# Problemy Decyzyjne dla Systemów Nieskończonych

Ćwiczenia 4

9 marca 2012

Na tych ćwiczeniach kończyliśmy temat automatów czasowych, zdefiniowaliśmy sieci Petriego i zaczęliśmy się przyglądać zastosowaniom WQO. O tym ostatnim już napiszę w następnych notatkach.

**1.** Udowodnij, że niepustość jest nierozstrzygalna dla aumatów czasowych z nieograniczoną możliwością stosowania operacji *plus* w dozorach.

**Wskazówki**

- zredukuj problem osiągalności dla maszyn dwulicznikowych
- taka maszyna ma 2 typy instrukcji:
  - $l : \text{inc(c)}, \text{goto } l'$
  - $l : \text{if } (c == 0) \text{ goto } l', \text{else dec(c)}, \text{ goto } l''$
- symulujemy ją za pomocą automatu czasowego, dostaniemy rezultat: maszyna dwulinicznikowa ma skończony bieg (czyli kończy w stanie $q_{halt}$) wtedy i tylko wtedy, gdy automat ma bieg skończony, czyli jego język jest niepusty
- wszystko jest dosyć prosto symulować, jedyną kwestią trudniejszą jest zwiększanie i zmniejszanie liczników
- robimy to następująco: konfiguracji maszyny o wartości licznika $c$ będzie odpowiadać konfiguracja automatu czasowego o różnicy wartości zegarów $x_c - x'_c$ równej $c$
- zwiększamy licznik używając pomocniczych zegarów $y$, $y'$ i $z$ w kolejnych stanach robiąc: $z := 0, y' := (x_c - x'_c)$, $z == 1, y := y', y' := 0$: teraz $y - y' = x_c - x'_c + 1$, następnie przepisujemy wykonujemy podobną procedurę powodując, że $x_c - x'_c = y - y'$
- zmniejszamy licznik podobnie, zerujemy łatwo

**2.** Przyjrzyj się sytuacjom, gdy dozwolone są przypisania typu: $x := x + 1$ lub $x := x - 1$.

**Wskazówki**

- rozważmy najpierw sytuację $x := x + 1$

- przyglądając się konstrukcji regionowej zauważamy, że wszystko działa
- teraz $x := x - 1$
- konstrukcja regionowa nie działa, bo trzeba byłoby dla dowolnie dużych wartości zegara $x$ pamiętać na przykład jego część ułamkową (by po wielu zmniejszeniach wartości móc ją znać)
- okazuje się, że ten przypadek ma niepustość nierozstrzygalną - ale tego nie dowodzimy na ćwiczeniach

**Uwaga 1.** *Zastanawialiśmy się co tak naprawdę powoduje, że liczba regionów w konstrukcji regionowej jest wykładnicza. Jedną sprawą jest, że dla n zegarów uporządkowań części ułamkowej jest $n!$, czyli to powoduje wykładniczość. Jednak problem jest PSPACE-trudny już dla 3 zegarów. Drugim powodem wykładniczości jest to, że stałe w dozorach reprentujemy binarnie, czyli wartość stałej jest wykładnicza w stosunku do jej reprezentacji. Pozostaje pytanie co dzieje się dla 1 lub 2 zegarów.*

**3.** Zaprojektuj jak najszybszy algorytm rozwiązujący problem niepustości dla automatów czasowych z jednym zegarem.

**Szkic rozwiązania** Na ćwiczeniach skupiliśmy się następującym pomyśle: że tak naprawdę dla każdego stanu istotna jest najmniejsza wartość zegara dla której dla się dotrzeć do tego stanu. Skonstruowaliśmy przeróbkę algorytmu Dijkstry, która realizuje to obliczenie. Będzie ona działać w czasie wielomianowym, gdyż może wykonać jedynie wielomianowo wiele faz poprawiania. Istotnie: każdy zegar może tylko wielomianowo wiele razy zmniejszać się w trakcie obliczenia jego minimalnej wartości, bo może przyjmować tylko wartości dane jako stałe w dozorach.

Jest też alternatywne rozwiązanie: mianowicie możemy jawnie zmodyfikować konstrukcję regionową. Niech stałe z dozorów to: $0 < C_1 < C_2 < \ldots C_k = C$. Normalnie rozważamy m.in. regiony $(i, i+1)$ dla $0 \leqslant i \leqslant C-1$, których jest wykładniczo wiele. Okazuje się, że możemy rozważać regiony $(C_i, C_{i+1}$ dla $1 \leqslant i \leqslant k-1$, których jest liniowo wiele plus regiony punktowe i skrajne regiony nieograniczone - czyli w sumie wielomianowo wiele regionów.

**4.** Rozstrzygnij dlaczego nie można tej konstrukcji (drugiej możliwości) powtórzyć dla więcej niż 1 zegara.

**Szkic rozwiązania** Spójrzmy na automat 2-zegarowy. Okazuje się, że dla każdego regionu punktowego $(x, y)$ trzeba również pamiętać region (lub zbiór regionów) postaci $(x-z, y-z)$ dla $z > 0$. To powoduje, że nawet dla stałych $C_1 = 1$, $C_2 = 1000000$ mamy dużo regionów, rzędu $1000000^2$.

**5.** Pokaż, że problem niepustości dla automatów czasowych z 2 zegarami jest NP-trudny.

**Wskazówki**

- zredukuj problem SUBSET SUM, czyli - dane: liczby $k_1, \ldots, k_n, s$, pytanie: czy istnieje taki podzbiór indeksów $P \subseteq \{1, \ldots n\}$, że $\sum_{i \in P} k_i = s$. Wiadomo, że ten problem jest NP-trudny (także NP-zupełny)
- czyli dla danych liczb $k_1, \ldots, k_n, s$ zbuduj automat czasowy z maksymalnie 2 zegarami taki, że jego język jest niepusty wtedy i tylko wtedy, gdy istnieje odpowiedni podzbiór $P$
- automat ma 2 zegary: $g$ - czas globalny i $l$ - czas lokalny. Zaczyna z oboma ustawionymi na 0. Następnie dla każdej z liczb $k_i$ wykonuje jedną z dwóch operacji: albo przechodzi do następnego stanu z warunkiem $l == 0$ albo z warunkiem $l == k_i, l := 0$. To odpowiada dodaniu do globalnego czasu 0 lub $k_i$, czyli pójście pierwszą opcją odpowiada nie wzięciu tego indeksu do $P$, a drugą wzięciu. Na końcu automat przechodzi do stanu akceptującego wtedy i tylko wtedy, gdy zegar $g = s$, czyli gdy czas globalny zsumował się do szukanej sumy $s$.

*Sieć Petriego* można zdefiniować na 2 sposoby: jeden żetonowy, drugi jako wektory z dodawaniem. Te sposoby są właściwie równoważne, my tu podamy wektorowy zwany Vector Addition System (VAS). VAS składa się ze skończonego zbioru wektorów $S = \{v_1, \ldots, v_n\} \subseteq \mathbb{Z}^k$ (czyli współrzędne mogą być ujemne). Przejścia VAS są postaci $v \mapsto v + v_i$, dla pewnego $1 \leqslant i \leqslant k$, gdzie zarówno $v$ jak i $v + v_i$ należą do $\mathbb{N}^k$ (czyli współrzędne są dodatnie).

Na przyszłych ćwiczeniach będziemy się m.in. zajmować sieciami Petriego.