Dr. Andreas Borchert
Michael Wiedemann
Abteilung Angewandte Informationsverarbeitung

13. Mai 2004
Blatt 3

Allgemeine Informatik II (SS 2004)

Abgabetermin: 27. Mai 2004

3 Parser für Lindenmayer-Systeme – 15 Punkte

Gegeben sei folgende Grammatik zur Beschreibung von einfachen deterministischen Lindenmayer-Systemen:

- V_T = { **Ident**, `":"`, `"->"`, `";"`, `"F"`, `"+"`, `"-"`, `"["`, `"]"` }
- V_N = { **CommandSequence**, **Command**, **Production**, **Start**, **System** }
- S = **System**
- Produktionsregeln in EBNF:

```
System = Start { Production } .
Start = CommandSequence ":" .
Production = Ident "->" CommandSequence ";" .
CommandSequence = { Command } .
Command = Ident | "F" | "+" | "-" | "[" CommandSequence "]" .
```

Als **Ident** sind dabei nur Großbuchstaben aus dem Bereich "A" bis "E" und "G" bis "Z" zugelassen. "F" darf nicht als **Ident** erkannt werden, da es sich dabei um den Vorwärts-Operator des Turtle-Grafiksystems handelt. Zwischen den Terminals sollten Leerzeichen einschließlich Tabulatoren und Zeilentrennern erlaubt sein.

Als Beispiel für einen Satz dieser Grammatik mögen Drachenkurven dienen:

```
FX:
X -> X+YF+;
Y -> -FX-Y;
```

Zu implementieren sind eine vollständige lexikalische und syntaktische Analyse von Sätzen dieser Grammatik. Im Erfolgsfalle muß dabei eine Datenstruktur entstehen, die neben der Startsequenz alle Produktionsregeln des beschriebenen Lindenmayer-Systems enthält. Um dies überprüfen zu können, sollten Sie eine Prozedur schreiben, die ausgehend von der Datenstruktur das gesamte System in lesbarer Form ausgibt, wofür geschickterweise wieder die gleiche Syntax verwendet wird.

Syntaxfehler müssen uneingeschränkt erkannt werden und sollten zu einer pauschalen Fehlermeldung führen. Wenn für ein **Ident** eine Produktionsregel mehrfach definiert wird, sollte dies ebenfalls als Fehler erkannt werden. Etwas aufwendiger und daher optional ist die Überprüfung, ob es für alle referenzierten **Ident** auch eine zugehörige Produktionsregel gibt.

Hinweise:

Beginnen Sie mit der Entwicklung einer geeigneten Datenstruktur für die Repräsentierung von Lindenmayer-Systemen. Da es nur begrenzt viele Möglichkeiten für **Ident** gibt, ist damit auch die Zahl der Produktionen limitiert. Für die Länge einer **CommandSequence** bzw. eines **Command** können Sie eine geeignete obere Grenze wählen.

Ihr Parser muß durch eine Prozedur repräsentiert werden, der einen Eingabestrom erhält vom Typ **Streams.Stream** und über einen **VAR**-Parameter (im Erfolgsfalle) die eingelesene Datenstruktur zurückliefert. Der Rückgabewert vom Typ **BOOLEAN** soll angeben, ob die Analyse erfolgreich war oder nicht. Der Parser darf auf keine globalen Variablen zugreifen.

Beachten Sie, daß Sie hier mehr als einen Attributtyp haben. Das Attribut, das der Parser zurückliefert, ist hier die Datenstruktur für das gesamte Lindenmayer-System. Non-Terminals wie **Command** oder **CommandSequence** haben jedoch sehr viel einfachere Attributtypen.

Viel Erfolg!