

Ableitungsbäume einer CFG

Aufgabenstellung

Hinweis: Aus Zeitgründen konnten wir diese Woche die Templates für Python und Haskell nicht gleichzeitig mit der Blattausgabe bereitstellen. Wir werden sie vermutlich in den nächsten Tagen (voraussichtlich Dienstag nachmittag) nachreichen.

In dieser Aufgabe haben Sie einen PDA (akzeptierend mit leerem Keller) gegeben und sollen diesen **exakt nach dem Verfahren aus der Vorlesung** (Satz 4.60) in eine äquivalente CFG umwandeln. Beachten Sie dabei folgendes:

- Das Startsymbol muss S heißen.
- Alle anderen Nichtterminale müssen Tripel der Form $[q_1; Z, q_2]$ sein, wobei q_1, q_2 Zustandsnamen aus dem Eingabe-PDA sind und Z ein Stacksymbol ist.
- Beachten Sie, dass wir für die Notation der Tripel Semikola verwenden, nicht Kommata wie in der Vorlesung!
- Die CFG, die Sie zurückgeben, muss *alle* Nichtterminale dieser Form enthalten, nicht nur die erreichbaren.
- Um dies zu präzisieren: Der Test erfolgt auf exakte Stringgleichheit, d.h. Ihre Ausgabe muss wirklich *exakt* die gleiche sein wie die Musterlösung, inkl. Zustandsnamen etc.

Implementierung

Machen Sie sich wie in den vorherigen beiden Aufgaben mit der Klasse `ContextFreeGrammar` vertraut. Machen Sie sich dann außerdem mit der Klasse `PDA` vertraut. Implementieren Sie schließlich die Methode `PDA.toCFG`, die die oben beschriebene Konstruktion ausführt. Beachten Sie dabei, dass ϵ -Transitionen das Label `Transition.EPSILON` haben und Sie diese anders behandeln müssen als “normale” Transitionen.

Tipp: Eine Hilfsfunktion `<A> Set<LinkedList<A>> lists(int n, Set<A> elems)`, die die Menge aller Listen der Länge n mit Elementen aus `elems` berechnet, könnte nützlich sein.

Eingabe

Ein PDA in der Syntax, die sich aus den Beispielen ablesen lässt.
Das Einlesen der Eingabe ist für Sie bereits in den Vorlagen implementiert.

Ausgabe

Eine CFG in der Syntax, die sich aus den Beispielen ablesen lässt.
Die Ausgabe in der korrekten Syntax (inkl. Sortierung) übernimmt das Template für Sie.

Beispiele

Weitere Beispiele finden Sie in dem .tar.gz-Archiv auf der Webseite.

Sample Input 1

```
PDA
Alphabet: a;b
States: q;p
Init: q
Stackalphabet: A;X
Startsymbol: X
Transitions:
q;a;X;q;A
q;a;A;q;AA
q;b;A;p;
p;b;A;p;
END
```

Sample Output 1

```
Grammar
Nonterminals: S, [p;A;p], [p;A;q], [p;X;p], [p;X;q], [q;A;p], [q;A;q], [q;X;p], [q;X;q]
Alphabet: a,b
Startsymbol: S
Productions:
S -> [q;X;p]
S -> [q;X;q]
[p;A;p] -> b
[q;A;p] -> a [q;A;p] [p;A;p]
[q;A;p] -> a [q;A;q] [q;A;p]
[q;A;p] -> b
[q;A;q] -> a [q;A;p] [p;A;q]
[q;A;q] -> a [q;A;q] [q;A;q]
[q;X;p] -> a [q;A;p]
[q;X;q] -> a [q;A;q]
END
```