

Inferenzmethoden

Teil V

Behandlung spezifischer Fragestellungen

Spezialisierte Beweistechniken



1. Verarbeitung mathematischer Theorien
2. Gleichheitsbehandlung
3. Zahlen und Induktion
4. Termersetzung und -auswertung

PRÄDIKATENLOGIK HAT GRENZEN

- **Zu wenig vorgegeben Strukturen**

- Keine Schlüsse über **Werte** von Termen möglich

- Interpretation von Gleichheit (z.B. $4+4=8$) ist nicht festgelegt

- Kein Schließen über **Datentypen** möglich

- Interpretation von $\forall x \ x=0 \vee x \geq 1$ nicht festgelegt

In informaler mathematischer Logik gelten diese Konzepte als fundamental

- **Erweiterung der Logik durch Axiome unpraktisch**

- + alle guten Eigenschaften der Logik bleiben erhalten

- Formales Schließen mühsam (zu viele Teilformeln)

- **Erweiterung von Semantik und Inferenzsystem**

- Mehr Theorie: Korrektheit, Vollständigkeit muß neu bewiesen werden

- + Formales Schließen “natürlich” und einfacher

Mehr in “Automatisierte Logik und Programmierung”

- **Entscheidungsprozeduren**

- Testverfahren für Gültigkeit spezieller Formelklassen
 - Folgt eine Gleichheit aus mehreren anderen
 - Folgt eine arithmetische Aussage aus mehreren anderen
- **Sehr effizient** für spezielle Anwendungsgebiete
- **Integration in Konnektionsmethode und Resolution schwierig**
 - Komplementaritätsbegriff muß erweitert werden
 - Betrachtung von mehr als zwei Literalen macht Suche ineffizienter

- **Rewrite-Verfahren**

- Bestimmung des Wertes von Termen durch “Umschreibung”
- Anwendung algebraischer Gesetze als Transformationsregeln
- Gleichungen erhalten eine Richtung
- **Integration in Unifikationsalgorithmus möglich**

Wie kann man dies effizient in der Beweissuche einsetzen?

Inferenzmethoden

Einheit 14

Theorie- und Gleichheitsbehandlung



1. Theorien & Unifikationstheorie
2. Gleichheitsbehandlung in Beweisverfahren
3. Entscheidungsprozeduren für Gleichheit

Verarbeitung mathematischer Standardtheorien

- **Theorien werden beschrieben durch Axiome**
 - “Grundwahrheiten” der Theorie, aus denen alles andere folgt
 - Reflexivität, Symmetrie, Transitivität, Substitutivität für Gleichheit
 - Assoziativität, Identität, Inverse für Gruppen
 - Peano-Axiome für natürliche Zahlen
- **Theorien verwenden oft spezielle Inferenzketten**
 - Standardargumente, die Axiome effizient in Schlußfolgerungen einsetzen
 - z.B. gezieltes Einsetzen von Substitutivität beim Gleichheitsschließen
- **Allgemeine Theoremeweiser unterstützen dies nicht**
 - Theoriespezifische Inferenzen passen nicht zum allgemeinen Verfahren
 - Beweiser müssen Axiome als zusätzliche Klauseln hinzunehmen
 - Alternative ist Integration der Theorie in die Unifikation

Beweise $Pa \wedge a=b \Rightarrow Pb$ aus Gleichheitsaxiomen

• Integration der Axiome in die Matrix

$$\left[\begin{array}{ccccccc} Pa^T & a=b^T & x=y^F & x=y^F & x=x^T & x=y^F & Pb^F \\ & & y=z^F & y=x^T & & Px^F & \\ & & x=z^T & & & Py^T & \end{array} \right]$$

2
1
3

- Beweissuchverfahren kann unverändert bleiben
- Zusätzliche Klauseln erhöhen Anzahl der Suchschritte erheblich

• Einbettung der Theorie durch spezielle Konnektionen

$$\left[\begin{array}{ccc} & \boxed{EQ} & \\ \text{1} & \text{2} & \text{3} \\ Pa^T & a=b^T & Pb^F \end{array} \right] \quad \text{wobei } \boxed{EQ} \equiv \left[\begin{array}{cccc} x=y^F & x=y^F & x=x^T & x=y^F \\ y=z^F & y=x^T & & Px^F \\ x=z^T & & & Py^T \end{array} \right]$$

- **Theoriekonnektion** beinhaltet Inferenzschritte der Theorie EQ
- Suchverfahren & Unifikationsmechanismus müssen angepasst werden

THEORIEKONNEKTIONEN

- **Erweiterter Komplementaritätsbegriff für Theorien**

- $P t_1^T$ und $P t_2^F$ sind **komplementär in der Theorie \mathcal{T}** , wenn $\sigma(t_1)$ und $\sigma(t_2)$ in \mathcal{T} gleich sind (σ zulässige Substitution)
- Erlaubt Verarbeitung theoriespezifischer Inferenzketten

- **Unifikation wird mehr als syntaktisches Gleichmachen**

$$\left[\begin{array}{c} \text{Arith} \\ P((x+x)-1)^T \end{array} \quad P(1)^F \right] \quad \sigma = [1/x]$$

- Konnektion benutzt Unifikation für einfache Arithmetik

$$\left[\begin{array}{c} \text{Group} \\ R(c, b)^T \end{array} \quad R(z \cdot (\bar{c} \cdot c), z \cdot (\bar{z} \cdot b))^F \right] \quad \sigma = [c/z]$$

- Konnektion benutzt Unifikation für Gruppentheorie

- **Allgemeiner Mechanismus noch wenig erforscht**

- Meist Integration von **Theorie-Unifikation** in konventionelle Beweiser
- Konnektionen können auch **unär**, **ternär** oder komplexer sein

ERWEITERUNG: GERICHTETE THEORIEKONNEKTIONEN

- **Theorieimplikation** $\Rightarrow_{\mathcal{T}}$
 - Implikation die in der Theorie \mathcal{T} gültig ist
- **Gerichtete σ -komplementäre Konnektion** (L^T, L'^F)
 - Es gilt $\sigma(L)=\sigma(L')$ oder $\sigma(L) \Rightarrow_{\mathcal{T}} \sigma(L')$
 - Richtung geht immer von Polarität T nach F
- **Unäre σ -komplementäre Konnektion** L^T oder L'^F
 - Es gilt $\sigma(L) \Rightarrow_{\mathcal{T}} \text{False}$ bzw. $\text{True} \Rightarrow_{\mathcal{T}} \sigma(L')$
 - Gültigkeit folgt alleine aus der Theorie, ohne Gegenliteral



Eine Formel F ist gültig in der Theorie \mathcal{T} , wenn es eine Multiplizität μ , eine zulässige Substitution σ und eine Menge \mathcal{C} von bezüglich \mathcal{T} σ -komplementären gerichteten Konnektionen gibt, so daß jeder Pfad durch F eine Konnektion aus \mathcal{C} enthält

- **Es gibt noch viele offene Fragen**

- Wie genau **Unifizierbarkeit modulo Theorie \mathcal{T}** definieren?
- Gibt es mgu's und, wenn ja, wieviele?
- Gibt es **Unifikationsalgorithmen** (als Entscheidungsprozeduren)?
- Was ist die **Komplexität** des Unifikationsverfahrens?

- **Bisher gibt es nur wenig allgemeine Lösungen**

- Erfolgreich nur für spezielle Theorien (Gleichheit, Gruppen, ...)

- **Es gibt verschiedene Typen von Unifikationstheorien**

- **unitär**: Genau ein allgemeinsten Unifikator (z.B. Standard-Unifikation)
- **finitär**: endlich viele allgemeinste Unifikatoren (z.B. AC-/Präfix-Unifikation)
- **infinite**: unendlich viele mgu's (z.B. Unifikation modulo Assoziativität)
- **leer**: keine allgemeinsten Unifikatoren (wenig erwünscht)

- **Wichtigste Grundbeziehung zwischen Objekten**

- Spezialbehandlung sehr lohnenswert
- Dargestellt als zweistelliges Prädikat zwischen Termen
- Prädikatszeichen \doteq in Infix-Notation

- **Charakterisiert durch 5 Grundeigenschaften**

- $x \doteq x$

Reflexivität

- $x \doteq y \Rightarrow y \doteq x$

Symmetrie

- $x \doteq y \wedge y \doteq z \Rightarrow x \doteq z$

Transitivität

- $x_i \doteq y \Rightarrow f(x_1, \dots, x_i, \dots, x_n) \doteq f(x_1, \dots, y, \dots, x_n)$

Substitutivität auf Funktionen (Schema)

- $x_i \doteq y \Rightarrow [P(x_1, \dots, x_i, \dots, x_n) \Leftrightarrow P(x_1, \dots, y, \dots, x_n)]$

Substitutivität auf Prädikaten (Schema)

Symmetrie und Transitivität sind ableitbar

Erweitere Formel um Axiome der Gleichheit

- **Verwende minimale Axiomenmenge**

- $x \doteq x$

- $x_i \doteq y \Rightarrow f(x_1, \dots, x_i, \dots, x_n) \doteq f(x_1, \dots, y, \dots, x_n)$

- $x_i \doteq y \Rightarrow [P(x_1, \dots, x_i, \dots, x_n) \Rightarrow P(x_1, \dots, y, \dots, x_n)]$

Das Schema der Substitutivität muß für jedes vorkommende Funktions- und Prädikatssymbol instantiiert werden

- **Erhebliche Vergrößerung des Suchraums**

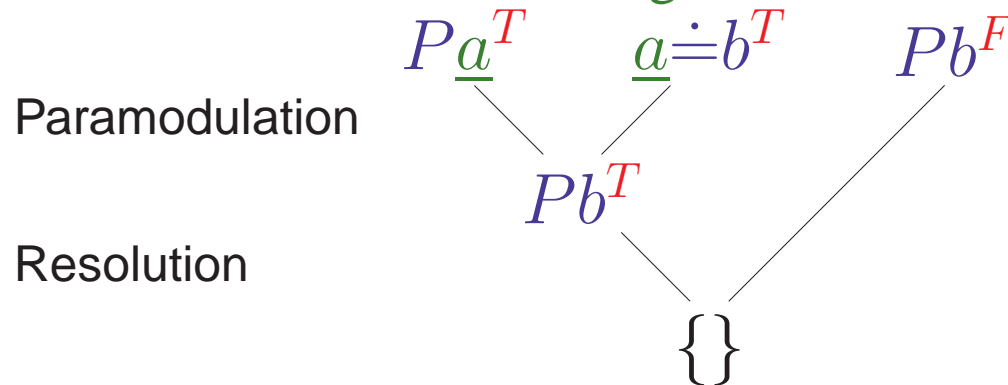
- Unbrauchbar für komplexe Formeln

- In der Praxis bei kleinen Formeln oft effizienter als Spezialverfahren

• Resolutionsähnliche Kalkülregel **Paramodulation**

- Wähle zwei **Elternklauseln** $\{L\} \cup C_1$ und $\{r \doteq s\} \cup C_2$ und **kennzeichne** einen Teilterm t von L
- Bestimme **allgemeinsten Unifikator** σ von r und t
- Generiere **Paramodulant** $\sigma(\{L'\} \cup C_1 \cup C_2)$, wobei $L' = "L[s/t]"$

Zusammen mit Resolution vollständig für Prädikatenlogik mit Gleichheit

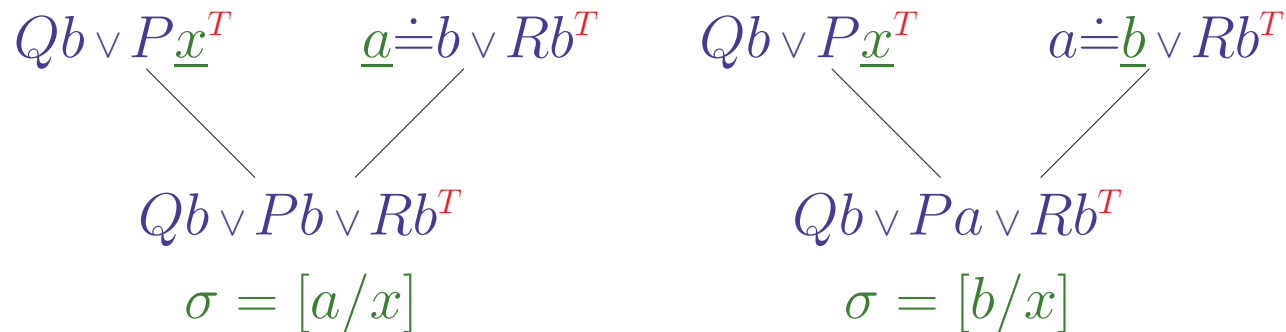


• **Alternative Sicht: bedingte Termersetzungsregel**

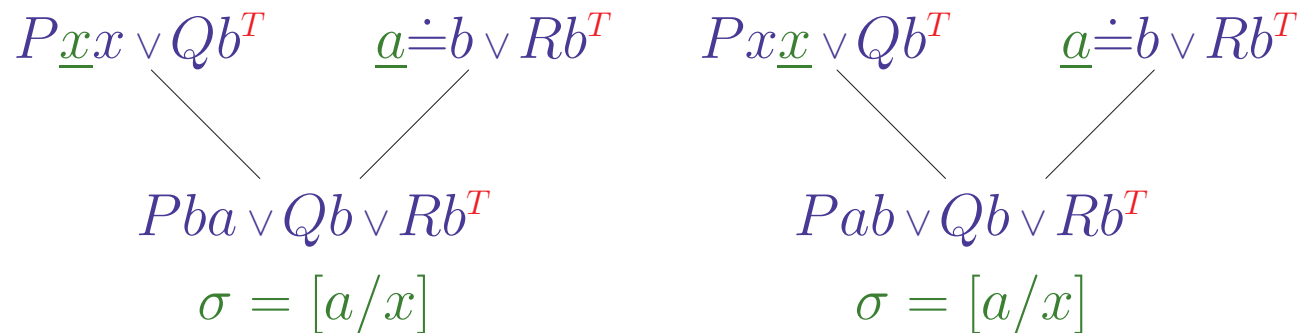
↪ Einheit 16

- C_2 ist Bedingung für die Ersetzung von r durch s in einem Literal L

- Welche **Richtung** einer Gleichheit wird gewählt?



- Welches **Variablenvorkommen** wird ersetzt?



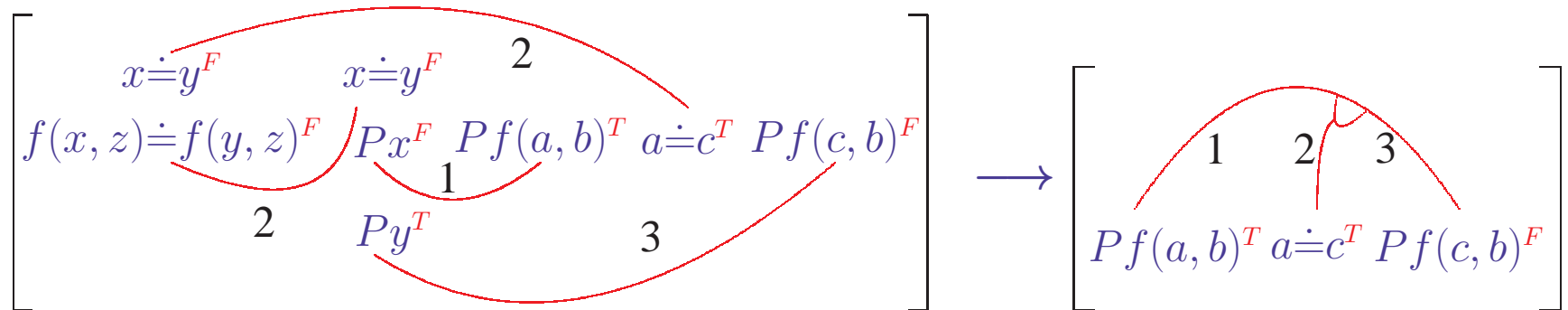
- **Lokale Sicht bedeutet großen Suchraum**

- Viele Paramodulanten möglich
- Effiziente Suchstrategien erforderlich

GLEICHHEITSKONNEKTIONEN

- **Axiomatische Gleichheitsbehandlung ist aufwendig**

- Einfache Beweise wie $Pf(a, b) \wedge a \doteq c \Rightarrow Pf(c, b)$ werden umständlich



- Menschen gehen direkter mit Gleichheiten um

- **Verdichte Beweisführung durch eq-Konnektion**

- Konnektion verbindet Literalpaar und ein oder mehrere Gleichungen

- Unifikation darf konnektierte Gleichheiten berücksichtigen

- **Strategische Steuerung wird aufwendiger**

- Welche Gleichheiten sind geeignet? (i.a. **unentscheidbares Problem**)

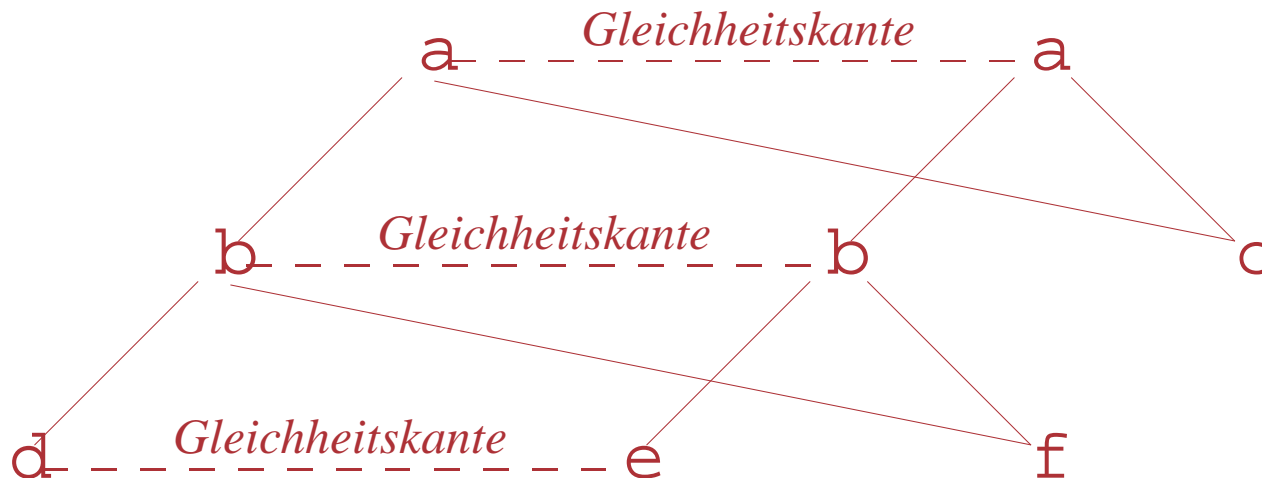
- Sehr kompliziert, wenn gleichzeitig Substitutionen zu bestimmen sind

Folgt eine Gleichheit aus anderen Gleichheiten?

- **Wichtig für praktische Beweisführung**
 - z.B.: $f(f(a, b), b) \doteq a$ folgt aus $f(a, b) \doteq a$
 $g(a) \doteq a$ folgt aus $g(g(g(a))) \doteq a$ und $g(g(g(g(g(a)))))) \doteq a$
 - Intuitiver Beweis (gezieltes Einsetzen) einfach
- **Quantorenfreie Gleichheit ist entscheidbar**
 - Einfache Theorie: Gleichheiten mit uninterpretierten Symbolen
 - Semantik: Reflexivität, Symmetrie, Transitivität, Substitution
- **Effiziente Verfahren verfügbar**
 - Berechnung der transitiven Hülle einer Äquivalenzrelation
 - Technisch: Kongruenzabschluß des Relationsgraphen
- **Entscheidungsprozedur ist keine Unifikation**
 - Verfahren überprüft Gleichheiten, aber instantiiert keine Variablen

GLEICHHEITSSCHLIESSEN DURCH KONGRUENZABSCHLUSS

Zeige: $a(b(d, f), c) \doteq a(b(e, f), c)$ **folgt aus** $d \doteq e$



1. **Verschmelze identische Knoten**
2. **Verbinde gleiche Knoten durch Gleichheitskante**
3. **Verbinde Wurzeln von Teilbäumen, die in allen Knoten gleich sind**
Gleichheit $\hat{=}$ Wurzeln der Termbäume sind verbunden

- **Notationen für gerichtete Graphen $G = (V, E)$**
 - $l(v)$: Markierung des Knoten v in G
 - $\delta(v)$: Anzahl der von v ausgehenden Kanten
 - $v[i]$: i -ter Nachfolgerknoten von v
 - u **Vorgänger** von v , wenn $v = u[i]$ für ein i
- **Begriffe für Äquivalenzrelationen R auf V**
 - **u und v kongruent unter R ($u \sim_R v$):**
 $l(u) = l(v)$, $\delta(u) = \delta(v)$ und für alle i $(u[i], v[i]) \in R$
 - **R abgeschlossen unter Kongruenzen:** $u \sim_R v \Rightarrow (u, v) \in R$
 - **Kongruenzabschluß R^* :** eindeutige minimale Erweiterung von R , die abgeschlossen unter Kongruenzen und Äquivalenzrelation ist
 $\hat{=}$ Menge aller Äquivalenzen, die logisch aus R folgen

Folgt $s \doteq t$ aus $s_1 \doteq t_1, \dots, s_n \doteq t_n$?

- **Konstruiere Graph G von $s, s_1, \dots, s_n, t, t_1, \dots, t_n$**
 - G besteht aus Termbäumen von $s, s_1, \dots, s_n, t, t_1, \dots, t_n$
 - Identische Teilausdrücke werden durch denselben Teilbaum dargestellt
- **Bestimme Kongruenzabschluß aller $s_i = t_i$ iterativ**
 - Start: R ist Identitätsrelation auf den Knoten von G ($R^* = R$)
 - Im Schritt i bestimme Kongruenzabschluß von $R^* \cup \{(\tau(s_i), \tau(t_i))\}$
($\tau(u)$: Wurzelknoten des Termbaums von u)
 - Repräsentiere R^* als Menge von Äquivalenzklassen $\{ [u]_R \mid u \in V \}$
($[u]_R \equiv \{x \in V \mid (x, u) \in R\}$)
- **Teste Äquivalenz von s und t**
 - $s = t$ gilt genau dann, wenn $(\tau(s), \tau(t)) \in R^*$

BERECHNE KONGRUENZABSCHLUSS VON $R \cup \{(u, v)\}$

- **Algorithmus MERGE(R, u, v)**

- Eingabe: gerichteter Graph $G = (V, E)$, $u, v \in V$

- Äquivalenzrelation R (abgeschlossen unter Kongruenzen)

- **Falls $u \sim_R v$, dann halte mit Ergebnis R**

- Es gilt $(R \cup \{(u, v)\})^* = R$

- **Andernfalls modifiziere R durch Verschmelzung**

- Setze $P_u := \{x \in V \mid \exists w \in [u]_R. x \text{ Vorgänger von } w\}$

- Setze $P_v := \{x \in V \mid \exists w \in [v]_R. x \text{ Vorgänger von } w\}$

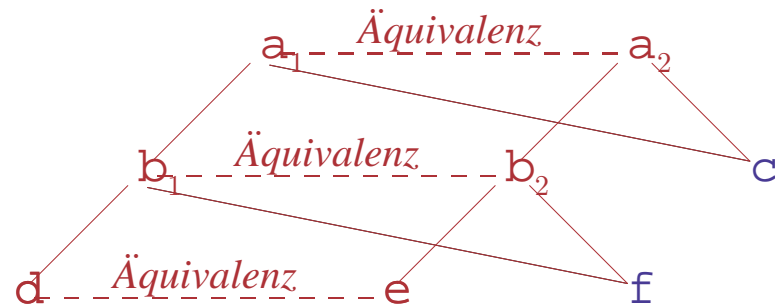
- Vereinige Äquivalenzklassen $[u]_R$ und $[v]_R$ in R

- Wiederhole für $x \in P_u$ und $y \in P_v$

- Falls $x \sim_R y$ und $[x]_R \neq [y]_R$ dann setze $R := \text{MERGE}(R, x, y)$

- **Halte mit der modifizierten Relation R als Ergebnis**

KONGRUENZABSCHLUSS: $d \doteq e \Rightarrow a(b(d, f), c) \doteq a(b(e, f), c)$



- **Graph ist Termbaum von $a(b(d, f), c)$ und $a(b(e, f), c)$**

- Identische Teilausdrücke benutzen denselben Teilbaum

- Initiale Relation: $R := \{ \{a_1\}, \{a_2\}, \{b_1\}, \{b_2\}, \{c\}, \{d\}, \{e\}, \{f\} \}$

- **Hinzunahme von $d \doteq e$**

Bestimme Vorgänger von $[d]_R$ ($\{b_1\}$) und $[e]_R$ ($\{b_2\}$)

- Vereinige $[d]_R$ und $[e]_R$: $R := \{ \{a_1\}, \{a_2\}, \{b_1\}, \{b_2\}, \{c\}, \{d, e\}, \{f\} \}$

Bestimme Vorgänger von $[b_1]_R$ ($\{a_1\}$) und $[b_2]_R$ ($\{a_2\}$)

- Vereinige $[b_1]_R$ und $[b_2]_R$: $R := \{ \{a_1\}, \{a_2\}, \{b_1, b_2\}, \{c\}, \{d, e\}, \{f\} \}$

Bestimme Vorgänger von $[a_1]_R$ (\emptyset) und $[a_2]_R$ (\emptyset)

- Vereinige $[a_1]_R$ und $[a_2]_R$: $R := \{ \{a_1, a_2\}, \{b_1, b_2\}, \{c\}, \{d, e\}, \{f\} \}$

Wurzelknoten der beiden Terme sind äquivalent

KONGRUENZABSCHLUSS: $g(g(g(a))) \doteq a$, $g(g(g(g(g(a)))) \doteq a$

- **Graph ist Termbaum von $g(g(g(g(g(a))))$**

- Initiale Relation: $R := \{ \{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_5\}, \{v_6\} \}$

- **Hinzunahme von $g(g(g(g(g(a)))) \doteq a$**

- $R := \{ \{v_1, v_6\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_5\} \}$ ist abgeschlossen

- **Hinzunahme von $g(g(g(a))) \doteq a$**

MERGE(R, v_3, v_6):

- $P_{v_3} := \{v_2\}$, $P_{v_6} := \{v_5\}$, $R := \{ \{v_1, v_6, v_3\}, \{v_2\}, \{v_4\}, \{v_5\} \}$

- Wegen $(v_3, v_6) \in R$ gilt $v_2 \sim_R v_5$ aber $[v_2]_R \neq [v_5]_R$

MERGE(R, v_2, v_5):

- $P_{v_2} := \{v_1\}$, $P_{v_5} := \{v_4\}$, $R := \{ \{v_1, v_6, v_3\}, \{v_4\}, \{v_2, v_5\} \}$

- Wegen $(v_2, v_5) \in R$ gilt $v_1 \sim_R v_4$ aber $[v_1]_R \neq [v_4]_R$

MERGE(R, v_1, v_4):

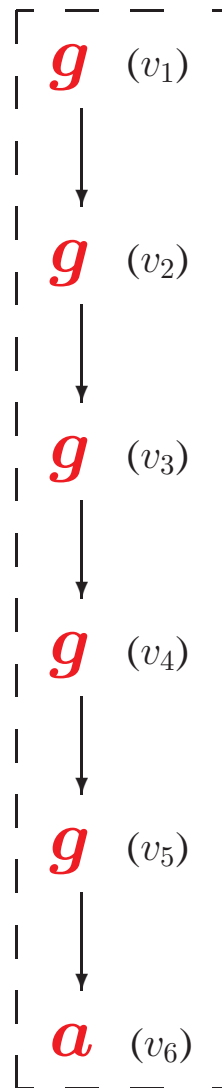
- $P_{v_1} := \{v_2, v_5\}$, $P_{v_4} := \{v_3\}$, $R := \{ \{v_1, v_6, v_3, v_4\}, \{v_2, v_5\} \}$

- Wegen $(v_6, v_4) \in R$ gilt $v_5 \sim_R v_3$ aber $[v_5]_R \neq [v_3]_R$

MERGE(R, v_5, v_3):

- $P_{v_5} := \{v_1, v_4\}$, $P_{v_3} := \{v_2, v_5, v_3\}$, $R := \{ \{v_1, v_6, v_3, v_4, v_2, v_5\} \}$

Alle Knoten sind äquivalent: $R=R^*$



- **Entscheidungsprozedur ist sehr effizient**
- **Kein Unifikationsverfahren**
 - Variablen werden nicht instantiiert
- **Direkt verwendbar für binäre Konnektionen**
 - Erweitert Unifikation um direkte Gleichheitsschlüsse
- **Unäre, ternäre etc. Schlüsse schwer zu integrieren**
 - Beweissuchverfahren müsste Multi-Konnektionen untersuchen anstelle der viel einfacheren Binärkonnektionen
 - **Strategische Steuerung wird aufwendig**
 - Welche Literale sind relevant?
 - Analyse verlangsamt Suchverfahren massiv

Immer noch offenes Forschungsthema