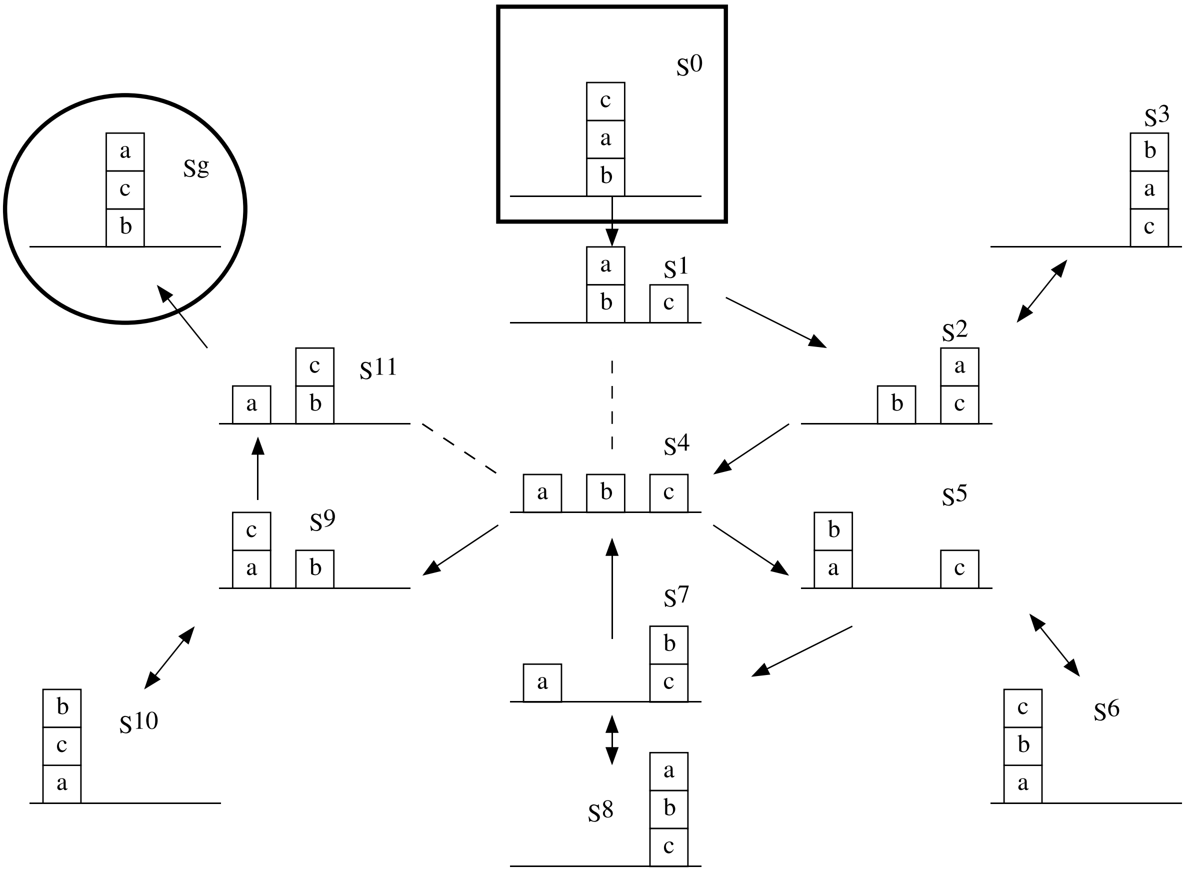

Figure 1: Depth first search.

Explorative behavior: *depth first search.*

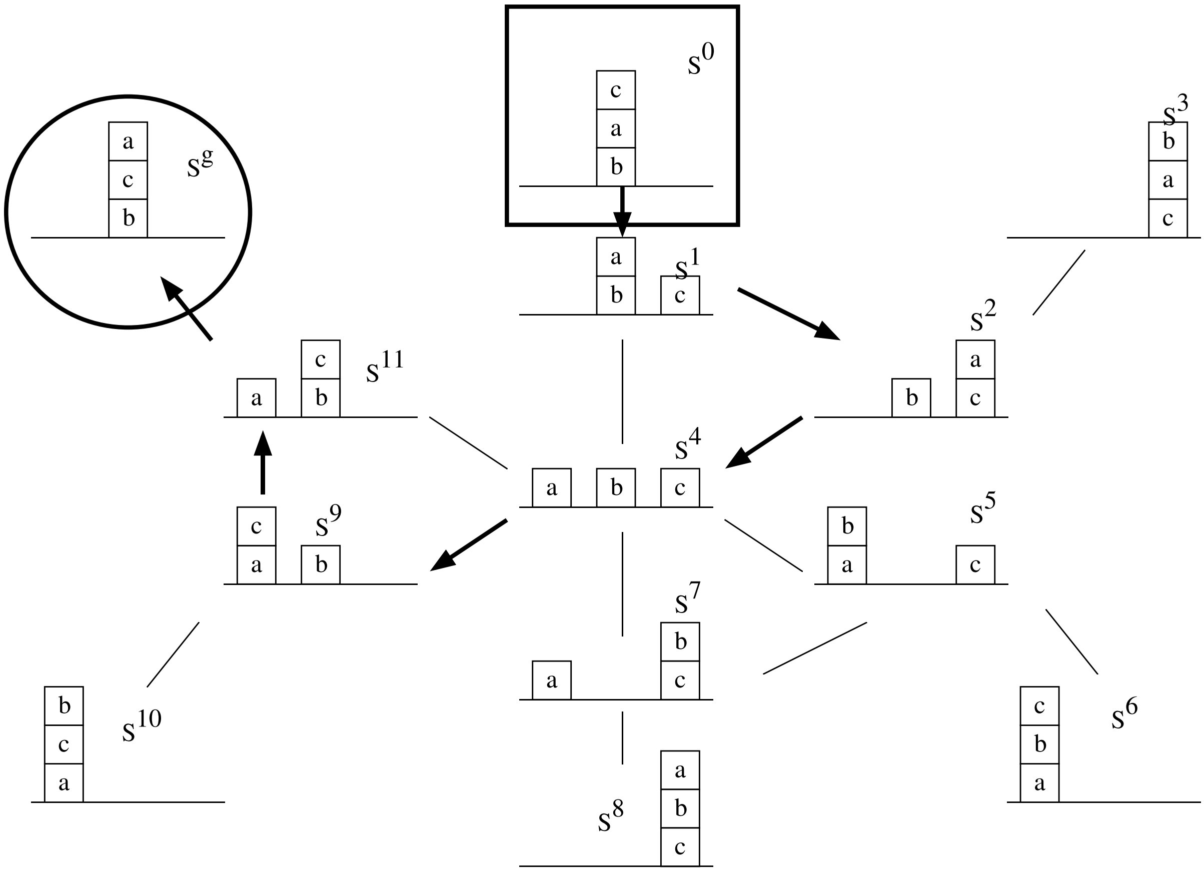

Figure 2: Depth first search and suggestion 1.

Suggestion 1:

if a block is in final position, do not move it.

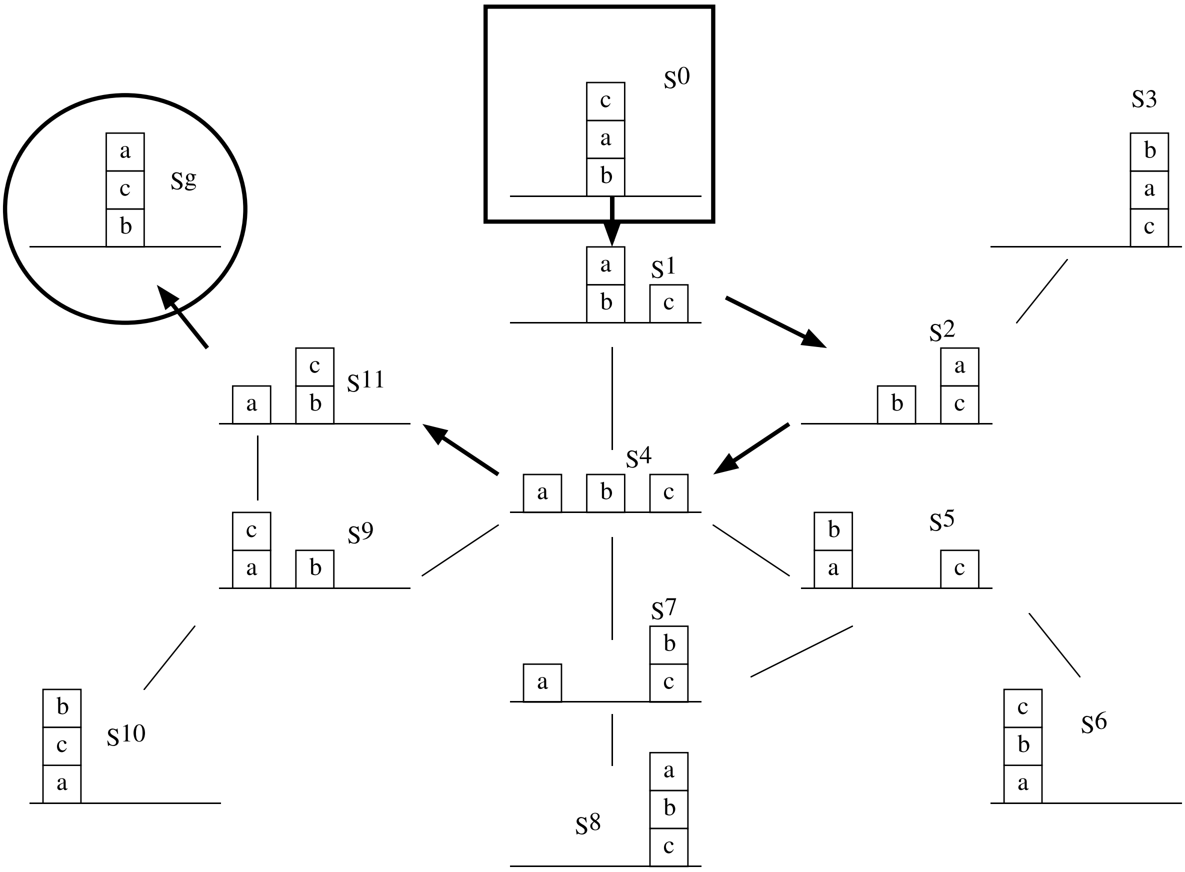

Figure 3: Depth first search, suggestion 1 and 2.

Suggestion 2:

> *if any block can be oved to final position, this should be done before any other type of move.*

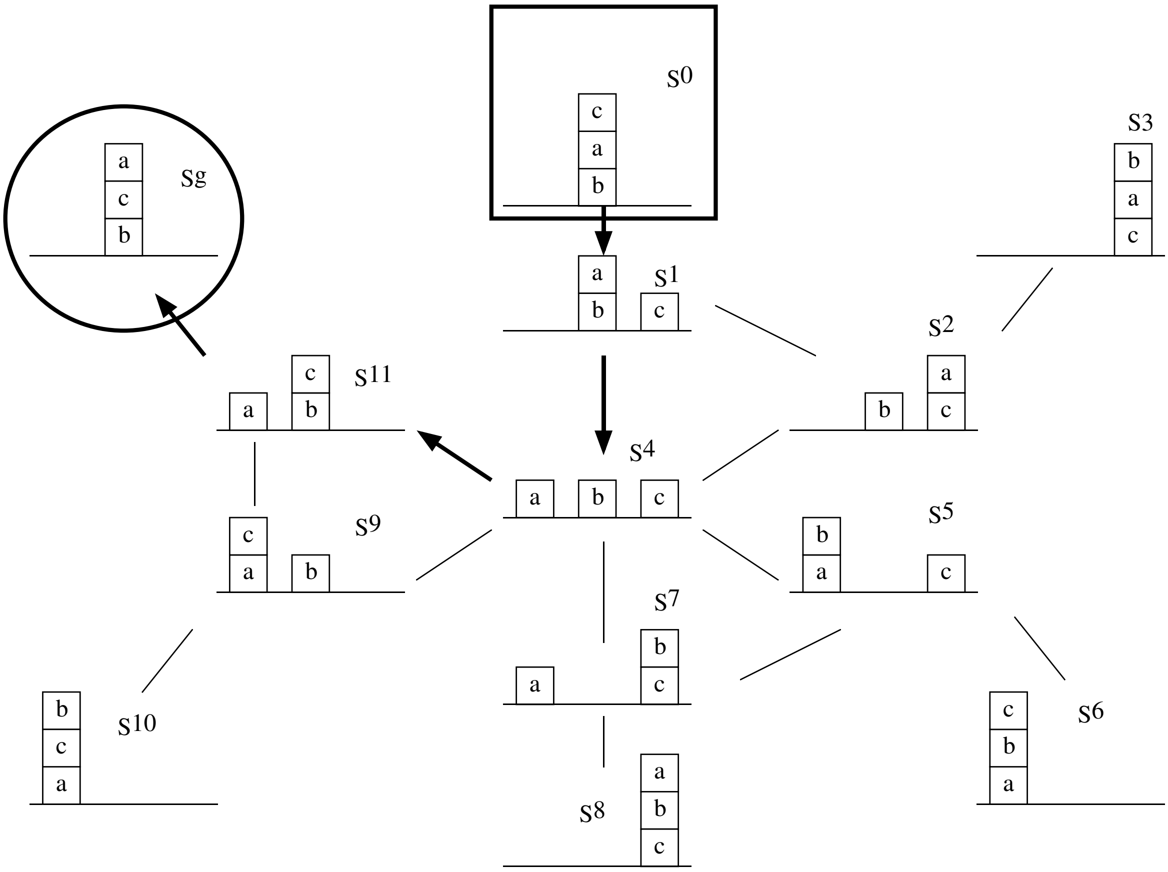

Figure 4: Depth first search, suggestion 1, 2 and 3.

Suggestion 3:

> *if there is no block that can be moved to final position, and there is a block that is above a block it ought to be above but it is not in final position, put it on the table.*

Initial configuration

(1) $on(C, A, S_0)$

(2) $on(A, B, S_0)$

(3) $on(B, Table, S_0)$

Goal configuration

(4) $on(A, C, S_g)$

(5) $on(C, B, S_g)$

(6) $on(B, Table, S_g)$

Unique names axiom

(7) $Table \neq A \wedge Table \neq B \wedge Table \neq C \wedge A \neq B \wedge A \neq C \wedge B \neq C$

Clear

$$(8) \quad x = Table \lor \neg \exists y \; on(y, x, s) \rightarrow clear(x, s)$$

Applicable action

$$(9) \quad \begin{aligned} & clear(x, s) \land clear(y, s) \land x \neq y \land x \neq Table \rightarrow \\ & \qquad applicable(move(x, y), s) \end{aligned}$$

Move action

$$(10) \quad applicable(move(x, y), s) \rightarrow on(x, y, result(move(x, y), s))$$

Frame axiom

$$(11) \quad \begin{aligned} & applicable(move(x, y), s) \land on(u, v, s) \land u \neq x \rightarrow \\ & \qquad on(u, v, result(move(x, y), s)) \end{aligned}$$

Final

$$(12) \quad on(x, Table, s) \wedge on(x, Table, s_g) \rightarrow final(x, s_g, s)$$

$$(13) \quad final(y, s_g, s) \wedge on(x, y, s) \wedge on(x, y, s_g) \rightarrow final(x, s_g, s)$$

Above

$$(14) \quad on(x, y, s) \rightarrow above(x, y, s)$$

$$(15) \quad on(x, z, s) \wedge above(z, y, s) \rightarrow above(x, y, s)$$

Under

$$(16) \quad on(x, y, s) \rightarrow under(y, x, s)$$

$$(17) \quad on(x, z, s) \wedge under(y, z, s) \rightarrow under(y, x, s)$$

Action selection rules

Bad move 1: *If a block is in final position, do not move it.*

$$final(x, s_g, s) \rightarrow bad(move(x, y), s, s_g) \tag{18}$$

Safe move 1: *If any block can be moved to final position, this should be done before any other type of move.*

$$\begin{aligned} &\neg final(x, s_g, s) \wedge final(x, s_g, result(move(x, y), s)) \rightarrow \\ &\qquad good(move(x, y), s, s_g) \end{aligned} \tag{19}$$

Safe move 2: *If there is no block that can be moved to final position, and there is a block that is above a block it ought to be above but it is not in final position, put it on the table.*

$$\begin{aligned} &\neg \exists w (final(w, s_g, result(move(w, z), s)) \wedge \neg final(w, s_g, s)) \wedge \\ &\qquad \neg final(x, s_g, s) \wedge above(x, y, s) \wedge above(x, y, s_g) \rightarrow \\ &\qquad\qquad good(move(x, Table), s, s_g) \end{aligned} \tag{20}$$

Depth First Search

$$\exists s_i(\forall x \forall y(on(x,y,s) \leftrightarrow on(x,y,s_i)) \wedge selected(x_1,s_i)) \rightarrow \\ bad(x_1,s,s_g) \tag{21}$$

$$\exists s_i \forall x \forall y(on(x,y,s_i) \leftrightarrow on(x,y,result(x_2,s)) \wedge \\ \neg \exists s_i \forall x \forall y(on(x,y,s_i) \leftrightarrow on(x,y,result(x_1,s)) \rightarrow \\ better(x_1,x_2,s,s_g) \tag{22}$$

$$\exists s_i \exists x_i \exists s_k(\forall x \forall y(on(x,y,result(x_2,s)) \leftrightarrow on(x,y,s_i)) \wedge \\ result(x_i,s_i) = s_k \wedge \forall x \forall y(on(x,y,s) \leftrightarrow on(x,y,s_k))) \wedge \\ \neg \exists s_i \exists x_i \exists s_k(\forall x \forall y(on(x,y,result(x_1,s)) \leftrightarrow on(x,y,s_i)) \wedge \\ result(x_i,s_i) = s_k \wedge \forall x \forall y(on(x,y,s) \leftrightarrow on(x,y,s_k))) \rightarrow \\ better(x_1,x_2,s,s_g) \tag{23}$$

1 Reasoning

Meta-Rules

$$x \neq y \wedge good(x, s, sg) \rightarrow bad(y, s, sg) \tag{24}$$

$$bad(x, s, sg) \rightarrow \neg good(x, s, sg) \tag{25}$$

$$\neg \forall x bad(x, s, sg) \tag{26}$$

$$x \neq y \wedge better(x, y, s, sg) \rightarrow bad(y, s, sg) \tag{27}$$

Consistency: Find an action that cannot be proved to be bad (see meta-rules) for the current goal and situation.

2 Reasoning Strategies

Express reasoning strategies, proof schemas.

Knowledge handling rules: three main components.

1. the class of facts or goal the rule can be used to know about
2. the inference operator that should be applied
3. the class of facts that act as hypotheses for the inference

Can be composed like mathematical functions, and combined procedurally like LISP programs.

1. $ON\text{-}NOW(S) \equiv CIRC(on(x,y,S) : NOW,\ \ 7)$

2. $ON\text{-}GOAL(S_g) \equiv CIRC(on(x,y,S_g) : GOAL,\ \ 7)$

$\boxed{\texttt{3.}}$ $DED(\forall x(on(x,y,S) \leftrightarrow on(x,y,S_g)) :\ ON\text{-}NOW(S),\ ON\text{-}GOAL(S_g))$

4. $APPLICABLE(S) \equiv CIRC(applicable(x,S) :\ ON\text{-}NOW(S),\ \ 8,\ \ 9,\ \ 7)$

$\boxed{\texttt{5.}}$ $DED(applicable(x,S) \wedge \forall y(applicable(y,S) \rightarrow y = x) :$
$APPLICABLE(S))$

6. $NEXT(S,X) \equiv CIRC(on(z,y,result(X,S)) :$
$APPLICABLE(S),\ ON\text{-}NOW(S),\ \ 10,\ \ 11,\ \ 7)$

7. $FINAL\text{-}NOW(S,Sg) \equiv CIRC(final(x,S_g,S) :\ ON\text{-}NOW(S),$
$ON\text{-}GOAL(S_g),\ \ 7,\ \ 12,\ \ 13)$

8. $ON\text{-}NEXT(S) \equiv CIRC(on(z,y,result(x,S)) :\ APPLICABLE(S),$
$ON\text{-}NOW(S),\ \ 10,\ \ 11,\ \ 7)$

9. $FINAL\text{-}NEXT(S,S_g) \equiv CIRC(final(z,S_g,result(x,S)) :\ ON\text{-}NEXT(S),$
$ON\text{-}GOAL(S_g),\ \ 7,\ \ 12,\ \ 13)$

$\boxed{\texttt{10.}}$ $DED(good(x,S,S_g) :\ FINAL\text{-}NOW(S,S_g),\ FINAL\text{-}NEXT(S,S_g),$
$ON\text{-}NOW(S),\ \ 19,\ \ 14,\ \ 15,\ \ 20,\ \ 16,\ \ 17,\ \ \mathbf{??})$

11. $ON\text{-}PAST(S) \equiv CIRC(on(x,y,s) :\ PAST(S))$

12. $SELECTED(S) \equiv CIRC(selected(x,s) :\ PAST(S))$

13. $BADS(S,S_g) \equiv CIRC(bad(x,S,S_g) :\ ON\text{-}PAST(S),\ ON\text{-}NOW(S),$
$ON\text{-}NEXT(S),\ FINAL\text{-}NOW(S,S_g),\ FINAL\text{-}NEXT(S,S_g),$
$SELECTED(S),\ \ 18,\ \ \mathbf{??},\ \ \mathbf{??},\ \ \mathbf{??},\ \ 21,\ \ 22,\ \ 23,\ \ 27,\ \ 7)$

$\boxed{\texttt{14.}}$ $DED(\neg bad(x,S,S_g) :\ BADS(S,S_g))$

Logical Spreadsheet Knowledge handling rules could be processed by a logical spreadsheet

- two dimensional table = (formula schemas $\times$ instances)
- formula schemas: axioms, intermediate results and conclussions
- intermediate results, expressed as knowledge handling rules, should be obtained by composition and combination of inference operations
- should allow building up proofs in a easy way
  - outlining their main steps and refining them progressively (top down)
  - building proofs out of simple initial results by inference composition and combination (bottom up)
- should access different theorem provers combining their functionalities

Elaboration: move tower

Tower (composite object)

(28) $tower(x_1, \ldots, x_n, s) \leftrightarrow \forall i(1 \leq i < n \rightarrow on(x_{i+1}, x_i, s))$

Note: block is a particular case.

On

(29) $on(t_1, t_2, s) \leftrightarrow on(bottom(t_1), top(t_2), s)$

Note: for blocks $bottom(x) = top(x) = x$

Top

(30) $tower(x_1, \ldots, x_n, s) \rightarrow x_n = top(x_1, \ldots, x_n, s)$

Bottom

(31) $tower(x_1, \ldots, x_n, s) \rightarrow x_1 = bottom(x_1, \ldots, x_n, s)$

Postponability as problem simplification

(32) $$postponable(g,\ problem) \rightarrow improves(problem,\ adjoin(problem - g,\ g)$$

Postponable move 1: *If a block is in final position, do not move it.*

Postponable move 2: *If a block is in final position, one need not think about putting anything on that block until it can be put in final position.*

Postponable move 2: *If a block is on the table but not in final position, one need not think about moving or putting anything on it until it can be put in final position.*

Simplification 1: *Remove all finished towers of blocks in final position.*

Simplification 2: *Remove all clear blocks on the table which are not in final position.*

Simplification 3: *Replace any unfinished tower of blocks in final position by a single block on the table and in final position.*

Simplification 1: *Replace any partially finished tower by a single block.*

Partially finished tower

$$(33)\quad partially - finished - tower(x_1, \ldots, x_n, s, sg) \leftrightarrow \\ \forall i(1 \leq i < n \rightarrow on(x_{i+1}, x_i, s) \wedge on(x_{i+1}, x_i, sg))$$

Safe moves

We need at most two moves to put a block in final position.

1. Clear final position by moving tower above it to the table.

2. Put block in final position by moving tower above it (including it) to final position.

Strategy: build towers bottom-up.

Can be improved:

1. some blocks can be put in final position by a single move

2. some blocks cannot be put in final position without their final position being cleared up