*, 2014; doi:10.1155/2014/834237.
- [31] ROTHEMUND P.: A DNA and restriction enzyme implementation of Turing machines. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society*, 1995; 27: 75-120.
- [32] ROZENBERG G, BACK T, KOK J.: *Handbook of Natural Computing*, Springer 2012.
- [33] SAINZ DE MURIETA I.: Inference models in DNA Computing, *Rozprawa doktorska, Departamento de Inteligencia Artificial, Facultad de Informática, Politécnica de Madrid*, 2013.
- [34] SAKAKIBARAY, SUYAMA A.: Intelligent DNA chips: logical operation of gene expression profiles on DNA computers. *Proceedings 11th Workshop on Genome Informatics*, 2000: 33-42.
- [35] SAKOWSKI S.: Rozszerzenie biomolekularnego automatu Shapiro, *Rozprawa doktorska, Wydział Automatyki, Elektroniki i Informatyki, Politechnika Śląska w Gliwicach*, 2010.
- [36] SAKOWSKI S, KRASIŃSKI T, SARNIK J, BŁASIAK J, WALDMAJER J, POPLAWSKI T.: A detailed experimental study of a DNA computer with two endonucleases, *Zeitschrift für Naturforschung*, 2017; 72(7-8): 303-313.
- [37] SAKOWSKI S, KRASIŃSKI T, SARNIK J, BŁASIAK J, WALDMAJER J, POPLAWSKI T.: Biomolecular computers with multiple restriction enzymes, *Genetics and molecular biology*, 2017; 40(4): 860-870.
- [38] SAKAMOTO K., GOUZU H., KOMIYA K., KIGA D., YOKOYAMA S., YOKOMORI T., HAGIVA M.: Molecular computation by DNA hairpin formation. *Science* 2000; 288: 1223–1226.
- [39] SIPSER M.: *Introduction to the Theory of Computation*, 2nd Edition, Int. Edition, 2006.
- [40] SORENI M., YOGEV S., KOSSOV E., SHOHAM Y., KEINAN E.: Parallel biomolecular computation on surfaces with advanced finite automata. *J. Am. Chem. Soc.*, 2005; 127: 3935-3943.
- [41] UNOLD O.: Jak liczy komputer DNA, *Delta – matematyka, fizyka, astronomia, informatyka*, 2008; 7(410): 8–10.
- [42] UNOLD O, TROĆ M.: Biomolekularne modele obliczeniowe. *Inżynieria Komputerowa*, 2005: 73-85.

- [43] UNOLD O, TROĆ M.: Restriction Enzyme Computation. *Proceedings 7th International Work Conference on Artificial and Natural Neural Networks, IWANN 2003, Lecture Notes in Computer Science*, 2003; 2686: 686–693.
- [44] UNOLD O., TROĆ M., DOBOSZ T., TRUSIEWICZ A.: Extended molecular computing model. *WSEAS Transactions on Biology and Biomedicine*, 2004; 1: 15-19.
- [45] WANG F., SU H., XU J., WANG Q., ZHOU X.: High-efficiency and integrable DNA arithmetic and logic system based on strand displacement synthesis. *Nature Communications*, 2019; 10: 5390.
- [46] WANG F., LV H., LI Q., LI J., ZHANG X., SHI J., WANG L., FAN C.: Implementing digital computing with DNA-based switching circuits. *Nature Communications*, 2020; 11: 121.
- [47] WATSON J., CRICK F.: A Structure for Deoxyribose Nucleic Acid. *Nature*, 1953; 171: 737-738.
- [48] WĄSIEWICZ P, MULAŁKA J, PŁUCIENNICZAK A.: Logical operations with DNA strands. *RSCTC'98, Proceedings of 1st International Conference on Rough Sets and Current Trends in Computing*, 1998.
- [49] WĄSIEWICZ P, JANCZAK T, MULAŁKA J, PŁUCIENNICZAK A.: The inference via DNA computing. *Congress on Evolutionary Computation, Washington, USA*, 1999; 2: 988-993.
- [50] WĄSIEWICZ P, BORSUK P, MULAŁKA J, WĘGLEŃSKI P.: Implementation of Data Flow Logical Operations via self-assembly of DNA. *Lecture Notes in Computer Science*, 1999; 1586: 174-182.
- [51] WĄSIEWICZ P, JANCZAK T, MULAŁKA J.: The inference based on molecular computing. *Cybernetics and Systems*, 2000; 31: 283-315.
- [52] WĄSIEWICZ P, MALINOWSKI A, NOWAK R, MULAŁKA J, BORSUK P, WĘGLEŃSKI P, PŁUCIENNICZAK A.: DNA computing: implementation of data flow logical operations. *Future Generations Computer Systems*, 2001; 17: 361-378.
- [53] WĄSIEWICZ P, MULAŁKA J.: Molecular genetic programming. *Soft Computing. Springer*, 2001; 5: 106-113.
- [54] WOOD D.: A DNA Computing Algorithm for Directed Hamiltonian Paths. *To appear in Genetic Programming: Proceedings of the Third Annual Conference, July 22–25, 1998. Morgan Kaufmann*. 1998.
- [55] YURKE B., TURBERFIELD A., MILLS A., SIMMEL F., NEUMANN J.: A DNA-fuelled molecular machine made of DNA. *Nature*, 2000; 406: 605–608.