

# Prolog: Suche

Michael Leuschel

# Tiefensuche

- kante(a,b). kante(a,c). kante(b,d). kante(c,e).
- start(a). ziel(e).

# Tiefensuche I

- $\text{dfs} \text{ :- start}(S), \text{dfs}(S).$
- $\text{dfs}(S) \text{ :- ziel}(S).$
- $\text{dfs}(S) \text{ :- kante}(S, S2), \text{dfs}(S2).$
  
- Mit Pfad als Ergebnis?

# Tiefensuche 2

- Mit Pfad als Ergebnis:

`dfs(Pfad) :- start(S), dfs(S,Pfad).`

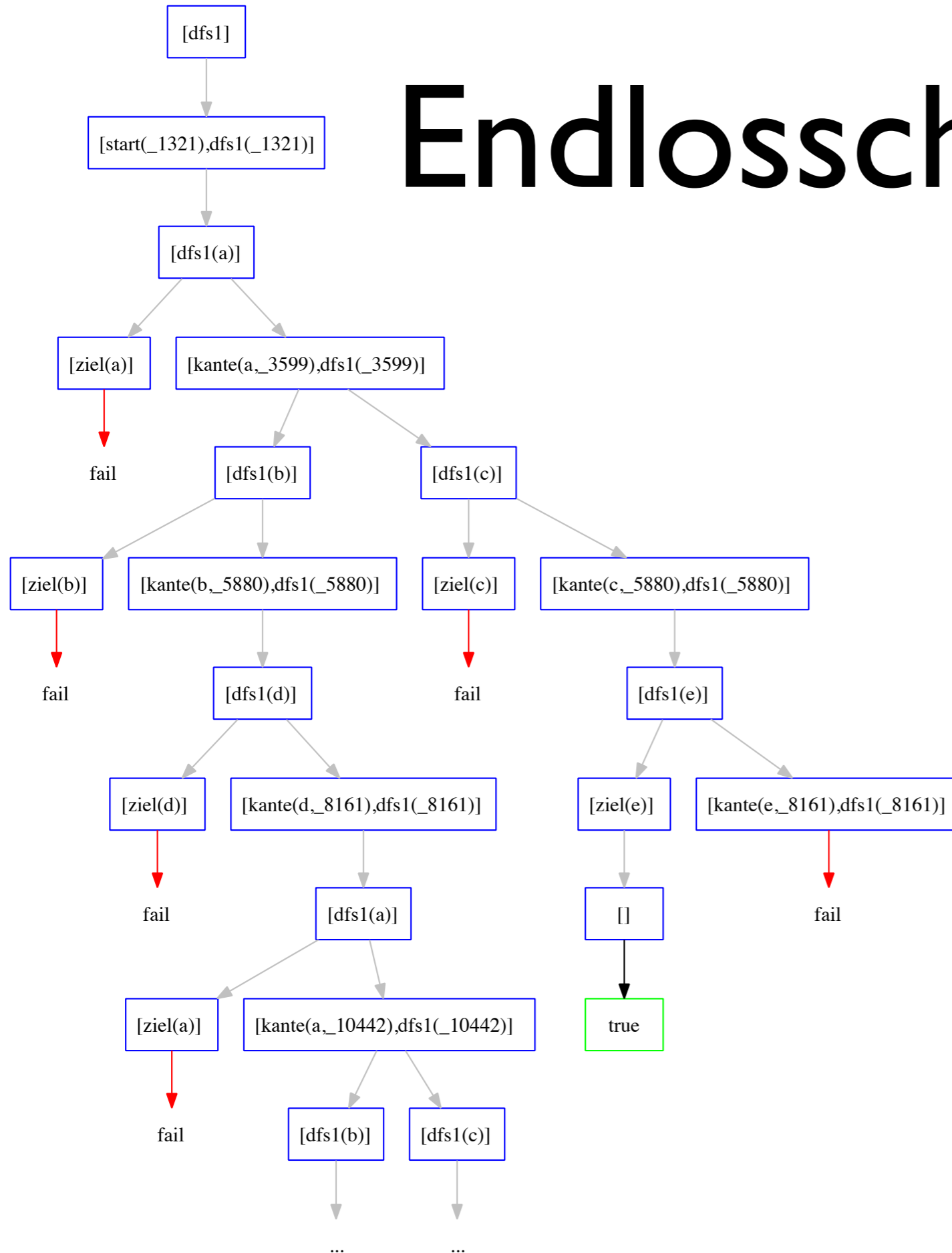
`dfs(S,[]) :- ziel(S).`

`dfs(S,[S|Rest]) :- kante(S,S2), dfs(S2,Rest).`

# Tiefensuche

- kante(a,b). kante(a,c). kante(b,d). kante(c,e).
- **kante(d,a).**
- start(a). ziel(e).
  
- Was passiert jetzt ?
- Warum? Lösungen ?

# Endlosschleifen



kante(a,b). kante(a,c). kante(b,d). kante(c,e). kante(d,a).

dfs1 :- start(S), dfs1(S).  
dfs1(S) :- ziel(S).  
dfs1(S) :- kante(S,S2), dfs1(S2).

# Schleifen Erkennen

- `dfs2(Pfad) :- start(S), dfs2(S,Pfad,[]).`
- `dfs2(S,[S],_) :- ziel(S).`
- `dfs2(S,[S|Rest],Hist) :- \+ member(S,Hist),  
kante(S,S2), dfs2(S2,Rest,[S|Hist]).`

# Version mit AVL Baum

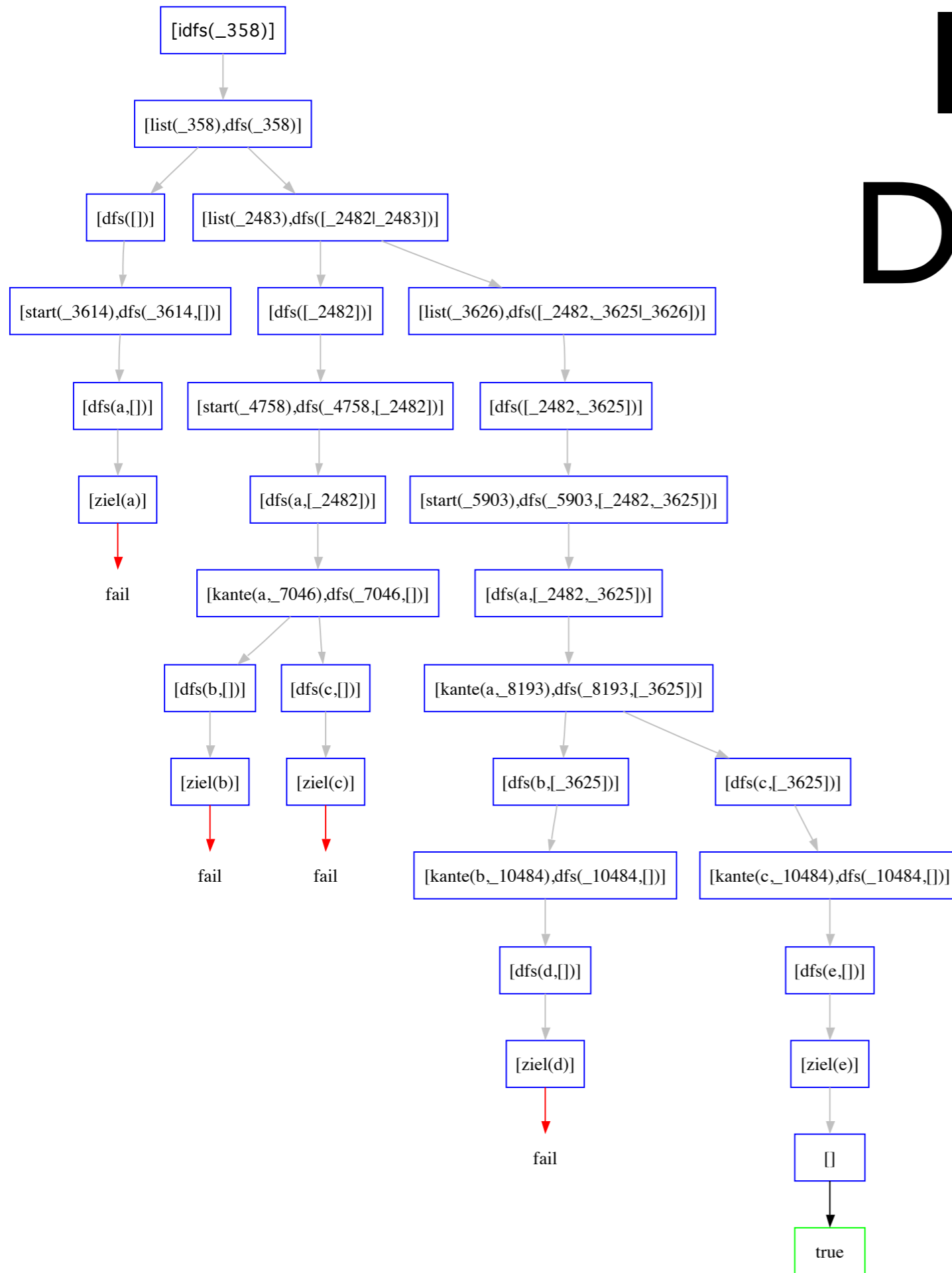
```
:- use_module(library(avl)).
```

```
dfsa(Pfad) :- start(S), empty_avl(E),dfsa  
(S,Pfad,E).
```

```
dfsa(S,[S],_) :- ziel(S).
```

```
dfsa(S,[S|Rest],Hist) :- \+ avl_fetch(S,Hist),  
kante(S,S2), avl_store(S,Hist,true,NewHist),  
dfsa(S2,Rest,NewHist).
```

# Iterative Deepening



# Iterative Deepening in Prolog

- `list([], list([_|T]) :- list(T).`
- `idfs(Pfad) :- list(Pfad), dfs(Pfad).`