Die Inferenzregeln von QUODLIBET

Bemerkung

Die hier angegebenen Inferenzregeln dienten der Motivation und dem Verständnis, welche Arten von Inferenzregeln benötigt werden. Sie müssen für den praktischen Gebrauch noch verfeinert werden.

Ziel

Angabe aller Inferenzregeln von QUODLIBET.

Dazu

Wir benötigen einige Bezeichnungen.

Bezeichnungen

Minimale Unterscheidungs-Positionen von t_1 und t_2

$\mathsf{MinDifPos}(t_1, t_2)$ besteht aus den minimalen Positionen aus $\{\pi \in \mathsf{Pos}(t_1) \cap \mathsf{Pos}(t_2) \mid \mathsf{top}(t_1/\pi) \neq \mathsf{top}(t_2/\pi)\}$.

Minimale Nicht-Konstruktor-Positionen in t

$\mathsf{MinNonCPos}(t)$ besteht aus den minimalen Positionen aus $\{\pi \in \mathsf{Pos}(t) \mid \mathsf{top}(t/\pi) \notin C \cup V^C\}$.

Die **C-Front** von t

Sei $t \in \mathsf{Term}(F, V)$. Seien $t_1, \ldots, t_m$ die verschiedenen maximalen Teilterme von t mit $\mathsf{top}(t_i) \notin C \cup V^C$. Seien $x_1, \ldots, x_m \in V^C$ neue Variable. Die C-Front $\hat{t}$ von t entsteht durch Ersetzen der t_i durch x_i.

Termmengen

$\mathcal{T}(\mathsf{sig}^C, V^C) = \mathsf{Term}(C, V^C)$.

Literale

$(\lambda_1, \ldots, \lambda_n)[m] = \lambda_m$ für $1 \leq m \leq n$.

Non-Applicative Inference Rules

Establishing Simple Tautologies

Complementary Literals:
`compl-lit` *m n*

$$\frac{\langle \Gamma, \lambda, \Delta, \lambda', \Pi \,;\, w\rangle}{}$$

if
- $(\Gamma, \lambda, \Delta, \lambda', \Pi)[m] = \lambda$
- $(\Gamma, \lambda, \Delta, \lambda', \Pi)[n] = \lambda'$
- $\lambda' =_{\mathrm{lit}} \overline{\lambda}$.

$\neq$-Tautology:
`=/=-taut` *m*

$$\frac{\langle \Gamma,\ t_1 \neq t_2 \,,\, \Delta \,;\, w\rangle}{}$$

if
- $(\Gamma,\ t_1 \neq t_2 \,,\, \Delta)[m] =\ t_1 \neq t_2$
- $t_1, t_2 \in \mathcal{T}(\mathsf{sig}^C, V^{\mathrm{C}})$
- t_1 and t_2 are not unifiable.

$<$-Tautology:
`<-taut` *m*

$$\frac{\langle \Gamma,\ () < (u_1, \ldots, u_k) \,,\, \Delta \,;\, w\rangle}{}$$

if
- $(\Gamma,\ () < (u_1, \ldots, u_k) \,,\, \Delta)[m] = () < (u_1, \ldots, u_k)$
- $k > 0$.

Decomposing Atoms

$=$-Decomposition:
`=-decomp` *m*

$$\frac{\langle \Gamma,\ t_1 = t_2 \,,\, \Delta \,;\, w\rangle}{\langle\, u_1 = v_1 \,,\, \Gamma,\ t_1 = t_2 \,,\, \Delta \,;\, w\rangle \ \ldots\ \langle\, u_k = v_k \,,\, \Gamma,\ t_1 = t_2 \,,\, \Delta \,;\, w\rangle}$$

if
- $(\Gamma,\ t_1 = t_2 \,,\, \Delta)[m] =\ t_1 = t_2$
- $\mathrm{top}(t_1) = \mathrm{top}(t_2)$
- $u_1 = v_1 \,, \ldots,\ u_k = v_k$ are exactly those equations in

  $\{\, t_1/p \doteq t_2/p \mid p \in \mathrm{MinDifPos}(t_1, t_2) \,\}$

  whose complements do not occur in Γ, Δ.

def-Decomposition:
`def-decomp` *m*

$$\frac{\langle \Gamma,\ \mathrm{def}(t) \,,\, \Delta \,;\, w\rangle}{\langle\, \mathrm{def}(u_1) \,,\, \Gamma,\ \mathrm{def}(t) \,,\, \Delta \,;\, w\rangle \ \ldots\ \langle\, \mathrm{def}(u_k) \,,\, \Gamma,\ \mathrm{def}(t) \,,\, \Delta \,;\, w\rangle}$$

if
- $(\Gamma,\ \mathrm{def}(t) \,,\, \Delta)[m] =\ \mathrm{def}(t)$
- $\mathrm{top}(t) \in (C \cup V^{\mathrm{C}})$
- $\mathrm{def}(u_1) \,, \ldots,\ \mathrm{def}(u_k)$ are exactly those definedness atoms in

  $\{\, \mathrm{def}(t/p) \mid p \in \mathrm{MinNonCPos}(t) \,\}$

  whose complements do not occur in Γ, Δ.

$<$-Decomposition:
`<-decomp` *m*

$$\frac{\langle \Gamma,\ t_1 < t_2 \,,\, \Delta \,;\, w\rangle}{\langle\, \mathrm{def}(u_1) \,,\, \Gamma,\ t_1 < t_2 \,,\, \Delta \,;\, w\rangle \ \ldots\ \langle\, \mathrm{def}(u_k) \,,\, \Gamma,\ t_1 < t_2 \,,\, \Delta \,;\, w\rangle}$$

if
- $(\Gamma,\ t_1 < t_2 \,,\, \Delta)[m] =\ t_1 < t_2$
- $\mathrm{top}(t_2) \in C$
- there are $\hat{t}_1, \hat{t}_2 \in \mathcal{T}(\mathsf{sig}^C, V^{\mathrm{C}})$ such that
  - for $i \in \{1, 2\}$, $\hat{t}_i$ is a C-front for t_i
  - for each $p_1 \in \mathrm{MinNonCPos}(t_1)$ and $p_2 \in \mathrm{MinNonCPos}(t_2)$, $t_1/p_1 = t_2/p_2$ iff $\hat{t}_1/p_1 = \hat{t}_2/p_2$
  - $|\hat{t}_1| < |\hat{t}_2|$ and $|\hat{t}_1|_x \leq |\hat{t}_2|_x$ for every $x \in V^{\mathrm{C}}$
  - $\mathrm{def}(u_1) \,, \ldots,\ \mathrm{def}(u_k)$ are exactly those definedness atoms in

    $\{\, \mathrm{def}(t_i/p) \mid i \in \{1, 2\} \wedge p \in \mathrm{MinNonCPos}(t_i) \,\}$

    that do not occur in $\overline{\Gamma}, \overline{\Delta}$.

Removing Redundant Literals

Multiple Literals:
`mult-lit` *m n*

$$\frac{\langle \Gamma, \lambda, \Delta, \lambda', \Pi \,;\, w\rangle}{\langle \Gamma, \lambda, \Delta, \Pi \,;\, w\rangle}$$

if
- $(\Gamma, \lambda, \Delta, \lambda', \Pi)[m] = \lambda$
- $(\Gamma, \lambda, \Delta, \lambda', \Pi)[n] = \lambda'$
- $\lambda' =_{\mathrm{lit}} \lambda$.

$=$-Removal:
`=-removal` *m*

$$\frac{\langle \Gamma,\ t_1 = t_2\,,\Delta\,;\ w\rangle}{\langle \Gamma, \Delta\,;\ w\rangle}$$

if
- $(\Gamma,\ t_1 = t_2\,, \Delta)[m] =\ t_1 = t_2$
- $t_1, t_2 \in \mathcal{T}(\mathsf{sig}^C, V^{\mathrm{C}})$
- t_1 and t_2 are not unifiable.

$<$-Removal:
`<-removal` *m*

$$\frac{\langle \Gamma,\ (u_1,\ldots,u_k) < ()\,,\Delta\,;\ w\rangle}{\langle \Gamma, \Delta\,;\ w\rangle}$$

if $(\Gamma,\ (u_1,\ldots,u_k) < ()\,, \Delta)[m] =$
$(u_1,\ldots,u_k) < ()\,.$

$\neq$-Removal:
`=/=-removal` *m*

$$\frac{\langle \Gamma,\ t_1 \neq t_2\,,\Delta\,;\ w\rangle}{\langle \Gamma, \Delta\,;\ w\rangle}$$

if
- $(\Gamma,\ t_1 \neq t_2\,, \Delta)[m] =\ t_1 \neq t_2$
- $\mathrm{top}(t_1) = \mathrm{top}(t_2)$
- each atom in $\{\, t_1/p \neq t_2/p \mid p \in \mathrm{MinDifPos}(t_1, t_2)\,\}$ occurs in Γ, Δ.

$\neg$def-Removal:
`~def-removal` *m*

$$\frac{\langle \Gamma,\ \neg\mathrm{def}(t)\,,\Delta\,;\ w\rangle}{\langle \Gamma, \Delta\,;\ w\rangle}$$

if
- $(\Gamma,\ \neg\mathrm{def}(t)\,, \Delta)[m] =\ \neg\mathrm{def}(t)$
- $\mathrm{top}(t) \in (C \cup V^{\mathrm{C}})$
- each atom in $\{\, \neg\mathrm{def}(t/p) \mid p \in \mathrm{MinNonCPos}(t)\,\}$ occurs in Γ, Δ.

Making Use of Negative Literals

Constant Rewriting:
`const-rewrite` *m n p*

$$\frac{\langle \Gamma, \lambda[t_1]_p, \Delta\,;\ w\rangle}{\langle \Gamma, \lambda[t_2]_p, \Delta\,;\ w\rangle}$$

if
- $(\Gamma, \lambda, \Delta)[m] = \lambda$
- $p \in \mathrm{Pos}(\lambda)$ and $\lambda/p = t_1$
- there is a literal $t_1 \dot{\neq} t_2$ in Γ, Δ such that $(\Gamma, \lambda, \Delta)[n] =\ t_1 \dot{\neq} t_2\,.$

$\neq$-Unification:
`=/=-unif` *m*

$$\frac{\langle \Gamma,\ t_1 \neq t_2\,,\Delta\,;\ w\rangle}{\langle \Gamma\tau, \Delta\tau\,;\ w\tau\rangle}$$

if
- $(\Gamma,\ t_1 \neq t_2\,, \Delta)[m] =\ t_1 \neq t_2$
- $t_1, t_2 \in \mathcal{T}(\mathsf{sig}^C, V^{\mathrm{C}})$
- $\tau = \mathrm{mgu}(\ t_1 = t_2\,, \mathrm{Var}(\Gamma, t_1, t_2, \Delta, w))$ exists.

Constructor Variable Addition:
`ctr-var-add` *m x*

$$\frac{\langle \Gamma,\ \neg\mathrm{def}(t)\,,\Delta\,;\ w\rangle}{\langle \Gamma,\ x \neq t\,, \Delta\,;\ w\rangle}$$

if
- $(\Gamma,\ \neg\mathrm{def}(t)\,, \Delta)[m] =\ \neg\mathrm{def}(t)$
- $x \in V^{\mathrm{C}} \setminus \mathrm{Var}(\Gamma, \Delta, t, w).$

Further Inference Rules for Order Atoms

Tuple $<$-Reduction:
`tuple <-reduct` *k*

$$\frac{\langle \Gamma,\ (t_1, u_1, \ldots, u_m) < (t_2, v_1, \ldots, v_n)\,, \Delta\,;\ w\rangle}{\langle\, t_1 < t_2\,, \Gamma,\ (t_1, u_1, \ldots, u_m) < (t_2, v_1, \ldots, v_n)\,, \Delta\,;\ w\rangle}$$

if
- $(\Gamma, (t_1, u_1, \ldots, u_m) < (t_2, v_1, \ldots, v_n)\,, \Delta)[k] =$
  $(t_1, u_1, \ldots, u_m) < (t_2, v_1, \ldots, v_n)$
- $m > 0\ \vee\ n > 0.$

Tuple $=$-Reduction:
`tuple =-reduct` *k*

$$\frac{\langle \Gamma,\ (t, u_1, \ldots, u_m) < (t, v_1, \ldots, v_n)\,, \Delta\,;\ w\rangle}{\langle \Gamma,\ (u_1, \ldots, u_m) < (v_1, \ldots, v_n)\,, \Delta\,;\ w\rangle}$$

if $(\Gamma, (t, u_1, \ldots, u_m) < (t, v_1, \ldots, v_n)\,, \Delta)[k] =$
$(t, u_1, \ldots, u_m) < (t, v_1, \ldots, v_n)\,.$

$<$-Monotonicity:
`<-mono` *m* p_1 p_2

$$\frac{\langle \Gamma,\ t_1[u_1]_{p_1} < t_2[u_2]_{p_2}\ ,\Delta\ ;\ w\rangle}{\langle\ u_1 < u_2\ ,\Gamma,\ t_1[u_1]_{p_1} < t_2[u_2]_{p_2}\ ,\Delta\ ;\ w\rangle}$$

if
- $(\Gamma,\ t_1 < t_2\ ,\Delta)[m] =\ t_1 < t_2$
- for $i \in \{1,2\}$, $p_i \in \mathrm{Pos}(t_i)\backslash\{\varepsilon\}$ and $t_i/p_i = u_i$
- there are $\hat{t}_1, \hat{t}_2 \in \mathcal{T}(\mathrm{sig}^C, V^{\mathrm{C}})$ and $x_1, x_2 \in V^{\mathrm{C}}\backslash\mathrm{Var}(t_1,t_2)$ such that
  - for $i \in \{1,2\}$, $\hat{t}_i = t_i[x_i]_{p_i}$
  - $|\hat{t}_1| \le |\hat{t}_2|$
  - for every $x \in V^{\mathrm{C}}\backslash\{x_1,x_2\}$, $|\hat{t}_1|_x \le |\hat{t}_2|_x$.

$<$-Transitivity:
`<-trans` *m* w_2

$$\frac{\langle \Gamma,\ w_1 < w_3\ ,\Delta\ ;\ w\rangle}{\langle\ w_1 < w_2\ ,\Gamma,\ w_1 < w_3\ ,\Delta\ ;\ w\rangle\ \langle\ w_2 < w_3\ ,\Gamma,\ w_1 < w_3\ ,\Delta\ ;\ w\rangle}$$

if $(\Gamma,\ w_1 < w_3\ ,\Delta)[m] =\ w_1 < w_3$.

Non-Applicative Case Analyses

Substitution Addition:
`subst-add` $\sigma_1\ \ldots\ \sigma_n$.

$$\frac{\langle \Gamma\ ;\ w\rangle}{\langle \Gamma\sigma_1\ ;\ w\sigma_1\rangle\ \ldots\ \langle \Gamma\sigma_n\ ;\ w\sigma_n\rangle}$$

if $\{\sigma_1,\ldots,\sigma_n\}$ is a cover set of substitutions for $\langle\Gamma\ ;\ w\rangle$.

Literal Addition:
`lit-add` $\lambda_1\ \ldots\ \lambda_n$.

$$\frac{\langle \Gamma\ ;\ w\rangle}{\langle \Lambda_1,\Gamma\ ;\ w\rangle\ \ldots\ \langle \Lambda_n,\Gamma\ ;\ w\rangle\ \langle \Lambda,\Gamma\ ;\ w\rangle}$$

if $\Lambda_1,\ldots,\Lambda_n,\Lambda$ is the case analysis resulting from literals $\lambda_1,\ldots,\lambda_n$ for $n > 0$.

Applicative Inference Rules

Non-Inductive Applicative Inference Rules

Non-Inductive Subsumption:
`axiom-subs` *DefRule* μ

$$\frac{\langle \Gamma\ ;\ w\rangle}{\langle \Lambda_1,\Gamma\ ;\ w\rangle\ \ldots\ \langle \Lambda_n,\Gamma\ ;\ w\rangle}\quad \text{with}\ \ \langle \Pi\ ;\ ()\rangle^{\mathcal{L}}$$

if there is a substitution μ and a clause Θ such that
- the clause representation of the defining rule *DefRule* is Π
- Γ,Θ contains $\mathrm{DefCond}(\mu,\Pi), \Pi\mu$
- $\Lambda_1,\ldots,\Lambda_n,\Lambda$ is the case analysis resulting from Θ.

`lemma-subs` *PS-Tree* μ

$$\frac{\langle \Gamma\ ;\ w\rangle}{\langle \Lambda_1,\Gamma\ ;\ w\rangle\ \ldots\ \langle \Lambda_n,\Gamma\ ;\ w\rangle}\quad \text{with}\ \ \langle \Pi\ ;\ \hat{w}\rangle^{\mathcal{L}}$$

if there is a substitution μ and a clause Θ such that
- the goal labeling the root of the proof state tree *PS-Tree* is $\langle\Pi\ ;\ \hat{w}\rangle$
- Γ,Θ contains $\mathrm{DefCond}(\mu,\Pi), \Pi\mu$
- $\Lambda_1,\ldots,\Lambda_n,\Lambda$ is the case analysis resulting from Θ.

Non-Inductive Rewriting:
`axiom-rewrite` *m p DefRule n* μ

$$\frac{\langle \Gamma,\lambda,\Delta\ ;\ w\rangle}{\langle \Lambda_1,\Gamma,\lambda,\Delta\ ;\ w\rangle\ \ldots\ \langle \Lambda_n,\Gamma,\lambda,\Delta\ ;\ w\rangle\ \langle \Lambda,\Gamma,\lambda[r\mu]_p,\Delta\ ;\ w\rangle}$$

with $\langle \Pi,\ l \doteq r\ ,\Sigma\ ;\ ()\rangle^{\mathcal{L}}$

if there is a position $p \in \mathrm{Pos}(\lambda)$, a substitution μ and a clause Θ such that
- $(\Gamma,\lambda,\Delta)[m] = \lambda$
- the clause representation of the defining rule *DefRule* is $\Pi,\ l \doteq r\ ,\Sigma$
- $(\Pi,\ l \doteq r\ ,\Sigma)[n] =\ l \doteq r$
- $\lambda/p = l\mu$
- $\Gamma,\Delta,\ l\mu = r\mu\ ,\Theta$ contains $\mathrm{DefCond}(\mu,(\Pi,\ l \doteq r\ ,\Sigma)), \Pi\mu, \Sigma\mu$
- $\Lambda_1,\ldots,\Lambda_n,\Lambda$ is the case analysis resulting from Θ.

`lemma-rewrite` *m p PS-Tree n* μ

$$\frac{\langle \Gamma, \lambda, \Delta \,;\, w\rangle}{\langle \Lambda_1, \Gamma, \lambda, \Delta \,;\, w\rangle \;\ldots\; \langle \Lambda_n, \Gamma, \lambda, \Delta \,;\, w\rangle \; \langle \Lambda, \Gamma, \lambda[r\mu]_p, \Delta \,;\, w\rangle}$$

with $\langle \Pi,\, l \doteq r \,,\, \Sigma \,;\, \hat{w}\rangle^{\mathcal{L}}$

if there is a position $p \in \mathrm{Pos}(\lambda)$, a substitution μ and a clause Θ such that

- $(\Gamma, \lambda, \Delta)[m] = \lambda$
- the goal labeling the root of the proof state tree *PS-Tree* is $\langle \Pi,\, l \doteq r \,,\, \Sigma \,;\, \hat{w}\rangle$
- $(\Pi,\, l \doteq r \,,\, \Sigma)[n] = \; l \doteq r$
- $\lambda/p = l\mu$
- $\Gamma, \Delta,\, l\mu = r\mu \,, \Theta$ contains $\mathrm{DefCond}(\mu, (\Pi,\, l \doteq r \,,\, \Sigma)), \Pi\mu, \Sigma\mu$
- $\Lambda_1, \ldots, \Lambda_n, \Lambda$ is the case analysis resulting from Θ.

Applicative Literal Removal:
`appl-lit-removal` *m* μ *Lma n*

$$\frac{\langle \Gamma, \lambda, \Delta \,;\, w\rangle}{\langle \Gamma, \Delta \,;\, w\rangle} \quad \text{with } \langle \Pi, \lambda', \Sigma \,;\, \hat{w}\rangle^{\mathcal{L}}$$

if there is a substitution μ such that

- $(\Gamma, \lambda, \Delta)[m] = \lambda$
- if Lma is a defining rule then Π, λ', Σ is the clause representation of Lma
- if Lma is a proof state tree then the root of Lma is labeled with $\langle \Pi, \lambda', \Sigma \,;\, \hat{w}\rangle$
- $(\Pi, \lambda', \Sigma)[n] = \lambda'$
- $\lambda =_{\mathrm{lit}} \overline{\lambda'\mu}$
- Γ, Δ contains $\mathrm{DefCond}(\mu, (\Pi, \lambda', \Sigma)), \Pi\mu, \Sigma\mu$.

Inductive Applicative Inference Rules

Inductive Subsumption:
`ind-subs` *PS-Tree* μ

$$\frac{\langle \Gamma \,;\, w\rangle}{\langle \Lambda_1, \Gamma \,;\, w\rangle \;\ldots\; \langle \Lambda_n, \Gamma \,;\, w\rangle \; \langle\, \hat{w}\mu < w \,, \Lambda, \Gamma \,;\, w\rangle} \quad \text{with } \langle \Pi \,;\, \hat{w}\rangle^{\mathcal{I}}$$

if there is a substitution μ and a clause Θ such that

- the goal labeling the root of the proof state tree *PS-Tree* is $\langle \Pi \,;\, \hat{w}\rangle$
- Γ, Θ contains $\mathrm{DefCond}(\mu, \Pi), \Pi\mu$
- $\Lambda_1, \ldots, \Lambda_n, \Lambda$ is the case analysis resulting from Θ.

Inductive Rewriting:
`ind-rewrite` *m p PS-Tree n* μ

$$\frac{\langle \Gamma, \lambda, \Delta \,;\, w\rangle}{\langle \Lambda_1, \Gamma, \lambda, \Delta \,;\, w\rangle \;\ldots\; \langle \Lambda_n, \Gamma, \lambda, \Delta \,;\, w\rangle \; \langle \Lambda, \Gamma, \lambda[r\mu]_p, \Delta \,;\, w\rangle \; \langle\, \hat{w}\mu < w \,, \Lambda', \Gamma, \lambda, \Delta \,;\, w\rangle}$$

with $\langle \Pi,\, l \doteq r \,,\, \Sigma \,;\, \hat{w}\rangle^{\mathcal{I}}$

if there is a position $p \in \mathrm{Pos}(\lambda)$, a substitution μ and a clause Θ such that

- $(\Gamma, \lambda, \Delta)[m] = \lambda$
- the goal labeling the root of the proof state tree *PS-Tree* is $\langle \Pi,\, l \doteq r \,,\, \Sigma \,;\, \hat{w}\rangle$
- $(\Pi,\, l \doteq r \,,\, \Sigma)[n] = \; l \doteq r$
- $\lambda/p = l\mu$
- $\Gamma, \Delta,\, l\mu = r\mu \,, \Theta$ contains $\mathrm{DefCond}(\mu, (\Pi,\, l \doteq r \,,\, \Sigma)), \Pi\mu, \Sigma\mu$
- $\Lambda_1, \ldots, \Lambda_n, \Lambda$ is the case analysis resulting from Θ
- $\Lambda' = \; l\mu = r\mu \,, \Lambda$.

Zur Korrektheit und Sicherheit von Inferenzregeln (1)

Beachte

Alle Inferenzregeln haben die Form

$$\frac{\langle \Gamma; \omega \rangle}{\langle \Gamma_1; \omega_1 \rangle \cdots \langle \Gamma; \omega_n \rangle} \quad (\text{mit } \langle \Pi; \omega' \rangle^{\mathcal{U}})$$

Dabei ist entweder $\omega_i \equiv \omega$ für alle i oder $\omega_i \equiv \sigma_i(\omega)$ für alle i.

Die Regel ist genau dann nicht-applikativ, falls der Teil (mit $\langle \Pi; \omega' \rangle^{\mathcal{U}}$) entfällt.

Beachte

Gilt $\Gamma \subseteq \Gamma_i$ für alle i, so ist die Regel sicher.

Frage

Wie zeigt man, dass die Regel korrekt ist?

Zur Korrektheit und Sicherheit von Inferenzregeln (2)

Beachte

a) Sei die Inferenzregel nicht-applikativ oder Π gültig. Die Inferenzregel ist korrekt, falls für jedes $\mathcal{A} \in$ DMod(spec) gilt

$$\mathcal{A} \models \Gamma \text{ , falls } \mathcal{A} \models \Gamma_i \text{ für alle } i$$

In diesem Fall bleibt die Größe der Gegenbeispiele erhalten.
Die Inferenzregel ist sicher, falls für jedes $\mathcal{A} \in$ DMod(spec) gilt

$$\mathcal{A} \models \Gamma_i \text{ für alle } i \text{ , falls } \mathcal{A} \models \Gamma$$

b) Sei die Inferenzregel applikativ, $\mathcal{U} = \mathcal{I}$ und Π nicht gültig.
Die Inferenzregel ist korrekt, falls für alle $\mathcal{A} \in$ DMod(spec) und jedes $\mathcal{A}$-Gegenbeispiel $(\langle \Gamma; \omega \rangle, \sigma)$ ein $\mathcal{A}$-Gegenbeispiel $(\langle \Pi; \omega' \rangle, \tau)$ gibt mit $(\langle \Pi; \omega' \rangle, \tau) <_{\mathcal{A}} (\langle \Gamma; \omega \rangle, \sigma)$.
In diesem Fall werden Gegenbeispiele durch kleinere ersetzt.

c) Sei Var$(\Gamma, \Gamma_1, \ldots, \Gamma_n) \subseteq V^C$. Es gilt
$\mathcal{A} \models \Gamma \text{ gdw } \mathcal{A} \models \tau(\Gamma)$ für jede induktive Substitution τ.
$\mathcal{A} \models \tau(\Gamma) \text{ gdw } \mathcal{A} \models \tau(\lambda)$ für ein $\lambda \in \Gamma$.