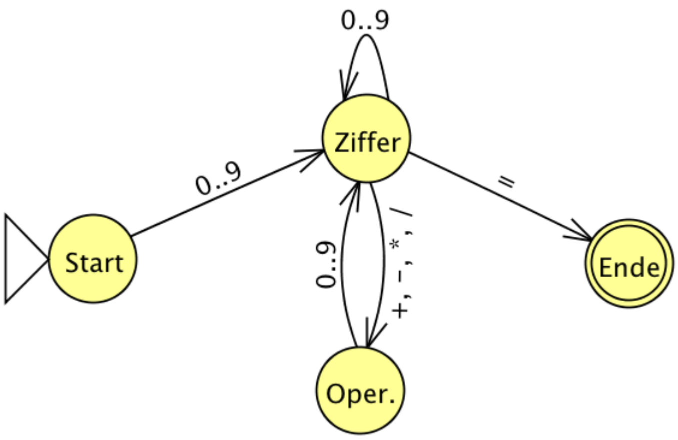
Projekt Theoretische Informatik

Fachliche Inhalte

Deterministische und nichtdeterministische Akzeptoren und Transduktoren.
Reguläre Grammatiken.

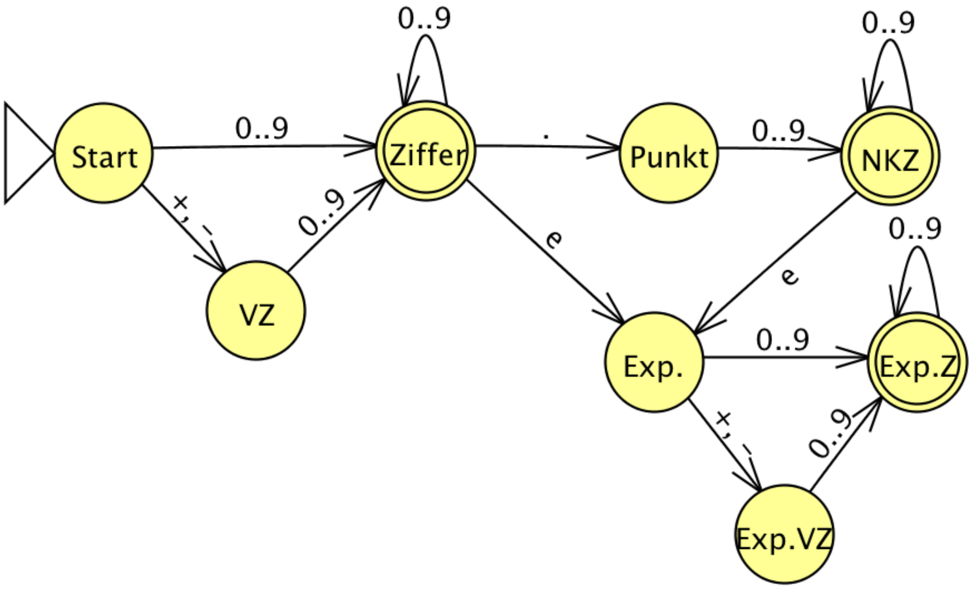
1. Akzeptor

Es soll die Eingabe eines Mini-Taschenrechners (nur Ziffern, Tasten für die vier Grundrechenarten und die Gleich-Taste) auf Korrektheit überprüft werden. Daraus lässt sich leicht folgender Akzeptor entwickeln:



Hiermit kann dann die Definition eines Akzeptors erarbeitet werden und die Möglichkeit, die Übergangsfunktion als Tabelle darzustellen beschrieben werden.

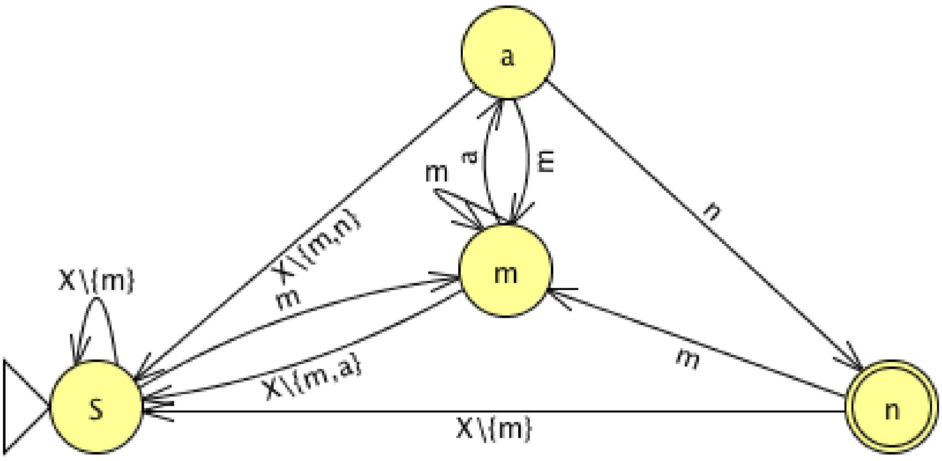
Nun sollten die Aufgaben 1-7 folgen, 5 und 7 können dabei weggelassen werden.
Bei den Aufgaben danken wir Herrn Rüdeger Baumann für die Erlaubnis, etliche Ideen aus seinem vergriffenen Schulbuch „Informatik für die Sekundarstufe II Band 2“ zu verwenden.



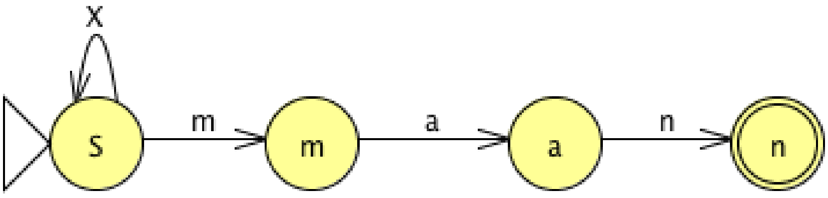
Dieser Akzeptor zu Aufgabe 1 kann schrittweise ergänzt werden und wird später noch benötigt.

Hinweis zu Aufgabe 4:
Hier kann man darüber sprechen, dass die Aufgabe d nicht sauber formuliert ist:
Was bedeutet „erweitern“ in diesem Zusammenhang?
Es fehlt sowohl in der Aufgabe wie in der Punktevergabe die Anforderung, dass der Automat Wörter ohne 007 oder 707 nicht akzeptiert.

Der „man“-Akzeptor aus Aufgabe 6 hat einen Graphen mit relativ komplizierten Übergängen:



An diesem Beispiel kann durch die verblüffende Vereinfachung gut der nichtdeterministische Akzeptor eingeführt werden:

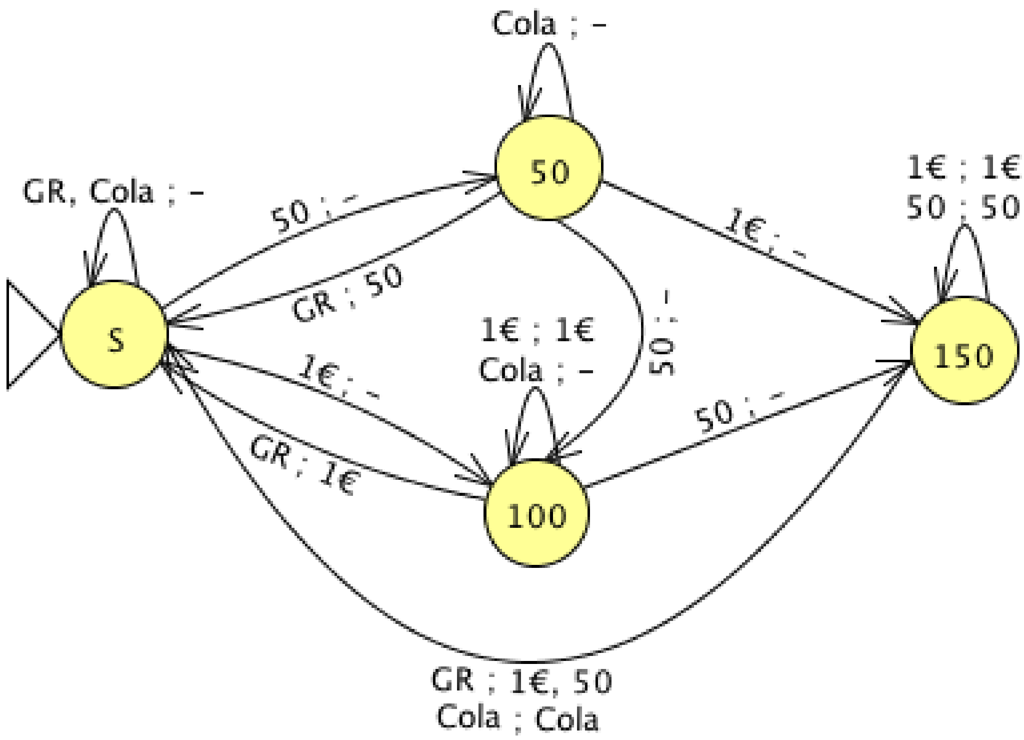


2. Transduktor

Der Transduktor wird als Akzeptor mit einer zusätzlichen Ausgabe entwickelt. Als Beispiel kann ein Cola-Automat dienen. Er erfüllt folgende Regeln:

- Eingaben sind ein 50 ¢-Stück, ein 1 €-Stück, eine Geldrückgabe- und eine Cola-Taste.
- Ausgaben sind ein 50 ¢-Stück, ein 1 €-Stück und eine Cola-Flasche.
- Die Cola-Flasche kostet 1,50 €.
- Überbezahltes Geld fällt durch den Automaten durch.

Man erhält folgenden Graphen des zugehörigen Transduktors:

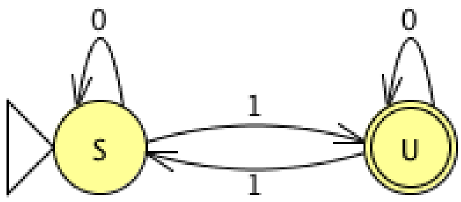


Der Graph sollte auch tabellarisch dargestellt werden.

Aufgabe 8 bietet ein weiteres Beispiel.

3. Reguläre Sprachen

Es kann z. B. aus einem Akzeptor für ungerade Parität die erste reguläre Grammatik entwickelt werden.



Man kommt so zu den Regeln: S → 0S | 1U | 1; U → 1S | 0U | 0.
Nun kann eine rechtsreguläre Grammatik definiert werden.

Die Konstruktion einer äquivalenten rechtsregulären Grammatik aus einem Akzeptor kann schematisch erfolgen.

Ebenso schematisch kann aus einer regulären Grammatik ein äquivalenter Akzeptor entwickelt werden.

Damit zeigt sich, dass reguläre Grammatiken und Akzeptoren dieselbe Sprachklasse beschreiben.

Die Aufgaben 9 - 12 festigen den Umgang mit regulären Sprachen.

Hinweis zu Aufgabe 11:
Die offizielle Modellösung dieser Abituraufgabe ist falsch. Sie erzeugt nur Wörter, die genau einmal 007 enthalten, etwa 0071007 lässt sich nicht erzeugen. Mit einem ähnlichen nichtdeterministischen Akzeptor wie bei Aufgabe 6 lässt sich eine sehr viel einfachere, dafür aber korrekte Grammatik konstruieren.

Am Beispiel der Sprache $L = \{ a^n b^n \mid n \in N \}$ können die Grenzen von regulären Sprachen gezeigt werden. Man kann die Annahme der Existenz eines zugehörigen Akzeptors zum Widerspruch führen. Wenn man für a und b Klammern setzt, bleibt auch diese Betrachtung im Themenrahmen der Beschreibung von Termen.

In Analogie lässt sich Aufgabe 13 betrachten.

Grammatiken

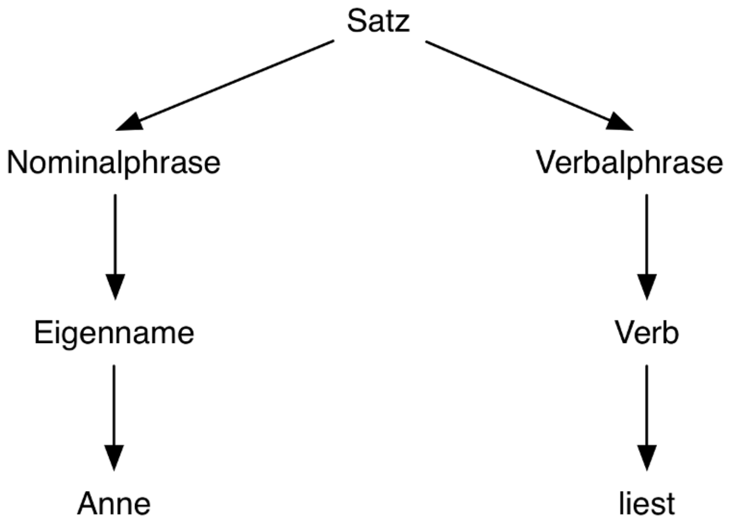
Aufgabe 16 zeigt, wie Chomsky als Linguist auf die Transformationsgrammatiken gekommen ist.

Am Beispiel dieser Aufgabe kann man auch die nötigen Schreibweisen für das Thema „Eindeutigkeit von Grammatiken“ der nächsten Aufgaben vorbereiten.

Hier etwa drei „verschiedene“ Ableitungen für „Anne liest“:
1. <Satz> → <NP> <VP> → <EN> <VP> → Anne <VP> → Anne <Verb> → Anne liest
2. <Satz> → <NP> <VP> → <EN> <Verb> → Anne liest
3. <Satz> → <NP> <VP> → <NP> <Verb> → <NP> liest → <EN> liest → Anne liest

Man sieht, dass diese Ableitungen strukturell identisch sind. Eine Möglichkeit, das zu zeigen, ist die Einschränkung auf die Linksableitung, bei der immer das am weitesten links stehende Nichtterminalzeichen ersetzt wird, also die erste Ableitung im Beispiel.

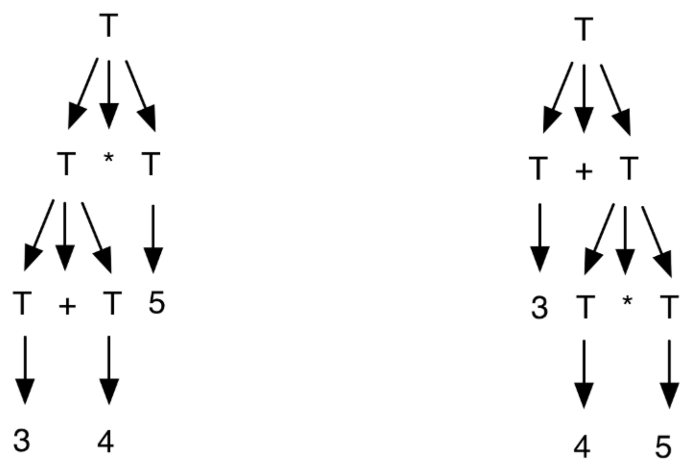
Eine andere Möglichkeit ist das Syntaxdiagramm, das für alle drei Ableitungen gleich aussieht, weil es sich um dieselbe Struktur handelt:



Aufgabe 17 führt nun wieder zum Term-Thema und mit Aufgabe 18 zur Eindeutigkeit von Grammatiken.

Eine Lösung von 17 hat die Produktionen: T → T+T | T-T | T*T | T/T | (T) | Zahl

Zusatz: Mehrdeutige Grammatiken
Tatsächlich aber findet man für den Term 3+4*5 zwei verschiedene Syntaxbäume:



Der linke Baum arbeitet den Term von links nach rechts ab und liefert 35, der rechte Term wäre für „Punkt- vor Strichrechnung“ korrekt und liefert 23.
Eine Grammatik, bei der es ein Wort gibt, das man mit zwei verschiedenen Syntaxbäumen ableiten kann, heißt mehrdeutig.
Da die Grammatik mehrdeutig ist, ist die scheinbar so praktische Grammatik für unsere Zwecke unbrauchbar. Aufgabe 21 motiviert zu einer eindeutigen Grammatik für Terme.

Aufgabe 19 zeigt, wie die natürliche Sprache mehrdeutig ist.

Ein Syntaxbaum gibt an, wie Anne mit einem Bild in der Hand das Zimmer betritt. „mit dem Bild" beschreibt hier die Art, wie sie das Zimmer betritt. Es ist hier eine adverbiale Bestimmung.

Ein anderer Syntaxbaum gibt an, wie Anne das Zimmer betritt, in dem ein Bild hängt. „mit dem Bild" beschreibt hier das Substantiv Zimmer genauer, ist hier ein Präpositionalattribut.

Aufgabe 20 zeigt die Mehrdeutigkeit an einem Beispiel aus der Informatik. Hier kann sich das ELSE mal auf das erste, mal auf das zweite IF beziehen.

Zusatz: 5. Monotone Sprachen und die Chomsky-Hierarchie

Am Beispiel der Sprache $L = \{ a^n b^n c^n \mid n \in N \}$ können die Grenzen von kontextfreien Sprachen gezeigt werden. (Der Beweis wäre aber hier zu aufwändig.)

Eine funktionsfähige Grammatik lässt sich leicht konstruieren.
Analog zu der Grammatik für $L = \{ a^n b^n \mid n \in N \}$ beginnt man mit den Produktionen:
S → aSbc | abc

Versucht man nun aaabbbccc abzuleiten, erhält man:
S → aSbc → aaSbcbc→ aaabcbcbc
Da die b und c die falsche Reihenfolge haben, muss man hier anders vorgehen. Man ersetzt das Terminalzeichen b durch das Nichtterminalzeichen B:
S → aSBc | abc
Hier also S → aSBc → aaSBcBc → aaabcBcBc
Nun benötigt man eine Regel, mit der man B im Kontext hinter einem c mit diesem vertauscht: cB → Bc
Damit erhält man → aaabBccBc → aaabBcBcc → aaabBBccc.
Nun darf nicht etwa die Regel B → b eingeführt werden, weil dann auch aabcbc abgeleitet werden könnte. Wieder benötigt man den Kontext: bB → bb.
Und zuletzt ergibt sich → aaabbBccc → aaabbbccc.

An diesem Beispiel können monotone und kontextsensitive Grammatiken eingeführt werden.

Wiederholung

Aufgabe 22 schließt das Gebiet der Akzeptoren und regulären Sprachen wiederholend ab.