

Chunking

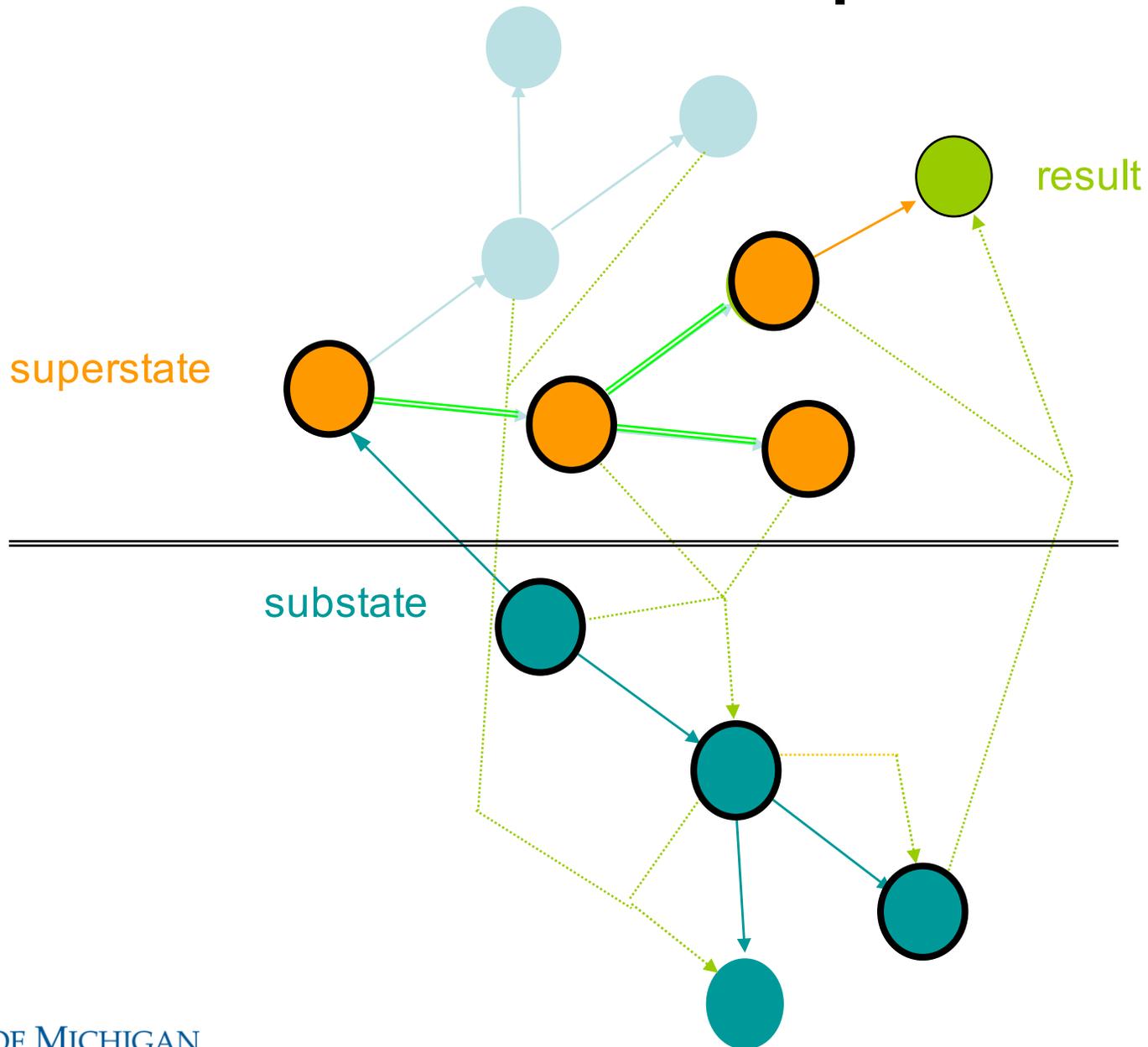
[15 min]

Soar Tutorial
July 2016

Learning/Chunking

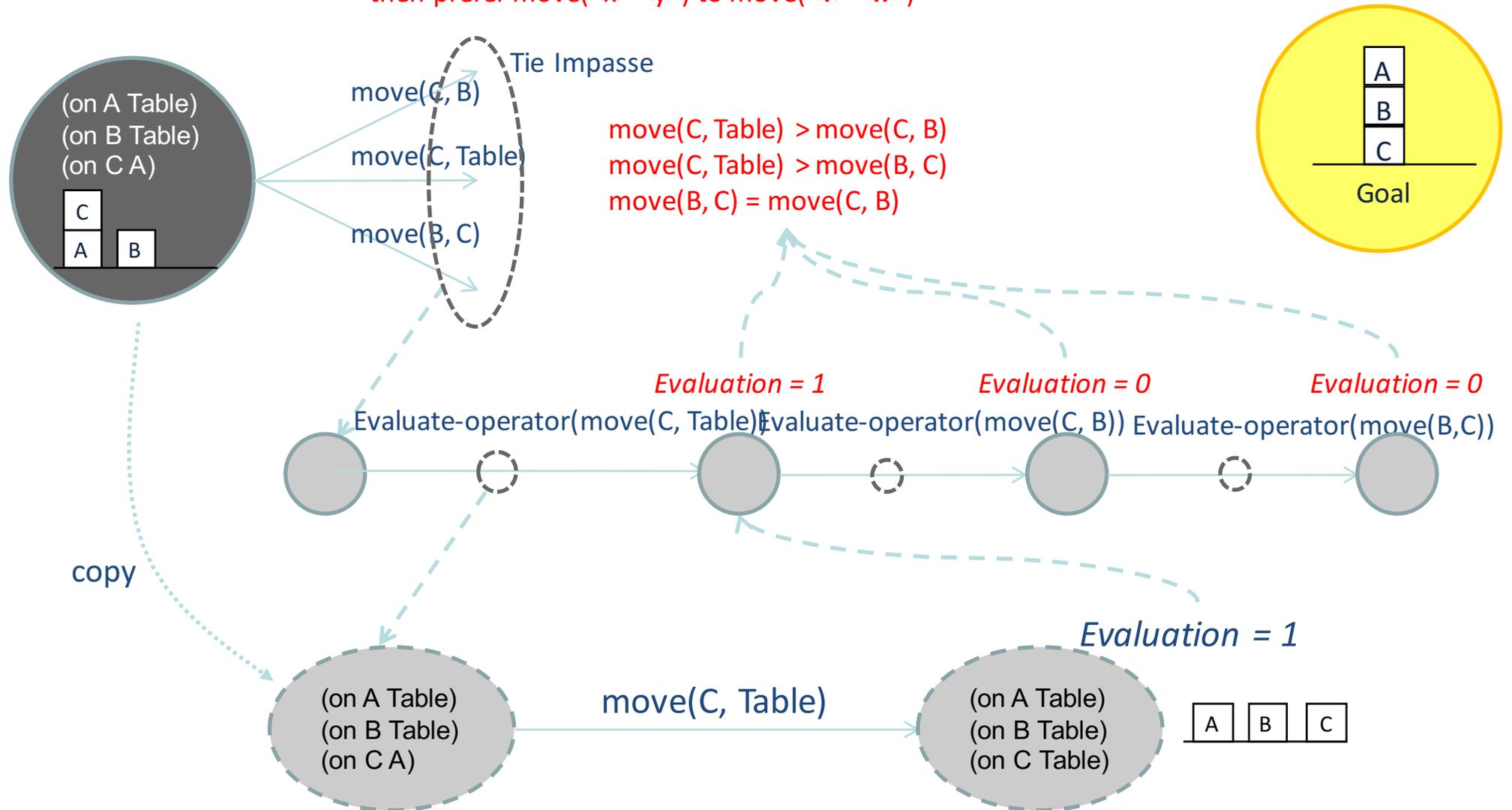
- Problem:
 - Subgoals “discover” knowledge to resolve impasses but it is lost after each problem solving episode
- Approach
 - Automatically build rules that summarize processing
 - Variablize justifications = chunks
 - Variablizes based on tests in rules that fired in substates
 - Conditions include only those test required to produce result
 - Chunks are built as soon as a result is produced
 - Immediate transfer is possible
 - One chunk for each result, where a result consists of connected WMEs that become results at the same time
 - Different results can lead to very different conditions
 - Improves generality of chunks

Chunk Example

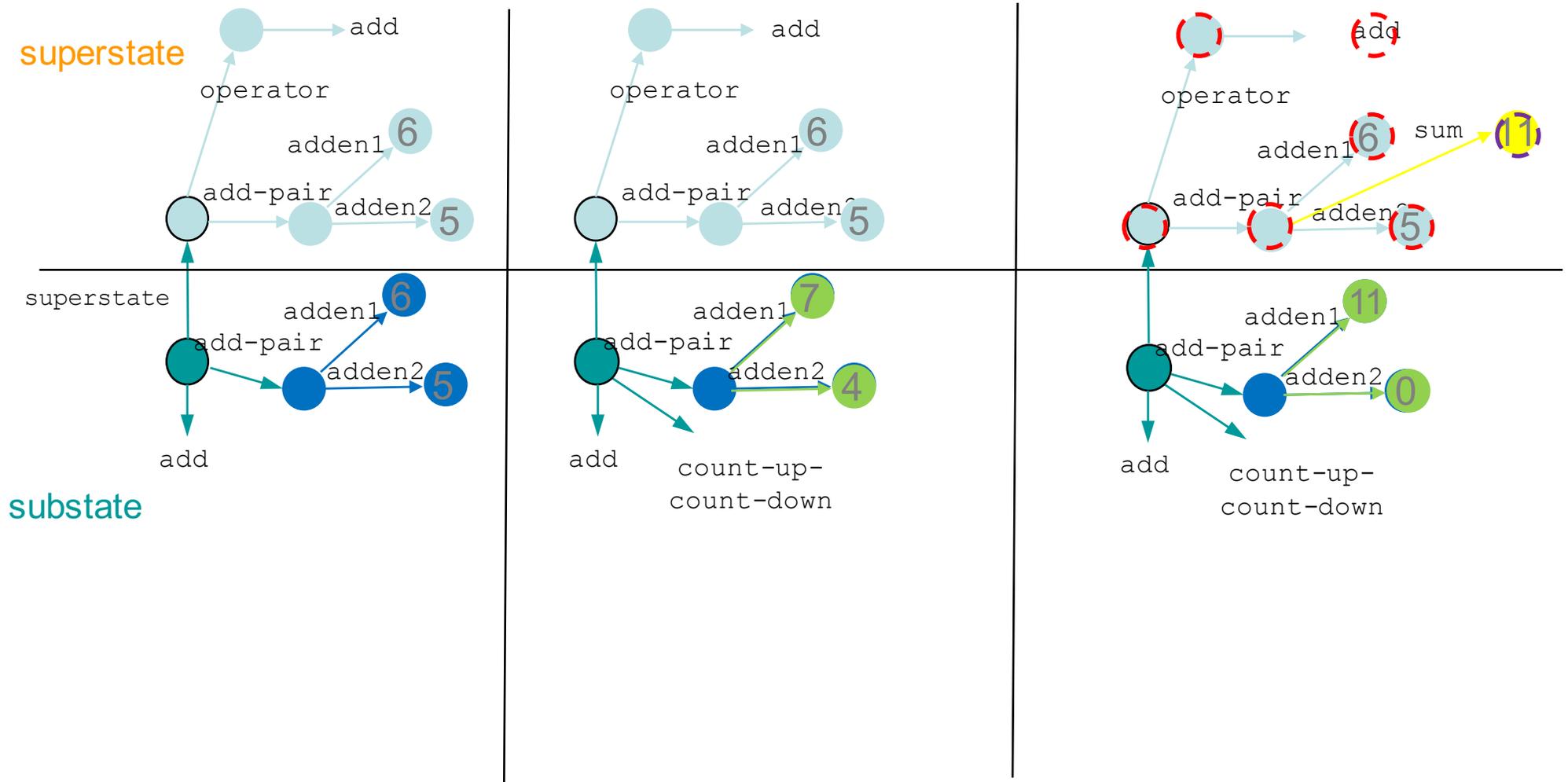


One-step Look-ahead: Learns selection rules

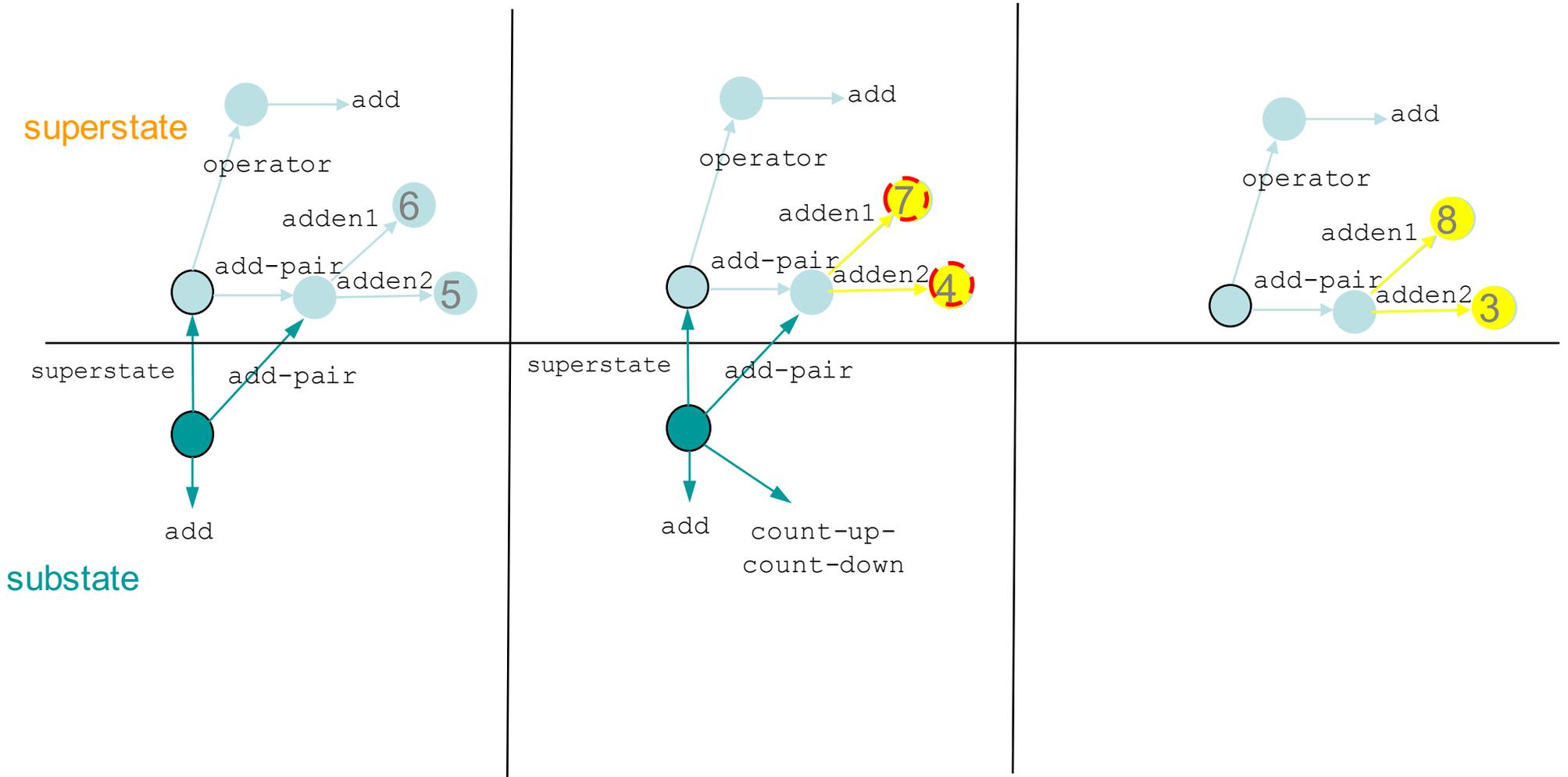
If operator $\text{move}(\langle x \rangle, \langle y \rangle)$ is proposed and $(\text{on } \langle x \rangle \langle y \rangle)$ is in the goal, and operator $\text{move}(\langle v \rangle, \langle w \rangle)$ is proposed, and $(\text{on } \langle v \rangle \langle w \rangle)$ is not in the goal, then prefer $\text{move}(\langle x \rangle, \langle y \rangle)$ to $\text{move}(\langle v \rangle, \langle w \rangle)$



Problem Solving in Substate



Problem Solving in Substate



Chunking Analysis

- Converts deliberate reasoning/planning to reaction
- Generality of learning based on generality of reasoning
 - Leads to many different types learning
 - If reasoning is inductive, so is learning
- Soar only learns what it thinks about
- All learning is impasse driven
 - Learning arises from a lack of knowledge