PS Software Engineering II

Aufgabenblock 4

Ausgabedatum: 19.04.2007
Abgabedatum: 26.04.2007

1 DFA

Sei das Alphabet $\Sigma = \{0, 1\}$ gegeben. Geben Sie Zustandsdiagramme und Simulatoreingaben für die DFAs an, welche folgende Sprachen akzeptieren:

(a) $\{\varepsilon\}$.

(b) $\{0, 1\}$.

(c) $\{\alpha | \alpha$ enthält den Substring 1010$\}$

(d) $\{\alpha | \alpha$ enthält nicht den Substring 1010$\}$

(e) $\{\alpha | \alpha$ beginnt mit 0 und $|\alpha| = 2k$ oder α beginnt mit 1 und $|\alpha| = 2k + 1$, für beliebiges $k \in \mathbb{N}_0\}$

2 NFA Simulator

Ein nichtdeterministischer endlicher Automat (NFA) ist definiert als ein Quintupel $(Z, \Sigma, \delta, z_1, F)$:

- Z ist eine endliche Menge von Zuständen
- Σ ist eine endliche Menge von Eingabesymbolen
- δ ist die Übergangsfunktion, d.h. eine Abbildung $Z \times \Sigma_\varepsilon \to \mathcal{P}(Z)$
  ($\mathcal{P}(Z)$ steht für die Potenzmenge von Z, $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$)
- $z_1 \in Z$ ist der Anfangszustand
- $F \subseteq Z$ ist eine Menge von Endzuständen

Die Unterschiede zwischen einem DFA und einem NFA sind leicht ersichtlich:

- Jeder Zustand eines DFA hat genau einen Übergang (d.h. einen ausgehenden Pfeil im Zustandsdiagramm) für jedes Element des Alphabets. Bei einem NFA ist das nicht notwendigerweise der Fall.
- Bei einem NFA sind zusätzlich zu Übergängen die mit einem Symbol des Alphabets bezeichnet sind sogenannte ε-Übergänge erlaubt. D.h. diese Übergänge werden auch ausgeführt wenn kein Eingabesymbol gelesen wird.

Die Abarbeitung eines NFA kann als die parallele Abarbeitung aller möglichen Wege gesehen werden. Sei ein NFA beispielsweise in einem Zustand z_1, das anstehende Eingabesymbol sei a und für dieses Symbol definiert die Übergangsfunkion mehrere Zielzustände (d.h. $|\delta(z_1, a)| > 1$). Nach dem Lesen des Eingabesymbols teilt sich der Automat in Kopien von sich selbst, um allen möglichen Wegen parallel folgen zu können. Diese Automaten arbeiten weiter wie der ursprüngliche Automat.
Sollte sich einer dieser Automaten wiederum in einer Konfiguration mit mehreren Zielzuständen befinden, teilt er sich ebenfalls auf.
Wenn sich ein Automat in einer Konfiguration $(z_2, b\alpha)$ befindet, in der keine Folgezustände möglich sind (d.h. $|\delta(z_2, b)| = 0$), dann terminiert dieser Automat.

Wenn sich einer dieser Automaten nach dem Lesen aller Eingabesymbole in einem gültigen Endzustand befindet, dann gilt die Eingabekette als akzeptiert.
ε-Übergänge werden analog behandelt: Sei ein Automat in einem Zustand z_3 und $|\delta(z_3, \varepsilon)| > 0$. Ohne dass das nächste Eingabesymbol gelesen wird splittet sich der Automat so auf, dass sowohl der ursprüngliche Automat erhalten bleibt, aber auch ein neuer Automat für jeden Zielzustand entsteht. Danach wird die Abarbeitung wie gehabt fortgesetzt.

(a) Entwickeln Sie einen Simulator für NFAs der eine Eingabedatei liest und die Abarbeitung des Automaten mit allen sich daraus ergebenden Konfigurationen darstellen kann. Das Format für eine Eingabedatei ist folgendem Beispiel[1] zu entnehmen:

```
# Configuration file for Nondeterministic Finite Automaton Simulator
#-----------------------------------------------------------------
#
# Q: States (comma separated string)
Q=a,b,c,d,e,f
#
# S: Alphabet (comma separated string)
S=0,1
#
# d: Transition Table (comma separated string with elements "p:s->q,"
#                      where p,q are states and s is a symbol)
#    Note: the string "$" is considered to be epsilon
d=a:0->e, a:$->c, a:1->b, b:0->e, b:1->c, c:0->d, c:1->c, d:0->d, d:1->d, e:0->e, e:1->f,
  f:0->e, f:1->c
#
# t: Start State (string)
t=a
#
# f: Accept States (comma separated string)
F=b,c,e,f
#
# I: Input Strings (comma separated string of symbols;
#                   different test string are separated by "|" )
I= |  | 0 | 1 | 0,1 | 1,0 | 0,0,0 | 0,0,1 | 0,1,0 | 0,1,1 | 1,0,0 | 1,0,1 | 1,1,0 | 1,1,1 |
   1,0,1,0,1,0 | 0,1,0,1,1,0,1,0,1 | 1,1,1,0,0
#EOF#
```

Anmerkungen zur gegebenen Eingabedatei:

(i) Die erste Kette von Eingabesymbolen entspricht der leeren Kette ε.

(ii) Die Definition für die Übergangsfunktion ist hier auf zwei Zeilen aufgeteilt; für Ihren Simulator können Sie annehmen, dass für die Definition lediglich eine Zeile zugelassen ist. Gleiches gilt für die Definition der Eingabesymbole.

Argumentieren Sie Ihre Designentscheidungen, vermeiden Sie unnötige Komplexität und legen Sie Wert auf die Eleganz der Lösung.

(b) Geben Sie den Zustandsgraphen an, welcher dem NFA der gegebenen Eingabedatei entspricht.

3 NFA

Sei das Alphabet $\Sigma = \{0, 1\}$ gegeben. Geben Sie Zustandsdiagramme und Simulatoreingaben für die NFAs an, welche folgende Sprachen akzeptieren:

[1] Die Datei steht auch auf der Website des Proseminars zur Verfügung.

(a) $\{\varepsilon\}$. Benutzen Sie lediglich einen Zustand.

(b) $\{0, 1\}$. Benutzen Sie lediglich zwei Zustände.

(c) $\{\alpha | \alpha$ enthält den Substring 1010$\}$.

(d) $\{\alpha | \alpha$ enthält eine gerade Anzahl von 0*en* oder die Anzahl der 1*en* in α ist durch 3 ohne Rest teilbar $\}$. (Anmerkung: 0 gilt auch als gerade Zahl.)