Teoria Współbieżności
Ćwiczenie 6

1 Cel ćwiczenia

Celem ćwiczenia jest zapoznanie studentów z zastosowaniem teorii śladów do szeregowania wątków. Kolejnym celem ćwiczenia jest zaprezentowanie studentów z zastosowaniem modelu gramatyk grafowych do reprezentacji wątków współbieżnych, gdzie wykonanie produkcji w gramatyce grafowej oznacza wykonania wątku. Jako przykład rozważony jest problem generacji jednowymiarowego ciągu elementów z pomocą gramatyki grafowej. Kolejnym celem ćwiczenia jest zapoznanie studentów z technikami implementowaniem wątków w Javie. W szczególności studenci zapoznani zostaną z implementacją schedulera szeregującego wątki Javy zgodnie z klasami Foaty uzyskanymi na podstawie analizy z użyciem teorii śladów.

2 Wprowadzenie teoretyczne

Teoria śladów [1, 2] służy głównie do modelowania zachowania systemów współbieżnych. Opisuje zależności pomiędzy elementami predefiniowanego zbioru niepodzielnych zadań obliczeniowych. Podstawowym pojęciem teorii śladów jest alfabet będący skończonym zbiorem niepodzielnych zadań obliczeniowych (tasków) - to znaczy niepodzielnych i składających się jedynie z instrukcji wykonywanych sekwencyjnie. Teoria śladów jest szeroko stosowanym formalizmem pozwalającym wyprowadzić optymalne zrównoleglenie dla zadanej grupy tasków.

W trakcie ćwiczenia studenci:

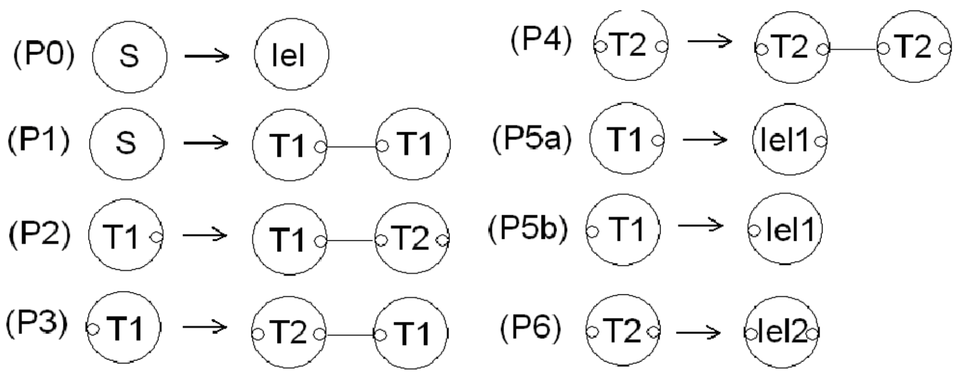
- Definiują alfabet w sensie teorii śladów dla zadanego problemu obliczeniowego
- Definiują relacje pomiędzy elementami alfabetu

- Obliczają postać normalną Foaty oraz budują graf Diekerta
- Wyprowadzają formalny model równoległości zadanego problemu obliczeniowego
- Implementują wyprowadzony wyżej model równoległy

3 Ćwiczenia

3.1 Generacja siatki 1D

Dla danego zbióru produkcji w gramatyce grafowej, odpowiedzialnego za generacje jednowymiarowych ciągu elementów.



oraz przykładowego wywodu w gramatyce (gdzie kolorem czerwonym zaznaczono miejsce zastosowania kolejnej produkcji)

$$(P1)\text{–}(P2)\text{–}(P3)\text{–}(P5a)\text{–}(P6)\text{–}(P5b)\text{–}(P6) \tag{1}$$

$$\begin{gathered} S_{(P1)} \to T1_{(P2)}\text{–}T1 \to T1\text{–}T2\text{–}T1_{(P3)} \to T1_{(P5a)}\text{–}T2\text{–}T2\text{–}T1 \to \\ \to |e|_1\text{–}T2_{(P6)}\text{–}T2\text{–}T1 \to |e|_1\text{–}|e|_2\text{–}T2\text{–}T1_{(P5b)} \to |e|_1\text{–}|e|_2\text{–}T2_{(P6)}\text{–}|e|_1 \to \\ \to |e|_1\text{–}|e|_2\text{–}|e|_2\text{–}|e|_1 \end{gathered} \tag{2}$$

opracowujemy alfabet w sensie teorii śladów reprezentujący wykonanie poszczególnych tasków – produkcji

$$A = \{P1, P2, P3, P5a, P6, P5b, P6\} \tag{3}$$

Następnie na podstawie analizy zależności pomiędzy produkcjami opracowywana jest relacja zależności (plus symetria)

$$D = \text{sym}\{\{(P1, P2), (P1, P3), (P2, P5a), (P2, P6), (P3, P5b), (P3, P6)\}^+\} \cup I_A \tag{4}$$

Dla słowo w alfabecie reprezentującego wywód w gramatyce

$$P1, P2, P3, P5a, P6, P5b, P6 \tag{5}$$

opracowywana jest postać normalna Foaty

$$FNF = [P1][P2, P3][P5a, P6, P5b, P6] \tag{6}$$

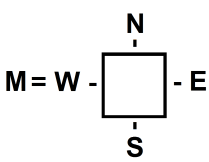
Bazując na opracowanej postaci normalnej Foaty oraz przedstawionym zbiorze produkcji tworzony jest program w języku JAVA generujące wątki odpowiadające produkcjom w gramatyce grafowej oraz szeregujący je zgodnie z klasami Foaty. Pełny kod programu znajduje się w repozytorium. https://github.com/macwozni/1DMeshParallel
Po omówieniu wszystkich elementów, zadawany jest inne wywód w gramatyce (inne słowo w sensie alfabetu teorii śladów) i studenci proszeni są o wygenerowanie nowych klas Foaty, oraz napisanie nowego mechanizmu szeregowania.

3.1.1 Zadanie

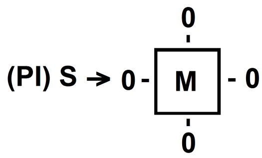
1. Zdefiniuj alfabet w sensie teorii śladów dla nowego dowolnego wywodu.
2. Zdefiniuj relację zależności D.
3. Oblicz klasy Foaty.
4. Zaimplementuj scheduler dla nowych klas Foaty (można wykorzystać załączony kod) i pokaż, że daje poprawny wynik.

3.2 Generacja siatki 2D

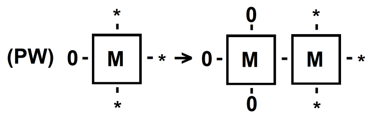
Dany jest element czworokątny



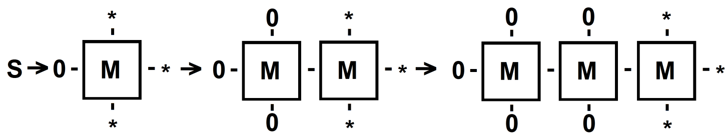
Etykiety N, S, W, E oznaczają odpowiednie kierunki sąsiedztwa: North, South, West, East. Dana jest następująca produkcja startow – generujące jeden element M (0 oznacza brak sąsiada w danym kierunku).



Następnie dodajemy produkcje generującą sąsiada z lewej strony ($*$ oznacza że tutaj może być sąsiad lub nic).



Przykładowy wywód: $PI \to PW \to PW$



3.2.1 Zadanie

Ćwiczenie polega na wykonaniu następujących kroków:

1. Proszę rozszerzyć gramatykę w taki sposób, aby była możliwa generacja siatek prostokątnych, dwuwymiarowych, o ilości elementów $N \times M$
2. Proszę napisać ciąg produkcji w gramatyce generujący siatkę prostokątną o 3×3 elementach
3. Bazując na ciągu produkcji w gramatyce generującej przedstawioną siatkę, proszę wskazać alfabet w sensie teorii śladów
4. Proszę napisać słowo (ciąg symboli z alfabetu) odpowiadających generacji siatki prostokątnej
5. Proszę wskazać relacje (nie)zależności dla alfabetu, w sensie teorii śladów

6. Proszę przekształcić ciąg symboli (słowo) do postaci normalnej Foaty
7. Proszę zaprojektować i zaimplementować algorytm współbieżny w oparciu o postać normalną Foaty. Parametr algorytmu to N = ilość kwadratów na każdym boku siatki

Literatura

[1] V. Diekert, Y. M'etivier - Partial commutation and traces, [w:] Handbook of Formal Languages, Springer, 1997, str. 457-553

[2] Diekert V., Rozemberg G. - The book of traces, 1995

[3] Wykład z przedmiotu "Teoria współbieżności" – rozdział dotyczący teorii śladów

[4] Bruce Eckel, "Thinking in Java" - rozdział o wątkach

Uwagi: sluzalec@agh.edu.pl