Theoretische Grundlagen des Software Engineering

11: Abstrakte Reduktionssysteme

Stephan Schulz
schulz@eprover.org

Reduktionssysteme

Definition: Reduktionssystem

Ein Reduktionssystem ist ein Tupel

$$(A, \rightarrow)$$

Dabei gilt:

- A ist eine beliebige Menge (die Trägermenge)
- $\rightarrow \subseteq A \times A$ ist eine zweistellige Relation über A (die Reduktionsrelation)

Ausprägungen

Abstrakte Reduktionssysteme

- Universelle Grundlagen

Wortersetzungssysteme

- Berechenbarkeitstheorie/Grammatiken/Turing- Maschinen

Termersetzungssysteme

- Theorembeweisen, Programmsynthese, Funktionale Programmiersprachen

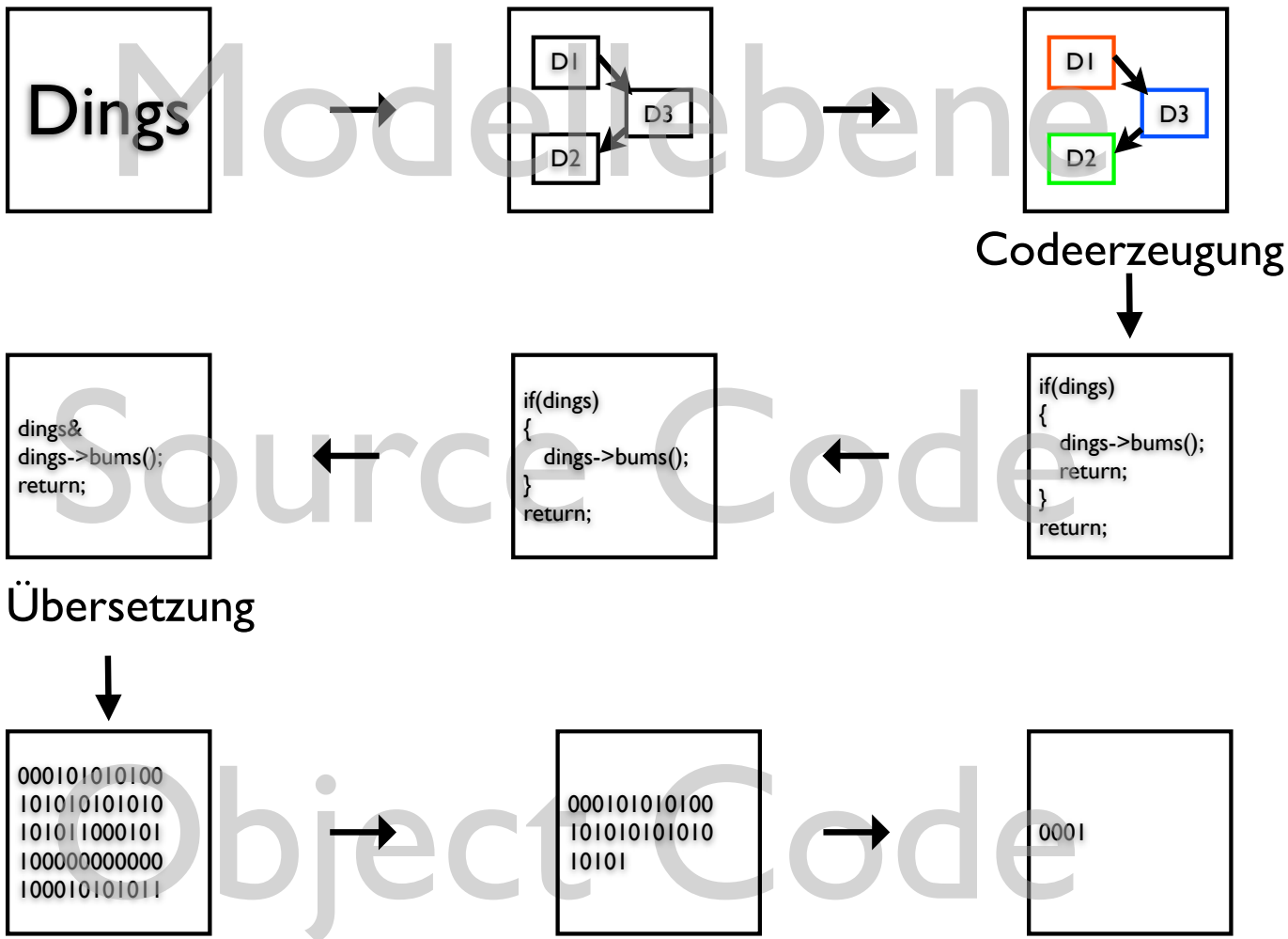
Polynom-Rewriting

- Computer-Algebra (Buchbergers Algorithmus/Gröbner-Basen)

Graph- Rewriting

- Verallgemeinerung von Termersetzung, Modellbasierte Entwicklung

Beispiel: SW- Entwickung



Interessante Fragen

Terminiert der Prozess?

- Als PL: Werden die Programmierer fertig?
- Als Programmierer: Wird der Compiler fertig?

Ist das Ergebnis unabhängig von der Reihenfolge?

- Designer: Kann ich verschiedene Entscheidungen unabhängig treffen?
- Compiler: Können Optimierungen in beliebiger Reihenfolge angewendet werden? Kann ich ein “optimales” Ergebnis garantieren?

Sind zwei Artefakte “gleich”?

- Compiler: Können sie substituiert werden?
- Designer: Können sie wiederverwendet werden?
- Rechtlich: Copyright!

Abstrakte Reduktionssysteme

Reduzibilität, Normalformen

Definition

Sei $(A, \rightarrow)$ ein Reduktionssystem und $x, y \in A$ beliebig

- x heißt reduzibel, falls $\exists y : x \rightarrow y$
- Wenn x nicht reduzibel ist, so heißt x irreduzibel oder in Normalform
- y heißt eine Normalform von x, falls $x \stackrel{*}{\rightarrow} y$ und y irreduzibel ist

Notation: Wenn x eine eindeutige Normalform hat, so bezeichnen wir diese mit $x\downarrow$

Nachfolger, Zusammenführbarkeit

Definition

Sei $(A, \rightarrow)$ ein Reduktionssystem und $x, y, z \in A$ beliebig

- y heißt direkter Nachfolger von x, falls $x \rightarrow y$
- y heißt Nachfolger von x, falls $x \overset{+}{\rightarrow} y$
- x und y heißen zusammenführbar, falls $\exists z : x \overset{*}{\rightarrow} z \overset{*}{\leftarrow} y$

Notation: Wenn x und y zusammenführbar sind, so schreiben wir $x \downarrow y$

Grundbegriffe

Beispiel

Sei $A = \mathbb{N}\backslash\{0, 1\}$ und $\rightarrow = \{(m, n) | m > n \text{ and } m \bmod n = 0\}$

- Also: $m \rightarrow n$ wenn n ein (echter) Teiler von m ist
- m ist in Normalform, wenn. . . m eine Primzahl ist
- p ist eine Normalform von m, wenn. . . p ein Primfaktor von m ist
- $m \downarrow n$, wenn. . . m und n einen gemeinsamen Teiler haben
- $\stackrel{+}{\rightarrow} = \ldots \rightarrow$ (denn Teilbarkeit und $>$ sind transitiv)
- $\stackrel{*}{\leftrightarrow} = \ldots A \times A$ (Idee: $a \rightarrow ab \leftarrow b$)

Konfluenz und Church-Rosser

Ziel: Eindeutige Normalformen

Idee

Wir wollen sicherstellen, das Ableitungen eineutige Ergebnisse liefern

- Implementierung von Rewriting zur Programausführung (λ- Kalkül, Haskell)
- Unabhängigkeit von Optimierungen/Transformationen

Wir wollen einfach feststellen, ob $a \downarrow b$

- Gleichheit von logischen/Artihmetischen Formeln
- Äquivalenz von Programmen

Was wird dazu genbraucht?

Konfluenz

Definition

Sei $(A, \rightarrow)$ ein Reduktionssystem

$\rightarrow$ heißt konfluent, genau dann, wenn

$\forall x, y_1, y_2 \in A : y_1 \overset{*}{\leftarrow} x \overset{*}{\rightarrow} y_2 \implies y_1 \downarrow y_2$

- Also: In einer konfluenten Reduktionsrelation kann jede Divergenz wieder zusammengeführt werden
- Korrolar: In einem konfluenten Reduktionssystem hat jedes Element höchstens eine Normalform

Diagrammatische Schreibweise: Konfluenz

Konfluenzdiagramm

Durchgezogene Linen: “für alle”

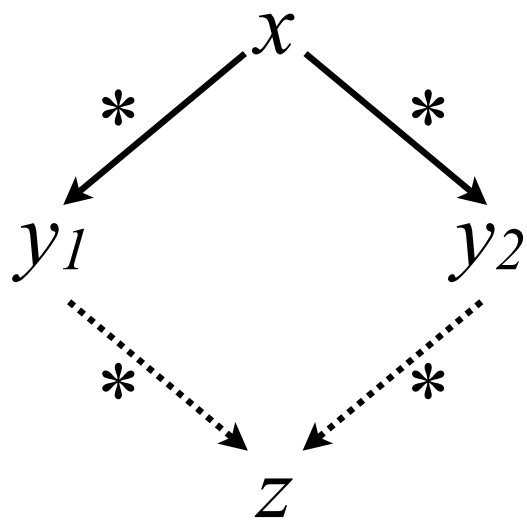
Gepunktete Linien: “es existiert”

Also:

$\forall x, y_1, y_2 : y_1 \overset{*}{\leftarrow} x \overset{*}{\rightarrow} y_2$

...

$\exists z : y_1 \overset{*}{\rightarrow} z \overset{*}{\leftarrow} y_2$



Church- Rosser

Alonzo Church $\rightarrow$

$\leftarrow$ J. Barkley Rosser

Definition

Sei $(A, \rightarrow)$ ein Reduktionssystem
$\rightarrow$ heißt Chuch- Rosser ("hat die Church- Rosser- Eigenschaft"), genau dann, wenn $\forall y_1, y_2 \in A : y_1 \overset{*}{\leftrightarrow} y_2 \implies y_1 \downarrow y_2$

- Also: In einem Church- Rosser- Reduktionssystem können beliebig verbundene Elemente nur durch Anwendung von $\rightarrow$ in einer Richtung zusammengeführt werden

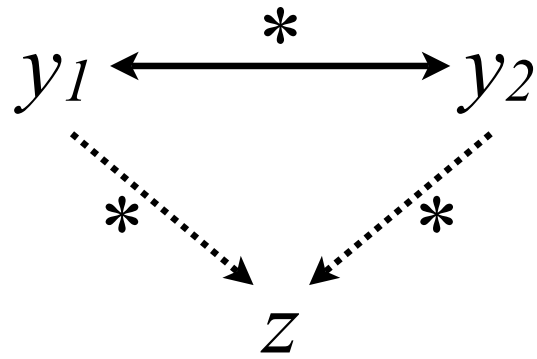
Diagrammatische Schreibweise: Church- Rosser

Church- Rosser- Diagramm

$$\forall y_1, y_2 : y_1 \overset{*}{\leftrightarrow} y_2$$

$$\ldots$$

$$\exists z : y_1 \overset{*}{\rightarrow} z \overset{*}{\leftarrow} y_2$$



Semi- Konfluenz

Definition

Sei $(A, \rightarrow)$ ein Reduktionssystem

$\rightarrow$ heißt semi-konfluent, genau dann, wenn

$\forall x, y_1, y_2 \in A : y_1 \leftarrow x \stackrel{*}{\rightarrow} y_2 \implies y_1 \downarrow y_2$

- Unterschied zur Konfluenz: Divergenz hat auf einer Seite nur einen Schritt!

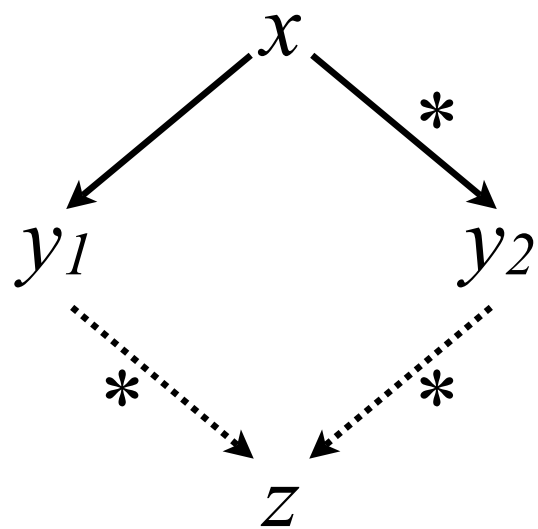
Diagramm: Semi-Konfluenz

Konfluenzdiagramm

$$\forall x, y_1, y_2 : y_1 \leftarrow x \stackrel{*}{\rightarrow} y_2$$

$$\ldots$$

$$\exists z : y_1 \stackrel{*}{\rightarrow} z \stackrel{*}{\leftarrow} y_2$$



Lokale Konfluenz

Definition

Sei $(A, \rightarrow)$ ein Reduktionssystem

$\rightarrow$ heißt lokal konfluent, genau dann, wenn

$\forall x, y_1, y_2 \in A : y_1 \leftarrow x \rightarrow y_2 \implies y_1 \downarrow y_2$

- Unterschied zur Konfluenz: Divergenz hat auf beiden Seiten nur einen Schritt!

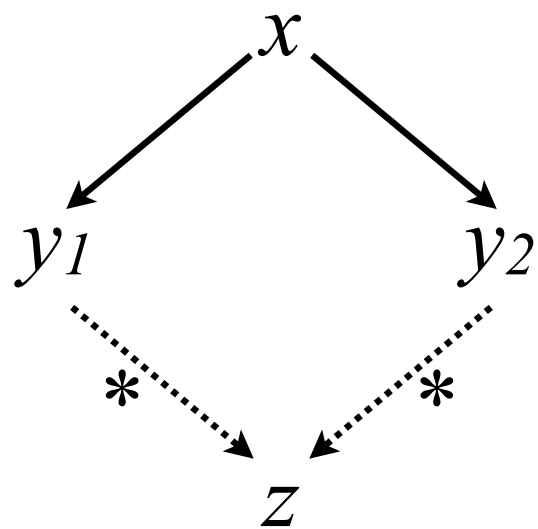
Diagramm: Lokale Konfluenz

Konfluenzdiagramm

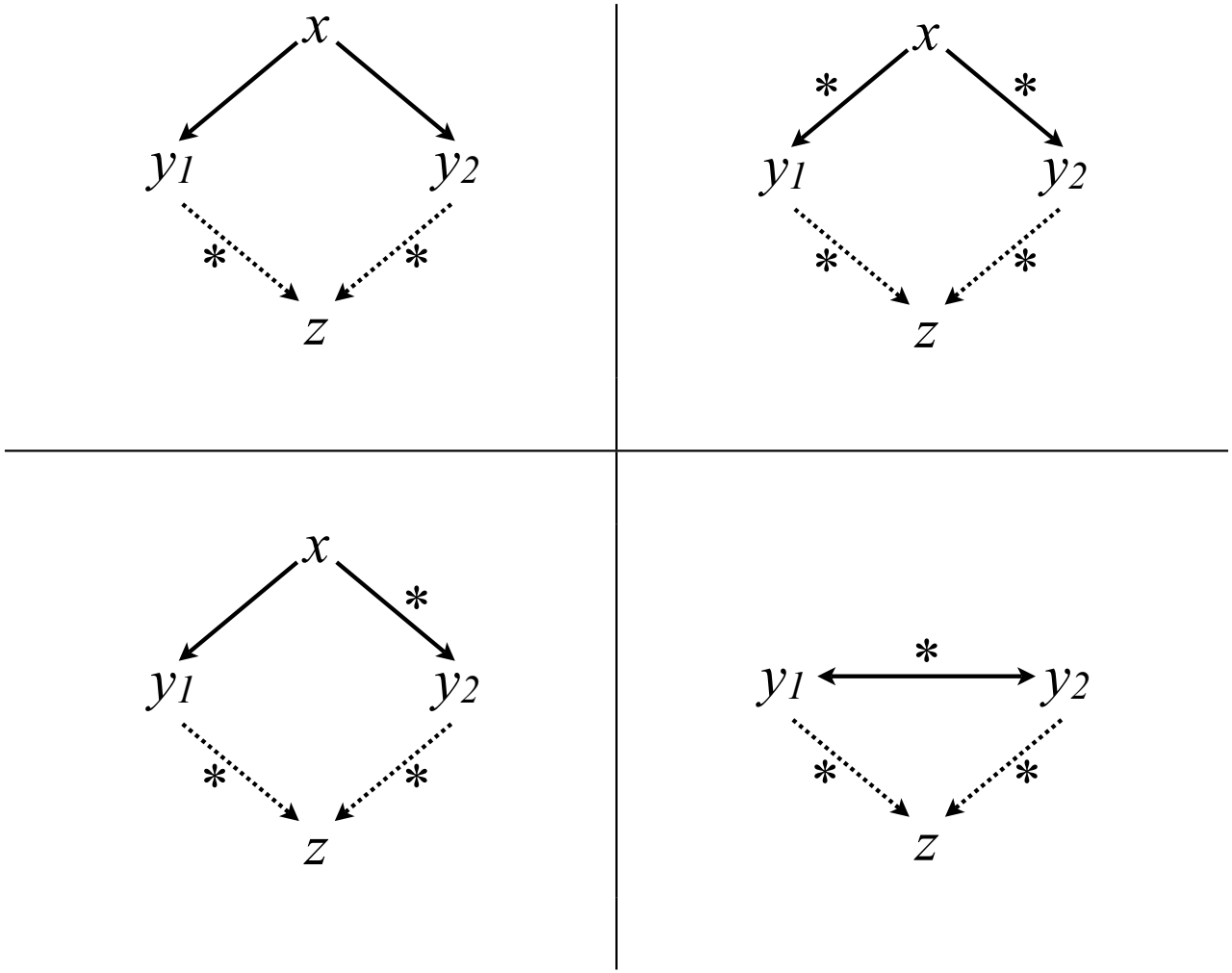
$\forall x, y_1, y_2 : y_1 \leftarrow x \rightarrow y_2$

$\ldots$

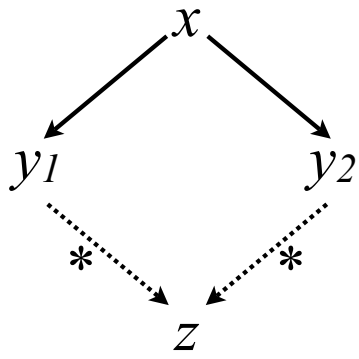
$\exists z : y_1 \stackrel{*}{\rightarrow} z \stackrel{*}{\leftarrow} y_2$



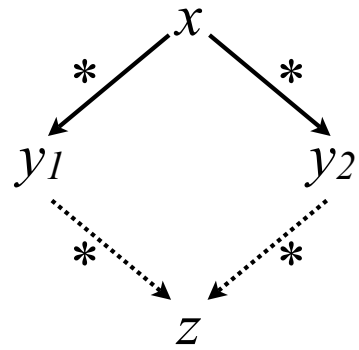
Quiz



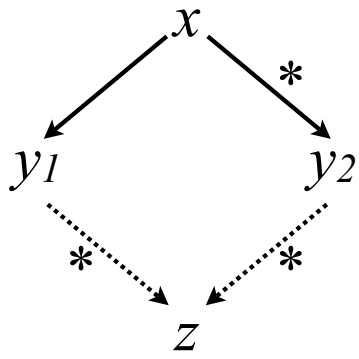
Lösung



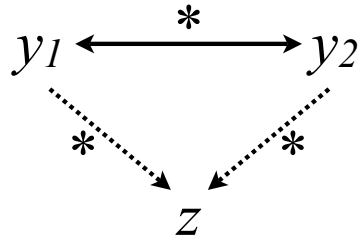
Lokale Konfluenz



Konfluenz



Semi- Konfluenz



Church- Rosser

Konfluenz und Church-Rosser

Satz

Sei $(A, \rightarrow)$ ein Reduktionssystem. Die folgenden 3 Eigenschaften sind äquivalent:

1. $\rightarrow$ ist Church- Rosser
2. $\rightarrow$ ist konfluent
3. $\rightarrow$ ist semi- konfluent

Anmerkung: Lokale Konfluenz ist i.A. schwächer!

Beweis: Per Ringschluss (1) $\Longrightarrow$ (2) $\Longrightarrow$ (3)

Church- Rosser impliziert Konfluenz

(1) $\Longrightarrow$ (2)

Behauptung: Sei $(A, \rightarrow)$ eine Reduktionssystem. Wenn $\rightarrow$ Church- Rosser ist, dann ist $\rightarrow$ konfluent

Beweis: Zu zeigen: $\forall x, y_1, y_2 : y_1 \overset{*}{\leftarrow} x \overset{*}{\rightarrow} y_2 : y_1 \downarrow y_2$

Sei also $x \in A$ beliebig und $\exists y_1, y_2 : y_1 \overset{*}{\leftarrow} x \overset{*}{\rightarrow} y_2$

Da $y_1 \overset{*}{\leftarrow} x \overset{*}{\rightarrow} y_2$ gilt insbesondere $y_1 \overset{*}{\leftrightarrow} y_2$

- ...denn $\overset{*}{\rightarrow} \subseteq \overset{*}{\leftrightarrow}$ und $\overset{*}{\leftarrow} \subseteq \overset{*}{\leftrightarrow}$ und
- ... $\overset{*}{\leftrightarrow}$ ist transitiv

Da $\rightarrow$ Church- Rosser ist, gilt $y_1 \downarrow y_2$

Da x beliebig war, folgt also die Behauptung.

q.e.d.

Konfluenz impliziert Semi- Konfluenz

(2) $\Longrightarrow$ (3)

Behauptung: Sei $(A, \rightarrow)$ eine Reduktionssystem. Wenn $\rightarrow$ konfluent ist, dann ist $\rightarrow$ semi- konfluent

Beweis: Zu zeigen: $\forall x, y_1, y_2 : y_1 \leftarrow x \stackrel{*}{\rightarrow} y_2 : y_1 \downarrow y_2$

Offensichtlich, denn $\rightarrow \subseteq \stackrel{*}{\rightarrow}$

q.e.d.

Semi- Konfluenz impliziert Church- Rosser

(3) $\Longrightarrow$ (1) – der spannende Teil

Behauptung: Sei $(A, \rightarrow)$ eine Reduktionssystem. Wenn $\rightarrow$ semi-konfluent ist, dann ist $\rightarrow$ Church- Rosser

Beweis: Zu zeigen: $\forall y_1, y_2 : y_1 \overset{*}{\leftrightarrow} : y_1 \downarrow y_2$.

Induktion über die Länge der Ableitungskette

Seien also $y_1, y_2 \in A$ beliebig und $y_1 \overset{n}{\leftrightarrow} y_2$

IA (n=0): $y_1 \overset{0}{\leftrightarrow} y_2 \implies y_1 = y_2 \implies y_1 \downarrow y_2$ (gleiche Elemente sind auch in 0 Schritten zusamengeführt).

Semi- Konfluenz impliziert Church- Rosser

IV: $y_1 \overset{n}{\leftrightarrow} y_2 \implies y_1 \downarrow y_2$

IS ($n \to n+1$): $y_1 \overset{n+1}{\leftrightarrow} y_2$, also $\exists y' : y_1 \leftrightarrow y' \overset{n}{\leftrightarrow} y_2$

Fallunterscheidung:

Fall 1: $\exists y' : y_1 \to y' \overset{n}{\leftrightarrow} y_2$

Nach IV: $y' \downarrow y_2$, also $\exists z : y' \overset{*}{\to} z \overset{*}{\leftarrow} y_2$

$\implies y_1 \to y' \overset{*}{\to} z \overset{*}{\leftarrow} y_2$

$\implies y_1 \overset{*}{\to} z \overset{*}{\leftarrow} y_2 \implies y_1 \downarrow y_2$

Fall 2: $\exists y' : y_1 \leftarrow y' \overset{n}{\leftrightarrow} y_2$

Nach IV: $y' \downarrow y_2$, also $\exists z : y_1 \leftarrow y' \overset{*}{\to} z \overset{*}{\leftarrow} y_2$

Da $\to$ semi-konfluent ist, gilt:

$\exists z' : y_1 \overset{*}{\to} z' \overset{*}{\leftarrow} z \overset{*}{\leftarrow} y_1$

$\implies y_1 \overset{*}{\to} z' \overset{*}{\leftarrow} y_2 \implies y_1 \downarrow y_2$ q.e.d.

Konvergenz und Termination

Konfluenz genügt nicht

Fakt

Es gibt Reduktionssysteme, die konfluent sind, aber keine eindeutigen Normalformen haben.

Wie kann das sein?

Konfluenz garantiert Eindeutigkeit einer Normalform, nicht Existenz

Termination, Beschränktheit

Definition

Sei $(A, \rightarrow)$ eine Reduktionssystem

- $\rightarrow$ heißt terminierend, wenn es keine unendliche Ableitung $a_0, \rightarrow a_1, \rightarrow a_2 \ldots$ gibt
- $\rightarrow$ heißt beschränkt, wenn gilt:
  $\forall a \in A \exists n \in \mathbb{N} \nexists b \in A : a \overset{n}{\rightarrow} b$
  Also: Für jedes Element gibt es eine "längste Kette"

Beschränktheit impliziert Termination, aber nicht umgekehrt (Übung)

Konvergenz

Definition

Sei $(A, \rightarrow)$ eine Reduktionssystem $\rightarrow$ heißt Konvergent genau dann, wenn

- $\rightarrow$ ist konfluent und
- $\rightarrow$ ist terminierend

Konvergenz garantiert:
(1) Jedes Element aus A hat eine eindeutige Normalform
(2) Alle Elemente einer $\overset{*}{\leftrightarrow}$- Äquivalenzklasse haben die selbe Normalform

Lokale Konfluenz und Termination

Satz

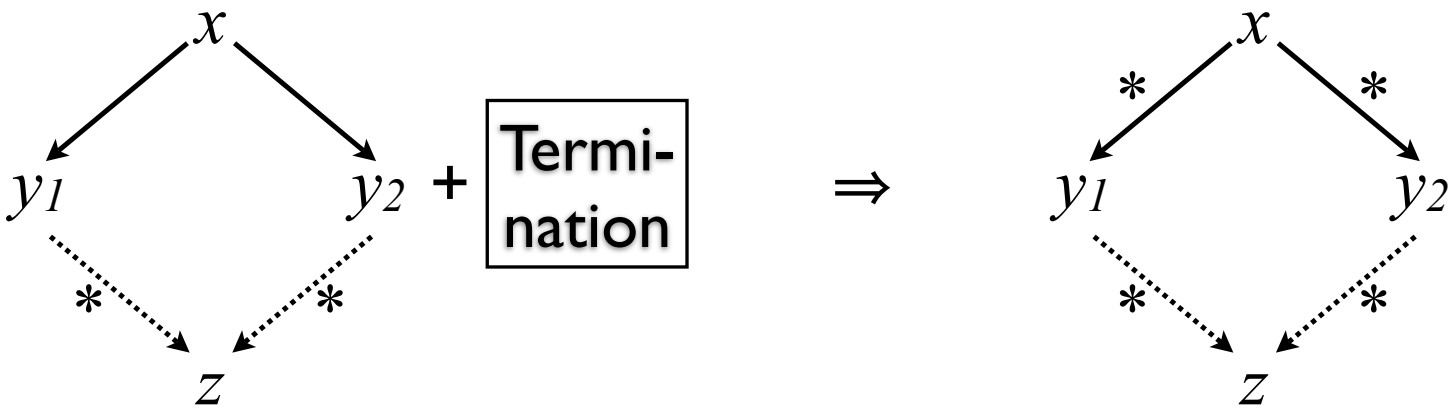
Sei $(A, \rightarrow)$ ein Reduktionssystem.

Sei $\rightarrow$ terminierend.

Dann gilt: $\rightarrow$ ist genau dann konfluent, wenn $\rightarrow$ lokal konfluent ist

Also: Für terminierende Reduktionssysteme fallen Konfluenz und lokale Konfluenz zusammen!

Satz im Bild



Beweisidee: Noethersche Induktion!

Zusammenfassung

Grundbegriffe

- Reduzibel, irreduzibel
- Normalform
- Zusammenführbar

Konfluenz

- Konfluenzbegiffe
- Church-Rosser
- Äquivalenz von Church-Rosser und Konfluenz

Termination

- Konvergenz
- Äquivalenz von lokaler Konfluenz und Konfluenz

Aufgaben

Geben Sie ein Reduktionssystem an, das lokal konfluent, aber nicht konfluent ist.

Geben Sie ein Reduktionssystem an, das terminierend, aber nicht beschränkt ist