Prof. Dr. Knut Reinert
Enrico Siragusa
Sascha Meiers
Christoph Hartmann

Institut für Informatik
AG Algorithmische Bioinformatik

Algorithmen und Datenstrukturen in der Bioinformatik
Zwölftes Übungsblatt WS 11/12

Abgabe Montag, 23.01.2012, 15:00 Uhr

Name: Übungsgruppe: A☐ B☐ C☐

Matrikelnummer:

Niveau I

Aufgabe 1: Reguläre Sprachen

Gegeben ist die folgende rechts-reguläre Grammatik $G = (\{S, X\}, \{a, b\}, P, S)$ mit den Produktionsregeln P:

I $S \rightarrow a$

II $S \rightarrow aX$

III $X \rightarrow a$

IV $X \rightarrow aX$

V $X \rightarrow bX$

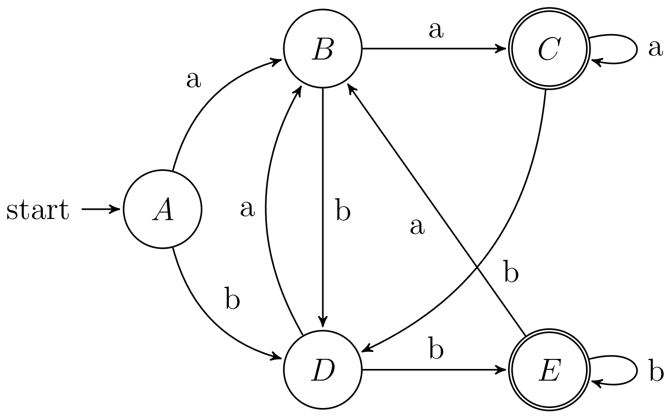
a) Wandeln Sie die Grammatik in einen nicht-deterministischen Automaten um. Dazu benötigen Sie einen Zustand für jede Variable, sowie einen weiteren für die Endzustände.

b) Wandeln Sie ihren *nfa* in einen *dfa* um. Die Zustandsmenge ihres neuen Automaten ist die Potenzmenge der Zustände des vorherigen, wobei Sie die vom Startknoten nicht erreichbaren Zustände weglassen können.

c) Geben Sie zuletzt einen regulären Ausdruck an, der diese Sprache beschreibt.

Aufgabe 2: Wahrscheinlichkeit

Gegeben ist ein *dfa* über dem Alphabet $\Sigma = \{a, b\}$.



Wie hoch ist die Wahrscheinlichkeit, dass ein zufälliges Wort aus 3 Buchstaben vom Automaten akzeptiert wird, wenn

a) die Wahrscheinlichkeit für a, $P(a) = \frac{1}{3}$ und die für b, $P(b) = \frac{2}{3}$ ist?

b) die Wahrscheinlichkeiten von a und b nicht mehr unabhängig sind, sondern in $\frac{3}{4}$ der Fälle auf ein a ein a folgt und auf ein b ein b?

Niveau II

Aufgabe 3: Rechts- und links-reguläre Grammatiken

Wie wir nun gesehen haben, lassen sich rechts-reguläre Grammatiken in *dfa* umwandeln. Die Beweise dazu finden Sie im Skript. Gilt dies jedoch auch für links-regulären Grammatiken? Überlegen Sie sich dazu folgendes:

a) Welche Sprache rezeugt die Grammatik, wenn Sie alle Produktionsregeln der Form $A \to Ba$ in die rechts-reguläre Form umdrehen (zu $A \to aB$)?

b) Wie kann man die gleiche Transformation auf Ebene der *nfa*s rückgängig machen?

Fassen Sie kurz die Beweisidee zusammen.

Aufgabe 3: L-Systeme (freiwillige, aber interessante Zusatzaufgabe)

Lindenmayer-Systeme sind Termersetzungssysteme und funktionieren analog zu formalen Grammatiken, mit einigen Unterschieden:

I Die Unterteilung zwischen Variablen und Terminalen entfällt.

II Der Umgang mit Produktionsregeln ist ein anderer: In jedem Schritt geht man das Wort vom ersten bis zum letzten Symbol durch und wendet für jedes Symbol die Produktionsregel an.

III Für jedes Symbol existiert genau eine Produktionsregel der Form $A \to x$. Ist diese nicht explizit gelistet, geht man von der Identitätsregel $A \to A$ aus.

Ein einfaches L-System ist gegeben durch die zwei folgenden Produktionsregeln:

I $A \to A - B$

II $B \to B + A$

Das Startwort lautet A.

a) Führen sie das System 5 Runden lang aus und notieren Sie die nach jeder Runde entstehenden Worte.

b) Interpretieren Sie jedes Wort als Turtle-Grafik-Zeichenanweisung (`http://en.wikipedia.org/wiki/Turtle_graphics`). Gehen Sie dabei das Wort von vorne bis zum Ende durch. Für A und B zeichnen Sie einen Strich, für – drehen sie die Zeichenrichtung um 90° gegen den Uhrzeigersinn, für + um 90° im Uhrzeigersinn. Zeichnen Sie die ersten 5 Iterationen, am besten auf Karopapier. Was fällt auf?

c) Sind die von L-Systemen erzeugten Sprachen regulär? Kurze Begründung.