New Mexico State University
Son Thanh To
Tran Cao Son
Enrico Pontelli

Contingent Planning as And/Or forward Search with Disjunctive Representation

Outline

- Contingent planning
- Approach overview
  - Employed **DNF representation**
  - **PrAO**: a new And/Or forward search with **novel pruning techniques**
- DNF representation: review & **extension**
- **PrAO** And/Or forward search algorithm
- Experimental evaluation
- Conclusion and future work

Outline

- Contingent planning
  - An example & formulation
  - And/Or search for solutions
- Approach overview
  - Employed DNF representation
  - **PrAO**: an And/Or forward search
- DNF representation: review & extension
- And/Or forward search algorithm: PrAO
- Experimental evaluation
- Conclusion and future work

Contingent Planning: An Example

at-same-room
¬bug-is-dead

¬at-same-room
¬bug-is-dead

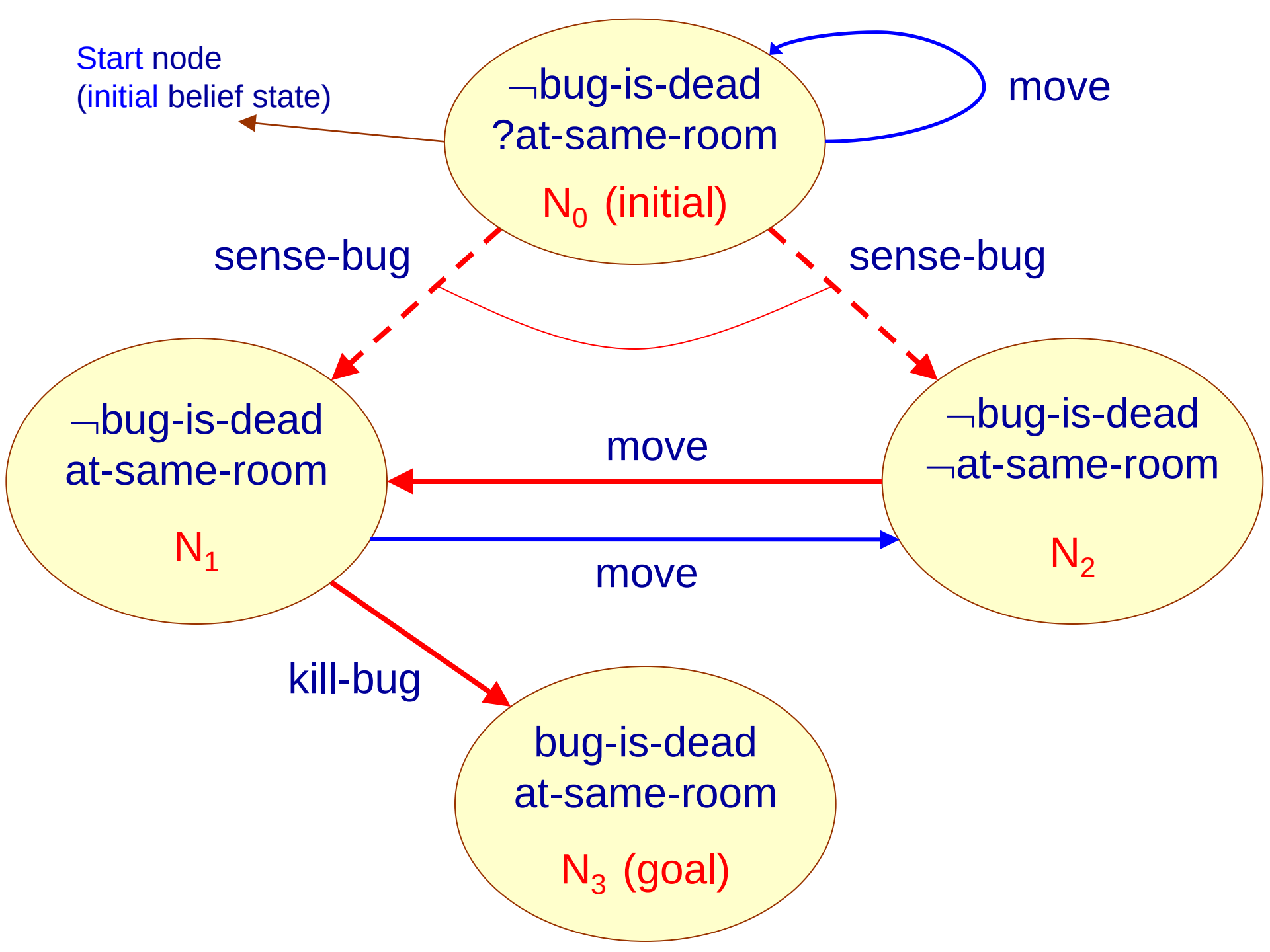
Contingent Planning Problem

- Problem $P = \langle F, A, \Omega, I, G \rangle$
  - **Fluents**: $F = \{bug\text{-}is\text{-}dead, at\text{-}same\text{-}room\}$
  - **Actions**: $A = \{move, kill\text{-}bug\}$
    - $pre(move) = \emptyset$
      $move: at\text{-}same\text{-}room \rightarrow \neg at\text{-}same\text{-}room$
      $move: \neg at\text{-}same\text{-}room \rightarrow at\text{-}same\text{-}room$
    - $pre(kill\text{-}bug) = at\text{-}same\text{-}room$
      $kill\text{-}bug: \emptyset \rightarrow bug\text{-}is\text{-}dead$
  - **Sensing**: $\Omega = \{sense\text{-}bug\}$, $pre(sense\text{-}bug) = \emptyset$
    - $l(sense\text{-}bug) = at\text{-}same\text{-}room$
  - **Initial State**: $I = \neg bug\text{-}is\text{-}dead$ (? $at\text{-}same\text{-}room$)
  - **Goal**: G = bug-is-dead

- The initial *Belief State:* set of states satisfying I:
  $BS(I) = \{ \{\neg bug\text{-}is\text{-}dead, at\text{-}same\text{-}room\}$,
  $\{\neg bug\text{-}is\text{-}dead, \neg at\text{-}same\text{-}room\} \}$

Outline

- Contingent planning
  - ✓Formulation
  - ➢And/Or search for solutions
- Approach overview
  - ▪Employed DNF representation
  - ▪**PrAO**: an And/Or forward search
- DNF representation: review & extension
- And/Or forward search algorithm: PrAO
- Experimental evaluation
- Conclusion and future work


Start node
(initial belief state)
¬bug-is-dead
?at-same-room
N0 (initial)
move
sense-bug
sense-bug
¬bug-is-dead
at-same-room
N1
move
¬bug-is-dead
¬at-same-room
N2
move
kill-bug
bug-is-dead
at-same-room
N3 (goal)

Overview of Our Approach

- **Extend DNF representation** for conformant planning (ICAPS-2009) to handle
  - Non-deterministic actions
  - Sensing actions
- **Develop PrAO**: an And/Or forward search with
  - Novel pruning techniques
  - The remaining search graph when a solution is detected is also the solution

Outline

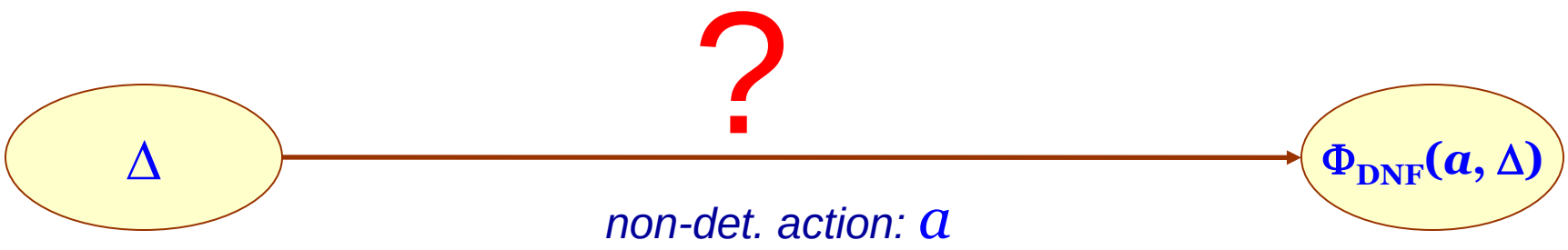
- ✓ Contingent planning
- ✓ Approach overview
- ➢ DNF representation
  - ➢ Brief review (conformant planning)
  - ▪ Extending for contingent planing
- And/Or forward search algorithm: PrAO
- Experimental evaluation
- Conclusion and future work

Outline

- ✓ Contingent planning
- ✓ Approach overview
- DNF representation
  - ✓ Review (conformant planning)
  - ➢ Extending for contingent planning
- And/Or forward search algorithm: PrAO
- Experimental evaluation
- Conclusion and future work

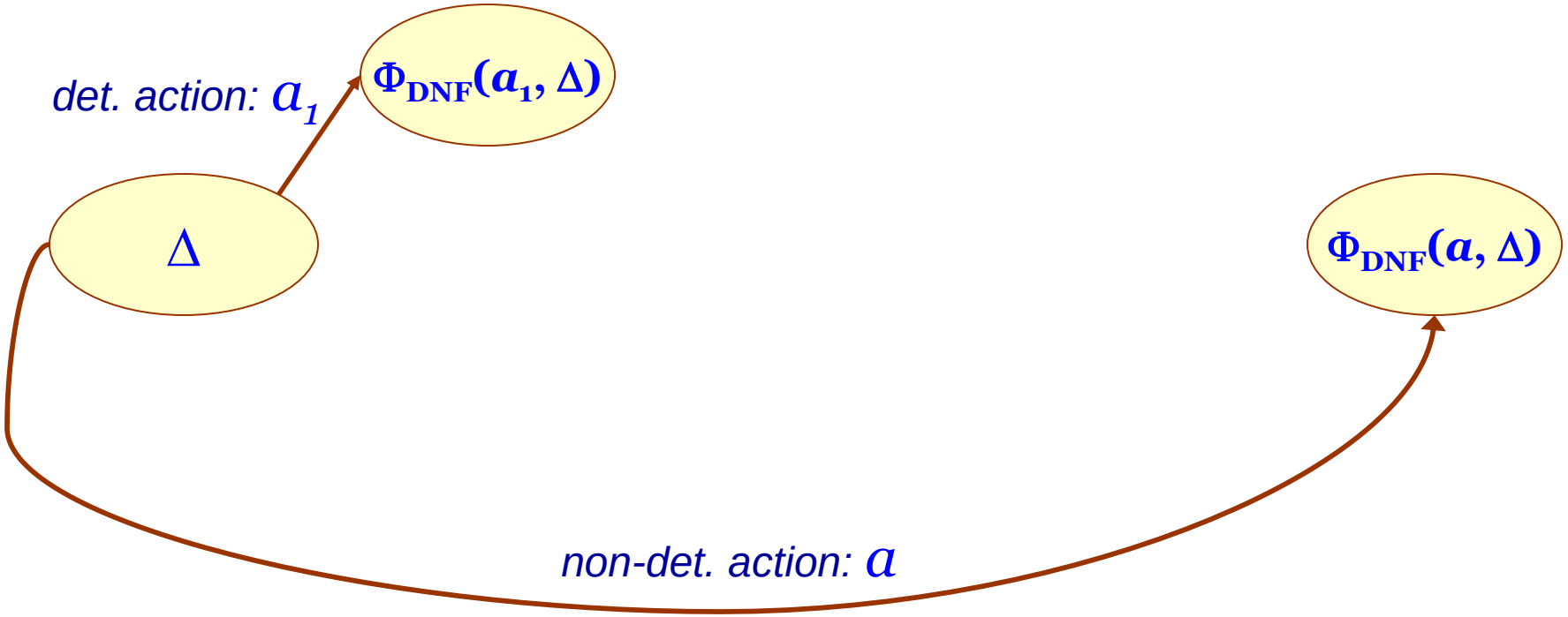
Extending to Non-deterministic action

- A non-deterministic action a contains a set of outcomes: o_1, …, o_n. (a is deterministic if $n = 1$)
- o_i is a set of conditional effects
- a_i : the deterministic action with set of effects o_i



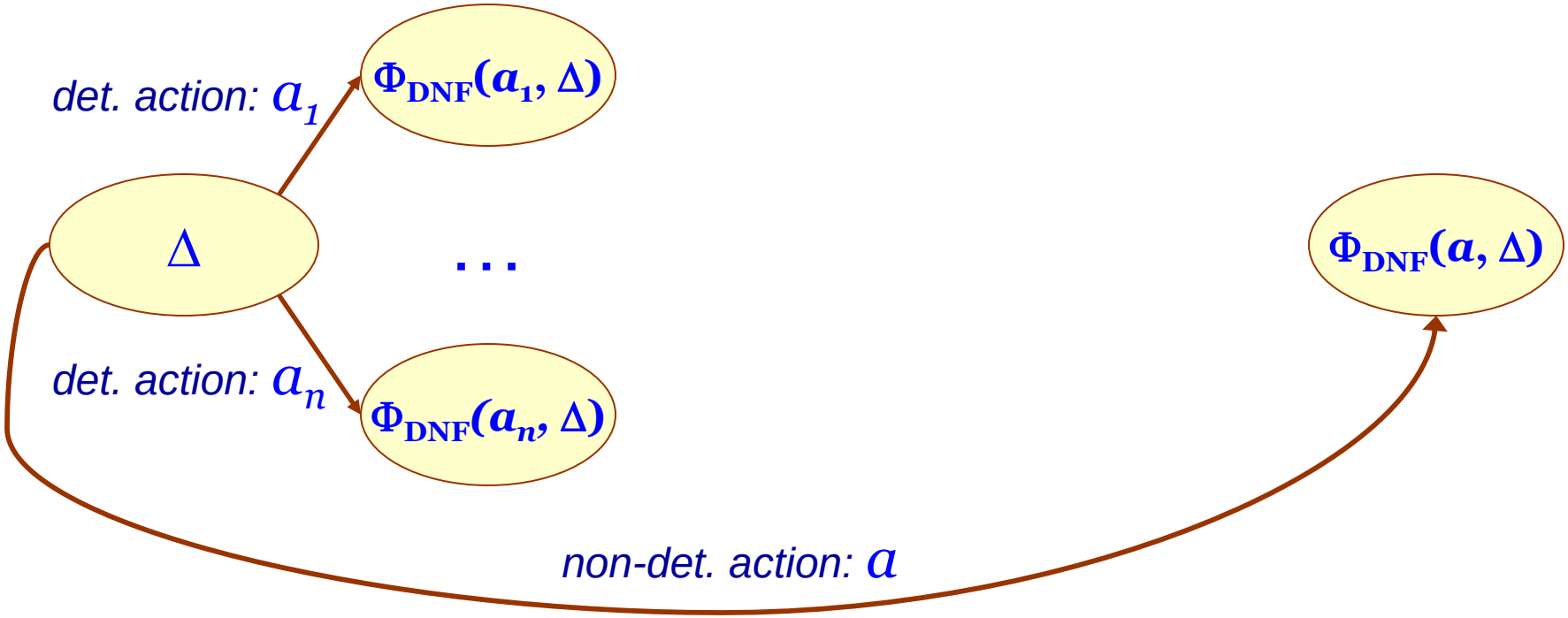
Extending to Non-deterministic action

- A non-deterministic action a contains a set of outcomes: o_1, …, o_n. (a is deterministic if $n = 1$)
- o_i is a set of conditional effects
- a_i : the deterministic action with set of effects o_i



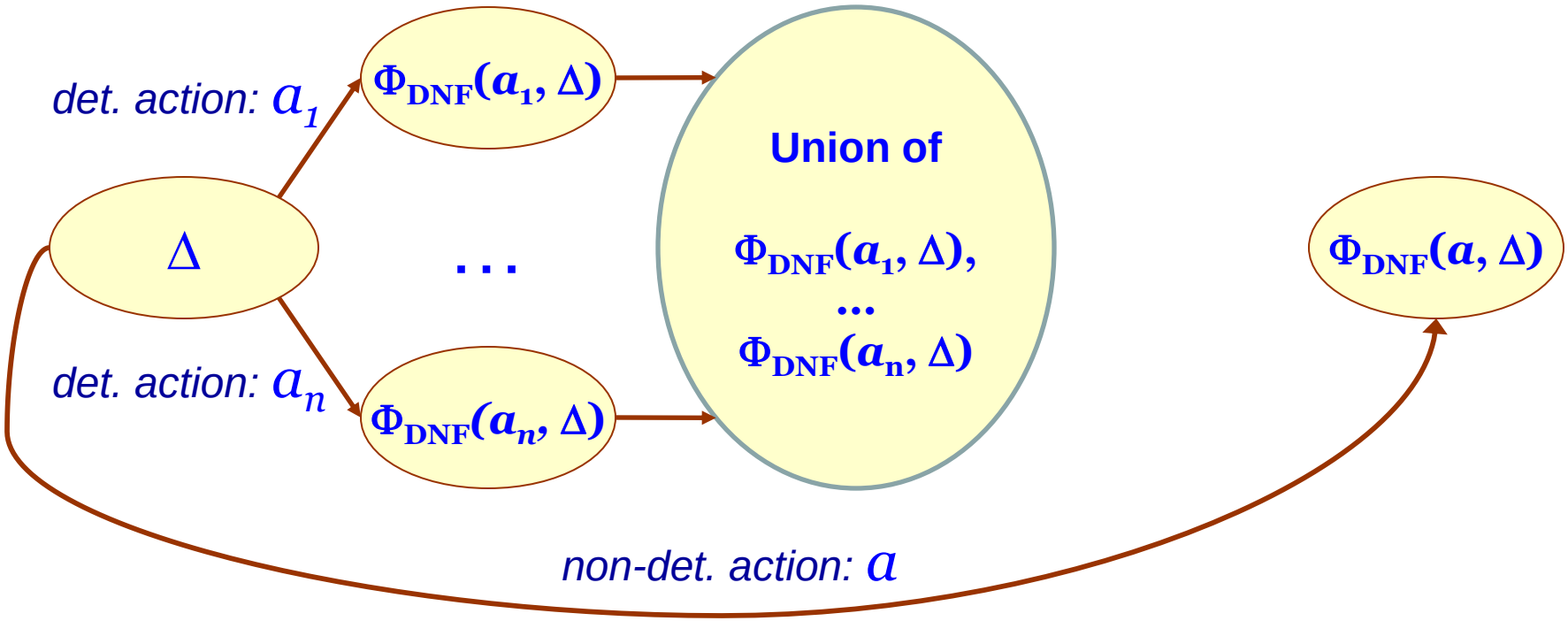
Extending to Non-deterministic action

- A non-deterministic action a contains a set of outcomes: o_1, …, o_n. (a is deterministic if $n = 1$)
- o_i is a set of conditional effects
- a_i : the deterministic action with set of effects o_i



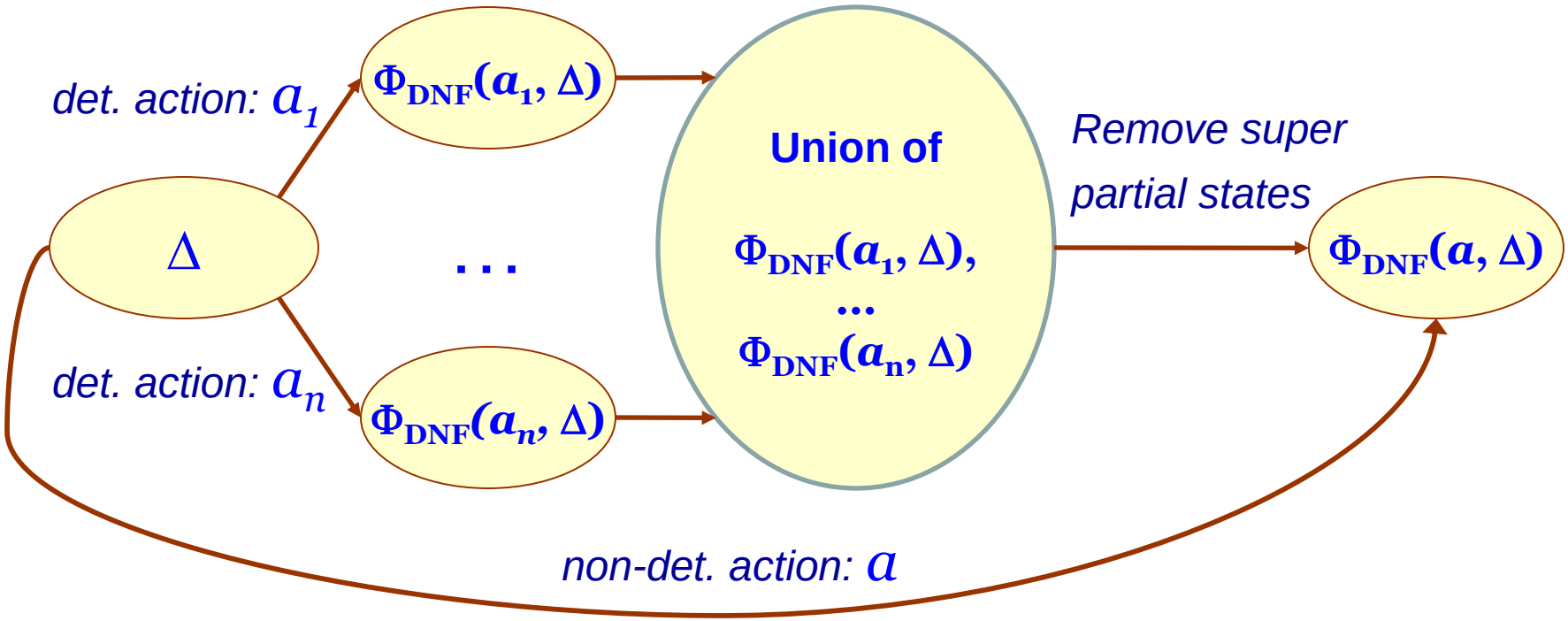
Extending to Non-deterministic action

- A non-deterministic action a contains a set of outcomes: o_1, …, o_n. (a is deterministic if $n = 1$)
- o_i is a set of conditional effects
- a_i : the deterministic action with set of effects o_i



Extending to Non-deterministic action

- A non-deterministic action a contains a set of outcomes: o_1, …, o_n. (a is deterministic if $n = 1$)
- o_i is a set of conditional effects
- a_i : the deterministic action with set of effects o_i



Successor states for Sensing Actions

- Sensing action ω:
  - $(\omega, S) \rightarrow (S^+, S^-)$
    - $S^+ = \{s \mid s \in S, s \models l(\omega)\}$
    - $S^- = \{s \mid s \in S, s \models \neg l(\omega)\}$
  - $(\omega, \Delta) \rightarrow (\Delta^+, \Delta^-)$: $\Delta^+ \equiv S^+$ $\Delta^- \equiv S^-$ $(\Delta^+ \models l(\omega)$ $\Delta^- \models \neg l(\omega))$
  - Initially: $\Delta^+ = \Delta^- = \varnothing$
  - For every $\delta \in \Delta$:
    - If $\delta \models l(\omega)$ then: $\Delta^+ = \Delta^+ \cup \{ \delta \}$
    - Else if $\delta \models \neg l(\omega)$ then: $\Delta^- = \Delta^- \cup \{ \delta \}$
    - Else:

      $\Delta^+ = \Delta^+ \cup \{ \delta^+ \}$ $\delta^+ = \delta \cup \{ l(\omega) \}$

      $\Delta^- = \Delta^- \cup \{ \delta^- \}$ $\delta^- = \delta \cup \{ \neg l(\omega) \}$

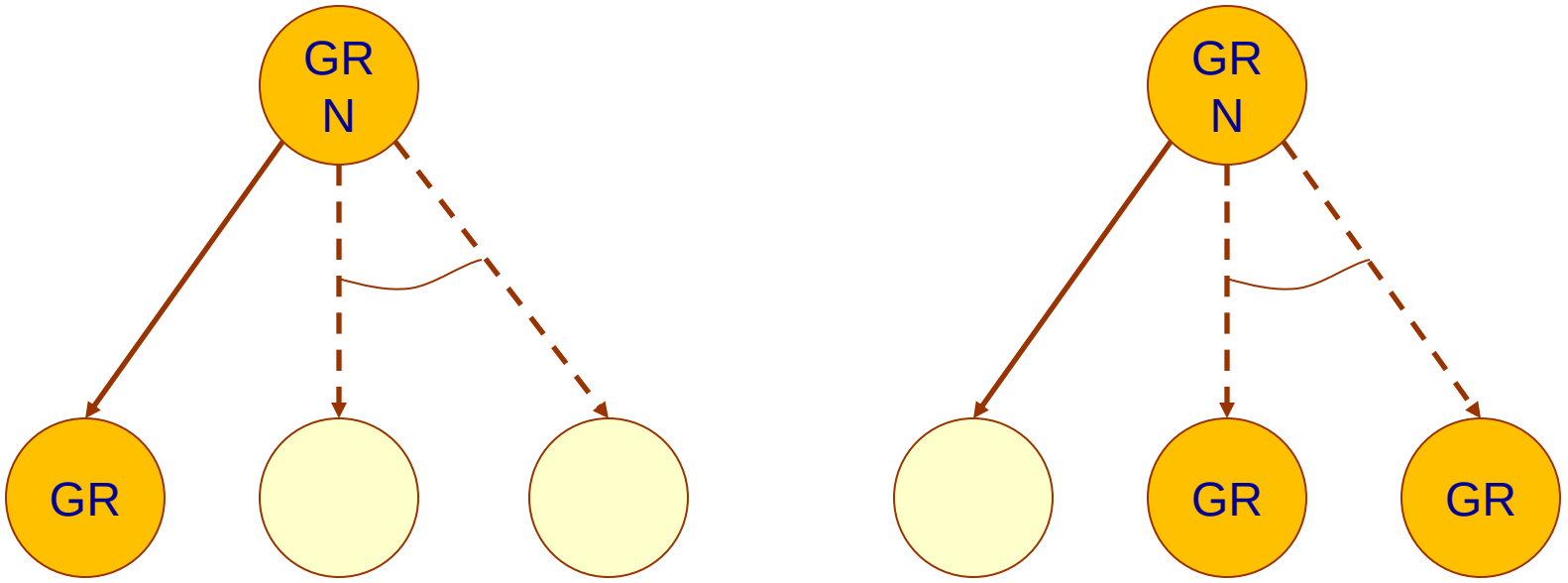
      Intuitively: $\delta \equiv \delta^+ \cup \delta^-$ $\Delta^+ \equiv S^+$ $\Delta^- \equiv S^-$

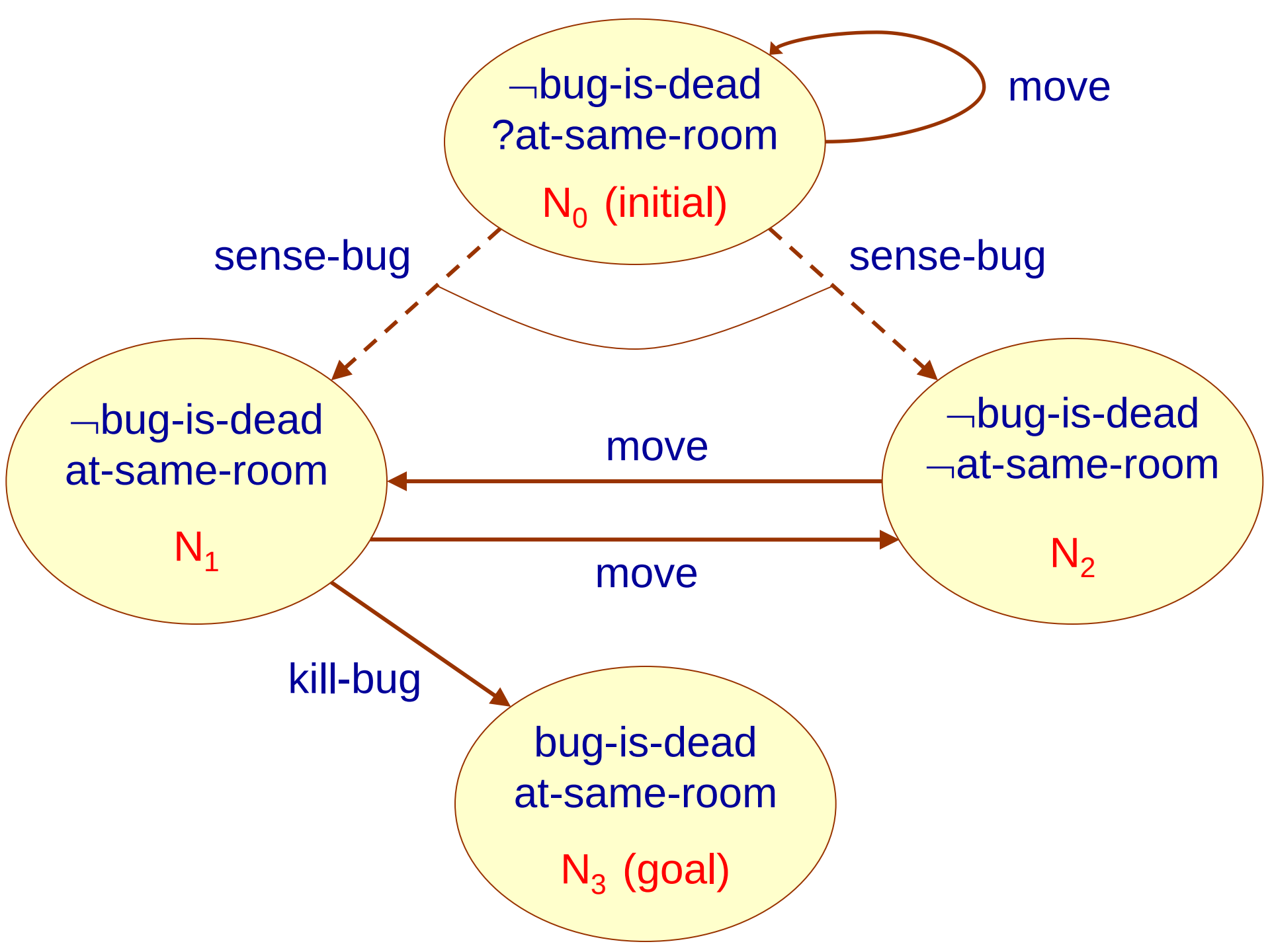
Outline

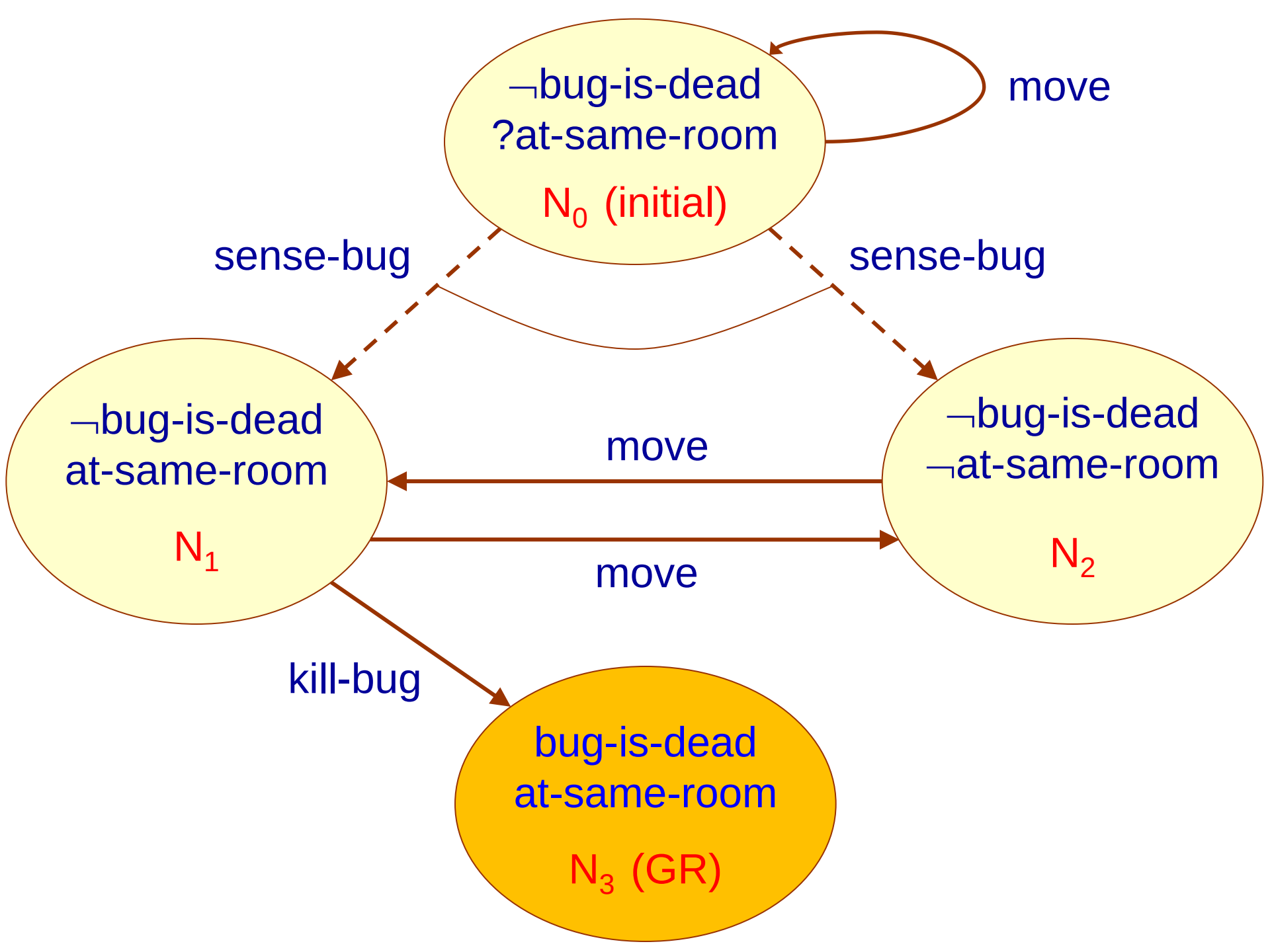
- ✓ Contingent planning
- ✓ Approach overview
- ✓ DNF representation: review & extension
- ➢ PrAO: a new And/Or forward search:
  - ➢ Search space is a **directed graph**
  - ➢ **Novel pruning techniques**
- • Experimental evaluation
- • Conclusion and future work

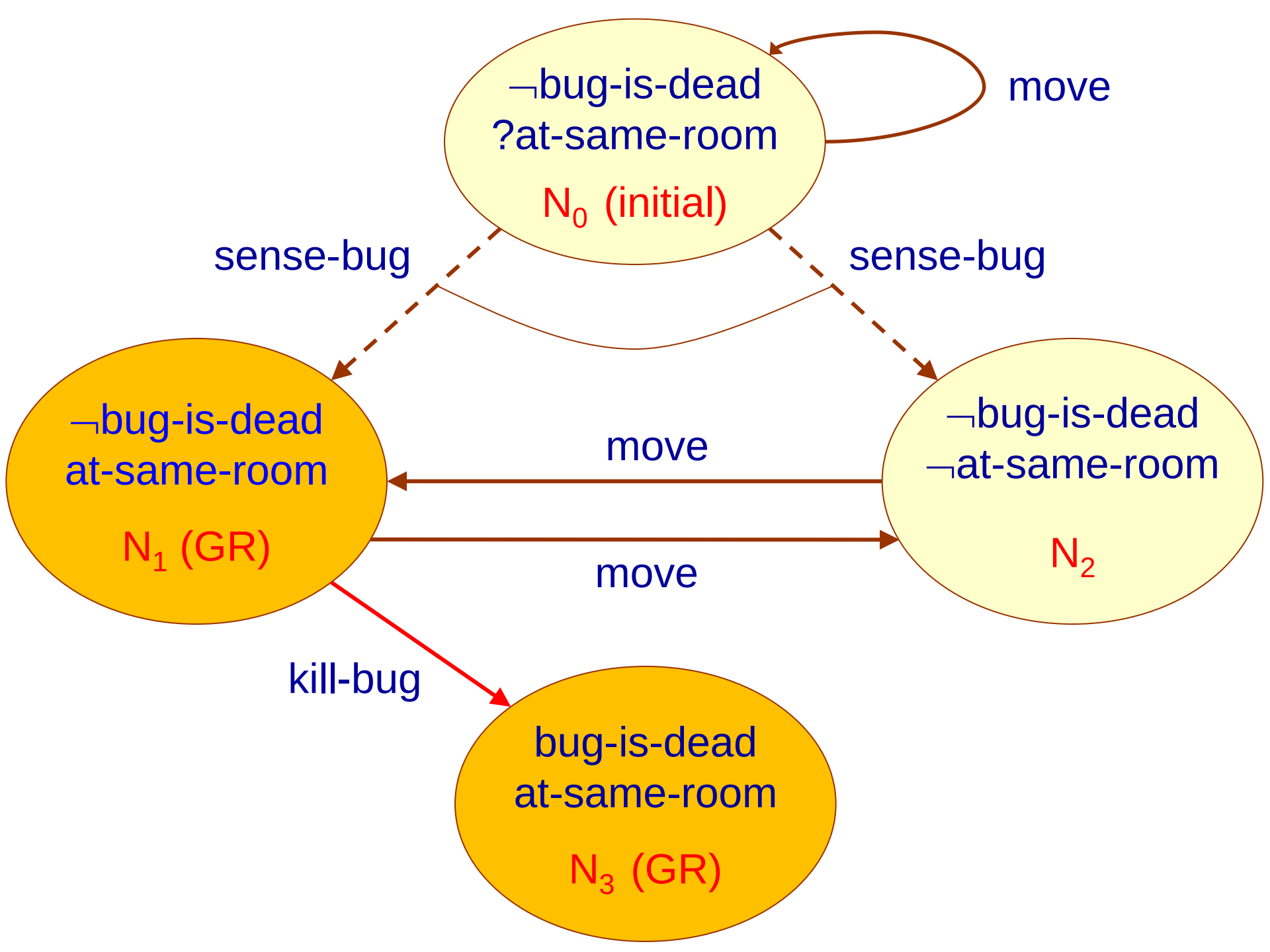
Goal Reachability

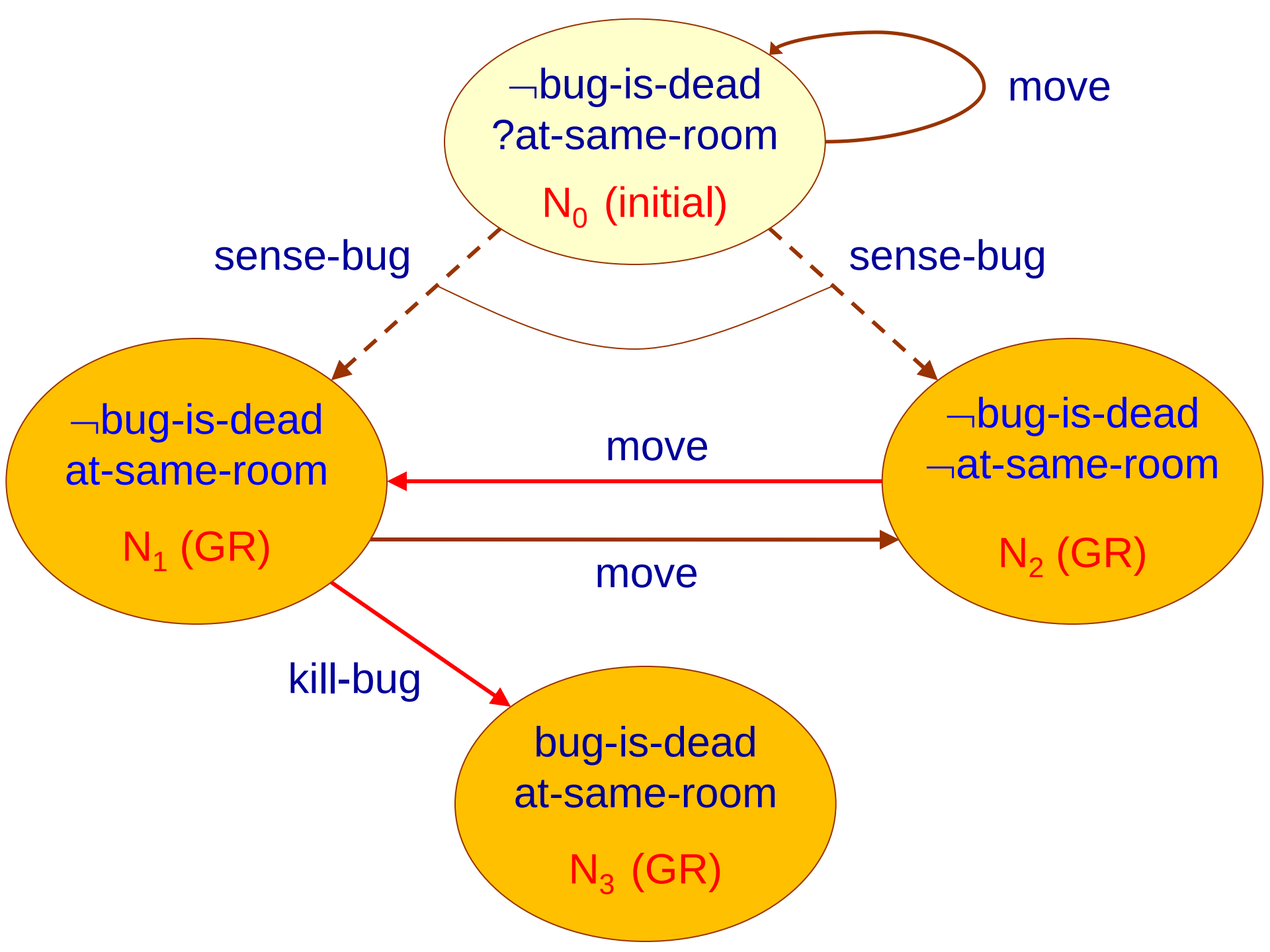
- Node N is ***goal reachable*** (GR) if:
  - Goal node (satisfies the goal), or
  - Has a goal-reachable or-child, or
  - Both dual and-children are goal-reachable

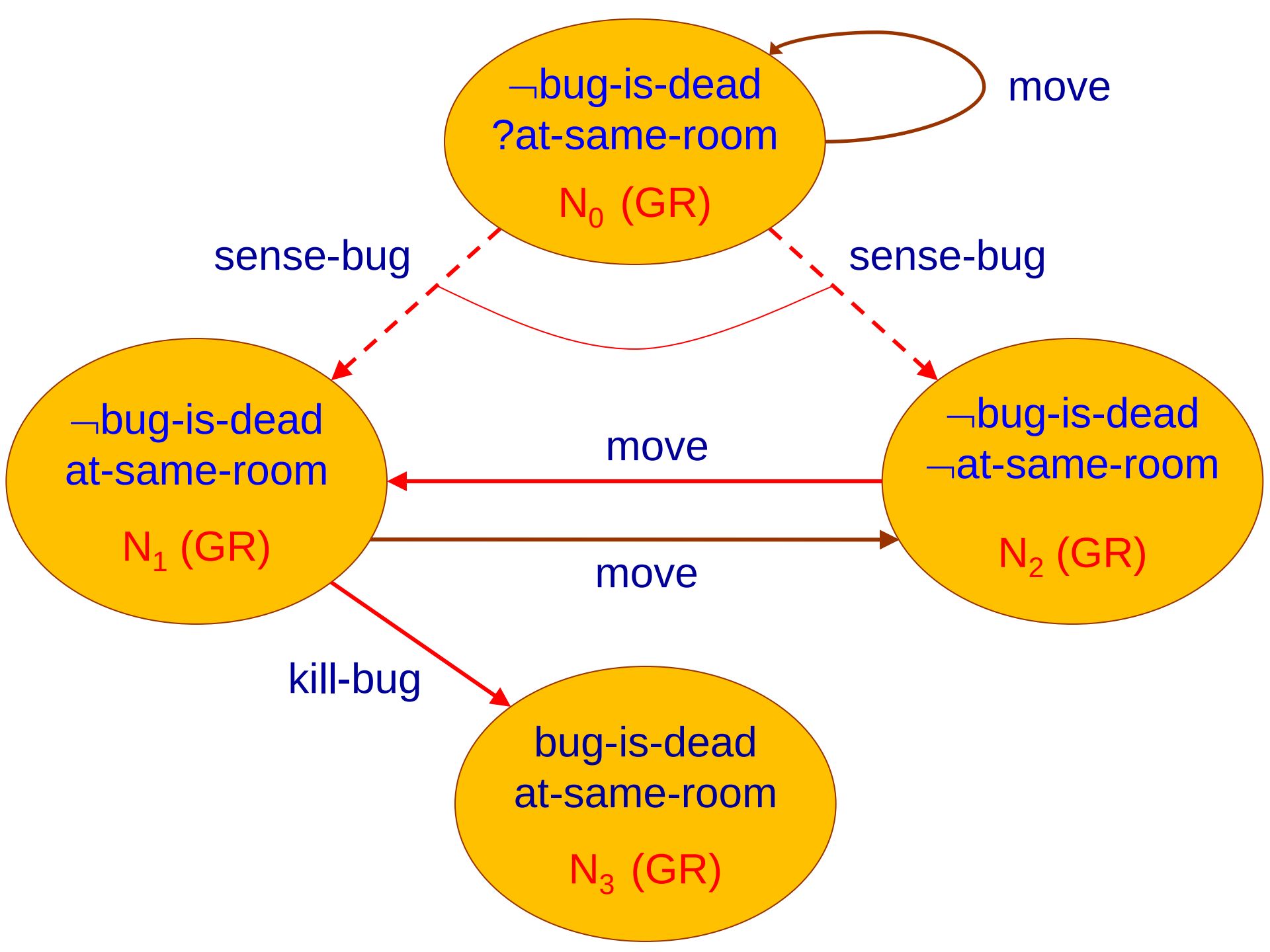



¬bug-is-dead
?at-same-room
N0 (initial)
move
sense-bug
sense-bug
¬bug-is-dead
at-same-room
N1
move
¬bug-is-dead
¬at-same-room
N2
move
kill-bug
bug-is-dead
at-same-room
N3 (goal)


¬bug-is-dead
?at-same-room
N0 (initial)
move
sense-bug
sense-bug
¬bug-is-dead
at-same-room
N1
move
¬bug-is-dead
¬at-same-room
N2
move
kill-bug
bug-is-dead
at-same-room
N3 (GR)

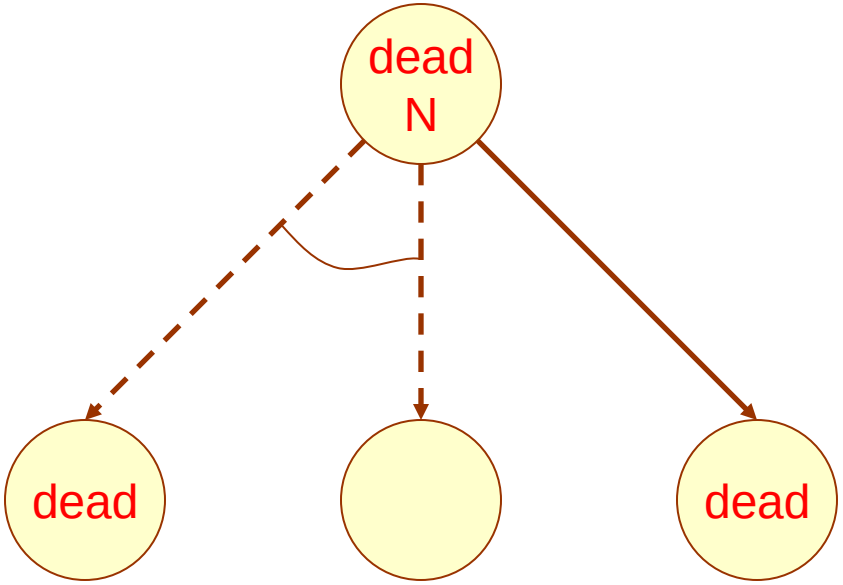

¬bug-is-dead
?at-same-room
N0 (initial)
move
sense-bug
sense-bug
¬bug-is-dead
at-same-room
N1 (GR)
move
move
¬bug-is-dead
¬at-same-room
N2
kill-bug
bug-is-dead
at-same-room
N3 (GR)


¬bug-is-dead
?at-same-room
N0 (initial)
move
sense-bug
sense-bug
¬bug-is-dead
at-same-room
N1 (GR)
move
move
¬bug-is-dead
¬at-same-room
N2 (GR)
kill-bug
bug-is-dead
at-same-room
N3 (GR)


¬bug-is-dead
?at-same-room
N0 (GR)
move
sense-bug
sense-bug
¬bug-is-dead
at-same-room
N1 (GR)
move
¬bug-is-dead
¬at-same-room
N2 (GR)
move
kill-bug
bug-is-dead
at-same-room
N3 (GR)

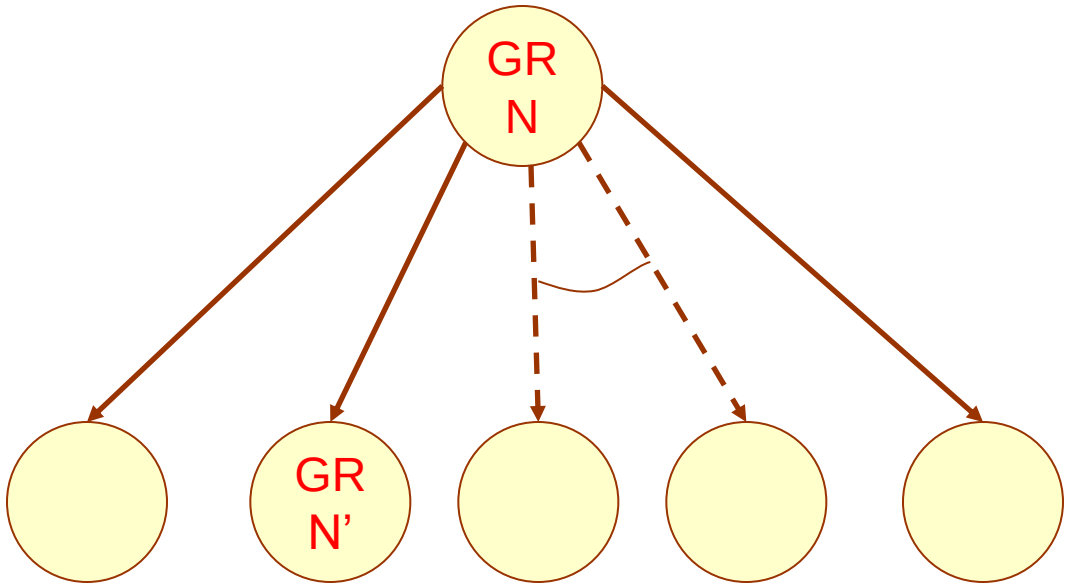
Dead Node Propagation

- Node *N* is **dead** if *N* ≠ goal and either:
  - N has no outgoing edges; or
  - Every or-child is dead, and for every pair of dual and-children, at least one is dead

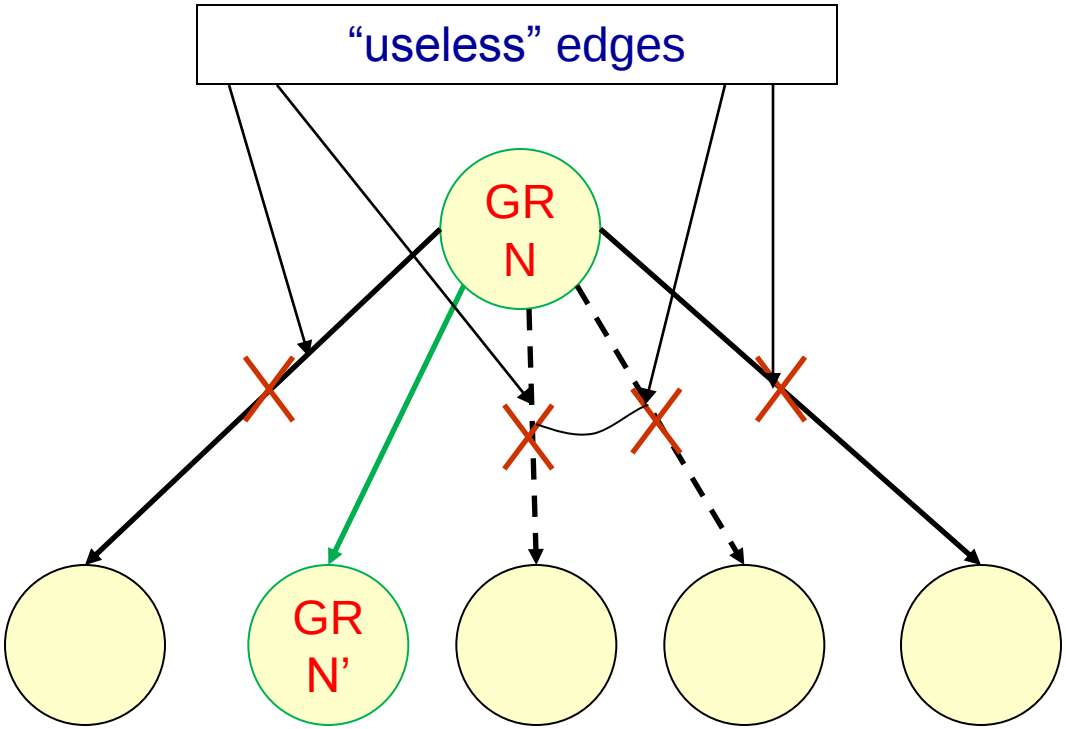


Pruning

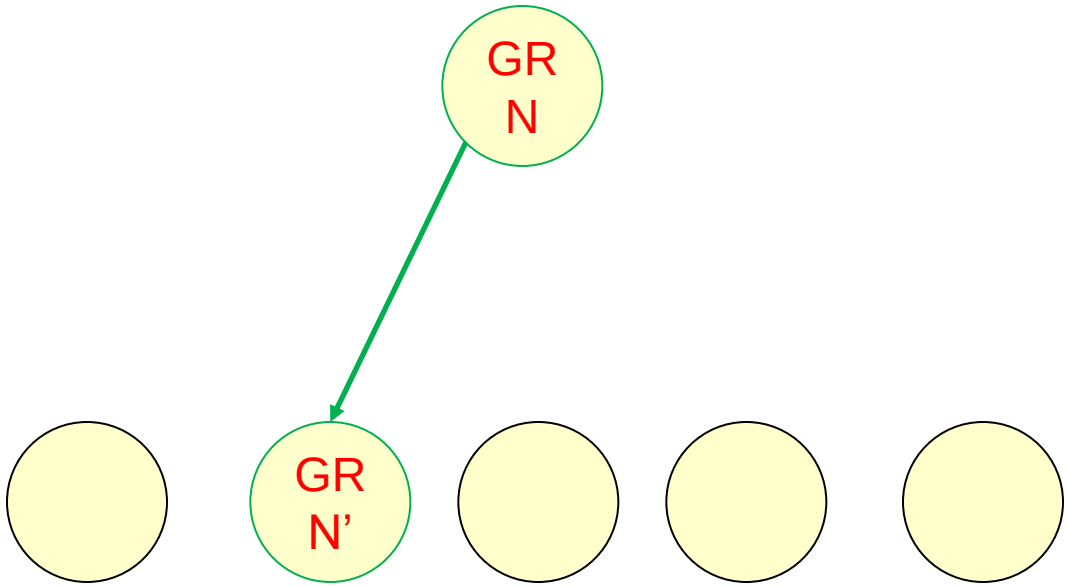
- Observation 1:
  - Some edges may become “useless” due to goal-reachable path



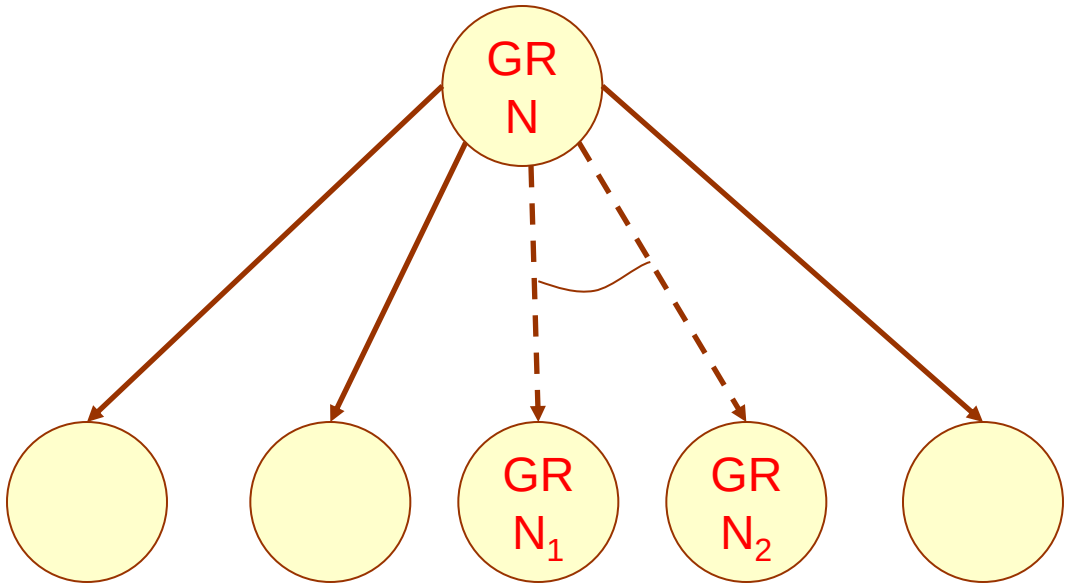
PrAO Algorithm: Pruning Technique



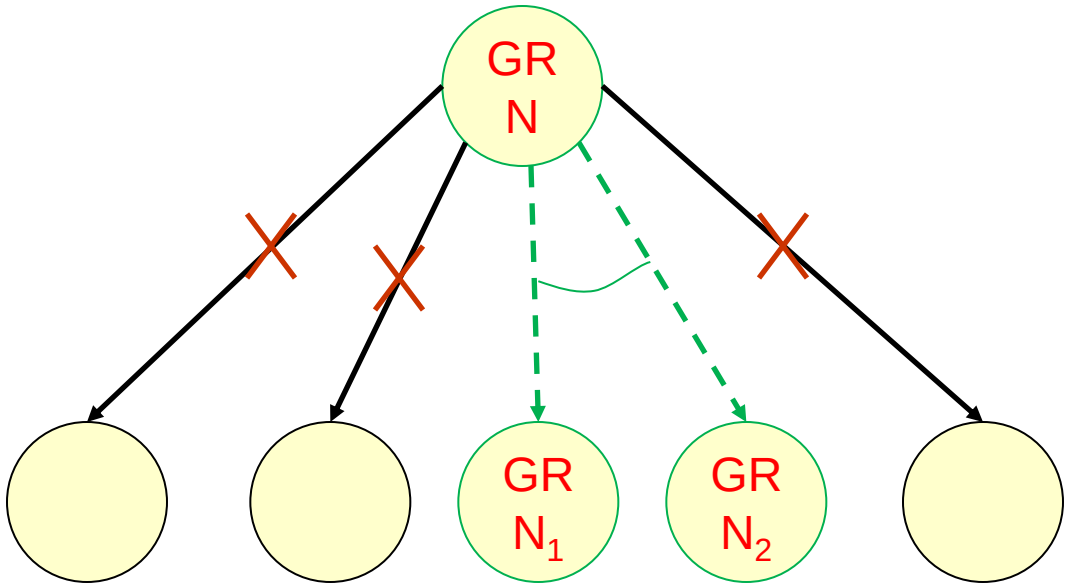
PrAO Algorithm: Pruning Technique



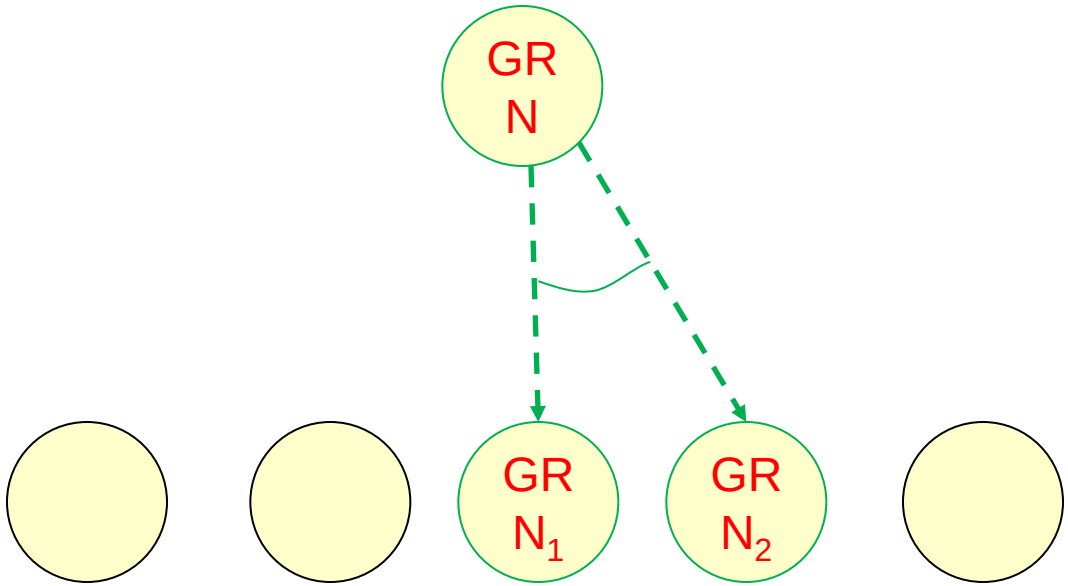
PrAO Algorithm: Pruning Technique



PrAO Algorithm: Pruning Technique

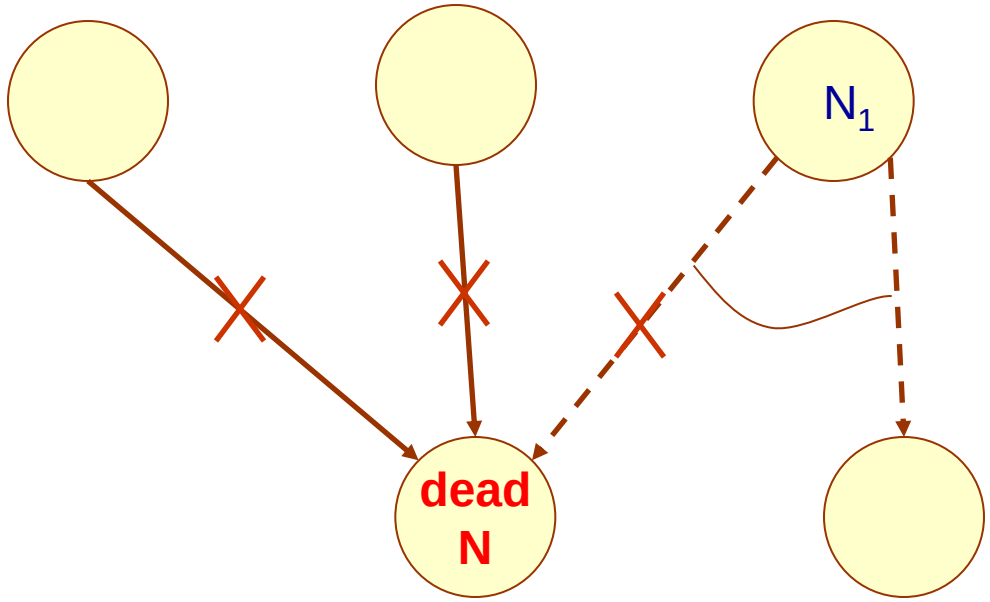


PrAO Algorithm: Pruning Technique



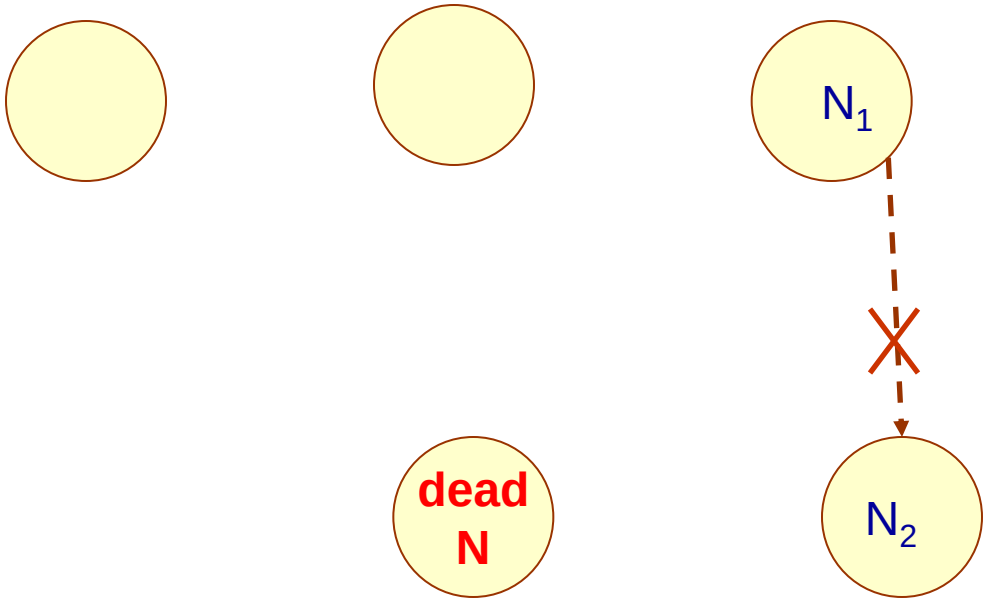
PrAO Algorithm: Pruning Technique

- Observation 2:
  - Edges can be removed due to a dead-node



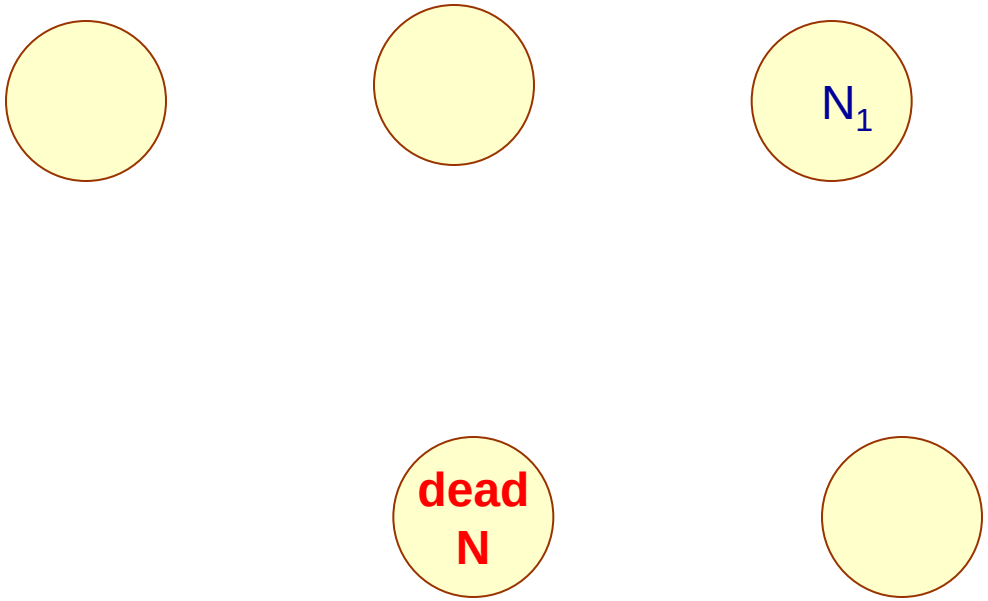
PrAO Algorithm: Pruning Technique

- Observation 2:
  - Edges can be removed by dead-nodes



PrAO Algorithm: Pruning Technique

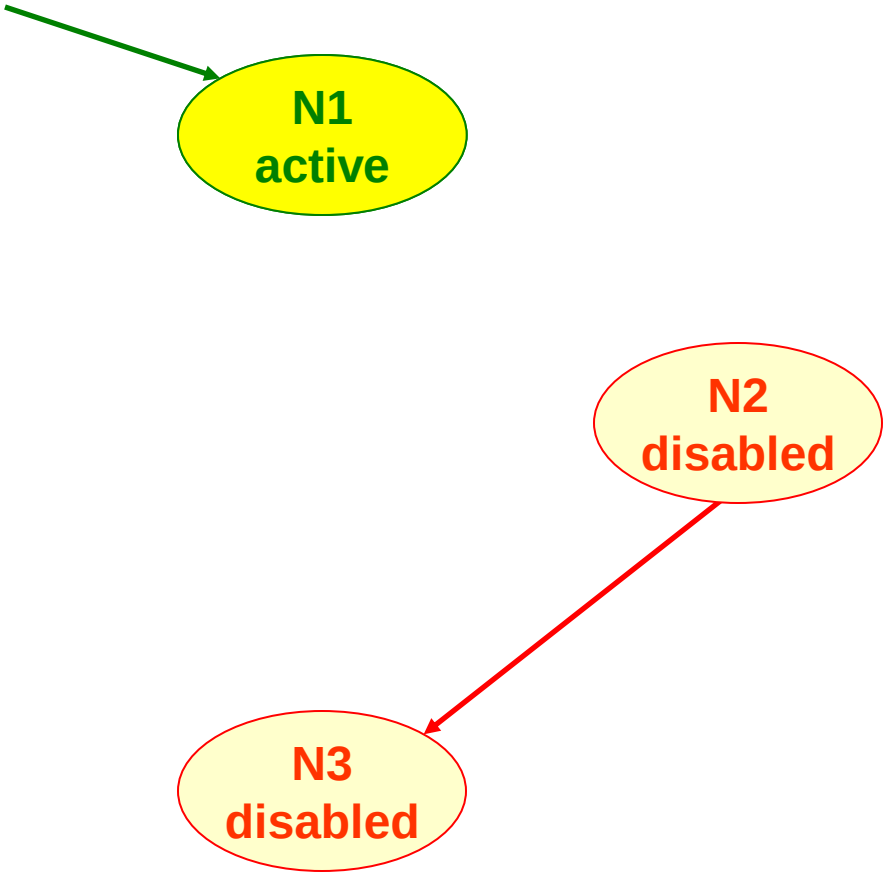
- Observation 2:
  - Edges can be removed by dead-nodes



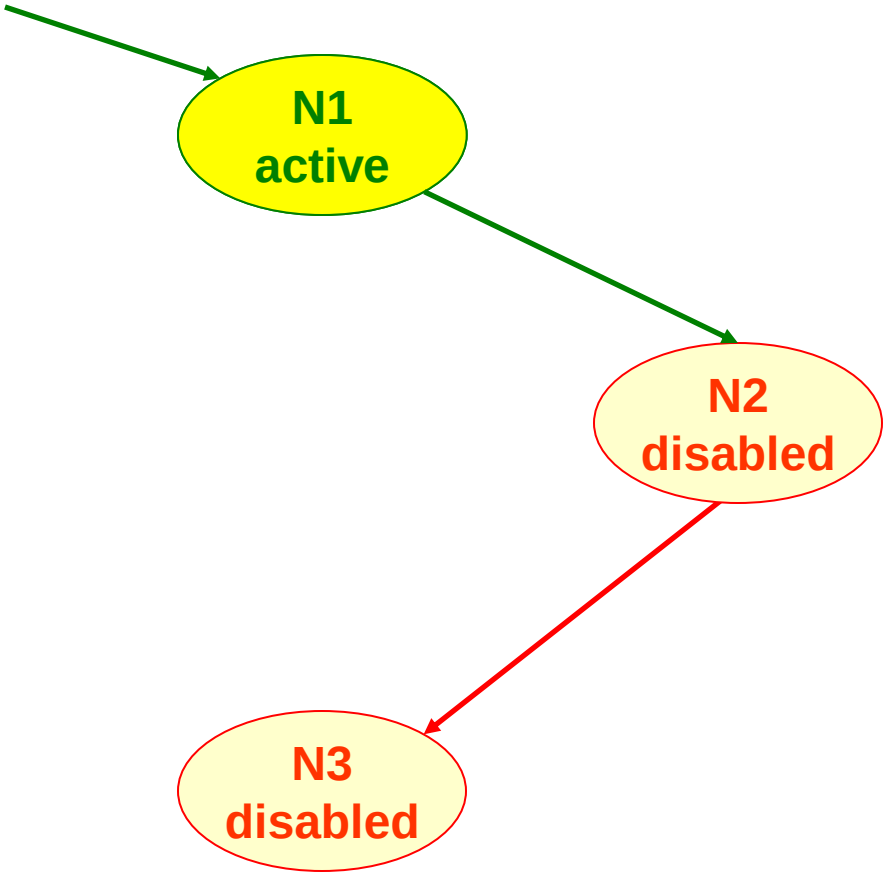
PrAO Algorithm: Pruning Technique

- Active nodes: can be reached from the start node
- Some nodes (and their descendants if exist) may become disabled (inactive) after removing useless edges
- Expand only active nodes
- A disabled node *N* may become active again if it is a child of being expanded node. If *N* is active then so are its descendants (*reactivate-propagation*).

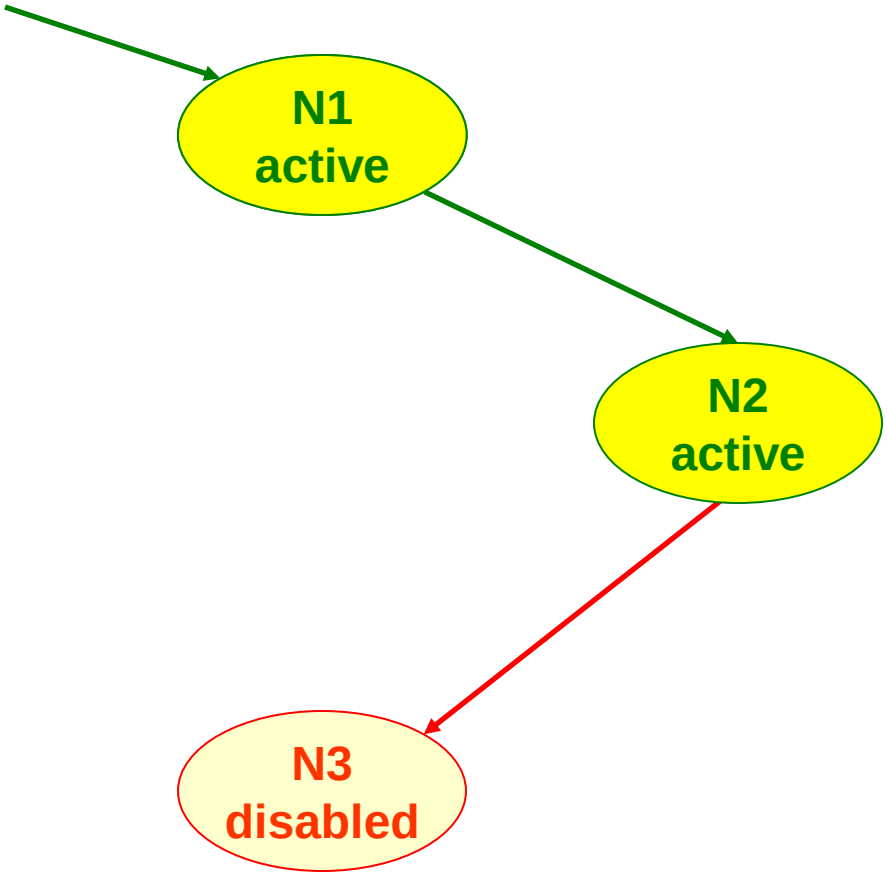
PrAO: Reactivate Disabled Nodes



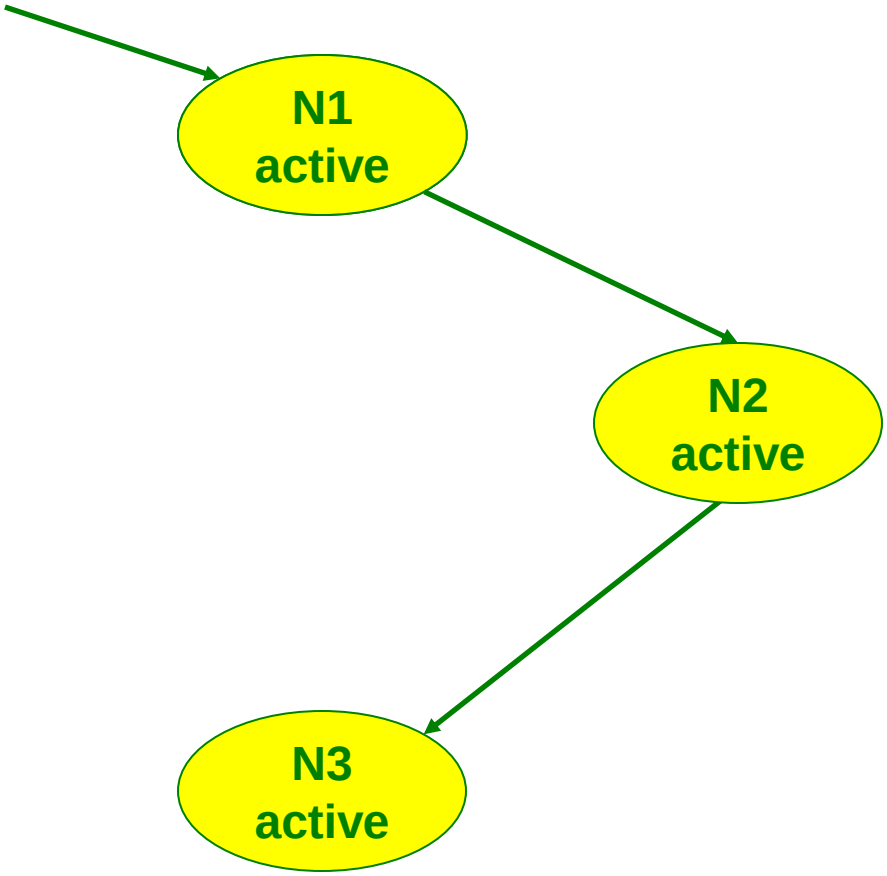
PrAO: Reactivate Disabled Nodes



PrAO: Reactivate Disabled Nodes



PrAO: Reactivate Disabled Nodes



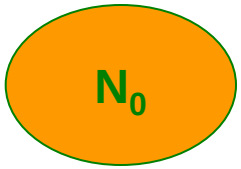
PrAO Algorithm

1. Initialize the front queue Q with the start node n_o (initial belief state).
2. Iteratively perform the following steps:
    3. Pick an active node n from Q with the best heuristic
    4. If no such node n exists, terminate the search
    5. Expand node n
    6. If n is goal-reachable, then
        7. Execute *goal-propagation(n)*
        8. If n_o is goal-reachable then extract and return the solution.
    9. Else if n is dead, then
        10. Execute *dead-end-propagation(n)*
        11. If n_o is a dead node then terminate the search
    10. Else for every disabled child n' of n, activate n' and execute *reactivate-propagation(n')*

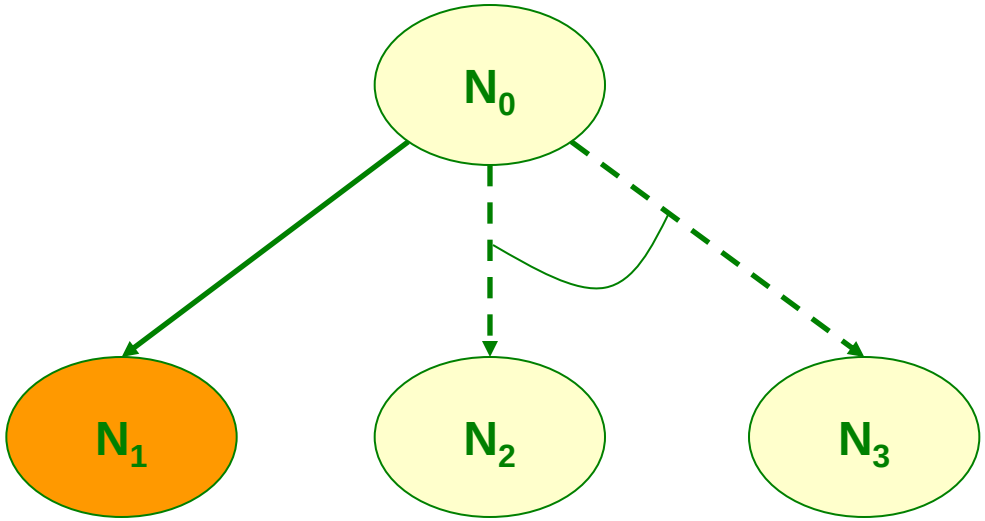
PrAO Algorithm

- Incorporate the pruning techniques in:
  - goal-propagation
  - dead-end-propagation
- Expand only active unexpanded nodes: avoid redundant expansion and reduce numbers of expanded/generated nodes
- PrAO is complete due to *reactivate-propagation*

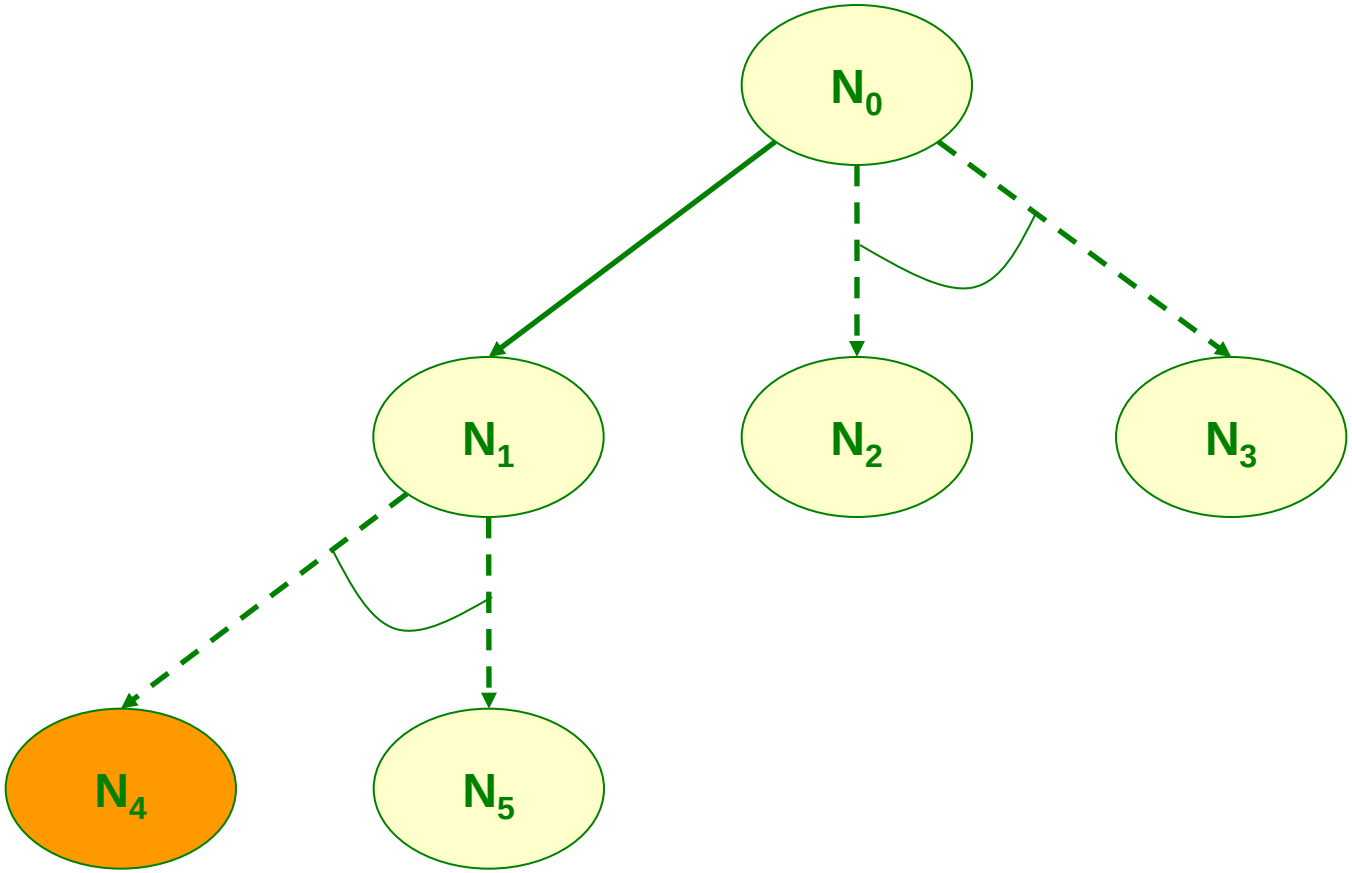
PrAO Algorithm



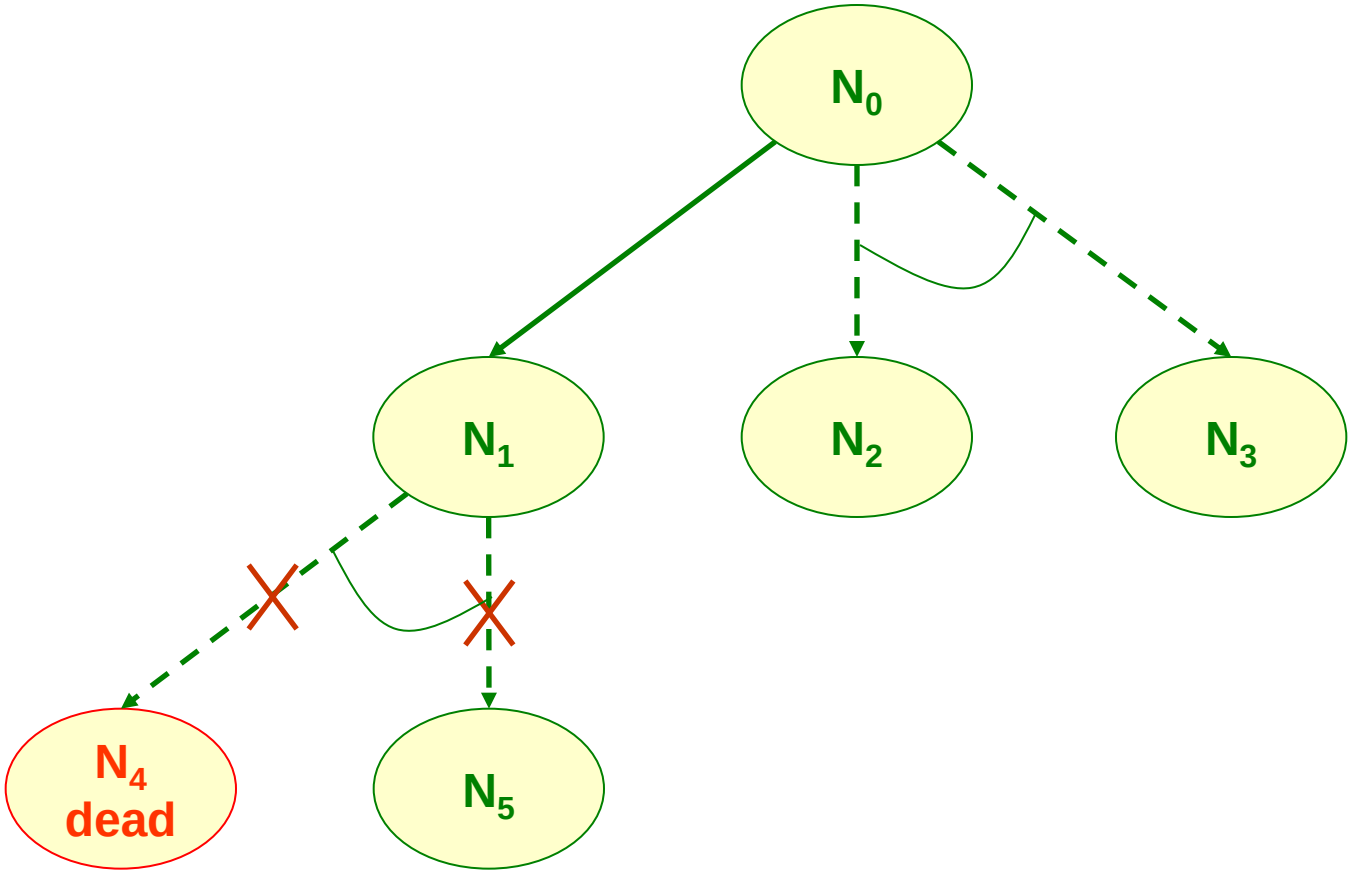
PrAO Algorithm



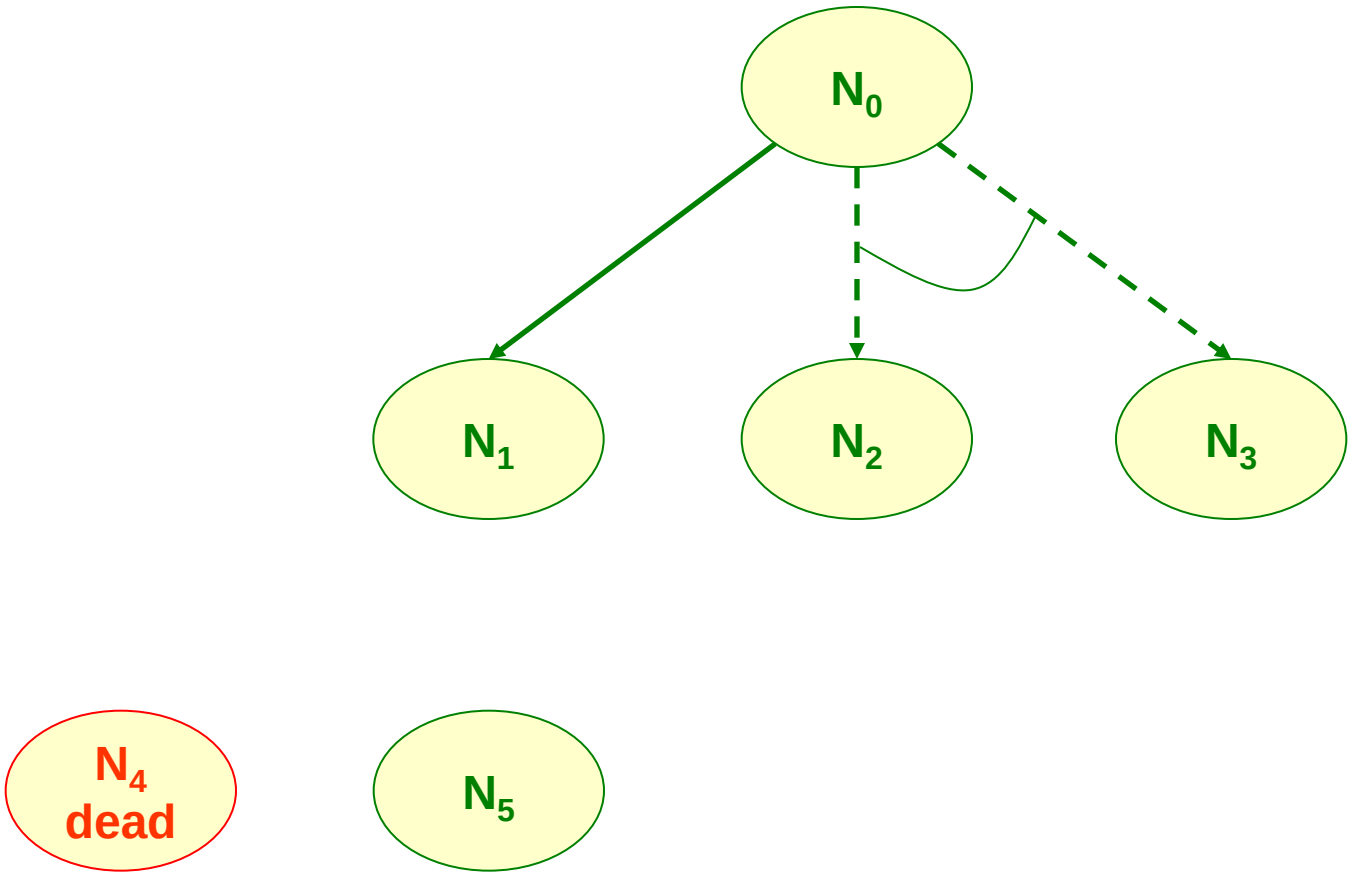
PrAO Algorithm



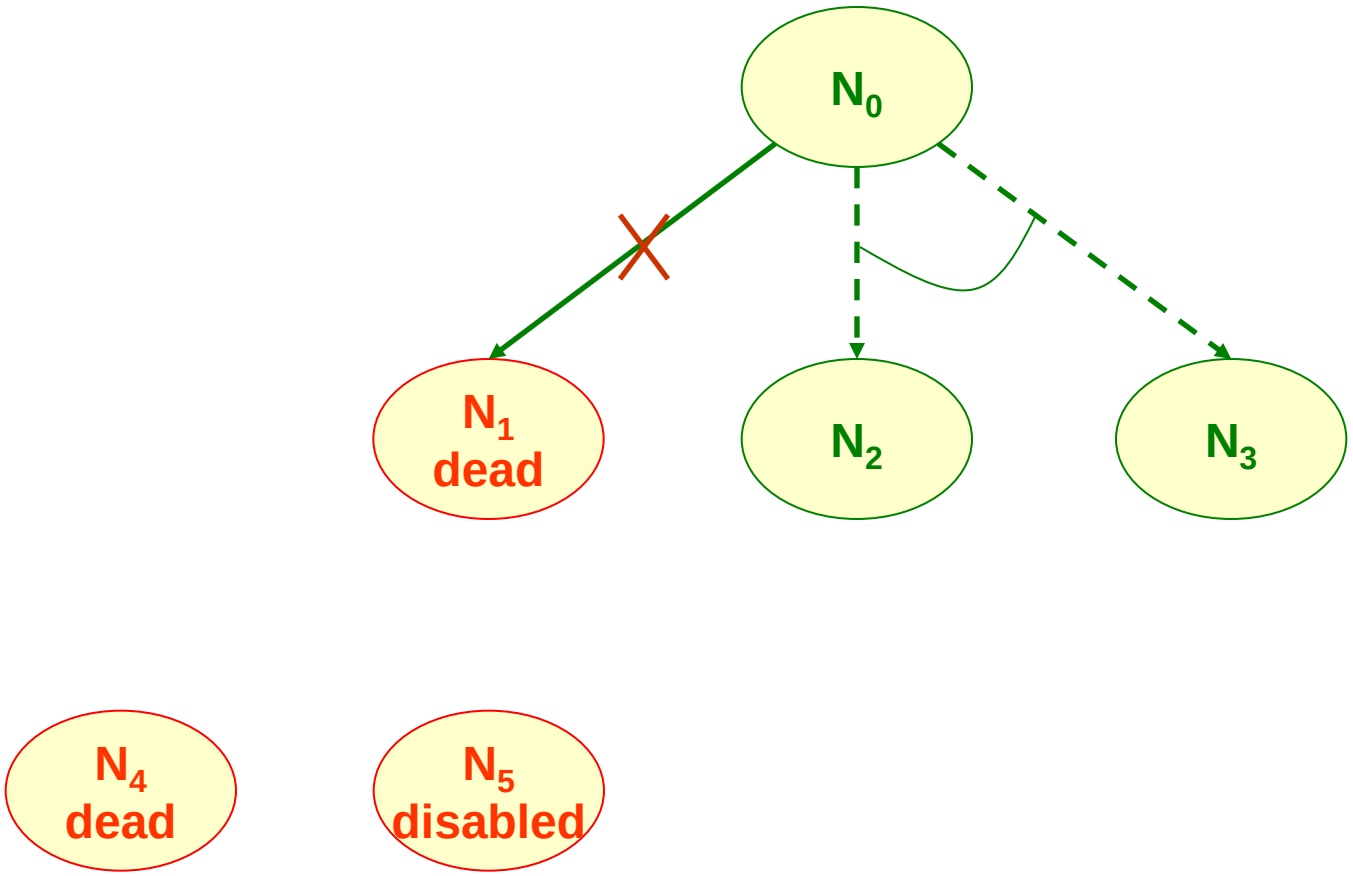
PrAO Algorithm



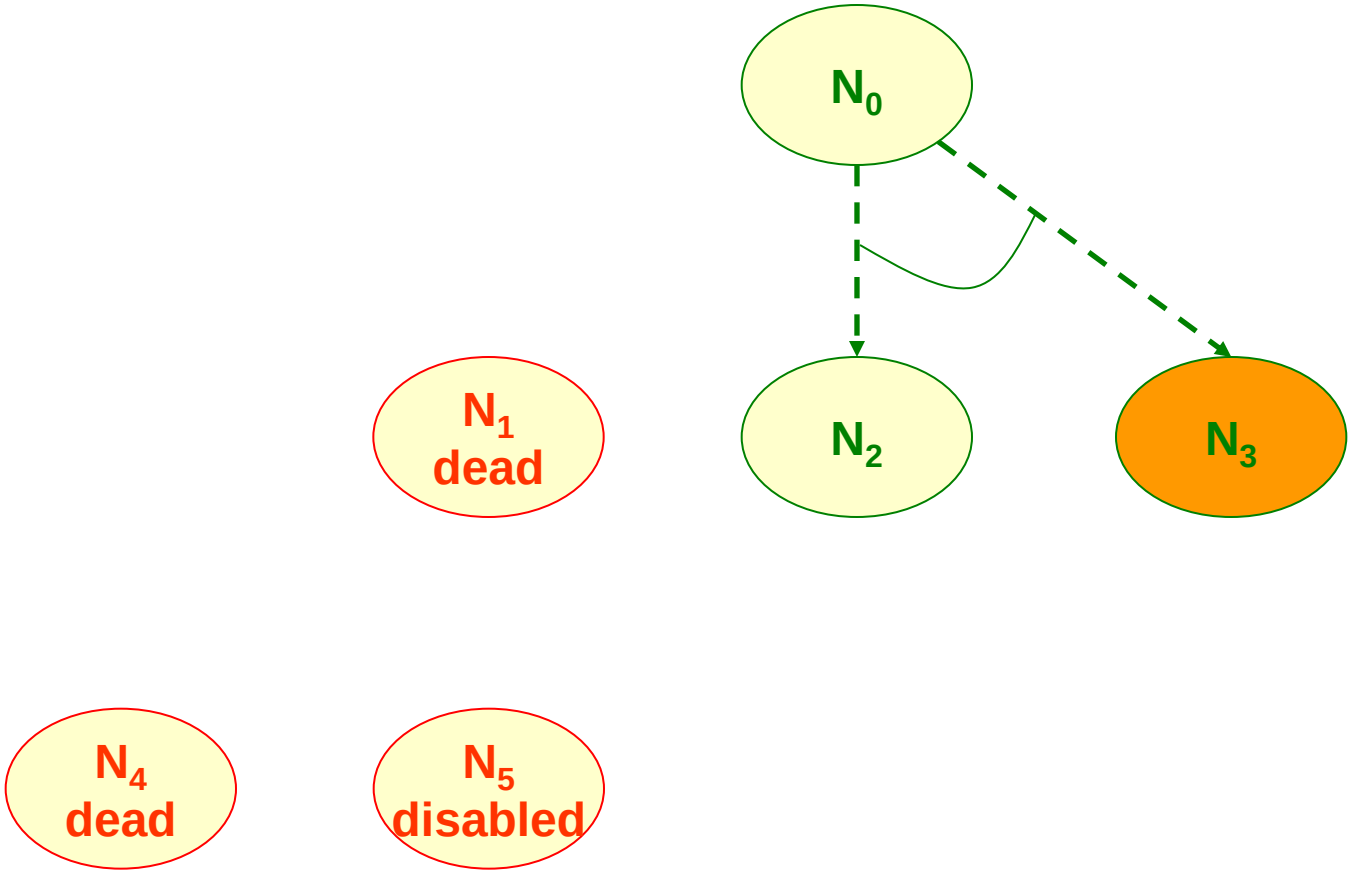
PrAO Algorithm



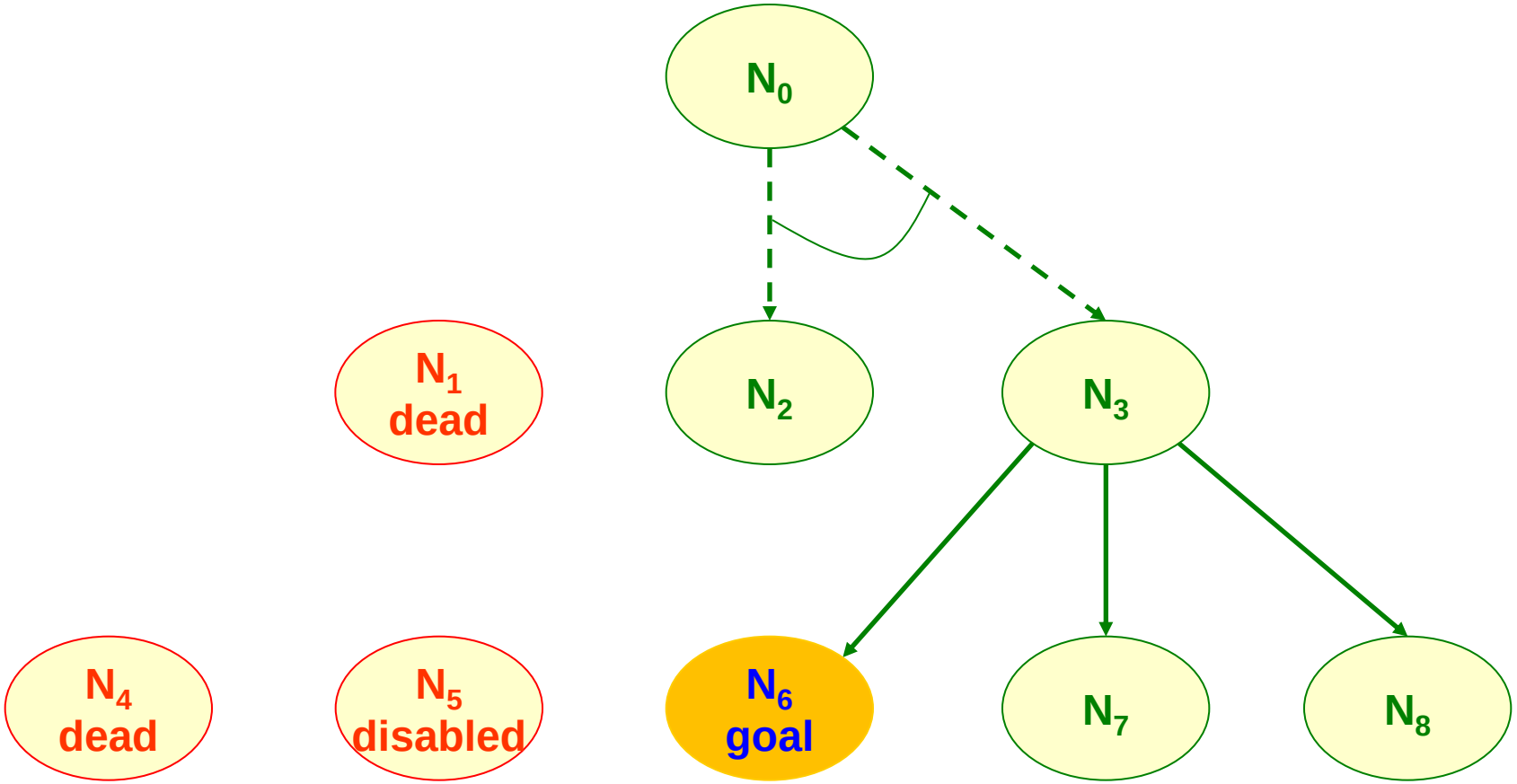
PrAO Algorithm



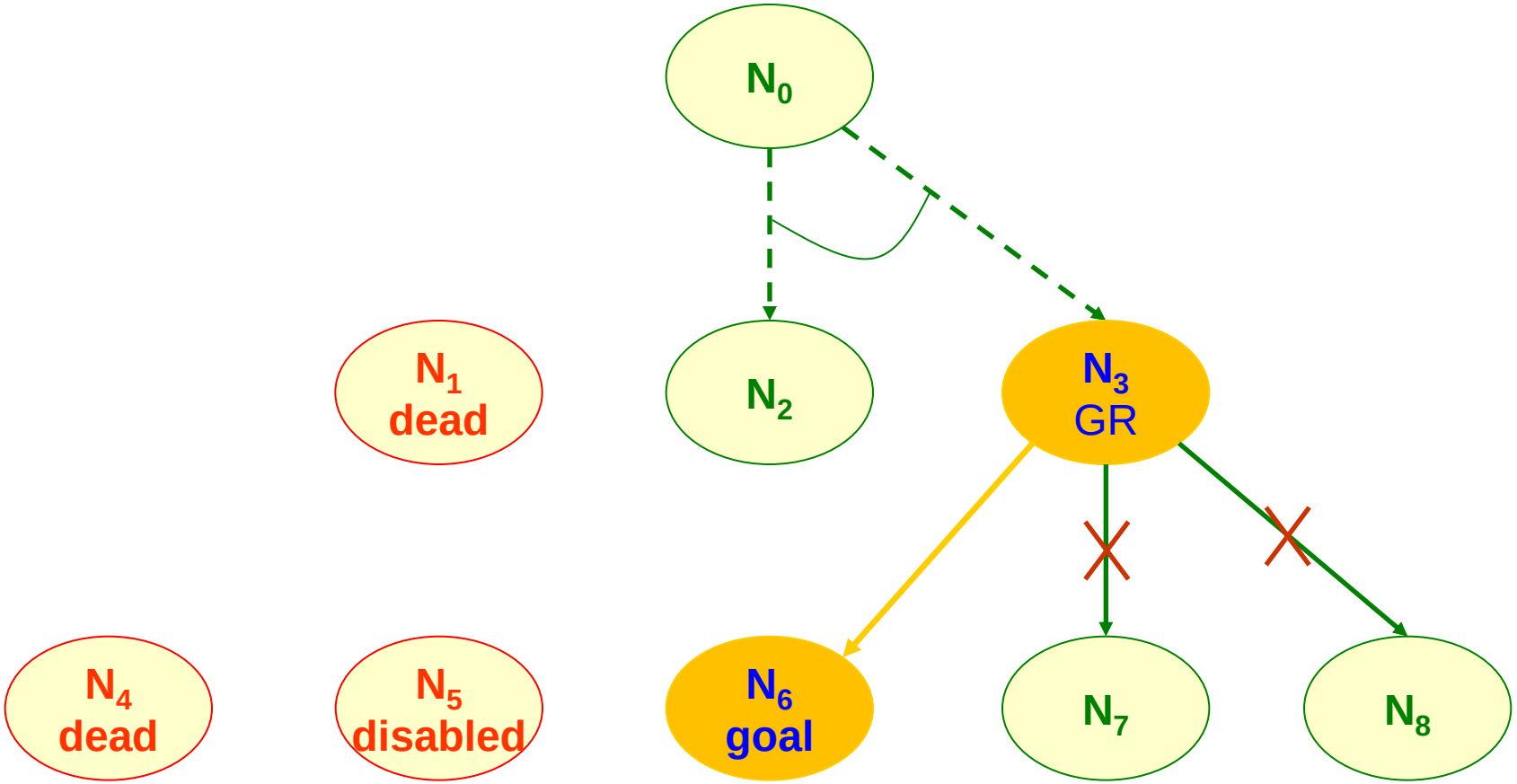
PrAO Algorithm



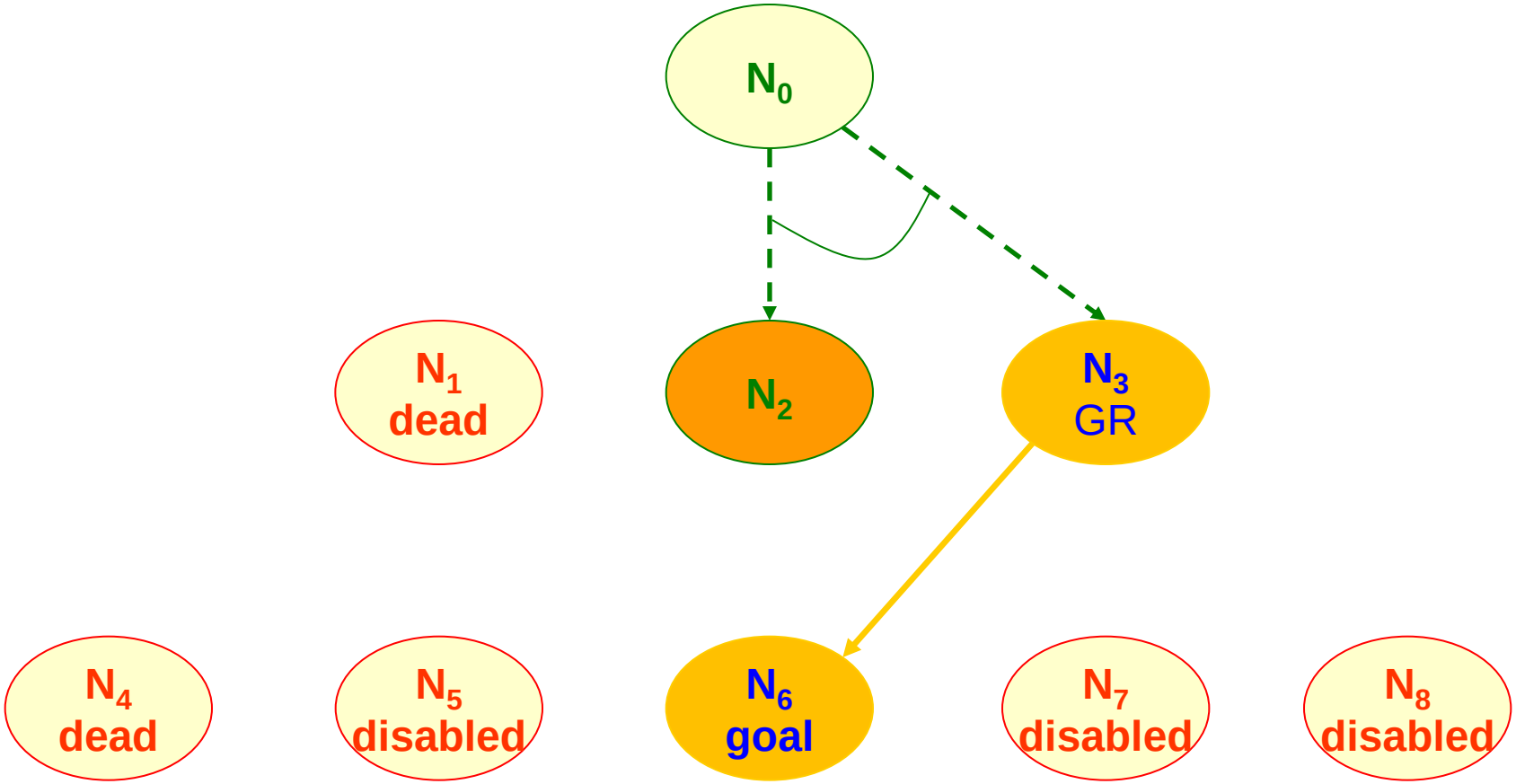
PrAO Algorithm



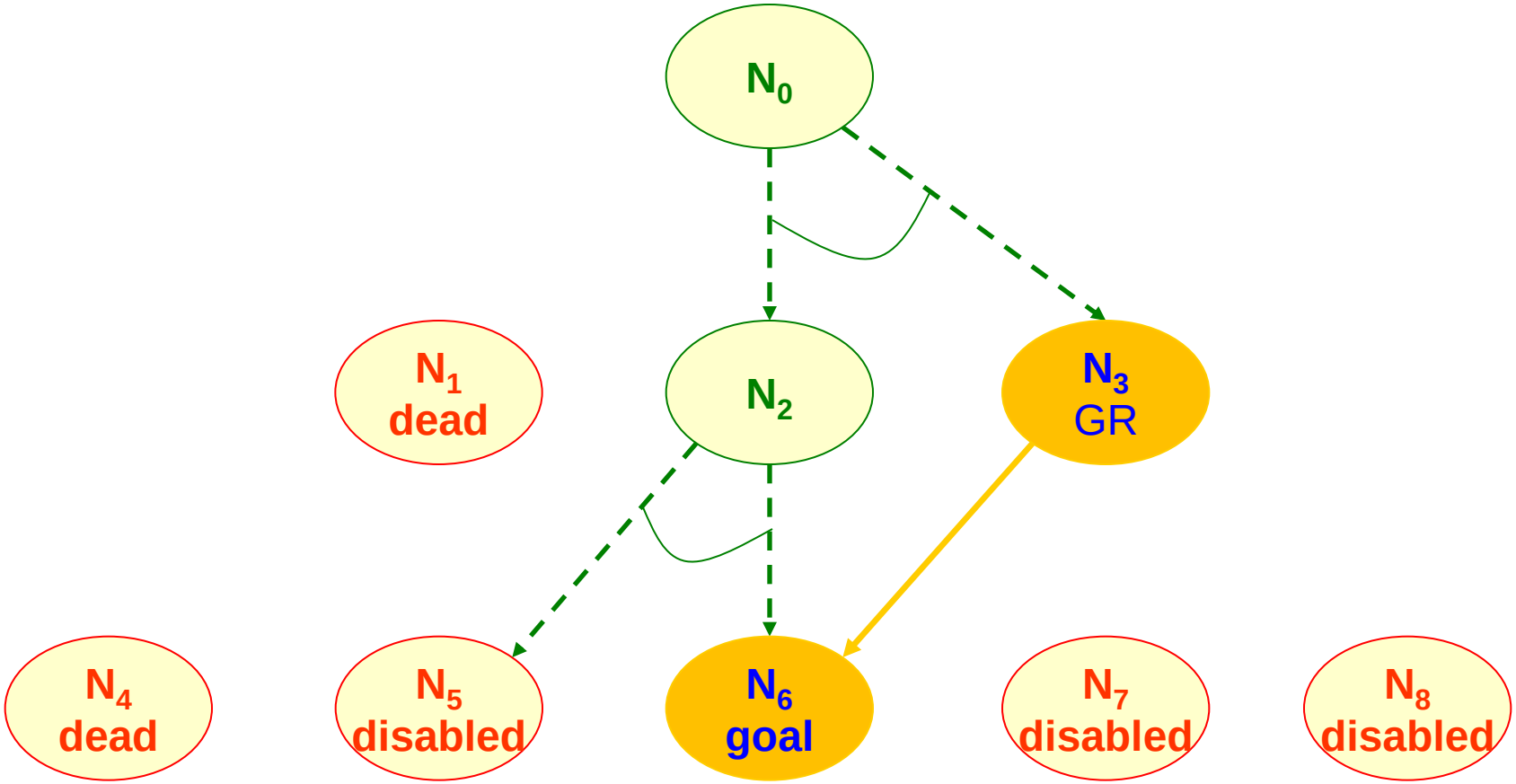
PrAO Algorithm



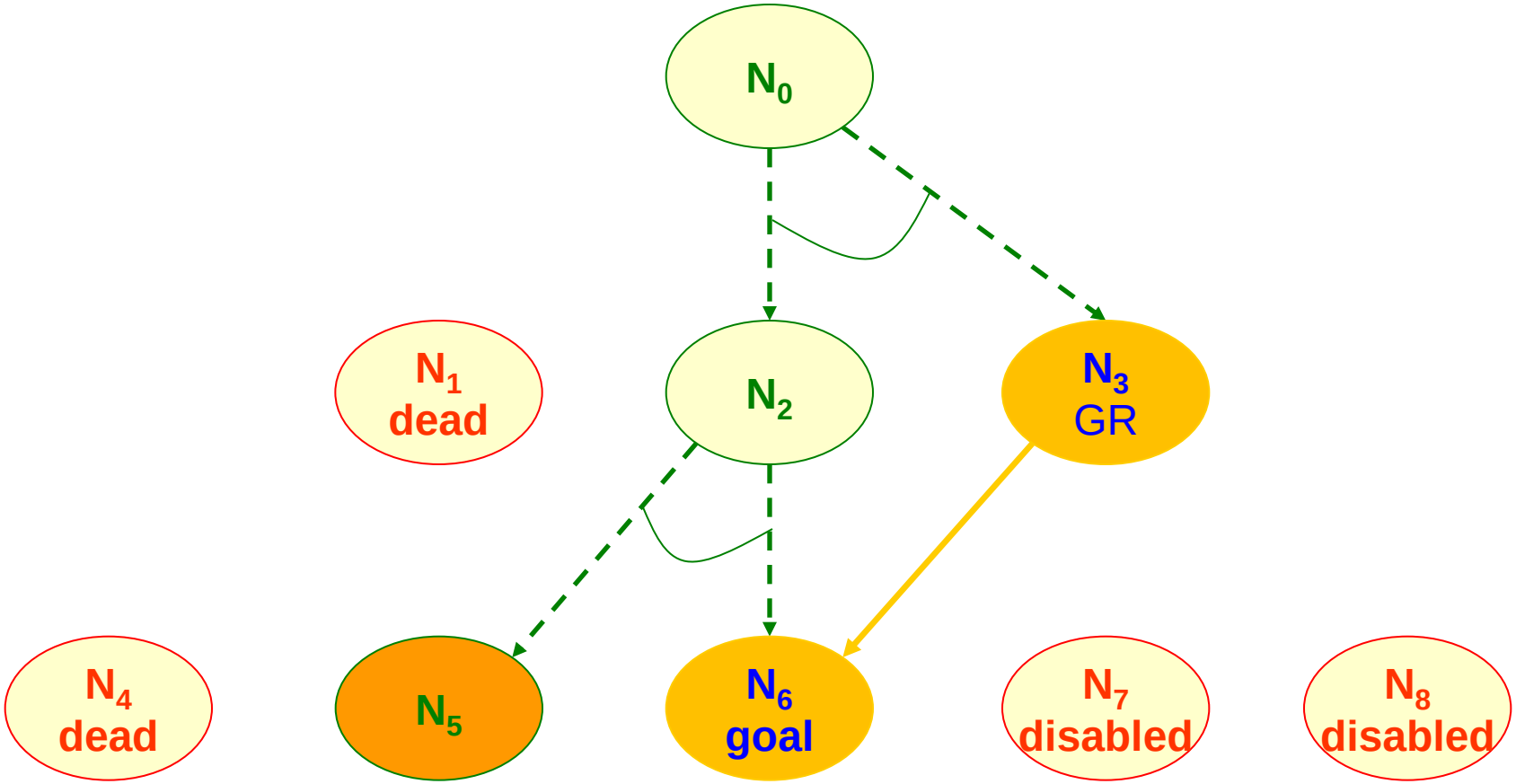
PrAO Algorithm



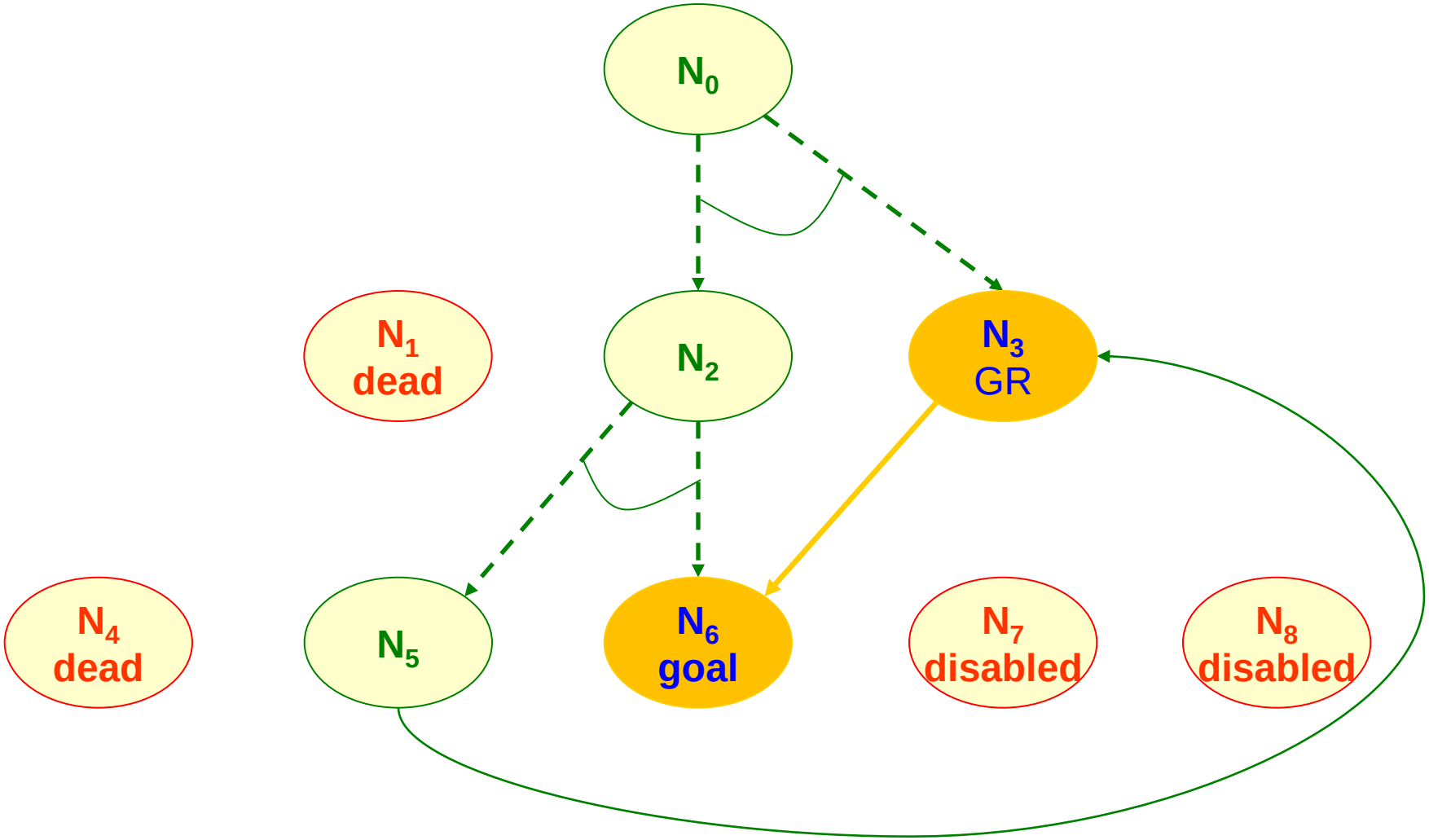
PrAO Algorithm



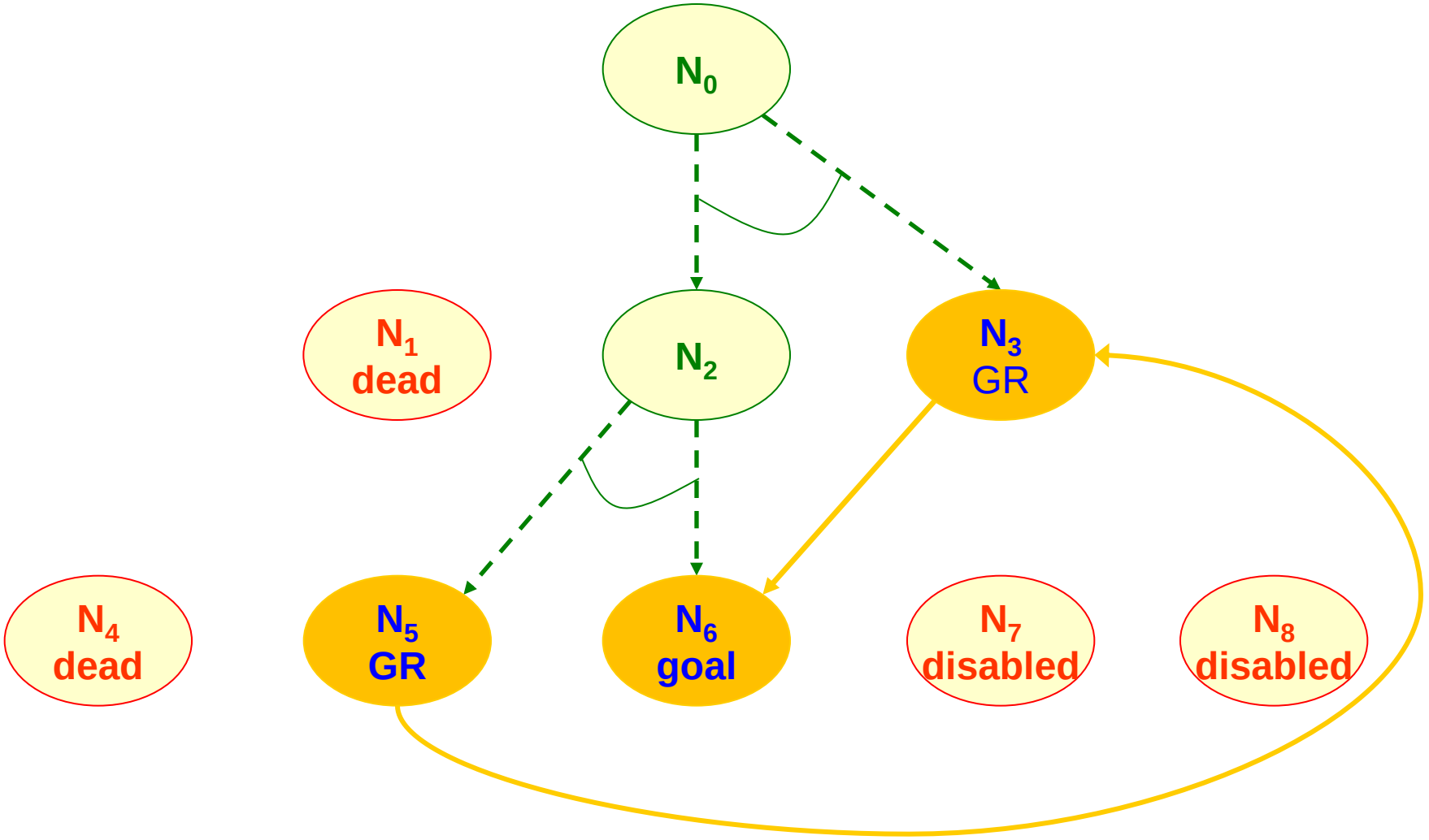
PrAO Algorithm



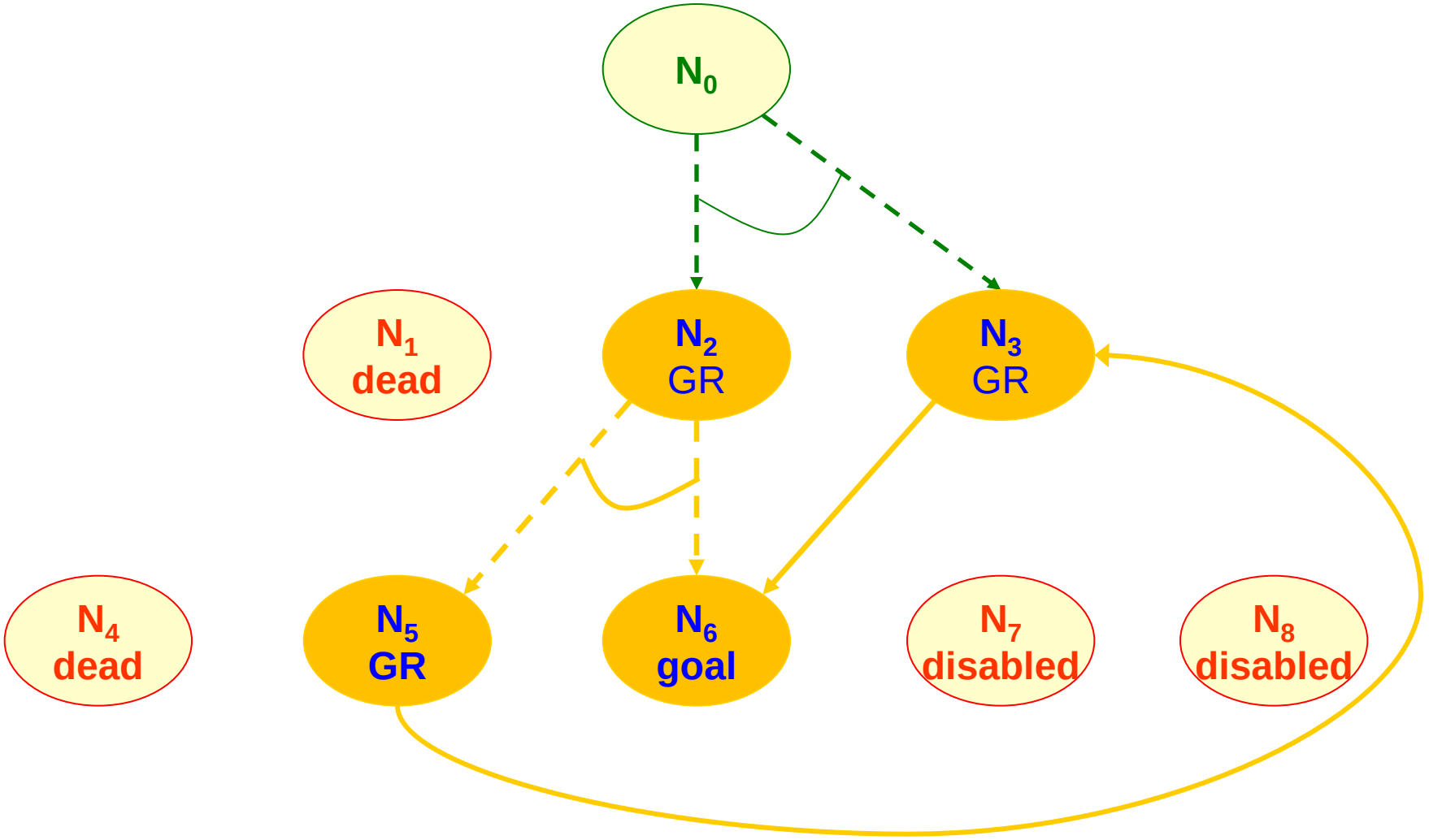
PrAO Algorithm



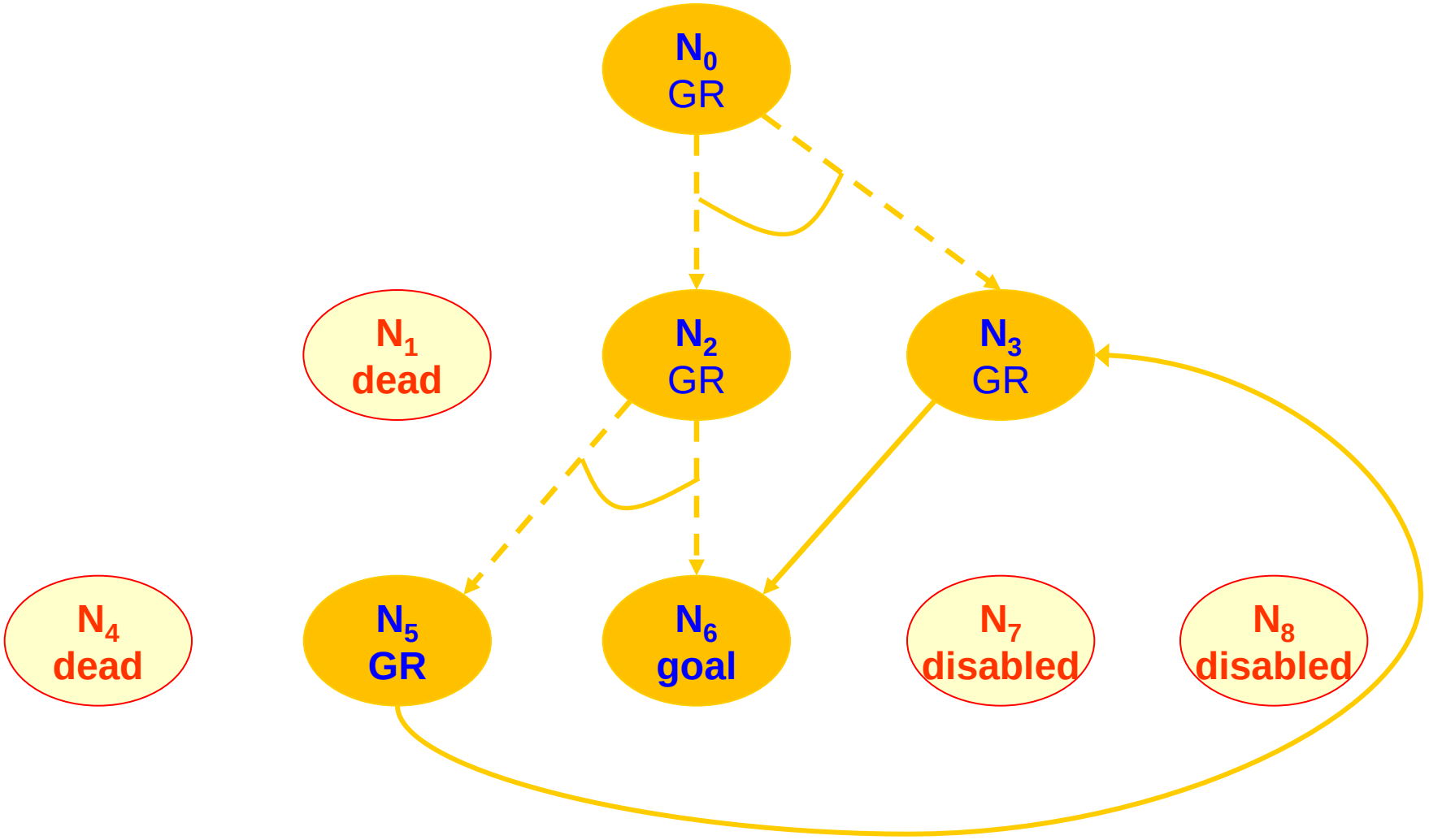
PrAO Algorithm



PrAO Algorithm



PrAO Algorithm



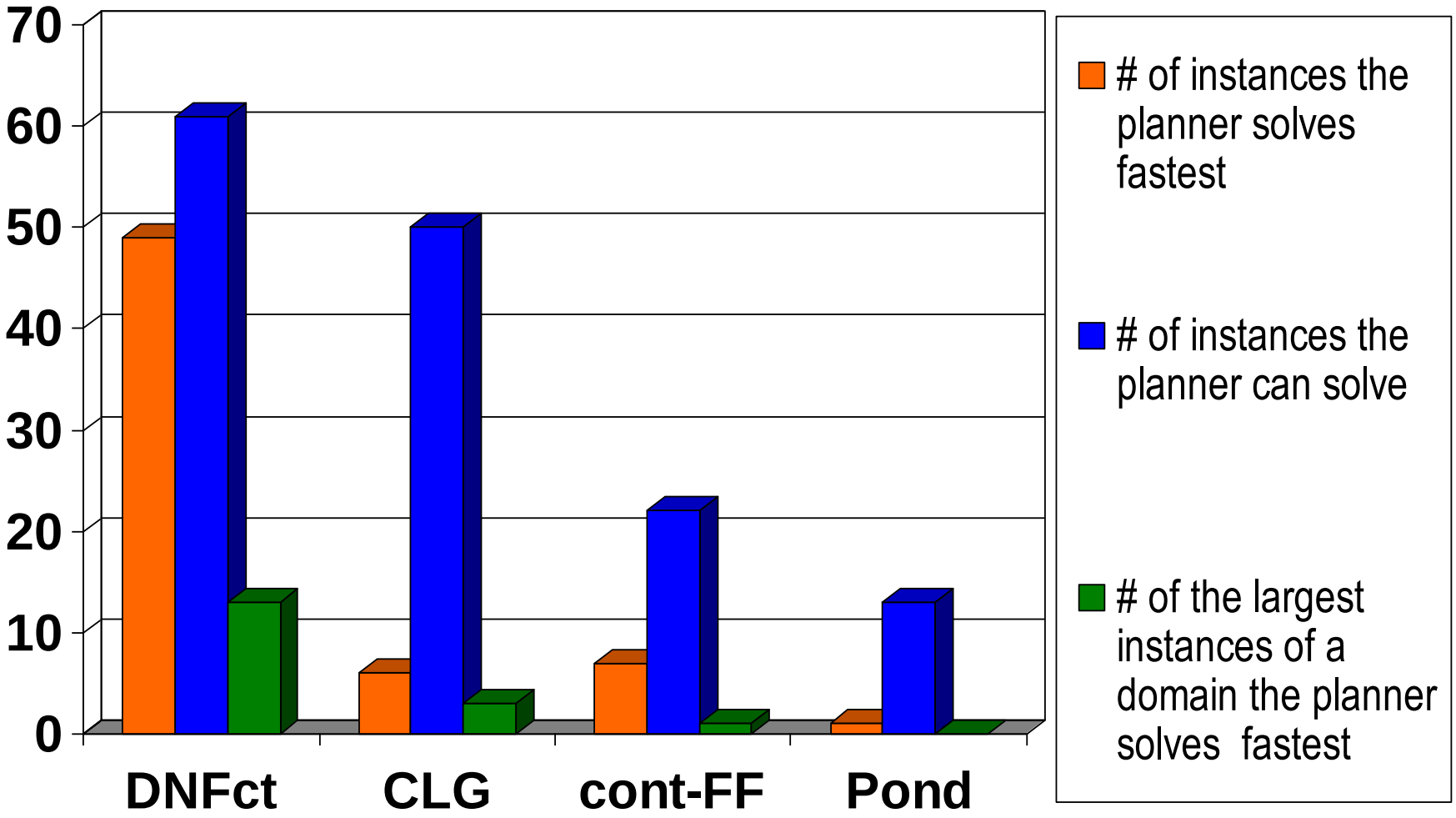
Outline

- ✓ Contingent planning
- ✓ Approach overview
- ✓DNF representation: review & extension
- ✓And/Or forward search: PrAO algorithm
- ➢ Experimental evaluation
  - •Experimental setup
  - ▪Performance comparison
- • Conclusion and future work

Experimental Setup

- Planners: compare DNF_{ct} with most state-of the-art:
  - CLG (Albore, Palacios, and Geffner IJCAI-2009)
  - Contingent-FF (Hoffmann and Brafman ICAPS-2005)
  - POND (Bryce, Kambhampati, and Smith JAIR-2006)
- Benchmarks:
  - From Pond, Contingent-FF, and CLG distribution.
  - Several obtained by our modification from conformant domains.
  - Total: 64 instances out of 17 domains.

Performance Comparison
on 64 Instances of 17 Domains



Outline

- ✓ Contingent planning
- ✓ Approach overview
- ✓ DNF representation: review & extension
- ✓ And/Or forward search: PrAO algorithm
- ✓ Experimental evaluation
- ➢ Conclusion and future work

Conclusion

- Extended DNF Representation to handle non-det & sensing actions in contingent planning:
  - Compact
  - Fast state computation
  - Complete
- Developed a new And/Or search algorithm PrAO with novel pruning techniques:
  - Avoid redundant expansion
  - Less expanded/generated nodes
  - Complete

Future Work

- DNF Representation
  - Not good when the size of DNF formulae representing belief states is too large.
- PrAO: And/Or forward search
  - In several problem instances, application of pruning techniques results in more generated/explored nodes.

Thank you!

Question?