Inferenzmethoden

Einheit 13

Zahlen und Induktion

1. Axiomatische Induktionsbehandlung
2. Induktion mit Theoriekonnektionen
3. Induktionslose Induktion

Induktion

Essentiell für mathematische Beweisführung

- **Ermöglicht Schlüsse über unendliche Konzepte**
  - Aussagen über beliebige Zahlen, Listen, Bäume, Graphen, Mengen, ...
  - Eigenschaften von Programmen (unabhängig von der konkreten Eingabe)
- **Grundform: schrittweise Induktion über $\mathbb{N}$**
  - Gilt $P(0)$ und folgt aus $P(x)$ immer $P(x+1)$, so gilt P für alle Zahlen
  - Übertragbar auf Listen, Bäume, Strings als strukturelle Induktion
- **Allgemeine Form: strukturelle Induktion**
  - Für Konzepte mit aufwendigerer rekursiver Definition
  - Gilt $P([])$ und folgt $P(a.l)$ aus $P(l)$ für jedes a, so gilt P für alle Listen
  - Gilt $P(\epsilon)$ und folgt $P(wa)$ aus $P(w)$ für jedes a, so gilt P für alle Strings
- **Erweiterung: wohlfundierte Induktion**
  - Reduktion des Problems mit wohlfundierter Ordnung $\succ$
  - Folgt $P(x)$ wenn $P(y)$ für alle $x \succ y$ gilt, so gilt P für alle Elemente
  - Wichtig, wenn Beweisargument "Rückwärtssprünge" macht

Axiomatische Definition natürlicher Zahlen

- **Fest definierte Prädikats- und Funktionssymbole**
  - $\boldsymbol{N(x)}$: x ist eine natürliche Zahl
  - **0**: Konstante Null
  - $\boldsymbol{x'}$: Postfix-Anwendung der Nachfolgerfunktion auf x
- **Induktionsaxiome für natürliche Zahlen**

$N(0)$	Erzeugungsaxiom für Null
$\forall x[N(x) \Rightarrow N(x')]$	Erzeugungsaxiom für Nachfolger
$\forall x[N(x) \Rightarrow x' \not\doteq 0]$	Eindeutigkeitsaxiom für Null
$\forall xy[N(x) \wedge N(y) \Rightarrow (x' \doteq y' \Rightarrow x \doteq y)]$	Eindeutigkeitsaxiom für Nachfolger
$P[0/x] \wedge \forall y[N(y) \Rightarrow (P[y/x] \Rightarrow P[y'/x])]$ $\Rightarrow \forall x(N(x) \Rightarrow P)$	Induktionsschema für jedes Prädikat zu instantiieren

x	*Induktionsvariable*
$P[0/x]$	*Induktionsanfang*
$[N(y) \Rightarrow (P[y/x] \Rightarrow P[y'/x])]$	*Induktionsschluß*
$P[y/x]$	*Induktionshypothese*
$P[y'/x]$	*Induktionskonklusion*

Hinzunahme von Induktionsaxiomen zur Formel

- **Beispiel:** $x\not\doteq 0 \Rightarrow \exists z(Nz \wedge x\doteq z')$

  Ergänze Gleichheits- und Zahlenaxiome; instantiiere Induktionsschema

$$
\begin{array}{ll}
 & \forall u\; u\doteq u \\
\wedge & \forall v\; (N(v) \Rightarrow N(v')) \\
\wedge & \{[0\not\doteq 0 \Rightarrow \exists v(Nv \wedge 0\doteq v')] \wedge \forall b[Nb \Rightarrow ((b\not\doteq 0 \Rightarrow \exists c(Nc \wedge b\doteq c')) \Rightarrow (b'\not\doteq 0 \Rightarrow \exists y(Ny \wedge b'\doteq y')))] \\
 & \quad \Rightarrow \forall x[Nx \Rightarrow (x\not\doteq 0 \Rightarrow \exists d(Nd \wedge x\doteq d'))]\} \\
\Rightarrow & \forall a[Na \Rightarrow (a\not\doteq 0 \Rightarrow \exists z(Nz \wedge a\doteq z'))]
\end{array}
$$

- **Matrix-Beweis in Nicht-Normalform**

$\sigma_2 = [b'/u, b/y, a/x, d/v, d'/z]$

$$
\left[\begin{array}{cccccc}
u\doteq u^T &
\left[\begin{array}{l}
\left[\begin{array}{cc} 0\doteq 0^F & \left[\begin{array}{c} Nv^F \\ 0\doteq v'^F \end{array}\right] \end{array}\right] \\
\left[\begin{array}{cccc} Nb^T & \left[\begin{array}{c} b\doteq 0^T \\ \left[\begin{array}{cc} Nc^T & b\doteq c'^T \end{array}\right] \end{array}\right] & b'\doteq 0^F & \left[\begin{array}{c} Ny^F \\ b'\doteq y'^F \end{array}\right] \end{array}\right] \\
\left[\begin{array}{c} Nx^F \\ x\doteq 0^T \\ \left[\begin{array}{cc} Nd^T & x\doteq d'^T \end{array}\right] \end{array}\right]
\end{array}\right] &
Na^T & a\doteq 0^F &
\left[\begin{array}{c} Nz^F \\ a\doteq z^F \end{array}\right] &
\left[\begin{array}{c} Nv^F \\ Nv'^T \end{array}\right]
\end{array}\right]
$$

Suchentscheidungen für Induktionsbeweise

Automatisierung von Induktionsbeweisen schwierig

- **Zusätzliche Alternativen bei der Beweisführung**
  1. Ist es nötig, einen Induktionsbeweis zu führen?
  2. Ist eine Verallgemeinerung der zu beweisenden Aussage nötig?
  3. Welche Teilformel ist als Induktionsformel auszuwählen?
  4. Welche Variable der Induktionsformel soll die Induktionsvariable sein ?
  5. Muß eine geschachtelte Induktion durchgeführt werden?
- **Ergibt Suchraum von beträchtlichem Ausmaß**
  – Fragen 1,2 nur vom menschlichem Systembenutzer zu entscheiden
  – Induktionsformel muß engen Zusammenhang zum Beweisziel haben ($\mapsto$ 3)
  – Anzahl der möglichen Induktionsvariablen (echte Alternativen) ist klein
  – Geschachtelte Induktionen nur, wenn weitere Variablen im Induktionsschluß
- **Stärkere heuristische Steuerung möglich**
  – Strukturanalyse liefert Menge revelanter (Theorie-)Konnektionen

Verfeinere Matrixcharaktisierung für Induktionsschritt

- **Induktionsschritt $P[y/x] \Rightarrow P[y'/x]$ ist gerichtet**
  - $P[y'/x]$ muß aus $P[y/x]$ arithmetisch folgen
  - Gerichtete Konnektionen mit Theorieimplikationen ersetzen Unifikatoren
- **$P[y'/x]$ ist strukturell ähnlich zu $P[y/x]$**
  - Teilformeln von $P[y'/x]$ entsprechen denen von $P[y/x]$
  - "Orthogonale" Konnektionen zwischen diesen Teilformeln reichen aus
- **$P[y/x] \Rightarrow P[y'/x]$ kann Fallanalyse benötigen**
  - z.B. bei $\exists y_h\; x \geq y_h^2 \wedge x < (y_h+1)^2 \;\Rightarrow\; \exists y\; x+1 \geq y^2 \wedge x+1 < (y+1)^2$ muß $x+1 \geq (y_h+1)^2$ und $x+1 < (y_h+1)^2$ unterschieden werden
  - Erlaube verschiedene (Teil-)Beweise unter verschiedenen Constraints
  - Disjunktion aller Constraints muß allgemeingültig sein
  - Constraints sollten dynamisch erzeugt werden

Erweiterung I: Gerichtete Konnektionen

- **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 $\sigma(L)\Rightarrow_{\mathcal{T}}\mathsf{False}$ bzw. $\mathsf{True}\Rightarrow_{\mathcal{T}}\sigma(L')$
  - Gültigkeit folgt alleine aus der Theorie, ohne Gegenliteral

$$\Downarrow$$

Eine Formel F ist gültig in einer 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

Erweiterung II: Orthogonale Konnektionen

$$\begin{bmatrix} \begin{bmatrix} x<y_h^{2F} & x<(y_h+1)^{2T} \end{bmatrix} & \begin{bmatrix} x+1<Y_c^{2T} \\ \\ x+1<(Y_c+1)^{2F} \end{bmatrix} \end{bmatrix}$$

- **(Bezüglich x) Orthogonale Formel $F \equiv H \Rightarrow C$**
  – Formel für die entweder $C = H[\rho(x)/x]$ oder $H = C[\rho(x)/x]$ gilt für eine Substitution ρ
  – H und C haben dieselbe Struktur
- **Orthogonale Konnektion (L^T, L'^F) in $F \equiv H \Rightarrow C$**
  – (L^T, L'^F) ist eine gerichtete Konnektion
  – L hat in H dieselbe relative Position wie L' in C

$$\Downarrow$$

Eine orthogonale Formel F ist gültig (in $\mathcal{T}$), wenn es eine zulässige Substitution σ gibt, so daß alle orthogonalen Konnektionen in F σ-komplementär sind

Erweiterung III: Constraints (1)

$$\left[\begin{array}{ccc} \left[\; x<y_h^{2F} \qquad x<(y_h+1)^{2T} \;\right] & \left[\begin{array}{c} x+1<Y_c^{2T} \\ \\ x+1<(Y_c+1)^{2F} \end{array}\right] & \boxed{x+1<(U+1)^{2F}} \end{array}\right]$$

- **Formel F ist σ-komplementär unter Constraint c**
  - Jeder Pfad durch F und c ist σ-komplementär
  - Der Constraint $x+1<(U+1)^{2F}$ macht den Induktionsschritt gültig
- **$\{c_1, .., c_n\}$ vollständige Menge von Constraints**
  - $\forall x_1..x_k\; c_1 \vee .. \vee c_n$ gültig, wobei $x_1..x_k$ alle freien Variablen der c_i
  - $\{\, x+1<(U+1)^{2F},\; x+1<(U+1)^{2T}\}$ wäre vollständig

$$\Downarrow$$

Eine Formel F ist gültig, wenn es eine vollständige Menge von Constraints $\{c_1,..,c_n\}$ und eine zulässige Substitution σ gibt, so daß F unter jedem Constraint c_i σ-komplementär ist

Erweiterung III: Constraints (2)

Wenn alle orthogonalen Konnektionen in einer orthogonalen Formel F unter einem atomaren Constraint c^j komplementär sind, dann ist F komplementär unter dem Constraint $(c^1 \wedge .. \wedge c^k)$

- **Konnektion (L^T, L'^F) komplementär unter c^j**
  - (L^T, L'^F) oder (c^j, L'^F) oder (L^T, c^j) ist komplementär
  - Jede Konnektion kann auf diese Art komplementär gemacht werden
- **Constraints liefern iterative Beweismethode**
  - Überprüfe orthogonale Konnektionen
  - Extrahiere atomaren Constraint c^j aus nichtkomplementärer Konnektion
  - F wird komplementär unter $c = (c^1 \wedge .. \wedge c^k)$
  - Prüfe Komplementärität von F unter $\neg c$

Induktionsbeweis für das Integerquadratwurzelproblem

$$\left[\begin{array}{ccc} \begin{bmatrix} x<y_h^{2F} & x<(y_h+1)^{2T} \end{bmatrix} & \begin{bmatrix} x+1<Y_c^{2T} \\ \\ x+1<(Y_c+1)^{2F} \end{bmatrix} & \begin{array}{c} x+1<(U+1)^{2F} \\ \\ x+1<(U+1)^{2T} \end{array} \end{array}\right]$$

- **Erster Teilbeweis mit orthogonalen Konnektionen**
  - $\boxed{\text{1a}}$ Theorieunifikation mit Rewriting liefert $\sigma_1=[y_h+1/Y_c]$
  - $\boxed{\text{1b}}$ Zweite Konnektion nicht komplementär $\rightsquigarrow$ Constraint $x+1<(y_h+1)^{2F}$
- **Zweiter Teilbeweis unter Constraint $x+1<(y_h+1)^{2T}$**
  - $\boxed{\text{2b}}$ Konnektion mit Constraint ergibt $\sigma_2=[y_h/Y_c]$ durch Unifikation
  - $\boxed{\text{2a}}$ Instantiierte zweite Konnektion ist komplementär in der Arithmetik
- **Beweis beschreibt implizit einen Algorithmus**

```
sqrt x = falls x=0 dann 0
         sonst setze y = sqrt(x-1)
               falls x<(y+1)^2 dann y sonst y+1
```

Entscheidungsprozeduren für die Arithmetik

- **Notwendig für praktische Beweisführung**
  - Arithmetisches Schließen taucht fast überall auf
  - Arithmetische Aussagen tauchen in vielen Erscheinungsformen auf

$$\texttt{x+1<y} \;\wedge\; \texttt{0<t} \;\Rightarrow\; \texttt{(x+1)*t < y*t}$$

entspricht $\texttt{x<y} \;\wedge\; \texttt{0<t} \;\Rightarrow\; \texttt{x*t < y*t}$

und $\texttt{x<y} \;\wedge\; \texttt{0}\leq\texttt{t} \;\Rightarrow\; \texttt{x*(t+1) < y*(t+1)}$

und $\texttt{x+1}\leq\texttt{y} \;\wedge\; \texttt{0<t} \;\Rightarrow\; \texttt{x*t < y*t}$

  - Formale Beweise simpler arithmetischer Aussagen sind nicht leicht

    “Wenn drei ganze Zahlen sich jeweils um maximal 1 unterscheiden, dann sind zwei von ihnen gleich”

- **Formale Arithmetik ist unentscheidbar**
  - Theorie ist gleichmächtig mit Theorie der berechenbaren Funktionen
  - Allgemeine Arithmetik ist nicht einmal vollständig axiomatisierbar

  Entscheidungsprozeduren sind nur für eingeschränkte Arithmetik möglich

Arith: Induktionsfreie Arithmetik

Entscheide arithmetische Probleme der Theorie $\mathcal{A}$

- **Syntax: elementar-arithmetische Formeln**
  - Terme aufgebaut aus ganzzahligen Konstanten, Variablen und +, -, *
    Andersartige Terme werden als Konstanten betrachtet
  - Atomare Formeln: $t_1\,\rho\,t_2$, wobei t_i Terme, $\rho \in \{<, \leq, >, \geq, =, \neq\}$
  - Formeln aufgebaut aus atomaren Formen mit $\neg$, $\wedge$, $\vee$ und $\Rightarrow$
  - Variablen sind implizit all-quantifiziert
- **Semantik charakterisiert durch Axiome**
  1. Gleichheitsaxiome mit eingeschränkter Substitutivität
  2. Axiome der Konstantenarithmetik
  3. Ringaxiome der ganzen Zahlen
  4. Axiome der diskreten linearen Ordnung
  5. Definitionsaxiome für Ordnungsrelationen und Ungleichheiten
  6. Monotonieaxiome
- **$\mathcal{A}$ ist als entscheidbar bekannt**
  - Mathematischer Beweis liefert ein ineffizientes Entscheidungsverfahren

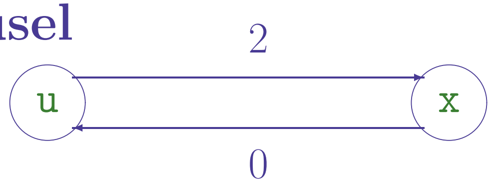
Arith: Arbeitsweise

Ausgangsformel: $A_1 \wedge \ldots \wedge A_n \;\Rightarrow\; C_1 \vee \ldots \vee C_m$ (A_i, C_j atomar)

1. **Normalisiere Formel für Widerspruchsbeweis**
   – Ziel ist Widerlegung von $A_1, \ldots, A_n, \neg C_1, \ldots, \neg C_m$
2. **Entferne Literale ohne atomare arithmetische Formeln**
   – Ersetze Teilterme, die nicht die Syntax von $\mathcal{A}$ erfüllen, durch Variablen
3. **Transformiere Ungleichungen $x{\neq}y$ in $x{\geq}y{+}1 \vee y{\geq}x{+}1$**
   – Erzeuge DNF und betrachte alle Klauseln separat
4. **Transformiere Terme in monadische lineare Polynome ($c{+}u_i$)**
   – Transformiere zunächst alle Komparanden in Standardpolynome
   – Ersetze nicht-konstante Anteile der Polynome durch neue Variablen
5. **Konvertiere Literale in Ungleichungen der Gestalt $u_i{\geq}c{+}u_j$**
   (u_i ist eine Variable oder die Zahl 0
6. **Erzeuge den Ordnungsgraphen der Klausel**
   – Ein Knoten für jede Variablen oder Konstante;
   – Eine Kante $u_i \xrightarrow{c} u_j$ repräsentiert $u_i{\geq}c{+}u_j$
7. **Teste Existenz positiver Zyklen im Graph** (Standardalgorithmus)
   – Positive Zyklen entsprechen einer widersprüchlichen Klausel

`Arith` Arbeitsweise: Beispiel 1

Beweise $\texttt{x+1<y}^2 \;\Rightarrow\; \texttt{x<y}^2$

1. **Erzeuge Formel für Widerspruchsbeweis:** $\texttt{x+1<y}^2\texttt{, }\neg\texttt{(x<y}^2\texttt{)}$
   Nach Auflösung der Negation $\texttt{x+1<y}^2\texttt{, x}\geq\texttt{y}^2$
2. **Entferne Literale ohne atomare arithmetische Formeln** $\surd$
3. **Transformiere Ungleichungen $x{\neq}y$ in $x{\geq}y{+}1 \vee y{\geq}x{+}1$** $\surd$
4. **Transformiere Terme in monadische lineare Polynome**
   $\texttt{x+1<u, x}\geq\texttt{u}$
5. **Konvertiere in Ungleichungen der Gestalt $u_i{\geq}c{+}u_j$**
   $\texttt{u}\geq\texttt{2+x, x}\geq\texttt{0+u}$
6. **Erzeuge den Ordnungsgraphen der Klausel**
   

7. **Standardalgorithmus findet positiven Zyklus im Graphen**
   Ausgangsformel war gültig

Arith Arbeitsweise: Beispiel 2

Beweise $\texttt{z-1<(x+y)}^2 \wedge \texttt{(x+y)}^2\texttt{<z+1} \Rightarrow \texttt{z=(x+y)}^2$

1. Erzeuge Beweisklausel: $\texttt{z-1<(x+y)}^2$, $\texttt{(x+y)}^2\texttt{<z+1}$, $\texttt{z}\neq\texttt{(x+y)}^2$

2. Entferne Literale ohne atomare arithmetische Formeln $\checkmark$

3. Transformiere Ungleichungen $x\neq y$ in $x\geq y+1 \vee y\geq x+1$

1. $\texttt{z-1<(x+y)}^2$, $\texttt{(x+y)}^2\texttt{<z+1}$, $\texttt{z<(x+y)}^2$
2. $\texttt{z-1<(x+y)}^2$, $\texttt{(x+y)}^2\texttt{<z+1}$, $\texttt{z>(x+y)}^2$

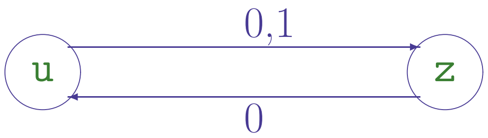
4. Transformiere Terme in monadische lineare Polynome

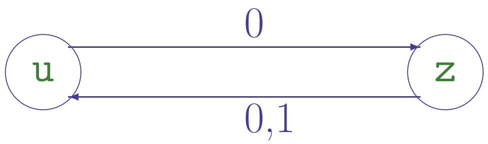
1. `z-1<u, u<z+1, z<u`
2. `z-1<u, u<z+1, z>u`

5. Konvertiere in Ungleichungen der Gestalt $u_i\geq c+u_j$

1. $\texttt{u}\geq\texttt{0+z}$, $\texttt{z}\geq\texttt{0+u}$, $\texttt{u}\geq\texttt{1+z}$
2. $\texttt{u}\geq\texttt{0+z}$, $\texttt{z}\geq\texttt{0+u}$, $\texttt{z}\geq\texttt{1+u}$

6. Erzeuge die Ordnungsgraphen der Klauseln





7. Standardalgorithmus findet je einen positiven Zyklus

Ausgangsformel war gültig

Andere Arten der Induktionsbehandlung

- **Verwendung von wohlfundierter Induktion**
  - $P[0/x] \wedge \forall y[N(y) \Rightarrow (P[y/x] \Rightarrow P[y'/x])] \quad \Rightarrow \forall x(N(x) \Rightarrow P)$
  - Standardinduktion führt zu einfach strukturierter Beweisführung
  - $\forall x(N(x) \;\Rightarrow\; \forall y[N(y) \Rightarrow (x \succ y \Rightarrow P[y/x])] \Rightarrow F) \;\Rightarrow\; F$
  - Vollständige Induktion liefert elegantere Beweise, gleiche Beweisstärke
  - Ordnung $\succ$ muß wohlfundiert sein
- **Konnektionsschemata für Induktion**
  - Das Extensionsverfahren mit Axiomen ist nicht vollständig
    Für Induktionsbeweise gilt kein Schnitteliminationssatz)
  - Gegenstück zur Induktionsregel des Sequenzenkalküls erforderlich
  - Unterstützung durch arithmetische Theoriekonnektionen
- **Definition von Zahlen in Logik zweiter Stufe**
  - Kein Induktionsschema erforderlich
  - Eleganter und vollständig, aber schwerer zu automatisieren
    Schnittelimination gilt für definierte Konzepte

Induktionslose Induktion mit Rewriting

- **Bedeutung von $\forall x F$ beschränkt auf Zahlen**
  - Nur Grundterme, die Zahlen darstellen (0, 0', 0'',...), einzusetzen
  - Logischer Allquantor gilt uneingeschränkt für alle Terme
  - "Beweise" Aussagen durch Termersetzung mit vollständigem Regelsystem
- **Superpositionsbeweise für $s_1{=}t_1, ..s_n{=}t_n \;\Rightarrow\; s{=}t$**
  - Erzeuge vollständiges Regelsystem $\mathcal{R}$ für $s_1{=}t_1, \ldots, s_n{=}t_n$
  - Zeige, daß Vervollständigung mit $s = t$ das System $\mathcal{R}$ nicht erweitert
  - $s{=}t$ muß bereits ableitbar gewesen sein
  - Nur Terme aus zur Verfügung stehenden Symbolen werden betrachtet

$$\vdash\; 0{+}x{=}x \;\wedge\; y'{+}z{=}(y{+}z)' \;\Rightarrow\; \forall uvw (u{+}v){+}w = u{+}(v{+}w)$$

  - Regeln: $0{+}x \longrightarrow x$, $y'{+}z \longrightarrow (y{+}z)'$ liefern $(u{+}v){+}w \longrightarrow u{+}(v{+}w)$
  - Erweitertes Regelsystem ist 'Quasi-reduzierbar'
- **Quasi-Reduzierbarkeit (aufwendig) entscheidbar**
  - Es gibt einfache hinreichende syntaktische Bedingungen an Form der Regeln