

# Tracing



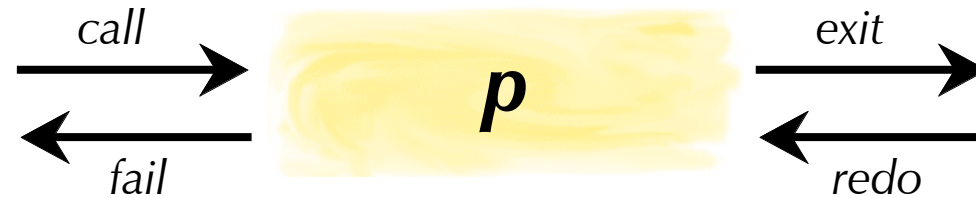
## Übersicht

- ◆ Kästchenmodell
- ◆ Beispiel: ?- person(Wer), weiblich(Wer).
- ◆ *trace* und *notrace*

## Ziel

- ◆ eine andere Art, Prologs Vorgehen zu verstehen
- ◆ sehr nützlich für die Fehlersuche in Prolog-Programmen

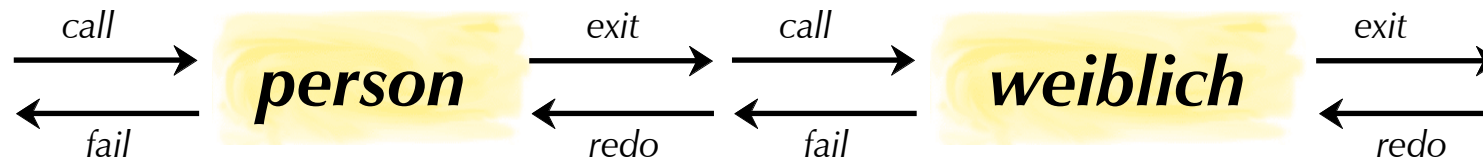
# Kästchenmodell



Ein Anfrage-Prädikat  $p$  kann als Kästchen mit vier »Ports« dargestellt werden.

- ◆ Zwei Eingänge
  - ◆ *call* —  $p$  wird zum ersten Mal aufgerufen
  - ◆ *redo* —  $p$  wird durch Backtracking aufgerufen
- ◆ Zwei Ausgänge
  - ◆ *exit* —  $p$  war erfolgreich
  - ◆ *fail* —  $p$  schlug fehl

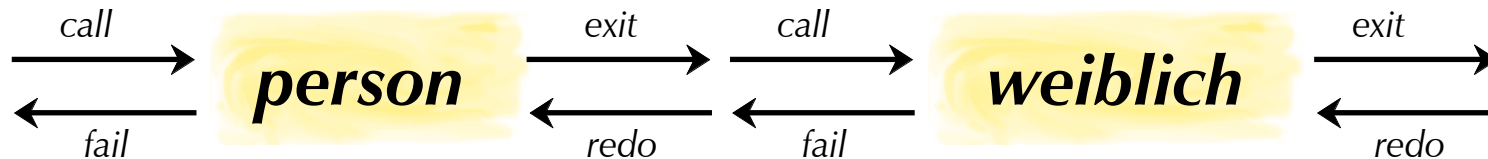
# Kästchenmodell



**Sind mehrere Prädikate in der Anfrage, kann man sie sich aneinanderghängt vorstellen.**

- ◆ Die Kette beginnt mit dem ersten *call* (ganz links)
- ◆ Die gesamte Anfrage gelingt mit dem letzten *exit* (ganz rechts)

# Ein Beispiel



?- **person(Wer), weiblich(Wer).**

- ◆ call: person(Wer)
- ◆ exit: person(hans)
  - ◆ call: weiblich(hans)
  - ◆ fail: weiblich(hans)
- ◆ redo: person(hans)
- ◆ exit: person(klara)
  - ◆ call: weiblich(klara)
  - ◆ exit: weiblich(klara)
- ◆ Wer = klara

```
person(hans).  
person(klara).  
person(sabrina).  
person(kevin).  
  
hund(fido).  
  
weiblich(klara).  
weiblich(sabrina).
```

Wissensbasis

# trace/0 · notrace/0

## Prolog kann dieses »Tracing« ausgeben.

- ◆ Einschalten mit *trace*, Ausschalten mit *notrace*
- ◆ Sehr nützlich beim Suchen von Programmierfehlern

| ?- **trace.**

{The debugger will first creep -- showing everything (trace)}

yes

{trace}

| ?- **person(Wer), weiblich(Wer).**

```
1 1 Call: person(_187) ?
1 1 Exit: person(hans) ?
2 1 Call: weiblich(hans) ?
2 1 Fail: weiblich(hans) ?
1 1 Redo: person(hans) ?
1 1 Exit: person(klara) ?
2 1 Call: weiblich(klara) ?
2 1 Exit: weiblich(klara) ?
```

Strichpunkt erzwingt Backtracking

Wer = klara ? ;

```
2 1 Redo: weiblich(klara) ?
2 1 Fail: weiblich(klara) ?
1 1 Redo: person(klara) ?
1 1 Exit: person(sabrina) ?
2 1 Call: weiblich(sabrina) ?
2 1 Exit: weiblich(sabrina) ?
```

Wer = sabrina ? ;

```
2 1 Redo: weiblich(sabrina) ?
2 1 Fail: weiblich(sabrina) ?
1 1 Redo: person(sabrina) ?
1 1 Exit: person(kevin) ?
2 1 Call: weiblich(kevin) ?
2 1 Fail: weiblich(kevin) ?
1 1 Redo: person(kevin) ?
1 1 Fail: person(_187) ?
```

no