

Erreichbare Symbole 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 eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ gegeben und sollen die Menge aller erreichbaren Nichtterminale, berechnen, d.h. $\{X \in V \mid \exists \alpha, \beta \in (\Sigma \cup V)^*. S \xrightarrow{*}_G \alpha X \beta\}$.

Implementierung

Machen Sie sich mit der Struktur der Klasse `ContextFreeGrammar` vertraut, insb. den Feldern und den Methoden `getProductionsByLhs` und `getProductionsByRhs`. Im Gegensatz zu den früheren Aufgaben hat jede Grammatik jetzt *explizite* Mengen für V und Σ und ein explizites Startsymbol S . Die Klasse baut auf den Klassen `Symbol`, `NonTerminal`, `Terminal` und `Production` auf. Die Klasse `Symbol` repräsentiert dabei ein Zeichen aus $\Sigma \cup V$, während die Unterklassen `Terminal` und `NonTerminal` jeweils Zeichen aus Σ bzw. V darstellen.

Vervollständigen Sie nun die Klasse `Reachability`, die für eine gegebene CFG die Menge der erreichbaren Nichtterminale berechnet.

Eingabe

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

Ausgabe

Alle erreichbaren Nichtterminale der Grammatik, jeweils eines pro Zeile, lexikographisch sortiert (Standard-Sortierung von Strings in Java).

Beispiele

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

Sample Input 1

```
Grammar
Nonterminals: A
Alphabet: a,b,c
Startsymbol: A
Productions:
A -> a | b | c
END
```

Sample Output 1

```
A
```

Sample Input 2

```
Grammar
Nonterminals: S,A,B
Alphabet: a,b
Startsymbol: S
Productions:
S -> | A B | A S B
A -> a
B -> b
END
```

Sample Output 2

```
A
B
S
```