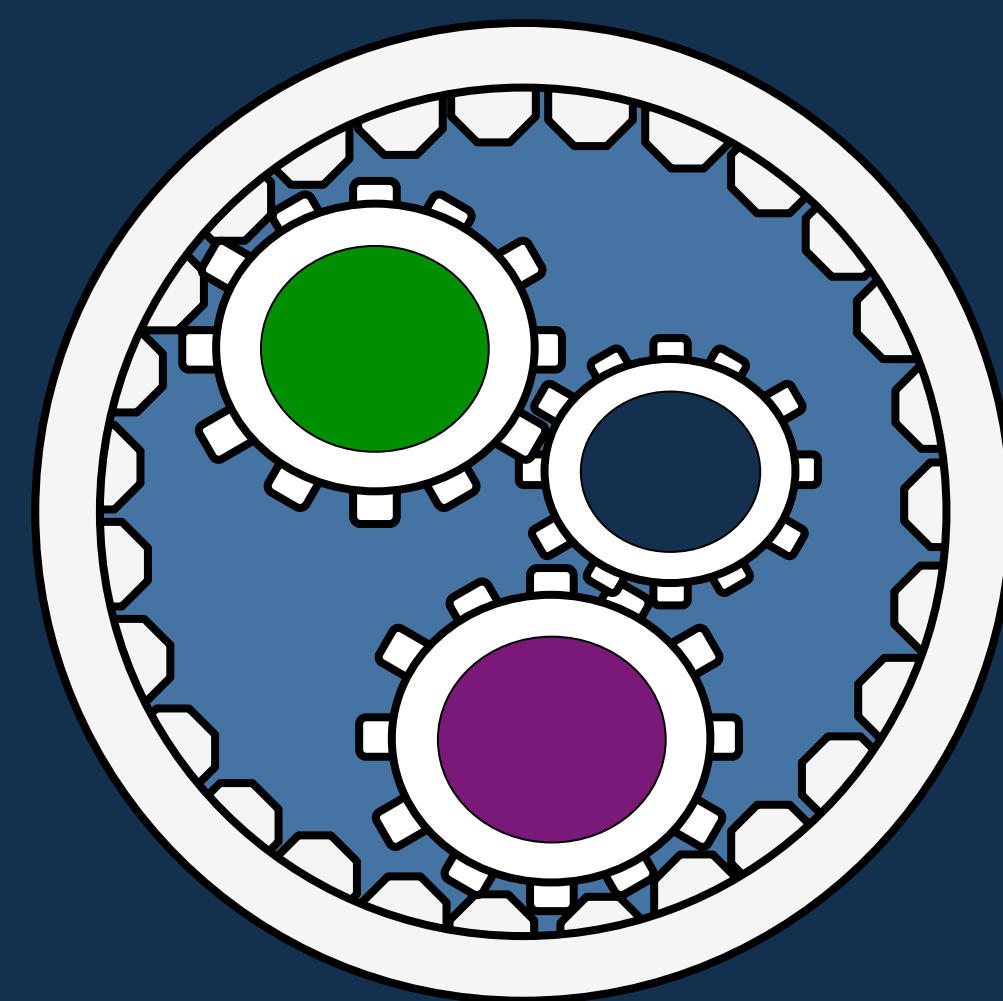
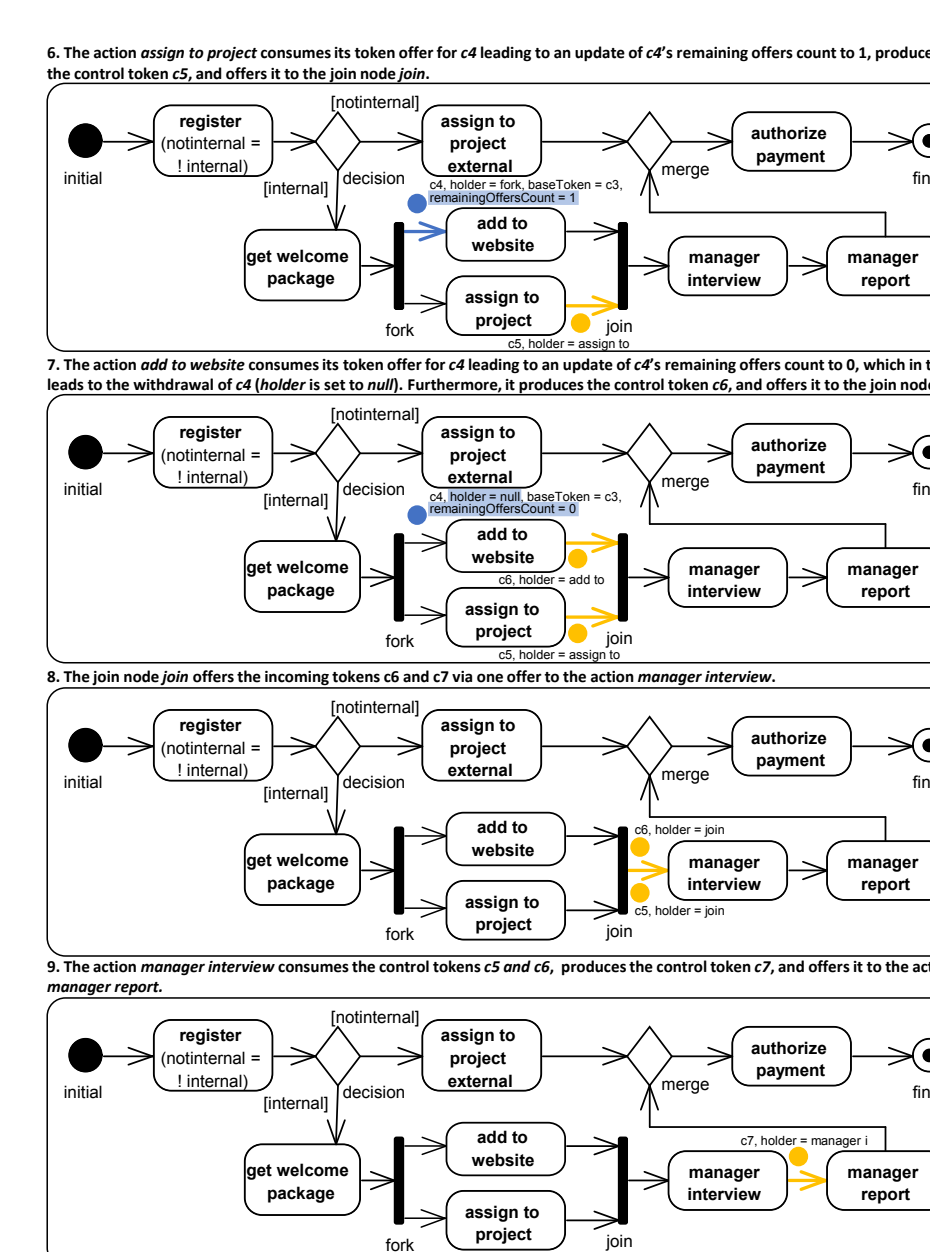
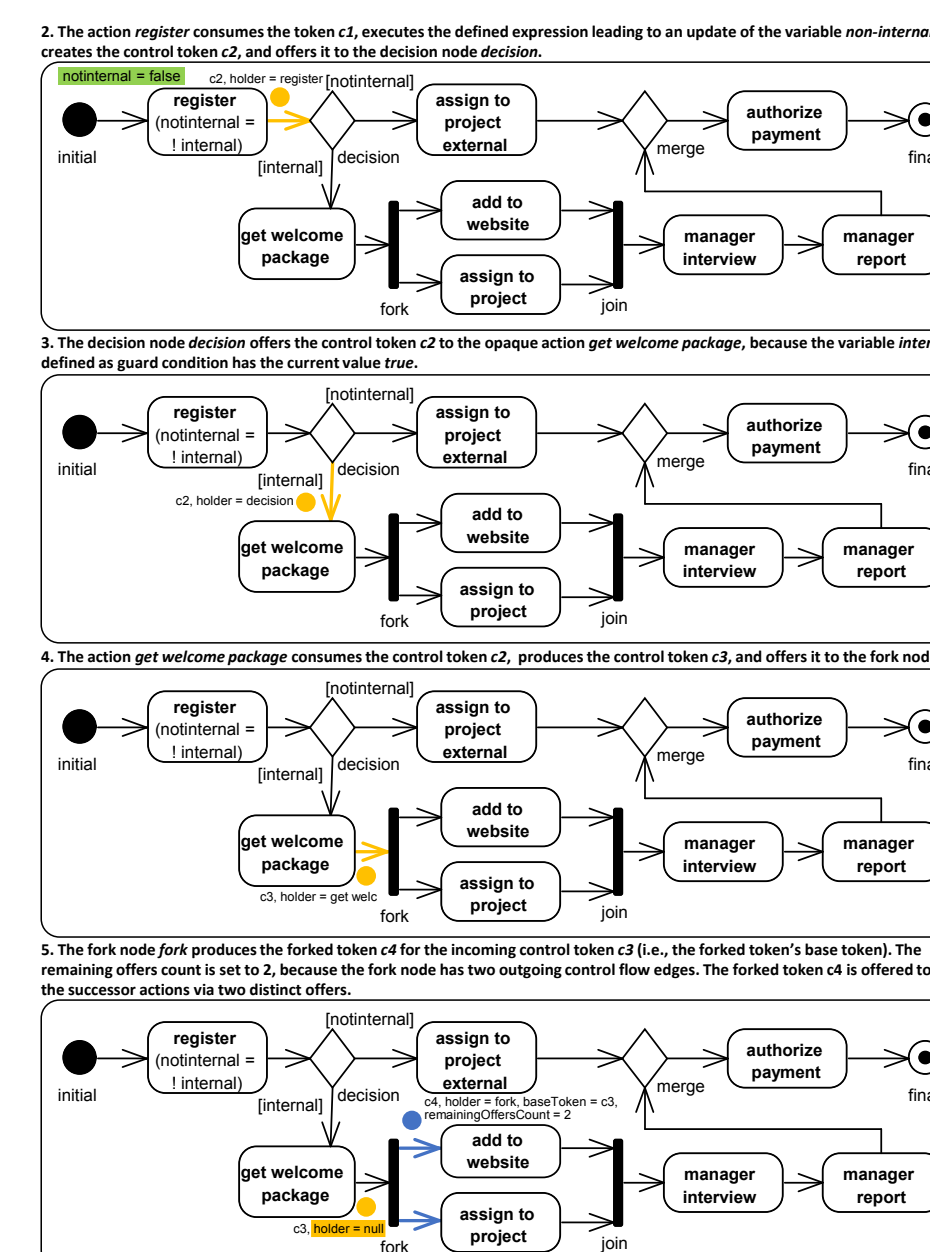
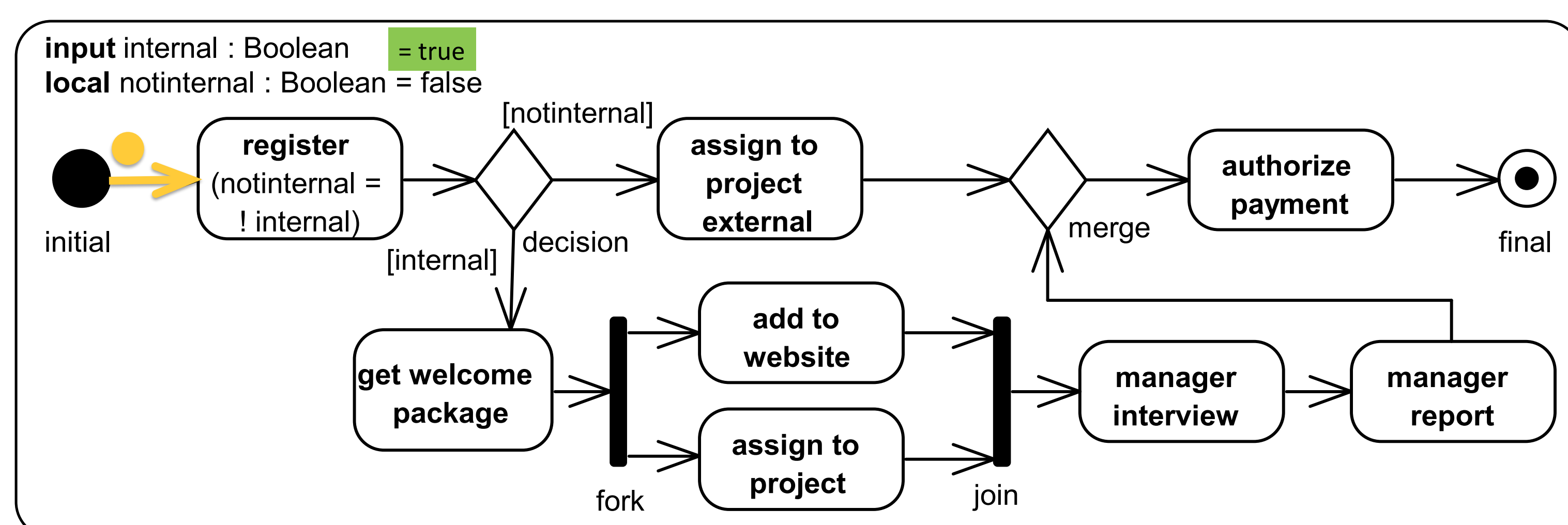


fUML Activity Diagrams with RAG-controlled Rewriting

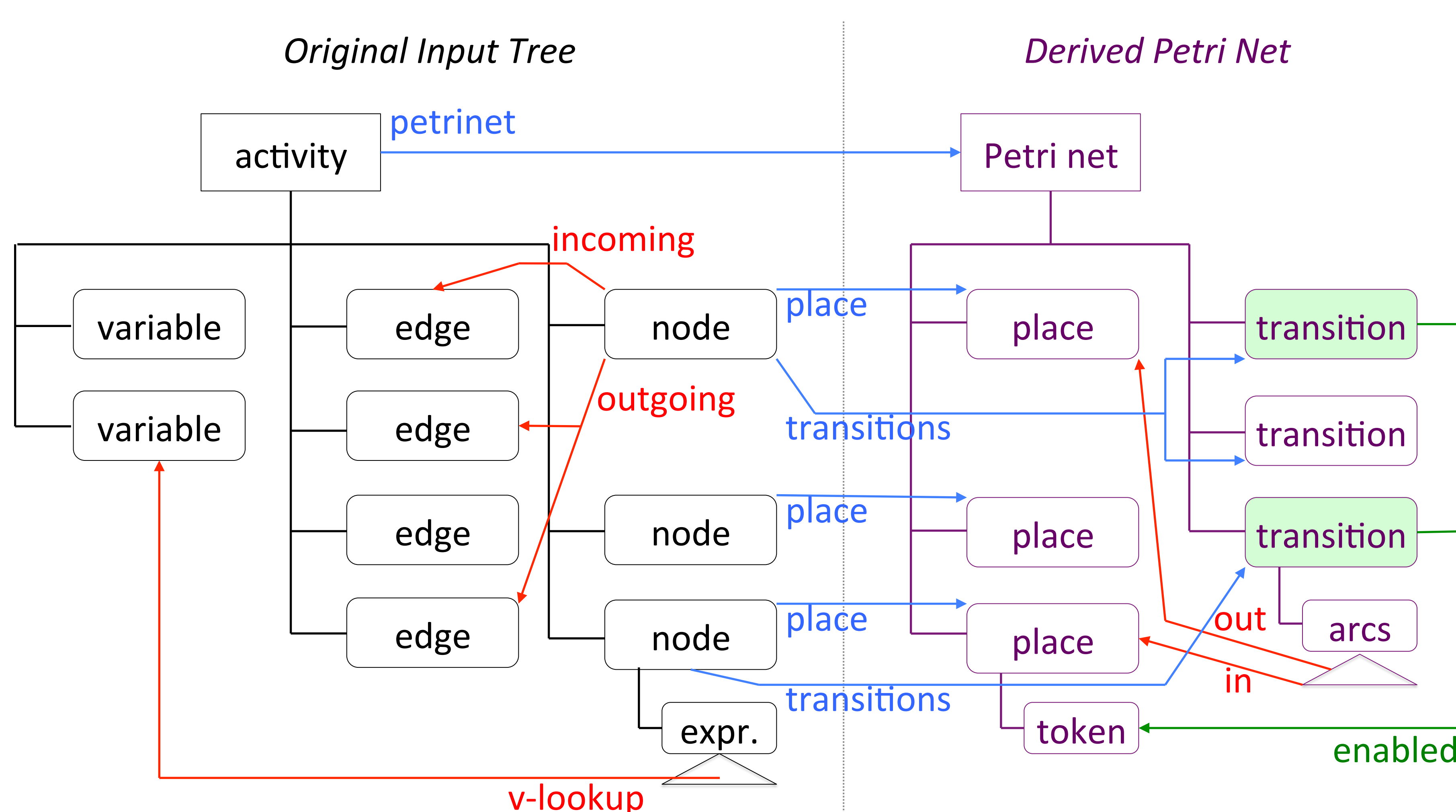


A RACR Solution of the 8th Transformation Tool Contest

TTC 2015 task: execution of fUML Activity Diagrams.



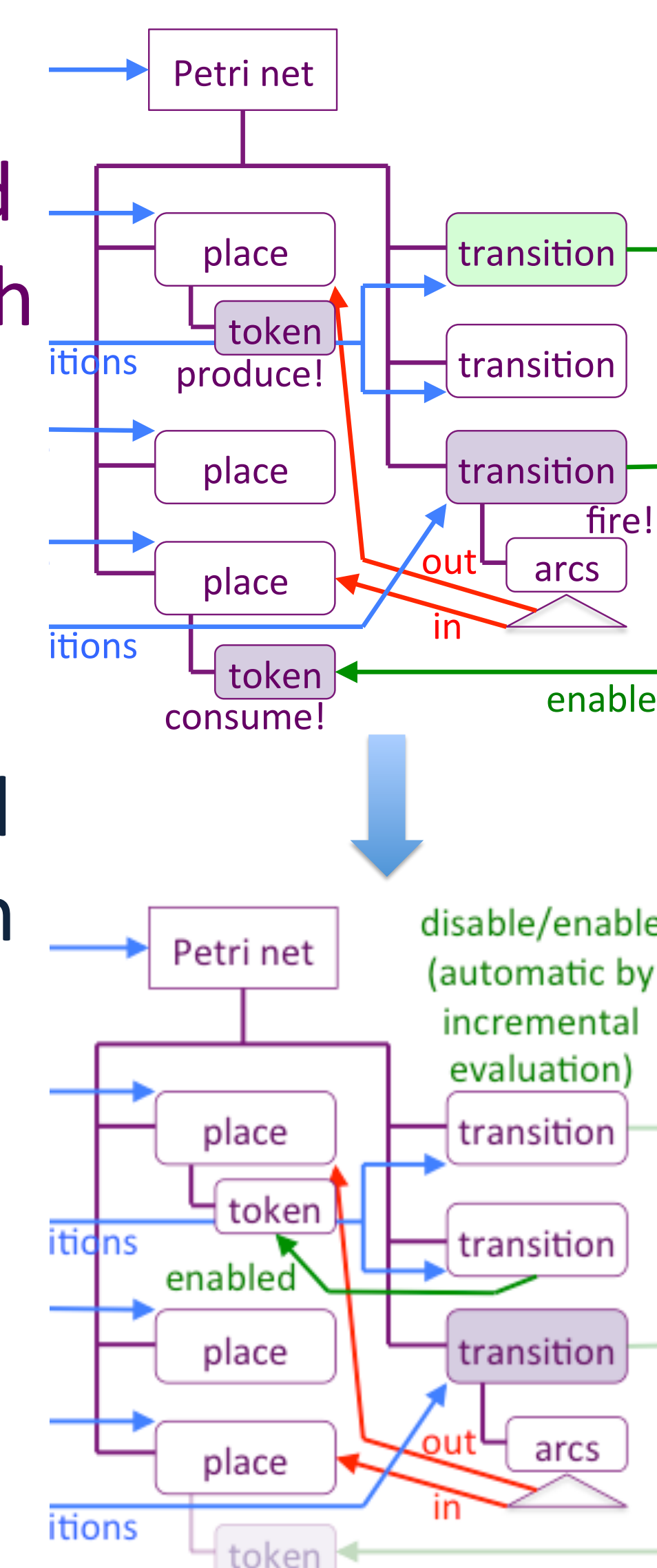
RACR Solution: diagram to Petri net interpreter, uses a reference attribute grammar to deduce memoized abstract syntax graph well-suited for execution by rewriting.



memoized, deduced abstract syntax graph

reuse of enabled analysis permits convenient, focused rewrite specification

incremental enabled analysis



semantic overlay graph (excerpt): ■ name analysis ■ code generation ■ enabled analysis

RAG-controlled rewriting: declarative, seamless combination of reference attribute grammars & graph rewriting (RACR: reference implementation, Scheme library).

- transformation-aware RAG-based analyses (incremental evaluation)
 - analyse-aware rewrite-based transformations (analyses deduce rewrites)
- mutual control

Efficient Analyses

Efficient Rewriting

Programmed /
RAG Controlled Rewriting

RACR

