Ćwiczenia 3 16.10

Alfabet z zależnością (Σ, D), gdzie

* $D \subseteq \Sigma^2$ relacja zależności zwrotna, symetryczna
* $I = \Sigma^2 \setminus D$ relacja niezależności

Równoważność śladowa $\equiv_D \subseteq \Sigma^* \times \Sigma^*$ to najmniejsza równoważność, taka że $xaby \equiv_D xbay$ dla każdych $x, y \in \Sigma^*$ i $(a,b) \in I$.

Inne spojrzenie na $\equiv_D$:

* niech $\sim$ będzie binarną relacją na Σ^*, taką, że $u \sim v$ wtw, gdy istnieją $x, y \in \Sigma^*$ i $(a,b) \in I$ spełniające $u = xaby$, $v = xbay$
* wtedy $\equiv_D$ jest symetrycznym, zwrotnym i przechodnim domknięciem $\sim$, innymi słowy $u \equiv_D v$ wtw, gdy istnieje sekwencja $(w_0, w_1, \dots, w_n)$, taka że $w_0 = u$, $w_n = v$ i $\forall\, 0 < i \leq n \quad w_{i-1} \sim w_i$

Śladem nazywamy klasę abstrakcji $\equiv_D$, przyjmujemy oznaczenie $[w]_D = \{v : v \equiv_D w\}$. Gdy D wynika z kontekstu, pomijamy je, pisząc $[w]$. **Językiem śladów** nazywamy dowolny ich podzbiór.

Język śladów T nad (Σ, D) jest **regularny**, jeśli istnieje istnieje język regularny $L \subseteq \Sigma^*$, który jest zamknięty na równoważność śladową i spełnia warunek:

$$T = [L] = \{[w] : w \in L\}$$

dla języków śladów analogicznie

Dla języka $L \subseteq \Sigma^*$ i słowa u **ilorazem lewostronnym** L względem u nazywamy $u^{-1}L = \{v : uv \in L\}$

Przykłady:

te pary liter można zamieniać w słowie

* $\Sigma = \{a, b, c\}$, $D = a \begin{array}{l} \diagup b \\ \diagdown c \end{array}$ $\Rightarrow$ $I = \{(b,c), (c,b)\}$

$[abbca]_D = \{abbca, abcba, acbba\}$

* $\Sigma = \{a, b, c\}$, $D = a - b \quad c$

$L = \{ab, abc, acb, cab\}$ zamknięty na $\equiv_D$

* $\Sigma = \{a, b, c\}$, $L = \{ab, acb, abca, acaa, acbca\}$

  $(ab)^{-1}L = \{\varepsilon, ca\} = (acb)^{-1}L$

* $\Sigma = \{a, b\}$ $L = (ab)^*$

  istnieją tylko dwa ilorazy lewostronne języka L:

  $L = \varepsilon^{-1}L = (ab)^{-1}L = (abab)^{-1}L = \ldots$,

  $bL = a^{-1}L = (aba)^{-1}L = (ababa)^{-1}L = \ldots$

Inne spojrzenie na ilorazy:

* słowa $x, y \in \Sigma^*$ mają taką samą przyszłość w języku L, oznaczenie $x \sim_L y$, jeśli nie istnieje $z \in \Sigma^*$, takie że $xz \in L \wedge yz \notin L$ lub $xz \notin L \wedge yz \in L$
* oczywiście $x \sim_L y \iff x^{-1}L = y^{-1}L$

Twierdzenie (Myhill, Nerode)

Język L jest regularny wtw $\sim_L$ ma skończoną liczbę klas abstrakcji. Co więcej, jest to liczba stanów w minimalnym automacie deterministycznym rozpoznającym język L.

1. Pokaż, że język śladów T jest regularny wtedy i tylko wtedy, gdy ma skończenie wiele ilorazów lewostronnych.

Mamy do udowodnienia dwie implikacje

1) T regularny $\Rightarrow$ ma skończenie wiele ilorazów

* oznaczmy alfabet z zależnością przez (Σ, D)
* niech $L \subseteq \Sigma^*$ będzie językiem regularnym zamkniętym na równoważność śladową, który odpowiada T
* L regularny, czyli z twierdzenia M-N ma skończenie wiele ilorazów lewostronnych
* oznaczmy je przez $L_1, L_2, \dots, L_k$
* załóżmy, że T ma $> k$ ilorazów lewostronnych i weźmy dowolne $k+1$ z nich oraz dla każdego wybierzmy dowolny ślad go generujący:

$$[s_1]^{-1} T_1, [s_2]^{-1} T_2, \dots, [s_{k+1}]^{-1} T_{k+1}$$

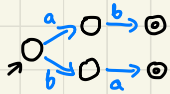
* z zasady szufladkowej co najmniej dwa ze słów $s\_$ generują ten sam iloraz lewostronny dla języka L, niech będą to s_i oraz s_j
* z drugiej strony $T_i \neq T_j$, więc BSO możemy przyjąć $[t] \in T_i$ i $[t] \notin T_j$ dla pewnego $t \in \Sigma^*$
* prowadzi to do sprzeczności, bo $s_i t \in L$ i $s_j t \in L$, a co za tym idzie (z zamkniętości L na $\equiv_D$) $\forall s' \in [s_i] \, \forall t' \in [t] \; s't' \in L$ i $\forall s' \in [s_j] \, \forall t' \in [t] \; s't' \in L$, czyli $[t]$ nie może rozróżniać przyszłości $[s_i]$ i $[s_j]$
* tym samym udowodniliśmy, że T ma dokładnie k ilorazów lewostronnych: $[L_1], [L_2], \ldots, [L_k]$

2) T ma skończenie wiele ilorazów $\Rightarrow$ jest regularny

* pokazujemy, że język $L = \bigcup \{ s : s \in T \}$ jest regularny i zamknięty na $\equiv_D$
* kolejne kroki przebiegają analogicznie do tych z pierwszej części dowodu

2. Automat deterministyczny jest „diamentowy”, jeśli dla każdego stanu q i każdych dwóch niezależnych liter a, b, takich że $q \xrightarrow{a} q'$ i $q \xrightarrow{b} q''$, istnieje stan p, spełniający $q' \xrightarrow{b} p$ oraz $q'' \xrightarrow{a} p$. Czy każdy język regularny zamknięty na równoważność śladową jest rozpoznawany przez automat „diamentowy”?

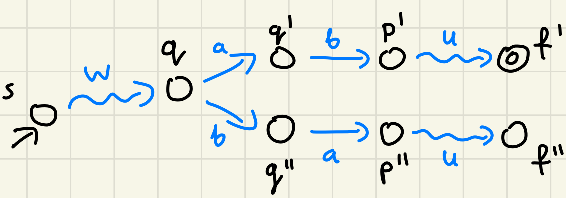
* weźmy dowolny deterministyczny automat A rozpoznający dany język L
* potencjalny problem:



może mieć za dużo stanów

* przekształcamy A, żeby otrzymać automat minimalny A' rozpoznający L (np. algorytmem Hopcrofta, $n \log n$)
* zgodnie z twierdzeniem M-N każdy stan w A' reprezentuje inną klasę abstrakcji relacji $\sim_L$ (w szczególności istnieje stan „śmietnik”)
* pokażemy, że A' jest automatem diamentowym

* załóżmy, że $q \xrightarrow{a} q'$ i $q \xrightarrow{b} q''$ dla pewnego stanu q oraz niezależnych liter a, b
* skoro A' jest deterministyczny, mamy $q' \xrightarrow{b} p'$ i $q'' \xrightarrow{a} p''$ dla pewnych stanów p', p''
* jeśli $p' = p''$, własność automatu diamentowego jest spełniona, dlatego załóżmy $p' \neq p''$
* p' i p'' muszą mieć inne „przyszłości" (na mocy twierdzenia M-N)
* stąd BSO istnieje słowo u, takie że z p' bieg po u jest akceptujący, a bieg z p'' po u – nie
* wystarczy wziąć dowolne słowo w, po którym istnieje bieg ze stanu początkowego do q, oraz zauważyć, że $wabu \in L$ i $wbau \notin L$ – sprzeczność z zamkniętością L na równoważność śladową



3. Czy sieć regionów skonstruowana z grafu konfiguracji sieci S może być podwójnie wykładnicza względem S ?

* zastanówmy się, jakie są zależności między kolejnymi przejściami

sieć Petriego	graf konfiguracji	sieć regionów
$\|P\| = n$	$\|V\| \leqslant 2^n$	$\|P'\| \leqslant ?$
$\|T\| = m$	$\|E\| \leqslant \binom{2^n}{2}$	$\|T'\| = m$
etykiety (labels)	$\|L\| = m$	

* pozostaje oszacować liczbę miejsc $|P'|$ w sieci regionów - wystarczy sprawdzić ile może być maksymalnie regionów w grafie konfiguracji

* w tym celu patrzymy na graf konfiguracji i analizujemy, w jaki sposób można scharakteryzować dowolny region $R \subseteq V$

* z definicji, dla każdej etykiety $l \in L$ musimy zdecydować, którego jest ona typu:

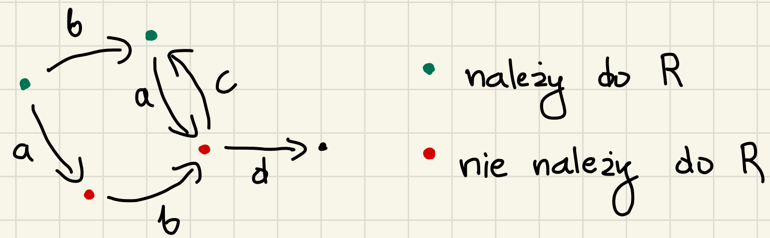
① wszystkie l-krawędzie prowadzą z R do $V \setminus R$

② —"— z $V \setminus R$ do R

③ —"— z R do R lub z $V \setminus R$ do $V \setminus R$

* pokażemy, że każdy region R można jednoznacznie opisać funkcją $f_R : L \to \{①, ②, ③\}$, która każdej etykiecie przypisuje jej typ w R

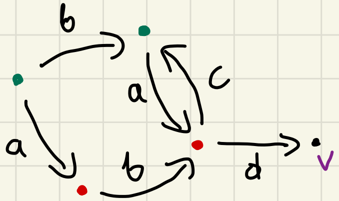
* rzeczywiście, jeśli $f_R(a) = ①$, czyli gdy wiemy, że wszystkie a-krawędzie prowadzą z R do $V \setminus R$, to wiemy już, jak rozdzielić końce wszystkich a-krawędzi, np.



* podobnie informacja, że $f_R(c) = ②$ dla pewnego $c \in L$, mówi nam jak rozdzielić końce c-krawędzi między R i $V \setminus R$

* a co z etykietami, którym przydzielono ③ ?

* robimy tak, że najpierw wyznaczamy przydział wierzchołków do R na podstawie ① i ② typu, a później „propagujemy" przynależenie do R, np.



$$f_R(d) = ③$$
$$\Downarrow$$
wierzchołek v nie należy do R, bo drugi koniec nie należy

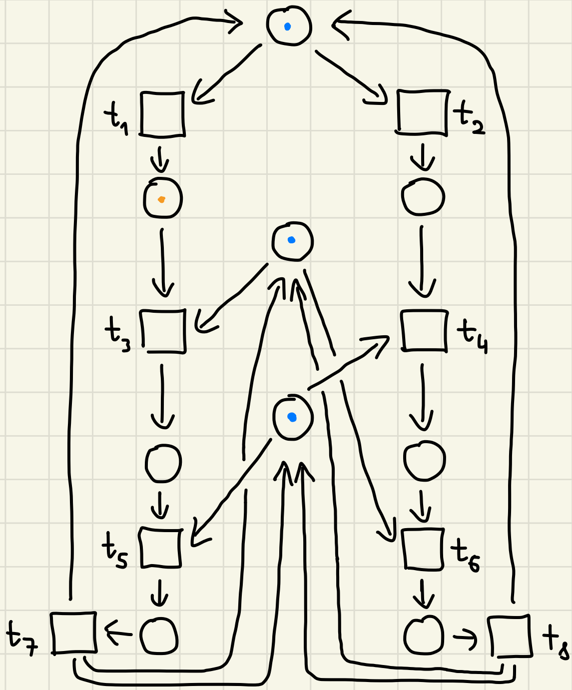
* procedura propagacji działa, bo graf konfiguracji jest spójny; pozostaje tylko szczególny przypadek, gdy $f_R(l) = ③$ dla każdego l, ale wtedy albo $R = V$, albo $R = \emptyset$, a te opcje pomijamy

* uwaga: dla ustalonego grafu konfiguracji nie każda funkcja f_R opisuje poprawny region, np. $f_R(a) = ①$, $f_R(b) = ②$ w powyższym przykładzie daje sprzeczność

* rozważenie takich funkcji f_R daje jednak górne ograniczenie na $|P'|$, bo regionów jest co najwyżej $|\{L \to \{①, ②, ③\}\}| = 3^{|L|} = 3^m$ ← pojedynczo wykładniczo

4. Czy istnieje ogólna sieć Petriego N oraz dwie konfiguracje M i M', M' > M, takie że M jest żywa i 1-ograniczona, a M' nie jest żywa?

* spójrzmy na sieć rozdzielania dwóch zasobów:

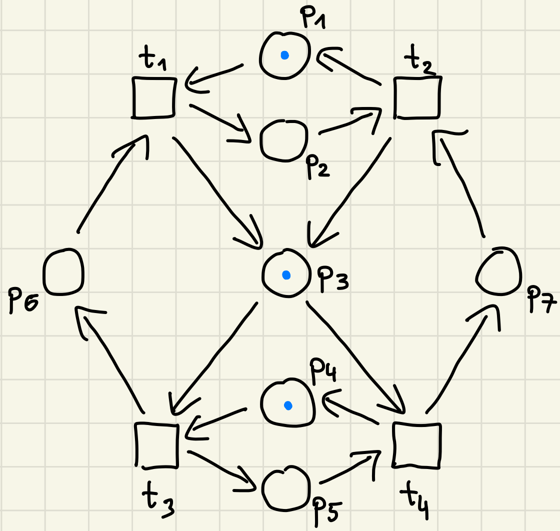


* konfiguracja niebieska jest żywa i 1-ograniczona, bo sieć za każdym razem „idzie w lewo albo prawo"

* po dołożeniu pomarańczowego żetonu i pójściu w prawo sieć może się zablokować po odpaleniu t_2, t_3, t_4

Inny kontrprzykład:

N:



$M(N) = (1,0,1,1,0,0,0)$ $\quad M'(N) = (1,0,1,2,0,0,0)$

* konfiguracja M jest żywa i 1-ograniczona, bo odpala cyklicznie:

$t_3 \to t_1 \to t_4 \to t_2 \to t_3 \to \dots$

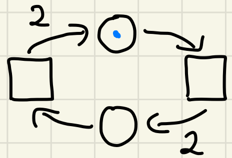
* konfiguracja M' osiąga blokadę po odpaleniu

$t_3 \to t_1 \to t_3$

w ogólności jest jednak 2-ograniczona

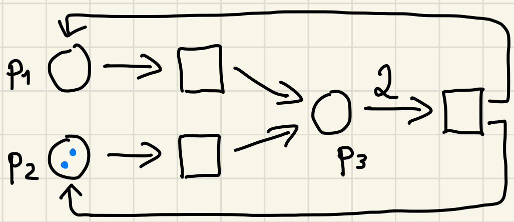
5. Ogólna sieć Petriego N z konfiguracją M jest żywa. Czy wynika stąd, że z dowolnej osiągalnej konfiguracji M' można wrócić do M?

* wystarczy wziąć sieć, która przemnaża żetony



5'. Co jeśli (N,M) jest 2-ograniczone?

* znowu można pokazać kontrprzykład



* dla dowolnej osiągalnej konfiguracji M' powyższej sieci mamy $M'(p_2) \leq 1$

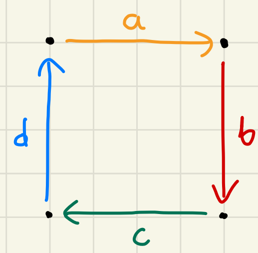
1^D. Dla danego alfabetu z zależnością (Σ, D) i języka L rozważmy pytanie, czy L jest zamknięty na równoważność śladową. Pokaż, że pytanie to jest rozstrzygalne dla L regularnego. Udowodnij, że problem staje się nierozstrzygalny, gdy założymy, że L jest bezkontekstowy.

2^D. Ogólna sieć Petriego N z konfiguracją M jest żywa i 1-ograniczona. Czy wynika stąd, że z dowolnej osiągalnej konfiguracji M' można wrócić do M?

zadanie do pomyślenia w domu (nieobowiązkowe)

Uwaga: na następnej stronie jest rozwiązanie zadania domowego 1^D z poprzednich ćwiczeń

1°. Rozważ cykl skierowany G o czterech wierzchołkach i czterech krawędziach etykietowanych kolejno literami a, b, c, d. Skonstruuj sieć o mniej niż czterech miejscach, której graf konfiguracji jest izomorficzny z G.



* pomysł: pracujemy z sieciami elementarnymi, więc jedną z konfiguracji może być konfiguracja pełna (po żetonie na każdym miejscu); jedyną aktywną tranzycją może być wtedy taka bez miejsc wyjściowych

* przykładowa sieć:

odpala cyklicznie

t_1, t_2, t_3, t_4