```c
#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;
}
```

```c
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;
}
```

RWTH AACHEN UNIVERSITY

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

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

Theorem

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

Theorem

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

$G: \; S \to aSa \mid bSb \mid aSb \mid \epsilon$