

Definition 3.10 Eine **Reduktionsstrategie** S ist eine Abbildung

$$S : T(\Sigma, V) \rightarrow 2^{\text{Pos}(T(\Sigma, V))}$$

wobei $S(t) \subseteq \text{Pos}(t)$ und es gilt:

- $t|_p$ ist ein Redex für alle $p \in S(t)$ und
- für alle $p_1, p_2 \in S(t)$ mit $p_1 \neq p_2$ gilt: p_1 und p_2 sind disjunkt.

Intuition: S legt die Positionen fest, bei denen als nächstes reduziert wird

Weiter definieren wir:

- $t_1 \rightarrow_R t_2 \rightarrow_R t_3 \rightarrow_R \dots$ heißt **S -Reduktion**, falls für $i = 1, 2, 3 \dots$ gilt: t_{i+1} entsteht aus t_i durch Anwendung von Reduktionsschritten an allen Positionen aus $S(t_i)$.
- S heißt **sequenziell** $:\Leftrightarrow \forall t \in T(\Sigma, V)$ gilt: $|S(t)| \leq 1$
- S heißt **normalisierend** $:\Leftrightarrow \forall t, t' \in T(\Sigma, V)$ mit $t' = t \downarrow_R$ gilt: jede S -Reduktion $t \rightarrow_R t_1 \rightarrow_R \dots$ endet in t' (d.h. die Strategie S berechnet immer die Normalform, falls diese existiert).

Beispiele für Reduktionsstrategien (im Folgenden nehmen wir an, dass $t \in T(\Sigma, V)$ nicht in Normalform ist):

Leftmost innermost (LI): $S(t) = \{p\}$ falls $t|_p$ Redex und $\forall q \neq p$ mit $t|_q$ Redex gilt: q rechts oder über p

Leftmost outermost (LO): $S(t) = \{p\}$ falls $t|_p$ Redex und $\forall q \neq p$ mit $t|_q$ Redex gilt: p über oder links von q

Parallel outermost (PO): $S(t) = \{p \mid t|_p \text{ Redex und } \forall q \neq p \text{ mit } t|_q \text{ Redex gilt: } q \not\leq p\}$

Die Strategie LI ist nicht normalisierend, wie man am folgenden Beispiel sieht:

$$R : \begin{array}{l} 0 * x \rightarrow 0 \\ a \rightarrow a \end{array}$$

Es gilt:

$$(0 * a) \downarrow_R = 0$$

Jedoch findet die LI-Strategie keine Normalform:

$$0 * \underline{a} \xrightarrow{LI} 0 * \underline{a} \xrightarrow{LI} 0 * \underline{a} \dots$$

Die Strategie LO ist auch nicht normalisierend für orthogonale TES:¹

$$R : \begin{array}{l} a \rightarrow b \\ c \rightarrow c \\ f(x, b) \rightarrow d \end{array}$$

¹Beachte: Die Strategie LO ist nicht identisch zur Pattern-Matching-Strategie aus Kapitel 2.3!

Auf Grund der Ableitung

$$f(c, \underline{a}) \rightarrow_R \underline{f}(c, b) \rightarrow_R d$$

sehen wir, dass $f(c, a) \downarrow_R = d$ gilt, aber es gibt nur eine LO-Ableitung:

$$f(\underline{c}, a) \rightarrow_R^{LO} \underline{f}(c, a) \rightarrow_R^{LO} f(\underline{c}, a) \dots$$

Dagegen berechnet die PO-Strategie die Normalform:

$$f(\underline{c}, \underline{a}) \rightarrow_R^{PO} \underline{f}(c, b) \rightarrow_R^{PO} d$$

Allgemein gilt:

Satz 3.6 (O'Donnell [O'Donnell 77]) *PO ist normalisierend für (schwach) orthogonale Systeme.*

Aber: PO ist aufwändiger zu implementieren als LO.

Im vorigen Beispiel könnte man auch "Rightmost outermost" benutzen, um die Normalform zu berechnen. Im Allgemeinen ist eine sequenzielle Strategie aber nicht ausreichend, sondern manchmal ist die parallele Reduktion essentiell:

Beispiel: „Paralleles Oder“:

$$\begin{aligned} R: \text{true} \vee x &\rightarrow \text{true} \\ x \vee \text{true} &\rightarrow \text{true} \\ \text{false} \vee \text{false} &\rightarrow \text{false} \end{aligned}$$

Nun soll der folgende Term ausgerechnet werden, wobei t_1 und t_2 irgendwelche Terme sind:

$$t_1 \vee t_2$$

Bei einer sequenziellen Strategie besteht nun die Gefahr, dass, falls versucht wird zuerst t_1 (oder t_2) zu reduzieren, könnte dies zu einer unendlichen Ableitung führen, wohingegen t_2 (oder t_1) zu **true** reduzierbar ist. Somit wäre in diesem Beispiel ein „sicheres“ Vorgehen, t_1 und t_2 parallel auszuwerten.

In diesem Beispiel ist R nicht orthogonal, da die ersten beiden Regeln überlappen. Daher stellt sich die Frage: Sind orthogonale Systeme sequenziell normalisierbar?

Leider gilt dies auch nicht immer, wie folgendes Beispiel von Berry [Berry 76] zeigt:

$$\begin{aligned} R: f(x, 0, 1) &\rightarrow 0 \\ f(1, x, 0) &\rightarrow 1 \\ f(0, 1, x) &\rightarrow 2 \end{aligned}$$

Dieses System ist orthogonal, aber wenn wir den Ausdruck $f(t_1, t_2, t_3)$ betrachten, ist es nicht klar, welches t_i zuerst reduziert werden muss.

Aus diesen Betrachtungen stellt sich die interessante Frage:

Gibt es für bestimmte Klassen von TES sequenzielle (und evtl. optimale) Reduktionsstrategien?

Ein erste Klasse können wir wie folgt definieren:

Definition 3.11 *Ein TES R heißt **linksnormal** (**left-normal**), falls für alle Regeln $l \rightarrow r \in R$ gilt: in l steht „hinter“ einer Variablen kein Funktionssymbol.*

Beispiele:

Das System

$$\begin{array}{l} 0 * x \rightarrow 0 \\ a \rightarrow a \end{array}$$

ist linksnormal. Dagegen ist die Regel

$$f(x, b) \rightarrow d$$

nicht linksnormal.

Satz 3.7 (O'Donnell [O'Donnell 77]) *LO ist normalisierend für linksnormale orthogonale TES.*

Anwendung: Die Kombinatorlogik (eine Implementierung des Lambda-Kalküls) oder auch ein Termersetzungssysteme R mit $\text{Var}(l) = \emptyset$ für alle $l \rightarrow r \in R$ sind linksnormal, sodass hierfür LO normalisierend ist.