Vortrag zum

Asf+Sdf Meta-Enviroment

von Carsten Evers

Gliederung des Vortrags

- Was bedeutet Asf und Sdf?
- Idee hinter Asf+Sdf
- SGLR-Parser
- Aufbau von Modulen
- Sdf-Dateien
- Die Benutzeransicht
- Operatoren
- Attribute + Bool-Beispiel
- Konflikte + Go/Run-Beispiel
- Weiterentwicklungen

Was bedeutet ASF und SDF?

Asf= *Algebraic Specification Formalism*

- abstrakte Syntax von Funktionen
- bedingte Gleichungen

Sdf = *Syntax Definition Formalism*

- zur Definition der Syntax
- abstrakt und konkret

Idee hinter Asf+Sdf

- modularer Aufbau einer abstrakten Grammatik
- Benutzung eines SGLR-Parsers zum Parsen von kontextfreien Grammatiken
- Sdf zur Umsetzung der Syntax
- Asf zur Umsetzung von algebraischen Strukturen
- Funktionen, Gleichungen, Logik

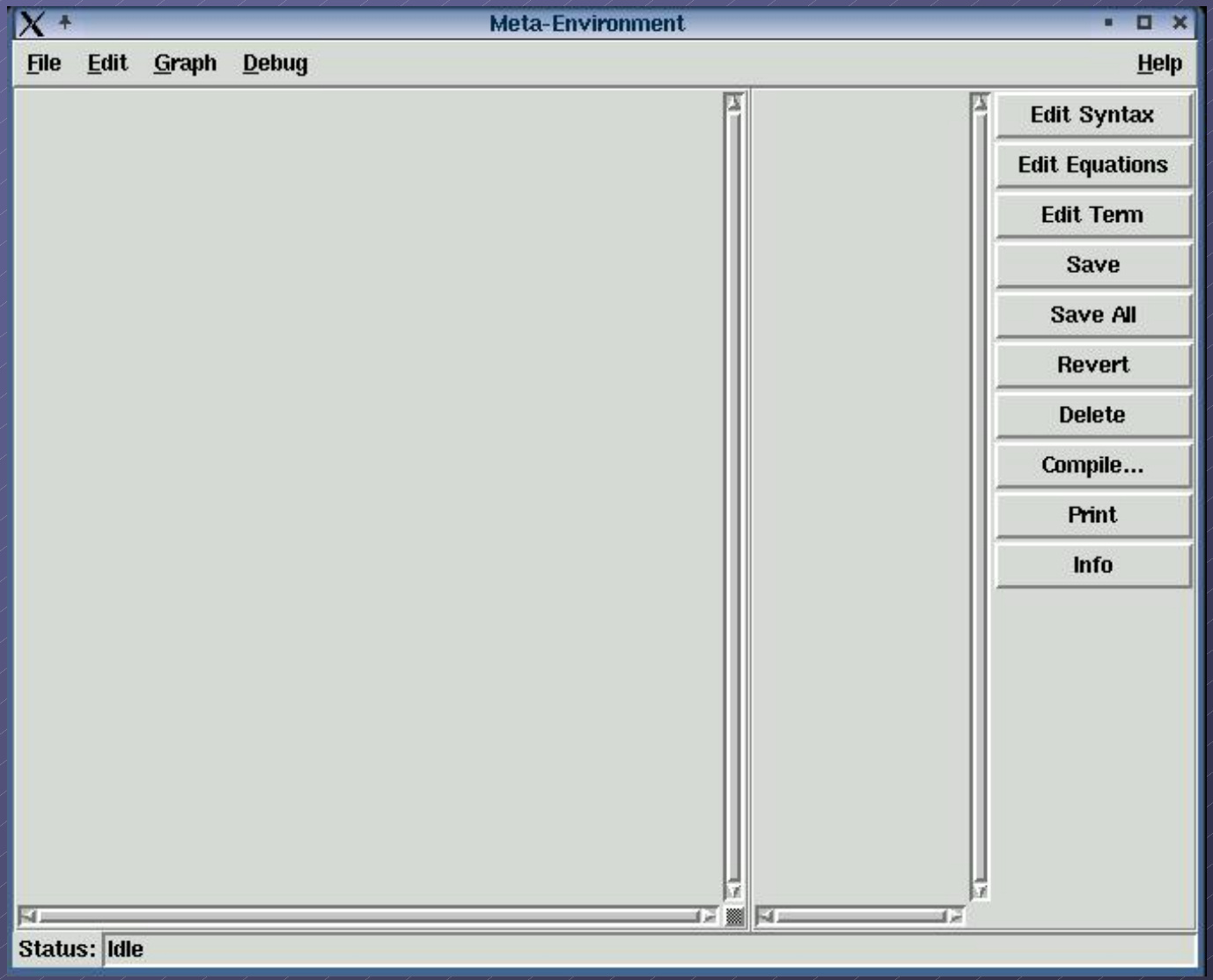
SGLR-Parser

- das Asf+Sdf Meta-Environment arbeitet mit einem „scannerless generalized-LR-parser”, kurz SGLR
- der Parser besitzt keinen separaten Scanner
- der *Generalized-LR Parser* generiert bei Mehrdeutigkeiten mehrere Parsebäume

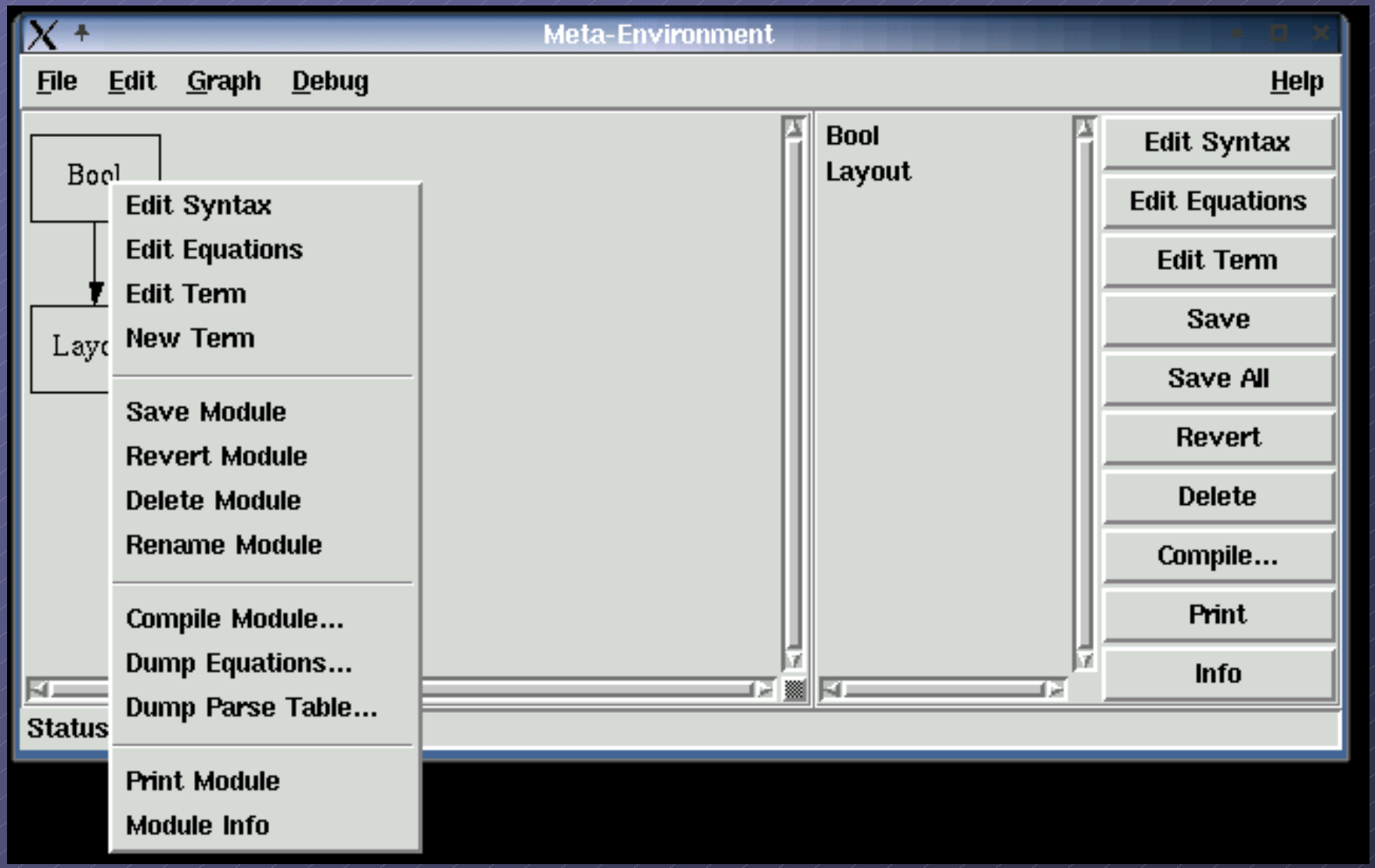
SGLR-Parser

- durch Prioritäten wird geregelt welche Teilbäume der erzeugten Grammatik zu einem einzelnen Parsebaum zuammengelegt werden
- *Lookaheads* mit unterschiedlichen Längen möglich
- Der so entstandene Parsebaum ist frei von Mehrdeutigkeiten

Die Benutzeransicht



Die Benutzeransicht



Aufbau eines Moduls

module<ModuleName>

<ImportSection>*
<ExportSection>*

equations

<ConditionalEquations>*

M.sdf

- Export- und Importsektion, können beide leer sein

module<ModuleName>

exports
<Gramma>+

hiddens
<Gramma>+

<imports>

- durch die Angabe einer <imports>-Sektion lassen sich andere Module in das aktuelle Modul einfügen
- das aktuelles Modul enthält alle sichtbaren Eigenschaften der importierten Module
- per <imports> wird das Konstruieren größerer Grammatiken erleichtert
- spätere Erweiterungen möglich

<sorts>

- alles, was in der <sorts>-Section definiert ist, ist ein Startsymbol
- Probleme beim Parsen werden durch Attribute gelöst
- Attribute bestimmen Vorrang, Reihenfolge, und nicht zu beachtende Regeln

<context-free syntax>

- Unterschied zwischen regulären Ausdrücken und <context-free syntax>–Sektion im Asf+Sdf Meta-Enviroment

- Verwendung von Layouts
- Verwendung von mehr als einem Symbol auf der linken Seite einer Regel

Sdf – lexical syntax

- <sorts> sind Nicht-Terminale und beginnen mit Großbuchstaben gefolgt von weiteren Buchstaben oder Zahlen
- “…“ - zur direkten Angabe von Zeichen
- […] - zur Definition von Zeichenklassen

- Beispiele:
- [a-zA-Z] [a-zA-Z0-9]* = Id
- Id ist ein Symbol vom Typ <sort>

Operatoren

- ? – Operator
- (…) – Sequenz Operator
- * , + , {}, \, \n, \r, \t, \nr
- 3+ - genau 3 mal das Symbol
- | - Alternativ Operator, ist rechts-assoziativ
- # - Tuple-Operator

Operatoren

- <<…>> Permutations- Operator, wie Tuple nur mit allen Kombinationen des Tuples
- Beispiele

1. Tuple

   (Bool # Id # Int) = (true, id345, 79)

2. alle Permutationen von Tuple

   <<Bool Id Int>> = (id345, 79, true)

Operatoren

- -> Funktionsoperator: (Bool Int) –> Int
- LAYOUT – ist eine vordefinierte Regel, bei der alles, was nach LAYOUT abgeleitet wird, überlesen wird
- Beispiel Sdf – Kommentare

```
lexical syntax
"%%" ~[\n]* [\n]   -> LAYOUT
"%" ~[\n%]+ "%"  -> LAYOUT
```

Operatoren

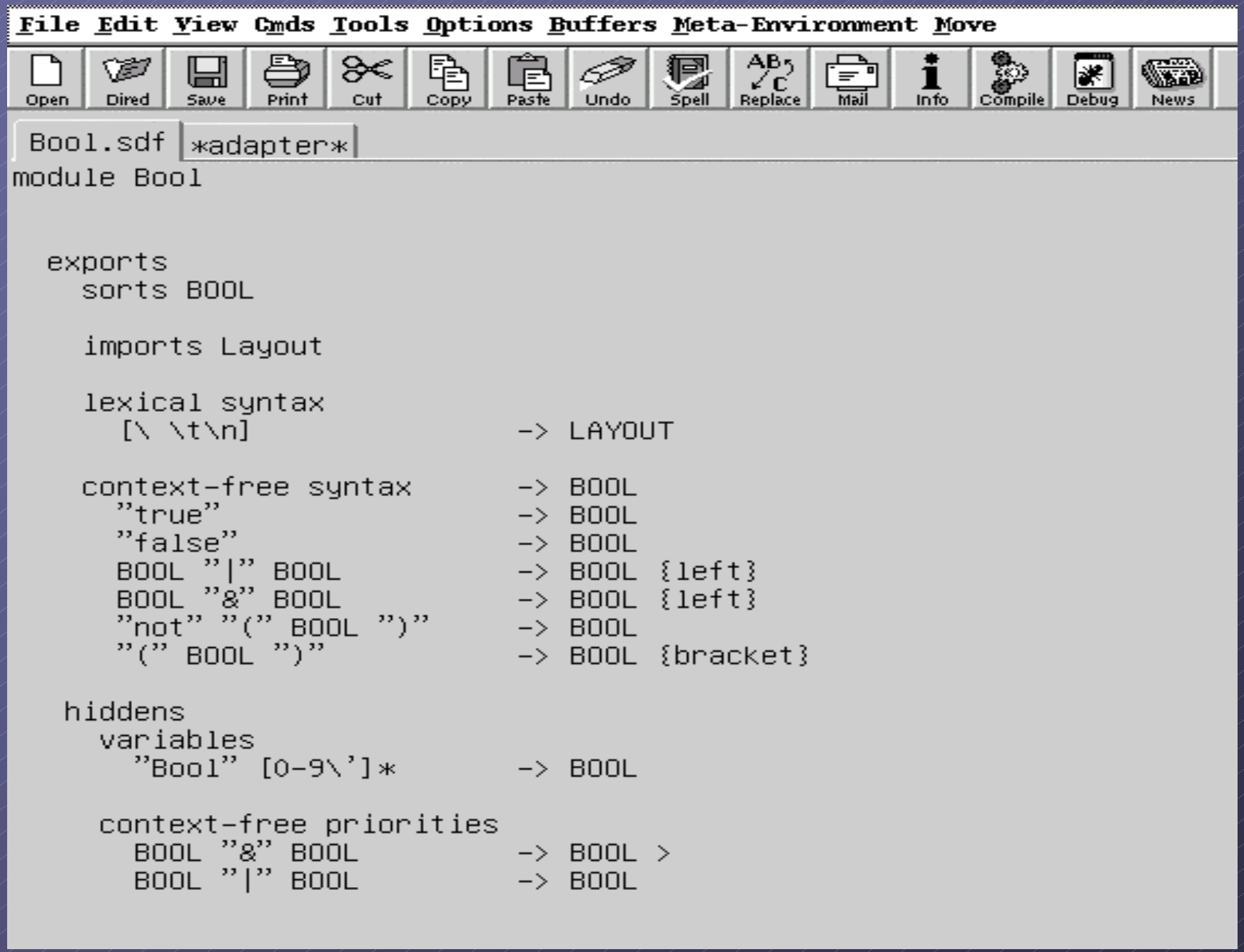
- Character-Class-Operatoren
- ~ - Komplementoperator
- / - Differenzoperator
- /\ - Durchschnittsoperator
- \/ - Vereinigungsoperator
- /, /\, \/ - rechts-assoziativ

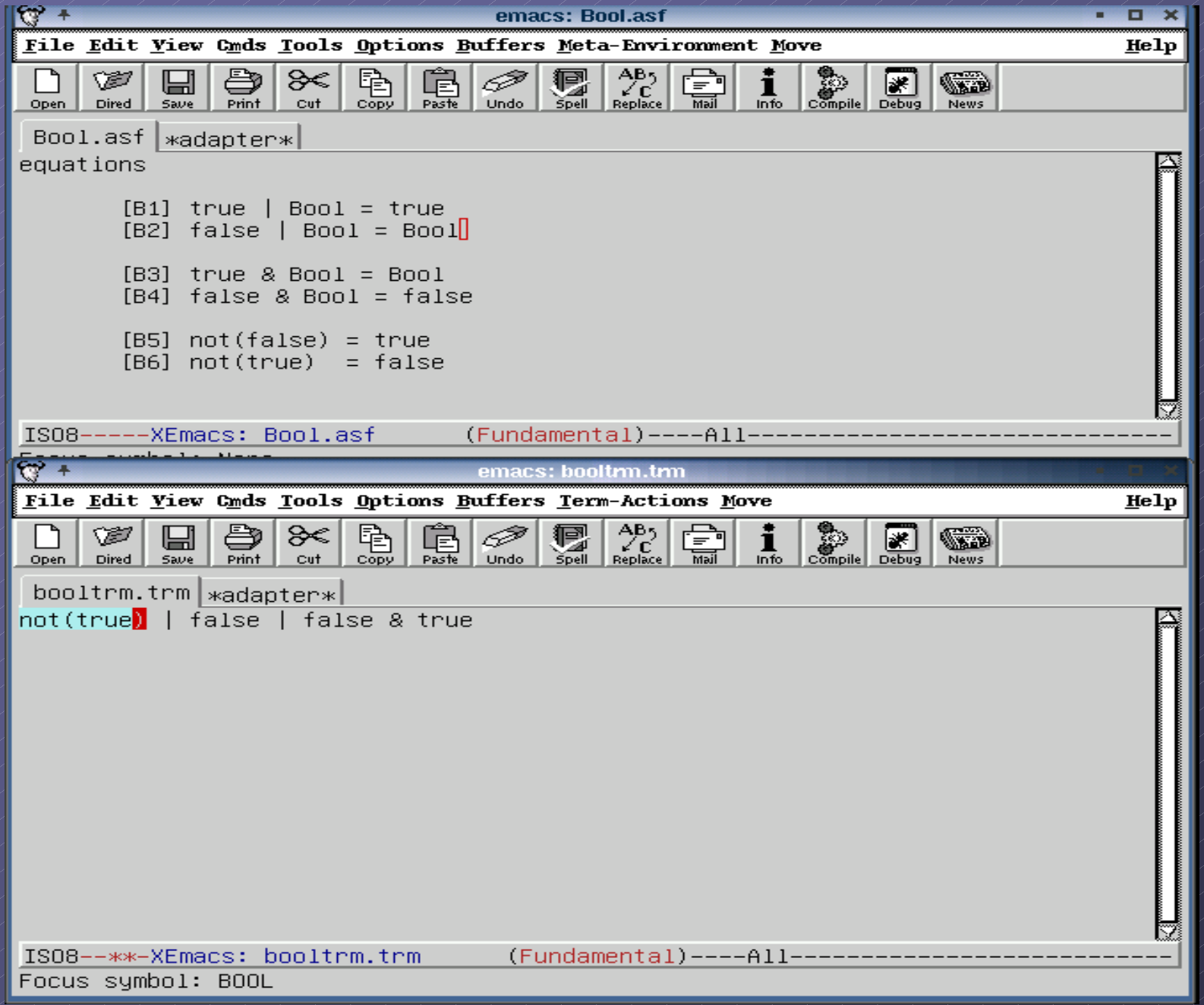
Attribute

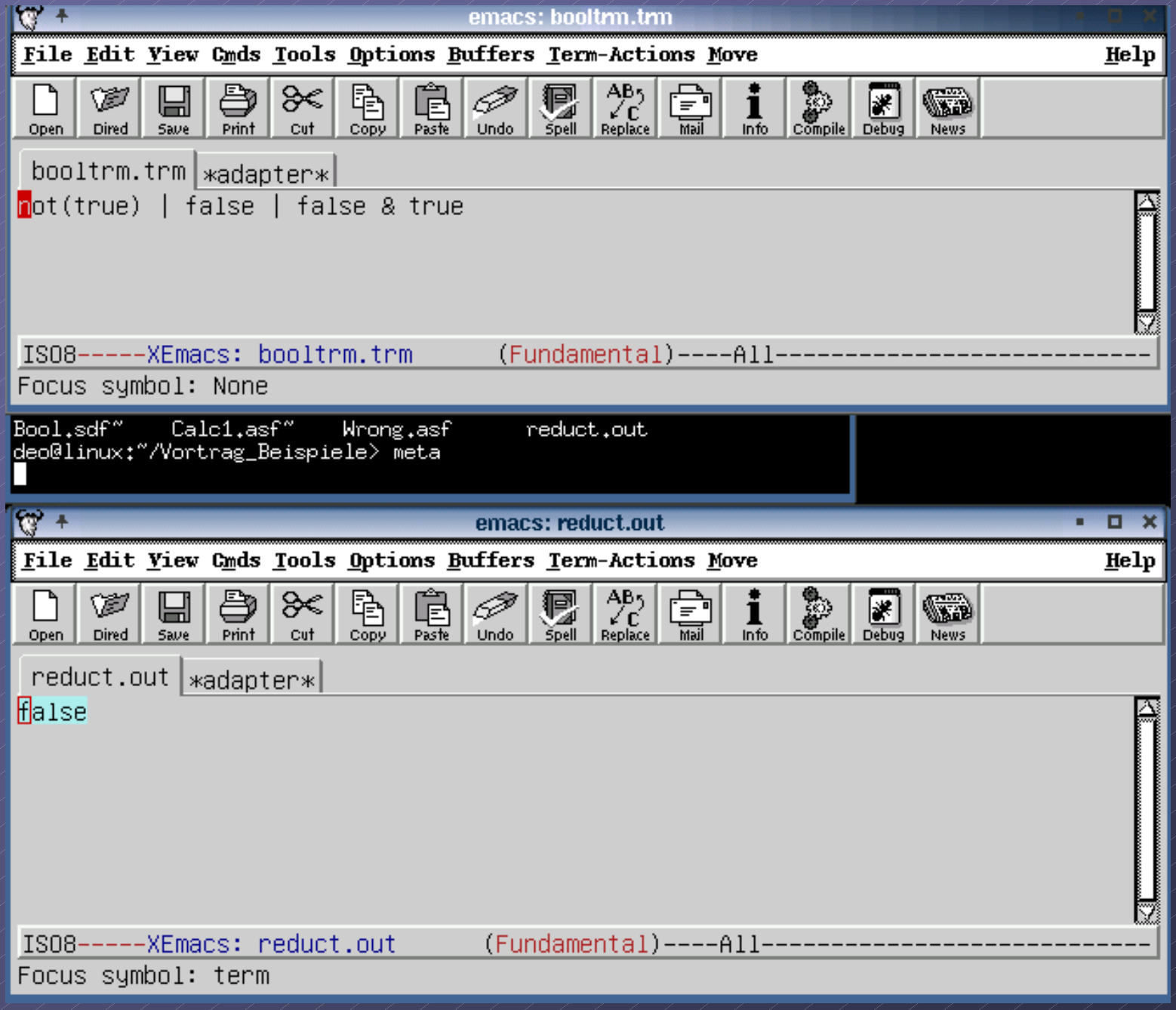
- spezielle Attribute zum Umgehen und Beseitigen von Kreisen und Mehrdeutigkeiten
- Attribute sind ganz rechts in geschweiften Klammern zu schreiben
- > - zur Bestimmung der Priorität zwischen mehrdeutigen Regeln

Attribute

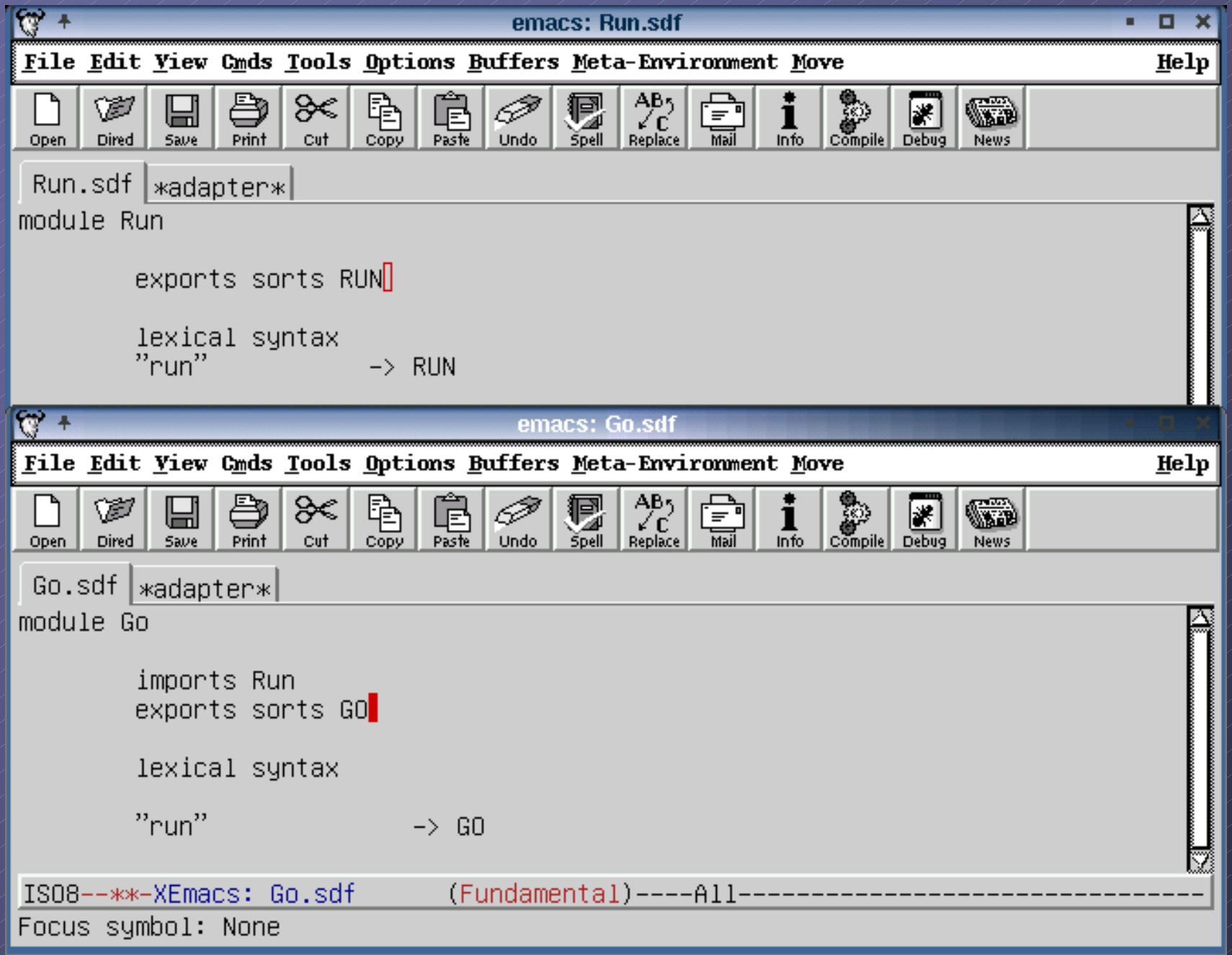
- {bracket} – zum Überschreiben von Prioritäten
- {left}, {assoc}, {right}, {non-assoc} – zur Festlegung der Assoziativität
- {prefer} - Regel mit höchster Priorität
- {avoid} - Regel mit niedrigster Priorität
- {reject} - entfernt Regel aus lexikalischen Konstruktionen

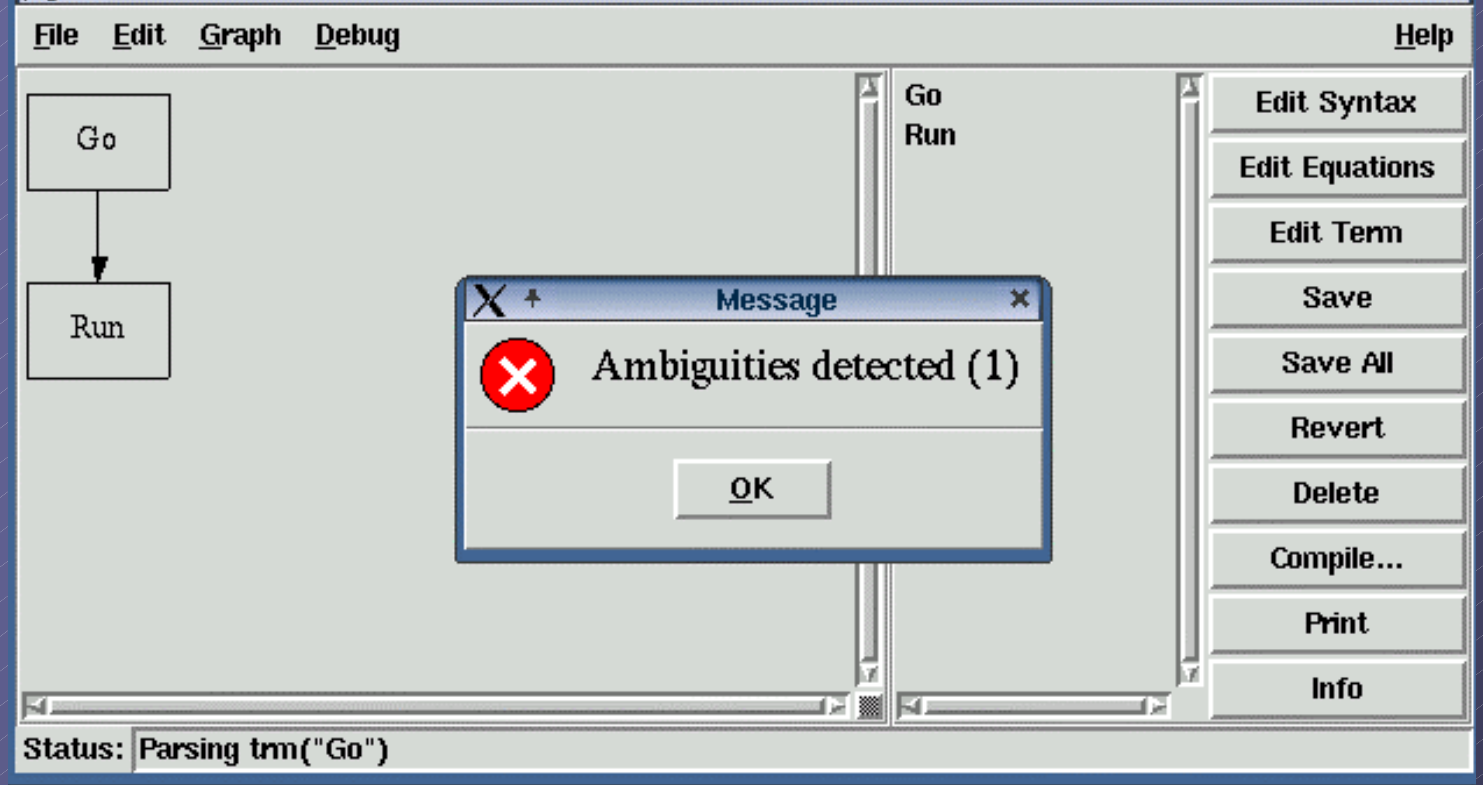

File Edit View Cmds Tools Options Buffers Meta-Environment Move
Open
Dired
Save
Print
Cut
Copy
Paste
Undo
Spell
Replace
Mail
Info
Compile
Debug
News
Bool.sdf
adapter
module Bool
exports
sorts BOOL
imports Layout
lexical syntax
[\ \t\n] -> LAYOUT
context-free syntax -> BOOL
"true" -> BOOL
"false" -> BOOL
BOOL "|" BOOL -> BOOL {left}
BOOL "&" BOOL -> BOOL {left}
"not" "(" BOOL ")" -> BOOL
"(" BOOL ")" -> BOOL {bracket}
hiddens
variables
"Bool" [0-9\']* -> BOOL
context-free priorities
BOOL "&" BOOL -> BOOL >
BOOL "|" BOOL -> BOOL

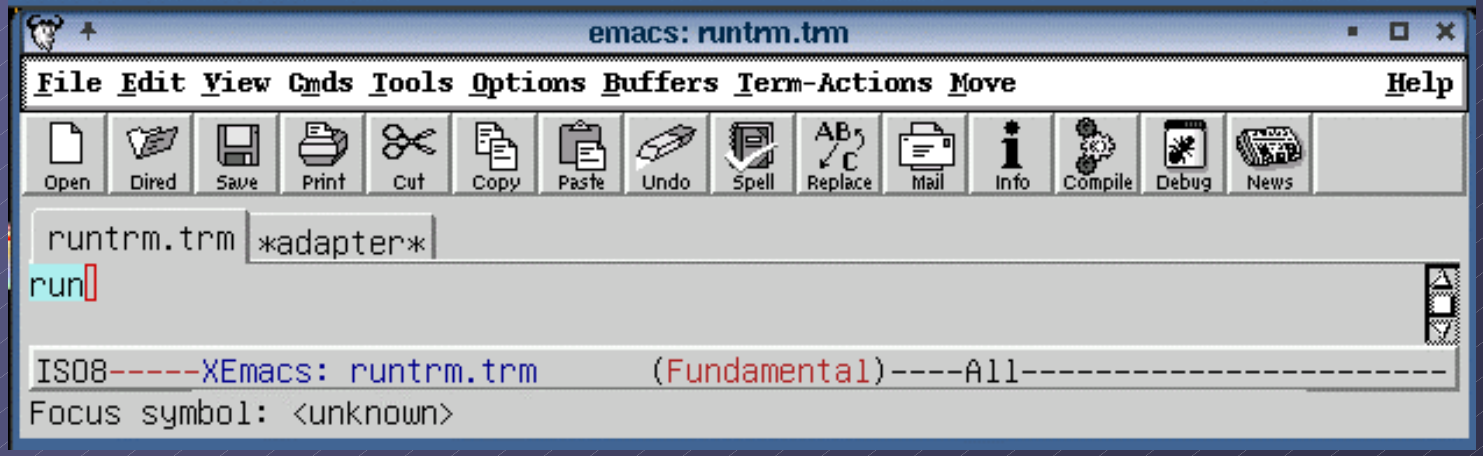

emacs: Bool.asf
File Edit View Cmds Tools Options Buffers Meta-Environment Move Help
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News
Bool.asf *adapter*
equations
[B1] true | Bool = true
[B2] false | Bool = Bool
[B3] true & Bool = Bool
[B4] false & Bool = false
[B5] not(false) = true
[B6] not(true) = false
ISO8-----XEmacs: Bool.asf (Fundamental)----All
emacs: booltrm.trm
File Edit View Cmds Tools Options Buffers Term-Actions Move Help
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News
booltrm.trm *adapter*
not(true) | false | false & true
ISO8--**-XEmacs: booltrm.trm (Fundamental)----All
Focus symbol: BOOL


emacs: booltrm.trm
File Edit View Cmds Tools Options Buffers Term-Actions Move Help
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News
booltrm.trm *adapter*
not(true) | false | false & true
ISO8-----XEmacs: booltrm.trm (Fundamental)----All----------
Focus symbol: None
Bool.sdf~ Calc1.asf~ Wrong.asf reduct.out
deo@linux:~/Vortrag_Beispiele> meta
emacs: reduct.out
File Edit View Cmds Tools Options Buffers Term-Actions Move Help
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News
reduct.out *adapter*
false
ISO8-----XEmacs: reduct.out (Fundamental)----All----------
Focus symbol: term

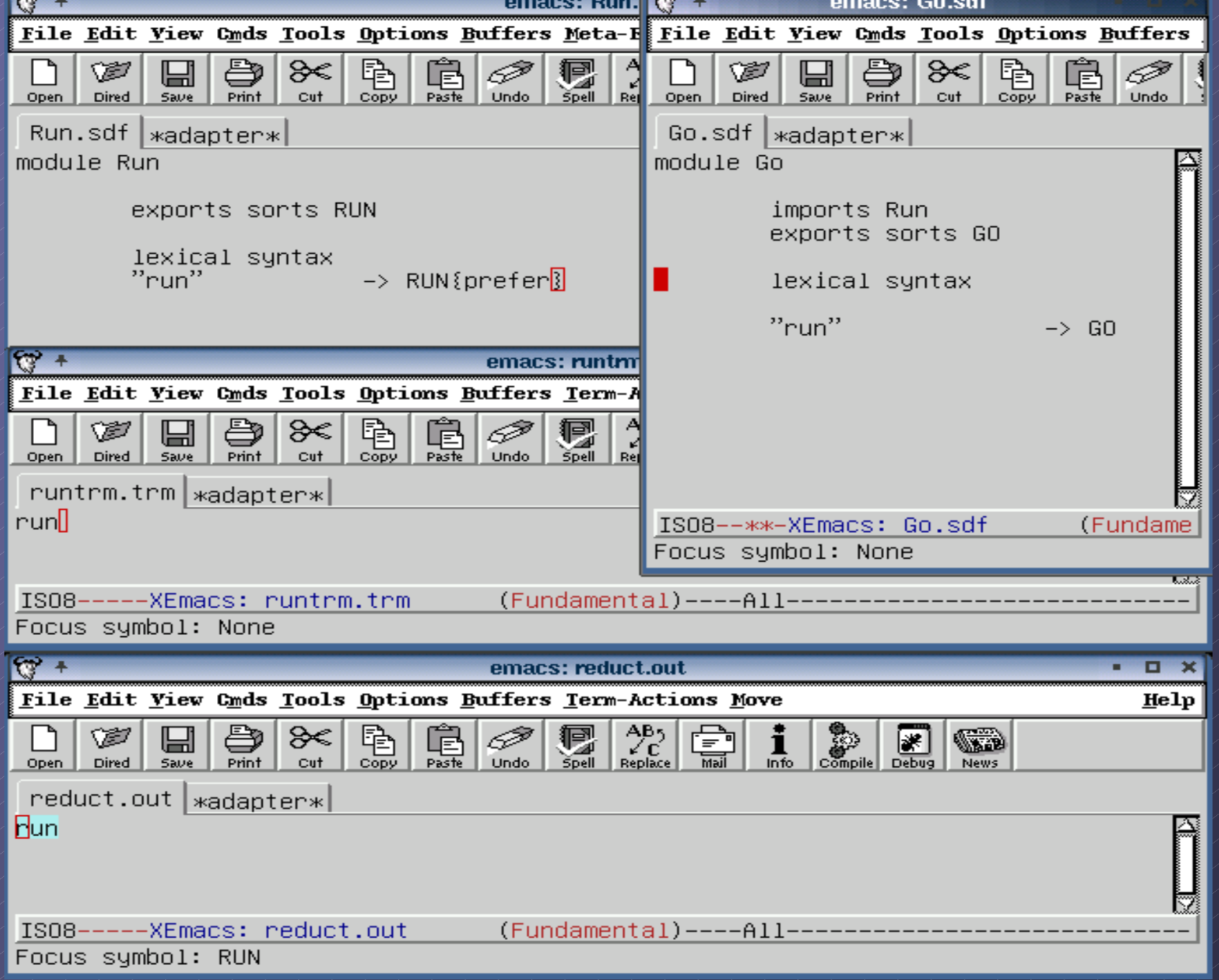

Meta-Environment
File
Edit
Graph
Debug
Help
Go
Run
Go
Run
Edit Syntax
Edit Equations
Edit Term
Save
Save All
Revert
Delete
Compile...
Print
Info
Status:
Idle


emacs: Run.sdf
File Edit View Cmds Tools Options Buffers Meta-Environment Move Help
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News
Run.sdf *adapter*
module Run
exports sorts RUN
lexical syntax
"run" -> RUN
emacs: Go.sdf
File Edit View Cmds Tools Options Buffers Meta-Environment Move Help
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News
Go.sdf *adapter*
module Go
imports Run
exports sorts GO
lexical syntax
"run" -> GO
ISO8--**-XEmacs: Go.sdf (Fundamental)----All----------
Focus symbol: None






```
Ambiguities(1):
[ambiguity(position(character(0),line(1),col(0),char(0)),productions([" GO  -> <START>"," RUN  -> <START>"]))]
```


File Edit View Cmds Tools Options Buffers Meta-E
Open Dired Save Print Cut Copy Paste Undo Spell
Run.sdf *adapter*
module Run
exports sorts RUN
lexical syntax
"run" -> RUN{prefer}
File Edit View Cmds Tools Options Buffers
Open Dired Save Print Cut Copy Paste Undo
Go.sdf *adapter*
module Go
imports Run
exports sorts GO
lexical syntax
"run" -> GO
ISO8--**-XEmacs: Go.sdf (Fundame
Focus symbol: None
emacs: runtrm
File Edit View Cmds Tools Options Buffers Term-A
Open Dired Save Print Cut Copy Paste Undo Spell
runtrm.trm *adapter*
run
ISO8-----XEmacs: runtrm.trm (Fundamental)----All----------
Focus symbol: None
emacs: reduct.out
File Edit View Cmds Tools Options Buffers Term-Actions Move Help
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News
reduct.out *adapter*
run
ISO8-----XEmacs: reduct.out (Fundamental)----All----------
Focus symbol: RUN

Konfliktbeseitigung

- Strategie: Beseitige die Konflikte mit den Vorrangregeln – *prefer* und *avoid*.
- Anzahl der Regeln mit höherer Priorität (*prefer*) und niedriger Priorität (*avoid*) wird vorher berechnet und verglichen
- ein Teilbaum wird dann entfernt, wenn er weniger *prefer*-Attribute und gleich viele oder weniger *avoid*-Attribute enthält
- bei gleicher Anzahl von *prefer*-Attributen wird der Teilbaum entfernt, der mehr *avoid*-Attribute besitzt

Variablen

- Variablen werden definiert, um ein noch nicht endgültig abgeleitetes Namensschema zu definieren und um semantische Aktionen in Regeln an diese Variablen zu binden.
- Beispiel: Booleans

List-Matching

Beispiel:

equations

[set] {Elem*1, Elem, Elem*2, Elem} =

{Elem*1, Elem, Elem*2}

„Elem“ sind zuvor definierte Variablen.

Default Equations

- alle *equations* haben dieselbe Priorität
- die linke Seite einer Regel wird selektiert und mit allen Regeln durchprobiert, bis ein Schema auf die Regel passt ->> Konflikte

Memo Functions

- legt eine Tabelle mit schon probierten Regeln an
- verringert unnötiges doppeltes Ausprobieren von schon probierten Regeln
- *memo functions* sind nicht für alle Funktionen sinnvoll
- gutes Beispiel: Fibonacci-Berechnung
- Beispielwerte: 10.4 s ohne, 3.3 mit *mf*

Was kann ASF+SDF sonst noch?

- Traversal Functions
- Transformer
- Accumulator
- Accumulator Transformer
- Buttom-up
- Top-Down

Ausgesuchte Komponenten

- *parse table*-Generation und *equations* über Menüpunkt
- *parsing*

  sglr –m –p M.trm.tbl < term.txt > term.tree

- Zurückschreiben eines Terms mit dem Evaluator

  asfe –e M.eqs < term.tree > reduct.tree

Stand-Alone-Modules

- Module können ausführbar erzeugt werden über:

  genmakefile –m M > Makefile

  make

- C-Code kann natürlich per Hand ergänzt werden

Weiterentwicklungen

- Kurzzeitziele < 6 Monate
- neues inneres Format
- *facelifting* des Benutzer-Interfaces
- Integration von TIDE, einem *debugger*
- *pretty printer*
- alles Optimieren

Weiterentwicklungen

- Zeitraum: 6 Monate bis 1.5 Jahre
- Traversalfunktionen im Compiler
- mehr Details über die benutzten Regeln in der Ausgabe
- Benutzeroberfläche mit Knöpfen zum An- und Ausschalten von externen Funktionen
- eigenständige Programmierumgebung
- Methoden zur Graphenmanipulation
- Einführen einer Onlinehilfe

Weiterentwicklungen

- Langzeitziele über 1.5 Jahre
- alles noch besser machen
- einfacheres *handling*, *framework*
- Bekanntmachung des Produkts + Feedback

Quellenangabe

- http://www.cwi.nl/htbin/sen1/twiki/bin/view/SEN1/MetaEnvironment
- http://www.cwi.nl/projects/MetaEnv/meta/doc/manual/user-manual.html
- manual.ps
- http://www.phil.uu.nl/sysdox/helpdesk/howto/asf+sdf.html – Apple Installation