Inferenzmethoden

Einheit 5

Unifikation

1. Wichtige Konzepte
2. Herbrand-Robinson Algorithmus
3. Martelli-Montanari Unifikation

Unifikation

- **Prädikatenlogische Beweise brauchen Substitutionen**
  - Konnektierte Literale müssen “gleich” gemacht werden können
  - Substitution darf γ-Variablen durch Terme ersetzen
  - Dieselbe Substitution muß alle Konnektionen komplementär machen
  - Technisches Problem: **Unifikation** (“Gleichmachen von Termen”)
- **Substitution: endliche Abbildung $\boldsymbol{\sigma : \mathcal{V} \to}$ Term** (vgl §1)
  - $\boldsymbol{\sigma = [t_1, .., t_n / x_1, .., x_n]} \mathrel{\hat{=}} \sigma(x_1){=}t_1, .., \sigma(x_n){=}t_n$
  - $\boldsymbol{\epsilon}$: leere (identische) Substitution
  - $\boldsymbol{A\sigma}$: Anwendung von σ auf Ausdruck A
  - $\boldsymbol{\tau\,\sigma}$: Komposition von τ und σ
- **Unifikationsproblem**
  - Sind die Literale $P(s_1, .., s_n)^T$ und $P(t_1, .., t_n)^F$ komplementär?

    Gegeben eine Gleichungsmenge $E = \{s_1 \doteq t_1, \ldots, s_n \doteq t_n\}$

    Gibt es eine Substitution σ mit $\sigma(t_i){=}\sigma(t_i)$ für alle $i \leq n$?
  - σ heißt Unifikator der Menge E (bzw. “von s und t” für $E = \{s \doteq t\}$)
  - E heißt lösbar, wenn es einen Unifikator für E gibt

Beispiele für Unifikatoren

- **Untersuche Komplementärität der Konnektion**

  $$P(x, f(gy), fx)^T \;—\; P(h(y,z), fz, f(h(u,v)))^F$$

- **Zugehörige Gleichungsmenge (auf Termen) ist**

  $$E = \{x \doteq h(y,z),\ fgy \doteq fz,\ fx \doteq fh(u,v)\}$$

- **Mögliche Unifikatoren**
  - $\sigma_1 = [h(y,z)/x,\ gy/z,\ y/u,\ z/v]$

    Ergibt $\{h(y,z) \doteq h(y,z),\ fgy \doteq fgy,\ fh(y,z) \doteq fh(y,z)\}$
  - $\sigma_2 = [h(y,gy)/x,\ gy/z,\ y/u,\ gy/v]$

    Ergibt $\{h(y,gy) \doteq h(y,gy),\ fgy \doteq fgy,\ fh(y,gy) \doteq fh(y,gy)\}$
  - $\sigma_3 = [h(u,gu)/x,\ gu/z,\ u/y,\ gu/v]$

    Ergibt $\{h(u,gu) \doteq h(u,gu),\ fgu \doteq fgu,\ fh(u,gu) \doteq fh(u,gu)\}$
  - $\sigma_4 = [h(ga,gga)/x,\ gga/z,\ ga/y,\ ga/u,\ gga/v]$

    Ergibt $\{h(ga,gga) \doteq h(ga,gga),\ fgga \doteq fgga,\ fh(ga,gga) \doteq fh(ga,gga)\}$

Gibt es Lösungen, die anderen vorzuziehen sind?

Allgemeinste Unifikatoren

- **Unifikatoren sollten möglichst stabil sein**
  - Mehrfache Anwendung von σ sollte nichts mehr ändern
  - Eine Substitution σ heißt idempotent falls $\sigma\,\sigma = \sigma$
  - $\sigma_2 = [h(y,gy)/x,\ gy/z,\ y/u,\ gy/v]$ ist idempotent
    $\sigma_1 = [h(y,z)/x,\ gy/z,\ y/u,\ z/v]$ ist nicht idempotent, da $\sigma_1\sigma_1 = \sigma_2$
- **Unifikatoren sollten möglichst allgemein sein**
  - σ ist allgemeiner als τ ($\sigma \geq \tau$), wenn $\tau = \sigma\vartheta$ für eine Substitution ϑ
  - σ ist allgemeinster Unifikator (mgu) einer Gleichungsmenge E, wenn σ Unifikator von E und allgemeiner als jeder Unifikator τ von E ist
  - $\sigma_2 = [h(y,gy)/x,\ gy/z,\ y/u,\ gy/v]$ ist allgemeinster Unifikator
  - Alle allgemeinsten Unifikator von E sind Varianten voneinander
    Für mgus σ und τ gibt es ϑ_1 und ϑ_2 mit $\tau = \sigma\vartheta_1$ und $\sigma = \tau\vartheta_2$
  - $\sigma_3 = [h(u,gu)/x,\ gu/z,\ u/y,\ gu/v]$ ist Variante von σ_2

Wie kann man Unifikatoren effizient bestimmen?

Unifikation – intuitiv

- **Schrittweises Angleichen der Terme von außen**
  - Eine Variable x unifiziert mit jedem Term t und liefert $\sigma=[t/x]$
  - Zwei Funktionsanwendungen $f(s_1, .., s_n)$ und $g(t_1, .., t_n)$ unifizieren nur, wenn $f=g$ und jedes s_i mit t_i unter derselben Substitution σ unifiziert

    σ kann Erweiterung aller Einzelsubstitutionen sein

    · $f(gx, ha)$ und $f(z, y)$ unifiziert mit $[gx/z]$ und $[ha/y]$ $\sigma=[gx/z, ha/y]$

    · $f(gx, ha)$ und $f(z, x)$ unifiziert mit $[gx/z]$ und $[ha/x]$ $\sigma=[g(ha)/z, ha/x]$

    · $f(gx, ha)$ und $f(g(hz), x)$ unifiziert mit $[hz/x]$ und $[ha/x]$ $\sigma=[ha/x, a/z]$

    · $f(gx, ha)$ und $f_1(z, y)$ unifiziert nicht, da $f\neq f_1$

    · $f(gx, ha)$ und $f(hz, y)$ unifiziert nicht, da gx nicht mit hz unifiziert

    · $f(gx, ha)$ und $f(ga, x)$ unifiziert nicht, da $[a/x]$ und $[ha/x]$ nicht kompatibel

    · $f(gx, ha)$ und $f(x, y)$ unifiziert nicht, da $[gx/x]$ zyklische Substitution

- **Aufgaben eines Unifikationsverfahrens**
  - Dekomposition von Termen bis zur Ebene der Variablen
  - Sequentieller Aufbau einer Substitution ersetzt Kompatibilitätstest
  - Occurs-Check: Test, ob Variable x in $\sigma(x)$ erscheint

Berechnung von Unifikatoren

Schrittweises Auflösen der Unterschiede zwischen Termen

- **Berechne die Differenz $\mathsf{DIFF}(s,t)$ der Terme s und t**

  Menge ungeordneter Termpaare mit der Eigenschaft

  – $\mathsf{DIFF}(s,t) = \begin{cases} \emptyset & \text{falls } s{=}t \\ \mathsf{DIFF}(s_1,t_1)\cup..\cup\mathsf{DIFF}(s_n,t_n) & \text{falls } s{=}f(s_1,..,s_n) \text{ und } t{=}f(t_1,..,t_n) \\ \{\{s,t\}\} & \text{sonst} \end{cases}$

  – $\mathsf{DIFF}(s_1,..,s_n,\ \ t_1,..,t_n) = \mathsf{DIFF}(s_1,t_1)\cup..\cup\mathsf{DIFF}(s_n,t_n)$

  Leicht zu implementierendes Konzept

- **Prüfe, ob $\mathsf{DIFF}(s,t)$ verhandlungsfähig ist**

  – $\mathsf{DIFF}(s,t)$ ist nicht leer

  – Alle Elemente sind Paare $\{x,t_0\}$ mit $x\in\mathcal{V}$ und x erscheint nicht in t_0

- **Reduktion einer verhandlungsfähigen Differenz $\mathsf{DIFF}(s,t)$**

  – Substitution $[t_0/x]$ mit $\{x,t_0\}\in\mathsf{DIFF}(s,t)$

  – Ermöglicht schrittweise Konstruktion eines Unifikators

Unifikationsalgorithmus von Herbrand–Robinson

Unifikation als Reduktion verhandlungsfähiger Differenzen

Eingabe		*Terme s und t*
Initialisierung		*Setze* $\sigma \leftarrow \epsilon$
Reduktion	*Solange*	$\mathsf{DIFF}(s\sigma, t\sigma)$ *verhandlungsfähig*
		wähle Reduktion ρ von $\mathsf{DIFF}(s\sigma, t\sigma)$
		Setze $\sigma \leftarrow \sigma\rho$
Ergebnis	*Falls*	$\mathsf{DIFF}(s\sigma, t\sigma) = \emptyset$
	dann	*σ ist mgu von $\{s, t\}$*
	sonst	*$\{s, t\}$ ist nicht unifizierbar*

Herbrand–Robinson Unifikation am Beispiel

$P(x, f(gy), fx)^T$ und $P(h(y,z), fz, f(h(u,v)))^F$

	σ	DIFF$(s\sigma, t\sigma)$
0	ϵ	$\{\{x, h(y,z)\}, \{z, gy\}, \{x, h(u,v)\}\}$
1	$[gy/z]$	$\{\{x, h(y,gy)\}, \{x, h(u,v)\}\}$
2	$[gy/z,\ h(u,v)/x]$	$\{\{u, y\}, \{v, gy\}\}$
3	$[gu/z,\ h(u,v)/x,\ u/y]$	$\{\{v, gu\}\}$
4	$[gu/z,\ h(u,v)/x,\ u/y,\ gu/v]$	$\emptyset$

$P(x, fc)^T$ und $P(fd, x)^F$

	σ	DIFF$(s\sigma, t\sigma)$
0	ϵ	$\{\{x, fd\}, \{x, fc\}\}$
1	$[fd/x]$	$\{\{d, c\}\}$
2		nicht verhandlungsfähig (keine Variable)

$P(x)^T$ und $P(fx)^F$

	σ	DIFF$(s\sigma, t\sigma)$
0	ϵ	$\{\{x, fx\}\}$
1		nicht verhandlungsfähig (occurs-check)

Eigenschaften des Herbrand–Robinson Verfahrens

Verhält sich das Verfahren wie gewünscht?

- **Terminierung** (Folie 9)
  – Für alle Terme s und t terminiert die Berechnung von $\mathsf{DIFF}(s,t)$
  – Für alle Terme s und t terminiert der Herbrand–Robinson Algorithmus
- **Korrektheit** (Folie 10)
  – Zwei Terme s und t sind genau dann unifizierbar, wenn der Herbrand–Robinson Algorithmus einen mgu von $\{s,t\}$ berechnet
- **Komplexität** (Folie 12)
  – Die Laufzeit des Herbrand–Robinson Algorithmus ist
  im Extremfall exponentiell
  aber im Mittel konstant

Beweis der Terminierung

- **Terminierung von $\mathsf{DIFF}(s,t)$**

$$\mathsf{DIFF}(s,t) = \begin{cases} \emptyset & \text{falls } s{=}t \\ \mathsf{DIFF}(s_1,t_1)\cup..\cup\mathsf{DIFF}(s_n,t_n) & \text{falls } s{=}f(s_1,..,s_n) \text{ und } t{=}f(t_1,..,t_n) \\ \{\{s,t\}\} & \text{sonst} \end{cases}$$

  – In jedem Schritt verringert sich die Termtiefe von s und t oder das Verfahren terminiert

  – Die Berechnung von $\mathsf{DIFF}(s,t)$ terminiert spätestens bei Termtiefe 0

- **Terminierung des Unifikationsalgorithmus**

  – Ist $\mathsf{DIFF}(s\sigma,t\sigma)$ verhandlungsfähig und ρ eine Reduktion, dann ist die Anzahl der Variablen in $s\sigma\rho$ und $t\sigma\rho$ geringer als die von $s\sigma$ und $t\sigma$

  – Der Unifikationsalgorithmus terminiert spätestens dann, wenn $s\sigma$ und $t\sigma$ keine Variablen mehr enthalten

Korrektheit I

- **Beweise Eigenschaft der Differenz unifizierbarer Terme**

  Sind s und t unifizierbar mit Unifikator τ, dann ist $\mathsf{DIFF}(s,t)$ *(leer oder) verhandlungsfähig und jede Reduktion ρ von* $\mathsf{DIFF}(s,t)$ *allgemeiner als τ*

- **Beweis durch Induktion über Termstruktur von s**
  – Ist $s = t$, so ist $\mathsf{DIFF}(s,t) = \emptyset$ und es gibt keine Reduktionen
  – Ist $s \in \mathcal{V}$, so ist $\mathsf{DIFF}(s,t) = \{(s,t)\}$ verhandlungsfähig, da s nicht in t vorkommen darf, und für die Reduktion $\rho = [t/s]$ gilt
  $s\tau = t\tau = s\rho\tau$ und $x\tau = x\rho\tau$ für alle $x \in \mathcal{V}$ mit $x{\neq}s$
  – Ist $s{=}f(s_1, .., s_n)$, dann ist $t{=}f(t_1, .., t_n)$ und alle s_i, t_i unifizierbar mit τ.
  Per Induktionsannahme sind alle $\mathsf{DIFF}(s_i,t_i)$ verhandlungsfähig (oder leer) und jede Reduktion ρ von $\mathsf{DIFF}(s_i,t_i)$ allgemeiner als τ.
  Dasselbe gilt dann auch für $\mathsf{DIFF}(s,t) = \bigcup \mathsf{DIFF}(s_i,t_i)$.

Korrektheit II

- **Es muß nur eine Richtung bewiesen werden**

  $\Rightarrow$: Seien s und t unifizierbar mit Unifikator τ

  Sei σ_i die Substitution nach Runde i, $s_i = s\sigma_i$, $t_i = t\sigma_i$, $D_i = \mathsf{DIFF}(s_i, t_i)$

  Dann ist D_i verhandlungsfähig oder leer und σ_i allgemeiner als τ

  Damit terminiert der Algorithmus nur, wenn $\mathsf{DIFF}(s\sigma,t\sigma) = \emptyset$ ist,

  also $s\sigma = t\sigma$ und σ allgemeiner als jeder Unifikator von s und t.

  $\Leftarrow$: Trivial: ist σ mgu von s und t, dann sind s und t unifizierbar

- **Beweis der Behauptung per Induktion**

  – Es ist $\sigma_0 = \epsilon \geq \tau$, $s_0 = s$, $t_0 = t$, $D_0 = \mathsf{DIFF}(s, t)$.

  Da s und t unifizierbar sind, ist D_0 verhandlungsfähig oder leer

  – Sei D_i verhandlungsfähig oder leer, $\sigma_i\tau_i = \tau$ und ρ Reduktion von D_i

  – Dann sind s_i und t_i unifizierbar mit Unifikator τ_i ($s_i\tau_i = s\tau = t\tau = t_i\tau_i$),

  und damit $\rho \geq \tau_i$, also $\rho\vartheta = \tau_i$ für eine Substitution ϑ.

  Es folgt $\sigma_{i+1}\vartheta = \sigma_i\rho\vartheta = \sigma_i tau_i = \tau$, also $\sigma_{i+1} \geq \tau$ und

  s_{i+1} und t_{i+1} unifizierbar mit ϑ, also D_{i+1} verhandlungsfähig oder leer.

Komplexität des Herbrand–Robinson Verfahrens

- **Occurs-check kann extrem zeitaufwendig werden**

	$P(fxx, fyy, fzz)^T$ und $P(y, z, v)^F$	
	σ	DIFF$(s\sigma, t\sigma)$
0	ϵ	$\{\{fxx, y\}, \{fyy, z\}, \{fzz, v\}\}$
1	$[fxx/y]$	$\{\{f(fxx, fxx), z\}, \{fzz, v\}\}$
2	$[fxx/y,\ f(fxx, fxx)/z]$	$\{\{f(f(fxx, fxx), f(fxx, fxx)), v\}\}$
3	$[fxx/y,\ f(fxx, fxx)/z,\ f(f(fxx, fxx), f(fxx, fxx))/v]$	$\emptyset$

- **Komplexität ist im schlimmsten Fall exponentiell**
  - Zeit für Occurs-check kann für Überprüfung von Konnektionen wie

    $P(fx_1x_1, fx_2x_2, ..., fx_nx_n)^T$ und $P(x_2, ..x_n, y)^F$

    exponentiell in n werden

Komplexität des Herbrand–Robinson Verfahrens II

- **Komplexität ist im Mittel konstant**
  - Nur wenige Terme sind überhaupt unifizierbar (früher Abbruch)
  - "pathologische Fälle" unifizierbarer Terme sind extrem selten
  - In Theorembeweisern dominiert Aufwand für Suche die Unifikation

    Prolog hat fast linearen Suchaufwand, keine Selbstreferenzen $\mapsto$ kein Occurs-check
- **Optimierung durch Pointerverwaltung**
  - Terme werden bei Differenzenbildung nicht explizit eingesetzt
  - Exponentielle Aufblähung wird durch Dag-Darstellung vermieden
  - Occurs-check verfolgt Pointer
  - Ergibt Fast lineare worst-case Komplexität,

    ist aber erheblich aufwendiger zu implementieren

Unifikationsalgorithmus von Martelli-Montanari

- **Interpretiere Unifikation als Gleichungsproblem**
  - Löse eine Gleichung $s\doteq t$
  - Erstelle elementare Gleichungsmenge als hinreichende Bedingung
- **Methode: Schrittweises Umschreiben der Gleichungen**
  - Differenzenbildung und Reduktion werden Transformationsregeln für Gleichungsmengen
- **Prädikatenlogik braucht vier Arten von Regeln**
  - Termdekomposition von $\{f(s_1,..,s_n)\doteq f(t_1,..,t_n)\}$ in $\{s_1\doteq t_1,..,s_n\doteq t_n\}$
  - Variablenelimination entfernt Gleichungen der Form $\{x\doteq t\}$ mit $x \not\in t$ und erweitert σ um $[t/x]$
  - Entfernung trivialer Gleichungen der Form $\{x\doteq x\}$
  - Umstellung von Gleichungen der Form $\{t\doteq x\}$ mit $t \not\in \mathcal{V}$ in $\{x\doteq t\}$

Für komplexere Logiken können weitere Regeln ergänzt werden

Martelli-Montanari Unifikationsalgorithmus

Eingabe *Terme s und t*

Initialisierung *Setze $E \leftarrow \{s \doteq t\}$ und $\sigma \leftarrow \epsilon$*

Transformation *Solange eine der folgenden Regeln anwendbar ist*

TD	$\{f(s_1,..,s_n) \doteq f(t_1,..,t_n)\} \cup E,\ \sigma$	$\leadsto$	$\{s_1 \doteq t_1,..,s_n \doteq t_n\} \cup E,\ \sigma$	
ET	$\{x \doteq x\} \cup E,\ \sigma$	$\leadsto$	$E,\ \sigma$	
VE	$\{x \doteq t\} \cup E,\ \sigma$	$\leadsto$	$E[t/x],\ \sigma[t/x]$	$(x \notin t)$
VE_u	$\{t \doteq x\} \cup E,\ \sigma$	$\leadsto$	$E[t/x],\ \sigma[t/x]$	$(x \notin t, t \notin \mathcal{V})$

Transformiere E, σ entsprechend

Ergebnis *Falls* $E = \emptyset$

dann *σ ist mgu von $\{s, t\}$*

sonst *$\{s, t\}$ ist nicht unifizierbar*

Martelli-Montanari Unifikation am Beispiel

$P(x, fgy, fx)^T$ und $P(h(y,z), fz, fh(u,v))^F$		
	$\{x\doteq h(y,z),\ fgy\doteq fz,\ fx\doteq fh(u,v)\}$,	ϵ
VE	$\{fgy\doteq fz,\ fh(y,z)\doteq fh(u,v)\}$,	$[h(y,z)/x]$
TD	$\{gy\doteq z,\ fh(y,z)\doteq fh(u,v)\}$,	$[h(y,z)/x]$
VE_U	$\{fh(y,gy)\doteq fh(u,v)\}$,	$[h(y,gy)/x, gy/z]$
TD	$\{h(y,gy)\doteq h(u,v)\}$,	$[h(y,gy)/x, gy/z]$
TD	$\{y\doteq u,\ gy\doteq v\}$,	$[h(y,gy)/x, gy/z]$
VE	$\{gu\doteq v\}$,	$[h(u,gu)/x, gu/z, u/y]$
VE_U	$\emptyset$,	$[h(u,gu)/x, gu/\{z,v\}, u/y]$

$P(x, fc)^T$ und $P(fd, x)^F$		
	$\{x\doteq fd,\ x\doteq fc\}$,	ϵ
VE	$\{fd\doteq fc\}$,	$[fd/x]$
TD	$\{d\doteq c\}$, Keine Regel anwendbar	$[fd/x]$

$P(x)^T$ und $P(fx)^F$		
	$\{x\doteq fx\}$, VE nicht anwendbar (occurs check)	ϵ

Terminierung

- **Jede Regel reduziert etwas**
  – VE reduziert Anzahl der Variablen in E, erhöht dabei aber möglicherweise die Größe der vorkommenden Terme
  – TD reduziert die Größe der Terme in E erhöht dabei aber möglicherweise die Anzahl der Gleichungen
  – ET reduziert die Anzahl der Gleichungen in E

- **Ein wohlfundiertes Terminierungsmaß wird kleiner**
  – Wähle als Maß Tripel der Form $(vars, tsize, eqs) \in \mathbb{N}^3$ in lexikographischer Ordnung
  – Dieses Maß wird in jedem Schritt durch Regelanwendungen kleiner
  – Da die lexikographische Ordnung auf $\mathbb{N}^3$ wohlfundiert ist, terminiert die Martelli-Montanari Unifikation

Korrektheit

- **Für unifizierbare Terme s, t wird ein mgu gefunden**
  - Sei E_i, σ_i die Gleichungsmenge/Substitution nach Runde i.

    E_i, σ_i ist äquivalent zu $\{s\dot{=}t\}$ und $\sigma_i \geq \tau$ für jeden Unifikator von s, t

    d.h. $\{s\dot{=}t\}$ und $E_i \cup \sigma_i$ (als Gleichungsmenge) haben dieselben Unifikatoren
  - Sind s, t unifizierbar und $E_i \neq \emptyset$, so ist eine der vier Regeln anwendbar

    Damit terminiert der Algorithmus nur mit $E = \emptyset$ und es ist

    σ Unifikator von s und t und $\sigma_i \geq \tau$ für jeden Unifikator von s, t.

- **Beweis der Behauptung per Induktion**
  - Sei τ Unifikator von s und t
  - Es ist $\sigma_0 = \epsilon \geq \tau$ und $E_0 = \{s\dot{=}t\}$, also E_0, σ_0 äquivalent zu $\{s\dot{=}t\}$
  - Sei $E_i \neq \emptyset$, E_i, σ_i äquivalent zu $\{s\dot{=}t\}$, und $\sigma_i \geq \tau$
    - TD: Es ist $\sigma_{i+1} = \sigma_i \geq \tau$ und $\{s_1\dot{=}t_1, .., s_n\dot{=}t_n\}, \sigma_{i+1}$
      äquivalent zu $\{f(s_1, .., s_n)\dot{=}f(t_1, .., t_n)\}, \sigma_i$
    - ET: Es ist $\sigma_{i+1} = \sigma_i \geq \tau$ und $E_i \setminus \{x\dot{=}x\}, \sigma_{i+1}$ äquivalent zu E_i, σ_i
    - VE: wegen $x\dot{=}t \in E_i$ ist $x\tau = t\tau$, also $[t/x] \geq \tau$ und damit auch
      $\sigma_{i+1} = \sigma_i[t/x] \geq \tau$ und $E_i[t/x], \sigma_{i+1}$ äquivalent zu E_i, σ_i

Komplexitätsaspekte

- **Naive Implementierung wäre exponentiell**

$P(y,z,v)^T$ und $P(fxx,fyy,fzz)^F$	
	$\{y \doteq fxx,\ z \doteq fyy,\ v \doteq fzz\}$
VE	$\{y \doteq fxx,\ z \doteq f(fxx,fxx),\ v \doteq fzz\}$
VE	$\{y \doteq fxx,\ z \doteq f(fxx,fxx),\ v \doteq f(f(fxx,fxx),f(fxx,fxx))\}$

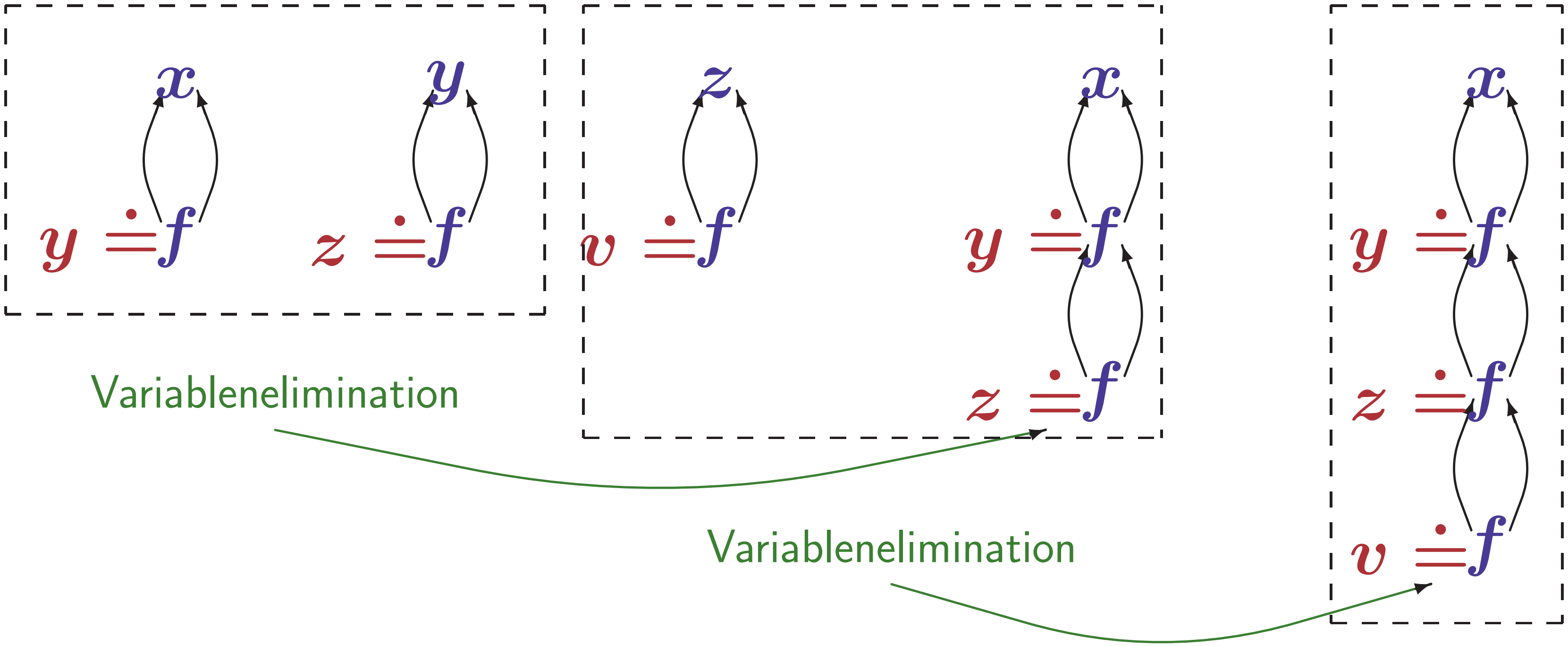
  – Occurs-check in VE wäre extrem zeitaufwendig

- **Höhere Effizienz durch Structure-Sharing**
  – Multigleichung $\{u,v,w\} \doteq hxy$ ersetzt $\{u \doteq hxy, v \doteq hxy, w \doteq hxy\}$
  – Dag-Darstellung von Termen spart Speicherplatz
  und vermeidet unnötige Dopplungen
  – Ermöglicht effizientere Operationen auf Termen und Gleichungen
  Occurs-check prüft mehrfache Teilterme nur einmal
  – Optimierungseffekt entspricht Pointerverwaltung
  und liefert fast lineare worst-case Komplexität

Structure-Sharing bei Martelli-Montanari

	$P(y,z,v)^T$ und $P(fxx, fyy, fzz)^F$
	$\{y\dot= fxx,\ \ z\dot= fyy,\ \ v\dot= fzz\}$
VE	$\{y\dot= fxx,\ \ z\dot= f(fxx,fxx),\ \ v\dot= fzz\}$
VE	$\{y\dot= fxx,\ \ z\dot= f(fxx,fxx),\ \ v\dot= f(f(fxx,fxx),f(fxx,fxx))\}$

Effizienz kommt aus Dag-Darstellung



Eigenschaften des Martelli-Montanari Verfahrens

- **Gut zu verallgemeinern und leicht zu implementieren**
  – Wenn Unifikation mehr als syntaktische Gleichheit verarbeiten soll
    z.B. Unifikation modulo Assoziativität, Kommutativität, Gleichheit,...
  – Auch verwendbar für Spezialunifikation in nichtklassischen Logiken
- **Sehr gute Komplexität**
  – Worst-case Komplexität fast linear
  – Im Mittel konstant, ds die meisten Terme nicht unifizierbar sind
- **Overhead größer als bei Herbrand-Robinson**
  – Aufbau von Structure Sharing etc. erzeugt signifikante Grundkosten
  – Lohnt nicht für Unifikation kleiner Terme
  – Herbrand-Robinson Unifikation ist in der Praxis meist schneller

Unifikation - Zusammenfassung

- **Es gibt viele weitere Unifikationsverfahren**
  - Huet's higher-order Unifikation — fast-linear, keine Entscheidungsprozedur
    Union/Find Algorithmus für den getypten λ-Kalkül
  - Paterson & Wegman's Unifikationsalgorithmus — linear, großer Overhead
  - Escalada-Imaz & Ghallab's Unifikation — fast-linear, geringer Overhead
  - Parallele Unifikationsalgorithmen sind nicht erfolgreich
    Unifikation ist inhärent sequentiell
- **Standard ist Robinson oder Martelli-Montanari**
  - Herbrand-Robinson ist das effektivste Verfahren in der Praxis
  - Martelli-Montanari ist effektiv für erweiterte Fragestellungen