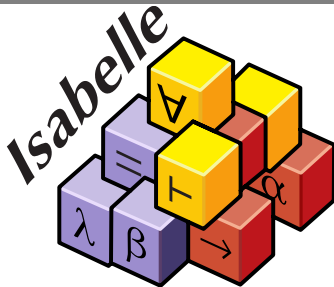


Theorembeweiserpraktikum

Anwendungen in der Sprachtechnologie

LEHRSTUHL PROGRAMMIERPARADIGMEN



Teil XXXI

Koinduktion

Was ist Koinduktion?

Duales Prinzip zu Induktion

Induktive Definition:

kleinster Fixpunkt, der die definierende Gleichung erfüllt.

Koinduktive Definition:

größter Fixpunkt, der die definierende Gleichung erfüllt.

Duales Prinzip zu Induktion

Induktive Definition:

kleinster Fixpunkt, der die definierende Gleichung erfüllt.

Induktionsprinzip: Um eine Eigenschaft für alle Elemente zu zeigen, genügt es sie für eine beliebige Menge zu zeigen, die die definierende Gleichung erfüllt. (Der kleinste Fixpunkt muss darin enthalten sein)

Koinduktive Definition:

größter Fixpunkt, der die definierende Gleichung erfüllt.

Koinduktionsprinzip: Jede Menge, die die definierende Gleichung erfüllt, ist in der koinduktiven Definition enthalten.

Beispiel: Reflexiv transitive Hülle

```
inductive rtc :: "('a ⇒ 'a ⇒ bool) ⇒ 'a ⇒ 'a ⇒ bool"  
for r :: "('a ⇒ 'a ⇒ bool)"  
where refl: "rtc r x x"  
  | trans: "r x y ⇒ rtc r y z ⇒ rtc r x z"
```

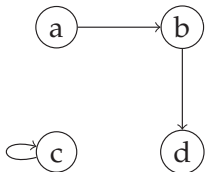
Beispiel: Reflexiv transitive Hülle

```
inductive rtc :: "('a  $\Rightarrow$  'a  $\Rightarrow$  bool)  $\Rightarrow$  'a  $\Rightarrow$  'a  $\Rightarrow$  bool"
```

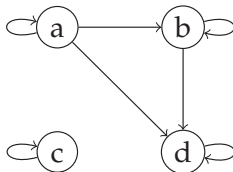
```
for r :: "('a  $\Rightarrow$  'a  $\Rightarrow$  bool)"
```

```
where refl: "rtc r x x"
```

```
| trans: "r x y  $\Longrightarrow$  rtc r y z  $\Longrightarrow$  rtc r x z"
```



(a) Graph r



(b) Inductive RTC r

aus © Andreas Lochbihler, DOI 10.5445/KSP/1000028867, KIT Scientific Publishing, Karlsruhe, 2012

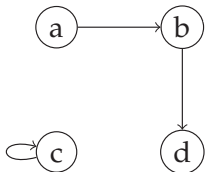
Beispiel: Reflexiv transitive Hülle

```
coinductive rtc :: "('a ⇒ 'a ⇒ bool) ⇒ 'a ⇒ 'a ⇒ bool"
```

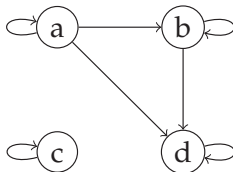
```
for r :: "('a ⇒ 'a ⇒ bool)"
```

```
where refl: "rtc r x x"
```

```
| trans: "r x y ⇒ rtc r y z ⇒ rtc r x z"
```



(a) Graph r



(b) Inductive RTC r

aus © Andreas Lochbihler, DOI 10.5445/KSP/1000028867, KIT Scientific Publishing, Karlsruhe, 2012

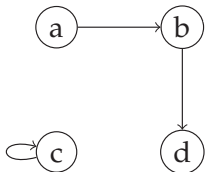
Beispiel: Reflexiv transitive Hülle

coinductive $rtc :: "('a \Rightarrow 'a \Rightarrow bool) \Rightarrow 'a \Rightarrow 'a \Rightarrow bool "$

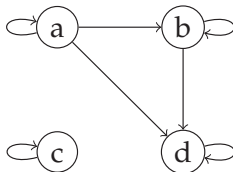
for $r :: "('a \Rightarrow 'a \Rightarrow bool) "$

where $refl: "rtc\ r\ x\ x"$

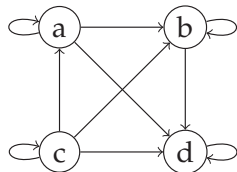
| $trans: "r\ x\ y \Longrightarrow rtc\ r\ y\ z \Longrightarrow rtc\ r\ x\ z"$



(a) Graph r



(b) Inductive RTC r



(c) Coinductive RTC r

aus © Andreas Lochbihler, DOI 10.5445/KSP/1000028867, KIT Scientific Publishing, Karlsruhe, 2012

Induktionsschema für rtc : $rtc.induct$

$$\begin{aligned}rtc\ r\ a\ b &\implies (\bigwedge x. P\ x\ x) \\ &\implies (\bigwedge x\ y\ z. r\ x\ y \implies rtc\ r\ y\ z \implies P\ y\ z \implies P\ x\ z) \\ &\implies P\ a\ b\end{aligned}$$

Induktionsschema für rtc : $rtc.induct$

$$\begin{aligned}rtc\ r\ a\ b &\implies (\bigwedge x. P\ x\ x) \\ &\implies (\bigwedge x\ y\ z. r\ x\ y \implies rtc\ r\ y\ z \implies P\ y\ z \implies P\ x\ z) \\ &\implies P\ a\ b\end{aligned}$$

Koinduktionsschema für rtc : $rtc.coinduct$

$$\begin{aligned}X\ a\ b &\implies \\ &(\bigwedge a\ b. X\ a\ b \implies \\ &\quad (\exists x. a = x \wedge b = x) \vee \\ &\quad (\exists x\ y\ z. a = x \wedge b = z \wedge r\ x\ y \wedge (X\ y\ z \vee rtc\ r\ y\ z))) \implies \\ &rtc\ r\ a\ b\end{aligned}$$

Beweismethoden: *coinduct/coinduction* – analog zu *induct/induction*

Beispiel: Lazy Listen

```
codatatype 'a llist = lnull: LNil | LCons (lhd: 'a) (ltl: "'a llist")
```

Auch hier *duale Sichtweise*: Elemente werden erzeugt, statt abgebaut.

Beispiel: Iterate

```
primcorec literate :: "('a ⇒ 'a) ⇒ 'a ⇒ 'a llist"  
where "literate f s = LCons s (literate f (f s))"
```

Mehr: siehe Sitzung.

Teil XXXII

Dokumentenerzeugung

Isabelle kann Theorien mit \LaTeX schön setzen.

Dazu muss man eine *Sitzung* definieren. Am einfachsten geht das mit
`isabelle mkroot -d name`.

Die Datei `ROOT` führt alle verwendeten Theorien auf.
Die Datei `document/root.tex` enthält den \LaTeX -Rahmen.

Man lässt Isabelle mit
`isabelle build -D .`
die Theorien verarbeiten und die PDF-Dateien erzeugen.

Normaler Text (einschließlich \LaTeX -Makros) kann mittels **text** `\text{bla bla}` eingefügt werden.

Kommentare (`\comment{bla bla}`) erscheinen *nicht* im Dokument!

Statt \LaTeX -Befehle wie `\section`, `\subsection` etc. in **text**-Blöcke einzubauen kann man die entsprechenden Isabelle-Befehle

- **chapter** (bei geeigneter `\documentclass`)
- **section**
- **subsection**
- **subsubsection**

verwenden.

Ähnlich wie in Markdown können Listen auch ohne \LaTeX -Befehl angegeben werden. Dazu werden spezielle Bullet-Points verwendet. Diese kann man in jEdit mittels `\item` für ungeordnete und `\enum` für geordnete Listen eingeben.

Man kann Ausdrücke verschiedener Art von Isabelle in das Dokument einfügen lassen:

Nach

definition $N :: nat$ **where** $"N = 0"$

theorem $great_result: "N = N * P"$ **unfolding** N_def **by** $simp$
wird aus

text \langle

*After defining $\@{\thm N_def}$ we were finally able
to prove $\@{\thm great_result}$.*

\rangle

in der Dokumentausgabe

After defining $N = 0$ we were finally able to prove $N = N * ?P$.

Neben `@{thm ...}` sind noch nützlich:

- `@{theory ...}` verweist auf einen (importierten) Theorie-Namen,
- `@{term ...}` setzt einen Term,
- `@{term_type ...}` ebenso, aber mit Typ,
- `@{typ ...}` setzt einen Typ,
- `@{value ...}` evaluiert einen Term und zeigt das Ergebnis,
- `@{text ...}` setzt beliebigen Text im Isabelle-Stil.

Während `@{thm ...}` garantiert, dass nur bewiesenes gedruckt wird, überprüfen die anderen nur die Typisierung, und mit `@{text ...}` lässt sich alles ausgeben.

Beim Ausgeben von Lemmas ist oft `@{thm great_result[no_vars]}` schöner als `@{thm great_result}`.

Standardmäßig enthält `document/root.tex` den Befehl `\input{session}` und `session.tex` (von Isabelle erstellt) enthält für jede Theorie `foo` eine Zeile `\input{Example.tex}`.

Man kann natürlich auch die Theorie-Dateien direkt in `document/root.tex` einbinden, etwa um dazwischen noch Text wie Kapitelüberschriften oder Einleitungen zu setzen.

Auch will man vielleicht in der Einleitung schon auf alle Definitionen und Ergebnisse vorgreifen. Dazu erstellt man z.B. eine Theorie `Introduction` und bindet diese in `document/root.tex` am Anfang ein.

Für Theorien, die in `ROOT` mit der Option `document = false` versehen sind, werden nicht in das Dokument aufgenommen (die trotzdem erzeugte `.tex`-Datei ist leer).

Mehr Informationen

zu mehr Anti-Quotations siehe das Isabelle Referenz-Handbuch
(`isabelle doc isar-ref`).

Für mehr \LaTeX -Spielereien wie z.B. die Ausgabe

$$\frac{P \ 0 \quad \bigwedge_{\text{nat.}} \frac{P \ \text{nat}}{P \ (\text{Suc} \ \text{nat})}}{P \ \text{nat}}$$

für

```
text  $\langle \backslash \text{begin}\{center\}$   
   $\@{\thm}\{mode=Rule\} \ \text{nat.}\ \text{induct}\ [no\_vars]\}$   
 $\backslash \text{end}\{center\} \rangle$ 
```

siehe `isabelle doc sugar`.