Plan:
- separowalność, kontekst badań
- regularna separowalność dla 1-VASS-ów
- zbiory semilinijne
- pojęcie algorytmu
- kluczowy lemat
- rozstrzygalne warunki

$K, L \in \mathcal{F}$

$S \in \mathcal{G}$

G-separowalność dla $\mathcal{F}$

Jeśli $\mathcal{F}$ efektywnie zamknięta na dopełnienie, to

$\mathcal{G}$-sep. dla $\mathcal{F}$ wygodnie pokazać, czy element

$\mathcal{F}$ należy do $\mathcal{G}$:

$K, \overline{K}$ $\mathcal{G}$-sep. $\Leftrightarrow K \in \mathcal{G}$

Historia:

- wiele wyników dla $\mathcal{F}$ regularne języki w $\mathcal{G}$ (od podklas, np. deterministyczne przez podklasy, FO - definiowalne itd.
- w 1975 Stigmanitski i Williams pokazali, że regularna separowalność języków bezkontekstowych jest nierozstrzygalna (dowód na ćwiczeniach)
- 1982 Hunt pokazał, że $\mathcal{F}$-sep. dla języków bezkontekst. jest nierozstrzygalna o ile $\mathcal{F}$ zawiera skończone boolowskie kombinacje $w\Sigma^*$ dla $w \in \Sigma^*$
- to chyba zostawiło badania dla $\mathcal{F} \ni$ Reg
- w 2015 pokazano, że PTL-zapytania jest rozstrzygalna dla:
  - CFL (języków bezkontekstowych)
  - języków VASS-ów
  - itd.
- Hipoteza Regularna separowalność jest rozstrzygalna dla języków VASS-ów (akceptacja przez stan / przez konfigurację).
  Otwarta
- Wyniki dla podklas:
  - ZZ-VASS-y (Clementowy algorytm)
  - 1-VASS-y (PSpace-sep.), regularność jest nierozstrzygalna
  - VASS-y z zakaz. przez konf.
  - itd.

Dla 2-VASS-ów otwarte.
Dla OCA reg-sep nierozstrzygalna.

Dziś: Reg-sep. jest rozstrzygalna dla 1-VASS-ów (Cadilhac).

Cel: Testowanie + wyprowadzenie zbiorów semilinijnych.

Niech K, L to języki 1-VASS-ów.

Reg-sep. jest „podobne" do niepustego przecięcia.

$K \cap L = \emptyset$? redukuje się do osiągalności w produktowym 2-VASS-ie.

Fakt Relacja osiągalności w 2-VASS-ie jest semilinijna i efektywnie obliczalna.

(tego używamy w algorytmie) ↑ ↑ (bardzo ważne pojęcie)

Zbiór $S \subseteq \mathbb{N}^d$ jest liniowy jeśli jest postaci:
$b + P^* = b + \mathbb{N}p_1 + \dots + \mathbb{N}p_k$,
gdzie $b \in \mathbb{N}^d$, $P \subseteq \mathbb{N}^d$, $P = \{p_1, \dots, p_k\}$

Zbiór jest semilinijny jeśli jest skończoną sumą skończonej liczby zbiorów liniowych.

Przykłady
- zbiór skończony
-

Fakt
Zbiory semilinijne są zamknięte na:
- skończone sumy ⇐ jasne
- skończone przecięcia
- dopełnienie ⇐ nieoczywiste

Wracamy do reg-sep 1-VASS-ów.

Tw.
Regularna separowalność 1-VASS-ów jest rozstrzygalna.
(Stała wykładnicza jest PSpace-zupełna)

Fakt
K, L języki 1-VASS-ów reg (akceptacja przez stan),
$K \cap L = \emptyset \Rightarrow K \mid_{\text{reg}} L$.

Czy dla akceptacji przez konfigurację też?

NIE!

Np. $\{a^n b^n \mid n \in \mathbb{N}\}$ i $\{a^* b^k \mid k \in \mathbb{N}, k \text{ stałe}\}$
są językami 1-VASS-ów (łatwo), ale nie są reg-sep. (ćwiczenia).

Pomysł: aproksymanty.

Idea: A to 1-VASS, nad $\mathbb{N}$, A_n to automat skończony, który liczy moduło n (z pomocą skończoności stanów).

Konkretnie A_n ma stany
$Q_n = Q \times \{0, 1, \dots, n-1\} \times \{\text{LOW}, \text{HIGH}\}$,
stan początkowy $(q_0, 0, \text{LOW})$, stan końcowy $(q_f, 0, \text{LOW})$
oraz dla tranzycji (q, a, q', z) A ma
$0 \le c < n$ ma:

$(q, c, \text{LOW}) \xrightarrow{} (q', c+z, \text{LOW})$ $\quad 0 \le c+z < n$

$(q, c, \text{LOW}) \xrightarrow{} (q', (c+z) \bmod n, \text{HIGH})$ $\quad n \le c+z$

$(q, c, \text{LOW}) \xrightarrow{} (q', (c+z) \bmod n, \text{LOW})$ $\quad c+z < 0$

$(q, c, \text{LOW}) \xrightarrow{} (q', (c+z) \bmod n, \text{HIGH})$

Fakt 1
Niech A to 1-VASS, $n, k > 0$. Wtedy:
$L(A) \subseteq L(A_n)$
$L(A_n) \subseteq L(A_k)$ jeśli $n \mid k$.

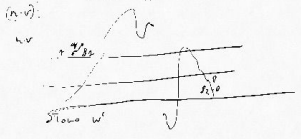
Nice to have
Tw.
Niech A, B to 1-VASS-y. Wówczas n.w.o.w.:
1. $L(A)$ i $L(B)$ są reg-separowalne
2. $L(A_n)$ i $L(B_n)$ są rozłączne dla pewnego $n > 0$
Idea: $L(A_n)$ i $L(B_n)$ są rozłączne dla pewnego $n > 0$.

D-d
jeśli coś separuje $L(A)$ i $L(B)$ to pewien $L(A_n)$.
3 ⇒ 2.

1. ⇒ 3
Niech n spełnia 2. ⇒ 1. Oczywiste.

Niech N będzie większe niż wszystkie liczby występujące w macierzach przejścia A i B oraz większe niż n.
Niech $\mathcal{R}$ będzie zbiorem (q, c, v) takich, że istnieje słowo w takie, że $(q_0, 0, \text{LOW}) \xrightarrow{w} (q, c, v)$ oraz $|w| < N$.
Niech $\mathcal{R}$ będzie zbiorem (q, c, v) takich, że istnieje słowo w takie, że $(q_0, 0, \text{LOW}) \xrightarrow{w} (q, c, v)$ oraz $|w| < N$.
Zatem L_n ma stanów. Niech $N = (n \cdot \nu)!$
Pokażemy, że wystarczy, że $w \in L(A_N) \cap \mathcal{R}$.

Jeśli $w \in L(A_N) \cap \mathcal{R}$,
to w' że istnieje $w' \in L(A) \cap \mathcal{R}$.
w którym cały czas bez wchodzenia w HIGH,
$L(A)$. w_{pp}.



Szczegóły pomijamy przez pompowanie y z xyz.

na wykładzie □

Algorytm ogólny
istnienia n t.ż.
Najpierw sprawdzamy warunek 2., tzn. czy
redukujemy do $L(A_n) \cap L(B) = \emptyset$.
Zakładamy dalej, że $L(A) \cap L(B) = \emptyset$, przy czym
zajmujemy się osiągalnością w 2-VASS-ach.
Jeśli $w \in L$
to $L(A) \cap L(B) = \emptyset$
$w = w_1 w_2 w_3 (w_2 A_n) \cap L(B)$, to

$((p_0, \gamma_0), 0, 0)$ ma t, że

$((p, \gamma), k, \ell) \xrightarrow{w_2} ((p, \gamma), k, \ell)$ w (N, N)

$((p, \gamma), k', \ell') \xrightarrow{w_3} ((p', \gamma'), k'', \ell'')$ w $(\mathbb{Z}, N)$

$\xrightarrow{w_3} ((p_f, \gamma_f), 0, 0)$ w $(\mathbb{Z}, N)$

n-osiągalność

Fakt $k, k' \ge n$ $n \mid \Delta$
$L(A) \mid_{\text{reg}} L(B) \Leftrightarrow n$-o

$P_2 = \{(k, \ell)$: warunek 2$\}$ (osiągalność dla każdego n)
$P_2 = \{(k, \ell, k-\Delta, \ell')$.
$P_3 = \{(k, \ell), (k', \ell')$: warunek 3$\}$
P_1, P_2, P_3 semilinijne i duże
$R = \{(k, k', \Delta)$: $\exists_{\ell, \ell'}$ które się obliczyć$\}$
R też semilin. i da się efektywnie policzyć

Fakt obliczyć
Wystarczy sprawdzić, czy dla każdego
istnieje $k, k' \ge n$ Δ t.ż.
dużego $n \in \mathbb{N}$
$(k, k', \Delta) \in R$.
(n-świadek) $n \mid \Delta$ t.że

Fakt
Wystarczy rozważyć R liniowe.

Fakt
$L = b + P^*$ zawiera n-świadków $\Leftrightarrow$

a) $\forall_{n \in \mathbb{N}}$ $\exists_{(k, k', a) \in L}$ t.że (dla wszystkich $n \in \mathbb{N}$)

b) $\forall_{n \in \mathbb{N}}$ $\exists_{(k, k', a) \in L}$ t.że $k, k' \ge n$
$n \mid \Delta$

Warunek a) Tutaj sprawdzamy.
Warunek b): $b[3] + P[3]^*$ ma zaw
wiązkę
⇕ wiele wektorów
$b[3] \in \ln(P[3]$ gdzie wektor

⇑ tutaj)
⇓ Tego się da.

Ćwiczenia: Mogą być na ćwiczeniach.