Prof. Dr. Hans-Jörg Kreowski **Juni 2007**

Formale Sprachen: DNA Computing

Insertion-Deletion-Systeme

Ein *Insertion-Deletion-System* (kurz: *insdel*-System) $\gamma = (V, T, A, R)$ besteht aus einem endlichen *Alphabet* V, einem *terminalen Teilalphabet* $T \subseteq V$, einer endlichen Menge von *Axiomen* (Startwörtern) $A \subseteq V^*$ und einer endlichen Menge von *Regeln* $R \subseteq V^* \times ((V^+ \times \{\lambda\}) \cup (\{\lambda\} \times V^+)) \times V^*$.

Die Anwendung von Regeln auf Wörter liefert eine binäre Relation von *Berechnungsschritten*:

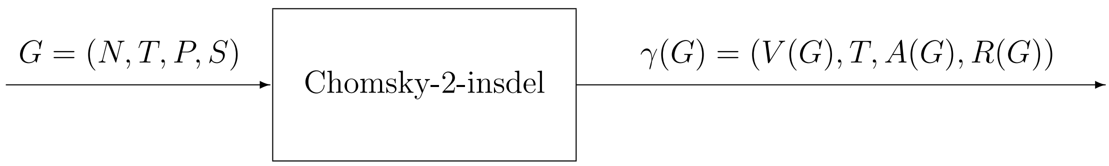
$$\text{Insertion} : w = xuvy \underset{r}{\Longrightarrow} xu\beta vy \text{ für } r = (u, \lambda/\beta, v) \in R$$
$$\text{Deletion} : w = xu\alpha vy \underset{r}{\Longrightarrow} xuvy \text{ für } r = (u, \alpha/\lambda, v) \in R$$

Der reflexive und transitive Abschluss aller Berechnungsschritte wird wie üblich mit $\overset{*}{\underset{R}{\Longrightarrow}}$ oder $\overset{*}{\Longrightarrow}$ bezeichnet und *Berechnungsrelation* genannt.

Die *erzeugte Sprache* besteht aus allen terminalen Wörtern, die sich aus Axiomen berechnen lassen:

$$L(\gamma) = \{w \in T^* \mid v \overset{*}{\underset{R}{\Longrightarrow}} w, v \in A\}.$$

Ein erstes grundlegendes Ergebnis zeigt, dass sich eine Chomsky-Grammatik korrekt durch ein Insertion-Deletion-System simulieren lässt:



mit $V(G) = N \cup T \cup \{K_u \mid (u{::=}v) \in P\} \cup \{E\}$
$\quad A(G) = \{SE^{\max}\}$
$\quad R(G) = \{(u, \lambda/K_u v, t) \mid (u{::=}v) \in P, t \in (N \cup T \cup \{E\})^{\max}\} \cup$
$\quad\quad \{(\lambda, uK_u/\lambda, \lambda) \mid \quad (u{::=}v) \in P\} \cup \{(\lambda, E^{\max}/\lambda, \lambda)\}$
wobei $\max = \max\{length(u) \mid (u{::=}v) \in P\}$.

Die Übersetzung bewahrt die Sprache:

$$L(G) = L(\gamma(G)).$$

Denn es gilt:

1. $S \xrightarrow{*} w$ impliziert $SE^{\max} \overset{*}{\Longrightarrow} wE^{\max}$
2. $wE^{\max} \Longrightarrow w$
3. $L(G) \subseteq L(\gamma(G))$
4. $SE^{\max} \overset{*}{\Longrightarrow} wE^{\max}$ mit $w \in (N \cup T)^*$ impliziert $S \xrightarrow{*} w$
5. $SE^{\max} \overset{*}{\Longrightarrow} w$ mit $w \in (N \cup T)^*$ impliziert $SE^{\max} \overset{*}{\Longrightarrow} wE^{\max} \Longrightarrow w$
6. $L(\gamma(G)) \subseteq L(G)$

Dabei folgt 3 aus 1 und 2 als Spezialfall und 6 entsprechend aus 4 und 5. 2 liefert die letzte Deletion-Regel. 5 ergibt sich daraus, dass das Löschen von $E^{\max}$ immer ganz am Ende gemacht werden kann. 1 und 4 zeigt man mit Induktion über die Länge der Ableitung beziehungsweise über die Zahl der Insertionen. Dabei kann man die letzte Insertion ans Ende schieben direkt vor die letzte Deletion, die dann zusammen gerade einen Ableitungsschritt bilden.

Betrachte eine Turing-Maschine $TM = (Z, A, d, s_0, F)$ mit $s_0 \in Z, F \subseteq Z, d \subseteq Z \times A' \times Z \times A' \times \{l, r, n\}$ und $A' = A \cup \{\Box\}$.

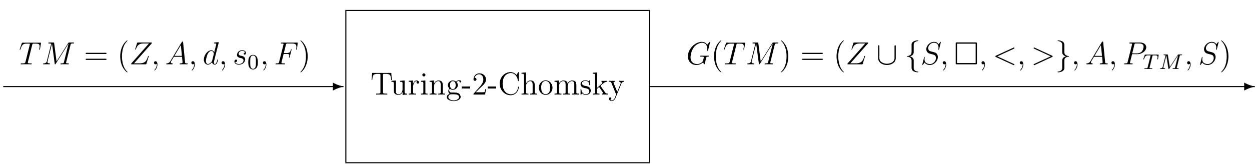
Berechnungsschritte verändern Konfigurationen, die den Bandinhalt, den aktuellen Zustand und das Zeichen repräsentieren, das als nächstes gelesen wird. Es steht rechts vom Zustand. Der rechte und linke Rand des Bandes sind markiert:

$< usav > \vdash < us'bv >$ für $(s', b, n) \in d(s, a)$
$< usav > \vdash < ubs'v >$ für $(s', b, r) \in d(s, a)$
$< ucsav > \vdash < us'cbv >$ für $(s', b, l) \in d(s, a)$
$< sav > \vdash < s'\Box\ bv >$ für $(s', b, l) \in d(s, a)$
$< us > \vdash < us\Box >$

Die erzeugte Sprache ist:

$$L(TM) = \{w \in A^* \mid < s_0 > \overset{*}{\vdash} < s''w\ \Box^i >, s'' \in F, i \in I\!N\}$$

Nach der Turingschen These wird jede berechenbare Funktion durch eine Turing-Maschine berechnet. Analog wird jede rekursiv aufzählbare Sprache durch eine Turing-Maschine erzeugt. Wer dem nicht traut, nehme eine Chomsky-Grammatik und baue eine Turing-Maschine, die die Spracherzeugung der Grammatik simuliert. Umgekehrt kann eine Turing-Maschine durch eine Chomsky-Grammatik simuliert werden:



wobei P_{TM} folgende Regeln enthält:

$sa ::= s'b$ für $(s', b, n) \in d(s, a)$
$sa ::= bs'$ für $(s', b, r) \in d(s, a)$
$csa ::= s'cb$ für $(s', b, l) \in d(s, a)$
$sa ::= s' \square b$ für $(s', b, l) \in d(s, a)$
$s > ::= s \square >$
$S ::= < s_0 >$
$< s'' ::= \lambda$ für $s'' \in F$
$\square > ::= >$
$> ::= \lambda$

Es ist fast offensichtlich, dass gilt:

$$< s_0 > \overset{*}{\vdash} < usv > \text{ gdw. } < s_0 > \overset{*}{\rightarrow} < usv >,$$

wenn man nur die ersten fünf Regeltypen verwendet, da die genau den Rechenschritten entsprechen. Mit den anderen Regeln folgt insbesondere, dass:

$$< s_0 > \overset{*}{\vdash} < s''w \, \square^i > \text{ impliziert } S \overset{*}{\rightarrow} w.$$

Da auch die Umkehrung gilt, ergibt sich die Korrektheit der Simulation:

$$L(G(FM)) = L(TM).$$