

Designing Scaffolding for Theorem Proving

Noboru Matsuda
University of Pittsburgh
May 23, 2003

The assumption is that even though expert problem-solving does not follow a possible cognitive model of strategic search shown below, providing training on backward and forward chaining based upon the cognitive model with thorough scaffolding eventually lead students to a mastery level. We do not exactly know when and how their behavior switches from a rigid rule-based one (shown below) to a typical human expert pattern.

Backward chaining

- *Show the student a single backward step*
- *Highlight a geometric configuration(s) relating to the backward step* [AW-P]

Select a goal

- *Provide a goal to prove*
- *Provide a list of goals at the fringe of backward chain*
- *Prompt the student to select a goal*

Recognize goals that have not yet achieved

- *Provide a list of goals in the goal stack, or highlight unachieved goals in the goal tree*
- *Prompt the student to recognize unachieved goals*

Select a goal to achieve first

- *Provide a goal from the goal stack*
- *Prompt the student to select a goal in the goal stack to achieve first*

Goal test (Examine if the goal has been asserted as a fact)

- *Carry out a goal test (i.e., tell the student if the goal has been asserted or not)*
- *Ask the student if the goal has been asserted or not*

- *Prompt the student to verify the goal*

Apply a rule

- *Show the student how to apply a rule*
- *Show the student a result of backward reasoning (i.e., subgoals) and ask which rule has been applied.*
- *Prompt the student to apply a rule backwards*

Select a rule

- *Provide a rule to apply backwards* [AW-E]
- *Provide a list of applicable rules*
- *Prompt the student to select a rule*

Recognize rules with consequence that unify the selected goal

- *Show a rule with consequence that unify with the goal*
- *Prompt the student to select a rule with consequence that unify with the goal*

Transform postulates into operational rules

- *Show a transformation of a rule*
- *Prompt the student to transform a postulate into a rule*

Identify the consequence

- *Show the consequence of the rule*
- *Prompt the student to identify the consequence of the rule*

Unify the consequence

- *Show a unification of the consequence and the goal*
- *Prompt the student to unify the consequence with the goal*

Select a single rule to apply

- *Select a rule from a conflict set*

- *Prompt the student to select a rule from a conflict set*

Instantiate the rule

- *Provide an instantiation of the rule*
- *Prompt the student to instantiate the rule*

Identify premises

- *Provide the premises*
- *Prompt the student to identify premises*

Unify the premises

- *Provide a unification of the premises with the propositions*
- *Prompt the student to unify the premises with the propositions*

Execute the rule

- *Execute the rule selected*
- *Prompt the student to execute the rule selected*

Check goal iteration (Verify that the premise is not in the goal stack)

- *Identify goal iteration, if any.*
- *Prompt the student to identify goal iteration*

Assert the premises of the rule as subgoals

- *Assert the premises of the rule* [AP-E]
- *Prompt the student to assert the premises* [AP-W]
- *Ask the student a relationship among the elements in the premises* [AP-R]
- *Ask the student elements appeared in the premises* [AP-O]

Forward chaining

- *Show the student a single step forward*
- *Highlight a geometric configuration(s) relating to the forward step* [AW-P]

Select a rule from the postulate pool

- *Select a rule* [AC-E]
- *Provide a set of rules*
- *Prompt the student to select a rule*

Transform postulates into operational rules

- *Show a transformation of a rule*
- *Prompt the student to transform a postulate into a rule*

Instantiate the rule

- *Provide an instantiation of the rule*
- *Prompt the student to instantiate the rule*

Identify premises and consequences

- *Identify premises and consequences of the rule*
- *Prompt the student to identify premises and consequences*

Unify premises and consequences

- *Unify premises and consequences of the rule with the problem*
- *Prompt the student to unify premises and consequence with the problem*

Execute the rule

- *Execute the rule selected*
- *Prompt the student to execute the rule selected*

Verify premises

- *Verify premises by identifying corresponding propositions*
- *Prompt the student to verify premises*

Assert consequences

- *Assert consequences as the true propositions*
- *Prompt the student to assert consequence* [AC-W]
- *Ask the student a relationship among the elements in the consequence* [AC-R]
- *Ask the student the elements appeared in the consequence* [AC-O]