Motivation – Englisch

Mary	and John	are		
a woman	and a man,	respectively.		
Mary,	John	and William	are	
a woman,	a man	and a man,	respectively.	
Mary,	John,	William	and Jenny	are
a woman,	a man,	a man	and a woman,	respectively.

$h(and) = h(are) = h(a) = h(respectively) = h(,) = h(.) = \lambda$
$h(woman) = a$, $h(X) = a$ für jeden Frauennamen X,
$h(man) = b$, $h(Y) = b$ für jeden Männernamen Y,

X – Menge der Vornamen, $Y = \{a\ woman,\ ,\ a\ man,\}$
$R = \{x, \mid x \in X\}\{x, \mid x \in X\}^+\{and\}X\{are\}YY^+\{and\}Y\{respectively.\}$

$h(Englisch \cap R) = \{ww \mid w \in \{a,b\}^+, |w| \geq 3\}$

Motivation – Schweizer Dialekt des Deutschen

Jan säit das mer em Hans hälfed.
(Jan sagt, dass wir Hans helfen.)

Jan säit das mer em Hans es Huus hälfed aastriche.
(Jan sagt, dass wir Hans helfen, das Haus zu streichen.)

Jan säit das mer d'chind em Hans es Huus lönd hälfed aastriche.
(Jan sagt, dass wir den Kindern erlauben, Hans zu helfen, das Haus zu streichen.)

$h(Schweizerdeutsch \cap R') = \{ww \mid w \in \{a,b\}^+\}$
für einen passenden Homomorphismus h und eine passende reguläre Sprache R'

Motivation – Programmiersprachen

```
begin  integer x;
       y := 1
end
```

mit $x, y \in \{a, b\}^*$

Ausdrücke dieser Form bilden eine reguläre Menge R

ALGOL-Programme dieser Form sind genau dann korrekt, wenn deklarierte Variable x mit benutzter Variabler y übereinstimmt

$h(ALGOL \cap R) = \{xx \mid x \in \{a, b\}^*\}$
für einen geeigneten Homomorphismus h

Grammatiken mit Auswahlkontext I

Definition:

Eine Grammatik mit Auswahlkontext ist ein Quadrupel $G = (N, T, P, S)$, wobei

— N, T, S wie bei einer Regelgrammatik spezifiziert sind,

— P eine endliche Menge von Tripeln $p = (r_p, E_p, F_p)$ ist, wobei jeweils

- $r_p = A_p \rightarrow w_p$ eine kontextfreie Regel mit $w_p \neq \lambda$ ist, und
- E_p und F_p Teilmengen von N mit $E_p \cap F_p = \emptyset$ sind.

Grammatiken mit Auswahlkontext II

Definition: Es seien $G = (N, T, P, S)$ eine Grammatik mit Auswahlkontext, x und y nichtleere Wörter über $N \cup T$. Wir sagen, dass y aus x durch Anwendung von $(A \to w, E, F)$ erzeugt wird, wenn folgende Bedingungen erfüllt sind:
— $x = uAv$, $y = uwv$ (kontextfreie Ersetzung)
— jedes Symbol aus E kommt in uv vor,
— kein Symbol aus F kommt in uv vor.

Bezeichnung: $x \Longrightarrow_p y$
$x \Longrightarrow_G y$ genau dann, wenn $x \Longrightarrow_p y$ für eine Regel $p \in P$
$\Longrightarrow_G^*$ reflexiver und transitiver Abschluss von $\Longrightarrow_G$

Definition: Die von einer Grammatik $G = (N, T, P, S)$ mit Auswahlkontext erzeugte Sprache $L(G)$ ist

$$L(G) = \{w \mid S \Longrightarrow_G^* w,\ w \in T^*\}.$$

Grammatiken mit Auswahlkontext III

$G_1 = (\{S, A, A', A_a, A_b, B, B'\}, \{a, b\}, \{p_0, p_1, \dots p_{10}\})$

$p_0 = (S \rightarrow AB, \emptyset, \emptyset)$, $p_1 = (A \rightarrow aA_a, \{B\}, \emptyset)$, $p_2 = (A \rightarrow bA_b, \{B\}, \emptyset)$,
$p_3 = (B \rightarrow aB', \{A_a\}, \emptyset)$, $p_4 = (B \rightarrow bB', \{A_b\}, \emptyset)$, $p_5 = (A_a \rightarrow A, \{B'\}, \emptyset)$,
$p_6 = (A_b \rightarrow A, \{B'\}, \emptyset)$, $p_7 = (B' \rightarrow B, \{A\}, \emptyset)$ $p_8 = (A \rightarrow A'', \{B\}, \emptyset)$,
$p_9 = (B \rightarrow \lambda, \{A''\}, \emptyset)$, $p_{10} = (A'' \rightarrow \lambda, \emptyset, \emptyset)$

$L(G_1) = \{ww \mid w \in \{a, b\}^* \}$

$G_2 = (\{S, A, A', B, C, D\}, \{a\}, \{p_0, p_1, \dots p_28\}, S)$

$p_0 = (S \rightarrow CA, \emptyset, \emptyset)$, $p_1 = (S \rightarrow BA, \emptyset, \emptyset)$, $p_2 = (A \rightarrow a, \{C\}, \emptyset)$,
$p_3 = (C \rightarrow a, \emptyset, \{A, A'\})$, $p_4 = (A \rightarrow A'A', \{B\}, \emptyset)$, $p_5 = (B \rightarrow D, \emptyset\{A\})$,
$p_6 = (A' \rightarrow A, \{D\}, \emptyset)$, $p_7 = (D \rightarrow B, \emptyset, \{A'\})$, $p_8 = (D \rightarrow C, \emptyset, \{A'\})$

$L(G_2) = \{a^{2^n+1} \mid n \geq 0\}$

Programmierte Grammatiken I

Definition:

Eine programmierte Grammatik ist eine Quadrupel $G = (N, T, Lab, P, S)$, wobei

— N, T, S wie bei einer Regelgrammatik spezifiziert sind,

— Lab eine endliche Menge von Labels ist,

— P eine endliche Menge von Quadrupeln $p = (l_p, A_p \to w_p, \sigma_p, \varphi_p)$ ist, wobei jeweils

- $l_p \in Lab$,
- $\sigma_p \subseteq Lab$ und
- $\varphi_p \subseteq Lab$

gelten.

Programmierte Grammatiken II

Definition: Die von einer programmierten Grammatik $G = (N, T, Lab, P, S)$ erzeugte Sprache besteht aus allen Wörtern $w \in T^*$, für die es eine Ableitung

$$S = w_0 \Longrightarrow_{p_1} w_1 \Longrightarrow_{p_2} w_2 \Longrightarrow_{p_3} \ldots \Longrightarrow_{p_k} w_k = w,$$

mit $k \geq 1$ so gibt, dass folgende Bedingungen für $1 \leq i \leq k$ gelten:
— $r_i = (l_i, A_i \to v_i, \sigma_i, \varphi_i)$,
— entweder gelten

$$w_{i-1} = w'_{i-1}A_i w''_{i-1},\ w_i = w'_{i-1}v_i w''_{i-1} \text{ für gewisse } w'_{i-1}, w''_{i-1} \in V_G^*,\ l_{i+1} \in \sigma_i$$

oder

$$A_i \text{ kommt in } w_{i-1} \text{ nicht vor, } w_{i-1} = w_i,\ l_{i+1} \in \varphi_i.$$

Programmierte Grammatiken III

$G'_1 = (\{S, A, B\}, \{a, b\}, \{q_0, q_1, ;q_8\}, \{r_0, r_1, r_2, \dots, r_8\}, S)$

$r_0 = (q_0, S \to AB, \{q_1, q_3, q_5, q_7\}, \emptyset)$,

$r_1 = (q_1, A \to aA, \{q_2\}, \emptyset)$, $r_2 = (q_2, B \to aB, \{q_1, q_3, q_5, q_7\}, \emptyset)$,

$r_3 = (q_3, A \to bA, \{q_4\}, \emptyset)$, $r_4 = (q_4, B \to bB, \{q_1, q_3, q_5, q_7\}, \emptyset)$,

$r_5 = (q_5, A \to a, \{q_6\}, \emptyset)$, $r_6 = (q_6, B \to a, \emptyset, \emptyset)$,

$r_7 = (q_7, A \to b, \{q_8\}, \emptyset)$, $r_8 = (q_8, B \to b, \emptyset, \emptyset)$,

$L(G'_1) = \{ww \mid w \in \{a, b\}^+ \}$

$G'_2 = (\{S, A\}, \{a\}, \{q_1, q_2, q_3\}, \{r_1, r_2, r_3\}, S)$

$r_1 = (q_1, S \to AA, \{q_1\}, \{q_2\})$, $r_2 = (q_2, A \to S, \{q_2\}, \{q_1, q_3\})$,

$r_3 = (q_3, S \to a, \{q_3\}, \emptyset)$

$L(G') = \{a^{2^m} \mid m \geq 0\}$

Resultate

$\mathcal{L}(P)$ – Menge der von programmierten Grammatiken erzeugten Sprachen,
$\mathcal{L}(RC)$ – Menge der von Grammatiken mit Auswahlkontext erzeugten Sprachen,

Lemma: Für jede Sprache $L \in \mathcal{L}(P)$, $L \subseteq T^*$ und jedes $a \in T$ gilt

$$\{w \mid aw \in L\} \in \mathcal{L}(P).$$

Satz: $\mathcal{L}(RC) = \mathcal{L}(P) \subset \mathcal{L}(CS)$

Satz: i) $\mathcal{L}(P)$ ist eine AFL.
ii) Das Mitgliedsproblem für programmierte Grammatiken is **NP**-vollständig.
iii) Das Leerheitsproblem für programmierte Grammatiken, bei denen jedes Fehlerfeld leer ist, ist entscheidbar.