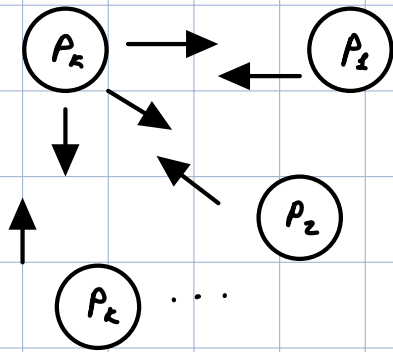


Лекция 10. Параллельный обход графа с расширяющейся передачей  
(в broadcast) от одного ко всем и сужающейся передаче  
(convergecast) от всех к одному



Алгоритм наводнения

$first = 1$

$\langle p_k, p_i \rangle ? GO(data) \{$

(1)  $if(first) \{$

(2)  $first := 0$

(3)  $foreach\ j \in neighbors \setminus \{p_k\} \{$

$\langle p_i, p_j \rangle ! GO(data)$

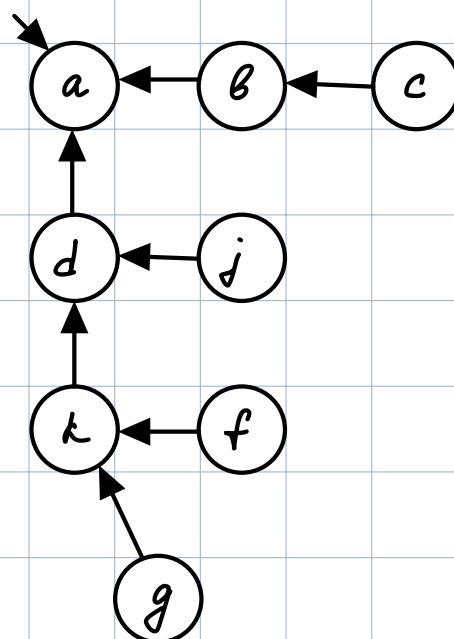
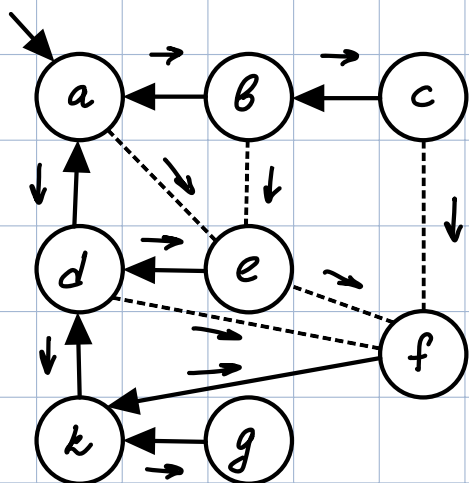
$\}$

(4)  $\}$

$\}$

Расширяющаяся / сужающаяся передача

С оставшим деревом и выделенным узлом (корень)



Если в дереве  $n$  узлов, то  $(n-1)$  дуг

$P_c$  - корень дерева - выделенный узел ( $parent_a = P_a$ )

$parent_i$  - родитель узла  $p_i$

$children_i$  - дети узла  $p_i$

### Алгоритм 3.2

а) распространение информации

$\langle p_k, p_i \rangle ? GO(data) \{$

(1)  $\text{foreach } j \in children_i \setminus \{k\} \{$

$\langle p_i, p_j \rangle ! GO(data)$

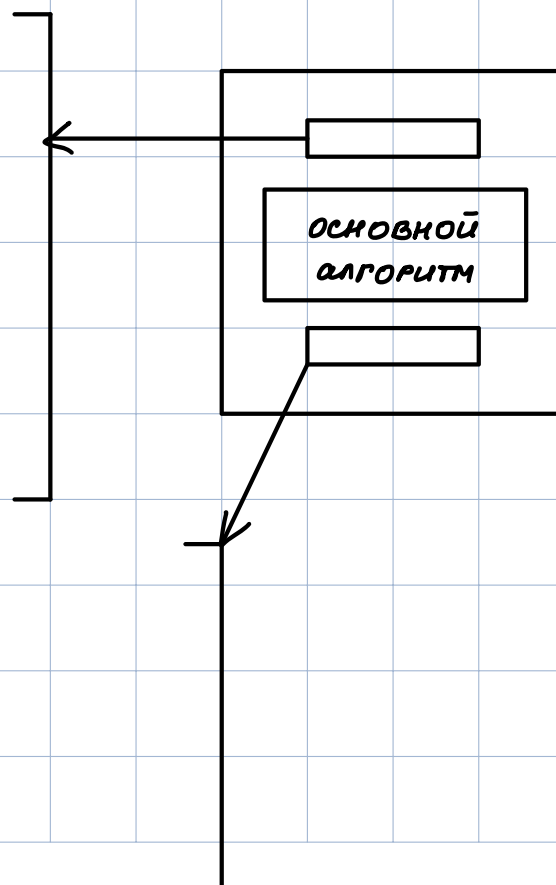
$\}$

$\}$

б) сбор информации

$\text{foreach } j \in children_i \{$

(2)  $\langle p_j, p_i \rangle ? BACK(val\_set_i)$



```

(3)   val_seti := (  $\bigcup_{j \in \text{children}_i}$  val_setj )  $\cup$  { < i, vi > }
(4)   k := parent
(5)   if (k  $\neq$  i) {
        < pi, pk > !BACK (val_seti) }
(6)   else {
        f(val_seti)
    }
    }

```

## Построение остоного дерева

### Структура данных

- neighbors<sub>i</sub> — множество соседей p<sub>i</sub> — отражает позицию p<sub>i</sub> в графе
  - parent<sub>i</sub> —
  - children<sub>i</sub> —
- } позиция p<sub>i</sub> в остоном дереве

Изначально parent<sub>i</sub> = 1, children<sub>i</sub> =  $\emptyset$

- expected\_msg<sub>i</sub> — ожидаемое количество сообщений
- FIFO каналы

## Алгоритм 33. Код для p<sub>i</sub>

< -, p<sub>a</sub> > ? START { // здесь считаем, что START получает только один процесс p<sub>a</sub>

(1) parent<sub>i</sub> := i

$children_i := \emptyset$

$expected\_msg_i := |neighbors_i|$

(2)  $foreach\ j \in neighbors_i\{$   
     $\langle p_i, p_j \rangle ! GO(data)$   
     $\}$   
 $\}$

$\langle p_j, p_i \rangle ? GO(data)\{$

(3)  $if\ (parent := 1)\{$

(4)  $parent_i := j$

$children_i := \emptyset$

$expected\_msg_i := |neighbors_i| - 1$

(5)  $if\ (expected\_msg_i = 0)\{$

(6)  $\langle p_i, p_j \rangle ! BACK(\{i, v_i\})$

(7)  $\}\ else\{$

$foreach\ k \in neighbors_i \setminus \{j\}\{$

$\langle p_i, p_k \rangle ! GO(data)\}$

(8)  $\}$

(9)  $\}\ else\{$

$\langle p_i, p_j \rangle ! BACK(\emptyset)$

(10)  $\}$

$\}$

$GO_{x,y} - BACK_{x,y}$

```

< pj, pi > ? BACK (val-set) {
(11)   expected_msgi := expected_msgi - 1
(12)   if (val-set ≠ ∅) {
        childreni := childreni ∪ { pj }
      }
(13)   if (expected_msg := 0) {
(14)     val-set = ∪x ∈ childreni val-setx ∪ { < i, vi > }
        pr := parenti;
(15)     if (pr ≠ i) {
          < pi, pr > ! BACK (val-set)
(16)     } else {
          f (val-set)
(17)     }
(18)   }
}

```

## Гоиимость

$C$  - количество каналов графа коммуникации

$2(n-1)$  - количество сообщений переданных остовому дереву

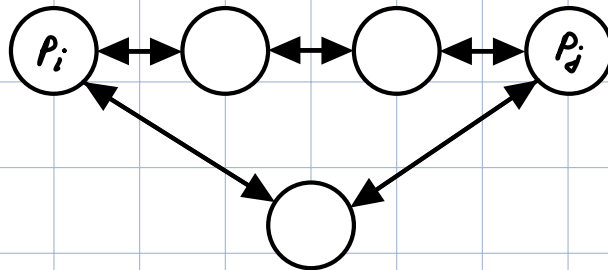
$4(e - (n-1))$  - общее количество сообщений по другим каналам

Итого  $2(2e - (n-1))$  - общее количество сообщений

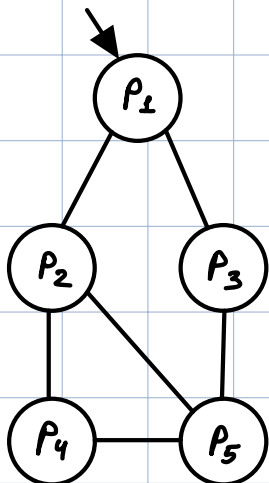
После GO и BACK будет  $2(n-1)$

Временная плотность -  $O(2)$

Оставное дерево с обходом в ширину

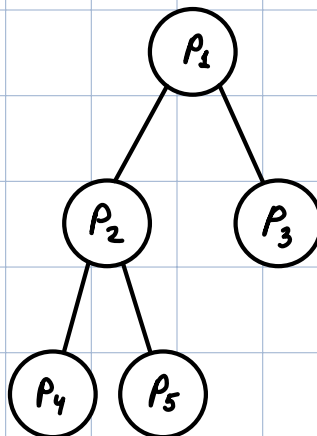


а) Граф коммуникации

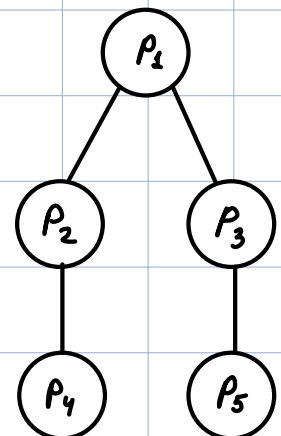


$P_1$	$P_2$	-	1
$P_1$	$P_3$	-	1
$P_1$	$P_4$	-	2
$P_1$	$P_5$	-	2

б) Оставное дерево



в) Оставное дерево



## Основное дерево в ширину без централизованного управления

Y. Cheung 1983

### Структура данных (у процесса $P_i$ )

$parent_i;$   
 $children_i;$   
 $expected\_msg_i;$  } подобно предыдущему

$level_i$  - для текущего расстояния до корня

$GO(d)$  - текущий уровень отправляемых сообщений

### Цель

$P_i$  получает сообщение  $GO(d)$

- $GO(d)$  - первое сообщение  $GO$ , то :

- $P_i$   $level_i := d + 1$
- устанавливает положение в основном дереве
- отправляет сообщение соседям

- $GO(d)$  - непервое сообщение  $GO$ , то :

- if  $level_i > d + 1$  :

a)  $level_i := d + 1$

b) меняем родителя

c)  $GO(d)$  - распространяем соседям

- иначе  $BACK(no)$

Ц / 3

1. Как изменится работа алгоритма, if каналы будут не FIFO?
2. Модифицировать алгоритм так, чтобы было несколько выделенных узлов, строящих свои остовные деревья.
3. Модифицировать алгоритм так, чтобы выделенных узлов было несколько, но строили 1 остовное дерево.
4. Построить 2 вычисления на каждый алгоритм