
Planning, Execution & Learning: Conformant and Conditional Planning

Reid Simmons

Conformant Planning

- Create a (Non-Branching) Plan that Achieves Goals Despite Uncertainty
 - Uncertainty in initial state
 - Uncertainty in effects
- No sensing actions
- Basic Idea:
 - *Remove uncertainty by forcing world into known states*

Conformant Graphplan (Smith & Weld, 1998)

- Extends Graphplan
 - Conditional effects
 - Multiple possible worlds
 - Uncertainty represented using disjunction
- Basic Approach:
 - Create parallel plan graphs for each possible world
 - Search backwards through each plan graph in parallel
 - **Confront** undesired effects in one possible world caused by choosing action in another possible world

Handling Conditional Effects

- Break Conditional Action into Mutually Exclusive *Aspects*
 - Separate action for each combination of conditions

Dunk(?pkg)

Pre: \neg Clogged

**Eff: Clogged \wedge
In(?pkg) \rightarrow \neg Armed**

Dunk₁(?pkg)

Pre: \neg Clogged \wedge In(?pkg)

Eff: Clogged \wedge \neg Armed

Dunk₂(?pkg)

Pre: \neg Clogged \wedge \neg In(?pkg)

Eff: Clogged

Flush()

Pre:

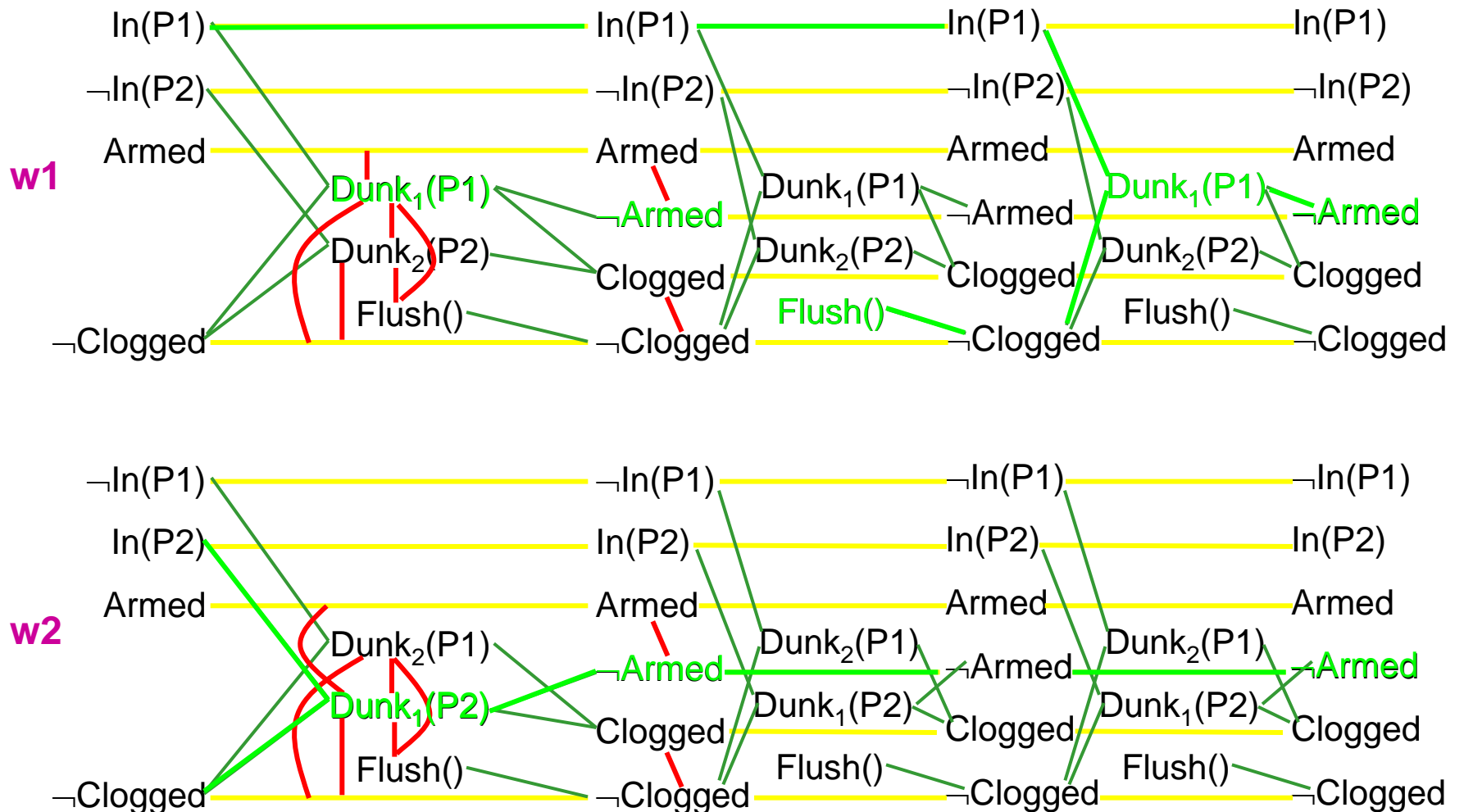
Eff: \neg Clogged

Possible Worlds Plan Graph

- Create Separate Plan Graph for Each Possible World
 - Same algorithm as regular Graphplan
 - One world for each possible initial state
 - Handle non-deterministic effects by splitting possible worlds
 - Can make more efficient by finding mutexes between worlds

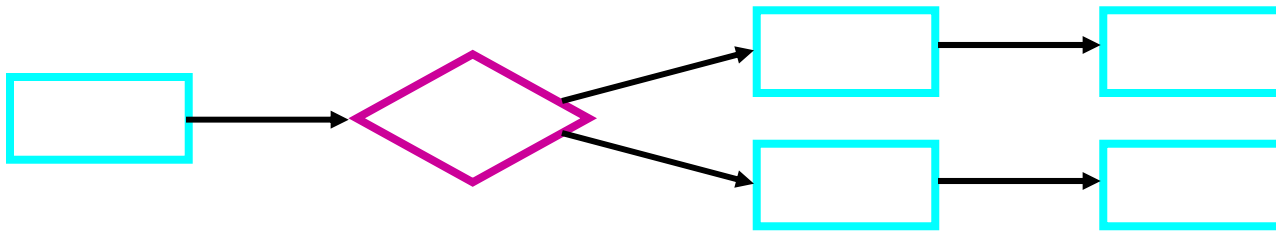
- Extract Solution for Goals in Each Possible World
 - Need to consider whether actions chosen for one possible world are mutex in any possible world
 - Can *confront* such interactions by adding negation of preconditions as subgoal

Bomb in the Toilet Domain



Conditional Planning

- Create Branching Plans
 - Take *observations* into account when selecting actions



- Observations Used to Handle Uncertainty
 - Uncertainty arises from non-deterministic actions
 - Uncertainty arises from lack of knowledge
- Planners Differ With Respect To:
 - Representation of uncertainty (logic, probabilities)
 - Representation of plans (trees, graphs)
 - Representation of observations
 - Search control

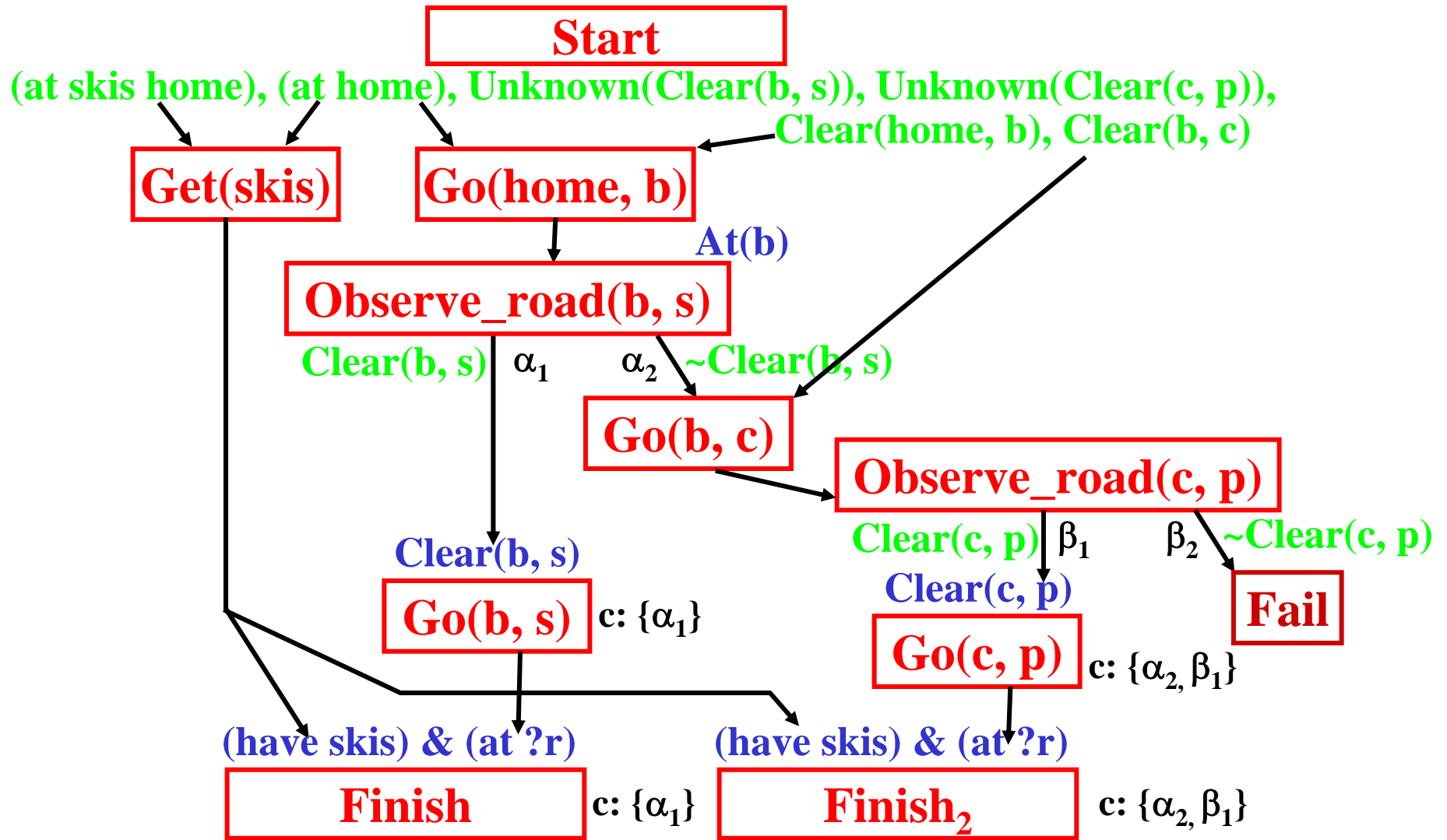
CNLP (Peot & Smith, 1992)

- Extensions to SNLP to Create Conditional Plans with Observations
- Extensions to SNLP Representation
 - Three-valued logic (True, False, *Unknown*)
 - Observation actions
Observe_Road (?loc1 ?loc2)
Pre: At(?loc1), Unknown (Clear(?loc1, ?loc2))
+ α_1 : Clear(?loc1, ?loc2)
+ α_2 : ~Clear(?loc1, ?loc2)
 - Contexts
 - Compatible observation labels

CNLP Extensions to SNLP

- “*Conditioning*”
 - Can remove threat by *separating contexts* (i.e., making them incompatible)
- Propagation of *context labels* and *reasons*
 - **Contexts**: What actions are incompatible
 - **Reasons**: what goals an action supports
- Tree-structured plan
 - Goal replication

Conditionally Planning a Ski Trip



CNLP Summary

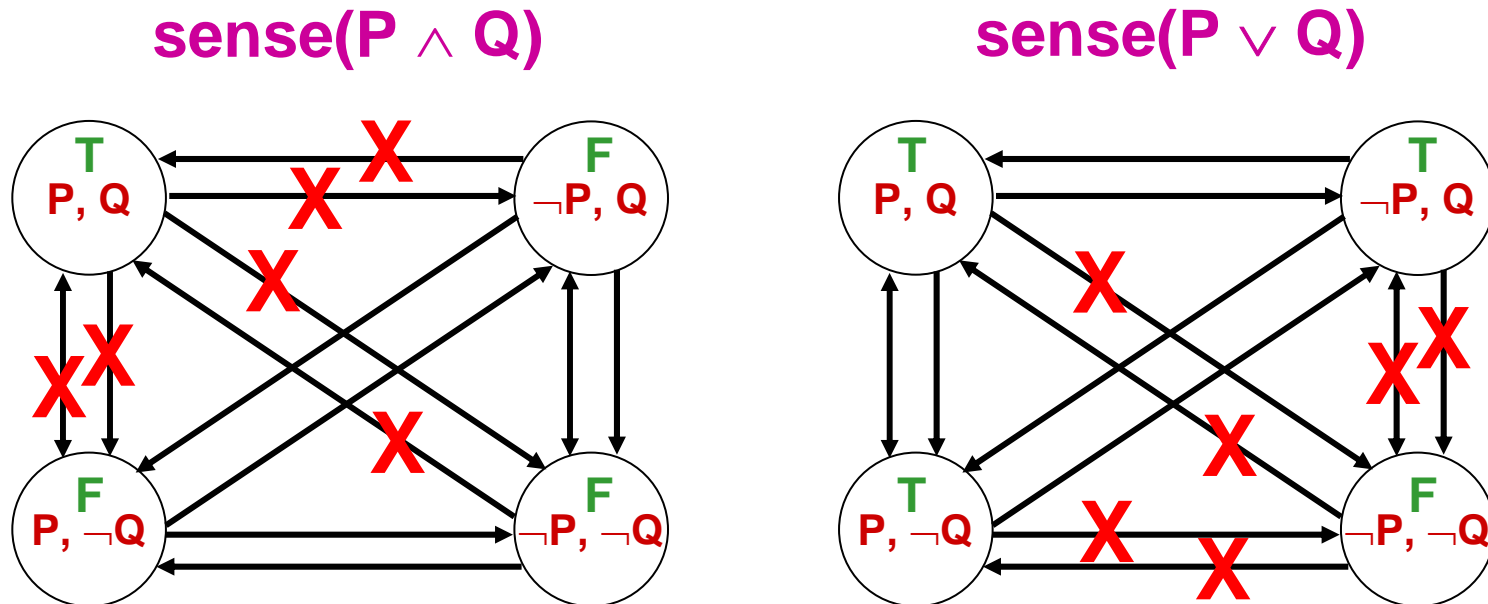
- Can Create Conditional Plans with Observation Actions
 - However, no explicit distinction between observations and causal effects
- Can Handle *Disjunctive Uncertainty*
 - No notion of which conditions more likely
 - Increases search space tremendously
- Can Plan with *Failure* as an Option

Sensory Graphplan (Weld, Anderson & Smith, 1998)

- Adds Sensing Actions to Conformant Graphplan
 - Distinguish sensory effects from causal effects
 - $\text{sense}(wff)$ is true if expression wff holds in the possible world
 - For simplicity, sense effects cannot occur in actions with preconditions (including being in conditional effects)
- Adds Knowledge Effects
 - $K(\neg u:v)$ is true if, being in world v the agent knows that u is not a consistent (accessible) possible world

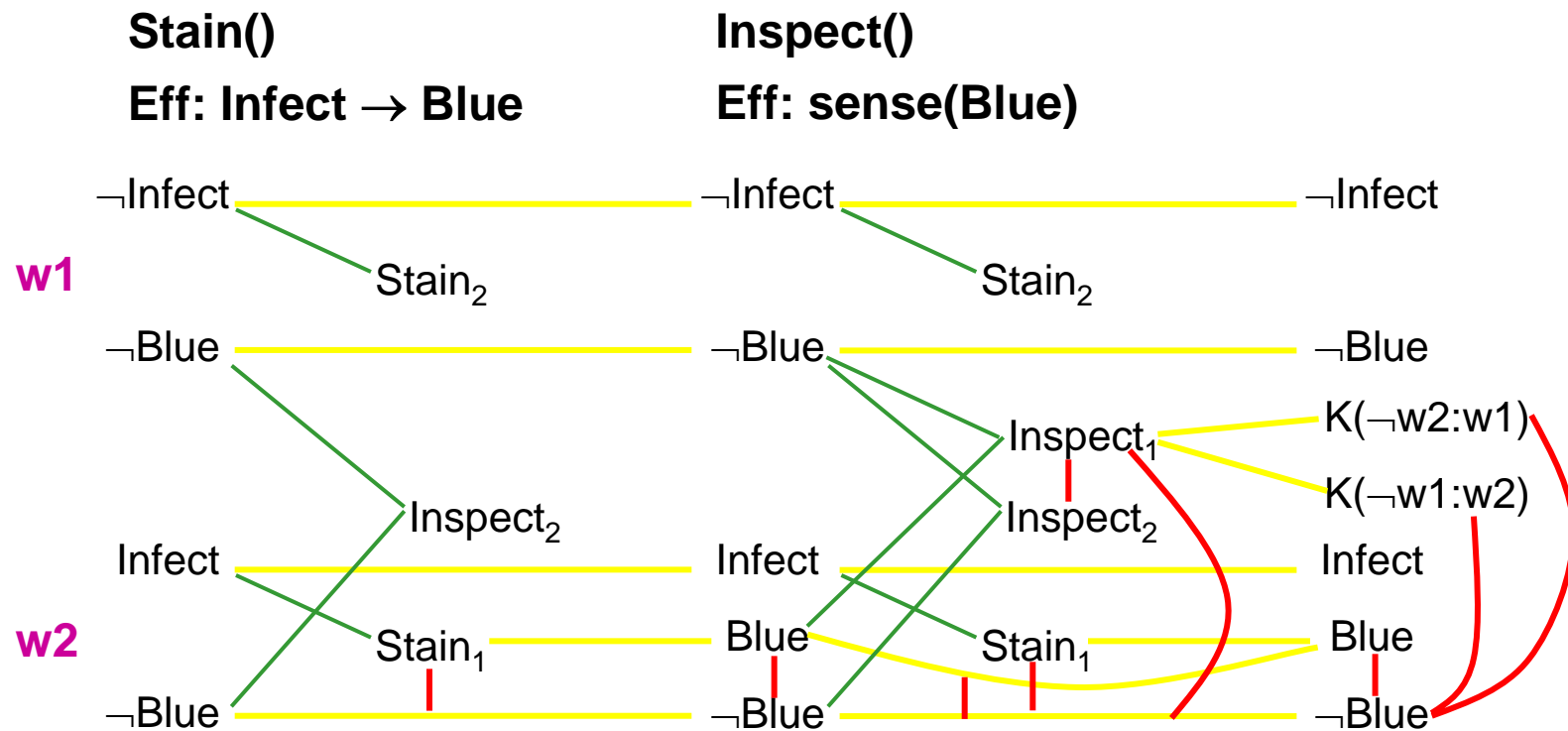
Sense Effects

- The Result of a Sense Effect is to Partition the Accessibility Relation Amongst Possible Worlds



Expanding Plan Graph

- Compute Knowledge Propositions at Each Action Level
 - Link knowledge propositions to sense effects
 - Link sense effects to literals in *all* relevant possible worlds



Solution Extraction

- Add *Conditioning* to CGP Solution Extraction
 - Add as subgoals knowledge propositions that partition aspects that are mutex with previously added aspects
 - Uses constraint satisfaction to maintain contexts
 - Planner does not have to commit to exactly what possible world the system is in
- Can Still Find Conformant Plans, if Exist
- *The First Conditional Planner with Reasonable Performance!!*