Cel: Ackermann-trudność osiągalności w VASSach.

Lipton '76 ExpSpace-trudność

Mayr '81 rozstrzygalność

Leroux, Schmitz '15 cubic-Ackermann

Leroux, Schmitz '19 Ackermann

Cz., Lazić, Lasota, Leroux, Mazowiecki '19 Tower-trudność

niezależnie, nieco inne techniki → Leroux & → Cz., Orlikowski Ackermann-trudność '21

Lasota ulepszenie argumentu '21

Tw. 1 (Lasota)

Problem osiągalności dla $(3d+2)$-VASSów jest $\mathcal{F}_d$-trudny.

Przypomnienie: $F_1(n) = 2n$, $F_{k+1}(n) = \underbrace{F_k \circ \dots \circ F_k}_{n}(1)$.

$F_2(n) = 2^n$ $F_3(n) = \left.2^{2^{\cdot^{\cdot^{2}}}}\right\} n$

$\mathcal{F}_d$ - klasa problemów rozwiązywalna w czasie (lub pamięci) rzędu $F_d(n)$, gdzie n to rozmiar wejścia.

$Ack(n) = F_n(n)$

$\mathcal{F}_d$-trudność (chyba) implikuje Ackermann-trudność, więc wystarczy udowodnić Tw. 1.

Można też argumentować inaczej: nasza konstrukcja da też Ackermann-trudność bezpośrednio.

Z czego redukujemy?

Automat d-licznikowy to d-VASS z zero-testami

Lemat 2

Dla $d \geq 3$ problem istnienia $F_d(n)$-ograniczonego biegu dla automatów 2-licznikowych jest $\mathcal{F}_d$-zupełny.

Bieg jest B-ograniczony gdy w każdej konfiguracji:
- każdy licznik jest $\leq B$
- suma liczników jest $\leq B$

↑ dla obu definicji Lemat 2 działa.

D-d (Lematu 2)

Należenie do $\mathcal{F}_d$ - po prostu przestrzeń konfiguracji jest rzędu F_d.

$\mathcal{F}_d$-trudność - tak samo jak nierozstrzygalność dla automatów 2-licznikowych, tylko startujemy z $\mathcal{F}_d$-trudnego problemu:

dana maszyna Turinga, liczba $n \in \mathbb{N}$, czy maszyna ma bieg akceptujący, który używa $\leq F_d(n)$ pamięci? □

Uwaga

Podobnie problem exp-ograniczony dla automatów 3-licznikowych jest PSpace-zupełny, a 2exp-ograniczony jest ExpSpace-zupełny.

Automat 2-licznikowy możemy symulować 2-VASSem, ale wyzwaniem są:

a) zapewnienie, że liczniki są B-ograniczone

b) zero-testy dla B-ograniczonych liczników.

Zakładamy bez straty ogólności, że zaczynamy z wartości liczników $(0,0)$.

Recepta na a) to dołożyć B.

Niech $x = y = 0$, $z = B$.

Wówczas trzymając $x + y + z = B$, czyli $z = B - x - y$ zapewniamy, że każdy bieg nowego VASSa jest B-ograniczony.

Okazuje się, że recepta na b) to trójka liczników (B, C, BC) dla wystarczająco dużych C. To jest pomysł Jerome'a Lerouxa z pracy o Tower-trudności.

Fakt

Trójka (B, C, BC) pozwala na $C/2$ zero-testów na B-ograniczonych licznikach.

D-d

Niech $(x, y, z) = (B, C, BC)$

Niech $x' = 0$.

Trzymamy $x + x' = B$ (więc x był B-ograniczony)

zero-test(x'):

transfer(x, x', z) → loop $x{-}{-}$ $x'{+}{+}$ $z{-}{-}$

transfer(x', x, z)

$y \mathrel{-}= 2$

z obniżony o maksymalnie $2B$, obniżenie o dokładnie $2B$ możliwe tylko gdy $x' = 0$ przed i po teście

Po $C/2$ zero testach sprawdzamy, czy $z = 0$

Równość $z = 0$ jest możliwa tylko gdy wszystkie $C/2$ zero-testów było poprawne!

Teraz dla kilku liczników

Niech $(x, y, z) = (B, C, BC)$

$x_1 = x_2 = \dots = x_k = 0$

Trzymamy $x_1 + x_2 + \dots + x_k + x = B$.

W zero-test(x_1) chcemy mieć możliwość obniżenia z o $2B \iff x_1 = 0$ oraz test nie zmienia wartości liczników

Robimy tak:

zero-test(x_1):

transfer(x_2, x_1, z)

transfer(x_3, x_2, z)

⋮

transfer(x_k, x_{k-1}, z)

transfer(x, x_k, z)

$\left.\begin{array}{l}\text{transfer}(x_k, x, z)\\ \vdots \\ \text{transfer}(x_2, x_3, z)\\ \text{transfer}(x_1, x_2, z)\end{array}\right\}$ w odwrotną stronę

Każdy token z $x_1 + \dots + x_k + x = B$ musi się ruszyć 2 razy, token z x_1 tylko raz. Dlatego odejmujemy $2B$ z licznika z wtw. gdy wszystko zostało wykonane w pełni oraz $x_1 = 0$.

Podobnie robimy zero-test(x_i) dla dowolnego x_i. □

Czyli pozostaje kwestia wygenerowania trójki $(F_d(n), C, F_d(n) \cdot C)$ dla dużego C. Będziemy zgadywać C, choć wystarczy rzędu $2 \cdot |Q| \cdot F_d(n)$, bo tyle może być zero-testów.

Idea: generujemy trójki (B, C, BC) dla małych B, a z nich trójki $(B', C', B'C')$ dla większych B'.

Przykład jak zrobić $B' = 2^B$.

(B, C, BC) umożliwia też $B/2$ zero-testów na C-ogr. licznikach, czyli dowolnych skoro C jest zgadnięte (dowolnie duże).

Startujemy $x = y = z = 0$, $u = 0$

$x{+}{+}$

loop $y{+}{+}$ $z{+}{+}$ — $(1, C', C')$

loop

flush$(x, u, 1)$ — loop $x{-}{-}$ $u \mathrel{+}= 1$

zero-test(x)

flush$(u, x, 2^8)$ → loop $u{-}{-}$ $x \mathrel{+}= 2^8$

zero-test(u)

flush$(z, u, 1)$

zero-test(z)

flush$(u, z, 2^8)$

zero-test(u)

Mamy $B/2$ zero-testów, więc $B/8$ iteracji pętli.

Dostaniemy $x = (2^8)^{B/8} = 2^B$

$z = y \cdot 2^B$,

czyli $(x, y, z) = (2^B, C', 2^B \cdot C')$.

F-amplifikator

$\underset{x_1\ \ x_2\ \ x_3}{(B, m, Bm)} \longrightarrow \boxed{\text{jeśli } Bm \text{ spadnie do } 0} \longrightarrow \underset{x_4\ \ x_5\ \ x_6}{(F(B), m', F(B) \cdot m')}$

Wystarczy mieć F_d-amplifikator, zrobimy F'_d-ampl., dla $F'_d \approx F_d$.

Lemat

Istnieje F'_d-amplifikator w wymiarze $3d+2$, dla $d \geq 2$.

D-d

Przez indukcję po d.

Dla $d = 2$ mieliśmy przykład.

Mamy (B, m, Bm), używamy do $B/2$ zero-testów.

$x = y = z = 0$

$x{+}{+}$

loop $y{+}{+}$ $z{+}{+}$

repeat $B/12$ times:

apply $(x, y, z) \xrightarrow{F_{k-1}\text{-ampl.}} (x', y', z')$

zero-test(x, y, z)

transfer $x' \to x$, $y' \to y$, $z' \to z$

zero-test(x', y', z')

Dostajemy $(x, y, z) = (F'_d(B), m', F'_d(B) \cdot m')$,

gdzie $F'_{k+1}(n) = \underbrace{F'_k \circ \dots \circ F'_k}_{n/12}(1)$

$n/12$ ← zamiast n, ale to mała różnica.

Dodaliśmy tylko 3 liczniki (te z (B, m, Bm)), więc faktycznie mamy $3d+2$ wymiar.

Chyba to w ogóle poprawia konstrukcję Sławka Lasoty do $3d+1$, a po drobnym triku do $3d$. □