6. Übungsblatt Informatik II
(Abgabe: 23.05.2005)

1. **Aufgabe:**
   auf Kopie

2. **Aufgabe:**

   (a) $T = \{a\}$
   $N = \{S\}$
   Startsymbol: S
   $S- > a \mid aS$

   (b) $T = \{a, b, c\}$
   $N = \{S, A, B\}$
   Startsymbol: S
   S $- > AB$
   A $- > a \mid aA$
   B $- > bc \mid bBc$

3. **Aufgabe:**

Listing 1: if-then-else-Auswertung

```
#include <iostream>

using namespace std;

int main() {

  bool left_asso = true;

  if (true)                 // wird immer ausgefhrt
    if (false)
      return 0;             // wird ohnehin nie ausgefhrt
    else                    // wird bei rechtsklammerung
      left_asso = false;    // ausgefhrt

  cout << "Dieser Compiler klammert " <<
       (left_asso?"links":"rechts")
       << "assoziativ" << endl;

  return 0;
}
```

4. **Aufgabe:** Gegeben ist die folgende Grammatik G:

6. Übungsblatt Informatik II
(Abgabe: 23.05.2005)

$$\begin{aligned} AA &\vdash VC \\ &\vdash AA + AA \\ &\vdash AA -_2 AA \\ &\vdash AA * AA \\ &\vdash AA/AA \\ &\vdash -_1 AA \\ &\vdash (AA) \end{aligned}$$

Sei $last(w)$ das letzte Zeichen in w.
Behauptung 1: links von $-_2$ steht VC oder)

Beweis Beh.1:Induktion über die Anzahl der Ableitungsschritte
$AA \vdash_G^i w \quad , \quad w = w_1 -_2 w_2 \quad , \quad last(w_1) \in \{), VC\}$.

IA $i = 3 \quad AA \vdash AA -_2 AA \vdash VC -_2 AA \vdash VC -_2 VC$
$\Rightarrow w_1 = VC \quad , \quad last(w_1) = VC \in \{VC,)\}$.
IS $i \rightarrow i+1$

- Fall1: $w = (AA1) \quad AA1 \in L(G), AA \vdash_G^i AA1$
  $AA1 = w_1' -_2 w_2' \quad w_1 = (w_1' \quad w_2 = w_2') \quad last(w_1) = last((w_1') \in \{VC,)\}$ nach IV.
- Fall2: $w = -_1 AA1 \quad AA1 \in L(G), AA \vdash_G^i AA1$
  $AA1 = w_1' -_2 w_2' \quad w_1 = -_1 w_1' \quad w_2 = w_2' \quad last(w_1) = last(-_1 w_1') = last(w_1') \in \{VC,)\}$.
- Fall3: $w = AA1\, op\, AA2 \quad op \in \{+, -_2, -_1, *, /\}$
  Hier sind 3 Fälle möglich:
  $-_2$ liegt in $AA1$. Die Behauptung folgt nach IV.
  $-_2$ liegt in $AA2$. Die Behauptung folgt nach IV.
  $-_2$ ist op. Die Behauptung folgt nach IV.

Behauptung 2: links von $-_1$ steht $*, (, -_2, -_1, +, /, \varepsilon$.

Beweis Beh.2:Induktion über die Anzahl der Ableitungsschritte
$AA \vdash_G^i w \quad , \quad w = w_1 -_1 w_2 \quad , \quad last(w_1) \in \{(, +, -_2, -_1, /, *, \varepsilon\}$.

IA $i = 2 \quad AA \vdash -_1 AA \vdash -_1 VC \quad , \quad w_1 = \varepsilon \quad , \quad w_2 = VC \quad , \quad last(w_1) = \varepsilon \in \{(, +, -_2, -_1, /, *, \varepsilon\}$.
IS $i \rightarrow i+1$

- Fall1: $w = (AA1) \quad AA1 \in L(G), AA \vdash_G^i AA1$
  $AA1 = w_1' -_1 w_2' \quad w_1 = (w_1' \quad w_2 = w_2') \quad last(w_1) = last((w') \in \{(, +, -_2, -_1, /, *, \varepsilon\}$ nach IV.
- Fall2: $w = -_1 AA1 \quad AA1 \in L(G), AA \vdash_G^i AA1$
  $AA1 = w_1' -_1 w_2' \quad w_1 = -_1 w_1' \quad w_2 = w_2' \quad last(w_1) = last(-_1 w_1') = last(w_1') \in \{(, +, -_2, -_1, /, *, \varepsilon\}$.

- Fall3: $w = AA1\,op\,AA2 \quad op \in \{+, -_2, -_1, *, /\}$
  Hier sind 3 Fälle möglich:
  $-_1$ liegt in $AA1$. Die Behauptung folgt nach IV.
  $-_1$ liegt in $AA2$. Die Behauptung folgt nach IV.
  $-_1$ ist op. Die Behauptung folgt nach IV.

Behauptung 3: Sei $w \in L(G)$ und sei $last(w)$ das letzte Zeichen in w. Dann gilt:
$last(w) \in \{VC,)\}$

Beweis Beh.3:Strukturelle Induktion
IA $AA \vdash VC \Rightarrow last(VC) = VC \in \{VC,)\}$
IS

- $w = (AA), AA \in L(G) \quad last(w) = last((AA)) =) \in \{VC,)\}$.
- $w = AA_1 + AA_2 \quad AA_1, AA_2 \in L(G) \quad last(w) = last(AA_2) \in \{VC,)\}$ nach IV.
- $w = AA_1 - AA_2 \quad AA_1, AA_2 \in L(G) \quad last(w) = last(AA_2) \in \{VC,)\}$ nach IV.
- $w = AA_1 * AA_2 \quad AA_1, AA_2 \in L(G) \quad last(w) = last(AA_2) \in \{VC,)\}$ nach IV.
- $w = AA_1 / AA_2 \quad AA_1, AA_2 \in L(G) \quad last(w) = last(AA_2) \in \{VC,)\}$ nach IV.
- $w = -_1 AA_1, AA_1 \in L(G) \quad last(w) = last(AA_1) \in \{VC,)\}$ nach IV.

$\Rightarrow$ Beh. 3.

5. **Aufgabe:**

Die Fälle bei denen 'int' oder ';' oder '= 1' in dem vx vorkommt müssen gar nicht betrachtet werden da man bei Erhöhung der v bzw x sofort aus der Sprache fällt. Wenn $vwx = a^j$ bzw b^j mit $j < n$ so liegt uv^iwx^iy auch nicht mehr in der Sprache da die Anzahl der $a's$ bzw $b's$ auf einer der beiden Seiten nicht mehr mit der Anzahl auf der anderen Seite übereinstimmt.

Aus dem gleichen Grund verlässt man die Sprache wenn vwx aus a^jb^j besteht, auch hier stimmt nach einer Erhöhung die Anzahl der $a's$ und die Anzahl der $b's$ auf einer der beiden Seiten nicht mehr mit der entsprechenden Anzahl auf der anderen Seite überein.

Der einzig verbleibende interressante Fall ist der bei dem $v = b^j, w =;$, und $x = a^j$ nach einer Erhöhung erhält man $int\ a^N b^{N+i}; a^{N+i} b^N = 1$ und hat die Sprache erneut verlassen. Man findet also keinen Fall bei dem a.), b.) und c.) gilt, und damit ist die Programmfamilie nicht kontextfrei.