Theoretische Informatik 2

Tutorium #4 9.5.2002

(Fabian Wleklinski)

Literatur

- Erk und Priese: „Theoretische Informatik“
  - Springer
- Hopcroft und Ullman: „Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie“
  - Addison-Wesley
- freier Parser-Generator:
  - http://www.devincook.com/GOLDParser/

Aufgabe 4.1 a), b)

- a) Beweis ist falsch!
  - Nehme einen NFA A, so dass für ein w:
    - $\delta(q_0,w) = \{p,q,\ldots\}$ mit
      - $p \in F$,
      - $q \notin F$
    - $\Rightarrow$ **$w \in T(A)$**
  - Konstruiere NFA A‘ gemäß Skript S. 44:
    - $F' := Q \setminus F$
    - $\Rightarrow p \notin F, q \in F$
    - $\Rightarrow$ **$w \in T(A')$ (!)**
  - $T(A') \neq \neg T(A)$!!!
  - Korrektur:
    - A ist DFA statt NFA!
- b) Der Satz ist richtig!
  - Konstruiere DFA A für L:
    - A := $(Q, \Sigma, \delta, q_0, F)$
    - T(A)=L
  - Konstruiere NFA B:
    - B := $(Q', \Sigma, \delta', q_{-1}, q_0)$
    - $Q' := Q \cup \{q_{-1}\}$
    - $\delta' : Q \times \Sigma \rightarrow P(Q)$
    - neuer Anfangszustand q_{-1}!

Aufgabe 4.1 b)

- Kehre Richtungen aller Kanten um!
  - $\forall p, q \in Q, a \in \Sigma$: $\delta(p,a)=q \Leftrightarrow \delta'(q,a) \ni p$
- ε-Übergänge von q_{-1} zu allen in A akzeptierenden Zuständen:
  - $\delta'(q_{-1},\varepsilon) := \{ f \mid f \in F \}$
- NFA B akzeptiert L^R!
  - Beweis für $L^R \subseteq L(B)$:
    - „es fehlen keine Wörter“
    - $\forall w \in T(A)$: es gibt in B einen Pfad für w^R von q_{-1} nach q_0
  - Beweis für $L^R \supseteq L(B)$:
    - „nicht zuviele Wörter“
    - $\forall w \in T(B)$: es gibt in A einen Pfad für w^R von q_0 zu einem $f \in F$
  - ◊
- $\Rightarrow$ reg. Sprachen sind abgeschlossen gegen Spiegelung! („L^R“)

Aufgabe 4.1 c)

- Der Satz ist richtig!
  - konstruiere DFA für L
    - T(A) = L
  - entferne unerreichbare Zustände
  - füge neuen Startzustand s hinzu
  - $\Rightarrow$ reg. Sprachen sind abgeschlossen gegen Suffix-Bildung!

Pumping-Lemma (REG)

- Grenze von DFAs?
  - d.h. Grenze von regulären Sprachen/ Ausdrücken!
- Wann ist eine Sprache nicht regulär? Beweis?
  1. Nerode-Index *oder:*
  2. Pumping-Lemma

- Pumping-Lemma:
  - L reguläre Sprache:
    $\exists\, n\ \forall\, z \in L$:
    $|z| \geq n$:
    - $\mathbf{z=uvw}$,
    - $\mathbf{1} \leq |\mathbf{v}|$ (< n) und
    - $\mathbf{uv^iw} \in \mathbf{L}$ $\quad(\forall i \geq 0)$
- Anschaulichkeit:
  - endliche Automaten
  - reguläre Ausdrücke
  - Sprache unendlich => Wiederholungen!

Pumping-Lemma (Beispiel 1)

- L_1:={ a^iba^i | i∈N }
  - Vermutung: L_1 ist **nicht** regulär
  - Beweis d. Widerspruch
  - Widerspruchsannahme: „L_1 ist regulär“
  - Pumping-Lemma sagt aus, dass es ein n geben muss, so daß ... (S. 36)!
  - Es gibt beliebig lange Wörter in L_1!
  - Betrachte das Wort $z=a^nba^n$ aus L_1
    - n ist nach wie vor die Konstante des Pumping-Lemma!
  - Wie sieht die Zerlegung $a^nba^n=uvw$ konkret aus?
    - 3 Möglichkeiten: je nachdem, wo sich das b befindet!

Pumping-Lemma (Beispiel 1)

- #1: „b ganz hinten“
  - Zerlegung in u, v, w:
    - $u=a^k$ $k \geq 0$
    - $v=a^j$ $j > 0$
    - $w=a^iba^n$ $i \geq 0$
      $k+j+i = n$
  - Pumpe das Wort z einmal auf
    - setze j auf „2“
  - erhalte das neue Wort $z' = uv^2w = a^ka^{2\cdot j}a^iba^n = a^{k+2\cdot j+i}ba^n = a^{n+j}ba^n$
  - da j>0: z‘ nicht in L_1!
- Nach Pumping-Lemma müsste z‘ in L_1 sein!
- Konsequenz aus dem WIDERSPRUCH:
  - Findet man mit Möglichkeiten #2 und #3 keine „bessere“ Zerlegung, **dann** war die Widerspruchs-annahme falsch, und L_1 kann nicht regulär sein!

Pumping-Lemma (Beispiel 1)

- #2: „b ganz vorne“
  - Zerlegung in u, v, w:
    - $u=a^n ba^i$ $i \geq 0$
    - $v=a^j$ $j > 0$
    - $w=a^k$ $k \geq 0$
      $k+j+i = n$
  - Analog zu #1 !
- #3: „b in der Mitte“
  - Zerlegung in u, v, w:
    - $u=a^k$ $k \geq 0$
    - $v=a^j ba^i$ $i,j \geq 0$
    - $w=a^l$ $l \geq 0$
    - $k+j=i+l=n$
  - Pumpe abermals das Wort z einmal auf
    - setze j auf „2“
  - erhalte das neue Wort $z' = uv^2w = a^k a^j b a^i a^j b a^i a^l$
  - da #b‘s > 1: z‘ nicht in L_1!
- Also: Sprache lässt sich nicht „pumpen“ => kann nicht regulär sein!
- ◊

Pumping-Lemma (Beispiel 2)

- L_2:={ a^p | p Primzahl }
  - Vermutung: L_2 ist **nicht** regulär
  - Beweis d. Widerspruch mit Pumping-Lemma
  - Es gibt beliebig lange Wörter in L_2!
  - Nehme die kleinste Primzahl $p \geq n$
  - betrachte $z=a^p$ aus L_1
  - zerlege a^p in uvw
    - $u=a^i$ $(i \geq 0)$
    - $v=a^j$ $(0<j<n)$
    - $w=a^k$ $(k \geq 0)$
      $i+j+k=p$
  - jedes uv^iw muss in L_2 liegen! (für jedes $i \geq 0$)
  - Fall 1: $i+k > 1$
    - pumpe (i+k) mal
    - $z'=uv^{i+k}w = a^i a^{j(i+k)} a^k = = a^{i+j(i+k)+k} = a^{(1+j)(i+k)}$
    - z‘ muss in L_2 liegen, aber: (1+j)(i+k) kann keine Primzahl sein!

Pumping-Lemma (Beispiel 2)

- Fall 2: i+k = 1
  - es muss j=p-1 sein!
  - o.B.d.A. sei i=0 und k=1
  - pumpe (j+2) mal
  - $z' = uv^{j+2}w = a^{j(j+2)}a =$ $= a^{j(j+2)+1} = a^{(j+1)(j+1)}$
  - z‘ muss in L_2 liegen, aber: (j+1)(j+1) kann keine Primzahl sein!
- L_2 ist nicht regulär!

- Weitere Beispiele siehe S. 37f

Reguläre Sprachen

- Entscheidbare Probleme für reguläre Sprachen:
  - L leer?
  - L endlich?
  - $L_1 \cap L_2$ leer?
  - $L_1 = L_2$?
- Alle endlichen Sprachen sind regulär!
- typisches Praxisbeispiel: „Getränkeautomat“

- Reguläre Sprachen
  - „rationale Sprachen“
  - „Typ-3-Sprache“
  - Sprachen der regulären Ausdrücke
  - Sprachen der DFAs und NFAs (etc.)
  - Sprachen der rechts- **oder** linkslinearen Grammatiken

Reguläre Sprachen

- Kriterien:
  - Automat (DFA, NFA, ...)
  - Nerode-Index
  - Pumping-Lemma
  - reg. Ausdruck
  - reg. Grammatik
- Abschlußeigenschaften:
  - $\cup$, $\cap$, *, $\bullet$, $\neg$
  - R (Spiegelung)
  - / (Quotient)
  - hom(), ε-frei hom(), inverser hom()

Pumping-Lemma (CFL)

- Pumping-Lemma:
  - L kontextfreie Sprache
    - also darf L auch eine reguläre Sprache sein!
  - $\exists\, n\ \forall\, z \in L$: $|z| \geq n$:
    - $\mathbf{z=uvwxy}$,
    - $|\mathbf{vwx}| < n$,
    - $|\mathbf{vx}| \neq \varepsilon$ und
    - $\mathbf{uv^iwx^iy} \in \mathbf{L}$ $(\forall i \geq 0)$
  - Nachteil:
    - Jede CFL erfüllt das PL – aber nicht jede Nicht-CFL erfüllt es nicht!
- Odgen‘s Lemma
  - „Verschärfung“ des Pumping-Lemma
    - Alle nicht-CFLs erfüllen das OL!
    - Siehe Skript S. 67
  - Es werden nur noch Teilfolgen der Wörter betrachtet
    - Pumping-Lemma ist der „Sonderfall“, dass man alle Buchstaben betrachtet!

Pumping-Lemma (Beispiel)

- L_{abc}:={ $a^i b^i c^i$ | i∈N }
  - Vermutung: L_{abc} ist **nicht** kontextfrei
  - Beweis d. Widerspruch
  - Widerspruchsannahme: „L_{abc} ist kontextfrei“
  - Pumping-Lemma sagt aus, dass es ein n geben muss, so daß ... (S. 65)!
  - Es gibt beliebig lange Wörter in L_{abc}!
  - Betrachte das Wort $z=a^n b^n c^n$ aus L_{abc}
    - n ist nach wie vor die Konstante des Pumping-Lemma!
  - Wie sieht die Zerlegung $a^n b^n c^n$=uvwxy konkret aus?
    - vor Allem v und x sind wichtig, da sie „gepumpt“ werden!

Pumping-Lemma (Beispiel)

- 2 Möglichkeiten
- #1: **v** und **x** bestehen jeweils nur aus einer Art von Symbolen
  - uv^2wx^2y hat ungleich viele a, b oder c!
  - WIDERSPRUCH!!!
- #2: **v** und/oder **x** bestehen aus mehr als einer Symbolart
  - uv^2wx^2y hat nicht mehr die Form a...b...c !
  - WIDERSPRUCH!!!
- L_{abc} ist nicht regulär!

Kontextfreie Sprachen

- Kriterien:
  - Automat (PDA, ...)
  - Pumping-Lemma
  - kontextfreie Grammatik
  - (Nerode-Index)
- Abschlußeigenschaften:
  - $\cup$, *, $\bullet$
  - R (Spiegelung)
  - $\cap_{REG}$ (Schnitt mit REG)
  - $/_{REG}$ (Quotient mit REG)
  - hom(), ε-frei hom(), inverser hom()