

Algorytmy Równoległe i Rozproszone

Część VIII - Systemy rozproszone, c. d.

Łukasz Kuszner
pokój 209, WETI

<http://www.kaims.pl/~kuszner/>
kuszner@kaims.pl

Oficjalna strona wykładu
<http://www.kaims.pl/~kuszner/ARiR/>
Wykład 15 godzin, Projekt 15 godzin

2011

Demony

Łamanie symetrii

Bizantyjscy . . .

Strona główna

Strona tytułowa

⏪ ⏩

◀ ▶

Strona 1 z 11

Powrót

Full Screen

Zamknij

Koniec

1. Systemy z demonem

Demon (ang. daemon, scheduler, adversary) „steruje systemem” wskazując wierzchołki do wykonania, kolejnego kroku obliczeń.

- Demon centralny
- Demon rozproszony
- R/W atomicity

[Strona główna](#)[Strona tytułowa](#)[◀](#) [▶](#)[◀](#) [▶](#)

Strona 3 z 11

[Powrót](#)[Full Screen](#)[Zamknij](#)[Koniec](#)

Algorytmy w modelu ze zmiennymi współdzielonymi i demonem

Niech p będzie warunkiem, a M akcją. Algorytm dla każdego wierzchołka u jest dany za pomocą reguł postaci:

R: **if** $p(u)$
 then M ,

Warunek p może dotyczyć stanu wierzchołka u oraz stanów wierzchołków z sąsiedztwa wierzchołka u , $N(u) = \{v \mid \{v, u\} \in E\}$. Jeśli warunek p dla wierzchołka u jest spełniony, to mówimy, że wierzchołek jest *aktywny*.

Demon centralny

Demon centralny spośród wierzchołków aktywnych wybiera jeden, który jako kolejny wykona ruch, w ten sposób żadne dwa ruchy nie są wykonywane w tym samym czasie.

Demony

Łamanie symetrii

Bizantyjscy . . .

Strona główna

Strona tytułowa

◀◀ ▶▶

◀ ▶

Strona 4 z 11

Powrót

Full Screen

Zamknij

Koniec

Demon rozproszony

Demon rozproszony spośród wierzchołków aktywnych wybiera dowolny podzbiór tych, które jako kolejne wykonają ruch.

Demony

Łamanie symetrii

Bizantyjscy . . .

Strona główna

Strona tytułowa

◀◀ ▶▶

◀ ▶

Strona 5 z 11

Powrót

Full Screen

Zamknij

Koniec

R/W atomicity

Przy konstruowaniu reguł pozwalaliśmy odczytać stany lokalne wszystkich wierzchołków w jednym kroku. Tu wprowadzamy ograniczenie do stanu tylko jednego sąsiada na ruch.

Demony

Łamanie symetrii

Bizantyjscy...

Strona główna

Strona tytułowa

◀◀ ▶▶

◀ ▶

Strona 6 z 11

Powrót

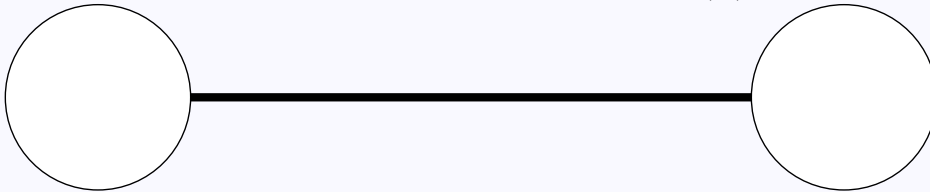
Full Screen

Zamknij

Koniec

2. Problem łamania symetrii

Rozpatrzmy graf jak na rysunku, w którym każdy wierzchołek v ma jedną zmienną lokalną f , inicjalnie $f(v) = 0$.



Algorytm 1: color

R1: if

$$\exists_{u \in N(v)} f(u) = f(v)$$

then $f(v) += 1$

[Strona główna](#)[Strona tytułowa](#)[◀](#) [▶](#)[◀](#) [▶](#)[Strona 7 z 11](#)[Powrót](#)[Full Screen](#)[Zamknij](#)[Koniec](#)

Ćwiczenie 1

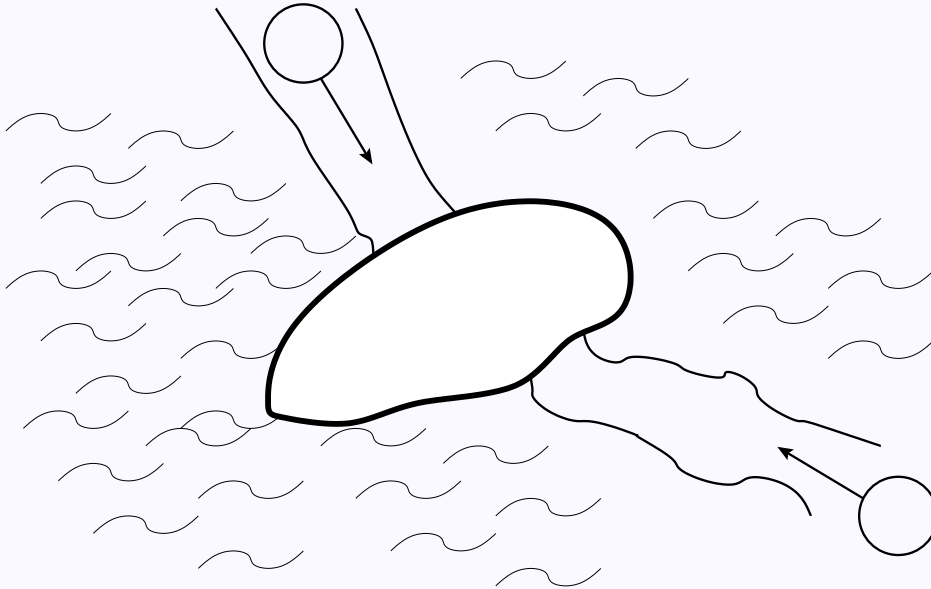
Porównaj działanie algorytmu w modelu z demonem centralnym i rozproszonym.

Ćwiczenie 2

Napisz odpowiedni algorytm dla modelu z przesyłaniem komunikatów.

3. Problem Bizantyjskich generałów

(ang. coordinated attack problem)

[Strona główna](#)[Strona tytułowa](#)[◀](#) [▶](#)[◀](#) [▶](#)[Strona 9 z 11](#)[Powrót](#)[Full Screen](#)[Zamknij](#)[Koniec](#)

[Strona główna](#)[Strona tytułowa](#)[◀](#) [▶](#)[◀](#) [▶](#)

Strona 10 z 11

[Powrót](#)[Full Screen](#)[Zamknij](#)[Koniec](#)

Dla dwóch procesorów p_i , jednobitowego wejścia: x_i oraz jednobitowego wyjścia y_i , gdzie $i = 1, 2$; nie istnieje algorytm, który komunikując się przez zawodne medium (wiadomość może być dostarczona w całości poprawnie, lub wcale) i spełnia trzy warunki

- Zgodność: $y_1 = y_2$;
- Poprawność: jeśli $x_1 = x_2 = 0$, to $y_1 = y_2 = 0$;
- Nietrywialność: istnieje wykonanie, dla którego $y_1 = y_2 = 1$;

Dowód nie wprost. Przypuśćmy, że taki algorytm istnieje. Rozpatrzmy wykonania, które prowadzą do zgodnej odpowiedzi $y_1 = y_2 = 1$ (nietrywialność). Spośród nich wybierzmy, te w której odebrano najmniej wiadomości. Odebrano co najmniej 1 wiadomość (poprawność i zgodność). Bez utraty ogólności możemy przyjąć, że ostatnią wiadomość otrzymał p_1 . Rozpatrzmy wykonanie, w którym ostatnia wiadomość zaginęła. Wtedy również $y_1 = y_2 = 1$ (zgodność). Sprzeczność.

Demony

Łamanie symetrii

Bizantyjscy . . .

Strona główna

Strona tytułowa

◀ ▶

◀ ▶

Strona 11 z 11

Powrót

Full Screen

Zamknij

Koniec