

Praktikum I – Reguläre Sprachen, DEAs

GTI SoSe 2015 Prof. A. Siebert

Für die meisten Probleme rund um Automaten und Grammatiken verwenden wir das frei verfügbare Programm JFLAP (<http://www.jflap.org>).

Sie sollen sich mit JFLAP vertraut machen – aber zunächst immer versuchen, die Lösungen ohne Unterstützung von JFLAP zu finden. In der Klausur werden Sie auf JFLAP nicht zugreifen können.

Stellen Sie bei JFLAP unter Preferences – Set the Empty String Character letzteren auf Epsilon (JFLAP-Default: λ statt ε).

Zeichnen Sie, soweit möglich, die Automaten ohne Überschneidungen.

Aufgabe 1.

Konstruieren Sie für die folgenden Regulären Ausdrücke über $\Sigma = \{f, g\}$ einen äquivalenten DEA. Verwenden Sie, falls notwendig, einen Fangzustand, so dass jeder Übergang wohl definiert ist.

- (a) f^*gg
- (b) $(gff)^*$
- (c) $(f+g)^*$
- (d) $f^* + g$

Aufgabe 2.

- a. Entwerfen Sie einen möglichst kleinen DEA für die Sprache $\Lambda = \{\text{egal, regal, real, ideal}\}$.
- b. Geben Sie für Λ einen äquivalenten, möglichst kurzen regulären Ausdruck an.

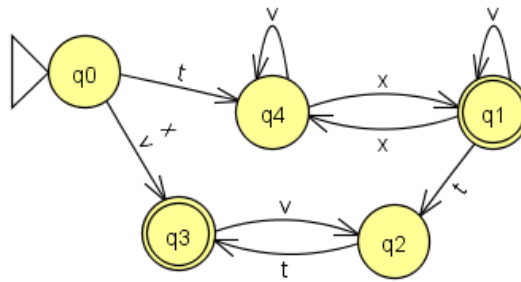
Aufgabe 3. (Klausuraufgabe Februar 2015)

Entwerfen Sie einen DEA für die Sprache $\Lambda = (ab)^*m^* + ab^*m$.

Aufgabe 4.

Geben Sie für den folgenden DEA über $\Sigma = \{t, v, x\}$ einen äquivalenten regulären Ausdruck an. Lösungsmethode ist hier "scharfes Hinschauen".

Einen (eher mühsamen) Algorithmus zur Umwandlung von DEAs in R.A.s finden Sie im Skript – Sie können sich schnell davon überzeugen, dass dieser (per Hand) auf Dauer kein Spaß ist.



Aufgabe 5. (Klausuraufgabe Februar 2015)

a. Sei Λ die Menge aller Worte über $\Sigma = \{p, s, t\}$, deren Länge durch 3 teilbar ist und die mit t aufhören, aber nicht mit t anfangen. Zu Λ gehören also z.B. pst , stt , $ssptpt$, aber nicht ε , pt , tst , ptp , $psst$.

b. Geben Sie einen zum DEA äquivalenten regulären Ausdruck an.

Testen Sie in JFLAP Ihren regulären Ausdruck, indem Sie ihn in einen minimalen DEA umwandeln (**Convert to NFA**, **Convert to DFA**, **Minimize DFA**). Das Ergebnis sollte (bis auf die Bezeichnung der Zustände) derselbe DEA sein wie in a.

Aufgabe 6. (Klausuraufgabe Juli 2013)

Geben Sie einen DEA an für die Suche nach dem Wort $w=auaau$, mit $\Sigma = \{a, u\}$. Übergänge zum Startzustand dürfen weggelassen werden.

Aufgabe 7.

a. Studieren und vervollständigen Sie das gegebene Programm `DEA.java`, welches das Wortproblem für DEAs lösen soll.

Die Eingabe des Programms ist also ein Wort über Σ . Die Ausgabe des Programms ist die Folge der durchlaufenen Zustände und eine Meldung, ob das Wort akzeptiert oder zurück gewiesen wird.

Die Ausgabe für den in `DEA.java` gegebenen DEA (Skript S. 16) sollte so aussehen

```
> java DEA ababa
z0 ==> z1 ==> z0 ==> z1 ==> z0 ==> z1
ababa accepted.
```

```
> java DEA abbb
z0 ==> z1 ==> z0 ==> z2 ==> z2
abbb rejected.
```

Sie müssen im Programm nur zwei kurze Code-Stücke ergänzen:

- Eine Schleife in `main()` über das Eingabewort `w`. Bei jedem Zeichen von `w` wird in die Übergangstabelle gegriffen, um den Folgezustand zu bestimmen. Der Folgezustand und ein `'==>'` werden ausgegeben.
- Die boolesche Methode `contained()`, die prüft, ob ein Zustand in einem Feld von Zuständen enthalten ist.

b. Testen Sie Ihr Programm mit dem DEA aus Aufgabe 4.

Z.B. sollten die Worte `txxvxx` und `tvxvtt` akzeptiert werden, nicht jedoch die Worte `vxx` oder `tvxx`.

Hierzu sollte nur der zu Beginn von `main()` explizit definierte DEA ausgetauscht werden. Der Rest des Programms muss unangetastet bleiben.