Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Tomasz Fryderyk Urbański
Nr albumu: 153507

O pewnych zagadnieniach złożoności obliczeniowej w sieciach boolowskich

Praca magisterska
na kierunku **INFORMATYKA**

Praca wykonana pod kierunkiem
prof. dr hab. Damiana Niwińskiego
Instytut Informatyki

Wrzesień 2000

Podpis autora:

Podpis promotora:

AMS Subject Classification 94C10 68Q15

Streszczenie

....

Słowa kluczowe: obwody boolowskie, dolne oszacowania, funkcje boolowskie, klasa AC^0

1 Wstęp

Wyniki dotyczące złożoności obliczeniowej zbioru liczb pierwszych były prezentowane w bardzo wielu pracach, jednak większość z nich dotyczyła górnych ograniczeń złożonościowych tego zbioru. Niestety, niewiele wiadomo o dolnych ograniczeniach złożoności problemu pierwszości. Jednym z nielicznych faktów znanych jest dolne oszacowanie dla złożoności pamięciowej, mówiące, że potrzebna jest przynajmniej logarytmiczna pamięć w celu rozpoznawania liczb pierwszych. Jednak nawet pośród tych kilku rezulatatów żaden nie dotyczy złożoności obwodów boolowskich. W niniejszej pracy zajmujemy się pierwszym twierdzeniem tego rodzaju udowodnionym w pracy [1] oraz komentarzu [2] przez Szparlińskiego, Allendera i Saksa. Wynik ten zrodził kilka nowych problemów, które są również przedmiotem poniższych rozważań.

Rozdział pierwszy poświęcony jest zagadnieniu rozmieszczenia liczb pierwszych, które jest nierozerwalnie związane z problemem obliczeniowym rozpoznawania pierwszości. Prezentujemy w nim jedno z najważniejszych twierdzeń tego rodzaju - twierdzenie Dirichleta dotyczące gęstości rozmieszczenia liczb pierwszych w pewnych ciągach arytmetycznych. Rozdział drugi zawiera podstawowe definicje i twierdzenia teorii obwodów boolowskich, które wykorzystujemy w dalszej cześci pracy. Przedmiotem dyskusji w rozdziale trzecim jest dowód wspomnianego twierdzenia autorstwa Szparlińskiego oraz wnioski zeń wynikające. Należy zauważyć, iż autorzy w swojej pierwszej pracy [1] podali dowód zawierający pewne luki. Zostały one następnie usunięte w komentarzu do pracy [2] poprzez wprowadzenie istotnych zmiany do konstrukcji będacej przedmiotem dowodu. W poniższej pracy powracamy do oryginalnej metody dowodowej uzupełniając stosowne luki. Rozdział czwarty rozwiązuje problem postawiony w pracy [1] dotyczący istnienia redukcji języka MOD_2 do $PRIMES$. Dowód prezentowany w tym rozdziale wykazuje istnienie takiej redukcji. Został on uzyskany przez autora pracy niezależnie od rozwiązania podanego w [2]. Ponadto w tej częsci pracy dowodzimy istnienia P-jednorodnej redukcji MOD_2 do $PRIMES^q$ dla liczb pierwszych nieparzystych q przy założeniu pewnej otwartej hipotezy w teorii liczb.

2 Oznaczenia wstępne

2.1 Twierdzenie Dirichleta

W dalszym ciągu potrzebna nam będzie pewna wiedza dotycząca rozmieszczenia liczb pierwszych. Prezentujemy zatem poniżej sformułowanie jednego z najważniejszych twierdzeń tego rodzaju udowodnionego przez P.Dirichleta.

Definicja 2.1. Symbolem $o(G(x))$ określali będziemy klasę funkcji $F(x)$ spełniających zależność:

$$\lim_{x\to\infty} \frac{F(x)}{G(x)} = 0 \tag{1}$$

Jeżeli $F(x)$, $G(x)$ będą dowolnymi funkcjami określonymi w pewnym zbiorze X, a przy tym istnieje taka stała B, że dla wszystkich $x \in X$ zachodzi nierówność

$$|F(x)| \leq BG(x) \tag{2}$$

to będziemy pisali:

$$F(x) = O(G(x)). \tag{3}$$

Liczbę liczb pierwszych mniejszych niż x oznaczymy przez $\pi(x)$. Ponadto wprowadzimy oznaczenie

$$\pi(x;k,l) = \sum_{p\leq x, p=l(\mathrm{mod}\ k)} 1 \tag{4}$$

Bez dowodu podamy poniższe twierdzenie odsyłając czytelnika do [5].

Twierdzenie 2.1. *(Dirchlet) Jeżeli dla pewnych k,l naturalnych $(k,l) = 1$, to wówczas przy x dążącym do nieskończoności mamy:*

$$\pi(x;k,l) = (\frac{1}{\varphi(k)} + f_p(x))\frac{x}{ln(x)}, \tag{5}$$

gdzie funkcja $f_p(x) = o(1)$.

Wniosek 2.2. *Niech p oznacza liczbę pierwszą, zaś α dowolną liczbę z przedziału (0,1). Wówczas istnieje liczba naturalna M taka, że dla wszystkich liczb od niej większych x oraz dowolnego $l \in \{1,\dots,p-1\}$ zachodzi:*

$$\frac{1-\alpha}{p-1}\frac{x}{ln(x)} \leq \pi(x;p,l) \leq \frac{1+\alpha}{p-1}\frac{x}{ln(x)} \tag{6}$$

Co więcej, możemy przyjąć $M = e^{\frac{D}{\alpha}p}$ dla pewnej stałej $D > 0$.

Dowód. Z twierdzenia 2.1 wynika istnienie M takiego, że dla dowolnych $x > M$ oraz $l \in \{1, \dots, p-1\}$ zachodzi:

$$(\frac{1}{p-1} - \frac{\alpha}{p-1})\frac{x}{ln(x)} \leq \pi(x; p, l) \leq (\frac{1}{p-1} + \frac{\alpha}{p-1})\frac{x}{ln(x)} \tag{7}$$

Wystarczy jedynie dobrać takie M, że dla dowolnych $x > M$ $|f_p(x)| < \frac{\alpha}{p-1}$ (zauważmy, że funkcja f_p nie jest zależna od wyboru l). Zatem, aby znaleźć górne oszacowanie dla M spełniających powyższe nierówności, musimy znać szybkość, z jaką funkcja $f_p(x)$ zbiega do zera. Odpowiedniej zależności dostarcza np. praca R.H.Hudsona [8]:

$$f_p(x) = O(ln(x)e^{-C\sqrt{ln(x)}}), \tag{8}$$

gdzie $C > 0$ pewna stała oraz $p \leq ln(x)$. Ponieważ $|f_p(x)| \leq Bln(x)e^{-C\sqrt{ln(x)}}$ (dla pewnej stałej B), musimy oszacować, dla jakich x zachodzi:

$$ln(x)e^{-C\sqrt{ln(x)}} \leq \frac{\alpha}{B(p-1)}. \tag{9}$$

Korzystając z nierówności $e^x \geq x^4$ (dla $x \geq 10$) łatwo udowodnić, że powyższa zależność jest spełniona dla $x \geq e^{\frac{D}{\alpha}p}$ dla pewnej stałej $D > 0$. A zatem okazaliśmy, że $M = e^{\frac{D}{\alpha}p}$ spełnia warunki twierdzenia, co kończy dowód. □

Niech $d_n = p_{n+1} - p_n$ stanowi odstęp między kolejnymi liczbami pierwszymi. Pojawia się naturalne pytanie o oszacowanie wzrostu d_n w stosunku do funkcji zależnych od p_n. Jednym z najbardziej znanych rezultatów tego rodzaju jest twierdzenie Czebyszewa. Będzie nam ono potrzebne w rozdziale 4.

Twierdzenie 2.3. *(Czebyszew) Dla każdego $n \leq 1$ $d_n < p_n$. A zatem $p_{n+1} < 2p_n$.*

Jedną z najdalej idących hipotez dotyczących tego problemu jest hipoteza Cramera sformułowana w 1937, która głosi, że:

$$d_n = O(ln(p_n)^2) \tag{10}$$

Wiadomo zaś, że $d_n \neq O(ln(p_n))$.

2.2 O pojęciu obwodu logicznego

Niech Σ oznacza pewien skończony alfabet. Wówczas napis $\Sigma^\star$ określał będzie zbiór wszystkich skończonych słów nad alfabetem Σ włączywszy słowo puste ϵ. Dowolny podzbiór $L \subset \Sigma^\star$ będziemy nazywali językiem nad alfabetem Σ. Często będziemy pisali $\wedge$, $\vee$, $\neg$ na oznaczenie znanych z logiki spójników logicznych 'i', 'lub' oraz 'nieprawda'.
Zbiór wszystkich funkcji postaci: $f : \{0,1\}^n \to \{0,1\}^m$ dla dowolnych $n, m > 0$ tworzy zbiór funkcji boolowskich. Wiele funkcji rozpatrywanych w tej pracy

będzie należało do tego zbioru np. pochodzące z logiki funkcje AND_n, OR_n oraz NEG zdefiniowane poniżej:

$$AND_n(x_0, \ldots, x_{n-1}) = \begin{cases} 0, & \text{jeżeli istnieje i t.że. } x_i = 0 \\ 1, & \text{w przeciwnym przypadku .} \end{cases} \tag{11}$$

$$OR_n(x_0, \ldots, x_{n-1}) = \begin{cases} 1, & \text{jeżeli istnieje i t.że. } x_i = 1 \\ 0, & \text{w przeciwnym przypadku .} \end{cases} \tag{12}$$

$$NEG(x) = \begin{cases} 1, & \text{jeżeli } x = 0 \\ 0, & \text{w przeciwnym przypadku .} \end{cases} \tag{13}$$

Notację $f \mid A$ zarezerwujemy dla oznaczenia obcięcia funkcji f do dziedziny A. Wprowadzimy też pewne oznaczenia dotyczące grafów skierowanych. Graf skierowany to z punktu widzenia formalnego para zbiorów (E, V), gdzie zbiór E jest nazywany zbiorem wierzchołków, zaś zbiór $V \subset E \times E$ określamy mianem zbioru krawędzi. Dla dowolnego wierzchołka w wyrażenie $In(w)$ określało będzie liczbę krawędzi wchodzących do wierzchołka, zaś $Out(w)$ liczbę krawędzi wychodzących. Wierzchołki w, dla których $In(w) = 0$ nazwiemy wierzchołkami początkowymi, zaś te spełniające warunek $Out(w) = 0$ końcowymi. Wszystkie wierzchołki, które nie są końcowe ani początkowe określali będziemy mianem wewnętrznych. Zdefiniujemy teraz pojęcie obwodu logicznego.

Definicja 2.2. Obwód logiczny $C = (E, V)$ to skończony, acykliczny graf skierowany, którego wierzchołki wewnętrzne etykietowane są funkcjami spełniającymi warunek: dla dowolnego wierzchołka wewnętrznego w funkcja f_w stanowiąca jego etykietę przyjmuje postać $f_w : \{0,1\}^{In(w)} \to \{0,1\}^{Out(w)}$.

Zauważmy, że obwód logiczny o n wierzchołkach początkowych i m końcowych oblicza w naturalny sposób pewną funkcję $f : \{0,1\}^n \to \{0,1\}^m$. Jeżeli bowiem każdemu wierzchołkowi początkowemu przypiszemy stałą 0 lub 1, to, korzystając z funkcji stanowiących etykiety wierzchołków wewnętrznych oraz warunku acykliczności grafu, możemy obliczyć wartości dla wierzchołków końcowych. Założywszy, że na zbiorze wierzchołków początkowych ustalony jest pewien porządek liniowy możemy rozpatrując wszystkie elementy zbioru $\{0,1\}^n$ obliczyć poszukiwaną funkcję.
Zdefinujemy teraz rozmiar i głębokość obwodu logicznego.

Definicja 2.3. Rozmiar obwodu logicznego C oznaczamy symbolem $size(C)$, jest on równy liczbie wierzchołków wewnętrznych grafu reprezentującego obwód C.

Definicja 2.4. Głębokość obwodu logicznego C oznaczamy symbolem $depth(C)$, jest ona równa maksymalnej liczbie wierzchołków wewnętrznych leżacych na pewnej ścieżce prowadzącej z wierzchołka początkowego do końcowego.

Pojęcie rozpoznawalności funkcji przez ciąg obwodów będzie kluczowe dla dalszych rozważań:

Definicja 2.5. Niech $f : \{0,1\}^\star \to \{0,1\}^m$ i niech $(C_n)_{n\geq 1}$ ciąg obwodów logicznych takich, że dla dowolnej liczby naturalnej n C_n posiada n wierzchołków początkowych oraz m wierzchołków końcowych. Powiemy, że ciąg obwodów $(C_n)_{n\geq 1}$ rozpoznaje funkcję f jeżeli dla każdego n obwód C_n rozpoznaje funkcję $f \mid \{0,1\}^n$.

Poniższa obserwacja pokazuje, że dla dowolnej funkcji $f : \{0,1\}^\star \to \{0,1\}^m$ istnieje pewien ciąg obwodów, który ją rozpoznaje.

Twierdzenie 2.4. *Niech $f : \{0,1\}^\star \to \{0,1\}^m$ oznacza dowolną funkcję. Wówczas istnieje ciąg obwodów $(C_n)_{n\geq 1}$ taki,że:*

1. *$(C_n)_{n\geq 1}$ rozpoznaje f,*
2. *$depth(C_n) \leq 3$ dla dowolnego n naturalnego,*
3. *$size(C_n) = O(n2^n)$,*
4. *wierzchołki wewnętrzne ciągu $(C_n)_{n\geq 1}$ są etykietowane funkcjami AND,OR oraz NEG.*

Dowód. Niech n dowolna liczba naturalna. Skonstruujemy obwód C_n rozpoznający funkcję $g = f|\{0,1\}^n$ i spełniający powyższe warunki. W tym celu rozpatrzymy dwa przypadki:

1. $g \equiv 0$. Wtedy oczywiście $g(x_0,\ldots,x_{n-1}) = AND(x_0, NEG(x_0))$ i konstrukcja obwodu C_n wynika bezpośrednio z tego przedstawienia.
2. Niech $g \not\equiv 0$. Wówczas możemy założyć, że istnieje $x = (x_0,\ldots,x_{n-1})$ takie, iż $g(x) = 1$. Funkcję g można zatem przedstawić w postaci:

$$g(x_0,\ldots,x_{n-1}) = \bigvee_{\substack{(\sigma_0,\ldots,\sigma_{n-1})\\ g(\sigma_0,\ldots,\sigma_{n-1})=1}} x_0^{\sigma_0} \wedge \cdots \wedge x_{n-1}^{\sigma_{n-1}}, \qquad (14)$$

   gdzie $x_i^0 = NEG(x_i)$, zaś $x_i^1 = x_i$. Nietrudno zauważyć, że powyższa równość pozwala skonstruować obwód C_n o głębokości 3 i rozmiarze rzędu $O(n2^n)$ rozpoznający funkcję g.

$\square$

Warto w tym miejscu zauważyć, że istnieją znacznie silniejsze oszacowania rozmiaru minimalnego obwodu rozpoznającego funkcję f. Zagadnienie to wiąże się z tzw. indeksami prostoty alternatywnej postaci kanonicznej i jest rozpatrywane np. w [9]. Zdefiniujemy teraz klasę funkcji rozpoznawalnych przez ciągi obwodów o możliwie najmniejszej - stałej głębokości.

Definicja 2.6. Powiemy, że funkcja $f : \{0,1\}^\star \to \{0,1\}^m$ należy do klasy FAC^0, jeżeli istnieje ciąg obwodów $(C_n)_{n\geq 1}$ rozpoznający tę funkcję, który dodatkowo spełnia poniższe warunki:

1. wierzchołki wewnętrzne obwodów C_n są etykietowane jedynie funkcjami AND,OR oraz NEG
2. istnieje stała $D > 0$, taka że dla każdego n $depth(C_n) < D$
3. istnieje wielomian P taki, że dla każdego n $size(C_n) < P(n)$

Czasami, aby zaznaczyć jaką stała oraz wielomian mamy na myśli będziemy pisali $f \in FAC^0\ (D, P(n))$

Niech L dowolny język nad alfabetem $\{0,1\}$.

Definicja 2.7. Funkcją charakterystyczną jezyka L nazywać będziemy funkcję $\chi_L : \{0,1\}^\star \to \{0,1\}$, która jest zdefiniowana następująco:

$$\chi_L(x) = \begin{cases} 1, & \text{jeżeli } x \in L \\ 0, & \text{w przeciwnym przypadku .} \end{cases} \tag{15}$$

Definicję klasy funkcji FAC^0 można w łatwy sposób rozszerzyć na klasę języków korzystając z pojęcia funkcji charakterystycznej języka.

Definicja 2.8. Powiemy, że język L należy do klasy AC^0, jeżeli $\chi_L \in FAC^0$.

Będziemy pisali $L \in AC^0(C, P(n))$, jeżeli $\chi_L \in FAC^0(C, P(n))$. Podamy teraz kilka definicji funkcji boolowskich wykorzystywanych w dalszym ciągu pracy.

Definicja 2.9. Niech $m \in \mathbf{N}$, $m \geq 2$ oraz x $= (x_n x_{n-1} ... x_1 x_0)_2$. Wówczas funkcje boolowskie MOD$_m$, PRIMES, MAJORITY, SUM, REM$_m$ zdefinujemy następująco:

$$MOD_m(x) = \begin{cases} 1, & \text{jeżeli } \sum_i x_i \equiv 0 \pmod{\mathrm{m}} \\ 0, & \text{w przeciwnym przypadku .} \end{cases} \tag{16}$$

$$PRIMES(x) = \begin{cases} 1, & \text{jeżeli } \sum_i x_i 2^i \in \mathbf{P} \\ 0, & \text{w przeciwnym przypadku .} \end{cases} \tag{17}$$

$$MAJORITY(x) = \begin{cases} 1, & \text{jeżeli } \sum_i x_i \geq n/2 \\ 0, & \text{w przeciwnym przypadku .} \end{cases} \tag{18}$$

$$SUM(x) = (y_l y_{l-1} ... y_1 y_0)_2, \ \text{,gdzie } \sum_i y_i 2^i = \sum_i x_i \tag{19}$$

$$REM_m(x) = (y_k y_{k-1} ... y_1 y_0)_2, \text{ gdzie } \sum_i y_i 2^i = \sum_i x_i \pmod{m}$$
$$\text{oraz } \sum_i y_i 2^i < m \quad (20)$$

Spośród powyższych funkcji żadna nie należy do klasy FAC^0. Wynika to pośrednio z poniższego twierdzenia, którego dowód można znaleźć na przykład w pracy [7].

Twierdzenie 2.5. *Języki MOD_n dla $n \geq 2$ nie należą do klasy AC^0.*

Zdefiniujemy klasę języków $ACC(q)$ analogicznie jak klasę AC^0 (patrz def. 2.6) z tą różnicą, że w warunku 1 dozwolimy dodatkowo, aby funkcja MOD_q mogła być etykietą wierzchołka wewnętrznego. Przyjmiemy ponadto, iż klasa języków ACC oznaczała będzie klasę $\bigcup_{q>1} ACC(q)$. Twierdzenie 2.5 można teraz wzmocnić następująco:

Twierdzenie 2.6. *(Smoleński) Jeżeli p jest liczbą pierwszą i $q > 1$ zawiera czynnik pierwszy różny od p, to $MOD_q \notin ACC(p^k)$ dla dowolnego $k > 1$.*

Dowód powyższego twierdzenia jest trudny i można go znaleźć np. w [3].
Przykładem nietrywialnej funkcji, która należy do klasy FAC^0 może być funkcja ADD(x,y): $\{0,1\}^{2n} \to \{0,1\}^{n+1}$, która dla dwóch liczb n-bitowych x,y oblicza ich $n+1$-bitową sumę. Warto w tym miejscu zauważyć, że funkcja mnożąca dwie liczby n-bitowe nie należy do klasy FAC^0, podobnie jak funkcja wykonująca dzielenie całkowite dwóch takich liczb (patrz np. [4]).

Twierdzenie 2.7. *Funkcja ADD należy do klasy $FAC^0(6, 20n^3)$.*

Dowód. Niech x,y będą dwiema liczbami n-bitowymi, których reprezentacje binarne są postaci: $(x_{n-1}, \ldots, x_0)$ oraz $(y_{n-1}, \ldots, y_0)$. Sumę tych dwóch liczb można zapisać przy użyciu $n+1$-bitów - $(z_n, \ldots, z_0)$. Zauważmy, że najprostsza, klasyczna metoda dodawania prowadziłaby do obwodu, którego głębokość jest rzędu $O(n)$. Problemem staje się w tym wypadku obliczanie przeniesienia, bowiem przy standardowej technice dodawania, aby znaleźć przeniesienie występujące na i-tej pozycji musimy znać przeniesienie z pozycji $i-1$-szej. Musimy zatem w pewien sposób zrównoleglić wykonywane obliczenia. Stosując to rozumowanie możemy zapisać następujący wzór pozwalający obliczać przeniesienie pojawiające się na i-tej pozycji:

$$CARRY_i = \bigvee_{j \in \{0,\ldots,i-1\}} ((x_j \wedge y_j) \wedge \bigwedge_{k \in \{j+1,\ldots,i-1\}} (x_k \vee y_k)) \quad (21)$$

Wzór powyższy należy tłumaczyć następująco: przeniesienie na i-tej pozycji mogło pojawić się po raz pierwszy na dowolnej pozycji j należacej do zbioru $\{0 \ldots i-1\}$. Aby jednak zachowało się ono aż do i-tej pozycji wśród par bitów x_k, y_k dla k pomiędzy $j+1$ a $i-1$ przynajmniej jeden bit z każdej pary musi być ustawiony. Zauważmy, że tę metodę obliczania przeniesienia można zrealizować przy pomocy obwodu o głębokości 3. Mając obliczony bit przeniesienia możemy podać wzory pozwalające obliczyć kolejne bity sumy:

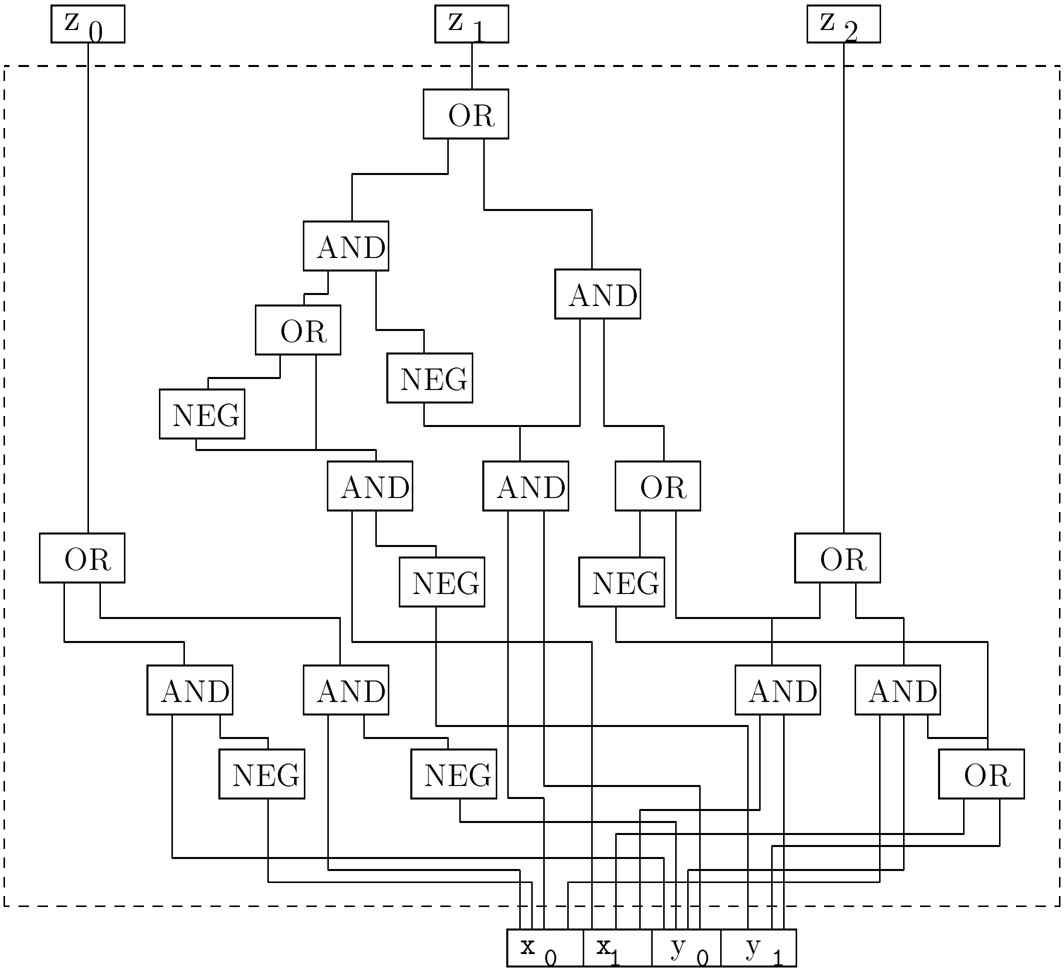
$$z_0 = (x_0 \wedge \neg y_0) \vee (\neg x_0 \wedge y_0)$$

$$z_i = (((x_i \wedge \neg y_i) \vee (\neg x_i \wedge y_i)) \wedge \neg CARRY_i) \vee$$

$$(((x_i \wedge y_i) \vee (\neg x_i \wedge \neg y_i)) \wedge CARRY_i) \text{ dla } i \in \{1, \dots, n-1\}$$

$$z_n = CARRY_n$$

Dzięki powyższym zależnościom możemy zbudować ciąg obwodów $(C_n)_{n \geq 1}$ rozpoznający funkcję ADD. Nietrudno zauważyć, że głębokość takiego obwodu wyniesie 6 (jest niezależna od rozmiaru składników sumowania), zaś rozmiar jest ograniczony np. przez wielomian $20n^3$. A zatem na mocy definicji 2.6 funkcja ADD należy do klasy FAC^0. W charakterze przykładu pokażemy schemat obwodu logicznego C_4 obliczającego sumę dwóch liczb 2-bitowych.



□

Następujące twierdzenie będziemy często wykorzystywali, aby dowieść, że pewna funkcja należy do klasy FAC^0.

Twierdzenie 2.8. *Niech f dowolna funkcja. Jeżeli istnieją funkcje g, h takie, że:*

1. $\forall x \, |g(x)| \leq Cln(|x|))$
2. $g \in FAC^0$
3. $f = h \circ g$

to wówczas $f \in FAC^0$.

Dowód. Skoro g należy do klasy FAC^0 to istnieje ciąg obwodów $(C_n)_{n\geq 1}$ rozpoznający funkcję g i spełniający warunki definicji 2.6. Zbudujemy teraz ciąg obwodów $(D_n)_{n\geq 1}$ rozpoznający funkcję h. W tym celu skorzystamy z twierdzenia 2.4. W myśl tego twierdzenia istnieje ciąg obwodów $(D_n)_{n\geq 1}$ ropoznający h oraz taki, że $\text{depth}(D_n) \leq 3$ i $\text{size}(D_n) \leq Bn2^n$ dla każdego $n \geq 1$ oraz pewnej stałej B. Skonstruujemy teraz ciąg obwodów $(E_n)_{n\geq 1}$, w którym dowolny obwód E_n powstaje poprzez zastosowanie obwodu $D_{\lceil Cln(n) \rceil}$ do wyjścia generowanego przez obwód C_n. Dzięki warunkowi 1 oraz 3 ciąg $(E_n)_{n\geq 1}$ rozpoznaje funkcję f. Łatwo zauważyć, że $\text{depth}(E_n) \leq 3 + \text{depth}(C_n)$ oraz $\text{size}(E_n) \leq \text{size}(C_n) + BCln(n)n^C$. A zatem f należy do klasy FAC^0, co kończy dowód. □

3 O pojęciu redukowalności

Powiemy, że język L_1 jest AC^0-redukowalny do języka L_2, jeżeli L_1 jest rozpoznawalny przez wielomianową rodzinę obwodów o stałej głębokości wykorzystującą bramki NOT oraz bramki OR, AND i L_2 o nieograniczonym wejściu (bramka L_2 przyjmująca na wejściu ciąg r bitów $b_r, b_{r-1}, ..., b_1, b_0$ daje w wyniku 1 o ile $b_r b_{r-1} ... b_1 b_0 \in L_2$). Na oznaczenie powyższej redukcji będziemy stosowali zapis $L_1 \leq_T^{AC^0} L_2$. Czasami pojawiała się też będzie mniej zwięzła notacja $L_1 \leq_T^{AC^0} L_2\ (D, P(n))$, w której explicite podajemy głębokość D obwodu obliczającego redukcję L_1 do L_2 oraz wielomian $P(n)$ ograniczający liczbę bramek w stosunku do rozmiaru wejścia jego n-tego sieci składowej. Oczywiście możemy bez utraty ogólności zakładać, iż powyższy wielomian ograniczony do liczb naturalnych jest funkcją rosnącą:

$$\forall\ i, j \geq 1\ j > i\ \Rightarrow\ P(j) > P(i) \tag{22}$$

Czasami będziemy też mieli do czynienia z wielomianami dwóch zmiennych $P(m, n)$, gdzie zmienna m parametryzuje pewną rodzinę obwodów. Wówczas milcząco zakładamy, że wielomian P jest rosnący ze względu na każda ze swoich zmiennych (ograniczamy się oczywiście do liczb naturalnych).
W analogiczny sposób możemy zdefiniować pojęcie AC^0-redukowalności funkcji $f : \{0,1\}^\star \to \{0,1\}^n$ do funkcji $g : \{0,1\}^\star \to \{0,1\}^m$ - $f \leq_T^{AC^0} g$ oraz języka $L \in \{0,1\}^\star$ do funkcji g - $L \leq_T^{AC^0} g$.
Podamy teraz prostą do udowodnienia obserwację, która w pewnym sensie stanowi motywację do wprowadzenia pojęcia redukowalności.

Obserwacja 3.1. *Niech L,K dowolne języki nad alfabetem Σ. Wówczas jeżeli $L \leq_T^{AC^0} K$ oraz $K \in AC^0$ to $L \in AC^0$.*

Powyższa obserwacja wykorzystywana jest często przy dowodzeniu, że pewien język L nie należy do AC^0, o ile wiemy, że inny język K nie należy do AC^0 i K jest AC^0-redukowalny do L.
Łatwo poczynić następującą obserwację:

Obserwacja 3.2. *Relacja $\leq_T^{AC^0}$ jest przechodnia tzn. dla dowolnych języków L_1,L_2,$L_3 \subseteq \{0,1\}^\star$ jeżeli $L_1 \leq_T^{AC^0} L_2$ oraz $L_2 \leq_T^{AC^0} L_3$, to zachodzi również $L_1 \leq_T^{AC^0} L_3$. Co więcej, jeżeli $L_1 \leq_T^{AC^0} L_2(C, P(n))$ oraz $L_2 \leq_T^{AC^0} L_3(D, R(n))$ dla pewnych stałych $C, D \in \mathbf{N}$ oraz wielomianów $P(n), R(n)$, to wówczas:*

$$L_1 \leq_T^{AC^0} L_3(C \cdot D, P(n)R(P(n))) \tag{23}$$

Dowód. Niech $(C_n)_{n \geq 1}$, $(D_n)_{n \geq 1}$ ciągi obwodów logicznych ustalające redukcję odpowiednio pomiędzy L_1 i L_2 oraz L_2 i L_3. Skonstruujemy ciąg obwodów $(E_n)_{n \geq 1}$ dowodzący istnienia redukcji L_1 do L_3. Niech n dowolna liczba naturalna. Obwód E_n powstaje z obwodu C_n przez wstawienie w miejsce wystąpień bramek L_2 obwodów D_n. Zauważmy, że tak skonstruowany ciąg $(E_n)_{n \geq 1}$ rozpoznaje język L_1. Ponadto size$(E_n) \leq CD$, bowiem w najgorszym przypadku na ścieżce prowadzącej z pewnego wierzchołka początkowego do końcowego w obwodzie C_n mogło pojawić się C bramek L_2 zastąpionych później obwodami D_n o głębokości D. Co więcej, rozmiar obwodu E_n jest ograniczony przez wielomian $P(n)R(P(n))$, gdyż obwód C_n może maksymalnie zawierać $P(n)$ bramek L_2, których rozmiar wejścia jest ograniczony również przez $P(n)$. Skoro jednak rozmiar obwodu $D_{P(n)}$ jest mniejszy niż $R(P(n))$, to uzyskujemy tezę twierdzenia. $\square$

Nietrudno dowieść prawdziwości następującego wniosku z twierdzenia 2.8:

Wniosek 3.3. *Niech L dowolny język nad alfabetem $\{0,1\}$, zaś g pewna funkcja postaci $g : \{0,1\}^\star \to \{0,1\}^m$ taka, że $\forall x\, |g(x)| \leq Cln(|x|))$ dla pewnego $C > 0$. Jeżeli istnieje funkcja h taka, że $\chi_L = h \circ g$, to wówczas $L \leq_T^{AC^0} g$.*

Zastosujemy powyższy wniosek do dowodu dwóch redukcji.

Twierdzenie 3.4. *Dla dowolnego $n \in \mathbf{N}$, $n \geq 2$ zachodzi:*

$$MOD_n \leq_T^{AC^0} SUM \tag{24}$$

Dowód. Zauważmy, że funkcja $SUM(x)$ spełnia warunek $\forall x |SUM(x)| \leq 2\ln(|x|))$. Ponadto istnieje funkcja h taka, że $MOD_n = h(SUM(x))$. Jest to oczywiste bowiem na podstawie sumy bitów liczby wejściowej x jesteśmy w stanie ocenić czy dzieli się ona przez liczbę n. A zatem z wniosku 3.3 natychmiast dostajemy tezę twierdzenia. $\square$

Twierdzenie 3.5.

$$MAJORITY \leq_T^{AC^0} SUM \tag{25}$$

Dowód. Podobnie jak w poprzednim dowodzie funkcja SUM spełnia odpowiedni warunek. Ponadto na podstawie sumy bitów liczby wejściowej x jesteśmy w stanie ocenić, czy jest ona większa od połowy długości zapisu liczby x. A zatem teza wynika natychmiast z wniosku 3.3. $\square$

Wyróżnimy teraz pewną podklasę wszystkich AC^0-redukcji.

Definicja 3.1. Niech L_1, L_2 dowolne języki nad alfabetem Σ. Powiemy, że redukcja $L_1 \leq_T^{AC^0} L_2$ wyznaczona przez ciąg obwodów logicznych $(C_n)_{n\geq 1}$ jest P-jednorodna jeżeli funkcja $n \mapsto C_n$ jest obliczalna przez deterministyczną maszynę Turinga w czasie wielomianowym.

4 Dolna granica dla problemu pierwszości

Jak wspomnieliśmy we wstępie E.Allender, M.Saks oraz I.Szparliński dowiedli w swojej pracy [1] prawdziwości następującego twierdzenia:

Twierdzenie 4.1. *Dla dowolnego $p \in \mathbf{P}$, $p \neq 2$ zachodzi:*

$$MOD_p \leq_T^{AC^0} PRIMES(36, P(p,n)), \tag{26}$$

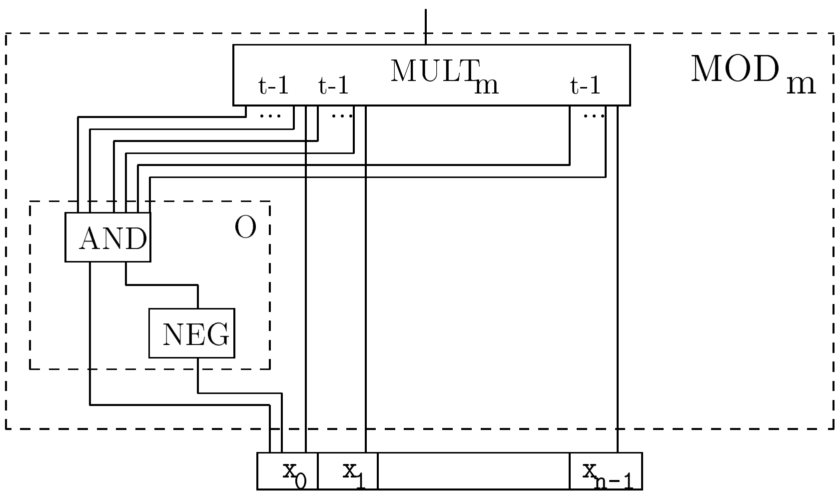
gdzie $P(p,n)$ pewien wielomian dwóch zmiennych (w istocie możemy przyjąć $P(p,n) = D(n^8 + n^2p^3)$ dla pewnej stałej $D > 0$).

Należy zwrócić uwagę, że w powyższym twierdzeniu głębokość redukcji nie zależy od p. Ponadto ze względu na charakter konstrukcji uzyskana redukcja nie będzie P-jednorodna. W celu udowodnienia twierdzenia 4.1 dowiedziemy kolejno prawdziwości dwóch twierdzeń pomocniczych.

Twierdzenie 4.2. *Niech $m \in \mathbf{N}$ będzie liczbą nieparzystą. Wówczas:*

$$MOD_m \leq_T^{AC^0} MULT_m(3, 3n^2) \tag{27}$$

Dowód. Zauważmy, że założenie o nieparzystości m jest istotne. Jeżeli bowiem $m = 2$, to język $MULT_2$ należy do klasy AC^0, zaś jak wynika z twierdzenia 2.5 język MOD_2 do tej klasy nie należy.
Ponieważ z założenia $(2, m) = 1$, to z twierdzenia Eulera otrzymujemy, iż $2^t = 1(\text{mod}\, m)$ dla $t = \varphi(m)$. Dokonując redukcji będziemy w dalszym ciągu korzystali ze stałej t. Wygląd obwodu obliczającego funkcję MOD_m prezentujemy na poniższym rysunku:



Wejściowy ciąg bitów dla n-tego obwodu $x = x_n x_{n-1}..x_1$ jest zamieniany na ciąg $x_n 0^{t-1} x_{n-1} 0^{t-1} ... x_1$ (pomiędzy każdymi dwoma bitami wejścia wstawiamy ciąg zer długości $t-1$). Następnie wynik tej operacji staje się argumentem dla pojedyńczej bramki $MULT_m$. Poprawność konstrukcji wynika łatwo z następujacej równości:

$$\sum_i x_i = \sum_i x_i 2^{ti} (\mathrm{mod}\, m) \tag{28}$$

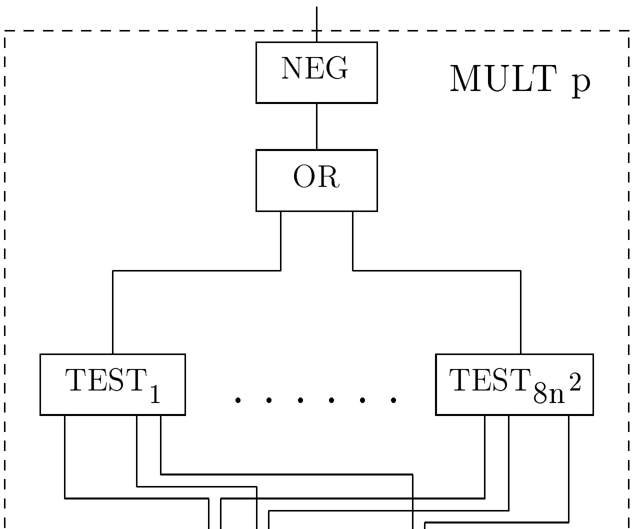
Liczba użytych bramek w n-tym obwodzie wynosi dokładnie: $1+2(n-1)(\varphi(n)-1)$, a zatem jest mniejsza niż $3n^2$. Tym samym dowód twierdzenia został zakończony. □

Twierdzenie 4.3. *Niech p będzie nieparzystą liczbą pierwszą. Wówczas:*

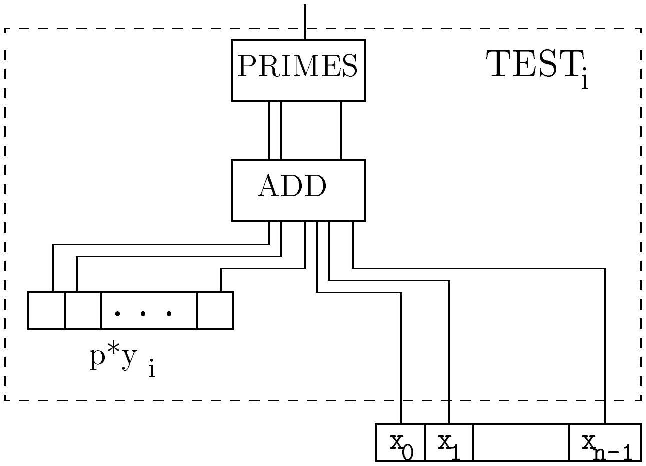
$$MULT_p \leq_T^{AC^0} PRIMES(12, C(n^3+p^3)). \tag{29}$$

dla pewnej stałej $C > 0$.

Dowód. Niech n będzie dowolną liczbą naturalną. Skonstruujemy obwód C_n zbudowany z bramek PRIMES, OR, AND obliczający dla liczb n-bitowych x wartość $MULT_p(x)$. Na początku ograniczymy się do przypadku, gdy $n > M$ dla pewnej stałej $M > 0$, której wartość wyliczymy w dalszym ciągu. Poniżej prezentujemy schemat obwodu C_n.



Niech $\{y_i\}_{i \leq 8n^2}$ będzie ciągiem liczb z przedziału $[1, 2^n]$. Obwód C_n zbudowany jest z bramek $TEST_i(y_i)$, których konstrukcja jest uzależniona od wyboru liczb y_i. Realizacja bramki $TEST_i$ przy użyciu bramek AND, OR i $PRIMES$ przedstawia się następująco (na mocy 2.7 funkcja ADD należy do klasy FAC^0):



Pozostaje dowieść, że istnieje ciąg liczb $\{y_i\}_{i \leq 8n^2}$, dla których obwód C_n oblicza funkcję $MULT_p$. Aby to okazać, weźmy dowolną liczbę n-bitową x. Jeżeli $p|x$ to oczywiście niezależnie od wyboru elementów ciągu $\{y_i\}$ $p \,|\, x+py_i$, $x+py_i <> p$ (możemy założyć, że $x > 0$) i każda z bramek $TEST_i$ daje na wyjściu 0. Stąd, zgodnie z oczekiwaniem, obwód C_n akceptuje x. Załóżmy, że $p \not| \, x$. Wówczas $x = l (\bmod p)$ dla pewnego $l \in \{1, \ldots, p-1\}$. Zakładając, że liczby y_i przyjmują wartości z przedziału $[1, 2^n]$ z jednakowym prawdopodobieństwem dostajemy:

$$P(TEST_i = 1) = \frac{\pi(x + p * 2^n; p, l) - \pi(x; p, l)}{2^n} \tag{30}$$

Chcemy dowieść, że istnieje stała $D > 0$ taka, iż dla wszystkich $n > M$ mamy:

$$P(TEST_i = 1) \geq \frac{D}{n} \tag{31}$$

Skorzystajmy teraz z wniosku 2.2 dla stałej $\alpha = \frac{1}{4}$. Wówczas dla $p2^n > e^{4Ep}$ (stała E pochodzi z zastosowanego wniosku) zachodzi:

$$\pi(x + p2^n; p, l) \geq \pi(p2^n; p, l) \geq \frac{3p2^n}{4(p-1)ln(p2^n)} \tag{32}$$

A zatem powyższa nierówność zachodzi m.in. dla $n > 8Ep$. Nietrudno zauważyć na podstawie tego samego wniosku, że dla $n > 8Ep$ zachodzi również:

$$\pi(x; p, l) \leq \pi(2^n; p, l) \leq \frac{5 * 2^n}{4(p-1)nln(2)} \leq \frac{8 * 2^n}{4(p-1)n} \tag{33}$$

Łącząc powyższe obserwacje z 30 dostaniemy dla $n > 8Ep$:

$$P(TEST_i = 1) \geq \frac{1}{4(p-1)} \left(\frac{3p}{\frac{ln(p)}{n} + 0.7} - 8 \right) \frac{1}{n} \tag{34}$$

Jeżeli przyjmiemy, że $M = 4(1+2E)p > max(4ln(p), 8Ep)$, to wówczas z powyższych rozważań wynika, iż stała D równa $\frac{3p-8}{4(p-1)}$ spełnia nierówność 31. Nietrudno dowieść, że dla dowolnej liczby pierwszej nieparzystej p $D \geq \frac{1}{8}$. A zatem udowodniliśmy, iż:

$$P(TEST_i = 1) \geq \frac{1}{8n} \tag{35}$$

dla dowolnej liczby pierwszej nieparzystej p i $n > M$, gdzie $M = 4(1+2E)p$. W dalszym ciągu będziemy chcieli oszacować prawdopodobieństwo z jakim obwód C_n wygeneruje niepoprawną odpowiedź równą 1 w przypadku, gdy $p \not| x$.

$$P(C_n = 1) = P(\bigcap_{i \leq 8n^2} \{TEST_i = 0\}) = \prod_{i \leq 8n^2} (1 - P(TEST_i = 1)) \tag{36}$$

Łacząc powyższy wynik z 35 dostaniemy:

$$P(C_n = 1) \leq (1 - \frac{1}{8n})^{8n^2} \tag{37}$$

Korzystając jednak ze znanej nierówności $1 - x \leq e^{-x}$ uzyskamy:

$$1 - \frac{1}{8n} \leq e^{-\frac{1}{8n}} \tag{38}$$

A stąd po przekształceniach:

$$8n^2 ln(1 - \frac{1}{8n}) \leq -n \tag{39}$$

Ostatecznie stosując funkcję $e(x)$ do obu stron nierówności i wykorzystując 37 mamy:

$$P(C_n = 1) \leq e^{-n} \tag{40}$$

A zatem procent obwodów, które będą działały błędnie w przypadku ustalonej liczby n-bitowej x jest mniejszy niż e^{-n}. Skoro jednak możliwych wyborów x jest co najwyżej 2^n oraz $e^{-n}2^n < 1$ dla dowolnego n naturalnego, to istnieje ciąg liczb $(y_i)_{i \leq 8n^2}$, dla którego obwód C_n realizuje funkcję $MULT_p$.
Skonstruowaliśmy zatem ciąg obwodów $(C_n)_{n \geq M}$, gdzie $M = 4(1+2E)p$ dla pewnej stałej $E > 0$, rozpoznający funkcję $MULT_p$. Nietrudno zauważyć uwzględniając twierdzenie 2.7 oraz fakt, iż każdorazowe użycie stałej 0 lub 1 w rzeczywistości wymaga użycia obwodu o głębokości 2, że depth$(C_n) < 12$. Warto w tym miejscu zwrócić uwagę na fakt, iż głebokość rozpatrywanych obwodów jest stała i niezależna od p. Spróbujmy teraz oszacować rozmiar obwodu C_n. Rozmiar każdego z obwodów $TEST_i$ jest ograniczony przez $20n^3 + 1 + 2log_2(p)n$ (korzystamy z twierdzenia 2.7 oraz konieczności umieszczenia odpowiednich obwodów reprezentujących stałe). A zatem size(C_n) jest mniejszy niż $8n^2 *$size$(TEST_i) + 2$. Biorąc pod uwagę, iż z $n > M$ wynika, że $n > log_2(p)$ dostajemy size$(C_n) < 200n^5$.

Pozostał jeszcze problem konstrukcji obwodów C_n dla $n < M$. Przyjmiemy, iż $C_n = C_M$ dla $n < M$. Aby konstrukcja była poprawna należy dodatkowo założyć, że na $M - n$ pozycjach wejściowych w obwodzie C_n ustawiamy zera. Musimy teraz poprawić oszacowanie rozmiaru obwodów C_n, które obecnie wynosi:

$$\mathrm{size}(C_n) < \max(200n^3, 200M^3) \tag{41}$$

Ostatecznie możemy przyjąć, że $\mathrm{size}(C_n) < C(n^3 + p^3)$ dla pewnej stałej $C > 0$. Dowód twierdzenia można zatem uznać za zakończony. □

Proof. (Twierdzenie 4.1) Chcąc uzyskać redukcję $MOD_p \leq_T^{AC^0} PRIMES$ skorzystamy z twierdzeń 4.2 i 4.3 oraz własności przechodniości AC^0-redukcji. W tym celu powołamy się na obserwację 3.2, uzyskując redukcję: $MOD_p \leq_T^{AC^0} PRIMES(3 * 12, 3n^2C(27n^6 + p^3))$. Ponieważ $3n^2C(27n^6 + p^3) < 81C(n^8 + n^2p^3)$, to powyższa zależność implikuje tezę twierdzenia. □

Twierdzenie 4.1 pozwala nam udowodnić zapowiadany wcześniej fakt, iż $PRIMES$ nie należy do klasy AC^0.

Twierdzenie 4.4. $PRIMES \notin AC^0$

Dowód. Załóżmy bowiem, że jest przeciwnie: $PRIMES \in AC^0$. Ponieważ jednak na mocy twierdzenia 4.1 $MOD_3 \leq_T^{AC^0} PRIMES$, to korzystając z obserwacji 3.1 dostaniemy $MOD_3 \in AC^0$. Ostatnia zależność stoi jednak w sprzeczności z twierdzeniem 2.5. □

Korzystając z twierdzenia Smoleńskiego można rozszerzyć poprzedni rezultat:

Twierdzenie 4.5. *Niech p będzie dowolną liczbą pierwszą. Wówczas:*

$$PRIMES \notin ACC(p) \tag{42}$$

Dowód. Korzystając z twierdzenia 4.1 dostaniemy, iż $MOD_3 \leq_T^{AC^0} PRIMES$ oraz $MOD_5 \leq_T^{AC^0} PRIMES$. Jeżeli zatem dla pewnej liczby pierwszej $p \neq 3$ mielibyśmy $PRIMES \leq_T^{AC^0} MOD_p$, to otrzymamy $MOD_3 \leq_T^{AC^0} MOD_p$ co przeczy twierdzeniu 2.6. Jeżeli zaś byłoby $PRIMES \leq_T^{AC^0} MOD_3$, to uzyskamy sprzeczność w postaci redukcji $MOD_5 \leq_T^{AC^0} MOD_3$. □

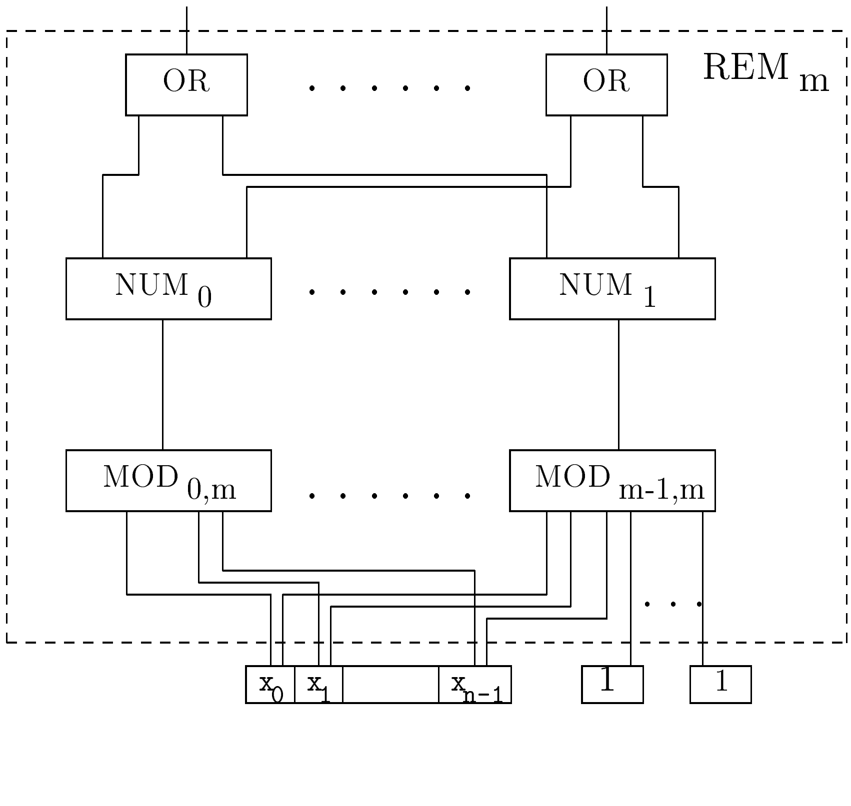
5 O redukcjach do języka PRIMES

W poprzednim rozdziale pokazaliśmy, iż $MOD_p \leq_T^{AC^0} PRIMES$ dla nieparzystych liczb pierwszych p. Pozostaje zatem otwarte pytanie, czy $MOD_2 \leq_T^{AC^0} PRIMES$. Poniżej zajmiemy się wykazaniem, że wspomniana redukcja jest prawdziwa. Twierdzenie, które w tym celu udowodnimy pomoże nam ponadto wykazać istnienie redukcji jeszcze innych języków do języka $PRIMES$.

Lemat 5.1. *Dla dowolnego* $m \in \mathbf{N}$, $m \geq 2$

$$REM_m \leq_T^{AC^0} MOD_m(5, 11m^2) \tag{43}$$

Dowód. Przy konstrukcji ciągu obwodów obliczających funkcję $REM_m(x)$ będziemy wykorzystywali pomocniczo funkcję $NUM_y(x) : \{0,1\} \to \{0,1\}^{\lfloor ln_2(y) \rfloor + 1}$. Funkcja ta dla argumentu 1 zwraca w wyniku zapis binarny liczby y, zaś w przypadku 0 jej wartością jest ciąg zer o odpowiedniej długości. Zauważmy, że NUM_y można rozpoznawać przy pomocy obwodu o głębokości 3 i rozmiarze $3(\lfloor ln_2(y) \rfloor + 1) \leq 3(y+1)$. Jest to fakt łatwy do udowodnienia, należy jedynie pamiętać, że stałe 0 i 1 potrzebne przy konstrukcji są w rzeczywistości obliczane przy użyciu wyrażeń $OR(NEG(x_i), x_i)$ oraz $AND(NEG(x_i), x_i)$ dla pewnego bitu wejściowego x_i. W dalszym ciągu pokażemy jak skonstruować obwód C_n obliczający dla n-bitowych liczb x wartość $REM_m(x)$. Poniżej prezentujemy schematyczny rysunek takiego obwodu:



Najniższa warstwa obwodu złożona jest z bramek $MOD_{i,m}$ dla $i \in \{0, \dots, m-1\}$. Dowolna bramka $MOD_{i,m}$ odpowiada bramce MOD_m, która poza n bitami wejściowymi ma wśród swoich argumentów i bitów równych 1. Bezpośrednio nad każdą z bramek $MOD_{i,m}$ znajduje się bramka NUM_j taka, że $j < m$ i $j + i = 0 \ (\bmod m)$. Ostatnia warstwa zbudowana jest z $\lfloor log_2(m) \rfloor + 1$ bramek OR. Pierwsza z tych bramek dostaje jako argumenty bity wyjściowe o numerze 0 z wszystkich bramek NUM_i, druga - wszystkie bity o numerze 1 itd. Łatwo zauważyć, że tak skonstruowany obwód oblicza funkcję REM_m. W przypadku bowiem, gdy $x = l (\bmod m)$, jedynie bramka $MOD_{m-l(modm),m}$ da na wyjściu wartość 1 i właściwa bramka NUM wygeneruje zapis binarny liczby l. Głębokość przedstawionego obwodu jest równa 5, zaś jego rozmiar jest niezależny od liczby n i niewiększy niż $m + \frac{3m(m-1)}{2} + m(3(m+1)) + log_2(m) + 1 \leq 11m^2$. □

Twierdzenie 5.2. *Niech $L \subseteq \{0,1\}^\star$. Jeżeli istnieje nieskończony ciąg liczb naturalnych $l_1 < l_2 < ...$, rosnące wielomiany $P(n,m)$,$R(n)$ oraz stała $C \in \mathbf{N}$, dla których zachodzą następujące zależności:*

(i) $\forall \ i \ l_{i+1} \leq R(l_i)$

(ii) $\forall \ i \ MOD_{l_i} \leq_T^{AC^0} L \ (C, P(l_i, n))$

to wówczas $SUM \leq_T^{AC^0} L$.

Dowód. Weźmy dowolne $n \in \mathbf{N}$. Zbudujemy obwód C_n obliczający funkcję SUM dla wejścia n-bitowego $x_{n-1}...x_1x_0$. Możemy ograniczyć się do przypadku $n \geq l_1$. Wówczas istnieje $i \geq 1$ t.że $l_i \leq n < l_{i+1}$. Zauważmy, że $SUM(x) = REM_{l_{i+1}}(x)$ dla dowolnego $x = (x_{n-1}...x_1x_0)_2$. Wynika to z faktu, iż $\sum_{i \in \{0,\dots,n-1\}} x_i \leq n < l_{i+1}$ oraz definicji (20). Z lematu (5.1), obserwacji (3.2) oraz założenia (ii) dostajemy $REM_{l_{i+1}} \leq_T^{AC^0} L(5C, 11l_{i+1}^2 P(l_{i+1}, 11l_{i+1}^2))$. Na mocy jednak założenia (i) mamy:

$$11l_{i+1}^2 P(l_{i+1}, 11l_{i+1}^2) \leq 11R(l_i)^2 P(R(l_i), 11R(l_i)^2) \tag{44}$$

Skoro jednak wielomiany P i R są rosnące oraz $l_i \leq n$ to:

$$11R(l_i)^2 P(R(l_i), 11R(l_i)^2) \leq 11R(n)^2 P(R(n), 11R(n)^2) \tag{45}$$

Dowiedliśmy zatem tezy twierdzenia $SUM \leq_T^{AC^0} L$, bowiem prawa strona powyższej nierówności jest wielomianem względem n. □

Możemy teraz wykorzystać powyższy dowód i pokazać:

Wniosek 5.3. $SUM \leq_T^{AC^0} PRIMES$

Dowód. Skorzystajmy z twierdzenia (5.2) przyjmując $L = PRIMES$, $l_i = p_{i+1}$ (gdzie p_i i-ta liczba pierwsza) oraz $R(n) = 2n$. Możliwość spełnienia założenia (ii) przy pewnej stałej C oraz wielomianie P(m,n) wynika bezpośrednio z twierdzenia (4.1). Z twierdzenia 2.3 natomiast dostajemy, że $p_{i+1} \leq 2p_i$ dla $i \geq 1$, co dowodzi prawdziwości założenia (i). □

Zapowiadane twierdzenie wynika natychmiast z powyższego wniosku, redukcji (3.4) i obserwacji (3.2).

Wniosek 5.4. $MOD_2 \leq_T^{AC^0} PRIMES$

Powołując się na redukcję (3.5) dostaniemy:

Wniosek 5.5. $MAJORITY \leq_T^{AC^0} PRIMES$

Zaobserwujmy, że redukcja $MOD_2 \leq_T^{AC^0} PRIMES$, której istnienia dowodzimy we wniosku 5.4 nie jest P-jednorodna. Powstaje pytanie, czy nie można znaleźć P-jednorodnej redukcji $MOD_2 \leq_T^{AC^0} PRIMES$? Niestety, próba odpowiedzi na to pytanie wymaga najprawdopodobniej znacznie większej wiedzy na temat rozmieszczenia liczb pierwszych niż jest nam w stanie dostarczyć obecnie teoria liczb. Poniżej prezentujemy dowód istnienia P-jednorodnej redukcji $MOD_2 \leq_T^{AC^0} PRIMES^q$ dla q nieparzystych,pierwszych przy założeniu prawdziwości hipotezy Cramera. Funkcja boolowską $PRIMES^q$ jest zdefiniowana analogicznie jak $PRIMES$ z tą różnicą, iż przyjmuje ona jako argument liczbę zapisaną w systemie o podstawie q (oczywiście nadal jej dziedzinę stanowi zbiór $\{0,1\}^\star$). Podobnie definujemy funkcję $MULT_p^q$. Nietrudno dowieść poniższego uogólnienia twierdzenia 4.2.

Twierdzenie 5.6. *Niech p, q dowolne liczby pierwsze takie, że $p \neq q$. Wówczas:*

$$MOD_p \leq_T^{AC^0} MULT_p^q \tag{46}$$

i redukcja ta jest P-jednorodna.

Potrzebne nam będzie dodatkowo następujące twierdzenie.

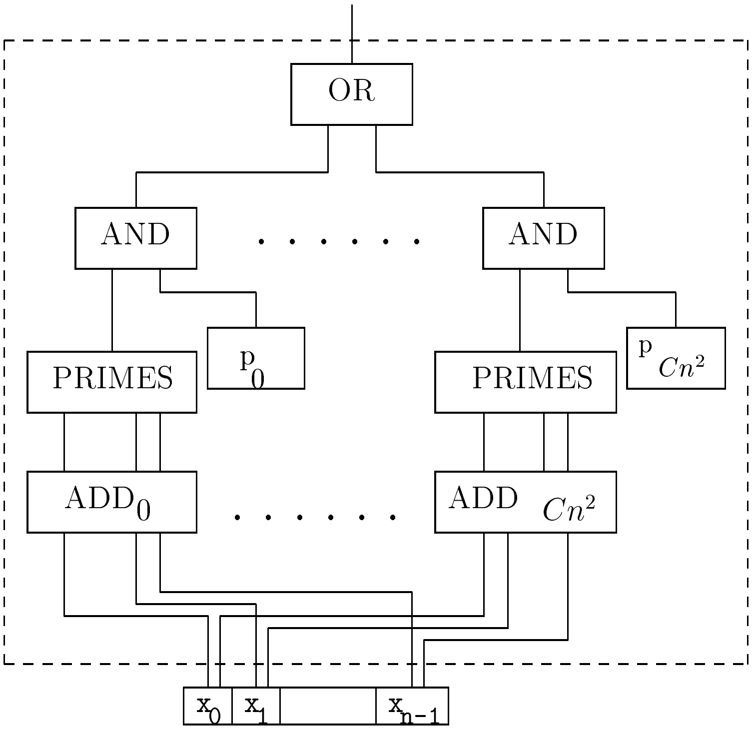
Twierdzenie 5.7. *Niech q będzie liczbą pierwszą. Wówczas jeżeli hipoteza Cramera jest prawdziwa to:*

$$MULT_2^q \leq_T^{AC^0} PRIMES^q \tag{47}$$

Co więcej, redukcja powyższa jest P-jednorodna.

Dowód. Z hipotezy Cramera wynika, iż dla dowolnej liczby naturalnej x w przedziale $[x, x + Clog^2(x)]$ znajduje się przynajmniej jedna liczba pierwsza ($C > 0$ jest pewną stałą wynikającą z tej hipotezy). Niech n dowolna liczba naturalna. Pokażemy konstrukcję ciągu obwodów $(C_n)_{n\geq 1}$ rozpoznających język $MULT_2^q$ przy użyciu bramek AND, OR, NEG oraz $PRIMES^q$. Poniżej znajduje się schemat obwodu C_n. Stałe p_i dla $i \in (0, Cn^2)$ występujące na tym schemacie są zdefiniowane w następujący sposób:

$$p_i = \begin{cases} 1, & \text{jeżeli } p_i \text{ jest liczbą nieparzystą} \\ 0, & \text{w przeciwnym przypadku .} \end{cases} \tag{48}$$



Bramki ADD_i użyte powyżej obliczają sumę liczby pojawiającej się na ich wejściu oraz indeksu i. Na mocy twierdzenia 2.7 są one oczywiście realizowalne przy pomocy obwodów o stałej głębokości. Pozostaje dowieść, że ciąg obwodów $(C_n)_{n\geq 1}$, których rozmiar jest wielomianowy, zaś głębokość stała rozpoznaje funkcję $MULT_2^q(x)$. Załóżmy, że x jest liczbą parzystą. Zauważmy, że na mocy hipotezy Cramera przynajmniej jedna z bramek $PRIMES$ musi dać wartość 1. Stała p_i, która jest związana z tą bramką musi mieć nieparzysty indeks bowiem w przeciwnym razie liczba pierwsza wskazywana przez bramkę $PRIMES$ byłaby parzysta. A zatem ostatecznie obwód wygeneruje 1. Podobnie można wykazać, że obwód da w wyniku 0 jeżeli x jest nieparzyste. □

Łącząc wyniki twierdzeń 5.6 oraz 5.7 dostajemy zapowiadany rezultat.

Twierdzenie 5.8. *Dla dowolnej nieparzystej liczby pierwszej q istnieje P-jednorodna redukcja $MULT_2^q$ do $PRIMES$.*

Bibliografia

[1] E. Allender, M. Saks and I. Szparliński *A Lower Bound for Primality*, ECCC Electronic Colloquium on Computational Complexity, Report No. **10** (1999)

[2] E. Allender, M. Saks and I. Shparlinski *A Lower Bound for Primality*, ECCC Electronic Colloquium on Computational Complexity, **Comment 1** on Report No. **10** (1999)

[3] H. Straubing *Finite Automata, Formal Logic and Circuit Complexity* , Progress in Theoretical Computer Science, Birkhauser (1994), 126-152

[4] I. Wegener *The complexity of boolean functions* , John Wiley Sons Ltd. (1987)

[5] W. Narkiewicz *Teoria liczb* , Biblioteka Matematyczna Tom 50, PWN (1990),

[6] P. Ribenboim *Mała księga wielkich liczb pierwszych* , Wydawnictwa Naukowo-Techniczne, (1997),

[7] J. Hastad *Computational Limitations for Small Depth Circuits.* MIT Press, Cambridge, MA, 1987

[8] R.H. Hudson *A formula for the exact number of primes below a given bound in any arithmetic progression.* Bull. Austral. Math. Soc., 1977

[9] S.W. Jabłoński *Wstęp do matematyki dyskretnej.* PWN, Warszawa, 1991