Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Sławomir Leszczyński
Nr albumu: 146602

Gry Artura-Merlina ze skończoną pamięcią

Praca magisterska
na kierunku INFORMATYKA

Praca wykonana pod kierunkiem
dra hab. Damiana Niwińskiego
Instytut Informatyki

Lipiec 2002

Pracę przedkładam do oceny

Data Podpis autora pracy:

Praca jest gotowa do oceny przez recenzenta

Data Podpis kierującego pracą:

Streszczenie

W pracy badano klasę języków rozpoznawanych przez gry Artura-Merlina w których Artur jest dwukierunkowym probablistycznym automatem skończonym o wielomianowym oczekiwanym czasie działania. Udowodniono, że jeśli doda się logarytmiczne ograniczenie na liczbę alternacji pomiędzy rundami w których Artur zadaje pytania Merlinowi, a rundami w których wykonuje ruchy probablistyczne to gry te rozpoznają wyłącznie języki regularne.

Słowa kluczowe

automaty skończone, automaty probablistyczne, gry Artura-Merlina, teoria złożoności obliczeniowej

Klasyfikacja tematyczna

68. Computer Science
68Q. Theory of computing
68Q15. Complexity classes

Spis treści

Rozdział 1

Wstęp

W informatyce teoretycznej jednym z ważnych zagadnień jest moc obliczeń zrandomizowanych. Ponieważ najsłynniejszy z problemów w tej klasie, czyli pytanie czy klasa BPP równa się PTIME, jest uważany za trudny, niektórzy badacze postanowili przyjrzeć się bardziej ograniczonym modelom obliczeń. Jednym z takich modeli są automaty skończone z ruchami probabilistycznymi, w skrócie PFA. Automaty te badał już Rabin w latach 60tych. Udało mu się wtedy udowodnić, że klasa języków rozpoznawanych przez PFA z głowicą poruszającą się tylko w jedną strone jest równa klasie języków regularnych.

W 1981 roku ku sporemu zaskoczeniu Freivalds [5] podał przykład języka nieregularnego $L_0 = \{0^m 1^m \mid m \geqslant 1\}$, który jest rozpoznawany przez 2PFA czyli PFA z głowicą mogącą przemieszczać się po taśmie w obu kierunkach. Od tego momentu badania nad tym modelem obliczeń nabrały rozpędu. Dwork i Stockmeyer w pracy [4] podali charakteryzację języków rozpoznawanych przez 2PFA oraz pokazali, że język $Center = \{u1v \mid u, v \in \{0,1\}^* \text{ i } |u| = |v|\}$ nie jest rozpoznawany przez 2PFA ale jest rozpoznawany przez tak zwane gry Artura-Merlina w ktorych Artur jest 2PFA. Mamy więc ścisłe zawieranie $2PFA \subsetneq AM(2PFA)$.
Wiadomo było, że język podany przez Freivaldsa wymaga od rozpoznającego go 2PFA co najmniej wykładniczego oczekiwanego czasu działania. W pracy [3] Dwork i Stockmeyer pokazali, że każdy 2PFA rozpoznający język nieregularny musi działać w co najmniej wykładniczym czasie. Mamy wiec rowność $ptime\text{-}2PFA = Regular$. W tej samej pracy zadali pytanie czy także AM(ptime-2PFA) rozpoznaje tylko języki regularne.
Problem ten probowali rozwiązać w pracy [2] Condon, Hellerstein, Pottle i Widgerson. Korzystając z metod liczenia złożoności komunikacyjnej języków przedstawili szacowanie które niestety nie było wystarczające aby problem rozstrzygnąć. Udowodnili jednak, że $AM(ptime\text{-}1PFA) = Regular$ oraz, że jeśli język jest nieregularny to albo on albo jego dopełnienie nie jest rozpoznawany przez grę Artura-Merlina w której Artur jest ptime-2PFA.

W pracy tej pokażemy, że jeśli ograniczymy do logarytmicznej liczbę alternacji pomiędzy rundami w których Artur zadaje pytania Merlinowi, a rundami w których wykonuje ruchy probablistyczne to język rozpoznawany przez taka grę Artura-Merlina (z wielomianowym ograniczeniem na oczekiwany czas działania Artura) jest regularny.

Rozdział 2

Podstawowe pojęcia

2.1. Automaty i gry Artura-Merlina

Definicja 1 `(Dwukierunkowy probablistyczny automat skończony - 2PFA)`
Dwukierunkowym probablistycznym automatem skończonym nazwiemy krotkę $\{Q, \Sigma, \delta, q_0, q_a, q_r\}$ gdzie:

- Q oznacza zbiór stanów automatu
- Σ oznacza alfabet wejściowy
- q_0, q_a i q_r oznaczają odpowiednio stan: początkowy, akceptujący i odrzucający
- δ jest funkcją przejścia zdefiniowaną następująco:

$$\delta : Q \times (\Sigma \cup \{\sharp, \$\}) \times Q \times \{lewo, prawo, zostań\} \to [0, 1]$$

oraz dla każdych ustalonych q i σ suma $\delta(q, \sigma, q', d)$ po wszystkich q' i d równa się 1.

Znaczenie funkcji przejścia jest takie, że automat w stanie q po przeczytaniu litery σ przechodzi do stanu q' i przesuwa głowicę tak jak wskazuje d. Zakładamy, że ze stanów akceptującego i odrzucającego nie ma przejść do innych stanów automatu oraz słowo wejściowe w jest ograniczone symbolami $\sharp$ (z lewej) i \$ (z prawej strony). Ponieważ symbole te nie należą do alfabetu wejściowego automat zawsze wie czy wyszedł poza słowo wejściowe. Zauważmy, że dla każdych ustalonych q i σ funkcja $\delta_{q\sigma}$ zdefiniowana jako $\delta_{q\sigma}(q', d) = \delta(q, \sigma, q', d)$ wyznacza przestrzeń probablistyczną. Ten zbiór przestrzeni indukuje nam dla każdego słowa przestrzeń probablistyczną na ciągach stanów automatu z której teraz skorzystamy. Powiemy, że automat akceptuje słowo z prawdopodobieństwem θ jeśli prawdopodobieństwo zdarzenia będącego ciągiem stanów kończących się stanem akceptującym wynosi θ. Analogicznie określamy prawdopodobieństwo odrzucenia słowa przez automat. Poniżej zdefiniujemy co to znaczy, że automat akceptuje język.

Definicja 2 `(Akceptacja dla 2PFA)`
Niech $L \subseteq \Sigma^*$, M będzie 2PFA z alfabetem wejściowym Σ oraz $0 \leqslant \varepsilon < 1/2$. Powiemy, że M *rozpoznaje* L *z prawdopodobieństwem błędu* ε jeśli

1. dla każdego $x \in L$, Pr[M akceptuje x]$\geqslant 1 - \varepsilon$
2. dla każdego $x \notin L$, Pr[M odrzuca x]$\geqslant 1 - \varepsilon$

M rozpoznaje L jeśli M rozpoznaje L z prawdopodobieństwem błędu ε dla jakiegoś $\varepsilon < 1/2$.

Przez *oczekiwany czas działania* 2PFA na słowie w będziemy rozumieć oczekiwaną liczbę kroków obliczenia na w po wszystkich losowych wyborach automatu. Powiemy, że automat działa w *oczekiwanym czasie* $T(n)$ jeśli dla wszystkich n oraz wszystkich słów w długości n oczekiwany czas działania automatu na w jest ograniczony przez $T(n)$.

Gry Artura-Merlina są rodzajem dowodów interakcyjnych w których strona przekonująca (Merlin) może poznać losowe bity z których już skorzystała strona weryfikująca (Artur). Dowód interakcyjny możemy opisać jako grę pomiędzy dwoma graczami, P - graczem przekonującym i V - graczem weryfikującym. Mając dany język L oraz słowo w gracz V musi zdecydować czy $w \in L$. Może przy tym korzystać z podpowiedzi gracza P. Zadaniem gracza P jest przekonanie gracza V, że słowo w należy do języka L bez względu na to czy tak jest naprawdę. Tak więc gracz V może zadawać pytania graczowi P ale nie powinien wierzyć odpowiedziom. Dowody interakcyjne stają się interesującym modelem obliczeń jeśli gracz weryfikujący nie jest w stanie samodzielnie rozstrzygnąć czy słowo należy do języka. Formalna definicja gry jest następująca:

Definicja 3 (`Gra Artura-Merlina`)
Protokół dla dwóch uczestników nazywamy grą Artura-Merlina jeśli:

1. Artur i Merlin są maszynami ze skończonym sterowaniem i własnymi taśmami roboczymi
2. istnieje taśma wejściowa z której mogą czytać zarówno Artur jak i Merlin
3. istnieje taśma komunikacyjna na której Artur może pisać, a Merlin czytać, taśma ta jest jednokierunkowa
4. istnieje taśma komunikacyjna na której Merlin może pisać, a Artur czytać, taśma ta jest jednokierunkowa
5. istnieje taśma z bitami losowymi z której może czytać tylko Artur, taśma ta jest jednokierunkowa
6. Merlin dowiaduje się jaka jest wartość bitów losowych dopiero w momencie gdy przeczytał je Artur

Niech $\varepsilon < 1/2$. Powiemy, że gra Artura-Merlina rozpoznaje język L jeśli:

- dla każdego słowa $w \in L$ Artur akceptuje w z prawdopodobieństwem większym od $1-\varepsilon$,
- dla każdego słowa $w \notin L$ oraz każdej maszyny mogącej pełnić rolę Merlina z którą komunikuje się Artur, prawdopodobieństwo, że Artur zaakceptuje w jest mniejsze od ε.

W dalszej części pracy ograniczymy się do gier Artura-Merlina w których Artur jest 2PFA z wielomianowym ograniczeniem na oczekiwany czas działania oraz logarytmiczną ($O(\log |w|)$) liczbą rund w których Artur komunikuje się z Merlinem.

Alternatywną charakteryzacją gier Artura-Merlina w których Artur jest PFA są niedeterministyczne PFA (patrz [2]). Ponieważ ta definicja czyni dowód łatwiejszym będziemy z niej korzystać w dalszej części pracy.

Definicja 4 `(Dwukierunkowy niedeterministyczny probablistyczny automat skończony - 2NPFA)`
Dwukierunkowym niedeterministycznym probablistycznym automatem skończonym nazwiemy krotkę $\{Q, \Sigma, \delta, q_0, q_a, q_r\}$ gdzie:

- Q oznacza zbiór stanów automatu i dzieli się na zbiór stanów niedeterministycznych N oraz zbiór stanów probablistycznych R
- Σ oznacza alfabet wejściowy
- q_0, q_a i q_r oznaczają odpowiednio stan: początkowy, akceptujący i odrzucający
- δ jest funkcją przejścia zdefiniowaną następująco:

$$\delta : R \times (\Sigma \cup \{\sharp, \$\}) \times Q \times \{lewo, prawo, zostań\} \to [0, 1]$$

$$\delta : N \times (\Sigma \cup \{\sharp, \$\} \times Q \times \{lewo, prawo, zostań\} \to \{0, 1\}$$

Dla każdych ustalonych $q \in R$ i σ suma $\delta(q, \sigma, q', d)$ po wszystkich q' i d równa się 1.

Znaczenie funkcji przejścia dla stanów probablistycznych jest identyczne jak w przypadku 2PFA. Natomiast dla stanów niedeterministycznych automat niedeterministycznie wybiera q' takie, że $\delta(q, \sigma, q', d) = 1$. Podobnie jak poprzednio zakładamy, że słowo wejściowe w jest ograniczone symbolami $\sharp$ i \$.

Definicja 5 `(Strategia niedeterministyczna)`
Niech $w = w_0 w_1 w_n \dots w_{n+1}$ będzie słowem w którym $w_0 = \sharp$, a $w_{n+1} = \$$. Niedeterministyczną strategią na w nazwiemy funkcję

$$S_w : N \times \{0, \dots, n+1\} \to Q \times \{-1, 0, 1\}$$

taką, że $\delta(q, \sigma, q', d) = 1$ jeśli $S_w(q, j) = (q', d)$ i $w_j = \sigma$

Innymi słowy funkcja S_w określa niedeterministyczny wybór jakiego dokona automat (działający zgodnie z tą strategią) w stanie $q \in N$ czytając literę σ na pozycji j. Gdy mamy już zdefiniowaną strategię możemy powiedzieć co to znaczy, że automat akceptuje język.

Definicja 6 `(Akceptacja dla 2NPFA)`
Niech $L \subseteq \Sigma^*$, M będzie 2NPFA z alfabetem wejściowym Σ oraz $0 \leqslant \varepsilon < 1/2$. Powiemy, że *$M$ rozpoznaje L z prawdopodobieństwem błędu ε* jeżeli

1. dla każdego $x \in L$, istnieje strategia S_x przy której automat akceptuje słowo z prawdopodobieństwem większym od $1 - \varepsilon$
2. dla każdego $x \notin L$ i dla wszystkich strategii S_x automat odrzuca słowo z prawdopodobieństwem większym $1 - \varepsilon$

M rozpoznaje L jeśli M rozpoznaje L z prawdopodobieństwem błędu ε dla jakiegoś $\varepsilon < 1/2$.

Podobnie jak dla 2PFA definiujemy *oczekiwany czas działania*. Tym razem jest to maksimum po wszystkich strategiach z oczekiwanej liczby kroków obliczenia na w dla możliwych losowych wyborów automatu.

Zdefiniowaną powyżej strategię możemy nazwać strategią pozycyjną - ponieważ zależy ona tylko od pozycji w słowie na jakiej znajduje się głowica automatu. Ogólniejszym typem strategii są strategie zależne od historii. W pracy [1] Anne Condon pokazała, że jeśli jakiś język jest akceptowany przez 2NPFA ze strategią zależną od historii to jest także akceptowany ze strategią pozycyjną. Dlatego poniżej ograniczymy się do strategii pozycyjnych.

2.2. Miara nieregularności

Definicja 7 (Relacja *n-podobieństwa)*
Niech $L \subseteq \Sigma^*$ oraz n będzie dodatnią liczbą naturalną. Relację $\sim_{L,n}$ nazwiemy relacją *n-podobieństwa* jeśli dla każdej pary słów w i w' o długości mniejszej lub równej n oraz dla każdego słowa v takiego, że $|wv| \leqslant n$ oraz $|w'v| \leqslant n$ słowo $wv \in L$ wtedy i tylko wtedy, gdy $w'v \in L$.

Słowa w i w' są *n-niepodobne* (oznaczenie $\nsim_{L,n}$) jeśli nie są *n-podobne*. Zauważmy, że jeśli słowa w i w' są *n-niepodobne* to musi istnieć krótkie słowo je rozróżniające. Czyli musi istnieć słowo v takie, że $|wv| \leqslant n$ oraz $|w'v| \leqslant n$ oraz $wv \in L$ i $w'v \notin L$ lub $wv \notin L$ i $w'v \in L$.

Definicja 8 (Miara nieregularności)
Przez miarę nieregularności $N_L(n)$ oznaczamy maksymalne k takie, że istnieje k różnych słów $w_1, \ldots, w_k$ które są parami *n-niepodobne*, to znaczy $w_i \nsim_{L,n} w_j$ dla wszystkich $1 \leqslant i < j \leqslant k$.

Prostą konsekwencją twierdzenia Myhilla-Neroda jest następujący fakt:

Fakt 2.2.1 *$N_L(n)$ jest ograniczone z góry przez stałą wtedy i tylko wtedy gdy L jest regularny.*

Poniższe ograniczenie pochodzi z pracy Kaneprsa i Freivaldsa [7]. Jest ono wzmocnieniem rezultatu z pracy [3].

Lemat 2.2.1 *Jeśli L nie jest regularny to $N_L(n) \geqslant \frac{n+3}{2}$ dla nieskończenie wielu n.*

2.3. Łańcuchy Markowa

Będziemy wykorzystywać łańcuchy Markowa ze skończoną przestrzenia stanów $\{1, 2, \ldots, m\}$. Konkretny łańcuch jest zdefiniowany przez macierz prawdopodobieństw przejść $P = \{p_{ij}\}_{i,j=1}^{m}$. Jeśli łańcuch jest w stanie i to w następnym kroku przesuwa się do stanu j z prawdopodobieństwem równym p_{ij}. Rozpatrywane przez nas łańcuchy będą miały wyróżniony stan początkowy 1 oraz dwa stany absorbujące $m-1$ oraz m. Oznaczmy przez $a(P)$ prawdopodobieństwo, że łańcuch P jest absorbowany w stanie m przy uruchomieniu w stanie początkowym. Przez $t(P)$ oznaczmy oczekiwany czas absorpcji (w stanie $m-1$ lub m). Interesuje nas jak będzie się zmieniało prawdopodobieństwo $a(P)$ przy małych zmianach prawdopodobieństw przejść.

Definicja 9 (*β-bliskość*)
Niech $\beta \geqslant 1$. Powiemy, że liczby p i p' są *β-bliskie* jeśli:

1. $p = p' = 0$, lub
2. $p > 0, p' > 0$ oraz $\beta^{-1} \leqslant \frac{p}{p'} \leqslant \beta$

Dwa łańcuchy Markowa P i P' określimy jako *β-bliskie* jeśli ich prawdopodobieństwa przejść p_{ij} oraz p'_{ij} są *β-bliskie*. W pracy [6] Greenberg i Weiss udowodnili następujący lemat:

Lemat 2.3.1 *Niech P i P' będą dwoma β-bliskimi łańcuchami Markowa o m stanach. Wtedy a(P) i a(P') są β^{2m}-bliskie.*

Definicja 10 (*β-bliskość modulo λ*)
Niech $\beta \geqslant 1$ i $\lambda \geqslant 0$. Powiemy, że liczby p i p' są *β-bliskie modulo λ* jeśli:

1. $p \leqslant \lambda$ i $p' \leqslant \lambda$ lub

2. $p > \lambda$ i $p' > \lambda$ oraz p i p' są β-*bliskie*

Podobnie mówimy, że dwa łańcuchy Markowa są β-*bliskie modulo* λ jeśli ich prawdopodobieństwa przejść są β-*bliskie modulo* λ.

Nieco zmodyfikowaną wersję lematu 2.3.1 udowodnili w swojej pracy [3] Dwork i Stockmeyer.

Lemat 2.3.2 *Niech P i P' będą dwoma β-bliskimi modulo λ łańcuchami Markowa o m stanach. Przez t oznaczmy maksimum z $t(P)$ i $t(P')$. Wtedy*

$$a(P') \geqslant (1 - 2\lambda m^3)\beta^{-2m} a(P) - 4\sqrt{\lambda m t}$$

Dla potrzeb naszej pracy wprowadzimy nowe pojęcie.

Definicja 11 (β-*bliskość modulo* (λ, η))
Powiemy, że dwa m stanowe łańcuchy Markowa P i P' są β-*bliskie modulo* (λ, η) jeśli istnieje taki podział macierzy przejść $m \times m$ na dwa zbiory U_λ i V_η, że odpowiadające sobie elementy zbioru $P(U_\lambda)$ i $P'(U_\lambda)$ są β-*bliskie modulo* λ oraz odpowiadające sobie elementy zbioru $P(V_\eta)$ i $P'(V_\eta)$ są β-*bliskie modulo* η.

Lemat 2.3.3 *Niech P i P' będą dwoma β-bliskimi modulo (λ, η) łańcuchami Markowa o m stanach. Przez $t_\lambda(P)$ oznaczmy oczekiwaną liczbę przejść łańcucha P przez elementy U_λ zbioru, a przez $t_\eta(P)$ oznaczmy oczekiwaną liczbę przejść łańcucha P przez elementy V_η zbioru. Niech $t_\lambda = max(t_\lambda(P), t_\lambda(P'))$,$t_\eta = max(t_\eta(P), t_\eta(P'))$ oraz $\nu = max(\lambda, \eta)$ Wtedy:*

$$a(P') \geqslant (1 - 2\nu m^2)\beta^{-2m} a(P) - 4\sqrt{\lambda m t_\lambda} - 4\sqrt{\eta m t_\eta}$$

Dowód:
Zmodyfikujemy dowód szacowania prawdopodobieństwa absorpcji dla łańcuchów β-*bliskich modulo* λ z pracy [3]. Powiemy, że liczba p jest θ-mała jeśli $p \leqslant \theta$. Idea dowodu jest następująca: najpierw przekształcimy łańcuchy P i P' w R i R' przez zmianę wszytkich λ i η-małych prawdopodobieństw przejść na zero. Pokażemy też, że prawdopodobieństwo $a(P)$ nie różni się zbytnio od $a(R)$ (to samo szacowanie będziemy mogli zastosować do P' i R'). Następnie możemy zastosować lemat 2.3.1 do łańcuchów R i R'.

Niech Δ będzie zdarzeniem polegającym na tym, że P jest zaabsorbowane w stanie m, oraz niech Φ będzie takim zdarzeniem, że P jest zaabsorbowane (w stanie $m-1$ lub m) zanim nastąpi przejście λ-małe ze zbioru U_λ lub przejście η-małe ze zbioru V_η. Przez Δ' oraz Φ' oznaczmy analogiczne zdarzenia dla procesu P'. Niech A będzie zmienną losową określającą oczekiwaną liczbę przejść przez elementy zbioru U_λ przed absorpcją, a B zmienną określającą liczbę przejść przez elementy zbioru V_η przed absorpcją. Niech $\overline{\Phi}$ będzie dopełnieniem Φ. Oznaczmy:

$$b_\lambda = (\lambda m t)^{-1/2}$$

$$b_\eta = (\eta m t)^{-1/2}$$

Ponieważ $E(A) = t_\lambda(P) \leqslant t_\lambda$ oraz $E(B) = t_\eta(P) \leqslant t_\eta$ zachodzą następujące nierówności:

$$\Pr[A > b_\lambda t_\lambda] \leqslant b_\lambda^{-1}$$
$$\Pr[B > b_\eta t_\eta] \leqslant b_\eta^{-1}$$

Z definicji Φ oraz nierówności $(1-x)^y \geqslant 1 - xy$ $(\star)$ która jest prawdziwa dla wszystkich $0 \leqslant x \leqslant 1, y \geqslant 1$ otrzymujemy:

$$\Pr[\overline{\Phi} \mid A \leqslant b_\lambda t_\lambda] \leqslant 1 - (1-\lambda m)^b_\lambda t_\lambda \leqslant \lambda m b_\lambda t_\lambda$$
$$\Pr[\overline{\Phi} \mid B \leqslant b_\eta t_\eta] \leqslant 1 - (1-\eta m)^b_\eta t_\eta \leqslant \eta m b_\eta t_\eta$$

Możemy więc napisać szacowanie

$$\Pr[\overline{\Phi}] \leqslant b_\lambda^{-1} + \lambda m b_\lambda t_\lambda + b_\eta^{-1} + \eta m b_\eta t_\eta \leqslant 2\sqrt{\lambda m t_\lambda} + 2\sqrt{\eta m t_\eta}$$

Z definicji prawdopodobieństwa warunkowego

$$\Pr[\Delta] = \Pr[\Delta \mid \Phi](1 - \Pr[\overline{\Phi}]) + \Pr[\Delta \mid \overline{\Phi}]\Pr[\overline{\Phi}]$$

możemy wyprowadzić

$$\Pr[\Delta \mid \Phi] - \Pr[\overline{\Phi}] \leqslant \Pr[\Delta] \leqslant \Pr[\Delta \mid \Phi] + \Pr[\overline{\Phi}]$$

Zauważmy, że $a(P) = \Pr[\Delta]$ z definicji. Zdefiniujemy łańcuchy Markowa R oraz R' takie, że $a(R) = \Pr[\Delta \mid \Phi]$ i $a(R') = \Pr[\Delta' \mid \Phi']$. Dzięki temu będziemy mieli szacowania

$$|a(P) - a(R)| \leqslant 2\sqrt{\lambda m t_\lambda} + 2\sqrt{\eta m t_\eta} \tag{2.1}$$
$$|a(P') - a(R')| \leqslant 2\sqrt{\lambda m t_\lambda} + 2\sqrt{\eta m t_\eta} \tag{2.2}$$

Niech σ_i będzie sumą $p_{ij} \in U_\lambda$ po wszystkich j takich, że p_{ij} jest λ-małe. Niech τ_i będzie sumą $p_{ij} \in V_\eta$ po j takich, że p_{ij} jest η-małe. Prawdopodobieństwa przejść w łańcuchu R są określone w sposób następujący:

- jeśli $p_{ij} \in U_\lambda$ jest λ-małe to $r_{ij} = 0$
- jeśli $p_{ij} \in V_\eta$ jest η-małe to $r_{ij} = 0$
- jeśli $p_{ij} \in U_\lambda$ nie jest λ-małe to $r_{ij} = \frac{p_{ij}}{1-\sigma_i}$
- jeśli $p_{ij} \in V_\eta$ nie jest η-małe to $r_{ij} = \frac{p_{ij}}{1-\tau_i}$

Łańcuch R' jest zdefiniowany analogicznie. Zauważmy, że jeśli $r_{ij} \neq 0$ to p_{ij} i r_{ij} są $(1 - max(\sigma_i, \tau_i))^{-1}$ bliskie. Ponieważ $\sigma_i \leqslant \lambda m$ oraz $\tau_i \leqslant \eta m$ to możemy napisać, że p_{ij} i r_{ij} są $(1-\nu m)^{-1}$ bliskie, a z tego wynika, że łańcuchy R i R' są $\beta(1-\nu m)^{-1}$ bliskie. Korzystając z lematu 2.3.1 oraz szacowań 2.1 i 2.2 możemy oszacować $a(P')$ względem $a(P)$ tak jak chcieliśmy.

$$\begin{aligned}
a(P') &\geqslant a(R') - 2\sqrt{\lambda m t_\lambda} - 2\sqrt{\eta m t_\eta} \\
&\geqslant \left(\frac{\beta}{1-\nu m}\right)^{-2m} a(R) - 2\sqrt{\lambda m t_\lambda} - 2\sqrt{\eta m t_\eta} \\
&\geqslant (1 - 2\nu m^2)\beta^{-2m} a(R) - 2\sqrt{\lambda m t_\lambda} - 2\sqrt{\eta m t_\eta} \qquad \text{z nierówności } (\star) \\
&\geqslant (1 - 2\nu m^2)\beta^{-2m}(a(P) - 2\sqrt{\lambda m t_\lambda} - 2\sqrt{\eta m t_\eta}) - 2\sqrt{\lambda m t_\lambda} - 2\sqrt{\eta m t_\eta} \\
&\geqslant (1 - 2\nu m^2)\beta^{-2m} a(P) - 4\sqrt{\lambda m t_\lambda} - 4\sqrt{\eta m t_\eta}
\end{aligned}$$

□

W dalszej części pracy zakładamy, że podstawą wszystkich logarytmów jest 2.

Rozdział 3

Dwukierunkowe NPFA z logarytmiczną liczbą alternacji

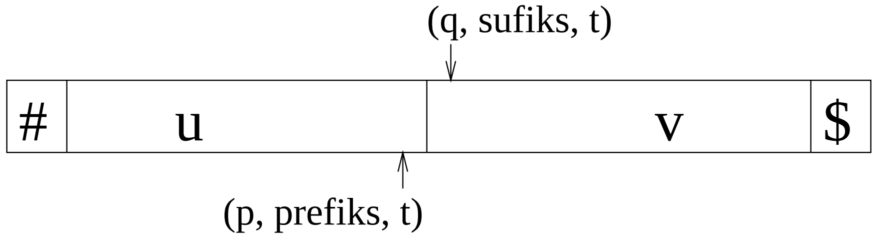
Twierdzenie 3.0.4 `(Siła wyrazu ptime-2NPFA z logarytmiczną alternacji)` *Dwukierunkowe niedeterministyczne probablistyczne automaty skończone działające w oczekiwanym czasie wielomianowym, z logarytmicznym ($O(\log n)$) ograniczeniem na liczbę alternacji między rundami probablistycznymi a niedeterministycznymi rozpoznają dokładnie klasę języków regularnych.*

Dowód:
Będziemy dowodzić nie wprost. Niech M będzie 2NPFA o logarytmicznej liczbie alternacji i oczekiwanym czasie działania $T(n)$. Niech L będzie językiem nieregularnym rozpoznawanym przez M. Ustalmy pewne n takie, że $N_L(n) \geqslant \frac{n+3}{2}$ oraz maksymalny zbiór W słów parami *n-niepodobnych*.

Będziemy symulować obliczenie automatu M na słowie uv za pomocą pewnego łańcucha Markowa H_{uv}. Przyjmijmy, że M ma c stanów, wtedy łańcuch Markowa będzie miał $4c+3$ stanów: trzy wyróżnione stany o etykietach Początkowy, Akceptujący i Odrzucający oraz $4c$ stanów etykietowanych (q, k, t) gdzie $q \in Q$ jest stanem automatu M, $k \in \{prefiks, sufiks\}$,a $t \in \{\textit{bez-alternacji}, alternacja\}$. Stan $(q, prefiks, t)$ odpowiada automatowi M będącemu w stanie q z głowicą ustawioną na ostatnim symbolu słowa $\sharp u$, natomiast stan $(q, sufiks, t)$ automatowi M w stanie q z głowicą na pierwszym symbolu słowa $v\$$. Stany Akceptujący i Odrzucający są stanami absorbującymi łańcucha H_{uv}.

Ustalmy niedeterministyczną strategię na słowie uv. Pojedyńcze przejście łańcucha H_{uv} odpowiada obliczeniu automatu M na prefiksie $\sharp u$ lub sufiksie $v\$$ słowa $\sharp uv\$$. Jeśli automat M uruchomiony w stanie początkowym po pewnej liczbie kroków obliczenia na prefiksie $\sharp u$ przejdzie do stanu q (gdzie q nie jest końcowym stanem automatu M) oraz ustawi głowicę na pierwszym symbolu słowa $v\$$, to łańcuch Markowa wykona przejście ze stanu Początkowy do stanu $(q, sufiks, t)$. Wartość t zależy od tego, czy w czasie obliczenia M na prefiksie $\sharp u$ automat przechodził tylko przez stany tego samego rodzaju (wtedy $t = \textit{bez-alternacji}$) czy też nastąpiła podczas tego obliczenia co najmniej jedna alternacja ze stanu probablistycznego do niedeterministycznego lub na odwrót (wtedy $t = alternacja$).

Prawdopodobieństwo przejścia ze stanu Początkowy łańcucha H_{uv} do stanu $(q, sufiks, t)$ jest równe odpowiedniemu prawdopodobieństwu tak zdefiniowanego obliczenia M na słowie u. Jeśli automat M z prawdopodobieństwem p przejdzie do stanu akceptującego przy obliczeniu



Rysunek 3.1: Stany łańcucha Markowa dla słowa uv

na słowie u, to łańcuch Markowa z prawdopodobieństwem p przejdzie ze stanu Poczatkowy do Akceptujący (to samo zachodzi dla stanu odrzucającego automatu M i Odrzucającego łańcucha Markowa).

Prawdopodobieństwa przejść ze stanów $(q, prefiks, t)$ oraz $(q, sufiks, t)$ określa się podobnie. Dlatego podamy odpowiednią definicje dla przejść ze stanów $(q, prefiks, t)$. Jeśli automat M uruchomiony na ostatnim symbolu słowa $\sharp u$ po pewnej liczbie kroków obliczenia na słowie $\sharp u$ nie dokonując (dokonując) po drodze alternacji między rundami probablistycznymi i deterministycznymi z prawdopodobieństwem p dojdzie do stanu q' i jego głowica znajdzie się na pierwszym symbolu słowa $v\$$, to w łańcuchu Markowa przejście ze stanu $(q, prefiks, t)$ do stanu $(q', sufiks, t)$ $((q', sufiks, alternacja))$ nastąpi także z prawdopodobieństwem p. Jeśli q' jest stanem końcowym przejście nastąpi do stanów Akceptujący lub Odrzucający.

Oznaczmy:

$$d = (4c + 3)^2$$
$$\eta = (\log(n))^{-2}$$
$$\lambda = (T(n))^{-2}$$
$$\mu = \frac{\log(1 - \varepsilon) + 1}{d}$$

Podzielmy zbiór przejść łańcucha Markowa na dwa zbiory U_λ i V_η. Do zbioru U_λ będą należeć wszystkie przejścia do stanów postaci $(q, k, \textit{bez-alternacji})$, a do zbioru V_η pozostałe przejścia. Zauważmy, że o ile przy obliczeniu łańcucha Markowa liczba kroków przez przejścia ze zbioru V_η może być duża (może być wiele przejść ze stanu postaci $(q, k, alternacja)$ do stanu $(q', k', alternacja)$) to liczba przejść w których alternacja naprawdę nastąpiła jest co najwyżej logarytmiczna. Przejścia w których alternacja nie nastąpiła odpowiadają niedeterministycznym rundom obliczenia automatu M i mają zawsze (przy ustalonej strategii) wartość 1. Dlatego liczba przejść tego rodzaju nie wpływa na to z jakim prawdopodobieństwem łańcuch wejdzie w stan Akceptujący.
Zdefiniujemy teraz relację równoważności $\equiv$ na słowach w W w następujący sposób: $w \equiv w'$ wtedy i tylko wtedy gdy:

1. dla wszystkich $p \in U_\lambda$ $H_w(p) \leqslant \lambda$ wttw gdy $H_{w'}(p) \leqslant \lambda$
2. dla wszystkich $p \in V_\eta$ $H_w(p) \leqslant \eta$ wttw gdy $H_{w'}(p) \leqslant \eta$

Takich klas abstrakcji jest nie więcej niż 2^d. Niech E będzie najliczniejszą z nich. Mamy więc $|E| \geqslant N_L(n)/2^d$. Wykonamy teraz rzutowanie łańcucha Markowa na produkt przestrzeni

$[\lambda, 1]^{|U'_\lambda|}$ oraz $[\eta, 1]^{|V'_\eta|}$, gdzie $U'_\lambda \subseteq U_\lambda$ jest zbiorem przejść (elementów łańcucha) większych od λ a $V'_\eta \subseteq V_\eta$ jest zbiorem przejść większych od η. Następnie do elementów tego produktu przestrzeni zastosujemy po współrzędnych funkcję log, otrzymując w ten sposób nową przestrzeń $[\log\lambda, 0]^{|U'_\lambda|} \times [\log\eta, 0]^{|V'_\eta|}$. Dzieląc przedziały $[\log\lambda, 0]$ oraz $[\log\eta, 0]$ na kawałki długości μ podzielimy przestrzeń $[\log\lambda, 0]^{|U'_\lambda|} \times [\log\eta, 0]^{|V'_\eta|}$ na co najwyżej $\lceil(-\log\lambda)/\mu\rceil^{|U'_\lambda|} \cdot \lceil(-\log\eta)/\mu\rceil^{|V'_\eta|}$ komórek o boku rozmiaru co najwyżej μ. Rzuty komórek na podprzestrzeń $[\log\lambda, 0]^{|U'_\lambda|}$ będziemy nazywać λ-komórkami, natomiast rzuty na podprzestrzeń $[\log\eta, 0]^{|V'_\eta|}$ nazywać η-komórkami. Powiemy, że łańcuch Markowa znajduje się w λ-komórce θ jeśli po zastosowaniu do niego rzutowania złożonego z funkcją logarytmiczną wynik zrzutowany na podprzestrzeń $[\log\lambda, 0]^{|U'_\lambda|}$ znajduje się w θ. Analogicznie określamy przynależność łańcucha do η-komórki. Zauważmy, że zmiana strategii automatu może mieć wpływ na to w jakiej η-komórce znajduje się symulujący go łańcuch Markowa, nie ma natomiast wpływu na to w jakiej λ-komórce się on znajduje.

Ponieważ $T(n)$ jest funkcją wielomianową to spełniona jest poniższa nierówność:

$$\lceil(-\log\lambda)/\mu\rceil^d \cdot 2^{\lceil(-\log\eta)/\mu\rceil^d} \leqslant N_L(n)/2^d \leqslant |E| \tag{3.1}$$

Ale wtedy z zasady szufladkowej wynika, że w E muszą istnieć dwa słowa u i u' oraz rozróżniający je sufiks v takie, że dla każdej pary strategii na uv i $u'v$ odpowiednie łańcuchy Markowa znajdują się w tej samej λ-komórce, a także zbiór η-komórek θ takich, że istnieje strategia na słowie uv dla której łańcuch Markowa znajduje się w θ musi być równy podobnie określonemu zbiorowi η-komórek dla strategii na słowie $u'v$. Z tych dwóch faktów możemy wyprowadzić:

1. dla każdej strategii S_1 na słowie uv oraz każdej strategii S_2 na słowie $u'v$ łańcuchy Markowa $H_{S_1}(uv)$ i $H_{S_2}(u'v)$ znajdują się w tej samej λ-komórce,

2. dla każdej ustalonej strategii S_1 na słowie uv możemy znaleźć taką strategię S_2 na słowie $u'v$, że łańcuchy Markowa $H_{S_1}(uv)$ oraz $H_{S_2}(u'v)$ znajdą się w tej samej η-komórce,

3. dla każdej ustalonej strategii S_2 na słowie $u'v$ możemy znaleźć taką strategię S_1 na słowie uv, że łańcuchy Markowa $H_{S_1}(uv)$ oraz $H_{S_2}(u'v)$ znajdą się w tej samej η-komórce.

Oznacza to tyle, że dla każdej strategii na słowie uv istnieje 2^μ-*bliska modulo* (λ, η). strategia na słowie $u'v$ i na odwrót. Jeśli bez zmniejszenia ogólności założymy, że $uv \in L$ to korzystając z lematu 2.3.3 możemy oszacować prawdopodobieństwo akceptacji $u'v$ przez M.

$$\begin{aligned}
\Pr[M \text{ akceptuje } u'v] &\geqslant (1 - 2\eta m^2)2^{-2\mu m}(1-\varepsilon) - 4\sqrt{\lambda m T(n)} - 4\sqrt{\eta m \log(n)} \\
&\geqslant (1 - 2m^2/\log(n))2^{-(\log(1-\varepsilon)+1)/2}(1-\varepsilon) - 4\sqrt{m/T(n)} - 4\sqrt{m/\log(n)} \\
&= (1 - 2m^2/\log(n))\sqrt{(1-\varepsilon)/2} - 4\sqrt{m/T(n)} - 4\sqrt{m/\log(n)} \\
&> 1/2
\end{aligned}$$

Ostatnia nierówność wynika z tego, że $\sqrt{(1-\varepsilon)/2} > 1/2$, a mianowniki pozostałych ułamków w poprzednim wyrażeniu mogą być dowolnie duże. Ponieważ słowo $u'v$ nie należy do L więc powinno być akceptowane z prawdopodobieństwem mniejszym od ε. Otrzymujemy więc sprzeczność.

□

Rozdział 4

Podsumowanie

W pracy pokazaliśmy, że dwukierunkowe niedeterministyczne probablistyczne automaty skończone działające w oczekiwanym czasie wielomianowym z nie więcej niż logarytmiczną liczbą alternacji między rundami probablistycznymi, a niedeterministycznymi rozpoznają dokładnie klasę języków regularnych. Alternatywne sformułowanie wyniku jest takie, że gry Artura-Merlina w których Artur jest ptime-2PFA z co najwyżej logarytmiczną liczbą alternacji między rundami w których Artur zadaje pytania Merlinowi, a rundami w których wykonuje ruchy probablistyczne rozpoznają dokładnie klasę języków regularnych.

Skonstruowany przez nas dowód da się też zastosować do wzmocnionej wersji twierdzenia. Łatwo zauważyć, że szacowanie 3.1 zachodzi także dla przypadku gdy η jest funkcją polilogarytmiczną. Rzeczywiście zachodzi następujący fakt.

Fakt 4.0.1 *Dwukierunkowe niedeterministyczne probablistyczne automaty skończone działające w oczekiwanym czasie wielomianowym, z polylogarytmicznym ($O(poly(\log n))$) ograniczeniem na liczbę alternacji między rundami probablistycznymi, a niedeterministycznymi rozpoznają dokładnie klasę języków regularnych.*

Wydaje się, że podobna technika dowodzenia może być skuteczna także przy bardziej ogólnej definicji automatu. Narzucającym się do zbadania przypadkiem są alternujące probablistyczne automaty skończone.

Spis literatury

[1] A. Condon, Computational models of Games, MIT Press, 1989

[2] A. Condon, L. Hellerstein, S. Pottle i A. Widgerson, On the power of finite automata with both nondeterministic and probabilistic states, STOC 1994

[3] C. Dwork i L. Stockmeyer, A time complexity gap for 2-way probabilistic finite automata, SIAM Journal of Computing, Vol. 19, No. 6, 1990, pp. 1011-1023

[4] C. Dwork i L. Stockmeyer, Finite state verifiers I: The power of interaction, Journal of the ACM, Vol. 39, No. 4, 1992, pp. 800-828

[5] R. Freivalds, Probabilistic two-way machines, Proc. International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, Springer-Verlag, New York, 1981, Vol. 118, pp. 33-45

[6] A.G. Greenberg, A. Weiss, A lower bound for probabilistic algorithm for finite state machines, J. Comput. Syst. Sci., 1986, Volume 33, pp. 88-105

[7] J. Kaneps, R. Freivailds, Minimal nontrivial space complexity of probabilistic one-way Turing machines, Proc. of the Conference on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, Springer-Verlag, 1990, Vol. 452, pp. 355-361