

# Übungsblock Prolog

## Debugging in Prolog mittels Trace-Funktion:

- Aufruf: `?-trace(ziel).`
- jeder Term einer Regel wird als zu berechnendes (Teil-)Ziel betrachtet
- „Boxenmodell“ für Auswertung:
  - Box ordnet jedem Teilziel vier Ports zu:
  - Call: erster Aufruf eines Ziels
  - Exit: Ziel wird verlassen, falls Aufruf erfolgreich war
  - Fail: Ziel wird unerfüllbar verlassen
  - Redo: erneuter Zielaufruf innerhalb des Backtrackings, d.h. Suche nach weiteren Möglichkeiten, Ziel zu erfüllen

# Übungsblock Prolog

Beispiel:

?- assertz(term3(X,Y):-term1(X,Z),term2(Z,Y)).

?- trace(*term1*). %trace mit allen Ports

?- trace(*term2*).

?- trace(*term3*).

?- term3(hugo,Y).

T Call: (7) term3(hugo,\_G324)

T Call: (8) term1(hugo,\_G396)

T Exit: (8) term1(hugo,eva)

T Call: (8) term2(eva,\_G324)

....

T Exit: (7) term3(hugo,bernd)

Y = bernd

# Übungsblock Prolog

## Debugging bestimmter Ports:

- ?- trace(*term1*,+*call*).    %zeigt alle call-Aufrufe von term1 an
- ?- trace(*term1*,+*fail*).    %zeigt alle fails von term1 an
- ?- trace(*term3*).    %zeigt alle Ports zu term3 an
- ?- trace(*term3*,-redo).    /\*zeigt die redo-Aufrufe von term3  
nicht an \*/
  
- ?- nodebug.    /\*stoppt den Debugger und damit  
das Anzeigen der Trace-Punkte \*/