

```

#include <stdlib.h>
#include <stdio.h>
char * s;
float A(), P(), Z();

float S() {
    float x = P();
    while(*s == '+' || *s == '-') {
        if(*s == '+') s++, x += P();
        if(*s == '-') s++, x -= P();
    }
    return x;
}

float P() {
    float x = A();
    while(*s == '*' || *s == '/') {
        if(*s == '*') s++, x *= A();
        if(*s == '/') s++, x /= A();
    }
    return x;
}

```

```

float A() {
    float x = 0;
    if(*s == '(') {
        s++;
        x = S();
        if(*s == ')') s++;
        else exit(1);
    }
    else x = Z();
    return x;
}

int main(int argc, char * argv[])
{
    float x;
    if(argc != 2) exit(1);
    s = argv[1];
    x = S();
    if(*s != 0) exit(1);
    printf("%f\n", x);
    return 0;
}

```

# Kontextfreie Sprachen

## Definition

- 1 Eine Sprache  $L$  ist *kontextfrei*, wenn es eine kontextfreie Grammatik  $G$  gibt, mit  $L = L(G)$ .
- 2 Eine CFG  $G$  ist *eindeutig*, wenn es zu jedem  $w \in L(G)$  genau einen Ableitungsbaum gibt.
- 3  $L$  ist eine *eindeutige kontextfreie Sprache*, wenn  $L = L(G)$  für eine eindeutige CFG  $G$ .

# Die $pre^*$ -Operation

## Definition

Es sei  $G = (N, T, P, S)$  eine CFG und  $L \subseteq (N \cup T)^*$  eine beliebige Sprache.

Wir definieren

$$pre_G^*(L) = \{ \alpha \in (N \cup T)^* \mid \alpha \xRightarrow{*}_G \beta \text{ für ein } \beta \in L \}.$$

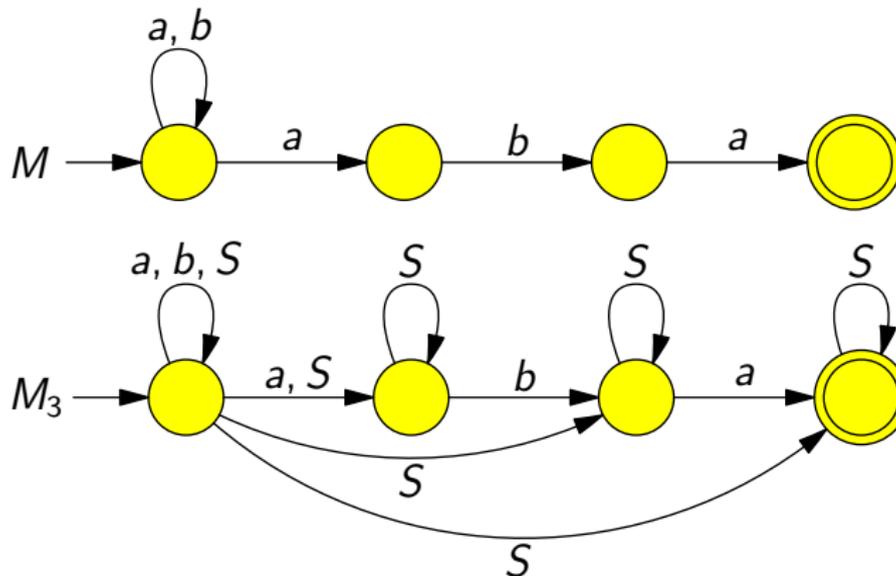
## Theorem

*Falls  $L$  regulär ist, dann ist auch  $pre_G^*(L)$  regulär.*

## Theorem

Falls  $L$  regulär ist, dann ist auch  $pre_G^*(L)$  regulär.

$G: S \rightarrow aSa \mid bSb \mid aSb \mid \epsilon$



# Sättigungsschritte

Sei  $L = L(M)$  für einen NFA  $M$ .

Wir transformieren  $M$  in einen NFA  $M'$  mit  $L(M') = pre_G^*(L)$  durch folgende Regeln:

Falls

- 1  $A \rightarrow \alpha \in P$  und
- 2  $p \in \hat{\delta}(q, \alpha)$

dann füge den Übergang  $q \xrightarrow{A} p$  zum Automaten hinzu.

Wiederhole dies, bis nichts mehr hinzugefügt werden kann.

## Definition

Es sei  $G = (N, T, P, S)$  eine CFG und  $M = (Q, \Sigma, \delta, q_0, F)$  ein NFA mit  $\Sigma = N \cup T$ .

Wir definieren für  $k \geq 0$ :

①  $M_G^0 = M$

②  $M_G^{k+1} = (Q, \Sigma, \delta', q_0, F)$  mit

$$\delta'(q, a) = \delta_{M_G^k}(q, a) \text{ falls } a \in T,$$

$$\delta'(q, A) = \delta_{M_G^k}(q, A) \cup \{p \in Q \mid p \in \hat{\delta}_{M_G^k}(q, \beta) \text{ und } A \rightarrow \beta \in P\}.$$

## Definition

Es sei  $L \subseteq \Sigma^*$  eine Sprache und  $G = (N, T, P, S)$  eine CFG mit  $\Sigma = N \cup T$ .

- 1  $pre^0(L) := L$
- 2  $pre^{k+1}(L) := \{ \alpha \mid \alpha \Rightarrow \beta \text{ f\"ur ein } \beta \in pre^k(L) \} \cup pre^k(L)$

Offensichtlich:

$$pre^*(L) = \bigcup_{n \geq 0} pre^n(L)$$

Induktion über  $k$ :  $pre^k(L) \subseteq L(M^k)$

Es sei  $\alpha \in pre^k(L)$ . Zwei Möglichkeiten:

- ①  $\alpha \in pre^{k-1}(L) \stackrel{IV}{\subseteq} L(M^{k-1}) \subseteq L(M^k)$ .
- ②  $\alpha \Rightarrow \beta$  und  $\beta \in pre^{k-1}(L) \subseteq L(M^{k-1})$ .

D.h.  $\alpha = \alpha_1 A \alpha_2$ ,  $\beta = \alpha_1 \gamma \alpha_2$  und  $A \rightarrow \gamma \in P$ .

Nach IV gilt  $\beta \in L(M^{k-1})$ , also

$$q_1 \in \hat{\delta}_{M^{k-1}}(q_0, \alpha_1)$$

$$q_2 \in \delta_{M^{k-1}}(q_1, A)$$

$$q_3 \in \hat{\delta}_{M^{k-1}}(q_2, \alpha_2) \text{ mit } q_3 \in F$$

Dann gilt  $q_2 \in \delta_{M^k}(q_1, A)$  und  $q_3 \in \hat{\delta}_{M^k}(q_0, \alpha)$ .

Induktion über  $k$ :  $L(M^k) \subseteq pre^*(L)$

Es sei  $\alpha \in L(M^k)$ .

Wieder zwei Möglichkeiten:

①  $\alpha \in L(M^{k-1})$ , dann  $\alpha \in pre^*(L)$  nach IV.

② Sonst:

$$\alpha = \alpha_0 A_0 \alpha_1 A_1 \dots \alpha_m A_m \alpha_{m+1}$$

$$\beta = \alpha_0 \beta_0 \alpha_1 \beta_1 \dots \alpha_m \beta_m \alpha_{m+1}$$

und  $\beta \in L(M^{k-1})$ ,  $A_i \rightarrow \beta_i \in P$  für  $0 \leq i \leq m$ .

Nach IV gilt  $\beta \in pre^*(L)$ .

Es gilt  $\alpha \xrightarrow{*} \beta$  und damit auch  $\alpha \in pre^*(L)$ .

# Beweis der zentralen Aussage

## Theorem

*Falls  $L$  regulär ist, dann ist auch  $pre_G^*(L)$  regulär.*

## Beweis.

Wir haben bereits gezeigt:

- 1  $pre^k(L) \subseteq L(M^k)$
- 2  $L(M^k) \subseteq pre^*(L)$

Außerdem:  $L(M^m) = L(M^{m+1}) = L(M^{m+2}) = \dots$  für ein  $m \in \mathbf{N}$ .

Es folgt daraus, daß  $L(M^m) = pre^*(L)$ .

$L(M^m)$  ist als Sprache eines NFAs regulär. □

# Das Wortproblem

## Definition

Das *Wortproblem* für CFG:

- Eingabe: Eine CFG  $G = (N, T, P, S)$  und ein Wort  $w \in T^*$
- Frage: Gilt  $w \in L(G)$ ?

## Theorem

*Das Wortproblem für CFG läßt sich in polynomieller Zeit lösen.*

## Beweis.

$w \in L(G)$  gdw.  $S \in pre^*({w})$ . □

# Unproduktive Symbole

## Definition

Finden von unproduktiven Symbolen einer CFG:

- Eingabe: Eine CFG  $G = (N, T, P, S)$
- Ausgabe: Eine Liste der unproduktiven Symbole von  $G$ :  
 $\{ A \in N \mid \text{es gibt kein } w \in T^* \text{ mit } A \xRightarrow{*} w \}$

## Theorem

*Unproduktive Symbole einer CFG lassen sich in polynomieller Zeit finden.*

## Beweis.

Die Menge der produktiven Symbole ist  $N \cap pre^*(T^*)$ . □