

Übungsblatt 3: Software-Entwicklung 1 (WS 2010/11)

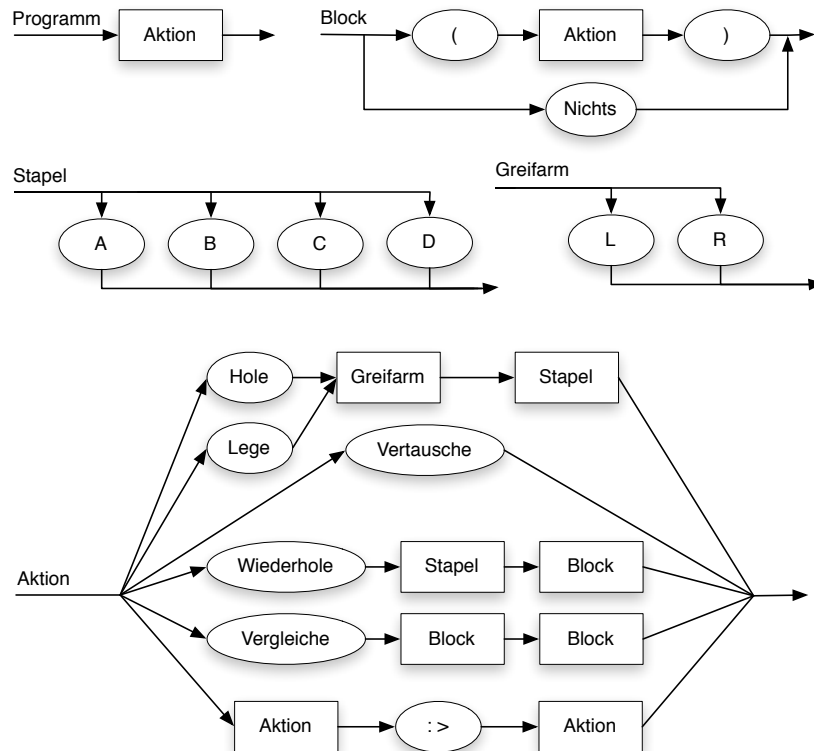
Ausgabe: in der Woche vom 8.11. bis zum 12.11.10

Abgabe: in der Woche vom 15.11. bis zum 19.11.10

Abnahme: max. zwei Tage nach der Übung

Aufgabe 1 Syntaxdiagramme (Präsenzaufgabe)

Gegeben seien folgende Syntaxdiagramme, welche die Sprache der Roboter-Algorithmen definieren (Fortsetzung von Blatt 2). Startsymbol ist Programm. Leere Blöcke werden durch Nichts dargestellt.



a) Entscheiden Sie, ob folgende Sätze zur eben definierten Sprache gehören (Leerzeichen dürfen Sie ignorieren, diese dienen nur der besseren Darstellung):

1. Hole A L :> Lege B L
2. Wiederhole D (Hole R D :> Lege R D)
3. Hole R B :> Wiederhole Nichts (Lege L B)
4. Lege L A :> Hole R B :> Wiederhole A (Vertausche)
5. Hole L C :> Hole R C :> Vergleiche Nichts (Lege L B)

b) Besitzen die in a) definierten Sätze eine sinnvolle Bedeutung (Semantik)? Diskutieren Sie.

Aufgabe 2 Roboter-Programme in Haskell (Einreichaufgabe)

In dieser Aufgabe arbeiten wir mit dem Haskell Compiler *ghc*. Auf der Vorlesungs-Seite finden Sie eine Bibliothek mit dem Namen `RoboLib.hs`, die Sie herunterladen und in Ihrem Home-Verzeichnis speichern sollen. Mit dieser Bibliothek können Sie Roboter-Programme nun nach folgendem Muster schreiben:

```
import RoboLib
main = runRobot prog
prog = <ROBOTER-PROGRAMM>
```

Als `<ROBOTER-PROGRAMM>` können Sie u.a. alle Programme eingeben, die durch das Syntaxdiagramm in [Aufgabe 1](#) beschrieben werden. Verwenden Sie Leerzeichen und weitere Klammern als Strukturierungshilfe.

Haben Sie das Programm in einer Datei mit dem Namen `robot.hs` gespeichert, übersetzen Sie es durch

```
ghc --make robot.hs
```

Der Compiler sollte die `RoboLib.hs` automatisch finden und ebenfalls übersetzen. Hat alles funktioniert, können Sie den Roboter mit dem folgenden Befehl starten:

```
./robot
```

Der Roboter kann beim Aufruf durch Parameter konfiguriert werden, über die Sie sich mit dem Parameter `-h` oder `--help` informieren können. Durch Drücken der Taste `h` innerhalb des Programms können Sie sich über die Befehle an den Roboter informieren.

Im Standardfall wird der Roboter mit 24 Karten gestartet, die in zufälliger Reihenfolge auf Stapel A liegen. Das Programm wird angezeigt und kann schrittweise oder auch komplett ausgeführt werden. Ein Zwischenzustand des Roboters kann beispielsweise so aussehen (mit `-k 16`):

Programm:		Protokoll:		
Wiederhole A		Start		
Hole L A		Prüfe, ob Stapel A leer ist: nein		
Lege L B		Karte 2 auf L von A geholt		
Ende		Karte 2 von L auf B gelegt		
Error		Prüfe, ob Stapel A leer ist: nein		
		Karte 4 auf L von A geholt		
		Karte 4 von L auf B gelegt		12
		Prüfe, ob Stapel A leer ist: nein		14
		Karte 13 auf L von A geholt		15
		Karte 13 von L auf B gelegt		9
		Prüfe, ob Stapel A leer ist: nein		6
		Karte 16 auf L von A geholt		11
		Karte 16 von L auf B gelegt		10
		Prüfe, ob Stapel A leer ist: nein		16
		Karte 7 auf L von A geholt		< 8 13
				7 \====/ 3 4
				/ / \ \ 5 2
				<-----> -----
				A B C

- a) Führen Sie Ihre Roboter-Programme von Blatt 2 aus und testen Sie sie mit verschiedenen Eingaben. Versuchen Sie häufige Aktionssequenzen, wie in 2b von Blatt 2 motiviert, wiederzuverwenden:

```
b_nach_c = Hole L B :> Lege L C
```

- b) Der Haskell-Roboter kann einen weiteren Stapel *D* verwenden. Schreiben Sie ein Roboterprogramm, dass die Karten vom Anfangsstapel gleichmäßig auf drei Stapel aufteilt. Wie sieht es ohne den Zusatzstapel *D* aus?

- c) Schreiben Sie ein Roboterprogramm, das den Kartenstapel so aufteilt, dass alle Karten mit ungradem Wert auf einem Stapel liegen, alle mit gradem auf einem anderen.
- d) (*freiwillige Zusatzaufgabe*) In Aufgabe 2b bzw. 2c von Blatt 2 haben Sie den sogenannten *Selectionsort* Algorithmus (Sortieren durch Auswahl) zum Sortieren benutzt und implementiert. Dieser wird Ihnen, zusammen mit anderen Sortieralgorithmen, über das Semester in der Vorlesung noch genauer begegnen.

Ebenfalls sehr einfach ist jedoch der duale *Insertionsort* Algorithmus (Sortieren durch Einfügen). Im Gegensatz zum Selectionsort wird hier nicht bei jeder Wiederholung die nächste Karte aus dem unsortierten Stapel gesucht, sondern es wird die Stelle gesucht, in die die oberste Karte des unsortierten Stapels eingefügt werden soll. Es wird also nicht die nächst kleinere/größere *ausgewählt*, sondern eine beliebige nächste Karte wird in einen bereits sortierten Stapel *eingefügt*.

Schreiben und testen Sie ein Roboterprogramm, das den Anfangsstapel mit Hilfe des Insertionsort Algorithmus sortiert und anschließend genauso zurücklässt wie in Aufgabe 2c von Blatt 2 verlangt. Sie tauschen also die Implementierung aus, erhalten aber die Funktionalität.

Aufgabe 3 Kontextfreie Grammatiken (Präsenzaufgabe)

Die formale Sprache L ist durch die Grammatik $\Gamma = (N, T, \Pi, V)$ mit

$$\begin{aligned} N &= \{S, U, V\} \\ T &= \{a, b, c, d\} \\ \Pi &= \left\{ \begin{array}{l} S \rightarrow c \\ U \rightarrow UU \\ U \rightarrow bS \\ V \rightarrow aV \\ V \rightarrow Ud \end{array} \right\} \end{aligned}$$

definiert.

- a) Welches ist der kürzeste Satz der Sprache L? Geben Sie alle Sätze der Sprache (mit Ableitung) an, die bis zu fünf Buchstaben besitzen.
- b) Überprüfen Sie, ob Ihre Ableitungen aus Aufgabenteil a *Linksableitungen* sind. Falls nicht, geben Sie zu den entsprechenden Sätzen auch eine Linksableitung an. Können Sie für einen der Sätze auch eine andere Linksableitung angeben?
- c) Ist die Grammatik Γ eindeutig? Wenn nicht, geben Sie ein Gegenbeispiel an.
- d) Wir möchten eine Datumsangabe modellieren, bei der chronologische Ordnung von Daten mit der Ordnung auf natürlichen Zahlen zusammenfällt. Dafür steht die Angabe vom Jahr vor dem Monat und dieser wiederum vor dem Tag. Angabe von Tag und Monat müssen entsprechend mit einer Null aufgefüllt werden, falls sie nur einstellig sind:

Beispiele:

1.1.2000		20000101
31.12.3000	werden zu	30001231
14.3.2517		25170314

Geben Sie eine kontextfreie Grammatik an, welche genau alle Datumsangaben zwischen dem 1.1.2000 und dem 31.12.3000 in diesem Format erzeugen kann. Gehen Sie vereinfachend davon aus, dass alle Monate 31 Tage haben. Sie können ein beliebiges Alphabet Ihrer Wahl zu Grunde legen und abkürzende Schreibweisen für Mengen verwenden.

Aufgabe 4 Syntax formaler Sprachen (Einreichaufgabe)

- Geben Sie eine kontextfreie Grammatik an, welche genau alle Datumsangaben zwischen dem 1.1.2010 und dem 31.12.2030 erzeugen kann und dabei die unterschiedliche Anzahl an Tagen der Monate berücksichtigt. Schaltjahre sollen nicht betrachtet werden, d.h. der Monat Februar wird mit konstant 28 Tagen angenommen. Für diese Aufgabe sollen sie **keine** abkürzenden Schreibweisen mehr verwenden!
- Geben Sie eine kontextfreie Grammatik an, die der Sprachdefinition mit Syntaxdiagrammen aus [Aufgabe 1](#) entspricht.
- Übersetzen Sie Grammatik Γ aus [Aufgabe 3](#) in ein Syntaxdiagramm.
- Versuchen Sie, ein möglichst minimales Syntaxdiagramm zu erstellen, das ebenfalls die Sprache L beschreibt.

Hinweis: Vereinfachen Sie schrittweise Ihr Syntaxdiagramm, indem Sie es auf eine Kombination der "Grundtypen von Syntaxdiagrammen" reduzieren und Nicht-Terminalsymbole eliminieren.

- Entscheiden Sie, ob der Satz $aaaabcbcbcbcbcd$ zur Sprache L gehört und geben Sie ggf. eine Ableitung an.
- Geben Sie eine *eindeutige* kontextfreie Grammatik an, welche die Sprache L beschreibt.

Aufgabe 5 Sprachen und Grammatiken (Einreichaufgabe)

Im Folgenden betrachten wir eine Familie von Grammatiken $\Gamma_i = (N, T, \Pi_i, Q)$, mit Nicht-Terminalsymbolen $N = \{Q, F, H, L\}$ und Terminalsymbolen $T = \{k, q, z, b\}$, die sich also nur in ihren Produktionen Π_i unterscheiden:

$$\begin{aligned} \Pi_1 &= \{ \} & \Pi_2 &= \{ Q \rightarrow \epsilon \} & \Pi_3 &= \left\{ \begin{array}{l} Q \rightarrow Q \\ Q \rightarrow k \end{array} \right\} & \Pi_4 &= \left\{ \begin{array}{l} Q \rightarrow kzQb \\ Q \rightarrow qb \end{array} \right\} & \Pi_5 &= \left\{ \begin{array}{l} Q \rightarrow F \\ Q \rightarrow H \\ F \rightarrow q \\ F \rightarrow QF \end{array} \right\} & \Pi_6 &= \left\{ \begin{array}{l} Q \rightarrow FH \\ F \rightarrow Q \\ H \rightarrow F \end{array} \right\} \\ \Pi_7 &= \left\{ \begin{array}{l} Q \rightarrow Hb \\ H \rightarrow HH \\ H \rightarrow kF \\ F \rightarrow qF \\ F \rightarrow z \end{array} \right\} & \Pi_8 &= \left\{ \begin{array}{l} Q \rightarrow Fb \\ H \rightarrow Hq \\ H \rightarrow \epsilon \\ F \rightarrow FF \\ F \rightarrow kHz \end{array} \right\} & \Pi_9 &= \left\{ \begin{array}{l} Q \rightarrow Hb \\ H \rightarrow HH \\ H \rightarrow kF \\ F \rightarrow qF \\ F \rightarrow H \end{array} \right\} & \Pi_{10} &= \left\{ \begin{array}{l} Q \rightarrow Hb \\ H \rightarrow Fq \\ H \rightarrow \epsilon \\ F \rightarrow FL \\ L \rightarrow kHz \end{array} \right\} \end{aligned}$$

- Wie viele verschiedene Sprachen werden durch die obigen Grammatiken definiert? Begründen Sie!
- Welche Sprachen werden definiert?

Hinweis: Verwenden Sie eine Mengennotation der Form $L(\Gamma_4) = \{(kz)^n qb^{n+1} \mid n \geq 0\}$, wobei k^n für die n -fache Wiederholung von k steht. Beispielsweise ist k^5 die Kurzschreibweise für $kkkkk$.

- Geben Sie eine kontextfreie Grammatik an, welche die Vereinigung all dieser Sprachen definiert.
- Überlegen Sie sich, wie Sie zeigen könnten, dass zwei Grammatiken dieselbe Sprache definieren. Geben Sie zwei verschiedene Grammatiken Ihrer Wahl an, und beweisen Sie, dass sie dieselbe Sprache definieren.