Inferenzmethoden

Einheit 14

Logiken höherer Stufe

1. Lineare Logik
2. Matrixmethoden für Fragmente
3. Higher-order Logik

Lineare Logik

Ressourcen-orientierte Logik

- **Schließen über Konsequenzen von Handlungen**
  - Formeln sind Ressourcen, die aufgebraucht werden
  - Manche Ressourcen werden als wiederverwendbar gekennzeichnet
  - Adäquater für Planung und Modellierung von Aktionen
  - Keine Frame-Axiome erforderlich
- **Viele andersartige Grundoperatoren erforderlich**
  - Mehrerere Varianten von Konjunktion, Disjunktion, Implikation
    Kommutative und nichtkommutative Versionen
    Idempotente und nicht-idempotente Versionen (z.B. $A \not\models A \wedge A$)
  - Additive, multiplikative und exponentielle Operatoren
- **Logik gilt nach 30 Jahren immer noch als kompliziert**
  - Semantik von der Fachwelt nur wenig verstanden
  - Bereits aussagenlogisch unentscheidbar
  - Nur in Fragmenten automatisierbar (↦ Proof Nets / Lineare Konnektionsmethode)

Syntax und Semantik

- **Syntax der (aussagenlogischen) linearen Logik**
  - $\mathbf{1}$, $\perp$, $\top$, $\mathbf{0}$ und elementare Propositionen sind (atomare) Formeln
  - Sind A und B Formeln, dann auch

    $A^{\perp},\ A{\otimes}B,\ A{\parr}B,\ A{\multimap}B,\ A\&B,\ A{\oplus}B,\ !A,\ ?A$
  - Prädikatenlogische Syntax ist analog zur Standardlogik
- **Keine leicht zu definierende Semantik**
  - Dialogische Semantik (Game Semantics) erklärt Operatoren
  - Einfachere Semantik basiert auf Beweisbarkeit im Sequenzenkalkül
- **Lineare Logik ist als substrukturelle Logik beschreibbar**
  - Sequenzenkalkül ohne allgemeine Kontraktion und Ausdünnung

    Nur wiederverwendbare Ressourcen können verdoppelt werden

    Alle anderen Hypothesen werden “verbraucht” (Linearität)
  - Regeln definieren Bedeutung der einzelnen Operatoren

Bedeutung der Konnektive

- $A^\perp$: Lineare Negation (Wechsel zwischen Verwenden und Erzeugen)
- $A \otimes B$: Multiplikative Konjunktion, *ein* A und *ein* B
- $A \& B$: Additive Konjunktion, freie Wahl zwischen A und B (nicht beide)
- $A ⅋ B \equiv (A^\perp \otimes B^\perp)^\perp$: Multiplikative Disjunktion
- $A \oplus B \equiv (A^\perp \& B^\perp)^\perp$: Additive Disjunktion, *genau ein* A oder *ein* B
- $A \multimap B \equiv (A \otimes B^\perp)^\perp$: Multiplikative Implikation
  - *genau ein* A wird verbraucht um *ein* B zu erzeugen
- $\mathbf{1}$: Multiplikative Einheit
  - Kann verbraucht werden, ohne daß etwas produziert werden muß
  - Kann ohne Ressourcen erzeugt werden
- $\top$: Additive Einheit
  - Kann nicht verbraucht werden
  - Kann erzeugt werden und dabei beliebige Ressourcen verbrauchen/erzeugen
- $\perp \equiv \mathbf{1}^\perp$: Multiplikativ leere Ressource
- $\mathbf{0} \equiv \top^\perp$: Additiv leere Ressource
- $!A$: A kann beliebig oft verwendet werden (Exponential)
- $?A \equiv (!(A^\perp))^\perp$: A kann beliebig oft erzeugt werden (Exponential)

BEISPIELFORMALISIERUNGEN

- **Chemische Reaktion:** 2xWasserstoff+Sauerstoff=2xWasser
  - Klassische Formel $H_2 \wedge H_2 \wedge O_2 \Rightarrow H_2O \wedge H_2O$ drückt dies nicht aus, denn dies wäre gleichwertig zu $H_2 \wedge O_2 \Rightarrow H_2O$
  - Lineare Logik beschreibt Ressourcen $H_2 \otimes H_2 \otimes O_2 \multimap H_2O \otimes H_2O$
- **Planungsprobleme mit Auswahl**
  - Für 6€ bekomme ich Zigaretten
Für 6€ bekomme ich ein Tagesticket Berlin ABC
  - Klassische Logik: $6€ \Rightarrow \text{Cig},\ 6€ \Rightarrow \text{ABC} \vdash 6€ \Rightarrow \text{Cig} \wedge \text{ABC}$
  - Lineare Logik: $6€ \multimap \text{Cig},\ 6€ \multimap \text{ABC} \vdash 6€ \multimap \text{Cig} \,\&\, \text{ABC}$
$6€ \multimap \text{Cig},\ 6€ \multimap \text{ABC} \vdash 12€ \multimap \text{Cig} \otimes \text{ABC}$

    oder sogar $!(6€ \multimap \text{Cig}),\ !(6€ \multimap \text{ABC})$
$\vdash 12€ \multimap (\text{Cig} \otimes \text{ABC}) \,\&\, (\text{Cig} \otimes \text{Cig}) \,\&\, (\text{ABC} \otimes \text{ABC})$
- **Tagesmenü eines Restaurants**
  - $25€ \multimap ((\text{Tomatensuppe} \oplus \text{Hühnersuppe}) \,\&\, \text{Salat}) \otimes (\text{Fisch} \,\&\, \text{Rind})$
$\otimes (5€ \multimap ((\text{Eiskreme} \,\&\, \text{Kuchen}) \otimes !\text{Kaffee}))$
  - In klassischer Logik nicht beschreibbar

Refinement Kalkül für Lineare Logik

Multiplikatives Fragment	
$H, H', A \multimap B \vdash G, G'$ ⊸L $H \vdash G, A$ $H', B \vdash G'$	$H \vdash G, A \multimap B$ ⊸R $H, A \vdash G, B$
$H, A \otimes B \vdash G$ ⊗L $H, A, B \vdash G$	$H, H' \vdash G, G', A \otimes B$ ⊗R $H \vdash G, A$ $H' \vdash G', B$
$H, H', A \parr B \vdash G, G'$ ⅋L $H, A \vdash G$ $H', B \vdash G'$	$H \vdash G, A \parr B$ ⅋R $H \vdash G, A, B$
$H, \mathbf{1} \vdash G$ **1**L $H \vdash G$	$\vdash \mathbf{1}$ **1**R
$\bot \vdash$ ⊥L	$H \vdash G, \bot$ ⊥R $H \vdash G$
Negation	
$H, A^{\bot} \vdash G$ $^{\bot}$L $H \vdash G, A$	$H \vdash G, A^{\bot}$ $^{\bot}$R $H, A \vdash G$
Allgemeine Regeln	
$H, H' \vdash G, G'$ `cut` $H \vdash G, A$ $H', A \vdash G'$	$A \vdash A$ `axiom`

Additives Fragment	
$H, \mathbf{0} \vdash G$ **0**L	$H \vdash G, \top$ ⊤R
$H, A \& B' \vdash G$ &L1 $H, A \vdash G$ $H, A \& B \vdash G$ &L2 $H, B \vdash G$	$H \vdash G, A \& B$ &R $H \vdash G, A$ $H \vdash G, B$
$H, A \oplus B \vdash G$ ⊕L $H, A \vdash G$ $H, B \vdash G$	$H \vdash G, A \oplus B$ ⊕R1 $H \vdash G, A$ $H \vdash G, A \oplus B$ ⊕R2 $H \vdash G, B$
Exponentiale	
$H, !A \vdash G$ `w-L` $H \vdash G$	$H \vdash G, ?A$ `w-R` $H \vdash G$
$H, !A \vdash G$ `c-L` $H, !A, !A \vdash G$	$H \vdash G, ?A$ `c-R` $H \vdash G, ?A, ?A$
$H, !A \vdash G$!L $H, A \vdash G$	$H^{!} \vdash G^{?}, !A$!R $H^{!} \vdash G^{?}, A$
$H^{!}, ?A \vdash G^{?}$?L $H^{!}, A \vdash G^{?}$	$H \vdash G, ?A$?R $H \vdash G, A$

Regeln für Quantoren wie bei Prädikatenlogik

H, H', G, G' sind multi-sets. Weakening und contraction gibt es nur für Exponentiale

$H^{!}$ und $G^{?}$ bestehen nur aus den entsprechenden Exponentialen

Die Regel `axiom` *darf keine verbleibenden Hypothesen haben. Es gibt keine* **0**R *oder* ⊤L *regel*

Beweis für $6€ \multimap Cig, 6€ \multimap ABC \vdash 12€ \multimap (Cig \otimes ABC)$

```
⊢ (6€⊸Cig) ⊸ (6€⊸ABC) ⊸ ((6€⊗6€)⊸(Cig⊗ABC))   BY ⊸R
1. 6€⊸Cig ⊢ (6€⊸ABC) ⊸ ((6€⊗6€)⊸(Cig⊗ABC))     BY ⊸R
1.1. 6€⊸Cig, 6€⊸ABC ⊢ (6€⊗6€)⊸(Cig⊗ABC)        BY ⊸R
1.1.1. 6€⊸Cig, 6€⊸ABC, 6€⊗6€ ⊢ Cig⊗ABC          BY ⊗L
1.1.1.1. 6€⊸Cig, 6€⊸ABC, 6€, 6€ ⊢ Cig⊗ABC       BY ⊗R
1.1.1.1.1. 6€⊸Cig, 6€ ⊢ Cig                     BY ⊸L
1.1.1.1.1.1. 6€ ⊢ 6€                            BY axiom
1.1.1.1.1.2. Cig ⊢ Cig                          BY axiom
1.1.1.1.2. 6€⊸ABC, 6€ ⊢ ABC                     BY ⊸L
1.1.1.1.1.1. 6€ ⊢ 6€                            BY axiom
1.1.1.1.1.2. ABC ⊢ ABC                          BY axiom
```

Beweis für $!(6€ \multimap Cig) \vdash 12€ \multimap (Cig \otimes Cig)$

$\vdash\ !(6€ \multimap Cig)\ \multimap\ ((6€ \otimes 6€) \multimap (Cig \otimes Cig))$	BY $\multimap$R
1. $!(6€ \multimap Cig) \vdash ((6€ \otimes 6€) \multimap (Cig \otimes Cig))$	BY $\multimap$R
1.1. $!(6€ \multimap Cig),\ 6€ \otimes 6€ \vdash Cig \otimes Cig$	BY c-L
1.1.1. $!(6€ \multimap Cig),\ !(6€ \multimap Cig),\ 6€ \otimes 6€ \vdash Cig \otimes Cig$	BY !L
1.1.1.1. $6€ \multimap Cig,\ !(6€ \multimap Cig),\ 6€ \otimes 6€ \vdash Cig \otimes Cig$	BY !L
1.1.1.1.1. $6€ \multimap Cig,\ 6€ \multimap Cig,\ 6€ \otimes 6€ \vdash Cig \otimes Cig$	BY $\otimes$L
1.1.1.1.1.1 $6€ \multimap Cig,\ 6€ \multimap Cig,\ 6€,\ 6€ \vdash Cig \otimes Cig$	BY $\otimes$R
1.1.1.1.1.1.1. $6€ \multimap Cig,\ 6€ \vdash Cig$	BY $\multimap$L
1.1.1.1.1.1.1.1. $6€ \vdash 6€$	BY axiom
1.1.1.1.1.1.1.2. $Cig \vdash Cig$	BY axiom
1.1.1.1.1.1.2. $6€ \multimap Cig,\ 6€ \vdash Cig$	BY $\multimap$L
1.1.1.1.1.1.2.1. $6€ \vdash 6€$	BY axiom
1.1.1.1.1.1.2.2. $Cig \vdash Cig$	BY axiom

Beweis für $((A \parr A^{\perp}) \otimes (B \parr A)) \parr (A \parr B)^{\perp}$

$\vdash\ ((A \parr A^{\perp}) \otimes (B \parr A)) \parr (A \parr B)^{\perp}$	BY $\parr$R
1. $\vdash\ (A \parr A^{\perp}) \otimes (B \parr A),\ \ (A \parr B)^{\perp}$	BY $^{\perp}$R
1.1. $A \parr B\ \vdash\ (A \parr A^{\perp}) \otimes (B \parr A)$	BY $\otimes$R
1.1.1. $\vdash\ A \parr A^{\perp}$	BY $\parr$R
1.1.1.1. $\vdash\ A, A^{\perp}$	BY $^{\perp}$R
1.1.1.1.1. $A\ \vdash\ A$	BY axiom
1.1.2. $A \parr B\ \vdash\ B \parr A$	BY $\parr$R
1.1.2.1. $A \parr B\ \vdash\ B, A$	BY $\parr$L
1.1.2.1.1. $B\ \vdash\ B$	BY axiom
1.1.2.1.2. $A\ \vdash\ A$	BY axiom

Matrixbasierte Beweissuche für Lineare Logik

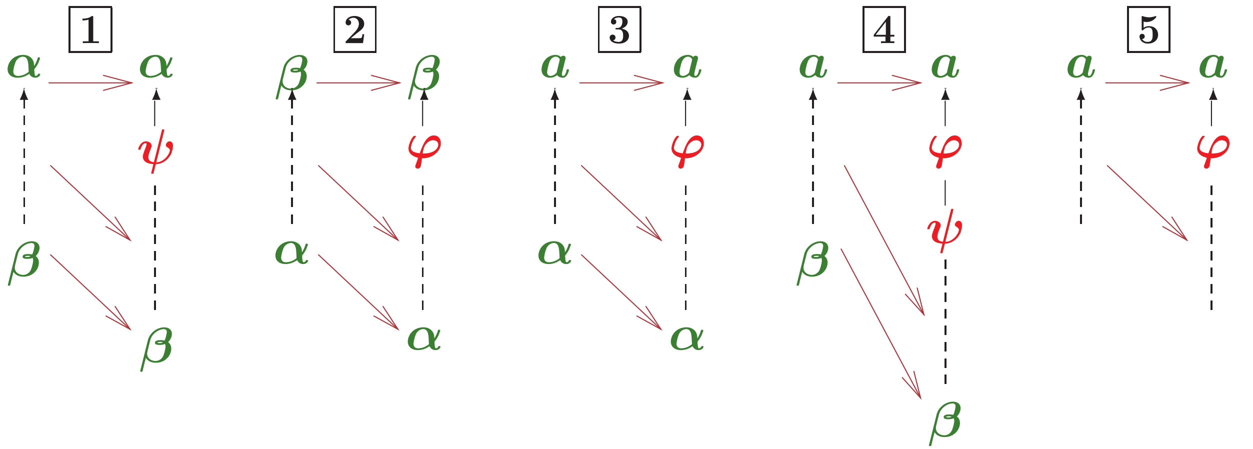
- **Erweiterte Matrixcharakterisierung der Gültigkeit**
  - **F ist gültig gdw. alle Pfade durch F komplementär**
  - Komplementaritätsbegriff muß ergänzt werden
    - · Unifizierbarkeit der konnektierten Terme (wenn Prädikatenlogik)
    - · Erreichbarkeit der Literale bei exakter Verwendung der Ressourcen
- **Erweitertes Beweissuchverfahren**
  - Unverändertes konnektionen-orientiertes Pfadüberprüfungsverfahren
  - Erweiterter Komplementaritätstest
    - · Termunifikation liefert Substitution σ_Q von γ-Variablen durch Terme
    - · Präfixunifikation liefert Substitution σ_L für lineare Präfixe
    - · Zusätzliche Linearitätstests sichern Ressourcenverwendung
  - Substitutionen liefern Reihenfolge von Regeln im Sequenzenbeweis
- **Bisher nur für multiplikatives Fragment gelungen**
  - Charakterisierung + Beweisverfahren für $\mathcal{MLL}$ ($\multimap$, $\otimes$, $\parr$, $\mathbf{1}$, $\bot$, $^\bot$)
  - Matrixcharakterisierung für $\mathcal{MELL}$ ($\mathcal{MLL}$ + Exponentiale $!$, $?$))
  - Für $\mathcal{ALL}$ ($\&$, $\oplus$, $\mathbf{0}$, $\top$, $^\bot$) gibt es nur separate Verfahren $\mapsto$ Galmiche

Lineare Positionsbäume

- **Zuordnung von Typen und Polaritäten** (analog zu $\wedge$, $\vee$, $\Rightarrow$, $\neg$)
  - Wurzel hat Polarität F
  - Nachfolgerpolaritäten und -typen werden tabellarisch bestimmt

α	$(A{\otimes}B)^T$	$(A{⅋}B)^F$	$(A{\multimap}B)^F$	β	$(A{\otimes}B)^F$	$(A{⅋}B)^T$	$(A{\multimap}B)^T$	o	$(A^{\perp})^T$	$(A^{\perp})^F$
α_1	A^T	A^F	A^T	β_1	A^F	A^T	A^F	o_1	A^F	A^T
α_2	B^T	B^F	B^F	β_2	B^F	B^T	B^T			

- **Ergänze lineare Positionen φ, ψ zum Formelbaum**
  - Wende die folgenden fünf Erweiterungsregeln solange wie möglich an



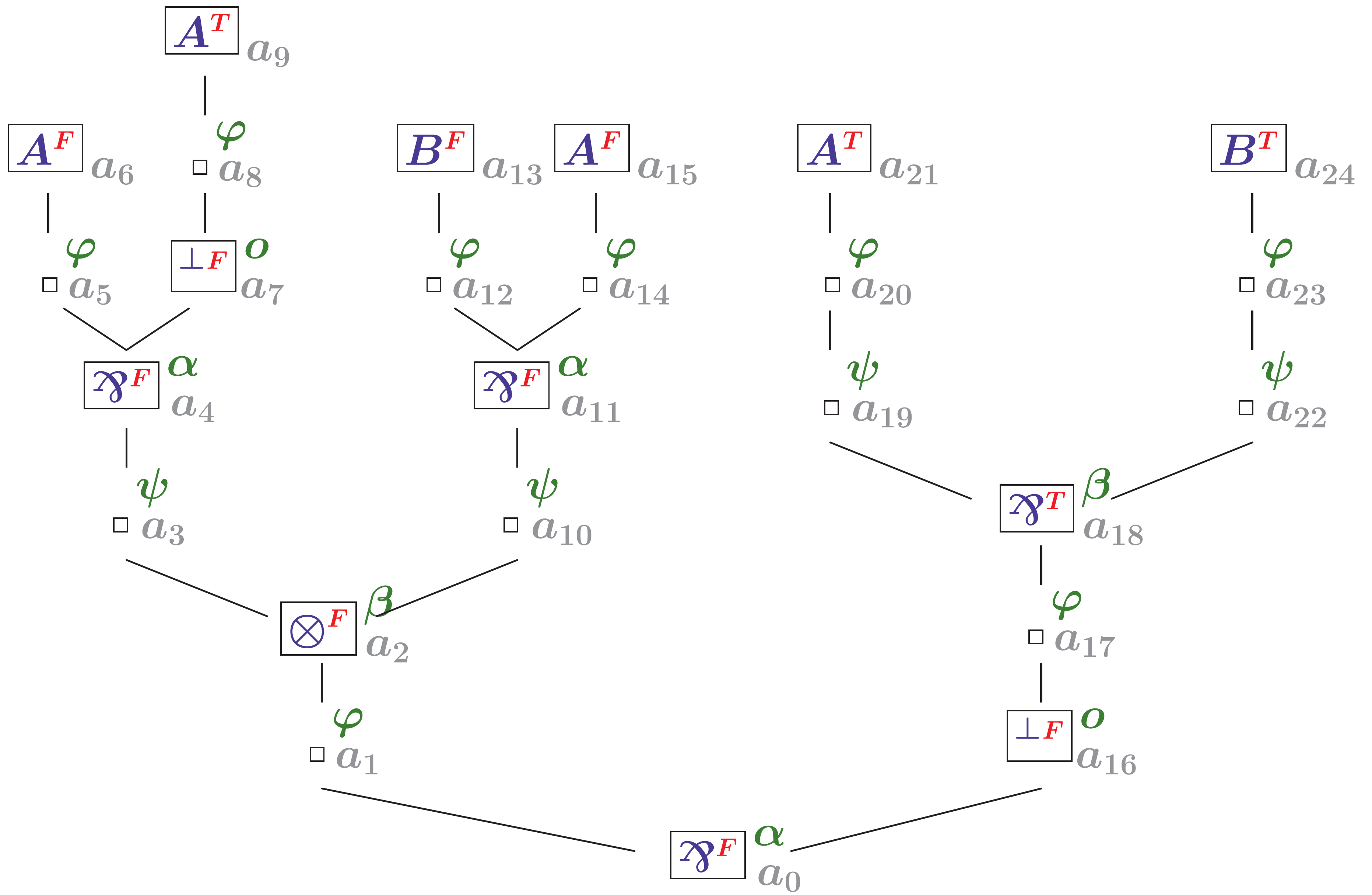
α, β, a stehen für α-, β-, atomare Positionen

- - - steht für beliebig viele o-Positionen

z.B. $\boxed{1}$: Füge ψ-Position unmittelbar vor α ein, wenn vor α ein β vorkommt

  - Entstehender Baum ist **linearer Positionsbaum** der Formel
  - φ-Positionen gelten als Variablen, ψ-Positionen als Konstante

Positionsbaum für $((A \parr A^{\perp}) \otimes (B \parr A)) \parr (A \parr B)^{\perp}$



Lineare Präfixe

- **Lineare Positionen codieren Linearitätsbedingungen**
  – Trennen α-Ebenen von β-Ebenen und atomare Positionen vom Rest
  – Helfen sicherzustellen, daß jede Position genau einmal verwendet wird und Sequenzenregeln in der richtigen Reihenfolge angewandt werden
  – Stellen sicher, daß Ressourcen korrekt aufgeteilt werden können
  – Technische Verwendung ähnlich wie bei konstruktiver Logik
- **Bestimme lineares Präfix eines Atoms P**
  – Liste der linearen Positionen zwischen Wurzel und P
- **Definiere lineare Substitution σ_L**
  – Abbildung von φ-Positionen in Strings über linearen Positionen
  – σ_L induziert Reduktionsordnung $\sqsubseteq_L$ auf linearen Positionen:
  Ist $\sigma_L(u) = v_1 ... v_n$ dann gilt $v_i \sqsubseteq_L u$ für jede ψ-Position v_i
- **Lineare Multiplizität nur für Exponentiale ($\mathcal{MELL}$)**

Komplementaritätsbedingungen

Komplexeres Kriterium als in konstruktiver Logik

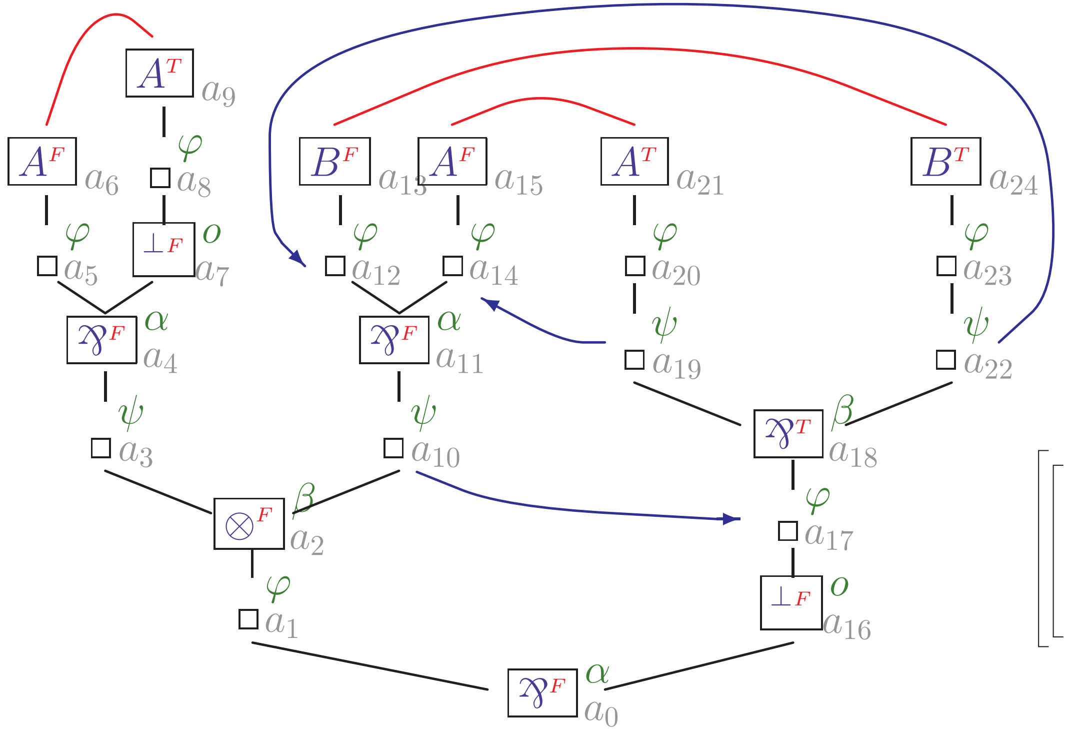
- **Aufspannende Menge $\mathcal{Con}$ von Konnektionen**
  - Jeder Pfad durch Positionsbaum enthält Konnektion aus $\mathcal{Con}$
- **Komplementarität jeder Konnektionen $\{u, v\}$**
  - $\sigma_L(pre(u)){=}\sigma_L(pre(v))$ (ggf. auch Komplementarität unter σ_Q)
- **Zulässigkeit von σ_L**
  - Ist $\sigma_L(pre(u)) = s_1 v s_2$, dann muß $\sigma_L(pre(v)){=}s_1 v$ sein
- **Linearität von $\mathcal{Con}$**
  - Jede atomare Position ist maximal einmal konnektiert
  - Keine Ressource wird mehrfach verwendet
- **Relevanz von F für $\mathcal{Con}$**
  - Jede atomare Position ist mindestens einmal konnektiert
  - Jede Ressource wird auch eingesetzt
- **Minimalität von $\mathcal{Con}$**
  - Keine echte Teilmenge von $\mathcal{Con}$ spannt den Positionsbaum auf

MATRIXCHARAKTERISIERUNG FÜR $\mathcal{MLL}$

- **Wichtige Erkenntnisse**
  - Ist eine aufspannde Paarung $\mathcal{C}on$ σ_L-komplementär für F, dann ist die induzierte Ordnung $\lhd = (< \cup \sqsubset_L)^+$ irreflexiv
  - Ist eine aufspannde Paarung $\mathcal{C}on$ σ_L-komplementär für F und σ_L zulässig, dann ist $\mathcal{C}on$ linear (vereinfacht die Matrixcharakterisierung)
  - Ist eine aufspannde Paarung $\mathcal{C}on$ σ_L-komplementär für F, linear und relevant, dann ist $\mathcal{C}on$ genau dann minimal, wenn $|\mathcal{C}on| = \#_\beta(F) + 1$ (vereinfacht den Test auf Minimalität)

Eine Formel F ist gültig in $\mathcal{MLL}$, wenn es eine zulässige Substitution σ_L und eine minimale Menge $\mathcal{C}on$ von σ_L-komplementären Konnektionen gibt, so daß F relevant für $\mathcal{C}on$ ist und jeder Pfad durch den Positionsbaum von F ein Element von $\mathcal{C}$ enthält

Matrixbeweis für $((A \parr A^\perp) \otimes (B \parr A)) \parr (A \parr B)^\perp$



$$\left[\left[\begin{matrix}\left[A_1a_3A_5\!:\!A^F \quad A_1a_3A_8\!:\!A^T\right] \\ \left[A_1a_{10}A_{12}\!:\!B^F \quad A_1a_{10}A_{14}\!:\!A^F\right]\end{matrix}\right] \left[\begin{matrix}A_{17}a_{19}A_{20}\!:\!A^T \\ A_{17}a_{22}A_{23}\!:\!B^T\end{matrix}\right]\right]$$

- **Alle vier Pfade sind durch drei Konnektion aufgespannt**
  - $\mathcal{C}on = \{\{a_6, a_9\},\ \{a_{15}, a_{21}\},\ \{a_{13}, a_{24}\}\}$
  - $\mathcal{C}on$ **ist minimal** (es gibt zwei β-Positionen in F)
  - F **ist relevant für** $\mathcal{C}on$ (alle atomaren Positionen erscheinen in $\mathcal{C}on$)
  - $\mathcal{C}on$ **ist** σ_L**-komplementär** für die zulässige Substition
    $\sigma_L = \{\varepsilon/A_1,\ \varepsilon/A_5,\ \varepsilon/A_8,\ a_{22}/A_{12},\ a_{19}/A_{14},\ a_{10}/A_{17},\ \varepsilon/A_{20},\ \varepsilon/A_{23}\}$
- **Die Formel ist gültig in $\mathcal{MLL}$**

Extensionsverfahren für $\mathcal{MLL}$

- **Pfadüberprüfungsverfahren wird geringfügig erweitert**
  - Regeln aus Einheit 6 müssen verwendete Konnektionen verwalten
  - Linearitätstest wird dynamisch im Extensionsschritt durchgeführt
    Guter Filter, um erfolglose Suchpfade frühzeitig zu eliminieren
  - Keine Reduktionsregel (Anwendung würde Linearität verletzen)
  - Relevanz und Minimalität muß separat am Ende geprüft werden
- **Komplementaritätstest wird erweitert**
  - Termunifikationsverfahren entfällt für $\mathcal{MLL}$
  - Präfixunifikationsverfahren mit Logik-spezifischen Regeln
  - Überprüfung der Zulässigkeit separat am Ende
- **Verfahren bisher nur auf $\mathcal{MLL}$ anwendbar**
  - Matrixcharakterisierung für $\mathcal{MELL}$ ist komplexer und benötigt
    Multiplizitäten, Weakening-Tabelle, aufwendigere Präfixunifikation

Konnektionskalkül für lineare Matrizen

- **Regeln verwalten Objekte der Form $\mathcal{C}, \mathcal{M}, \mathcal{P}, \sigma, \mathcal{C}on$**
  - Aktuelle Klausel, Restmatrix, Pfad, Substitution, Konnektionsmenge
  - Substitution σ und Konnektionsmenge $\mathcal{C}on$ werden lokal bestimmt, und an Unterziele weitergereicht und von dort zurückgereicht
- **Startregel**
  - Für Beweis von $\mathcal{M}$ wähle Startziel $\mathcal{C}{=}\mathcal{M}$, setze $\mathcal{P}{=}\{\}$, $\sigma{=}[\,]$, $\mathcal{C}on{=}\{\,\}$

$$
\boxed{
\begin{array}{lr}
\vdash \mathcal{M} & \text{ret } \sigma, \mathcal{C}on \\
\qquad \vdash \mathcal{M}, \mathcal{M}, \{\}, [\,], \{\,\} & \text{ret } \sigma, \mathcal{C}on \\
\qquad \vdash \text{admissible}(\sigma) & \\
\qquad \vdash |\mathcal{C}on| = \#_\beta(\mathcal{M}) + 1 & \\
\qquad \vdash \text{atoms}(\mathcal{M}) \subseteq \bigcup \mathcal{C}on &
\end{array}
}
\quad \texttt{Start}
$$

  - Substitution und Konnektionsmenge werden im Unterziels bestimmt und auf Zulässigkeit, Minimalität und Relevanz geprüft

Regeln des linearen Konnektionskalküls

- **Bereinigung: Abschluß des aktuellen Pfades**
  - $\mathcal{C},\mathcal{M},\mathcal{P},\mathcal{C}on$ ist beweisbar, wenn $\mathcal{C}$ leer ist

    $\boxed{\vdash \{\},\mathcal{M},\mathcal{P},\sigma,\mathcal{C}on \qquad \text{ret } \sigma,\mathcal{C}on}$ Axiom

  - Regel schließt einen Ast im Konnektionskalkülbeweis
  - Eingabesubstitution und -konnektionsmenge werden Rückgabewert

- **Extension: Verlängerung des aktuellen Pfades**
  - $\mathcal{C},\mathcal{M},\mathcal{P},\mathcal{C}on$ ist beweisbar, wenn es $L\in\mathcal{C}$, $L'\sim_\alpha\mathcal{P}\cup\{L\}$ gibt, sodaß $c=\{L,L'\}$ komplementär, $c\in\mathcal{C}on$ oder $c\cap\bigcup\mathcal{C}on=\emptyset$ (Linearität) und $L'_\beta(\mathcal{P}\cup\{L\}),\mathcal{M},\mathcal{P}\cup\{L\}$ sowie $\mathcal{C}\cap L_\beta,\mathcal{M},\mathcal{P}$ beweisbar sind

$$\boxed{\begin{array}{ll}
\vdash \mathcal{C},\mathcal{M},\mathcal{P},\sigma,\mathcal{C}on & \text{ret } \sigma_2,\mathcal{C}on_2\\
\quad\vdash L'_\beta(\mathcal{P}\cup\{L\}),\mathcal{M},\mathcal{P}\cup\{L\},\sigma\rho,\mathcal{C}on\cup\{L,L'\} & \text{ret } \sigma_1,\mathcal{C}on_1\\
\quad\vdash \mathcal{C}\cap L_\beta,\mathcal{M},\mathcal{P},\sigma_1,\mathcal{C}on_1 & \text{ret } \sigma_2,\mathcal{C}on_2\\
\quad\vdash L\in\mathcal{C} & \\
\quad\vdash L'\sim_\alpha\mathcal{P}\cup\{L\} & \\
\quad\vdash \sigma\rho(L')=\sigma\rho(\overline{L}) & \\
\quad\vdash \{L,L'\}\in\mathcal{C}on \vee \{L,L'\}\cap\bigcup\mathcal{C}on=\emptyset &
\end{array}}$$ Extension L,L',ρ

  - Bisherige Substitution und Konnektionsmenge kann erweitert werden
  - Resultat des ersten Teilziels wird weitergereicht, dann zurückgegeben

Extensionsbeweis für $((A \parr A^{\perp}) \otimes (B \parr A)) \parr (A \parr B)^{\perp}$

$\vdash \mathcal{M}$	BY Start	$\sigma_3,\ \{\{A^F, A^T\}, \{B^F, B^T\}, \{A^T, A^F\}\}$
1. $\vdash \mathcal{M},\ \mathcal{M},\ \{\},\ [],\ \{\}$	BY Extension A^F, A^T, ρ_1	$\sigma_3,\ \{\{A^F, A^T\}, \{B^F, B^T\}, \{A^T, A^F\}\}$
1.1. $\vdash \{\},\ \mathcal{M},\ \{A^F\},\ \sigma_1,\ \{\{A^F, A^T\}\}$	BY Axiom	$\sigma_1,\ \{\{A^F, A^T\}\}$
1.2. $\vdash \{B^F, A^F\},\ \mathcal{M},\ \{\},\ \sigma_1,\ \{\{A^F, A^T\}\}$	BY Extension B^F, B^T, ρ_2	$\sigma_3,\ \{\{A^F, A^T\}, \{B^F, B^T\}, \{A^T, A^F\}\}$
1.2.1. $\vdash \{A^T\},\ \mathcal{M},\ \{B^F\},\ \sigma_2,\ \{\{A^F, A^T\}, \{B^F, B^T\}\}$		
	BY Extension A^T, A^F, ρ_3	$\sigma_3,\ \{\{A^F, A^T\}, \{B^F, B^T\}, \{A^T, A^F\}$
1.2.1.1. $\vdash \{\},\ \mathcal{M},\ \{B^F, A^T\},\ \sigma_3,\ \{\{A^F, A^T\}, \{B^F, B^T\}, \{A^T, A^F\}\}$		
	BY Axiom	$\sigma_3,\ \{\{A^F, A^T\}, \{B^F, B^T\}, \{A^T, A^F\}$
1.2.1.2. $\vdash \{\},\ \mathcal{M},\ \{B^F\},\ \sigma_3,\ \{\{A^F, A^T\}, \{B^F, B^T\}, \{A^T, A^F\}\}\}$		
	BY Axiom	$\sigma_3,\ \{\{A^F, A^T\}, \{B^F, B^T\}, \{A^T, A^F\}$
1.2.2. $\vdash \{\},\ \mathcal{M},\ \{\},\ \sigma_3,\ \{\{A^F, A^T\}, \{B^F, B^T\}\}$		
	BY Axiom	$\sigma_3,\ \{\{A^F, A^T\}, \{B^F, B^T\}, \{A^T, A^F\}$

$\rho_1 = \{\varepsilon/A_5,\ \varepsilon/A_8\}$ $\qquad$ $\rho_3 = \{a_{22}/A_{12},\ a_{19}/A_{14},\ a_{10}/A_{17},\ \varepsilon/A_{20}\}$

$$\left[\left[\begin{matrix} \left[A_1a_3A_5{:}A^F \quad A_1a_3A_8{:}A^T \right] \\ \left[A_1a_{10}A_{12}{:}B^F \quad A_1a_{10}A_{14}{:}A^F \right] \end{matrix} \right] \begin{bmatrix} A_{17}a_{19}A_{20}{:}A^T \\ A_{17}a_{22}A_{23}{:}B^T \end{bmatrix} \right]$$

$\rho_2 = \{\varepsilon/A_1,\ a_{22}/A_{12},\ \varepsilon/A_{23}\}$

$$\begin{aligned} \sigma_1 &= \rho_1 &&= \{\varepsilon/A_5,\ \varepsilon/A_8\} \\ \sigma_2 &= \sigma_1\rho_2 &&= \{\varepsilon/A_1,\ \varepsilon/A_5,\ \varepsilon/A_8,\ a_{22}/A_{12},\ \varepsilon/A_{23}\} \\ \sigma_3 &= \sigma_2\rho_3 &&= \{\varepsilon/A_1,\ \varepsilon/A_5,\ \varepsilon/A_8,\ a_{22}/A_{12}, \\ & && \quad a_{19}/A_{14},\ a_{10}/A_{17},\ \varepsilon/A_{20},\ \varepsilon/A_{23}\} \end{aligned}$$

LINEARE PRÄFIX-UNIFIKATION

Unifiziere Präfix-Strings konnektierter Atome

- **Ähnlich zur Präfix-Unifikation für konstruktive Logik**
  - Gleiches Transformationsverfahren im Stil von Martelli-Montanari
  - Regeln wie bei konstruktiver Logik
  - Regeln R_2, R_4, R_6, R_7 können entfallen, da alle $\mathcal{MLL}$-Präfixe die Form $\psi_1\varphi_1\psi_2\varphi_2\ldots\psi_n\varphi_n$ oder $\varphi_1\psi_2\varphi_2\ldots\psi_n\varphi_n$ haben

- **Transformationsregeln für Lineare Logik**

R_1	$\{\varepsilon = \varepsilon \vert \varepsilon\}, \sigma$	$\rightsquigarrow$	$\{\}, \sigma$
R_3	$\{Xs = \varepsilon \vert Xt\}, \sigma$	$\rightsquigarrow$	$\{s = \varepsilon \vert t\}, \sigma$
R_5	$\{Vs = z \vert \varepsilon\}, \sigma$	$\rightsquigarrow$	$\{s = \varepsilon \vert \varepsilon\}, [z/V] \cup \sigma$
R_8	$\{Vs^+ = \varepsilon \vert V_1t\}, \sigma$	$\rightsquigarrow$	$\{V_1t = V \vert s^+\}, \sigma$
R_9	$\{Vs^+ = z^+ \vert V_1t\}, \sigma$	$\rightsquigarrow$	$\{V_1t = V' \vert s^+\}, [z^+V'/V] \cup \sigma$
R_{10}	$\{Vs = z \vert Xt\}, \sigma$	$\rightsquigarrow$	$\{Vs = zX \vert t\}, \sigma$ ($V{\neq}X$, und $s{=}\varepsilon$, $t{\neq}\varepsilon$, oder X Konstante)

Lineare Logik im Rückblick

- **Ressourcenlogik mit andersartigen Grundoperatoren**
  - Additive, multiplikative und exponentielle Operatoren
  - Logik höherer Stufe
  - Klassische und nichtklassische Logik simulierbar
- **Modifikation des Matrixkalküls erforderlich**
  - Formelbaum muß um spezielle lineare Positionen erweitert werden
  - Komplementarität verlangt Unifizierbarkeit linearer Präfixe
  - Ressourcenverwaltung verlangt Linearität, Relevanz, Minimalität
  - Charakterisierung bisher nur für $\mathcal{MLL}$ und $\mathcal{MELL}$ möglich
    Integration des additiven Fragments bisher nicht gelungen
  - Wird mit $\mathcal{MELL}$ die Grenze von Matrixkalkülen erreicht?
- **Erweiterung des Extensionsverfahrens entsprechend**
  - Pfadüberprüfungsverfahren muß um Linearitätstest erweitert werden
  - Komplementaritätstest benötigt lineare Präfix-Unifikation
  - Implementiert als kompakte Beweiser linTap und linCop für $\mathcal{MLL}$

Higher-order Logik

Logik höherer Stufe

- **Logik erster Stufe hat nur einfache Variablen**
  – Keine Quantifizierung über Funktions- oder Prädikatensymbole erlaubt
- **Konstrukte höherer Stufe kommen in der Realität vor**
  – Funktionen (2. Stufe): “Bestimme $x+2$ bei Eingabe x” $\lambda x.x+2$
  – Induktionsprinzip: $\forall P.P(0) \wedge (\forall y{:}\mathbb{N}.P(y) \Rightarrow P(y+1)) \Rightarrow \forall x{:}\mathbb{N}.P(x)$
  – Zwischenwertsatz: $\forall f{:}\mathbb{R}\to\mathbb{R}.\forall a < b{:}\mathbb{R}.(f(a)>0 \wedge f(b)<0) \Rightarrow \exists x{:}\mathbb{R}.f(x)=0$
  – Funktionale (3. Stufe): Ableitungsoperator $\lambda f.df/dx$
  – Quantifizierung über Funktionale, ...
- **Logik höherer Stufe hat keine Einschränkungen**
  – Extrem einfache Syntax: x, $\lambda x.t$, $f(a)$, $\forall x\, P$, $P \Rightarrow Q$
  – Fester Auswertungsmechanismus Reduktion: $(\lambda x.t)(a) \longrightarrow t[a/x]$
- **Logik höherer Stufe ist minimale Grundlagentheorie**
  – Keine Abstützung auf Mengentheorie erforderlich
  – Reduktion erklärt Wert von Ausdrücken
  – Mathematische Konzepte werden nicht über Axiome erklärt sondern als definitorische Abkürzung für komplexe Terme (logizistischer Ansatz)

Logik höherer Stufe – Syntax

- **Alphabet für erlaubte Symbole**
  - $\mathcal{V}$: Variablensymbole
- **Terme**
  - Variablen $\boldsymbol{x} \in \mathcal{V}$
  - $\boldsymbol{\lambda x.t}$, wobei $x \in \mathcal{V}$ und t Term — λ-Abstraktion
  - $\boldsymbol{f\ t}$, wobei t und f Terme — Applikation
  - $(\boldsymbol{t})$, wobei t Term
  - $(\boldsymbol{P \Rightarrow Q})$, wobei P,Q Terme
  - $(\forall \boldsymbol{x}\ \boldsymbol{P})$, wobei $x \in \mathcal{V}$ und P Term
- **Konventionen**
  - Applikation bindet stärker als λ-Abstraktion
  - Applikation ist links-assoziativ: $\boldsymbol{f\ t_1\, t_2 \hat{=} (f\ t_1)\ t_2}$
  - Notation $f(t_1,\ldots t_n)$ entspricht iterierter Applikation $f\ \ t_1\ldots t_n$

Logik höherer Stufe – (Wert-)Semantik

- **α-Konversion**: Umbenennung gebundener Variablen
  - Ersetze Teilterm der Gestalt $\lambda x.t$ durch $\lambda z.t[z/x]$ (z neue Variable)
  - Ersetze Teilterm der Gestalt $\forall x\ P$ durch $\forall z\ P[z/x]$
  - Terme t und u sind kongruent (α-konvertibel), wenn sie auseinander durch endlich viele Umbenennungen gebundener Variablen entstehen
- **(β-)Reduktion**: Auswertung von Termen
  - Ersetze Teilterm der Gestalt $(\lambda x.t)(s)$ (Redex) durch $t[s/x]$ (Kontraktum)
  - t ist reduzierbar auf u ($\boldsymbol{t \xrightarrow{*} u}$), wenn u aus t durch endlich viele Reduktionen und Umbenennungen entsteht ($\hat{=}$ Termersetzung)
- **(Semantische) Gleichheit $t = s$**
  - Es gibt einen Term u mit $t \xrightarrow{*} u$ und $s \xrightarrow{*} u$
- **Normalform von t**: Wert eines Terms
  - Irreduzibler (Redex-freier) Term u mit $t = u$
  - Der Wert eines Terms ist eindeutig ($\xrightarrow{*}$ ist konfluent)
  - Nicht jeder Term hat einen Wert (keine starke Normalisierbarkeit)
- **Semantik von $\forall$ und $\Rightarrow$ analog zur Prädikatenlogik**

Definition logischer Standardkonzepte

Junktoren werden zu benutzerdefinierten Funktionen

Konjunktion	$\wedge$	$\equiv$	$\lambda A.\lambda B.\forall P\,(A \Rightarrow (B \Rightarrow P)) \Rightarrow P$
Disjunktion	$\vee$	$\equiv$	$\lambda A.\lambda B.\forall P\,((A \Rightarrow P) \Rightarrow (B \Rightarrow P)) \Rightarrow P$
Falschheit	$\perp$	$\equiv$	$\forall P\,P$ (*intuitionistisch*)
Negation	$\neg$	$\equiv$	$\lambda A.\forall P\,(A \Rightarrow P)$
Existenzquantor	$\exists \boldsymbol{x}\ \boldsymbol{A}$	$\equiv$	$\forall P\,(\forall x\,(A \Rightarrow P)) \Rightarrow P$
Gleichheit	$\doteq$	$\equiv$	$\lambda x.\lambda y.\forall P\,(\,P(x) \Rightarrow P(y)\,)$

Definition von Zahlen: Church Numerals

$$
\begin{array}{lll}
\overline{n} & \equiv \lambda f.\lambda x.\, f^n\, x & \equiv \lambda f.\lambda x.\, \underbrace{f\,(f \ldots (f\,x) \ldots)}_{n\text{-mal}} \\
\mathsf{s} & \equiv \lambda n.\lambda f.\lambda x.\, n\, f\, (f\, x) & \\
\mathsf{add} & \equiv \lambda m.\lambda n.\lambda f.\lambda x.\, m\, f\, (n\, f\, x) & \\
\mathsf{mul} & \equiv \lambda m.\lambda n.\lambda f.\lambda x.\, m\, (n\, f)\, x & \\
\mathsf{exp} & \equiv \lambda m.\lambda n.\lambda f.\lambda x.\, n\, m\, f\, x & \\
\mathsf{zero} & \equiv \lambda n.\, n\, (\lambda n.\mathsf{F})\, \mathsf{T} & \\
\mathsf{p} & \multicolumn{2}{l}{\equiv \lambda n.\, \mathsf{snd}(\,(\mathtt{n}\ (\lambda \mathtt{fx.}\ (\mathsf{s},\ \mathsf{match}\ \mathtt{fx}\ \mathsf{with}\ (\mathtt{f}\,,\mathtt{x}) \mapsto \mathtt{f}\ \mathtt{x}))\ (\lambda \mathtt{z.}\overline{\mathtt{0}},\ \overline{\mathtt{0}}))\,)} \\
N & \equiv \lambda x.\forall P\ (\forall y\, P(y) \Rightarrow P(\mathsf{s}\, y)) \Rightarrow P(\overline{0}) \Rightarrow P(x) & \boxed{N(x) \mathrel{\hat{=}} x \in \mathbb{N}}
\end{array}
$$

Abkürzungen

$$
\begin{array}{ll}
\mathsf{T} & \equiv \lambda x.\lambda y.x \\
\mathsf{F} & \equiv \lambda x.\lambda y.y \\
\mathsf{if}\ b\ \ \mathsf{then}\ s\ \ \mathsf{else}\ t & \equiv b\, s\, t \\
(s\,,t) & \equiv \lambda p.\, p\, s\, t \\
\mathsf{match}\ pair\ \mathsf{with}\ (x\,,y) \mapsto t & \equiv pair\ (\lambda x.\lambda y.t)
\end{array}
$$

Church Numerals: Auswertung von Funktionen

$$
\begin{array}{lcll}
\mathsf{s}\,\overline{n} & \equiv & (\lambda n.\lambda f.\lambda x.\ n\,f\,(f\,x))\,(\lambda f.\lambda x.\ f^{n}\,x) & \\
 & \longrightarrow & \lambda f.\lambda x.\ (\lambda f.\lambda x.\ f^{n}\,x)\,f\,(f\,x) & \\
 & \longrightarrow & \lambda f.\lambda x.\ (\lambda x.\ f^{n}\,x)\,(f\,x) & \\
 & \longrightarrow & \lambda f.\lambda x.\ f^{n}\,(f\,x) & \\
 & \longrightarrow & \lambda f.\lambda x.\ f^{n+1}\,x & \equiv\ \overline{n+1}
\end{array}
$$

$$
\begin{array}{lcll}
\mathsf{add}\,\overline{m}\,\overline{n} & \equiv & (\lambda m.\lambda n.\lambda f.\lambda x.\ m\,f\,(n\,f\,x))\,\overline{m}\,\overline{n} & \\
 & \longrightarrow & (\lambda n.\lambda f.\lambda x.\ \overline{m}\,f\,(n\,f\,x))\,\overline{n} & \\
 & \longrightarrow & \lambda f.\lambda x.\ \overline{m}\,f\,(\overline{n}\,f\,x) & \\
 & \equiv & \lambda f.\lambda x.\ (\lambda f.\lambda x.\ f^{m}\,x)\,f\,(\overline{n}\,f\,x) & \\
 & \longrightarrow & \lambda f.\lambda x.\ (\lambda x.\ f^{m}\,x)\,(\overline{n}\,f\,x) & \\
 & \longrightarrow & \lambda f.\lambda x.\ f^{m}\,(\overline{n}\,f\,x) & \\
 & \equiv & \lambda f.\lambda x.\ f^{m}\,((\lambda f.\lambda x.\ f^{n}\,x)\,f\,x) & \\
 & \longrightarrow & \lambda f.\lambda x.\ f^{m}\,((\lambda x.\ f^{n}\,x)\,x) & \\
 & \longrightarrow & \lambda f.\lambda x.\ f^{m}\,(f^{n}\,x) & \\
 & \longrightarrow & \lambda f.\lambda x.\ f^{m+n}\,x & \equiv\ \overline{m+n}
\end{array}
$$

Beweisführung in Logik höherer Stufe

Extensionsverfahren ähnlich wie bei Prädikatenlogik

- **Erweiterte Matrixcharakterisierung der Gültigkeit**
  - ***F* ist gültig gdw. alle Pfade durch *F* komplementär**
  - Komplementarität mit erweitertem Substitutionsbegriff
    - · Prädikats- und Funktionssymbole dürfen ersetzt werden
    - · Konnektierte Terme müssen nur semantisch gleich sein
- **Erweitertes Beweissuchverfahren** → TPS (Andrews)
  - Konnektionen-orientiertes Pfadüberprüfungsverfahren
  - Komplementaritätstest mit Unifikation höherer Stufe → Huet
    - · i.A. unentscheidbar (!) und erheblich komplizierteres Verfahren
- **Beweise sind normalerweise sehr kurz und elegant**
  - Aber erheblich schwerer zu finden

Beweis der Grundeigenschaften der Gleichheit

$$\doteq \;\equiv\; \lambda x.\lambda y.\forall P(P(x) \Rightarrow P(y))$$

- **Reflexivität:** $a \doteq a$

$$\begin{bmatrix} Pa^{T} & Pa^{F} \end{bmatrix}$$

- **Kommutativität:** $a \doteq b \Rightarrow b \doteq a$

$$\begin{bmatrix} Xa^{F} & Pb^{T} & Pa^{F} \\ Xb^{T} & & \end{bmatrix} \qquad [\lambda z.\neg Pz/X]$$

- **Transitivität:** $a \doteq b \wedge b \doteq c \Rightarrow a \doteq c$

$$\begin{bmatrix} Xa^{F} & Yb^{F} & Pa^{T} & Pc^{F} \\ Xb^{T} & Yc^{T} & & \end{bmatrix} \qquad [P/X, P/Y]$$

- **Substitutivität:** $Pa \wedge a \doteq b \Rightarrow Pb$

$$\begin{bmatrix} Pa^{T} & Xa^{F} & Pb^{F} \\ & Xb^{T} & \end{bmatrix} \qquad [P/X]$$

Intuitionistische Logik höherer Stufe

- **Unterstützt formale Manipulation von Algorithmen**
  - Synthese aus Spezifikationen, Optimierung, Verifikation, ...
  - Einheitliche Sprache für Spezifikation, Programmierung, Deduktion...
- **(Zur Zeit noch) viele verschiedene Formulierungen**
  - Martin-Löf'sche Typentheorie (Computational Type Theory)
  - Kalkül der Konstruktionen
  - System F
  - LCF (Logik berechenbarer Funktionen)

    ⋮
- **Beweissysteme interaktiv mit taktischer Steuerung**
  - **AUTOMATH** (historischer Vorläufer)
  - **Nuprl, MetaPRL** (Computational Type Theory)
  - **Alf / Agda, ...** (Martin-Löf Typentheorie)
  - **Lego**, **Coq** (Kalkül der Konstruktionen)
  - **Cambridge LCF**
- **Viele erfolgreiche Anwendungen**