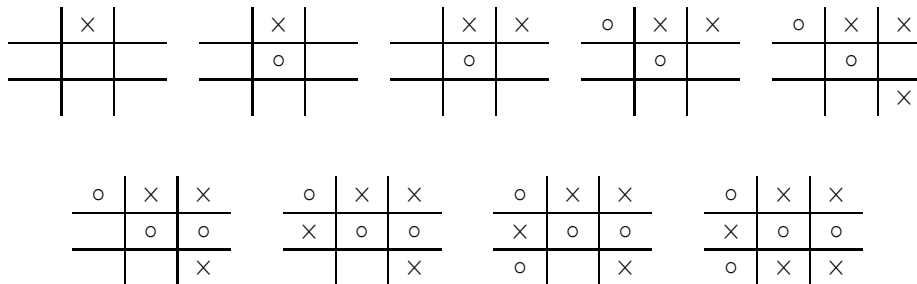


Aufgabe 1 (Tic-Tac-Toe)**20 Punkte**

In dieser Aufgabe soll das bekannte Spiel Tic-Tac-Toe mit Hilfe einer attribuierten Grammatik beschrieben werden.



Die obenstehende Graphik zeigt einen typischen Verlauf (x beginnt) eines Spiels, das unentschieden endet, da es keinem der beiden Spieler gelingt, eine waagerechte, senkrechte oder diagonale Reihe von drei Symbolen zu vervollständigen.

Sie dürfen davon ausgehen, dass das Spiel nach der ersten Dreierreihe sofort beendet wird. Doppelte Feldbelegungen kommen nicht vor und dürfen hier vernachlässigt werden. Verwenden Sie zur Beschreibung der belegten Felder bzw. Züge Zahlen von 1 bis 9. Für den oben dargestellten Spielverlauf lautet die Zugfolge 8 9 3 4 2 5 7 6 1, wobei die ersten fünf Züge vom ersten und die letzten vier vom zweiten Spieler ausgeführt werden.

- (a) Erstellen Sie eine Grammatik mit Attributen und semantischen Regeln, die eine gültige Zugfolge des Spiels Tic-Tac-Toe beschreibt. Das Ergebnis des Spiels soll als ganze Zahl ausgegeben werden, wobei 0 für ein Unentschieden steht und 1 bzw. 2 für einen Sieg des ersten bzw. zweiten Spielers. **15 Punkte**
- (b) Geben Sie den Datenflussgraphen, d.h. den Ableitungsbaum mit Belegungen der Attribute, für die Zugfolge 5 8 2 6 4 an. Tragen Sie dabei die Werte der vererbten Attribute links neben das Nichtterminal und die Werte der synthetisierten Attribute rechts neben das Nichtterminal ein. **5 Punkte**

Aufgabe 2 (Beseitigung von Linksrekursionen)**15 Punkte**

Im Kurstext wurde gezeigt, wie man Linksrekursion aus einem Übersetzungsschema, in dem jedes Grammatiksymbol ein synthetisiertes Attribut besitzt, entfernt. Verallgemeinern Sie diese Technik für den Fall, daß zwei synthetisierte Attribute auftreten. Betrachten Sie als Ausgangspunkt das Schema

$$A \rightarrow A_1 Y \quad \{ A.a_1 := g_1(A_1.a_1, A_1.a_2, Y.y) \\ A.a_2 := g_2(A_1.a_1, A_1.a_2, Y.y) \},$$

$$A \rightarrow X \quad \{A.a_1 := f_1(X.x) \\ A.a_2 := f_2(X.x) \}.$$

15 Punkte Aufgabe 3 (Übersetzung von Postfix-Ausdrücken)

8 Punkte (a) Entwickeln Sie eine attributierte Grammatik, die in der Lage ist, Terme über positive Ganzzahlen (einschließlich 0) und die Operatoren $\{+, -, *\}$ in Postfix-Notation in äquivalente Ausdrücke in Infix-Notation zu übersetzen. Für die Ausgangssprache sei $\Sigma = \{_, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *\}$, wobei „_“ ein zwischen Zahlen und Operatoren einfach zu verwendendes Trennsymbol sei.

1 Punkt (b) Welche der verwendeten Attribute sind *vererbt*, welche *synthetisiert*?

6 Punkte (c) Zeigen Sie die Datenflussgraphen für die Berechnungen Ihrer Grammatik auf folgenden Wörtern:

17_3+_2_*

9_265_-_3_14_*_+_

50 Punkte Aufgabe 4 (myPS-4 „Semantische Prüfungen“)

In der letzten Kurseinheit wurde in der Aufgabe zu myPS auf die Implementierung der Übersetzungsregeln für `union`, `concat`, `plot` und `scaletobox` verzichtet. Bitte holen Sie die Implementierung der zugehörigen Übersetzungsregeln für die genannten Funktionen jetzt nach.

Damit sind wir nun in der Lage, korrekte myPS Programme in Postscript zu überführen. Dabei werden bislang nur syntaktische Fehler durch den Parser abgefangen. Andere Fehler, z.B. Typkonflikte, werden erst zur „Laufzeit“, d.h. durch den Postscript-Interpreter erkannt. Einige dieser Prüfungen können und sollen jedoch schon während der Übersetzungszeit stattfinden.

Im zweiten Teil der Aufgabe soll damit begonnen werden, entsprechende Prüfungen in den Compiler einzubauen:

1. Nur deklarierte Variablen werden benutzt.
2. Variablen dürfen nur einmal deklariert werden.
3. Die Typen innerhalb von Zuweisungen, Termen und Ausdrücken müssen zueinander passen.

Fügen Sie die oben angegebenen Prüfungen in Ihren Compiler ein.