

Compilerbau

Sommersemester 2004

5. Übungsblatt

Abgabetermin: 1. Juni 2004, 12 Uhr

Aufgabe 1:

8 Punkte

Sei $\Sigma = \{a, b\}$. Gib jeweils eine kontextfreie Grammatik für die folgenden Sprachen an:

- $\{uv \mid u, v \in \Sigma^*; v = u^R\}$;
- $\{uv \mid u, v \in \Sigma^*; \#_a u = \#_b v\}$;
- $\{uv \mid u, v \in \Sigma^*; |u| = |v|; u \neq v\}$,

wobei w^R das reverse, also rückwärts gelesene Wort w , $\#_x w$ die Anzahl der Vorkommen des Symbols x in w , und $|w|$ die Länge des Wortes w bezeichnet.

Aufgabe 2:

6 Punkte

Gib eine kontextfreie Grammatik an, die die Syntax von Prolog-Programmen spezifiziert. Definiere dazu die geeigneten Token-Klassen. Ein Prologprogramm ist z.B.:

```
parent(jon, bon) .  
parent(bon, jovi) .  
ancestor(X, Y) :- parent(X, Y) .  
ancestor(X, Y) :- parent(X, Z), ancestor(Z, Y) .
```

Ein Prolog-Programm besteht also aus einer Menge von Klauseln. Eine Klausel ist entweder ein Faktum (wie z.B. `parent(jon, bon) .`) oder eine Regel (z.B. `ancestor(X, Y) :- parent(X, Y) .`). Der Einfachheit halber verzichten wir auf built-in Prädikate wie z.B. `assert`, `retract` und Arithmetik. Ausser Prädikate können Terme in Regeln und Fakten Variablen, Zahlen, Atome oder Konstruktor-Anwendungen sein. Ein Atom oder Konstruktor ist ein Name, der mit einem Kleinbuchstaben beginnt. Eine Variable ist ein Name, der mit einem Grossbuchstaben oder Unterstrich anfängt.

Aufgabe 3:

6 Punkte

In einer erweitert kontextfreien Grammatik haben die Produktionen die Form $A \rightarrow e$ mit $A \in N$ und $e \in E_{NUT}$ (e ist ein regulärer Ausdruck über Terminalen und Nichtterminalen). Anwenden einer Produktion $A \rightarrow e$ heißt, dass A durch ein Wort $\alpha \in [e]$ ersetzt wird. Zum Beispiel, mit Hilfe der Regel $A \rightarrow (aB)^*$ erhält man: $cAc \rightarrow caBaBc$. Gib eine Transformation an, die zu einer erweitert kontextfreien Grammatik eine äquivalente (d.h. die selbe Sprache definierende) kontextfreie Grammatik konstruiert.