# Inferenzmethoden

## Einheit 6

## Unifikation

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

# Unifikatoren

- **Prädikatenlogische Beweise brauchen Substitution**
  - Konnektierte Literale müssen "gleich" gemacht werden können
  - Substitution darf $\gamma$-Variablen durch Terme ersetzen
  - Dieselbe Substituition muß alle Konnektionen komplementär machen
- **Technisches Problem: Unifikation**
  - Gleichmachen einer Menge von Termen durch Substitution
- **Unifikator einer Menge $S = \{t_1, .., t_n\}$ von Termen**
  - Substitution $\sigma$ mit $\sigma(t_1) = \ldots = \sigma(t_n)$
- **Unifikator von Paarmengen $\{\{s_1, t_1\}, .., \{s_n, t_n\}\}$**
  - Unifikator der Termtupel $(s_1, .., s_n)$, $(t_1, .., t_n)$
- **Allgemeinster Unifikator von $S$ (mgu)**
  - Unifikator $\sigma$, so daß $\tau = \sigma\tau$ für jeden Unifikator $\tau$ von $S$ gilt

**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 $\sigma$ unifiziert

    $\sigma$ kann Erweiterung aller Einzelsubstitutionen sein

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

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

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

$f(gx,ha)$ 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

## Unterschiede zwischen Termen schrittweise auflösen

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

  Menge ungeordneter Termpaare mit der Eigenschaft

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

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

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

  – Elemente sind Paare $\{x,t_0\}$ mit $x\in\mathcal{V}$, $x$ erscheint nicht in $t_0$ (occurs-check)

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

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

  **Unifikation $\hat{=}$ Reduktion verhandlungsfähiger Differenzen**

# Unifikationsalgorithmus von Herbrand–Robinson

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

# Herbrand–Robinson Unifikation am Beispiel

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

| | $\sigma$ | $\mathsf{DIFF}(s\sigma, t\sigma)$ |
|---|---|---|
| 0 | $[]$ | $\{\{x, h(y, z)\}, \{z, gy\}, \{x, h(u, v)\}\}$ |
| 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 | $[gu/z,\ h(u, v)/x,\ u/y,\ gu/v]$ | $\{\}$ |

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

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

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

| | $\sigma$ | $\mathsf{DIFF}(s\sigma, t\sigma)$ |
|---|---|---|
| 0 | $[]$ | $\{\{x, fx\}\}$ |
| 1 | | nicht verhandlungsfähig (occurs-check) |

# Herbrand–Robinson Unifikation am Beispiel

| | $P(fxx, fyy, fzz)^T$ und $P(y, z, v)^F$ | |
|---|---|---|
| | $\sigma$ | $\mathsf{DIFF}(s\sigma, t\sigma)$ |
| 0 | $[]$ | $\{\{fxx, y\}, \{fyy, z\}, \{fzz, v\}\}$ |
| 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 | $[fxx/y,\ f(fxx, fxx)/z,$ $f(f(fxx, fxx), f(fxx, fxx))/v]$ | $\{\}$ |

**Zeit für Occurs-check wächst exponentiell**

# Eigenschaften des Herbrand–Robinson Verfahrens

- **Exponentiell im schlimmsten Fall**
  - Occurs-check kann extrem zeitaufwendig werden, wenn Terme wachsen

    Beispiel: $P(fx_1x_1, fx_2x_2, ..., fx_nx_n)^T$ und $P(x_2, ..x_n, y)^F$
- **Konstant im Mittel**
  - Nur wenige Terme sind überhaupt unifizierbar (früher Abbruch)
  - "pathologische Fälle" unifizierbarer Terme sind extrem selten
- **Optimierung durch Pointerverwaltung**
  - Terme werden bei Differenzenbildung nicht explizit eingesetzt
  - Exponentielle Aufblähung wird durch Dag-Darstellung vermieden
  - Occurs-check verfolgt Pointer
  - Aufwendiger zu implementieren
  - Fast linear auch im schlimmsten Fall

# Unifikationsalgorithmus von Martelli-Montanari

- **Interpretiere Unifikation als Gleichungsproblem**
  - Löse eine Gleichung $s \doteq t$
  - Erstelle elementare Gleichungsmenge als hinreichende Bedingung
- **Schrittweises Umschreiben der Gleichungen**
  - Differenzenbildung und Reduktion werden Transformationsregeln für Gleichungsmengen
- **Effiziente Datenstruktur**
  - Structure-Sharing in Termen und Gleichungen
- **Gut zu verallgemeinern**
  - Wenn Unifikation mehr als syntaktische Gleichheit verarbeiten soll z.B. Unifikation modulo Assoziativität, Kommutativität, Gleichheit,...

# Unifikationsalgorithmus von Martelli-Montanari

**Gegeben:** **Zu lösende Gleichung $s \doteq t$**

**Ziel:** **Idempotente Gleichungsmenge $\{x_1 \doteq t_1, \ldots, x_k \doteq t_k\}$** $(x_i \notin t_j)$

**Methode:** **Anwendung von Transformationsregeln**

**Regeln:** **Termdekomposition**

$$\{f(s_1, \ldots, s_n) \doteq f(t_1, \ldots, t_n)\} \cup E \leadsto \{s_1 \doteq t_1, \ldots, s_n \doteq t_n\} \cup E.$$

**Entfernung trivialer Gleichungen**

$$\{x \doteq x\} \cup E \leadsto E$$

**Umstellung**

$$\{t \doteq x\} \cup E \leadsto \{x \doteq t\} \cup E, \qquad \text{wenn } t \notin \mathcal{V}$$

**Variablenelimination**

$$\{x \doteq t\} \cup E \leadsto \{x \doteq t\} \cup E[t/x], \qquad \text{wenn } x \notin t \text{ und } x \in E$$

# Eigenschaften des Martelli-Montanari Verfahrens

- **Korrektheit**
  - Termdekomposition zerlegt Gleichungen, bis ein Term Variable ist oder Terme als nicht unifizierbar identifiziert
  - Umstellung stellt Variablen auf linke Seite
  - Variablenelimination instantiiert restliche Gleichungen
  - Iteration ergibt Menge von Gleichungen $x_i \doteq t_i$ mit $x_i \in \mathcal{V}$ und $x_i \notin t_j$
- **Effizienz durch Datenstruktur Multigleichung**
  - Dag-Darstellung von Termen und Gleichungen
  - $\{u, v, w\} \doteq hxy$ ersetzt mehrere Gleichungen $u \doteq hxy, v \doteq hxy, w \doteq hxy$
  - Optimierungseffekt ähnlich zur Pointerverwaltung
- **Komplexität fast linear**
  - Auch im schlimmsten Fall

# Martelli-Montanari Unifikation am Beispiel

| | |
|---|---|
| $P(x, fgy, fx)^T$ und $P(h(y,z), fz, fh(u,v))^F$ | |
| | $\{x \doteq h(y,z),\ fgy \doteq fz,\ fx \doteq fh(u,v)\}$ |
| Termdekomposition | $\{x \doteq h(y,z),\ gy \doteq z,\ x \doteq h(u,v)\}$ |
| Variablenelimination | $\{x \doteq h(y,z),\ gy \doteq z,\ h(y,z) \doteq h(u,v)\}$ |
| Termdekomposition | $\{x \doteq h(y,z),\ gy \doteq z,\ y \doteq u,\ z \doteq v\}$ |
| Umstellung | $\{x \doteq h(y,z),\ z \doteq gy,\ y \doteq u,\ z \doteq v\}$ |
| Variablenelimination | $\{x \doteq h(y,gy),\ z \doteq gy,\ y \doteq u,\ gy \doteq v\}$ |
| Umstellung | $\{x \doteq h(y,gy),\ \{z,v\} \doteq gy,\ y \doteq u\}$ |
| Variablenelimination | $\{x \doteq h(u,gu),\ \{z,v\} \doteq gu,\ y \doteq u\}$ |
| $P(x, fc)^T$ und $P(fd, x)^F$ | |
| | $\{x \doteq fd,\ x \doteq fc\}$ |
| Variablenelimination | $\{x \doteq fd,\ fd \doteq fc\}$ |
| Termdekomposition | $\{x \doteq fd,\ d \doteq c\}$ Keine Regel anwendbar |
| $P(x)^T$ und $P(fx)^F$ | |
| | $\{x \doteq fx\}$ Zielsituation nicht erreicht |

# Martelli-Montanari Unifikation am Beispiel

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

## Effizienz kommt aus Dag-Darstellung