

Theorie der Programmierung

Übung 03 — Polynomordnungen

Leon Vatthauer

16. Mai 2025

Definition 2.42 (Polynomordnung)

Eine **monotone polynomielle Interpretation** $\mathcal{A}$ für Σ besteht aus

- einem streng monotonen Polynom $0 \neq p_f \in \mathbb{N}[x_1, \dots, x_n]$ für jedes $f/n \in \Sigma$,
- einer Teilmenge $\emptyset \neq A \subseteq \mathbb{N} \setminus \{0\}$ die unter den p_f abgeschlossen ist.

Definition 2.42 (Polynomordnung)

Eine **monotone polynomielle Interpretation** $\mathcal{A}$ für Σ besteht aus

- einem streng monotonen Polynom $0 \neq p_f \in \mathbb{N}[x_1, \dots, x_n]$ für jedes $f/n \in \Sigma$,
- einer Teilmenge $\emptyset \neq A \subseteq \mathbb{N} \setminus \{0\}$ die unter den p_f abgeschlossen ist.

Die hierdurch induzierte **Polynomordnung** auf Termen ist definiert durch

$$t \succ_{\mathcal{A}} s : \iff p_t >_A p_s$$

Wobei p_t definiert ist durch:

$$p_x = x \qquad p_{f(t_1, \dots, t_n)} = p_f(p_{t_1}, \dots, p_{t_n}).$$

Definition 2.42 (Polynomordnung)

Eine **monotone polynomielle Interpretation** $\mathcal{A}$ für Σ besteht aus

- einem streng monotonen Polynom $0 \neq p_f \in \mathbb{N}[x_1, \dots, x_n]$ für jedes $f/n \in \Sigma$,
- einer Teilmenge $\emptyset \neq A \subseteq \mathbb{N} \setminus \{0\}$ die unter den p_f abgeschlossen ist.

Die hierdurch induzierte **Polynomordnung** auf Termen ist definiert durch

$$t \succ_{\mathcal{A}} s : \Longleftrightarrow p_t >_A p_s$$

Wobei p_t definiert ist durch:

$$p_x = x \qquad p_{f(t_1, \dots, t_n)} = p_f(p_{t_1}, \dots, p_{t_n}).$$

Korollar 2.45

Sei $\rightarrow_0$ ein TES und $>_A$ eine Polynomordnung. Falls

$$t \rightarrow_0 s \Rightarrow t \succ_A s \qquad \text{für alle Terme } t, s,$$

dann ist $\rightarrow$ stark normalisierend.

Terminationsbeweise

Sei $T = (\Sigma, \rightarrow_0)$ ein Termersetzungssystem. Wir wollen zeigen, dass T stark normalisierend (SN) ist, dazu gehen wir wie folgt vor:

1. Stelle für jedes $f/n \in \Sigma$ ein Polynom $p_f(x_1, \dots, x_n)$ auf.
2. Wähle eine Domäne $\emptyset \neq A \subseteq \mathbb{N} \setminus \{0\}$ die unter den p_f abgeschlossen ist. (Meistens eine Menge der Form $\mathbb{N}_{\geq k} = \{n \in \mathbb{N} \mid n \geq k\}$)
3. Zeige $\forall t, s. t \rightarrow_0 s \Rightarrow p_t >_A p_s$.

Aufgabe 1

Terminationsbeweise mittels Polynomordnung

Wir betrachten das folgende TES für $\Sigma = \{\odot/2, \oplus/2\}$:

$$x \oplus (y \odot z) \rightarrow_0 (x \oplus y) \odot (x \oplus z) \tag{1}$$

$$(x \odot y) \odot z \rightarrow_0 x \odot (y \odot z) \tag{2}$$

Aufgabe 1

Terminationsbeweise mittels Polynomordnung

Wir betrachten das folgende TES für $\Sigma = \{\odot/2, \oplus/2\}$:

$$x \oplus (y \odot z) \rightarrow_0 (x \oplus y) \odot (x \oplus z) \quad (1)$$

$$(x \odot y) \odot z \rightarrow_0 x \odot (y \odot z) \quad (2)$$

Wir wollen zeigen, dass das TES SN ist. Dazu verwenden wir die folgenden Polynome:

$$p_{\oplus}(x, y) := xy$$

$$p_{\odot}(x, y) := 2x + y + 1$$

Terminationsbeweise mittels Polynomordnung

Wir betrachten das folgende TES für $\Sigma = \{\odot/2, \oplus/2\}$:

$$x \oplus (y \odot z) \rightarrow_0 (x \oplus y) \odot (x \oplus z) \quad (1)$$

$$(x \odot y) \odot z \rightarrow_0 x \odot (y \odot z) \quad (2)$$

Wir wollen zeigen, dass das TES SN ist. Dazu verwenden wir die folgenden Polynome:

$$p_{\oplus}(x, y) := xy$$

$$p_{\odot}(x, y) := 2x + y + 1$$

1. Es sei $\mathcal{A} = \langle \mathbb{N}, p_{\oplus}, p_{\odot} \rangle$ die einfachste auf $p_{\oplus}$ und $p_{\odot}$ basierende polynomielle Interpretation von Σ . Begründen Sie, dass die von $\mathcal{A}$ induzierte Polynomordnung $\succ_{\mathcal{A}}$ nicht geeignet ist, um die Termination des Systems zu zeigen.

Aufgabe 1

Terminationsbeweise mittels Polynomordnung

Wir betrachten das folgende TES für $\Sigma = \{\odot/2, \oplus/2\}$:

$$x \oplus (y \odot z) \rightarrow_0 (x \oplus y) \odot (x \oplus z) \quad (1)$$

$$(x \odot y) \odot z \rightarrow_0 x \odot (y \odot z) \quad (2)$$

Wir wollen zeigen, dass das TES SN ist. Dazu verwenden wir die folgenden Polynome:

$$p_{\oplus}(x, y) := xy$$

$$p_{\odot}(x, y) := 2x + y + 1$$

2. Finden Sie eine geeignete Domäne $B \subseteq \mathbb{N}_{\geq 1}$ für welche die durch die Interpretation $\mathcal{B} = \langle B, p_{\oplus}, p_{\odot} \rangle$ induzierte Polynomordnung $\succ_{\mathcal{B}}$ die Termination des Systems zeigt.

Aufgabe 1

Terminationsbeweise mittels Polynomordnung

Wir betrachten das folgende TES für $\Sigma = \{\odot/2, \oplus/2\}$:

$$x \oplus (y \odot z) \rightarrow_0 (x \oplus y) \odot (x \oplus z) \quad (1)$$

$$(x \odot y) \odot z \rightarrow_0 x \odot (y \odot z) \quad (2)$$

Wir wollen zeigen, dass das TES SN ist. Dazu verwenden wir die folgenden Polynome:

$$p_{\oplus}(x, y) := xy$$

$$p_{\odot}(x, y) := 2x + y + 1$$

3. Wir ersetzen die zweite Reduktionsregel durch

$$x \odot (y \odot z) \rightarrow_0 (x \odot y) \odot z. \quad (3)$$

Zeigen Sie unter Verwendung einer Polynomordnung, dass das so erhaltene System ebenfalls stark normalisierend ist.

Aufgabe 2

Woher kommen diese Polynome?

Wir betrachten das folgende in Haskell-Syntax gegebene Programm:

data Nat = *Z* | *S* **Nat**

$$x + Z = x$$

$$x + (S\ y) = S\ (x + y)$$

$$d\ Z = Z$$

$$d\ (S\ x) = S\ (S\ (d\ x))$$

$$q\ Z = Z$$

$$q\ (S\ x) = q\ x + S\ (d\ x)$$

Aufgabe 2

Woher kommen diese Polynome?

Wir betrachten das folgende in Haskell-Syntax gegebene Programm:

data Nat = Z | S Nat

$x + Z = x$

$x + (S\ y) = S\ (x + y)$

$d\ Z = Z$

$d\ (S\ x) = S\ (S\ (d\ x))$

$q\ Z = Z$

$q\ (S\ x) = q\ x + S\ (d\ x)$

1. Drücken Sie dieses Programm als TES über der Signatur $\Sigma = \{0/0, s/1, +/2, d/1, q/1\}$ aus.

Aufgabe 2

Woher kommen diese Polynome?

Wir betrachten das folgende in Haskell-Syntax gegebene Programm:

data Nat = **Z** | **S Nat**

$$x + Z = x$$

$$x + (S\ y) = S\ (x + y)$$

$$d\ Z = Z$$

$$d\ (S\ x) = S\ (S\ (d\ x))$$

$$q\ Z = Z$$

$$q\ (S\ x) = q\ x + S\ (d\ x)$$

2. Verwenden Sie eine Polynomordnung um zu beweisen, dass das System stark normalisierend ist.
Hinweis: Wählen Sie $p_s(x) := x + 1$ und leiten Sie hieraus geeignete Werte für p_+ , p_d , p_q und p_0 her.

Woher kommen diese Polynome?

Wir betrachten das folgende in Haskell-Syntax gegebene Programm:

data Nat = Z | S Nat

$$x + Z = x$$

$$x + (S\ y) = S\ (x + y)$$

$$d\ Z = Z$$

$$d\ (S\ x) = S\ (S\ (d\ x))$$

$$q\ Z = Z$$

$$q\ (S\ x) = q\ x + S\ (d\ x)$$

3. Können wir also schließen, dass jedes aus den obigen Funktionen zusammengesetzte Programm terminierend sein wird? Wird die Termination eines solchen Programmes davon abhängen, ob eine strikte oder eine nicht-strikte Auswertungsstrategie verwendet wird?