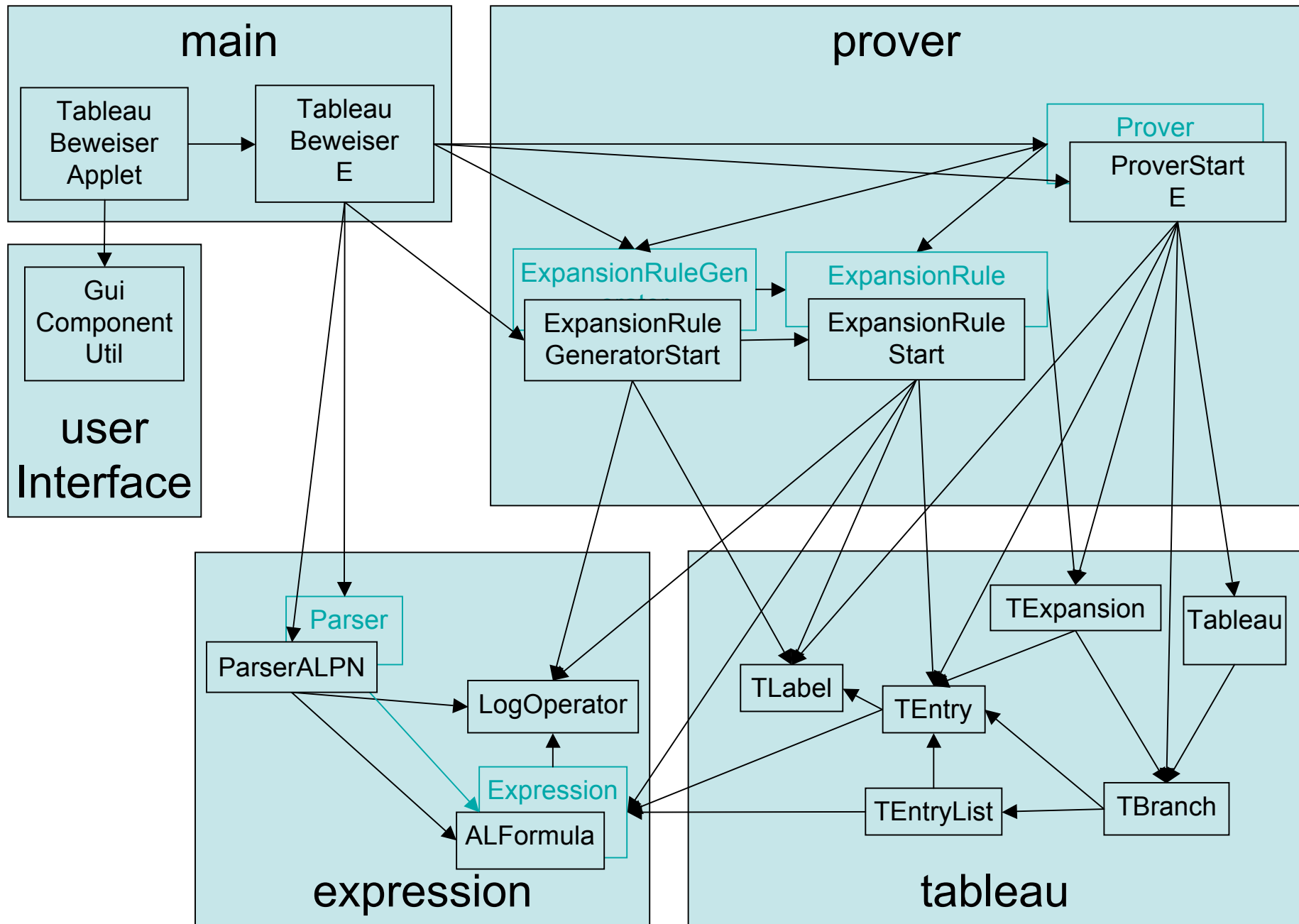
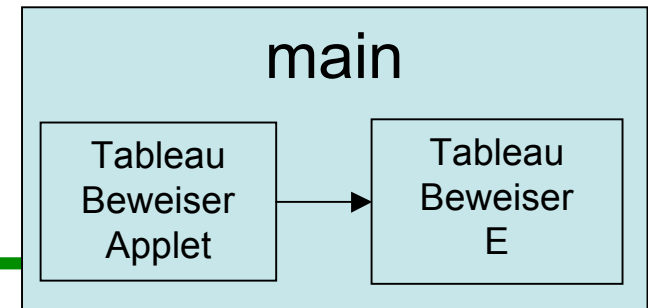

Reasoning Services

Carola Eschenbach, Özgür Özçep
Universität Hamburg, Dept. Informatik
AB Wissens- und Sprachverarbeitung (WSV)

Wintersemester 2006



package main

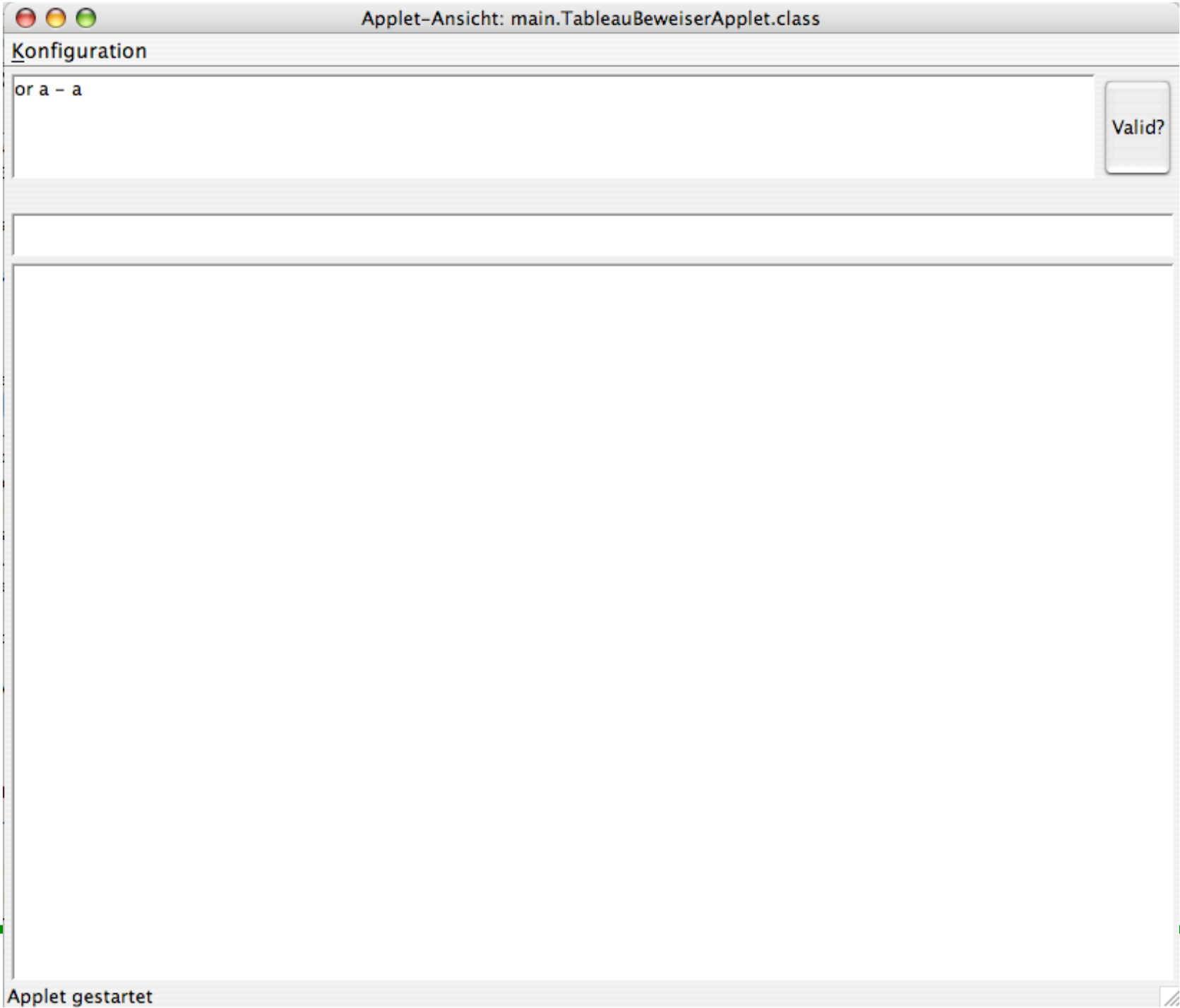


Zwei Klassen

- von denen es im Normalfall nur eine Instanz gibt

TableauBeweiserApplet

- erweitert die Klasse JApplet
- ist für die Gestaltung des (Haupt-)Fensters verantwortlich
- erzeugt die Komponenten des Fensters
- erzeugt eine Instanz von TableauBeweiser und übergibt die Felder zur Füllung durch andere Klassen



package userInterface

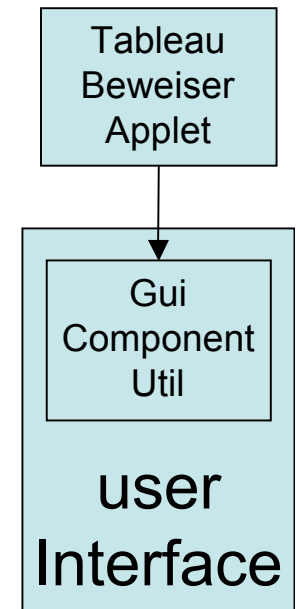
Derzeit nur eine Klasse

- von der es im Normalfall genau eine Instanz gibt

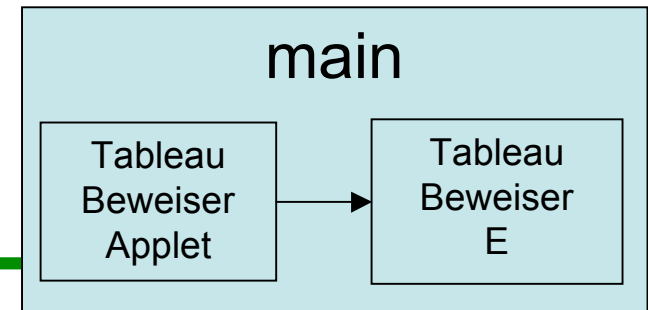
GuiComponentUtil

- Hilfsmodul von TableauBeweiserApplet
- enthält Information zum Layout des (Haupt-)Fensters
- Methoden zur Gestaltung der Felder und des Buttons
- Methode zur Generierung des Menus

- definiert die Aktionen, auf die main.TableauBeweiser reagieren muss.



package main



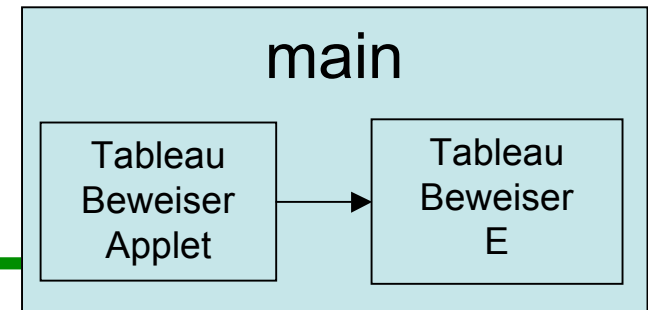
Zwei Klassen

- von denen es im Normalfall nur eine Instanz gibt

TableauBeweiser

- implementiert Interface ActionListener
- ist für die Verarbeitung der Benutzereingaben (über die Fensterkomponenten) verantwortlich
- erzeugt und verwaltet
 - Parser,
 - Regelgenerator und
 - den eigentlichen TableauBeweiser
- (Es sind Mechanismen vorgesehen, mit denen auch alternative Parser, Regelgeneratoren und Tableau-Beweiser einbindbar sind.)

package main

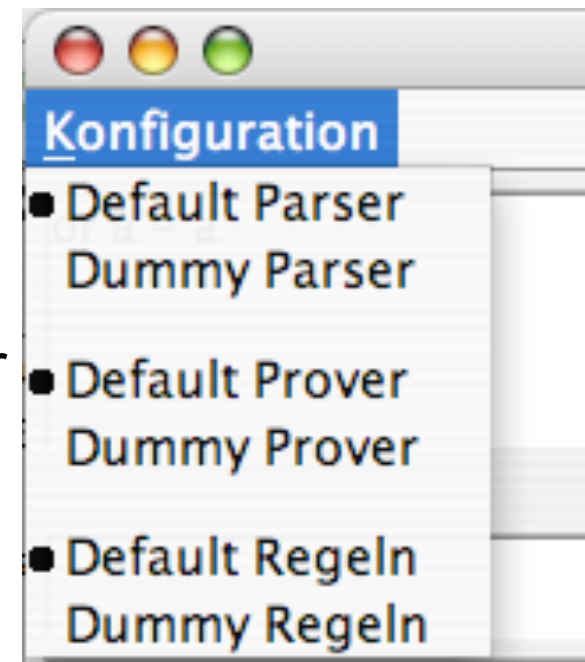


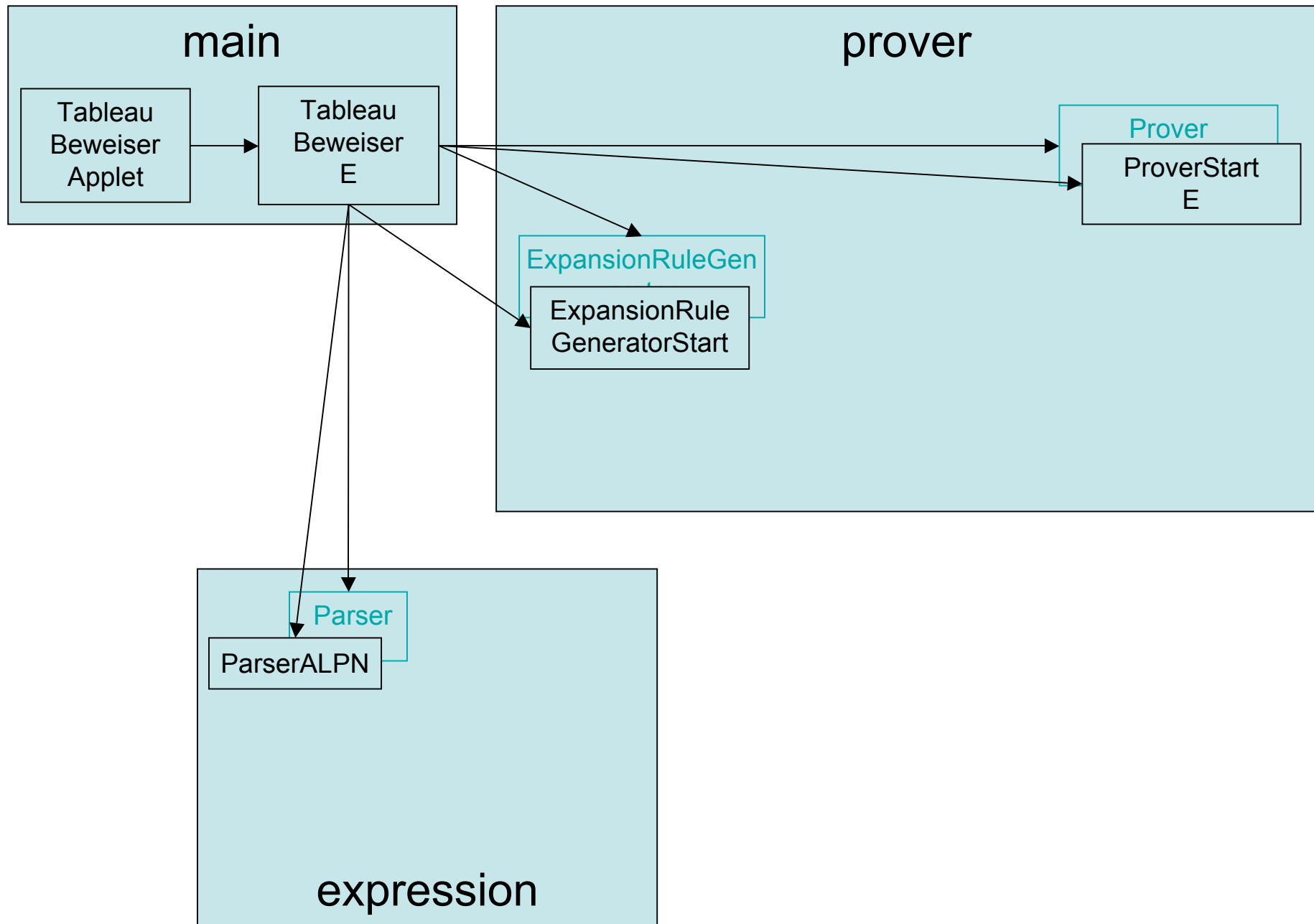
Zwei Klassen

-

TableauBeweiser

- erzeugt und verwaltet
 - Parser,
 - Regelgenerator und
 - den eigentlichen TableauBeweiser
- (Es sind Mechanismen vorgesehen, mit denen auch alternative Parser, Regelgeneratoren und Tableau-Beweiser einbindbar sind.)





package expression

Zwei Interfaces, 4 Klassen

- nicht dargestellt:
Klasse ExpressionException

LogOperator

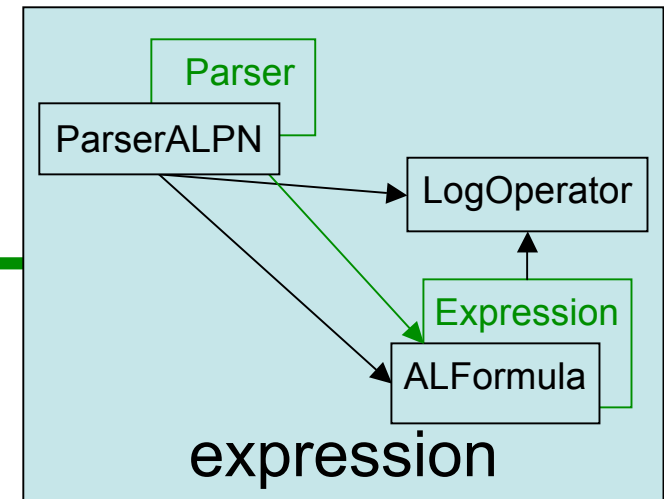
- enum-Klasse, Aufzählung der logischen Operatoren

Expression

- Interface für eine Datenstruktur der zu verarbeitenden Ausdrücke

ALFormula

- implementiert Expression
- Datenstruktur für aussagenlogische Formeln



package expression

Zwei Interfaces, 4 Klassen

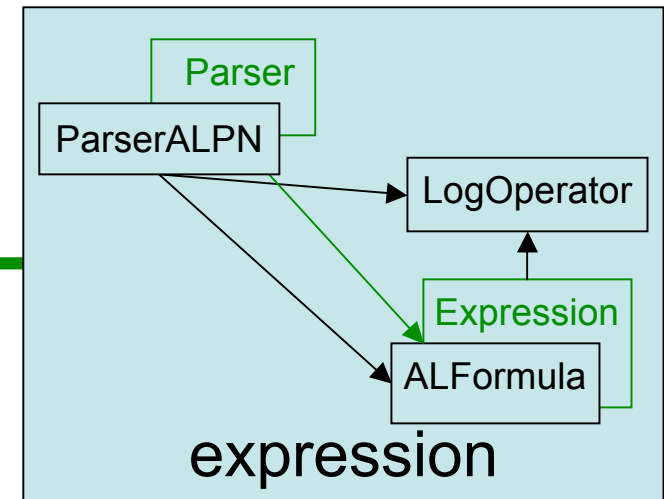
- nicht dargestellt:
Klasse ExpressionException

Parser <E extends Expression>

- Interface zum Parsen von Zeichenketten

ParserALPN

- implementiert Parser <ALFormula>
- Parser für aussagenlogische Formeln in Präfixnotation



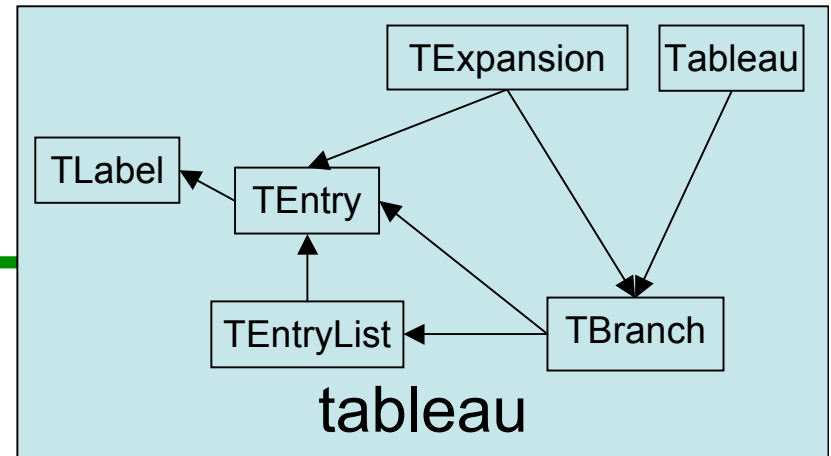
package tableau

7 Klassen

- nicht dargestellt:
Klasse TableauException

Vorbemerkung

- Die Klassen sind weitgehend passiv
- und 'wissen' nichts von Tableau-Expansion oder Strategien
- 'reine Verwaltungs-Klassen'
- Die Verarbeitung erfolgt durch Klassen des Packets prover.



package tableau

7 Klassen

- nicht dargestellt:
Klasse TableauException

TLabel

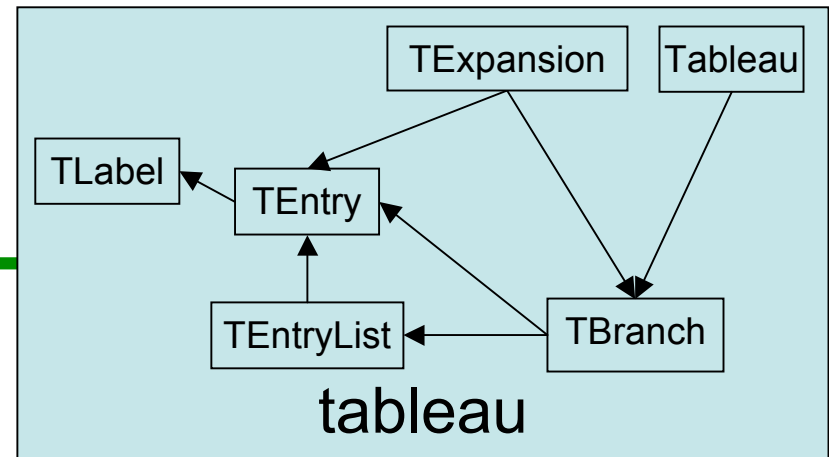
- enum-Klasse für die möglichen Markierungen

TEntry

- Klasse für Tableau-Einträge
- TLabel + Expression (Inhalt)
- werden nach der Erzeugung nie verändert

TEntryList

- Sammlung von TEntries, als TreeMap implementiert, damit Einträge gleichen Inhalts leicht gefunden werden können



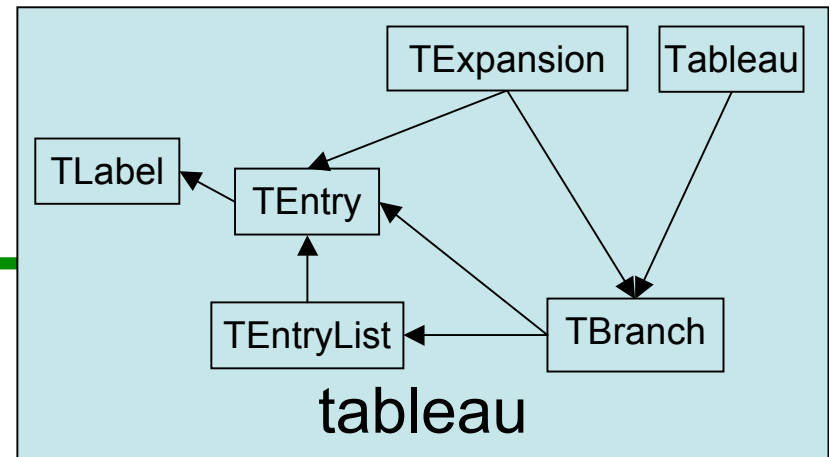
package tableau

7 Klassen

- nicht dargestellt:
Klasse TableauException

TBranch

- Klasse für Tableau-Zweige
- unterschiedliche Listen für Einträge mit atomaren Formeln, bereits expandierte Einträge, noch expandierbare Einträge
- erlaubt Ergänzung von Einträgen und 'Markierung' als expandiert
- Abschluss wird beim Einfügen atomarer Einträge geprüft und durch Statusvariable markiert
- keine doppelten Einträge



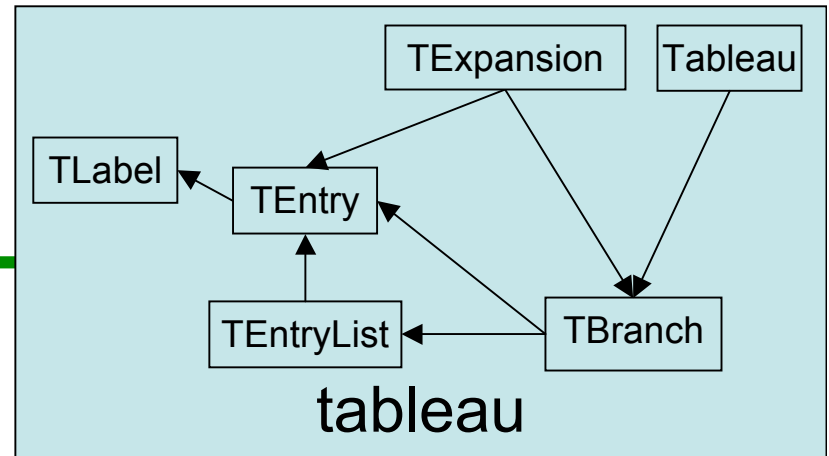
package tableau

7 Klassen

- nicht dargestellt:
Klasse TableauException

Tableau

- Klasse für Tableaux
- interne Unterscheidung von aktuell bearbeitetem Zweig (currentBranch) und den restlichen Zweigen
- Tableau-Expansion erfolgt durch Ersetzung des aktuellen Zweigs durch eine Liste von Zweigen
- Wenn der aktuelle Zweig geschlossen ist, wird er gelöscht und durch einen anderen ersetzt (so es noch einen gibt)
- Ist der letzte Zweig geschlossen, dann auch das Tableau



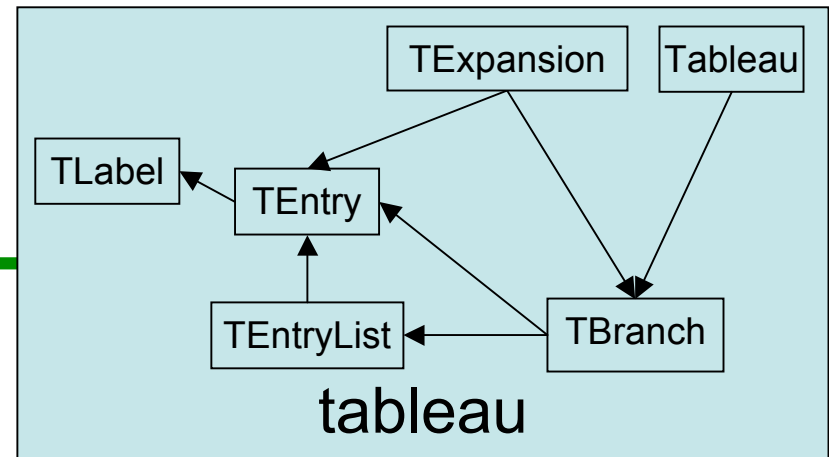
package tableau

7 Klassen

- nicht dargestellt:
Klasse TableauException

TExpansion

- Klasse zur Repräsentation einzelner Expansionsschritte
- enthält ursprünglichen T-Eintrag und Listen von Listen von Ergänzungen zum Tableau-Zweig
- Methode applyTo(TBranch) liefert Array von TBranch



package prover

3 Interfaces, 4 Klassen

- nicht dargestellt:
Klasse ProverException

ExpansionRule

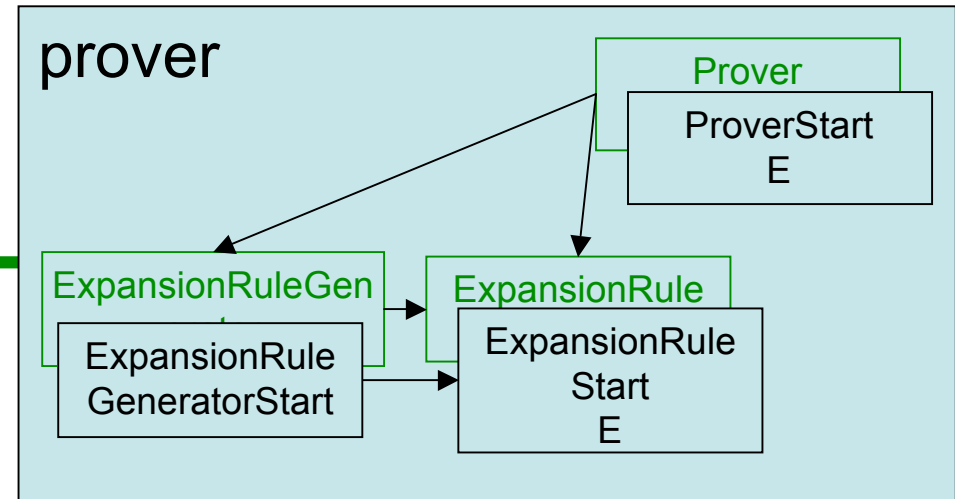
ExpansionRuleStart

ExpansionRuleGenerator

ExpansionRuleGeneratorStart

Prover

ProverStart



package prover

3 Interfaces, 4 Klassen

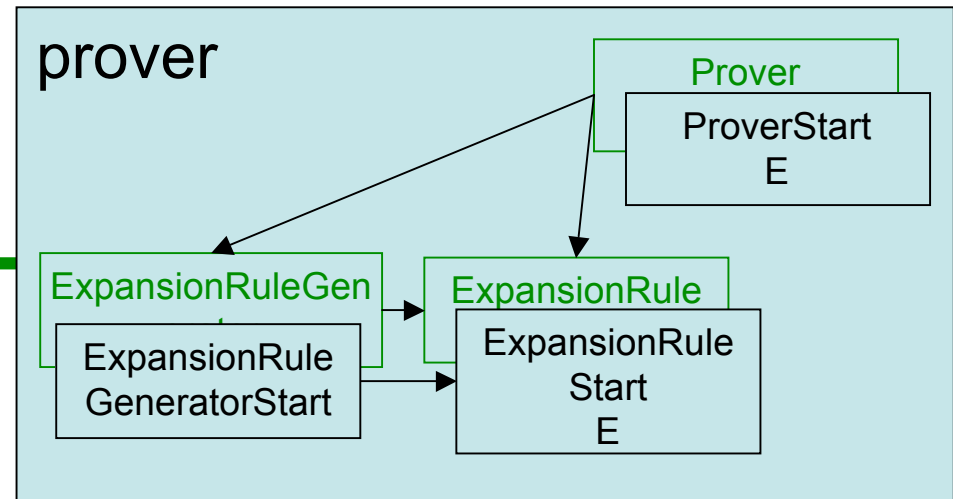
- nicht dargestellt: ProverException

ExpansionRule

- Interface für Implementationen von Expansionsregeln

ExpansionRuleStart

- implementiert ExpansionRule
- spezifiziert durch: TLabel, LogOperator, 2-dimensionales Array von TLabel
 - verschiedene Zweige
 - Label der Teilformel im jeweiligen Zweig
- liefert nur die erste gefundene Expansion



ExpansionRuleStart

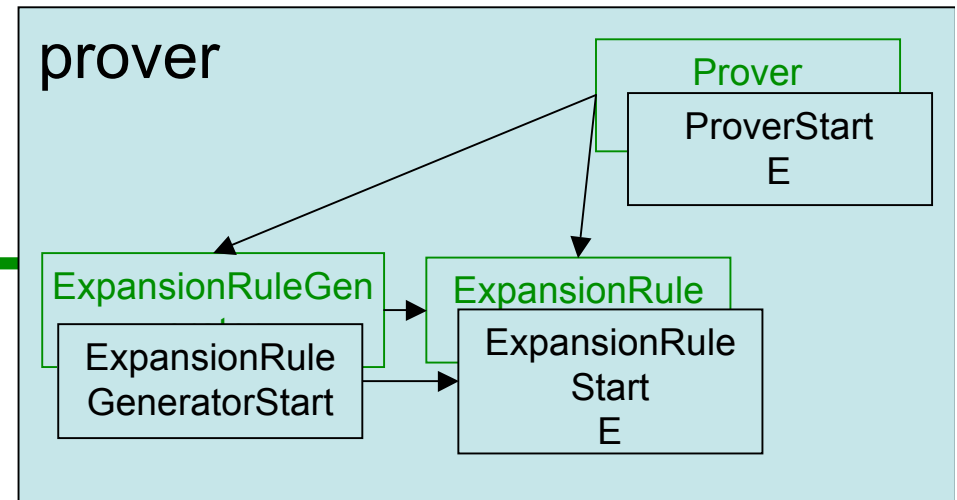
```
//      T X v Y
//      -----
//      T X | T Y
new ExpansionRuleStart(
    TLabel.True,
    LogOperator.Disjunktion,
    new TLabel[][]{ {TLabel.True, null},
                    {null, TLabel.True}});

//      F X v Y
//      -----
//      F X
//      F Y
new ExpansionRuleStart(
    TLabel.False,
    LogOperator.Disjunktion,
    new TLabel[][]{ {TLabel.False, TLabel.False}});
```

package prover

3 Interfaces, 4 Klassen

- nicht dargestellt:
Klasse ProverException



ExpansionRuleGenerator <R extends ExpansionRule>

- Interface für Generierung von Expansionsregeln

ExpansionRuleGeneratorStart

- implementiert ExpansionRuleGenerator
<ExpansionRuleStart>
- Regeln für Konjunktion, Disjunktion, Negation in
zweiwertigen Logiken

package prover

3 Interfaces, 4 Klassen

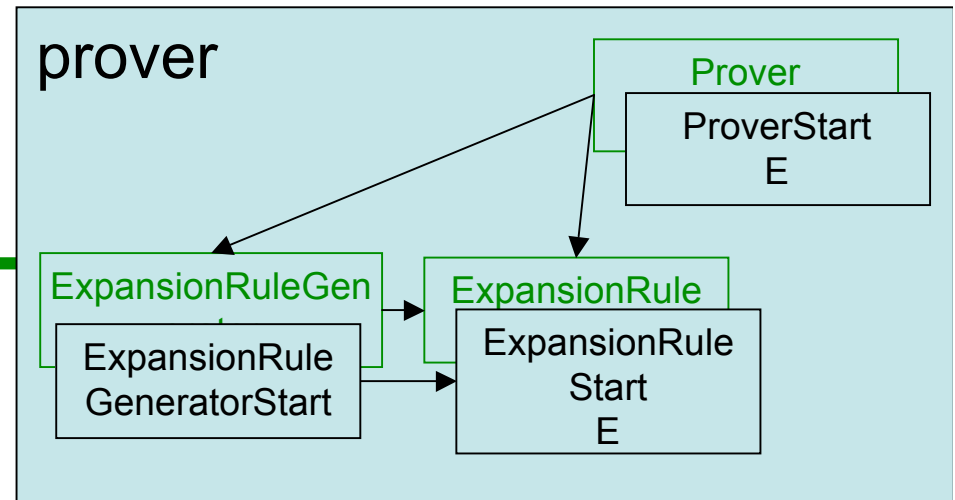
- nicht dargestellt:
Klasse ProverException

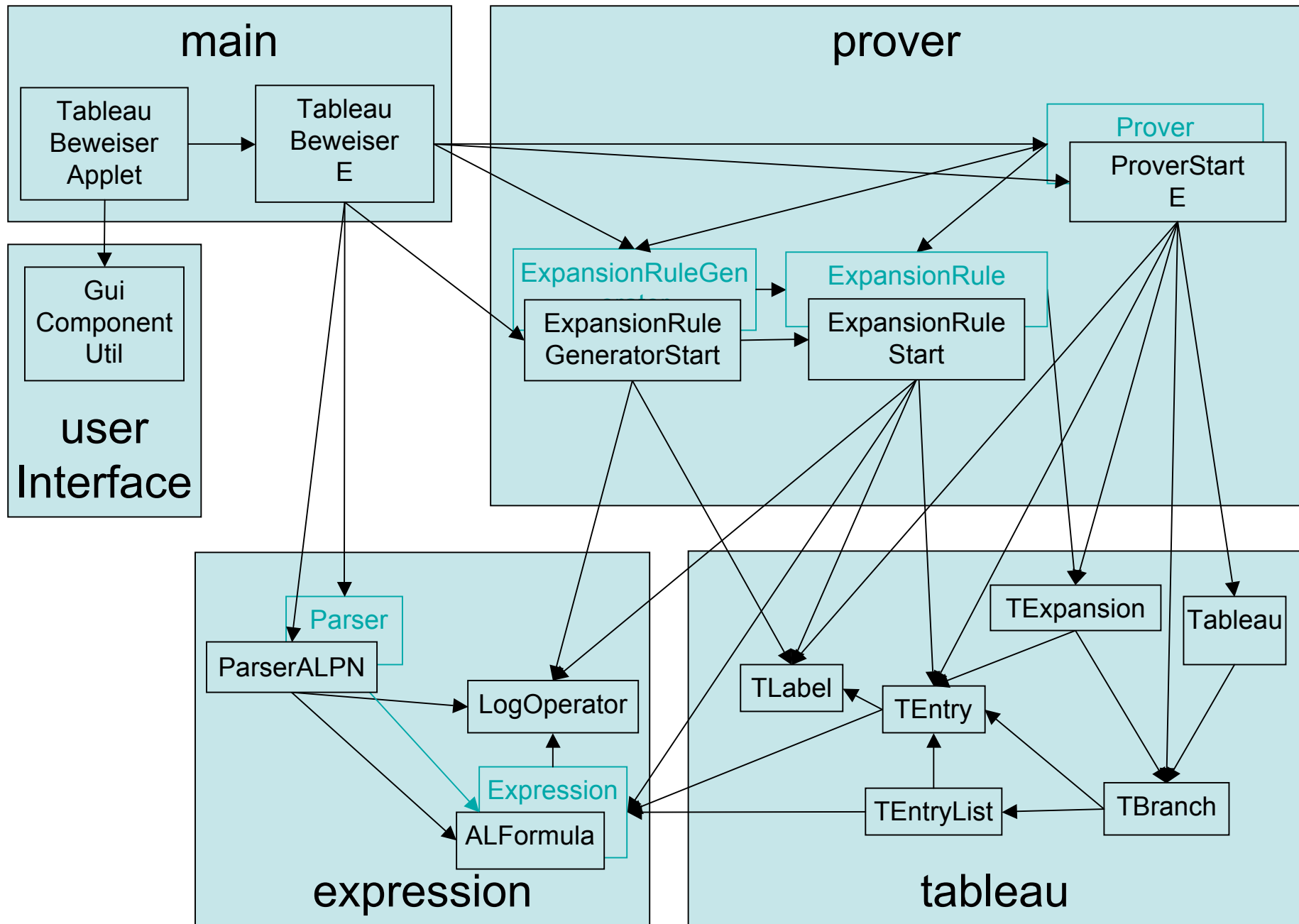
Prover

- Interface für Beweiser mit austauschbarem Regelgenerator

ProverStart

- implementiert Prover
- Expandiert das Tableau indem es den aktuellen Zweig expandiert und durch die generierten Zweige ersetzt.
- Verwendet nicht-verzweigende Regeln vor verzweigenden.
- Terminiert, wenn Tableau geschlossen oder aktueller Zweig nicht mehr expandierbar.





Aufgabe: Vertrautmachen mit den Klassen und Packeten

Führt alle erforderlichen Ergänzungen aus, damit

- folgende weitere Junktoren verarbeitet werden:
 - im: Implikation
 - bim: Biimplikation
 - xor: Exklusives Oder
- Macht Testläufe mit einschlägigen Tautologien
 - im a a ; eq a a ; - xor a a ; eq eq a b and im a b im b a ;
eq eq a b - xor b a ...
- Macht Testläufe mit falsifizierbaren Formeln
 - im a - a ; eq a b ; xor a a ...

Aufgabe: Vertrautmachen mit den Klassen und Packeten

Verändert die Ausgabe des Beweisers so,

- dass zum Schluss angegeben wird, wieviele Expansionsschritte ausgeführt wurden, wieviele Zweige insgesamt erzeugt wurden und wie lang der längste Zweig war, bevor er abgeschlossen wurde,
- Bestimmt den Aufwand der Expansion von
 - or or or or and a b and a c and - a d and - a - d and - b - c
- dass auch die im aktuellen Zweig bereits expandierten Formeln mit ausgegeben werden
- dass zum Schluss das gesamte erzeugte Tableau mit allen abgeschlossenen Zweigen ausgegeben wird
- Testen nicht vergessen

Aufgabe: Vertrautmachen mit den Klassen und Packeten

Schreibt einen neuen Regelgenerator

- der das sog. Semantic Branching realisiert

- z.B.
$$\frac{TP \vee Q}{\begin{array}{c|c} TP & FP \\ & TQ \end{array}}$$

- Erlaubt die Auswahl durch den Benutzer

Bestimmt den Aufwand der Expansion von

- or or or or and a b and a c and - a d and - a - d and - b - c
- durch die beiden alternativen Regelmengen

Klärt, warum diese Regel auch korrekt ist und zu weniger aufwendigen Ableitungen führt

Aufgabe: Vertrautmachen mit den Klassen und Packeten

Verändern Sie die Expansionsstrategie

- so, dass in einem Expansionsschritt alle nicht-verzweigenden Expansionen durchgeführt werden
- und auch die resultierenden Einträge, die nicht-verzweigende Expansion erlauben, gleich mitgemacht werden.

- Testen nicht vergessen

Aufgabe: Vertrautmachen mit den Klassen und Packeten

Ergänzt einen Parser,

- der Infixnotation mit Klammerung bei binären Junktoren verarbeitet
- Testen nicht vergessen

Wenn noch Zeit ist

- ergänzt den Parser so, dass er auch unnötige Klammern erlaubt
- oder Klammerersparnisregeln berücksichtigt
- Testen nicht vergessen