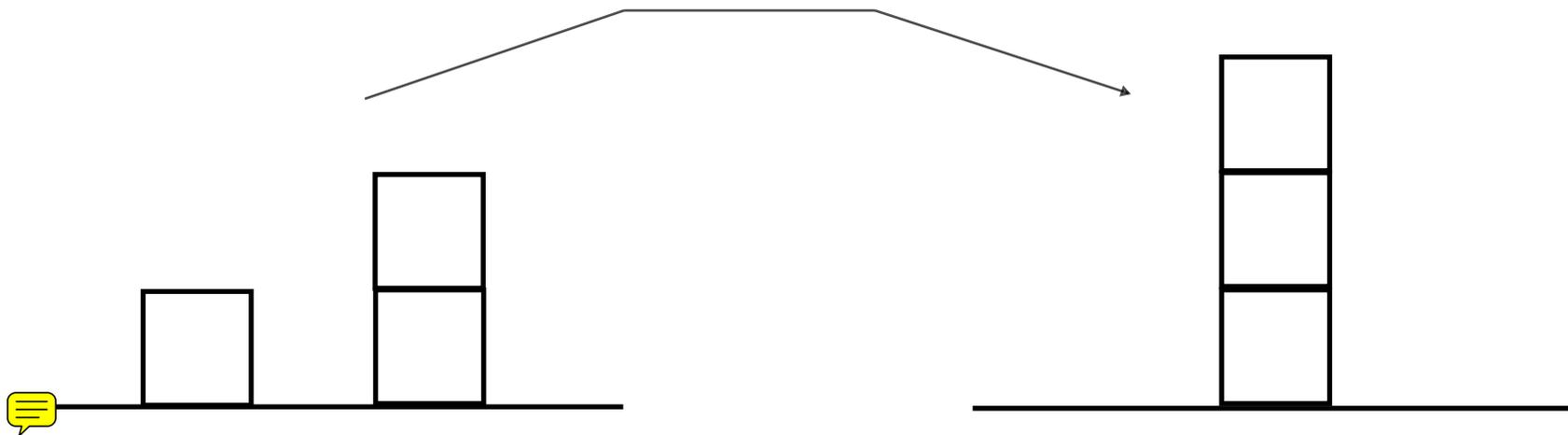


Inhalt

Logische Grundlagen des Planens

Logische Grundlagen des Planens



Der Situationskalkül

Der Situationskalkül - Aktionen

Der Situationskalkül - Situationen

Der Situationskalkül - Veränderungen zwischen Situationen

Der Situationskalkül - Ausführungsbedingungen und Effektaxiome

$\text{Poss}(\text{STACK}(x, y), s)$

$$\equiv \text{ONTABLE}(x, s) \wedge \text{CLEAR}(x, s) \wedge \text{CLEAR}(y, s) \wedge x \neq y$$

$\text{Poss}(\text{UNSTACK}(x, y), s)$

$$\equiv \text{ON}(x, y, s) \wedge \text{CLEAR}(x, s)$$

Der Situationskalkül - Ausführungsbedingungen und Effektaxiome

Ausführungsbedingung \Rightarrow Effekt

$\text{Poss}(a, s) \Rightarrow \text{Fluent}(\dots, \text{do}(a, s))$

$\text{Poss}(\text{STACK}(x, y), s) \Rightarrow \text{ON}(x, y, \text{do}(\text{STACK}(x, y), s))$

$\text{Poss}(\text{STACK}(x, y), s) \Rightarrow \neg \text{CLEAR}(y, \text{do}(\text{STACK}(x, y), s))$

Der Situationskalkül - Zielbeschreibungen

$$\exists t \text{ ON}(A, B, t) \wedge \text{ON}(B, C, t)$$

Der Situationskalkül - Zielbeschreibungen

$$\exists t \text{ ON}(A, B, t) \wedge \text{ON}(B, C, t)$$

$$\text{Poss}(\text{STACK}(A, B), S_0)$$

$$\equiv \text{ONTABLE}(A, S_0) \wedge \text{CLEAR}(A, S_0) \wedge \text{CLEAR}(B, S_0)$$

$$\text{ON}(A, B, \text{do}(\text{STACK}(A, B), S_0))$$

$$\neg \text{CLEAR}(B, \text{do}(\text{STACK}(A, B), S_0))$$

$$\neg \text{ONTABLE}(A, \text{do}(\text{STACK}(A, B), S_0))$$

$$\text{ON}(A, B, \text{do}(\text{STACK}(A, B), S_0)) \wedge \text{ON}(B, C, \text{do}(\text{STACK}(A, B), S_0))$$

Der Situationskalkül - Probleme

Der Situationskalkül - Rahmenproblem

$$\text{ON}(x, y, \mathbf{s}) \wedge x \neq u \Rightarrow \text{ON}(x, y, \text{do}(\text{STACK}(u, v), \mathbf{s}))$$

$$\neg \text{ON}(x, y, \mathbf{s}) \wedge (x \neq u \vee y \neq v) \Rightarrow \neg \text{ON}(x, y, \text{do}(\text{STACK}(u, v), \mathbf{s}))$$

Der Situationskalkül - Qualifikationsproblem

Der Situationskalkül - Verzweigungsproblem

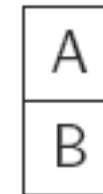
Der Situationskalkül - Plangenerierung

1. $\{\neg \text{ONTABLE}(x, s), \neg \text{CLEAR}(x, s), \neg \text{CLEAR}(y, s), \text{ON}(x, y, \text{do}(\text{STACK}(x, y), s))\}$
2. $\{\neg \text{ONTABLE}(x, s), \neg \text{CLEAR}(x, s), \neg \text{CLEAR}(y, s), \neg \text{CLEAR}(y, \text{do}(\text{STACK}(x, y), s))\}$
3. $\{\neg \text{ONTABLE}(x, s), \neg \text{CLEAR}(x, s), \neg \text{CLEAR}(y, s), \neg \text{ONTABLE}(x, \text{do}(\text{STACK}(x, y), s))\}$
4. $\{\neg \text{ON}(x, y, s), x = u, \text{ON}(x, y, \text{do}(\text{STACK}(u, v), s))\}$
5. $\{\text{ON}(x, y, s), x = u, \neg \text{ON}(x, y, \text{do}(\text{STACK}(u, v), s))\}$
6. $\{\text{ON}(x, y, s), y = v, \neg \text{ON}(x, y, \text{do}(\text{STACK}(u, v), s))\}$
7. $\{\neg \text{ONTABLE}(x, s), x = u, \text{ONTABLE}(x, \text{do}(\text{STACK}(u, v), s))\}$
8. $\{\text{ONTABLE}(x, s), x = u, \neg \text{ONTABLE}(x, \text{do}(\text{STACK}(u, v), s))\}$
9. $\{\neg \text{CLEAR}(x, s), x = v, \text{CLEAR}(x, \text{do}(\text{STACK}(u, v), s))\}$
10. $\{\text{CLEAR}(x, s), \neg \text{CLEAR}(x, \text{do}(\text{STACK}(u, v), s))\}$
11. $\{\neg \text{ON}(x, y, s), \neg \text{CLEAR}(x, s), \neg \text{ON}(x, y, \text{do}(\text{UNSTACK}(x, y), s))\}$
12. $\{\neg \text{ON}(x, y, s), \neg \text{CLEAR}(x, s), \text{ONTABLE}(x, \text{do}(\text{UNSTACK}(x, y), s))\}$
13. $\{\neg \text{ON}(x, y, s), \neg \text{CLEAR}(x, s), \text{CLEAR}(y, \text{do}(\text{UNSTACK}(x, y), s))\}$
14. $\{\neg \text{ON}(x, y, s), x = u, \text{ON}(x, y, \text{do}(\text{UNSTACK}(u, v), s))\}$
15. $\{\text{ON}(x, y, s), \neg \text{ON}(x, y, \text{do}(\text{UNSTACK}(u, v), s))\}$
16. $\{\neg \text{ONTABLE}(x, s), \text{ONTABLE}(x, \text{do}(\text{UNSTACK}(u, v), s))\}$
17. $\{\text{ONTABLE}(x, s), x = u, \neg \text{ONTABLE}(x, \text{do}(\text{UNSTACK}(u, v), s))\}$
18. $\{\neg \text{CLEAR}(x, s), \text{CLEAR}(x, \text{do}(\text{UNSTACK}(u, v), s))\}$
19. $\{\text{CLEAR}(x, s), x = v, \neg \text{CLEAR}(x, \text{do}(\text{UNSTACK}(u, v), s))\}$
20. $\{\text{ONTABLE}(A, S_0)\}$
21. $\{\text{ONTABLE}(B, S_0)\}$
22. $\{\text{ON}(C, B, S_0)\}$
23. $\{\text{CLEAR}(A, S_0)\}$
24. $\{\text{CLEAR}(C, S_0)\}$
25. $\{\neg \text{ON}(A, B, t)\}$

Anfangszustand

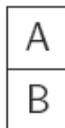
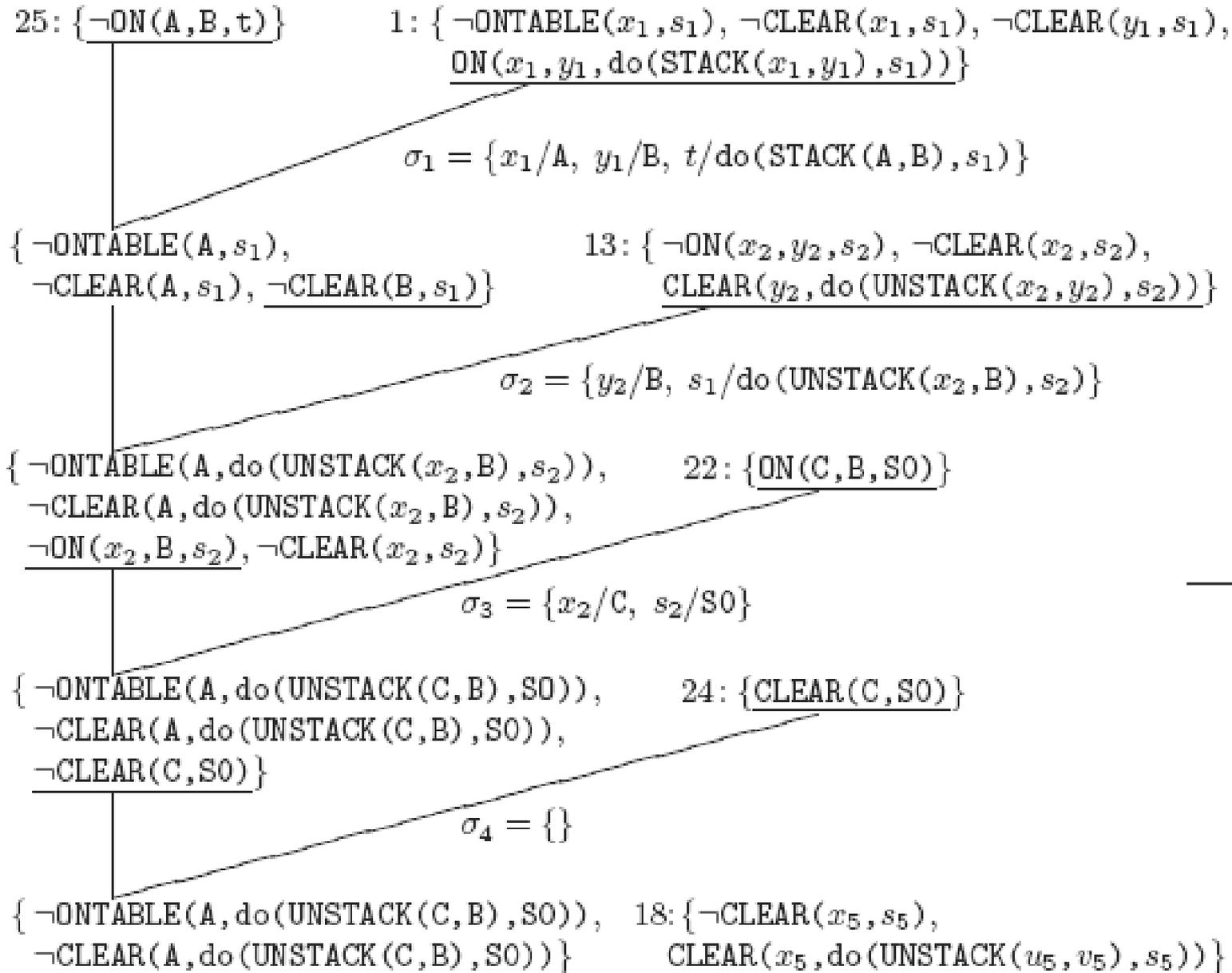


Ziel

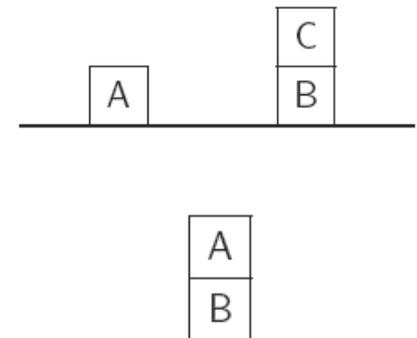
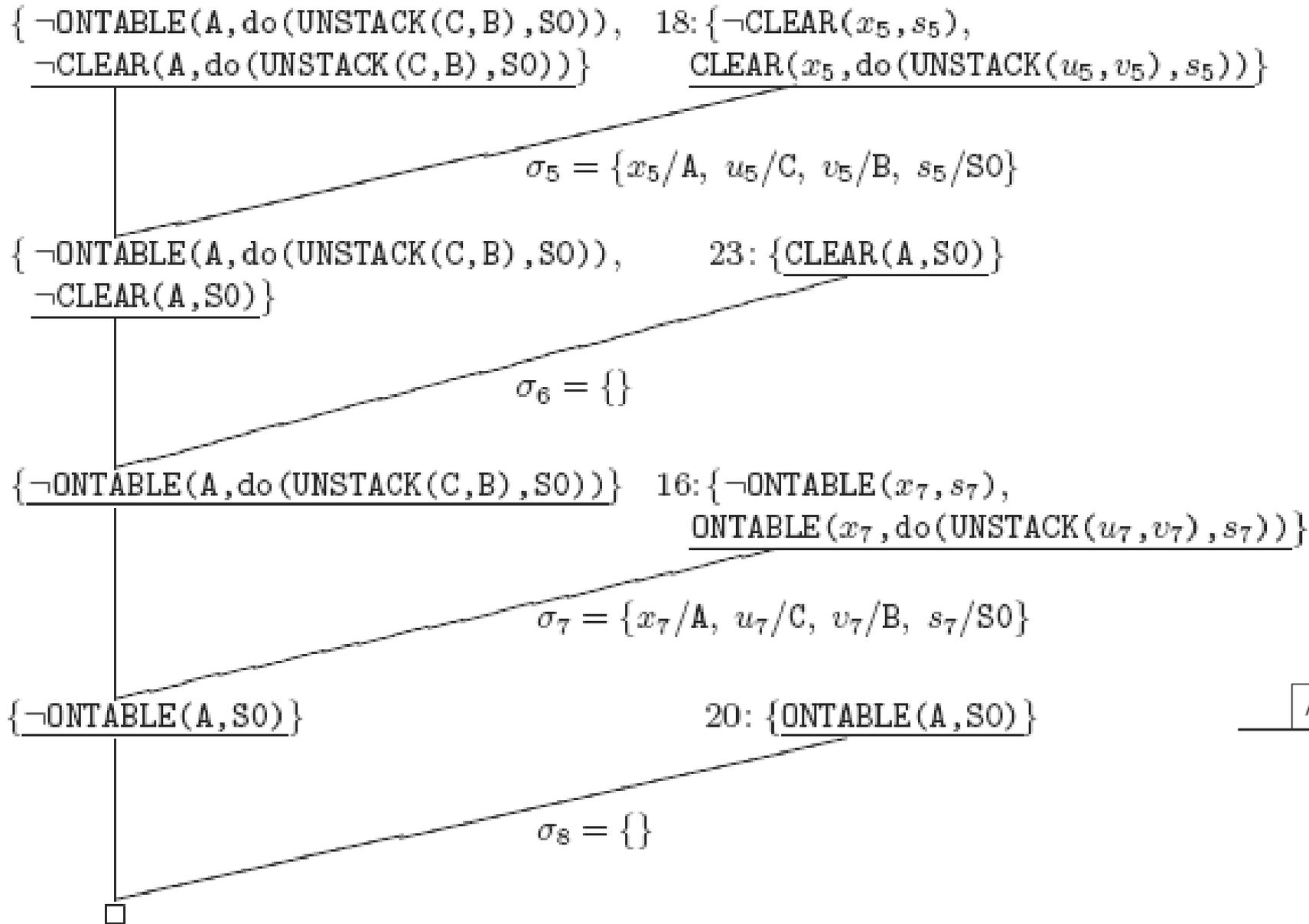


Effektaxiome (1-3 und 11-13),
Rahmenaxiome (4-10 und 14-19),
Ausgangszustand (20-24) und
negierte Zielbeschreibung (25) in
Klauselform

Der Situationskalkül - Plangenerierung



Der Situationskalkül - Plangenerierung



Der Situationskalkül - Plangenerierung

$$\sigma = \sigma_8 \circ \sigma_7 \circ \dots \circ \sigma_1$$

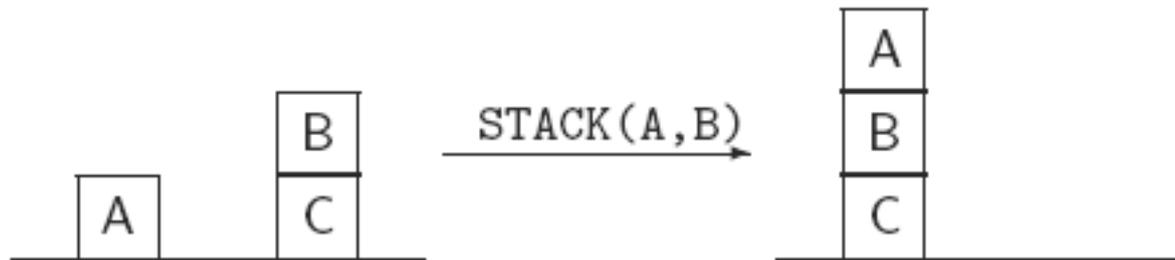
σ

STRIPS

STRIPS - Zustände und Zielbeschreibungen

STRIPS - Operatoren

STRIPS - Operatoren



Datenbasis vorher:

ONTABLE(A)
 CLEAR(B)
 CLEAR(A)
 ON(B, C)
 ONTABLE(C)

D-list

A-list

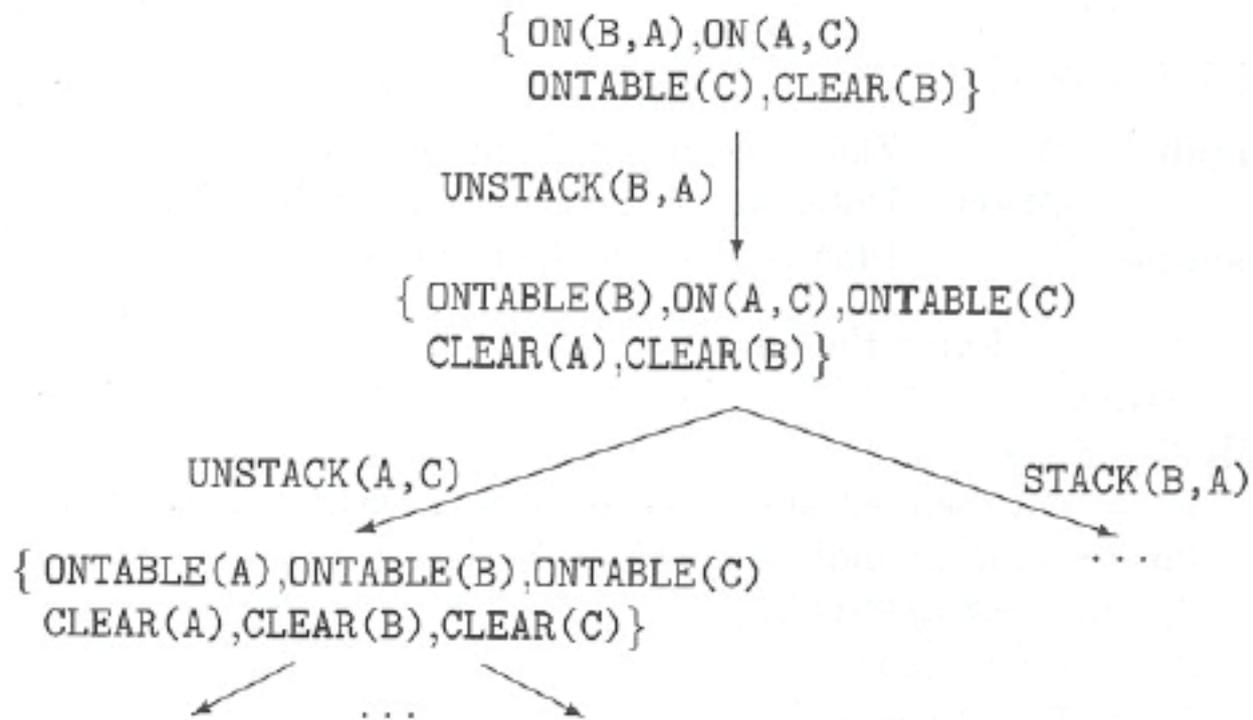
unverändert

Datenbasis nachher:

ON(A, B)
 CLEAR(A)
 ON(B, C)
 ONTABLE(C)

STRIPS - Operatoren

STRIPS - Planen mit Vorwärtssuche



STRIPS - Planen mit Rückwärtssuche

STRIPS - R-STRIPS Algorithmus

Eingabe: G Ziel (Menge von Grundliteralsen)
Start Datenbasis (Menge von Grundliteralsen)
Ausgabe: P Plan (Liste von Operatoren)

1. $P := []$ % leerer Plan als Initialisierung
2. $S := \text{Start}$
3. **while** $G \not\subseteq S$ **do**
4. $g :=$ ein Element aus G , das nicht in S enthalten ist
5. $Op :=$ eine Grundinstanz (C, D, A) eines Operators R ,
 so dass $g \in A$
6. $P_C := \text{R-STRIPS}(C, S)$
7. $S := Op(P_C(S))$
8. $P := P + P_C + [Op]$
- end while**
9. **return**(P)

STRIPS - R-STRIPS Algorithmus

MOVE(x, y, z) nimmt Block x von Block y herunter und setzt ihn auf Block z (entspricht der Sequenz [UNSTACK(x, y), STACK(x, z)])

STRIPS-Regel für MOVE:

MOVE(x, y, z): C: ON(x, y), CLEAR(x), CLEAR(z)
 D: ON(x, y), CLEAR(z)
 A: ON(x, z), CLEAR(y)

Ausgangsdatenbasis **Start** und Ziel **G**:

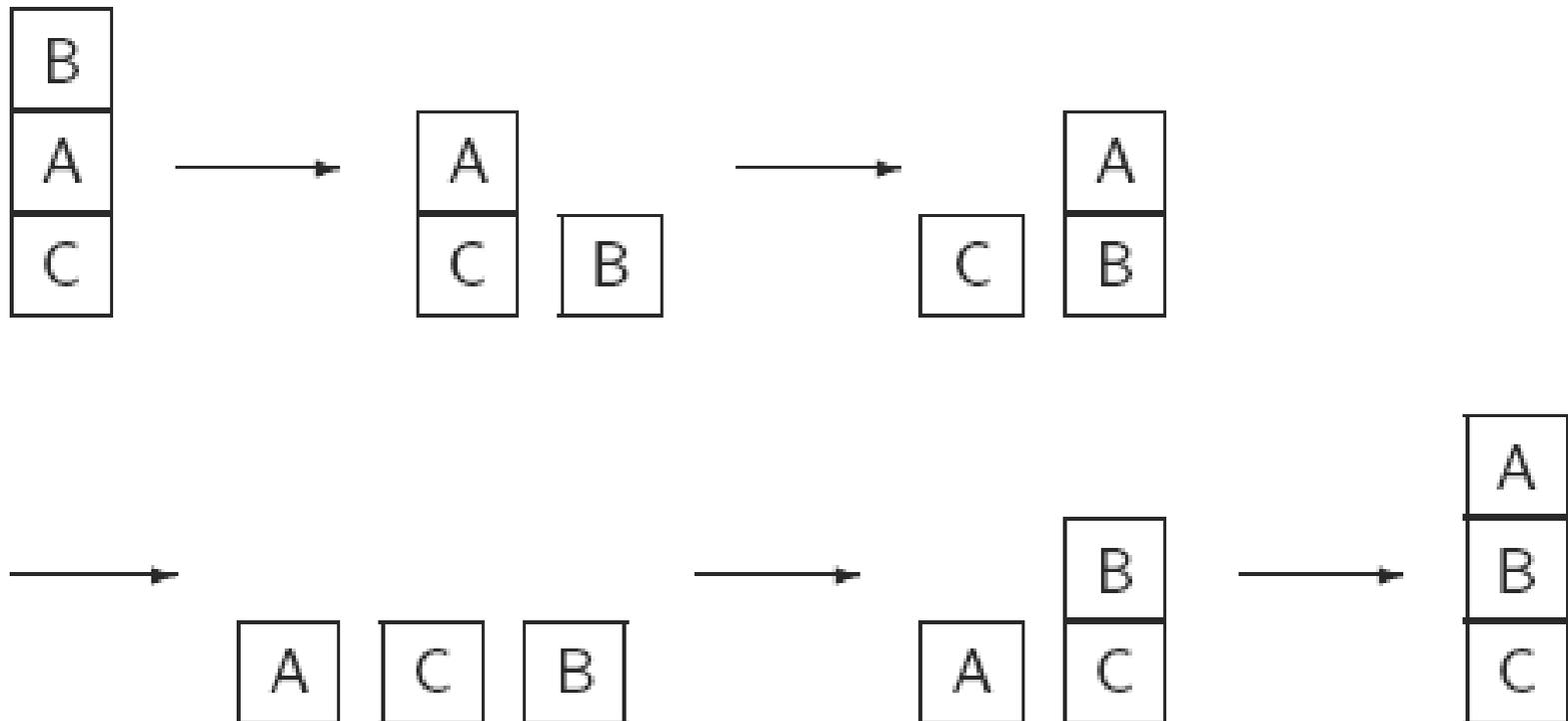


STRIPS - R-STRIPS Algorithmus

Erstes Teilziel : $ON(A, B)$

Erster Operator : $MOVE(A, C, B)$

R-STRIPS \rightarrow $[UNSTACK(B, A), MOVE(A, C, B), UNSTACK(A, B),$
 $STACK(B, C), STACK(A, B)]$



STRIPS - Behandlung des Rahmenproblems

STRIPS - Sussman-Anomalie



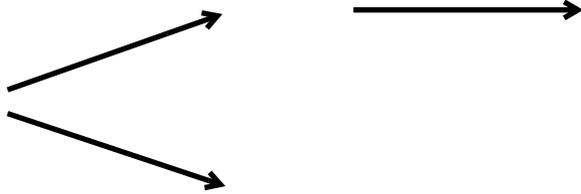
Start- und Zielzustand seien wie oben angegeben.

- 1. Teilziel $ON(A,B)$:
 $P = [UNSTACK(C,A), STACK(A,B)];$
- 2. Teilziel $ON(B,C)$:
 $P' = [UNSTACK(A,B), \dots]$
Das erste Teilziel muss also wieder zerstört werden.

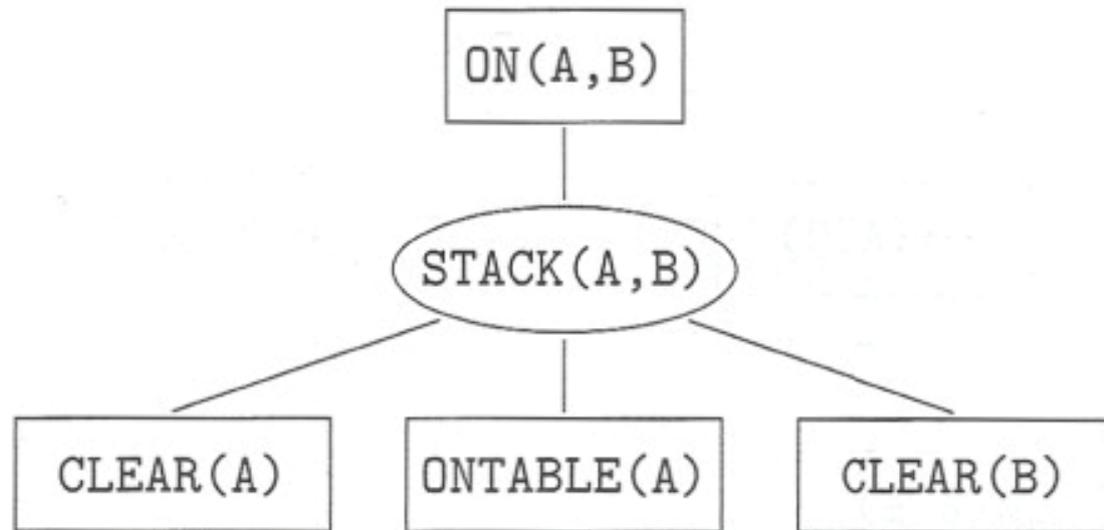
STRIPS

- 1. Teilziel $ON(B,C)$:
 $P = [STACK(B,C)]$
- 2. Teilziel $ON(A,B)$:
 $P' = [UNSTACK(B,C), \dots]$
Auch hier muss das erste Teilziel wieder zerstört werden.

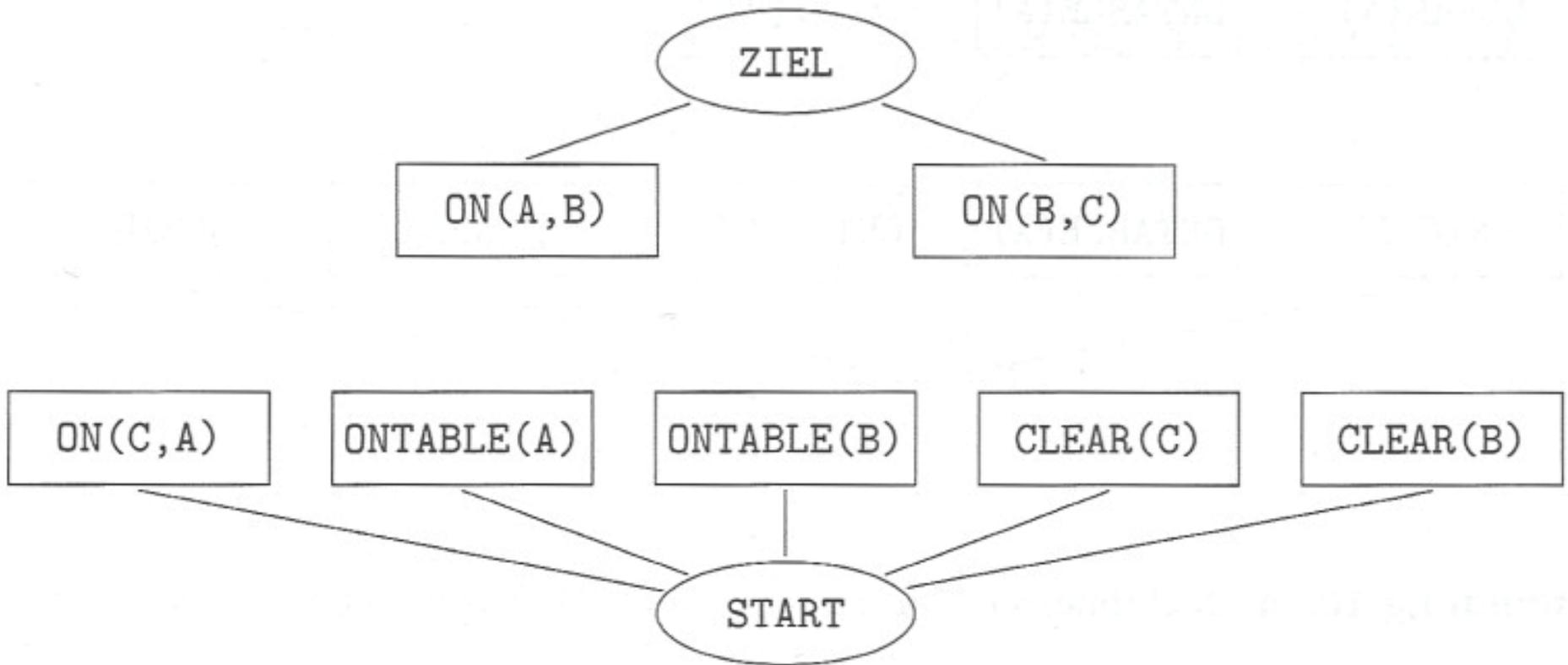
Nichtklassische Planungssysteme



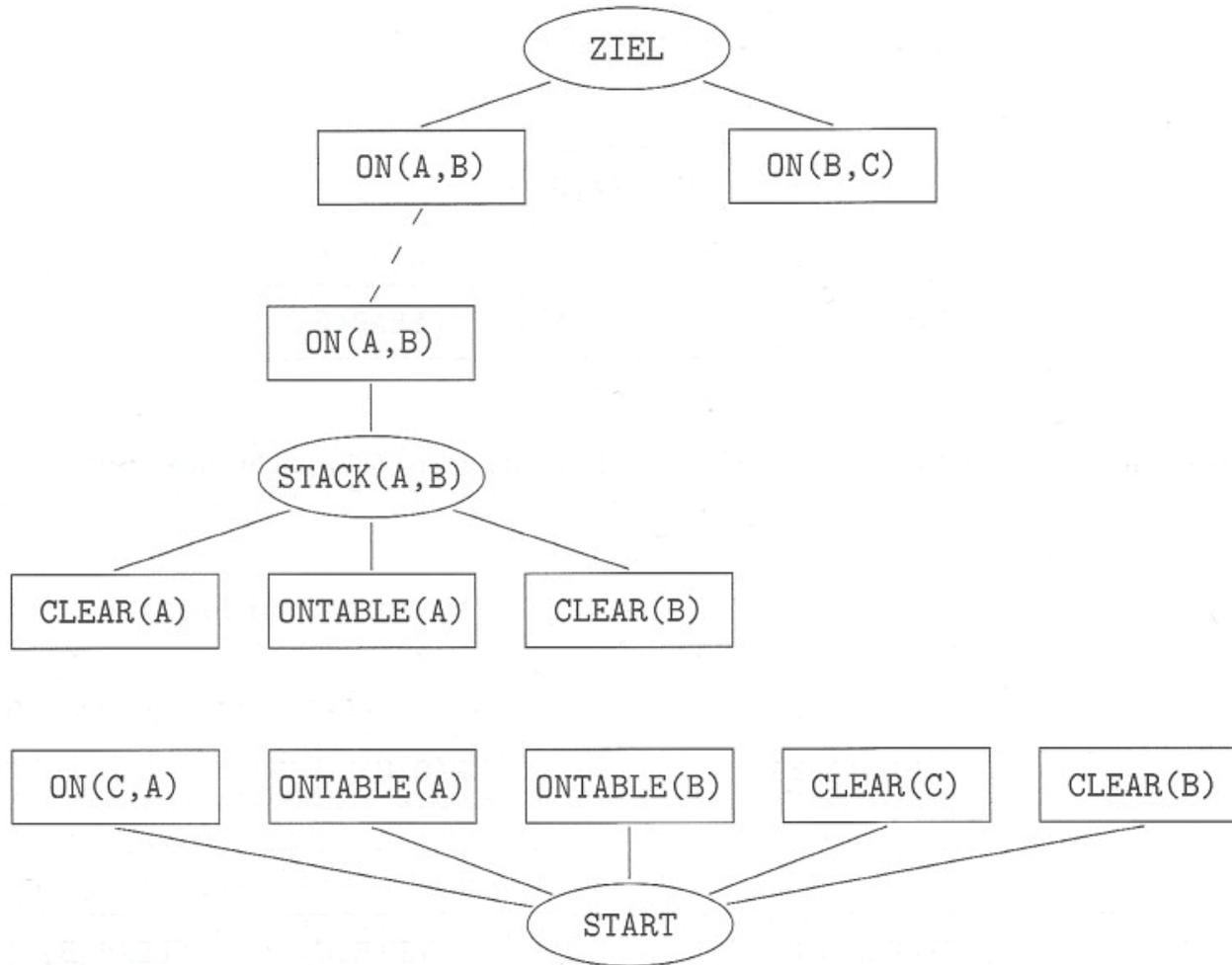
Nichtklassische Planungssysteme



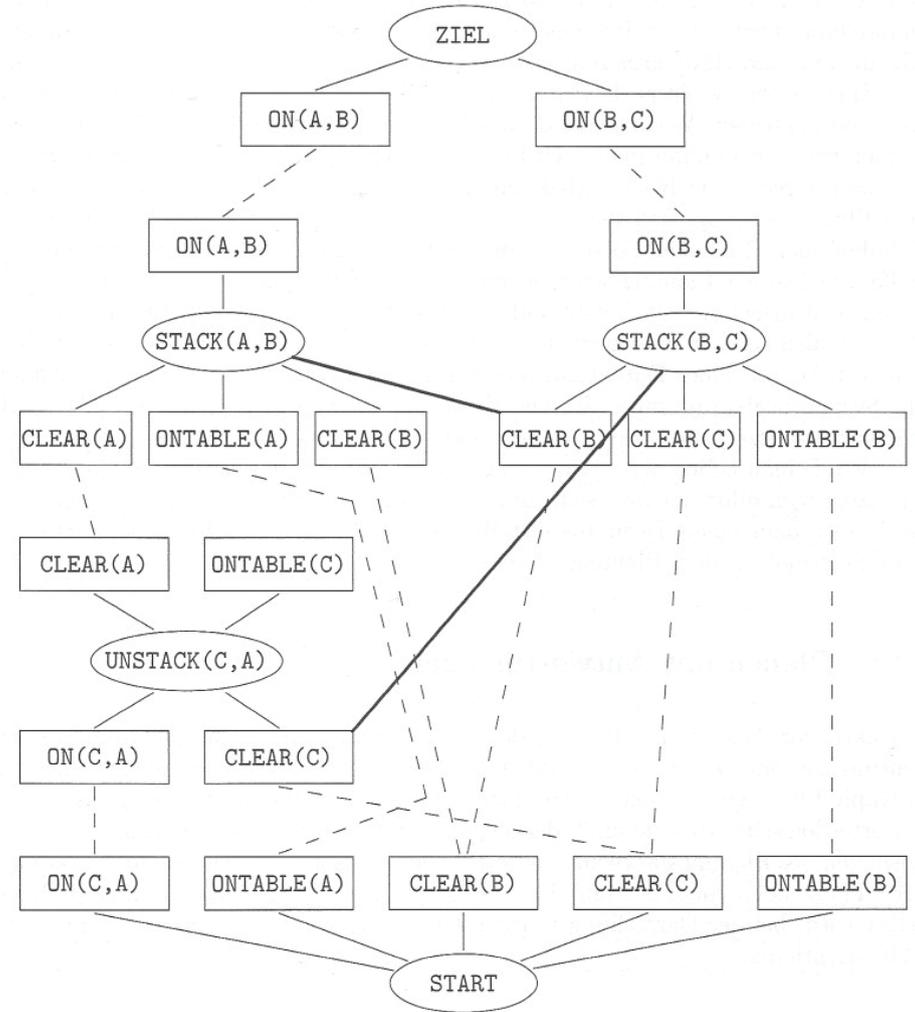
Nichtklassische Planungssysteme



Nichtklassische Planungssysteme



Nichtklassische Planungssysteme



Planen mit Antwortmengen

Planen mit Antwortmengen - SMODELS



Planen mit SMOBELS

Planen mit SMOBELS

```
:- not on(3,2,lasttime).           % Ziel:
:- not on(2,1,lasttime).           %
:- not on(1,table,lasttime).       %      3      6
:- not on(6,5,lasttime).           %      2      5
:- not on(5,4,lasttime).           %      1      4
:- not on(4,table,lasttime).       % -----
```

Planen mit SMOBELS

Planen mit SMOBELS

```
time(0..lasttime).

location(B) :- block(B).
location(table).

% GENERATE
{move(B,L,T) : block(B) : location(L)} grippers :- time(T),
                                                    T < lasttime.

% DEFINE

% Effekt, wenn ein Block bewegt wird
on(B,L,T+1) :- move(B,L,T),
               block(B), location(L), time(T), T < lasttime.

% was sich nicht verändert
on(B,L,T+1) :- on(B,L,T), not -on(B,L,T+1),
               block(B), location(L), time(T), T < lasttime.

% Eindeutigkeit des Ortes
-on(B,L1,T) :- on(B,L,T), L ≠ L1,
               block(B), location(L), location(L1), time(T).
```

Planen mit SMOBELS

```
% TEST

% zwei verschiedene Blöcke können nicht auf demselben Block sein
:- 2 {on(B1,B,T) : block(B1)},
    block(B), time(T).

% ein Block, der nicht frei ist, kann nicht bewegt werden
:- move(B,L,T), on(B1,B,T),
    block(B), block(B1), location(L), time(T), T < lasttime.

% ein Block kann nicht auf einen Block gestellt werden, der
% zur gleichen Zeit bewegt wird
:- move(B,B1,T), move(B1,L,T),
    block(B), block(B1), location(L), time(T), T < lasttime.

% DISPLAY

hide.

show move(B,L,T).
```

Planen mit SMOBELS

```
const grippers = 2.  
const lasttime = 3.  
  
block(1..6).  
  
% DEFINE  
  
on(1,2,0).  
on(2,table,0).  
on(3,4,0).  
on(4,table,0).  
on(5,6,0).  
on(6,table,0).  
  
% TEST  
  
:- not on(3,2,lasttime).  
:- not on(2,1,lasttime).  
:- not on(1,table,lasttime).  
:- not on(6,5,lasttime).  
:- not on(5,4,lasttime).  
:- not on(4,table,lasttime).
```

%	Initialzustand:		
%			
%			
%	1	3	5
%	2	4	6
%	-----		

%	Ziel:	
%		
%	3	6
%	2	5
%	1	4
%	-----	

Planen mit SMOBELS

Planen mit SMODELS

Ausblick und Anwendungen

Ausblick und Anwendungen
