Fragen, Antworten und Kommentare zur aktuellen Vorlesung

Das Video zur Lösung der Aufgabe 17 finden Sie unter: https://youtu.be/GC2uiSQuqEs . In 17a fehlt die Regel Start -> a und die untere Regel ist korrekt B->b.

Das Video zur Lösung der Aufgabe 18 finden Sie unter: https://youtu.be/TJr6yJwjmx4 .

Frage: Kettenregeln tragen ja nichts zum erzeugten Wort bei, kann man die nicht einfach weglassen?

Antwort: Nein, wie folgendes Beispiel zeigt: Start -> A A -> B B -> C C -> D D -> aa
Die erzeugte Sprache ist {aa}, durch weglassen der Kettenregeln wäre es die Sprache {}.

Frage: Ich habe Schwierigkeiten bei der Anwendung der semiformalen Algorithmen.

Antwort: Probieren Sie es zusätzlich mit einer der Praktikumsaufgaben aus und schreiben Sie Schritt für Schritt die Werte aller Variablen hin. Zentrales Ziel muss es aber sein die Idee des Algorithmus zu verstehen und so anwenden zu können. Z. B.:

Entfernung von ε-Regeln: Suche alle Nichtterminale die nach ableitbar ε sind und ersetze diese in den Anderen Regeln durch das leere Wort wodurch neue zusätzliche rechte Seiten der Regeln entstehen

Entfernung von Ketten-Regeln: Suche für alle Nichtterminale die Nichtterminale heraus, die als Kette ableitbar sind. Streich dann Ketten der Form A -> B und ergänze für alle Regeln B ->xyz eine Regel der Form A -> xyz. Dies entspricht anschaulich dem Überspringen von Kettenregeln.

Frage: Gibt es Regeln mit denen eine kontextfreie Grammatik erstellt werden kann?

Antwort: Feste klare Regeln gibt es nicht, wir werden später sehen warum („Sprachäquivalenz von kontextfreien Grammatiken ist nicht entscheidbar“). Generell haben die unendlichen kontextfreien Sprachen eine Art Wachstumspunkt, an denen sich eine Struktur immer wiederholt. Ein klassisches Beispiel ist die Sprache $\{a^n b^n \mid n > 0\}$. Zunächst ist es sehr sinnvoll, dass in jeder Regel die Anzahl der erzeugten a und b gleich ist (muss nicht sein, aber sehr sinnvoll). Dann ist zu erkennen, dass der Wachstumspunkt der Sprache zwischen a und b liegt, da hier immer neue a und b hinzukommen können. An diesem Wachstumspunkt muss dann ein Nichtterminal stehen, dass das Konzept umsetzt, hier Start -> a Start b | ab

Der Wachstumspunkt und Stellen davor können durch andere Sprachanteile „verunreinigt“ sein. Diese sind dann getrennt davon zu erstellen, z. B. $\{x^j a^n y^k b^n z^l \mid j,k,l,n > 0\}$, dabei haben j, k, l nichts mit n zu tun, sonst wäre die Sprache nicht mit einer kontextfreien Grammatik erzeugbar. Die Grammatik lautet

Start0 -> X xa Start bz Z

Start -> a Start b | Y

X -> X x | ε Y -> Y y | y Z -> Z z | ε

Das neue Start-Zeichen Start0 sorgt dafür, dass mindestens ein x, a, b, z im Ergebnis stehen, was mit „> 0" gefordert wird. Dass mindestens ein y im Ergebnis steht, wird später mit den Y-Regeln geklärt.

Eine weitere Möglichkeit sind nebeneinanderstehende Wachstumspunkte, die aber nichts miteinander zu tun haben. Dies deutet auf eine Konkatenation von kontextfreien Sprachen hin, die auch immer kontextfreie Sprachen garantiert. Beispiel: $\{a^nb^na^mb^m \mid n,m > 0\}$ mit Start0 -> Start Start mit den Regeln für Start aus der obersten Grammatik.

Eine letzte Möglichkeit ist es, dass kontextfreie Sprachen ineinander geschachtelt sind. Hier müssen erst die äußeren und dann die inneren Formen typischerweise mit einem neuen Nichtterminal abgeleitet werden. Beispiel:

$\{a^nc^md^mb^n \mid n,m > 0\}$ mit Start -> a Start b | a X b X -> c X d | cd.

Die Regeln geben nur eine grobe Orientierung. Abschließend wird das Beispiel mit der Sprache $\{a^nb^mc^{n+m} \mid n,m > 0\}$ betrachtet. Für jeweils ein c muss es ein a oder ein b geben. Der Wachstumspunkt liegt damit zwischen b und c. Weiterhin liegt eine Schachtelung mit äußeren a und inneren b vor, so dass erst a und dann mit einem weiteren Nichtterminal b erzeugt werden. Hier ist wieder die Randbedingung „>0" zu beachten. Die Regeln lauten

Start -> a Start c | a X c X -> b X c | bc

Frage: Wie geht das mit den Kettenregeln?

Antwort: Grob sind es zwei Schritte: Stellen Sie fest, welches Nichtterminal aus welchem abgeleitet werden kann (Algorithmus im Skript, geht auch anschaulich)

Start -> AB A -> ABBA | B B -> b | A

Als „spannende" Kettenregeln wird A -> B und B -> A gefunden.

Jetzt wird im zweiten Schritt für jede dieser Regeln die Kettenregel gelöscht und die rechte Seite durch alle aus dieser rechten Seite direkt ableitbaren rechten Seiten ersetzt, die keine Kettenregeln sind. Konkret

A -> B streichen gibt B -> b | A, aber das zweite ist eine Kettenregel, bleibt nur b als rechte Seite über, daraus folgt als Zwischenergebnis A -> ABBA | b (für ABBA interessiert sich hier niemand, da dies keine Kettenregel ist).

Nächster Schritt: B->A streichen. Die einzige Nicht-Kettenregel für A lautet A -> ABBA, so dass die Regel B -> b | ABBA am Ende lautet. Die vollständige Grammatik ist:

Start -> AB A -> ABBA | b B -> b | ABBA

Bitte beachten, wenn es Regeln der Art A -> B B -> C C -> D gibt müssten diese Ersetzungen für alle Paare (A,B), (A,C), (A,D), (B, C), (B, D), (C, D) gemacht werden; also gibt es eine Regel A -> B , wird diese gestrichen und alle rechten Seiten für B, C und D als neue Regeln ergänzt.

Frage: Wenn ich die Regel A -> ABBA in Chomsky-Normal-Form kann doch auch A->X1 X2 und dann X1 -> AB und X2-> BA nutzen?

Antwort: Das geht in diesem Fall auch und benötigt sogar weniger Nichtterminalzeichen, wobei der Ansatz nur sehr gut bei 2^n-Zeichen als Baumstruktur funktioniert. Passt der Wert nicht, kann der Baumansatz für die ersten Zeichen, aber mit dem Kettenansatz aus der Vorlesung verknüpft und so die Anzahl der Nichtterminalzeichen reduziert werden. Das wird aber hier nicht gefordert, da nur die Machbarkeit im Mittelpunkt steht. Beispiel: aus A -> ABCDEF wird mit der Optimierung X1 -> AB X2->CD X3->EF X4 -> X1 X2 und A -> X4 X3