Cel: wstęp do VASSów

Za tydzień: rozstrzygalność osiągalności

Za dwa tygodnie: Ackermann-trudność

Plan:
- ciekawy przykład
- pokrywalność w ExpSpace (technika Rackoffa)
- techniki przydatne w konstrukcjach długich biegów (wybrane)

Dużo o niskich wymiarach, które wydają się ciekawe. Co wiadomo?

Kodowanie liczb w tranzycjach

	unarne	binarne
1-VASS	NL-zup. Łatwe	NP-zup. Haase et al. '09
2-VASS	NL-zup. Englert et al. '16	PSpace-zup. Blondin et al. '15
3-VASS	NL-tr. w $\mathcal{F}_7$	PSpace-trudność w $\mathcal{F}_7$

← ćwiczenie

↑ wiel. bieg

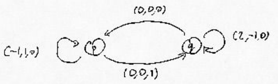
↖ ↗ trudne

↑ w 2-VASSie zawsze jest wykładniczy bieg

↑ możliwe, że zawsze wykł. bieg

W 2-VASSach zbiór osiągalności i relacja też są semiliniowe. W 3-VASSach nie.

Słynny przykład z pracy Hopcroft, Pansiot '79



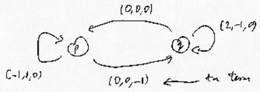
$$\begin{tikzcd} p(k,0,n) \arrow[r] & p(0,k,n) \arrow[r] & q(0,k,n+1) \arrow[r] & q(2k,0,n+1) \arrow[d] \\ & & & p(2k,0,n+1) \end{tikzcd}$$

$$p(1,0,0) \to p(x,y,n)$$
$$\Downarrow$$
$$x+y \le 2^n$$

Reach(p(1,0,0)) nie jest semiliniowy

* * *

Prawie ten sam przykład pokazuje, że zbiór osiągalności może być skończony, ale wykładniczy (dla unarnych 3-VASSów).



← tu teraz -1 zamiast 1

$$p(k,0,n) \to p(2k,0,n-1)$$
$$p(1,0,n) \to p(2^n,0,0) \to p(2^n-a,a,0)$$
dla $a \in \{0,1,\dots,2^n\}$

Ale Reach (1,0,n) jest skończony, łatwo to zobaczyć.

Problem Pokrywalności

Dane: VASS V, konfiguracje s,t

Pytanie: czy $s \xrightarrow{V} t'$ t.że $t' \geq t$?

Lipton '76: ExpSpace-trudność

Rackoff '78: w ExpSpace (istnieje bieg $\leq$ 2-exp)

Tw. (Rackoff, zmodyfikowana wersja)

Rozważmy d-VAS o normie M i dwie jego konfiguracje $s,t \in \mathbb{N}^d$ t.że norma(t) $\leq M$ (norma(s) może być dowolna). Jeśli jest bieg z s pokrywający t, to jest również taki bieg o długości $\leq (M+1)^{(4d)^{d-1}}$.

maks. z wszystkich liczb

D-d

Indukcja. Dla $d=1$ łatwo.

Załóżmy tezę dla d, dowodzimy dla $d+1$.

Niech $s \xrightarrow{\rho} t'$ w $d+1$-VASSie.

Niech $B_d = (M+1)^{(4d)^{d-1}}$.

Są dwa przypadki:

1) norma każdej konf. na ρ jest $< (M+1)\cdot B_d$

2) przeciwnie.

W przypadku 1) możemy oszacować długość biegu przez $C = ((M+1)\cdot B_d)^d$. Potem pokażemy, że $C \leq B_{d+1}$.

W przypadku 2) mamy

$$s \xrightarrow{\rho_1} u \xrightarrow{\rho_2} t'$$

pierwsza konf. o normie $\geq (M-1) B_d$

Dł. ρ_1 jest $\leq C$, co wynika z poprzedniego rozumowania.

Załóżmy bez straty ogólności, że $u[d+1] \geq (M+1)B_d$.

Z zał. ind. $u[1..d]$ pokrywa $t[1..d]$ ścieżką dł. $\leq B_d$. Ta ścieżka jest też OK na wsp. $d+1$. Czyli $u \xrightarrow{\rho_2'} t''$ t.że $t'' \geq t$ oraz dł. $\rho_2' \leq B_d$. Teraz wystarczy pokazać, że

$$C + B_d \leq B_{d+1}.$$

Mamy:

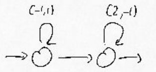
$$C+B_d \leq (M+1)\cdot C = (M+1)\cdot\left((M+1)\cdot B_d\right)^{d+1}$$
$$= (M+1)\left((M-1)\cdot(M+1)^{(4d)^{d-2}}\right)^{d+1}$$
$$= (M+1)^{((4d)^{d-1}+1)(d+1)+1} \leq (M+1)^{4\cdot(4d)^{d-1}(d+1)} \leq (M+1)^{(4(d+1))^d} = B_{d+1}.$$

□

B_d jest wciąż 2-exp, czyli OK.

Teraz techniki, które pozwalają na tworzenie długich biegów

Programy licznikowe to inny sposób prezentacji VASSów, przykładowo.



```
loop  x -= 1  y += 1
loop  x += 2  y -= 1
```

Cel: unarny 3-VASS z wykł. biegiem

Bazujemy na iloczynie teleskopowym

$$\frac{k}{k-1}\cdot\frac{k-1}{k-2}\cdot\ldots\cdot\frac{3}{2}\cdot\frac{2}{1} = k$$

Następujący 3-VASS mnoży po kolei licznik y przez maksymalnie $\frac{i}{i-1}$, a na koniec wartość y może być Nk (gdy była N na początku).

A więc w poprawnym biegu wszystkie mnożenia muszą być dokładne, co wymusza wykładniczy bieg.

```
x += 1   y += 1
loop   x += 1   y += 1            ← tu wartość to (N,N,0)
for i = k downto 2 do
    loop   y -= 1   z += 1        } słabe mnożenie y przez $\frac{i}{i-1}$
    loop   y += i   y -= (i-1)    }
loop   x -= 1   y -= k            ← (N,Nk,0) w poprawnym biegu
```

Po ostatniej pętli for w poprawnym biegu musi być

$$\left(N, N\cdot\frac{k}{k-1}\cdot\frac{k-1}{k-2}\cdot\ldots\cdot\frac{i+1}{i}, 0\right) = \left(N, \frac{Nk}{i}, 0\right)$$

Zatem $\forall_{i \in \{2,\dots,k-1\}}\ i \mid Nk$, czyli

NWW$(2,\dots,k-1) \mid Nk$, czyli N wykładnicze.

* * *

Technika controlling-counter

Przykład: chcemy przetestować licznik x trzy razy na zero, w konf. C_1, C_2, C_3.

$$\overset{0}{C_0} \xrightarrow{y_1} \overset{x_1}{C_1} \xrightarrow{y_2} \overset{x_2}{C_2} \xrightarrow{y_3} \overset{x_3}{C_3} \longrightarrow$$

Chcemy $x_1 = x_2 = x_3 = 0$.

Wystarczy sprawdzić, czy $x_1+x_2+x_3 = 0$.

Pomysł: stworzymy licznik c (controlling counter), który na końcu biegu będzie liczył $x_1+x_2+x_3$ i w docelowej konf. będzie miał mieć wartość 0.

$x_1 = y_1$

$x_2 = y_1 + y_2$

$x_3 = y_1 + y_2 + y_3$

$$x_1+x_2+x_3 = 3y_1 + 2y_2 + y_2$$

Każdy x+=1 przed k zero-testami jest symulowany przez c+=k. Podobnie x-- jest sym. przez c-=k.

To działa też dla wielu liczników.

Przykład

```
                       lub downto
                       ↙
        for i=1 to k do                          → (2^k,0)
(1,0) →     loop  x -= 1   y += 2
            loop  x += 1   y -= 1                → (1,0)
```

Po dodaniu licznika kontrolującego c:

```
          for i = k downto 1 do
              loop  x -= 1   y += 2   c += i          (2^k,0,c)
(1,0,0) →                                        →
              loop  x += 1   y -= 1   c -= 1          o ile c=0
```