

Inferenzmethoden

Teil IV

Jenseits von Prädikatenlogik



ES GIBT MEHR ALS KLASSISCHE PRÄDIKATENLOGIK

- **Konstruktive Logik: mehr als nur Wahrheit**
 - Interpretiere logische Symbole als **Konstruktion** eines Nachweises
 - Gut verwendbar als Logik der Berechnung und Programmierung
 - Ursprünglicher Name: **Intuitionistische Logik**
- **Modallogiken: zusätzliche Quantoren \diamond , \square**
 - Ist Gültigkeit einer Aussage möglich oder zwingend notwendig?
- **Logik höherer Stufe: freie Quantifizierung**
 - Formeln dürfen auch über Funktionen und Prädikate quantifizieren
- **... und noch vieles mehr**
 - Lineare, nichtmonotone, Relevanz-, Beschreibungs-, Temporallogik, ...
 - Kombinationen: konstruktive Logik höherer Stufe, Typentheorie, ...

Lassen sich Beweisverfahren entsprechend anpassen?

VORAUSSETZUNGEN FÜR NICHTKLASSISCHES BEWEISEN

- **Normalformen sind nicht immer möglich**
 - In konstruktiver Logik ist jede Formel in DNF ungültig
 - Eine Klausel müsste, für sich alleine betrachtet, gültig sein
 - In linearer Logik gibt es mehrere Konjunktionsbegriffe
- **Beweise müssen mehr Information enthalten**
 - In konstruktiver Logik enthalten Beweise algorithmische Lösungen
 - In Modallogiken entstehen gesicherte Aussagen aus Möglichkeiten
 - Lineare Logik beschreibt die Verarbeitung von Ressourcen
- **Konnektionsmethode muß erweitert werden**
 - Beweissuchverfahren für Nichtnormalform-Matrizen
 - Verwaltung logik-spezifischer Zusatzinformation in den Literalen und Verallgemeinerung des Komplementaritätsbegriffs
 - Komplementaritätstest mit erweiterten Unifikationsverfahren

Inferenzmethoden

Einheit 14

Die Konnektionsmethode:

Behandlung von Nicht-Normalform-Matrizen



1. Anpassung der Grundkonzepte
2. Verfahren für Nicht-Normalform-Matrizen
3. Pfadexploration auf Formelbäumen

DEDUKTION OHNE NORMALFORMBILDUNG

● Normalform-Matrizen sind zu einfach

- Matrix \equiv Menge von Klauseln in α -Beziehung
- Klausel \equiv Menge von Literalen in β -Beziehung
- Im Formelbaum müssten alle α -Knoten vor den β -Knoten erscheinen
- Nur wenige Formeln werden von Normalform-Matrizen repräsentiert

● Normalisierung ist “unnatürlich”

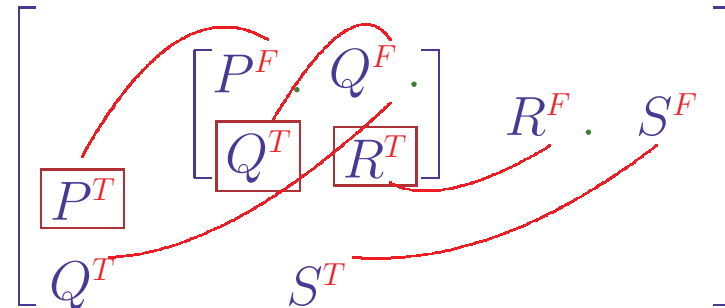
- Oft exponentielle Aufblähung der Formel ↪ Effizienzprobleme
- Originalformel selten rekonstruierbar ↪ Unverständliche Beweise
- Normalformtransformationen jenseits von Prädikatenlogik kaum möglich

● Erweitere Extensionsverfahren auf Formelbäume

- Verwende allgemeine Konzepte anstelle der vereinfachten Klauselform
- Matrizen sind komplexer (Mengen von Matrizen kleinerer Tiefe)
- Pfadbegriff ist feiner (Menge von Literalen in α -Beziehung)
- Aufwendigere Pfadüberprüfung – sonst keine grundsätzliche Änderung

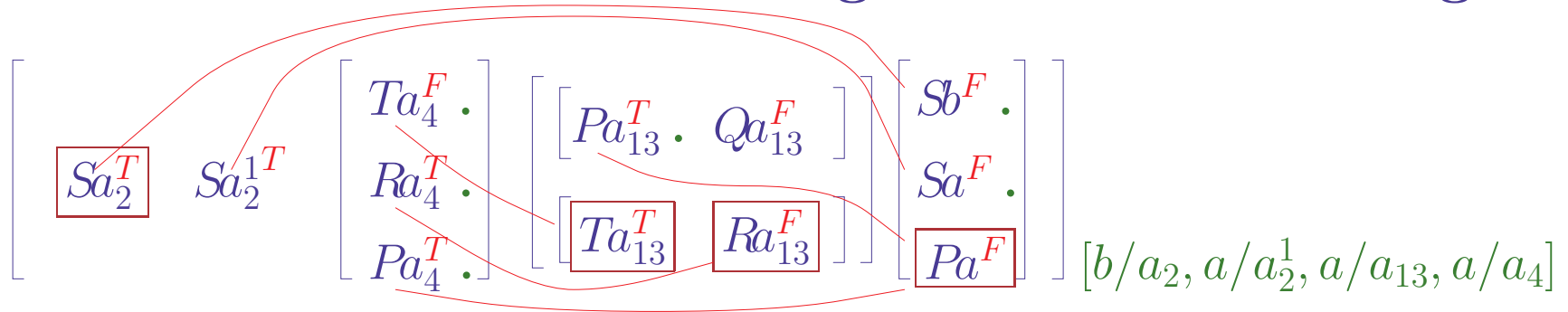
NICHT-NORMALFORM – WAS MUSS ANGEPASST WERDEN?

● Wichtige Änderungen in der Aussagenlogik

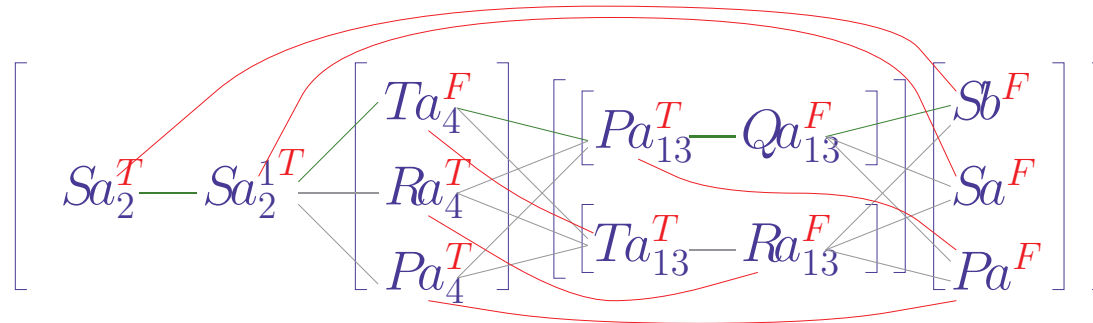


- Welche Literale gehören zum aktuellen Pfad?
- Was ist die “aktuelle Klausel”?
- Extension: Welcher Teil der Matrix kann noch konnektiert werden?
- Bereinigung: Wann ist eine “Klausel” abgeschlossen

● Keine zusätzlichen Änderungen für Prädikatenlogik



GRUNDKONZEPTE DES MATRIXKALKÜLS WIEDERHOLT



- **α/β -Beziehung** zwischen Literalen

- $u \sim_{\alpha} v$: $u \neq v$ und größter gemeinsamer Vorfahr im Formelbaum hat Typ α
- $u \sim_{\beta} v$: $u \neq v$ und größter gemeinsamer Vorfahr im Formelbaum hat Typ β

- **Matrix (der Tiefe n)**

- Literal oder Menge von Matrizen der maximalen Tiefe n-1
- Submatrizen stehen in α - bzw. β -Beziehung (gerade/ungerade Tiefe)
- Präsentation: α -Beziehungen nebeneinander, β -Beziehungen übereinander

- **Pfad**

- (Maximale) Menge von Literalen in gegenseitiger α -Beziehung
- Implementierung verwendet induktive Definition auf Formelbaum

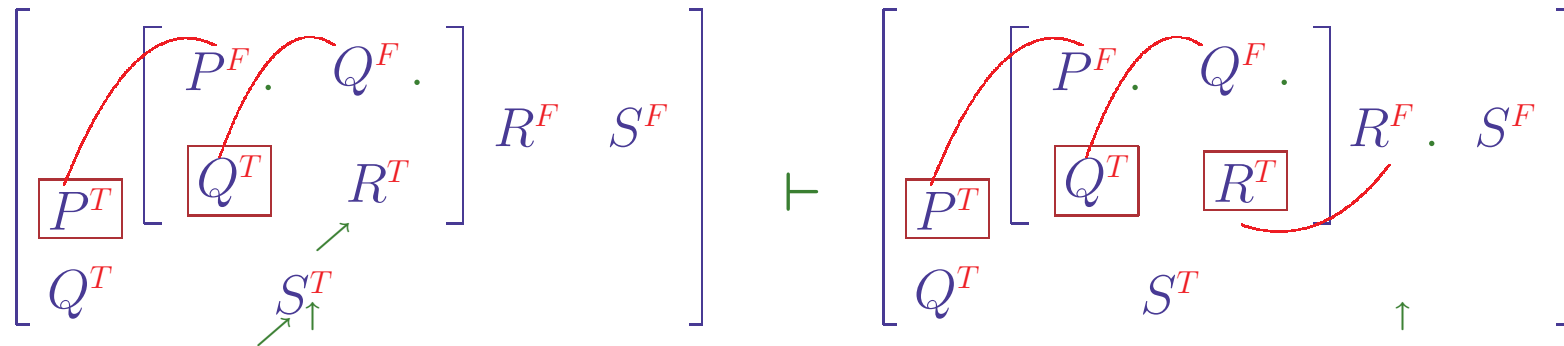
- **σ -komplementäre Konnektion**

- Paar $\{X_1^T, X_2^F\}$ von Literalen, deren Formeln unter σ gleich sind

ALLGEMEINES EXTENSIONSVERFAHREN: GRUNDKONZEPTE

- **Aktueller (aktiver) Pfad \mathcal{P}**
 - Nichtkomplementäre Menge von Literalen in gegenseitiger α -Beziehung
- **Offene Teilmatrix $\mathcal{M}_{\mathcal{P}}$** $\hat{=}$ ungenutzter Teil der Matrix
 - Menge von Literalen, die zum aktuellen Pfad \mathcal{P} in α -Beziehung stehen
- **Teilziel** $\hat{=}$ (Teil-)Klausel
 - Menge von Literalen in gegenseitiger β -Beziehung
 - $C_{\beta}(L, \mathcal{P})$: maximales Teilziel in $\mathcal{M}_{\mathcal{P}}$, das L enthält $\hat{=}$ Klausel von L
- **Aktuelles (aktives) Teilziel \mathcal{C} (zu \mathcal{P})** $\hat{=}$ abgeschlossene Literale der aktuellen Klausel
 - Teilziel, das ausschließlich aus Literalen der offenen Teilmatrix besteht
- **Aktives Ziel $(\mathcal{P}, \mathcal{C})$**
 - Aktueller Pfad \mathcal{P} und passendes aktuelles Teilziel \mathcal{C}
- **Offenes Ziel \mathcal{E} (zu $(\mathcal{P}, \mathcal{C})$)** $\hat{=}$ offene Literale der aktuellen Klausel
 - Menge der Literale der offenen Teilmatrix, die zu \mathcal{C} in β -Beziehung stehen
 - Eindeutig durch die Matrix und das aktive Ziel $(\mathcal{P}, \mathcal{C})$ bestimmt

EXTENSIONSSCHRITT AUF NICHT-NORMALFORM-MATRIZEN



↑ markiert **aktuelle "Klausel"**

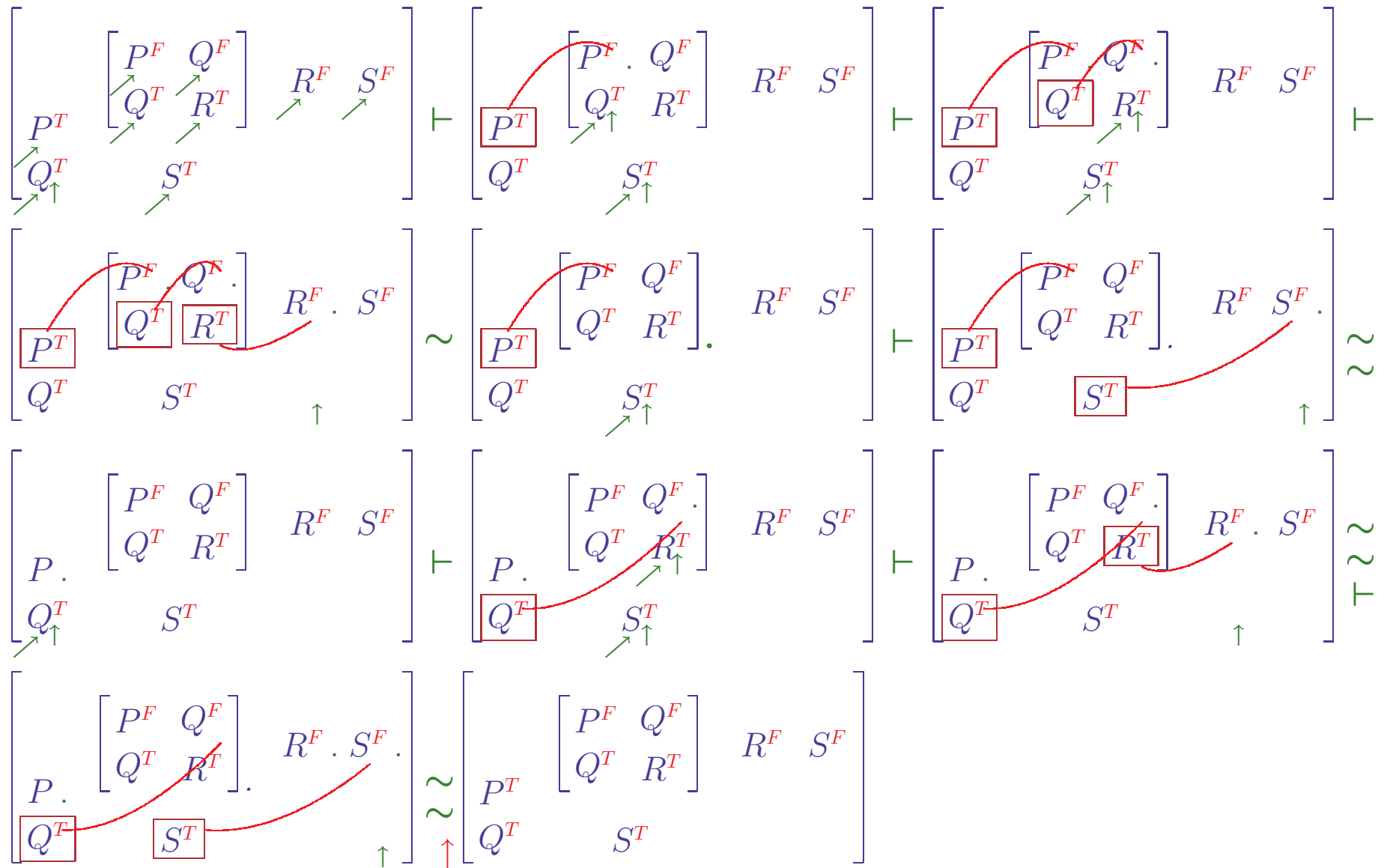
\boxed{P} markiert Literale des **aktuellen Pfades \mathcal{P}**

↗ markiert Literale des offenen Ziels \mathcal{E}

· markiert abgeschlossene Teilpfade

1. Wähle ein Literal L des zu $(\mathcal{P}, \mathcal{C})$ offenen Ziels \mathcal{E} markiert mit ↗
2. Erweitere den aktuellen Pfad \mathcal{P} um L markiere mit Box \boxed{L}
3. Wähle ein mit L konnektiertes Literal \bar{L} der offenen Teilmatrix
Vermerke Alternativen in **Alternativenmenge**
4. Wähle Teilmenge \mathcal{C} der zu \bar{L} in β -Beziehung stehenden Literale, die mit dem aktuellen Pfad \mathcal{P} konnektiert sind, und eine Substitution ρ , welche die mit σ modifizierten Konnektionen komplementär macht markiere mit ·
 - Erweitere σ mit ρ ; vermerke alternative Teilmengen und Substitutionen
 - Breche den Extensionsschritt ab, falls es keine solche Teilmenge gibt

EXTENSIONSBEWeis AUF NICHT-NORMALFORM MATRIZEN



EIN UNIFORMES VERFAHREN FÜR FORMELBÄUME

- **CP₁-ähnliches Beweisverfahren ist kompliziert**

- Trickreiche Erweiterung des klauselbasierten Verfahrens → Bibel 1987
- Schwer als korrekt und vollständig zu beweisen

- **2-D Matrizen sind nur eine Illustration**

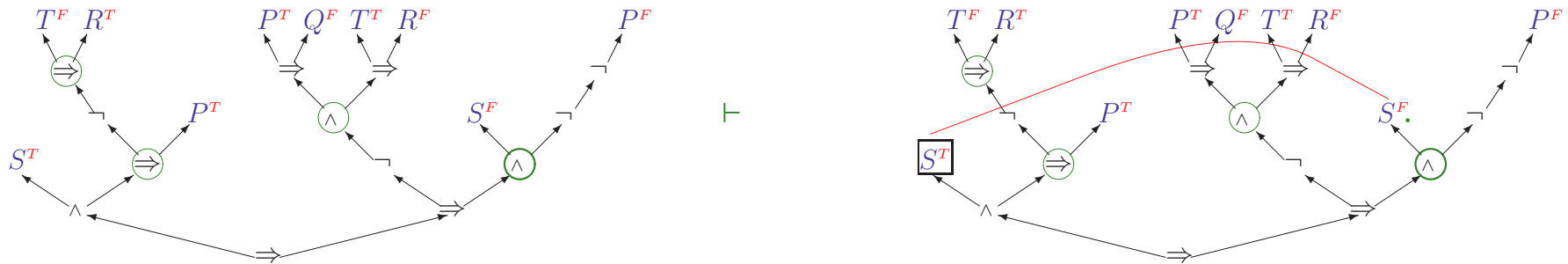
- Markierungen \uparrow , \nearrow , \boxed{L} , \cdot sind nur optische Hilfsmittel
- Implementiertes Verfahren verarbeitet Verwaltungsvariablen für aktuelle Pfade, offene Ziele, offene Teilmatrix, Konnektionen, etc

- **Beschreibe Prozedur mit Formelbaumkonzepten**

- Herleitung des Verfahrens direkt aus dem Charakterisierungstheorem
- Korrektheit und Vollständigkeit “leicht” zu beweisen
- Methodik auch jenseits von Prädikatenlogik erster Stufe anwendbar

Details in Fachaufsätzen auf der Veranstaltungswebseite

REFORMULIERUNG DES EXTENSIONSSCHRITTS



$$\mathcal{P} = \{ \}$$

$$\mathcal{C} = \{ \}$$

$$\mathcal{E} = \{S^T, T^F, R^T, P_1^T, P_2^T, Q^F, T^T, R^F, S^F, P^F\}$$

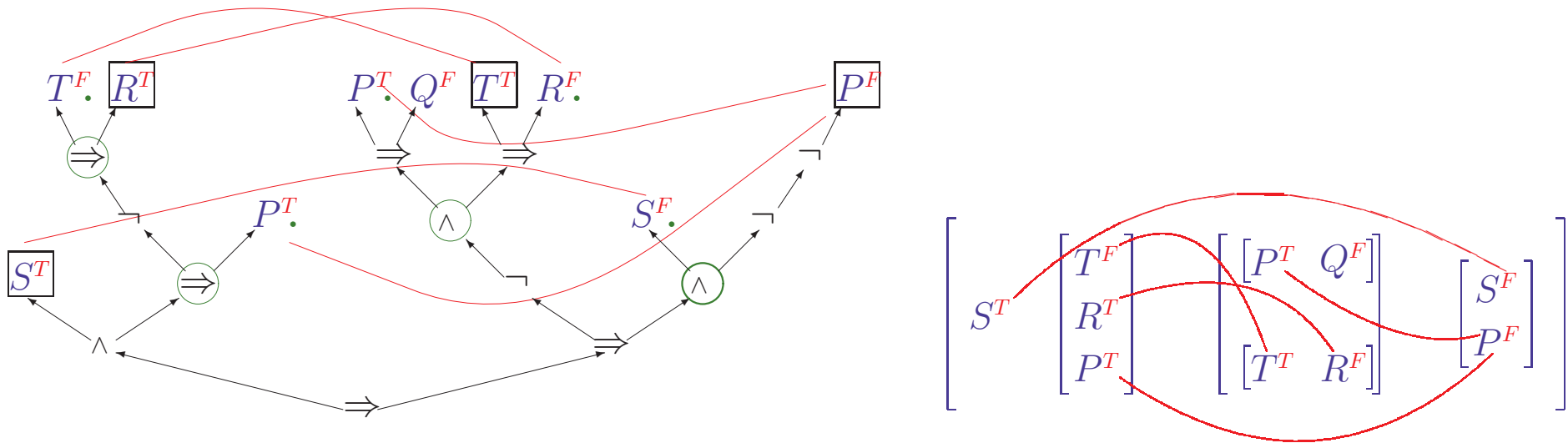
$$L = S^T, \quad \mathcal{P} = \{S^T\}, \quad \mathcal{M}_{\mathcal{P}} = \{T^F, R^T, P_1^T, P_2^T, Q^F, T^T, R^F, S^F, P^F\}$$

$$\bar{L} = S^F, \quad C_{\beta}(\bar{L}) = \{S^F, P^F\}, \quad \mathcal{C} = \{S^F\}$$

$$\mathcal{E} = \{P^F\}$$

1. Wähle ein Literal L des zu $(\mathcal{P}, \mathcal{C})$ offenen Ziels \mathcal{E}
2. Erweitere den aktuellen Pfad \mathcal{P} um L
3. Wähle ein mit L konnektiertes Literal \bar{L} der offenen Teilmatrix
Vermerke Alternativen in **Alternativenmenge**
4. Wähle Teilmenge \mathcal{C} der Literale von $C_{\beta}(\bar{L}, \mathcal{P})$, die mit dem aktuellen Pfad \mathcal{P} konnektiert sind, und eine Substitution ρ , welche die mit σ modifizierten Konnektionen komplementär macht
 - Erweitere σ mit ρ und vermerke alternative Teilmengen und Substitutionen
 - Breche den Extensionsschritt ab, falls es keine solche Teilmenge gibt

EXTENSIONSBEWeis AUF FORMELBÄUMEN AM BEISPIEL



Start	$\mathcal{P} = \{\}$ $\mathcal{C} = \{\}$ $\mathcal{E} = \{S^T, T^F, R^T, P_1^T, P_2^T, Q^F, T^T, R^F, S^F, P^F\}$
Erster Schritt	$L = S^T$, $\mathcal{P} = \{S^T\}$, $\mathcal{M}_{\mathcal{P}} = \{T^F, R^T, P_1^T, P_2^T, Q^F, T^T, R^F, S^F, P^F\}$ $\bar{L} = S^F$, $C_{\beta}(\bar{L}) = \{S^F, P^F\}$, $\mathcal{C} = \{S^F\}$, $\mathcal{E} = \{P^F\}$
Zweiter Schritt	$L = P^F$, $\mathcal{P} = \{S^T, P^F\}$, $\mathcal{M}_{\mathcal{P}} = \{T^F, R^T, P_1^T, P_2^T, Q^F, T^T, R^F\}$ $\bar{L} = P_2^T$, $C_{\beta}(\bar{L}) = \{P_2^T, T^T, R^F\}$, $\mathcal{C} = \{P_2^T\}$, $\mathcal{E} = \{T^T, R^F\}$
Dritter Schritt	$L = T^T$, $\mathcal{P} = \{S^T, P^F, T^T\}$, $\mathcal{M}_{\mathcal{P}} = \{T^F, R^T, P_1^T, R^F\}$ $\bar{L} = T^F$, $C_{\beta}(\bar{L}) = \{T^F, R^T, P^T\}$, $\mathcal{C} = \{T^F, P^T\}$, $\mathcal{E} = \{R^T\}$
Vierter Schritt	$L = R^T$, $\mathcal{P} = \{S^T, P^F, T^T, R^T\}$, $\mathcal{M}_{\mathcal{P}} = \{R^F\}$ $\bar{L} = R^F$, $C_{\beta}(\bar{L}) = \{R^F\}$, $\mathcal{C} = \{R^F\}$, $\mathcal{E} = \{\}$

BEWEISBARKEIT ALGORITHMISCH

- **Beweisbarkeit von $(\mathcal{P}, \mathcal{C})$** (bezüglich F)
 - Alle Pfade, die $\mathcal{P} \cup \{v\}$ für ein Literal $v \in \mathcal{E}$ erweitern, sind komplementär unter einer Substitution σ
 - Entspricht Gültigkeit der verbleibenden Teilmatrix
- **Satz: F gültig g.d.w. aktives Ziel (\emptyset, \emptyset) beweisbar**
... für eine Multiplizität μ und eine zulässige Substitution σ
Beweis des Satzes stützt sich auf das Charakterisierungstheorem
- **Satz: Ein aktives Ziel $(\mathcal{P}, \mathcal{C})$ ist beweisbar g.d.w.**
 - (1) Das offene Ziel \mathcal{E} ist leer, oder
 - (2) Für ein $L \in \mathcal{E}$ ist $(\mathcal{P}, \mathcal{C} \cup \{L\})$ beweisbar und es gibt eine komplementäre Konnektion $\{L, \bar{L}\}$ mit $\bar{L} \in \mathcal{P}$, oder
$$\bar{L} \sim_{\alpha} \mathcal{P} \cup \{L\} \text{ und } (\mathcal{P} \cup \{L\}, \{\bar{L}\}) \text{ beweisbar}$$Grundlage für Beschreibung des Extensionsverfahrens auf Formelbäumen

DAS UNIFORME PFADSUCHVERFAHREN (FUNKTIONAL)

- $\text{prove}(F, n)$

$$= \begin{cases} \text{provable}(\emptyset, \emptyset, []) & \text{falls dies erfolgreich ist} \\ \text{wobei } \mu, \text{CON} = (n, \text{connections}(F^\mu)) & \\ \text{prove}(F, n+1) & \text{sonst} \end{cases}$$
- $\text{provable}(\mathcal{P}, \mathcal{C}, \sigma)$

$$= \begin{cases} \text{check-extension}(\mathcal{E}, \sigma) & \text{falls } \mathcal{E} \neq \emptyset \\ \text{wobei } \mathcal{E} = \{v \in F \mid v \sim_\alpha \mathcal{P} \wedge v \sim_\beta \mathcal{C}\} & \\ \sigma & \text{sonst} \end{cases}$$
- $\text{check-extension}(\mathcal{E}, \sigma)$

$$= \begin{cases} \text{check-connections}(\mathcal{D}, A, \sigma) & \text{falls dies erfolgreich ist} \\ \text{wobei } A \in \mathcal{E} \text{ beliebig, und} & \\ \mathcal{D} = \{\bar{A} \in F \mid \{A, \bar{A}\} \in \text{CON} \wedge (\bar{A} \in \mathcal{P} \vee \bar{A} \sim_\alpha (\mathcal{P} \cup \{A\}))\} & \\ \text{check-extension}(\{v \in \mathcal{E} \mid v \sim_\alpha A\}, \sigma) & \text{sonst (und } \mathcal{E} \neq \emptyset) \end{cases}$$
- $\text{check-connections}(\mathcal{D}, A, \sigma)$

$$= \begin{cases} \text{provable}(\mathcal{P}, \mathcal{C} \cup \{A\}, \sigma_2) & \text{falls dies erfolgreich ist} \\ \text{wobei } \bar{A} \in \mathcal{D} \text{ beliebig, } \sigma_1 = \text{unify-check}(A, \bar{A}, F^\mu, \sigma) & \\ \text{und } \sigma_2 = \begin{cases} \sigma_1 & \text{falls } \bar{A} \in \mathcal{P} \\ \text{provable}(\mathcal{P} \cup \{A\}, \{\bar{A}\}, \sigma_1) & \text{sonst} \end{cases} & \\ \text{check-connections}(\mathcal{D} - \{\bar{A}\}, A, \sigma) & \text{sonst (und } \mathcal{D} \neq \emptyset) \end{cases}$$

KORREKTHEIT & VOLLSTÄNDIGKEIT

Eine Formel F ist genau dann prädikatenlogisch gültig, wenn $\text{prove}(F, 1)$ mit einer geeigneten Unifikations- und Zulässigkeitsprüfung `unify-check` erfolgreich terminiert

- $\text{prove}(F, 1)$ beweist F oder terminiert nicht
 - Ist F gültig, so liefert $\text{prove}(F, 1)$ eine zulässige Substitution, die alle Pfade komplementär macht
 - Ist F nicht gültig, so wird die Multiplizität unendlich oft erhöht
- **Korrektheitsbeweis durch simultane Induktion**
 - Abgestützt auf die Sätze über Beweisbarkeit aktiver Ziele
- **Implementierung in funktionaler/logischer Sprache**
 - Rekursive Verwaltung der Alternativen durch den Compiler
 - Imperative Implementierung erfordert höheren Verwaltungsaufwand und ist fehleranfälliger