

# Algorytmy Równoległe i Rozproszone

## Część I - Wprowadzenie i Sieci porównujące

Łukasz Kuszner  
pokój 209, WETI  
<http://www.sphere.pl/~kuszner/>  
kuszner@sphere.pl

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

2007/08

### Spis treści

<b>1</b>	<b>Plan wykładu</b>	<b>1</b>
<b>2</b>	<b>Literatura</b>	<b>2</b>
<b>3</b>	<b>Sieci porównujące</b>	<b>3</b>
3.1	Schematy komparatora . . . . .	3
<b>4</b>	<b>PRAM - przykład</b>	<b>4</b>
4.1	Iloczyn skalarny . . . . .	4
<b>5</b>	<b>Model rozproszony</b>	<b>5</b>
5.1	Rozproszony algorytm dla problemu: „wybór lidera” . . . . .	5

## 1 Ogólny plan wykładu

- Wstęp
- Sieci porównujące,
- Układy kombinacyjne,
- Algorytmy w modelu PRAM,
- Algorytmy w modelu rozproszonym.

Notatki

## 2 Literatura

- T. H. Cormen, C. E. Leiserson and R. L. Rivest, „Introduction to Algorithms”, The MIT Press/McGraw-Hill Company, 1990 (wydanie polskie WNT).
- Raymond Greenlaw, H. James Hoover, Walter L. Ruzzo „Limits to Parallel Computation: P-Completeness Theory”, Oxford University Press, 1998.
- Hagit Attiya „Lecture Notes for Course Distributed Algorithms”, 1994.

Notatki

- C. Xavier, S. S. Iyengar, „Introduction to Parallel Algorithms”, Wiley-IEEE, 1998.
- Gerard Tel, „Introduction to Distributed Algorithms”, Cambridge University Press, 2nd edition, 2000.
- Shlomi Dolev. Self-Stabilization. MIT Press, 2000.

Notatki

- J. Jaja, „An Introduction to Parallel Algorithms”, Addison-Wesley, Reading, MA, 1992. ■
- S.G. Akl, „The Design and Analysis of Parallel Algorithms”, Prentice-Hall, 1989. ■
- Lecture Notes Repository: Uniwersytet w Paderborn.

Notatki

- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar „Introduction to Parallel Computing”, Addison Wesley, 2003. ■
- Hagit Attiya, Jennifer Welch „Distributed Computing: Fundamentals, Simulations, and Advanced Topics”, 2nd Edition. ■
- Guy E. Blelloch, Bruce M. Maggs: Parallel Algorithms. The Computer Science and Engineering Handbook, 1997: 277-315.

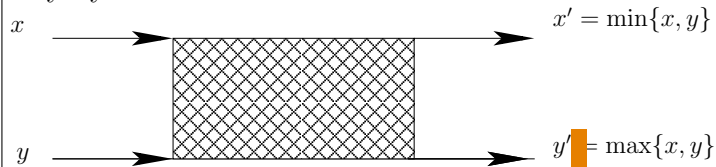
Notatki

### 3 Sieci porównujące - przykład

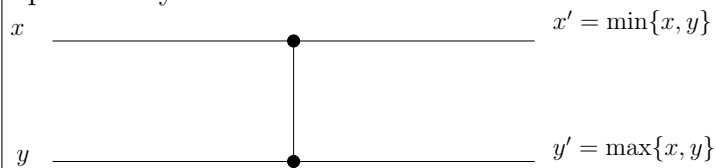
Sieć porównująca składa się z połączonych ze sobą komparatorów.

#### 3.1 Schematy komparatora

zwykły:

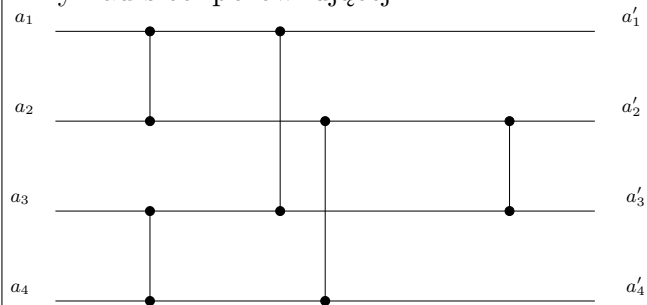


uproszczony:



Notatki

#### Przykład sieci porównującej



Notatki

#### Ćwiczenie 1

Sprawdź, że sieć porównująca z poprzedniego slajdu jest siecią sortującą.

Notatki

## 4 PRAM - przykład

### 4.1 Iloczyn skalarny

We: Tablice współrzędnych  $a[1 : n]$  i  $b[1 : n]$

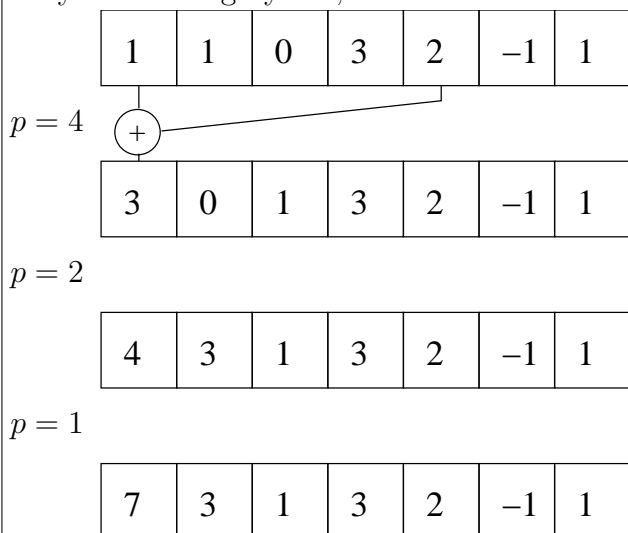
Wy: Liczba będąca iloczynem skalarnym wektorów  $a$  i  $b$ .

**Algorytm 1:** Iloczyn skalarny

```
1: for  $i = 1$  to  $n$  in parallel do
2:    $c_i = a_i * b_i$ 
3: end for
4:  $p = n/2$ 
5: while  $p > 0$  do
6:   for  $i = 1$  to  $p$  in parallel do
7:      $c_i = c_i + c_{i+p}$ 
8:   end for
9:    $p = p/2$ 
10: end while
```

Notatki

Przykład dla algorytmu, dla  $n = 7$ .



Notatki

## 5 Model rozproszony - przykład

### 5.1 Rozproszony algorytm dla problemu: „wybór lidera”

Przypuśćmy, że mamy rozproszony system asynchroniczny, w którym jednostki obliczeniowe tworzą pierścień.

Problem polega na skonstruowaniu algorytmu, który pozwoli wyróżnić jeden z procesorów. Oczekujemy, że po zakończeniu działania algorytmu dokładnie jeden procesor zasygnalizuje „jestem liderem”, a wszystkie pozostałe „nie jestem liderem”.

Notatki

Przypuśćmy, że każdy z procesorów ma unikalny identyfikator. Procesory tworzą pierścień, możemy więc mówić o lewym i prawym sąsiedzie. Inicjalnie każdy z procesorów wysyła wiadomość do swojego lewego sąsiada, w której zapisuje swój identyfikator.

Następnie każdy procesor odbierając wiadomość od prawego sąsiada przesyła ją dalej, jeśli zapisany na niej identyfikator jest większy od jego własnego identyfikatora i kasuje, jeśli jest mniejszy.

Procesor, który otrzymał z powrotem swój własny identyfikator ogłasza się liderem i wysyła wiadomość kończącą działanie algorytmu.

Notatki

### Ćwiczenie 2

Uzasadnij, że liczba wiadomości wysłanych w trakcie działania algorytmu jest  $O(n^2)$ .

Notatki