

## 4. Algoritmus RETE

---

# Algoritmus RETE

24. října 2023

## 4. Algoritmus RETE

---

# Motivace použití algoritmu RETE

Předpoklady:

- při modifikaci pracovní paměti se obvykle mění jen malá část jejího obsahu
- více pravidel obvykle sdílí stejné elementární podmínky

## 4. Algoritmus RETE

---

### Postup použití algoritmu

1. **Inicializace** (jednorázová akce)
  - a. Vytvoření RETE sítě
  - b. Nastavení počátečních podmínek
2. **Inference** (opakované vykonávání algoritmu)
  - a. Provedení aplikovatelného pravidla
  - b. Šíření nového faktu RETE sítí

## 4. Algoritmus RETE

---

### Příklad

- Báze znalostí:

$A(x) \& B(x) \& C(y) \Rightarrow$  přidej  $D(x)$

$A(x) \& B(y) \& D(x) \Rightarrow$  přidej  $E(x)$

$A(x) \& B(x) \& E(x) \Rightarrow$  zruš  $A(x)$

- Pracovní paměť:

$A(1), A(2), B(2), B(3), B(4), C(5)$

## 4. Algoritmus RETE

---

### Tvorba sítě

- Báze znalostí:

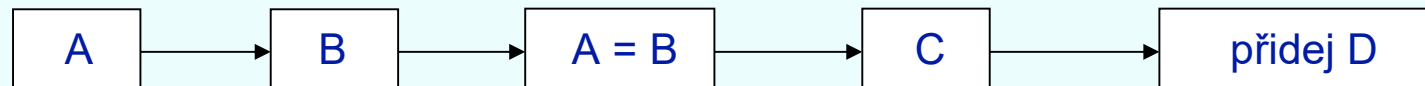
$A(x) \& B(x) \& C(y) \Rightarrow$  přidej  $D(x)$

$A(x) \& B(y) \& D(x) \Rightarrow$  přidej  $E(x)$

$A(x) \& B(x) \& E(x) \Rightarrow$  zruš  $A(x)$

- Pracovní paměť:

$A(1), A(2), B(2), B(3), B(4), C(5)$



## 4. Algoritmus RETE

### Tvorba sítě

- Báze znalostí:

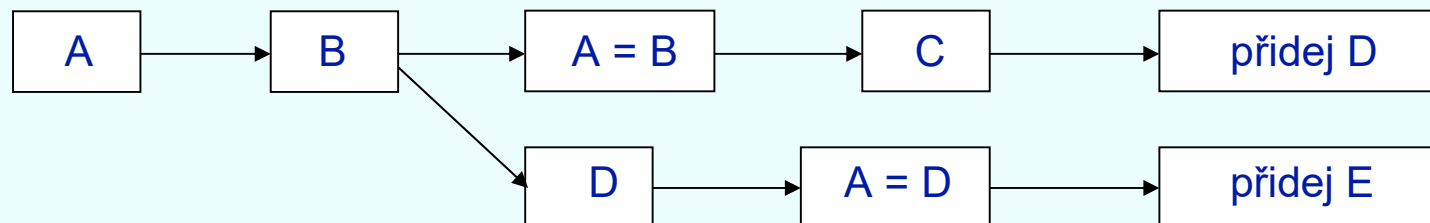
$A(x) \& B(x) \& C(y) \Rightarrow$  přidej  $D(x)$

$A(x) \& B(y) \& D(x) \Rightarrow$  přidej  $E(x)$

$A(x) \& B(x) \& E(x) \Rightarrow$  zruš  $A(x)$

- Pracovní paměť:

$A(1), A(2), B(2), B(3), B(4), C(5)$



## 4. Algoritmus RETE

### Tvorba sítě

- Báze znalostí:

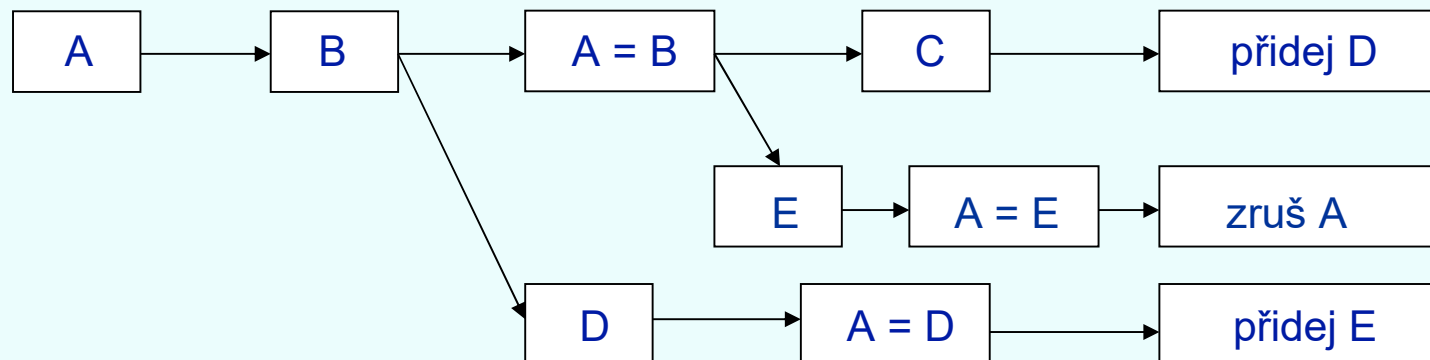
$A(x) \& B(x) \& C(y) \Rightarrow$  přidej  $D(x)$

$A(x) \& B(y) \& D(x) \Rightarrow$  přidej  $E(x)$

$A(x) \& B(x) \& E(x) \Rightarrow$  zruš  $A(x)$

- Pracovní paměť:

$A(1), A(2), B(2), B(3), B(4), C(5)$



## 4. Algoritmus RETE

### Nastavení počátečních podmínek

- Báze znalostí:

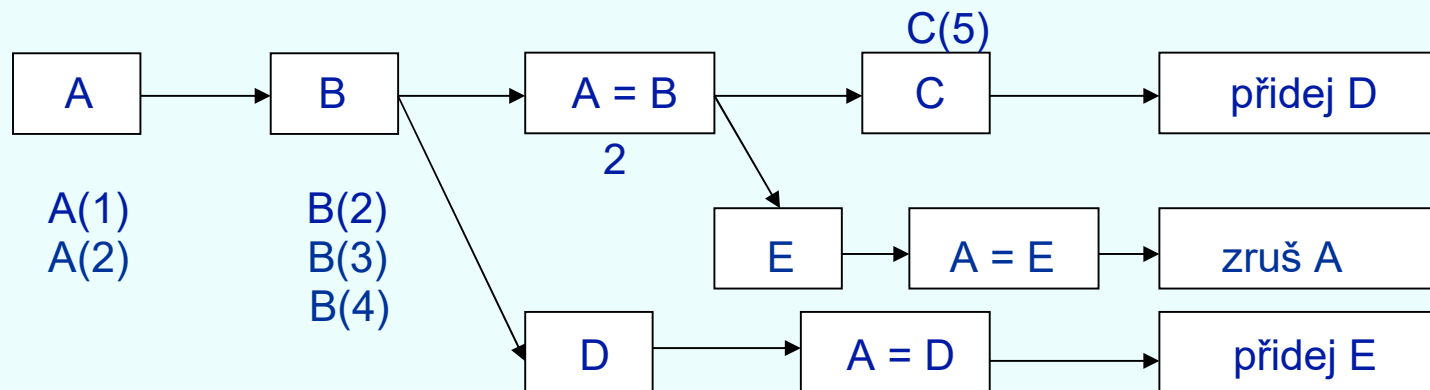
$A(x) \& B(x) \& C(y) \Rightarrow$  přidej  $D(x)$

$A(x) \& B(y) \& D(x) \Rightarrow$  přidej  $E(x)$

$A(x) \& B(x) \& E(x) \Rightarrow$  zruš  $A(x)$

- Pracovní paměť:

$A(1), A(2), B(2), B(3), B(4), C(5)$





## 4. Algoritmus RETE

### Provedení pravidla

- Báze znalostí:

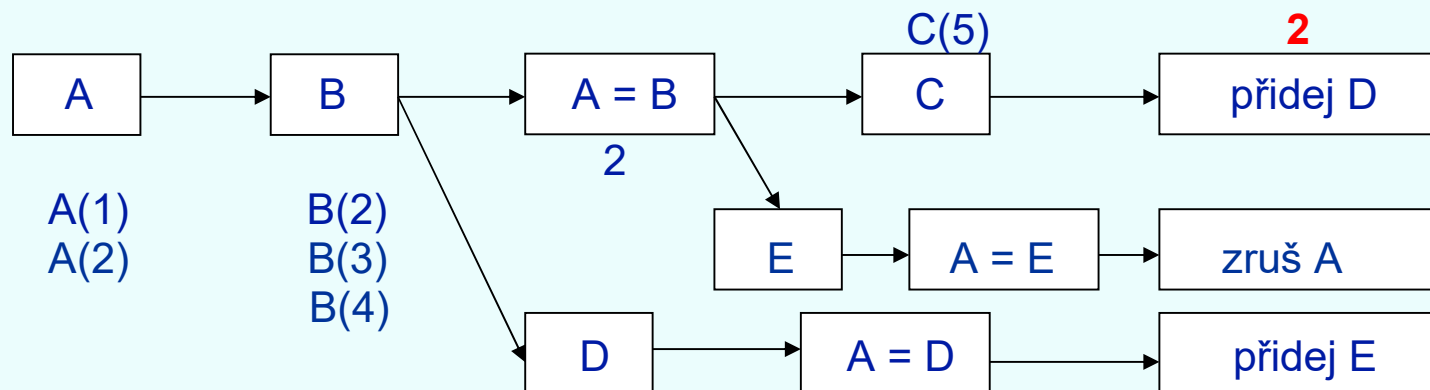
$A(x) \& B(x) \& C(y) \Rightarrow$  přidej  $D(x)$

$A(x) \& B(y) \& D(x) \Rightarrow$  přidej  $E(x)$

$A(x) \& B(x) \& E(x) \Rightarrow$  zruš  $A(x)$

- Pracovní paměť:

$A(1), A(2), B(2), B(3), B(4), C(5)$



## 4. Algoritmus RETE

### Šíření hodnot

- Báze znalostí:

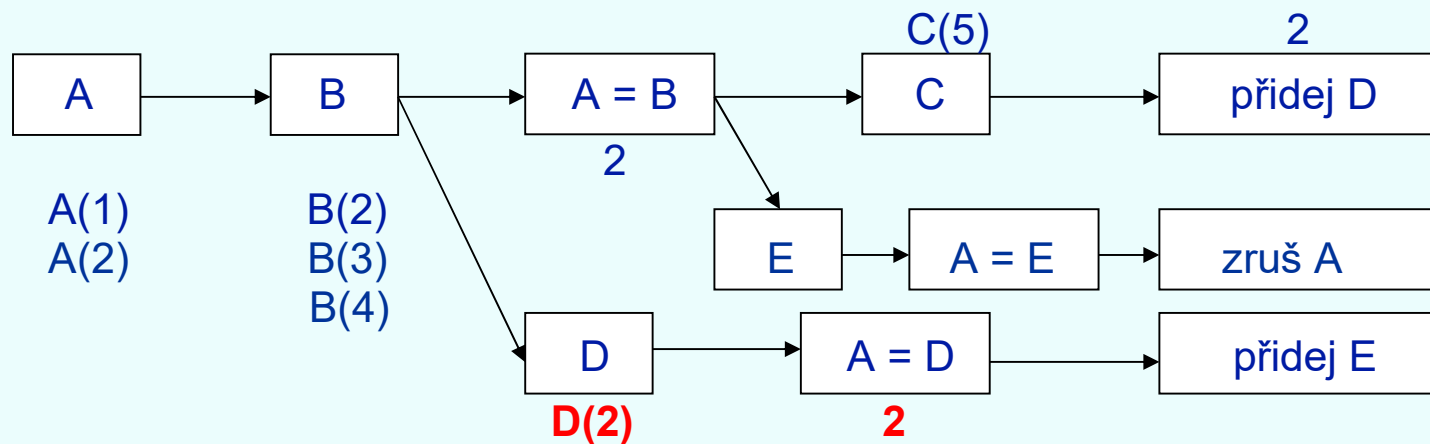
$A(x) \& B(x) \& C(y) \Rightarrow$  přidej  $D(x)$

$A(x) \& B(y) \& D(x) \Rightarrow$  přidej  $E(x)$

$A(x) \& B(x) \& E(x) \Rightarrow$  zruš  $A(x)$

- Pracovní paměť:

$A(1), A(2), B(2), B(3), B(4), C(5)$



## 4. Algoritmus RETE

### Provedení pravidla

- Báze znalostí:

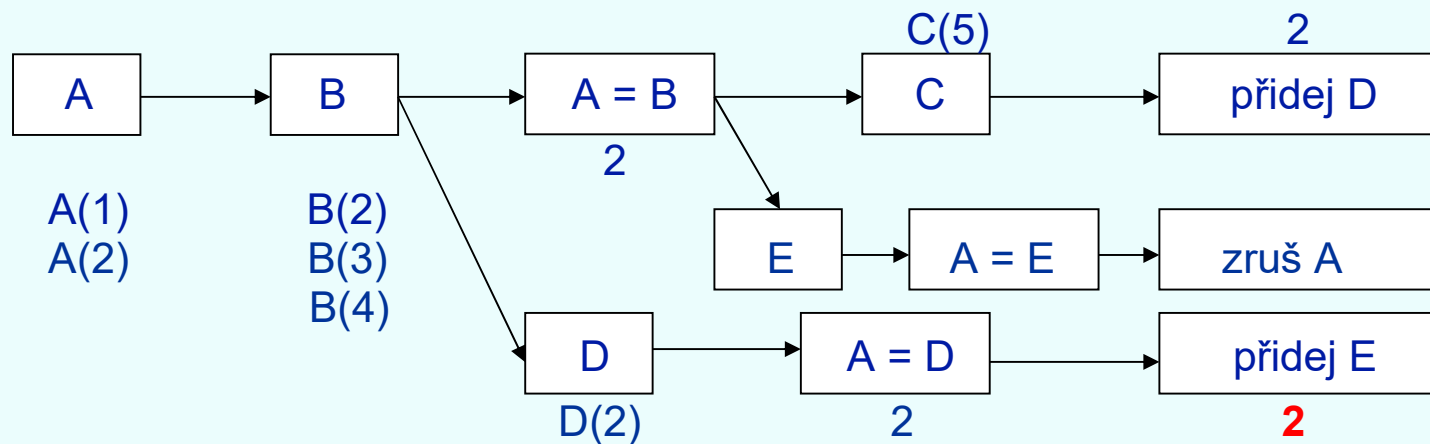
$A(x) \& B(x) \& C(y) \Rightarrow$  přidej  $D(x)$

$A(x) \& B(y) \& D(x) \Rightarrow$  přidej  $E(x)$

$A(x) \& B(x) \& E(x) \Rightarrow$  zruš  $A(x)$

- Pracovní paměť:

$A(1), A(2), B(2), B(3), B(4), C(5)$



## 4. Algoritmus RETE

### Šíření hodnot

- Báze znalostí:

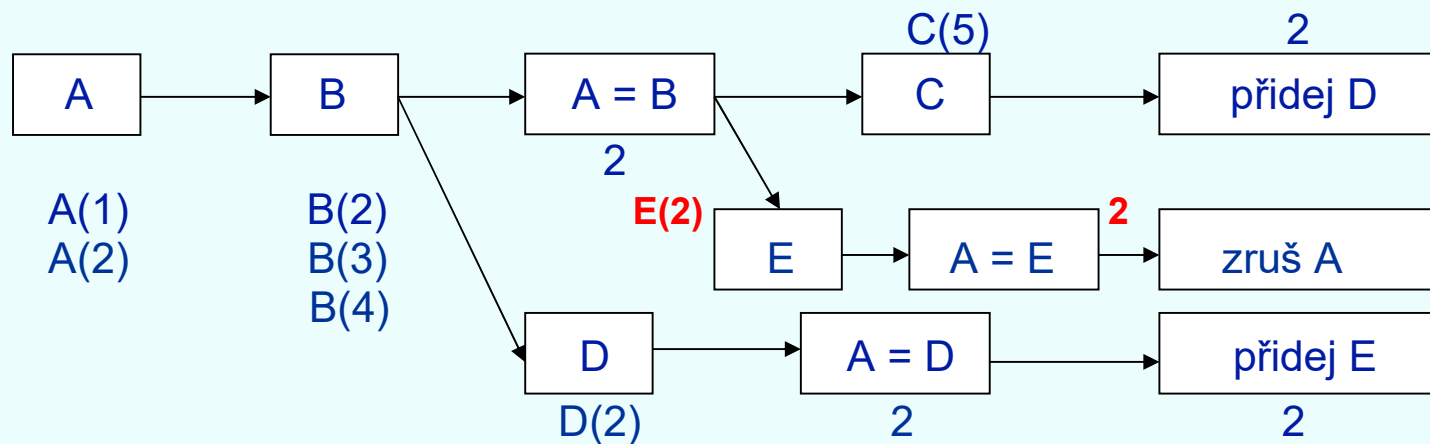
$A(x) \& B(x) \& C(y) \Rightarrow$  přidej  $D(x)$

$A(x) \& B(y) \& D(x) \Rightarrow$  přidej  $E(x)$

$A(x) \& B(x) \& E(x) \Rightarrow$  zruš  $A(x)$

- Pracovní paměť:

$A(1), A(2), B(2), B(3), B(4), C(5)$



## 4. Algoritmus RETE

### Provedení pravidla

- Báze znalostí:

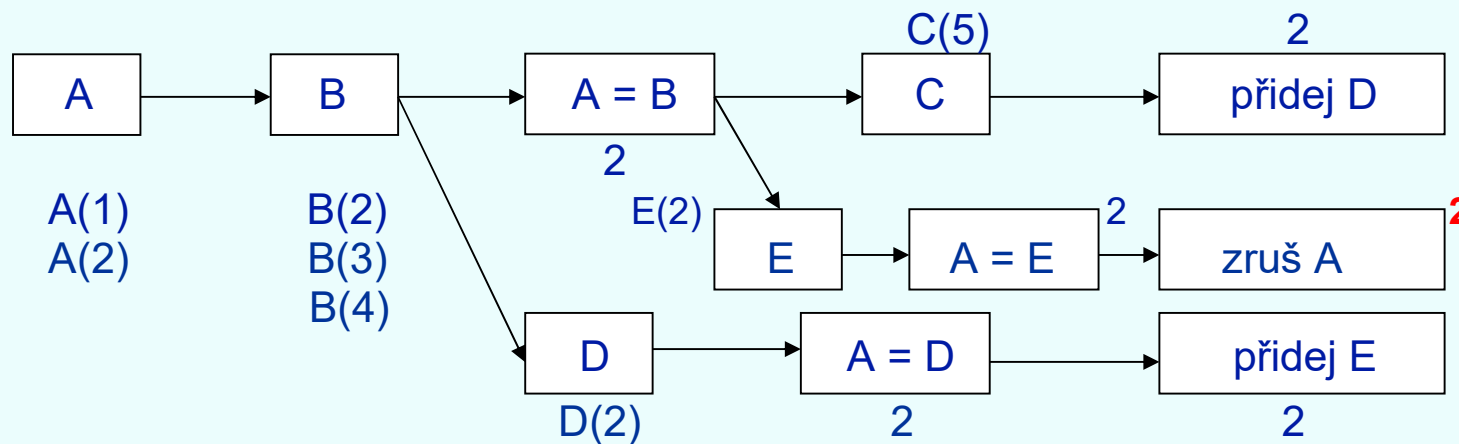
$A(x) \& B(x) \& C(y) \Rightarrow$  přidej  $D(x)$

$A(x) \& B(y) \& D(x) \Rightarrow$  přidej  $E(x)$

$A(x) \& B(x) \& E(x) \Rightarrow$  zruš  $A(x)$

- Pracovní paměť:

$A(1), A(2), B(2), B(3), B(4), C(5)$



## 4. Algoritmus RETE

### Šíření hodnot

- Báze znalostí:

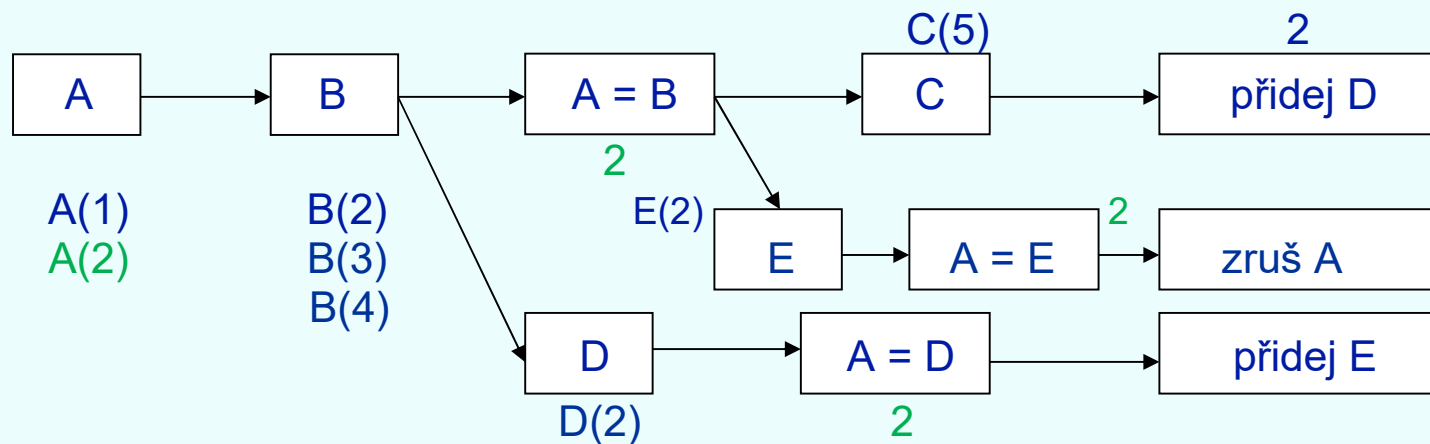
$A(x) \& B(x) \& C(y) \Rightarrow$  přidej  $D(x)$

$A(x) \& B(y) \& D(x) \Rightarrow$  přidej  $E(x)$

$A(x) \& B(x) \& E(x) \Rightarrow$  zruš  $A(x)$

- Pracovní paměť:

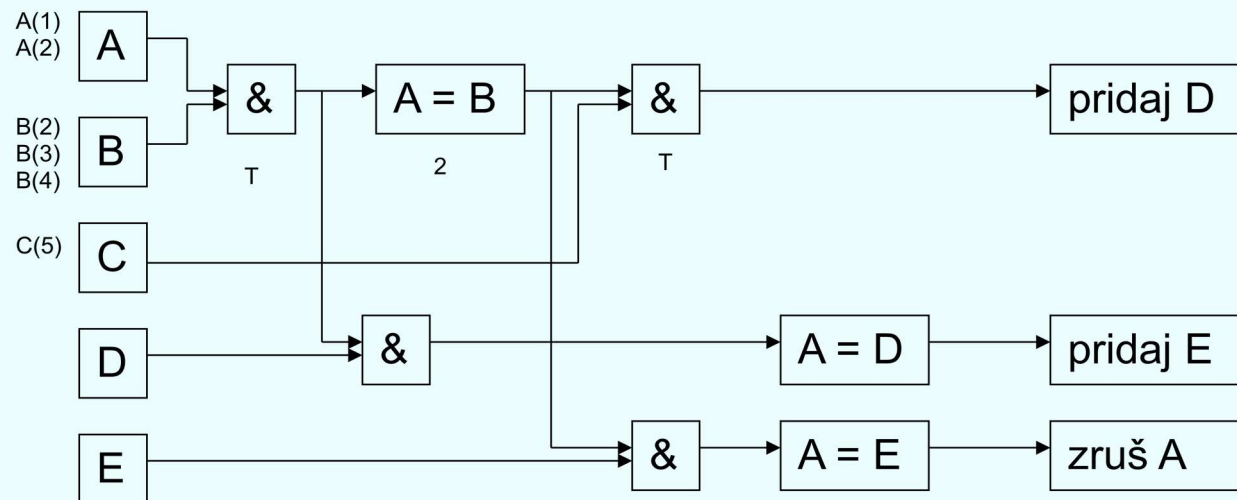
$A(1), A(2), B(2), B(3), B(4), C(5)$



## 4. Algoritmus RETE

### Alternativní tvar sítě

- Báze znalostí:  
 $A(x) \& B(x) \& C(y) \Rightarrow$  přidej  $D(x)$   
 $A(x) \& B(y) \& D(x) \Rightarrow$  přidej  $E(x)$   
 $A(x) \& B(x) \& E(x) \Rightarrow$  zruš  $A(x)$
- Pracovní paměť:  
 $A(1), A(2), B(2), B(3), B(4), C(5)$



## 4. Algoritmus RETE

---