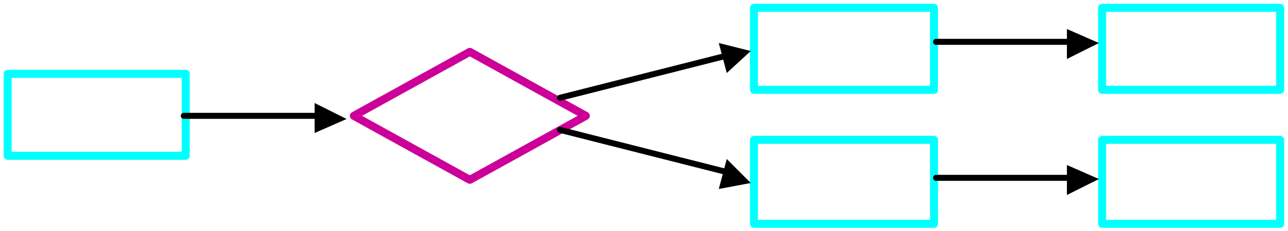
Planning, Execution & Learning
1. Conditional Planning

Reid Simmons

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)
  - Observations actions

    Observe_Road (?loc1 ?loc2)

    Pre: Unknown (Clear(?loc1, ?loc2))

    $+\alpha_1$: Clear(?loc1, ?loc2)

    $+\alpha_2$: ~Clear(?loc1, ?loc2)
  - Contexts
    - Compatible observation labels

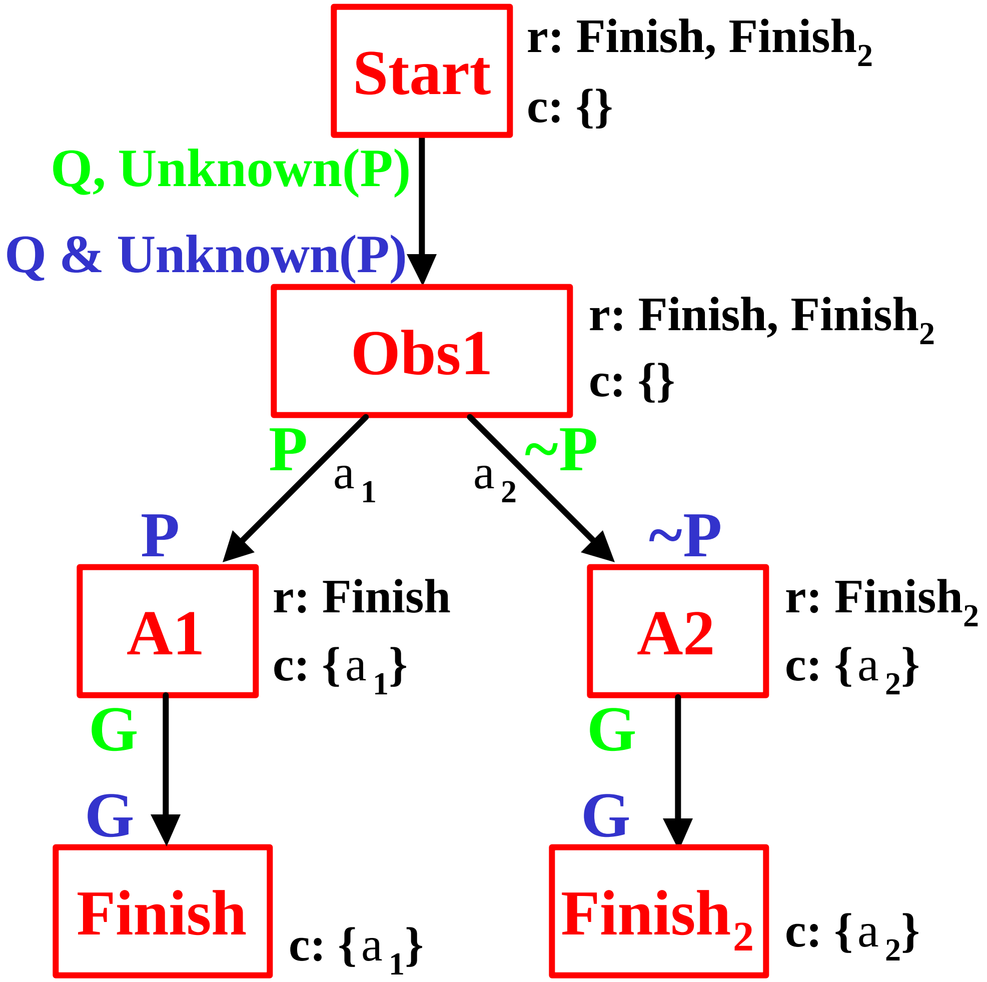
CNLP Extensions to SNLP

- ***“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

Adding Conditional Operators



Start

r: Finish, Finish_2

c: {}

Q, Unknown(P)

Q & Unknown(P)

Obs1

r: Finish, Finish_2

c: {}

P

a_1

a_2

~P

P

~P

A1

r: Finish

c: $\{a_1\}$

A2

r: Finish_2

c: $\{a_2\}$

G

G

G

G

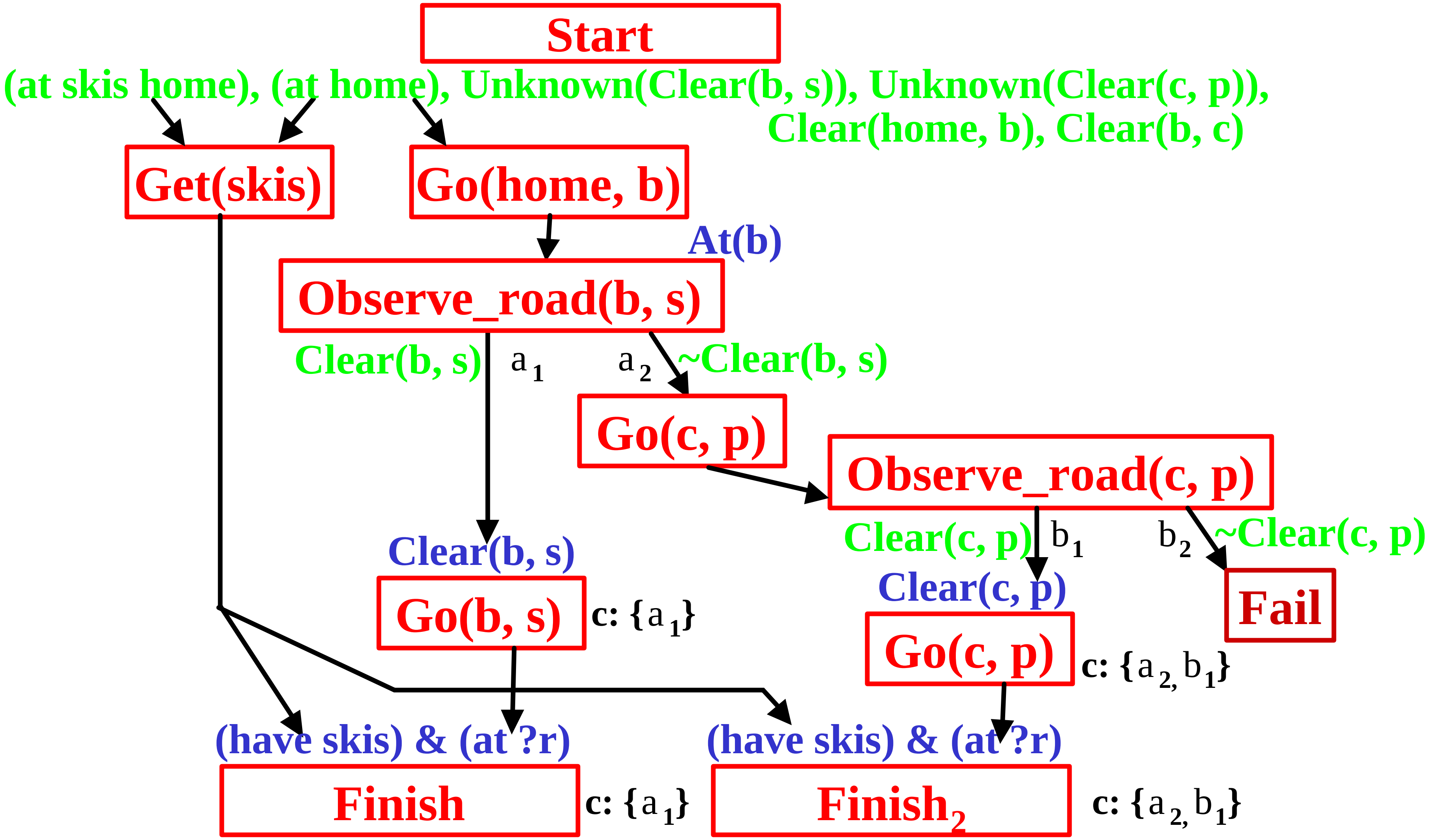
Finish

c: $\{a_1\}$

Finish_2

c: $\{a_2\}$

Conditionally Planning a Ski Trip



Start

(at skis home), (at home), Unknown(Clear(b, s)), Unknown(Clear(c, p)),
Clear(home, b), Clear(b, c)

Get(skis)

Go(home, b)

At(b)

Observe_road(b, s)

Clear(b, s) a_1

a_2 ~Clear(b, s)

Go(c, p)

Observe_road(c, p)

Clear(c, p) b_1

b_2 ~Clear(c, p)

Clear(b, s)

Go(b, s) c: {a_1}

Clear(c, p)

Fail

Go(c, p) c: {a_2, b_1}

(have skis) & (at ?r)

Finish c: {a_1}

(have skis) & (at ?r)

Finish$_2$ c: {a_2, b_1}

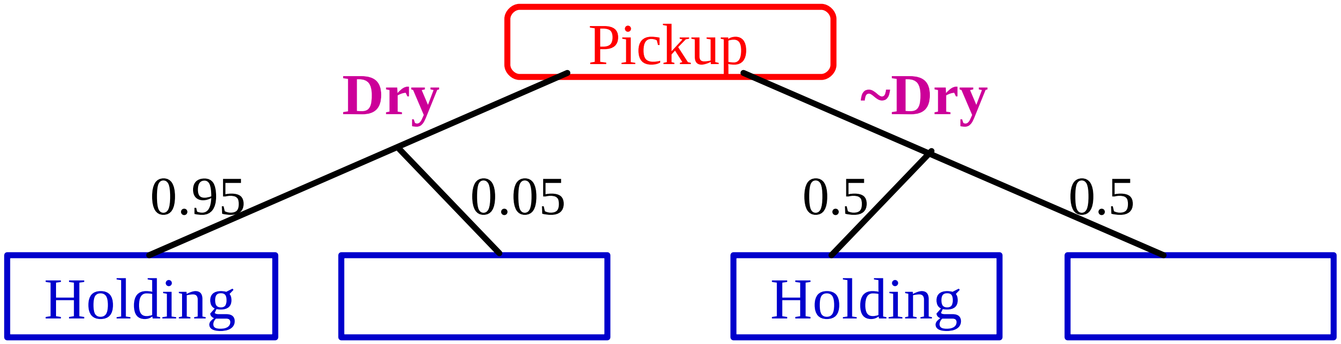
CNLP Summary

- 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

Buridan (Kushmerick, 1995)

- Plan to Achieve Goals with Probability Greater Than a Given Threshold

- Extensions to SNLP Representation
  - Probabilistic (not-deterministic) outcomes of actions
    - Conditioned on current state
    - Mutually exclusive and exhaustive ***"triggers"***
  - No preconditions (**!**)
    - Action can occur anywhere



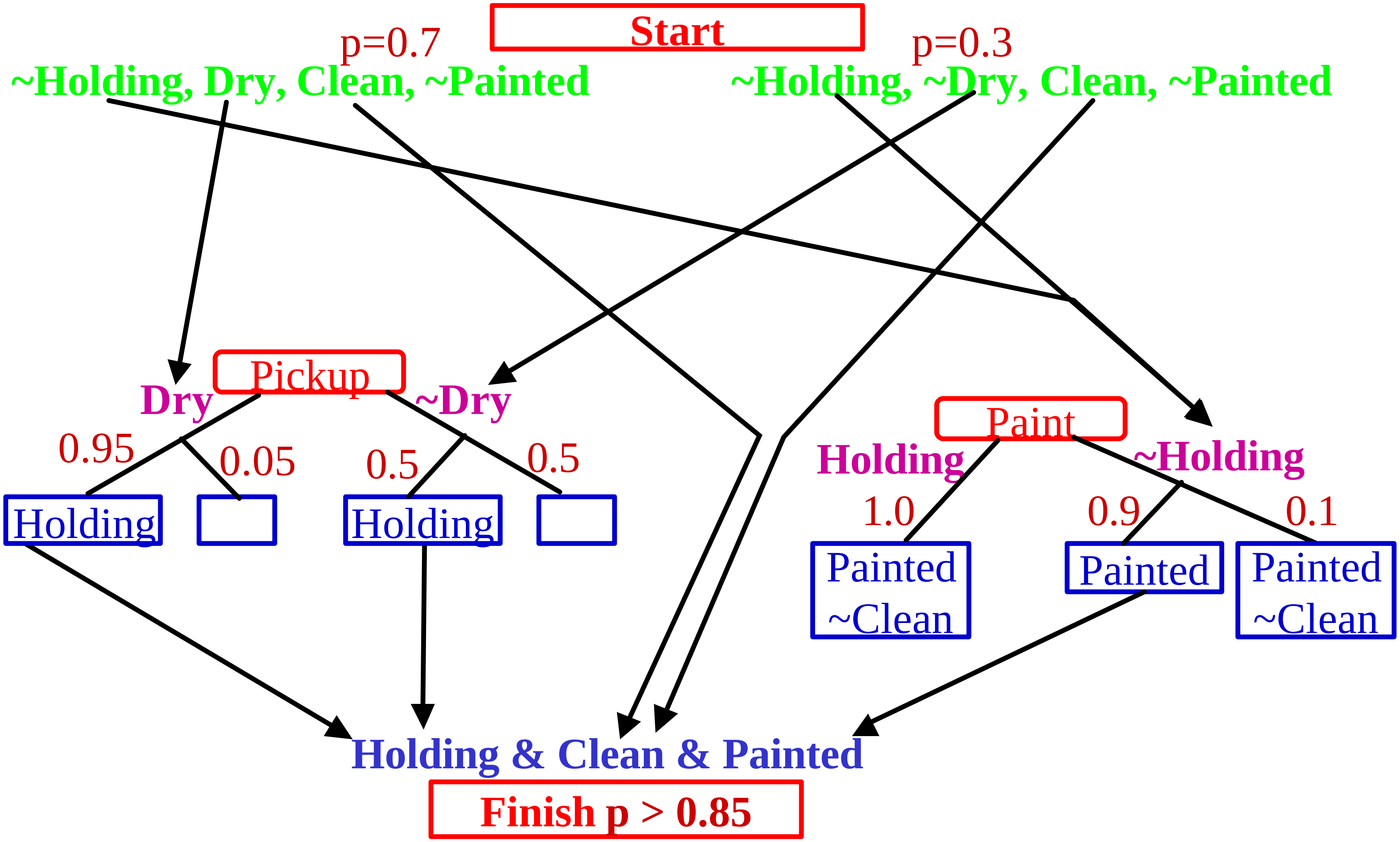
Buridan Extensions to SNLP

- Multiple Causal Links
  - Each link increases probability of achievement

- *Confrontation*
  - Reduce likelihood of threat by action A_1 by adding another action A_2 that makes it less likely for A_1 to have the undesired effect

- Plan Assessment
  - Estimate probability of plan success
  - NP-hard, in general

Buridan (Partial) Plan



Start

p=0.7

~Holding, Dry, Clean, ~Painted

p=0.3

~Holding, ~Dry, Clean, ~Painted

Dry

Pickup

~Dry

0.95

0.05

0.5

0.5

Holding

Holding

Paint

Holding

~Holding

1.0

0.9

0.1

Painted ~Clean

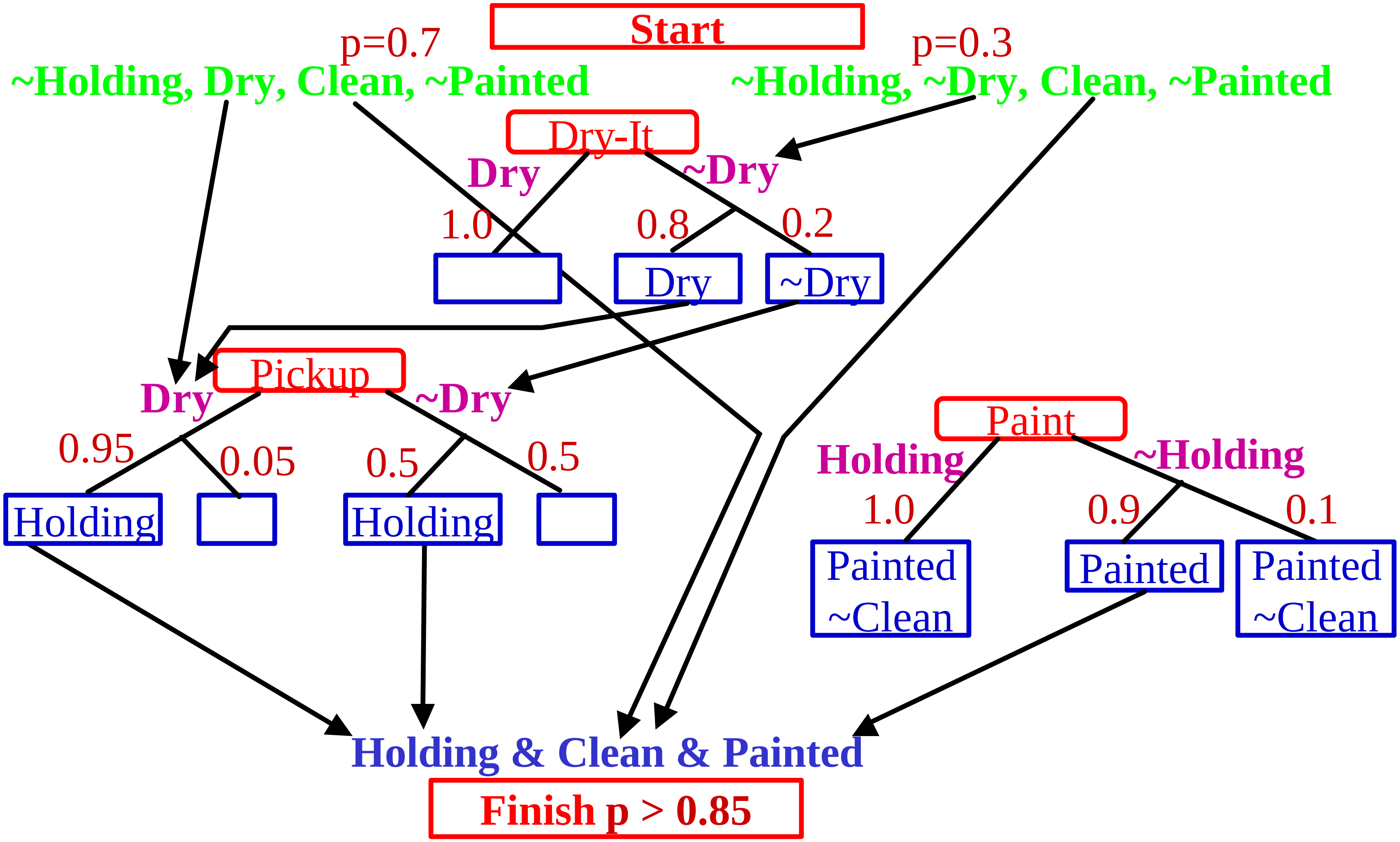
Painted

Painted ~Clean

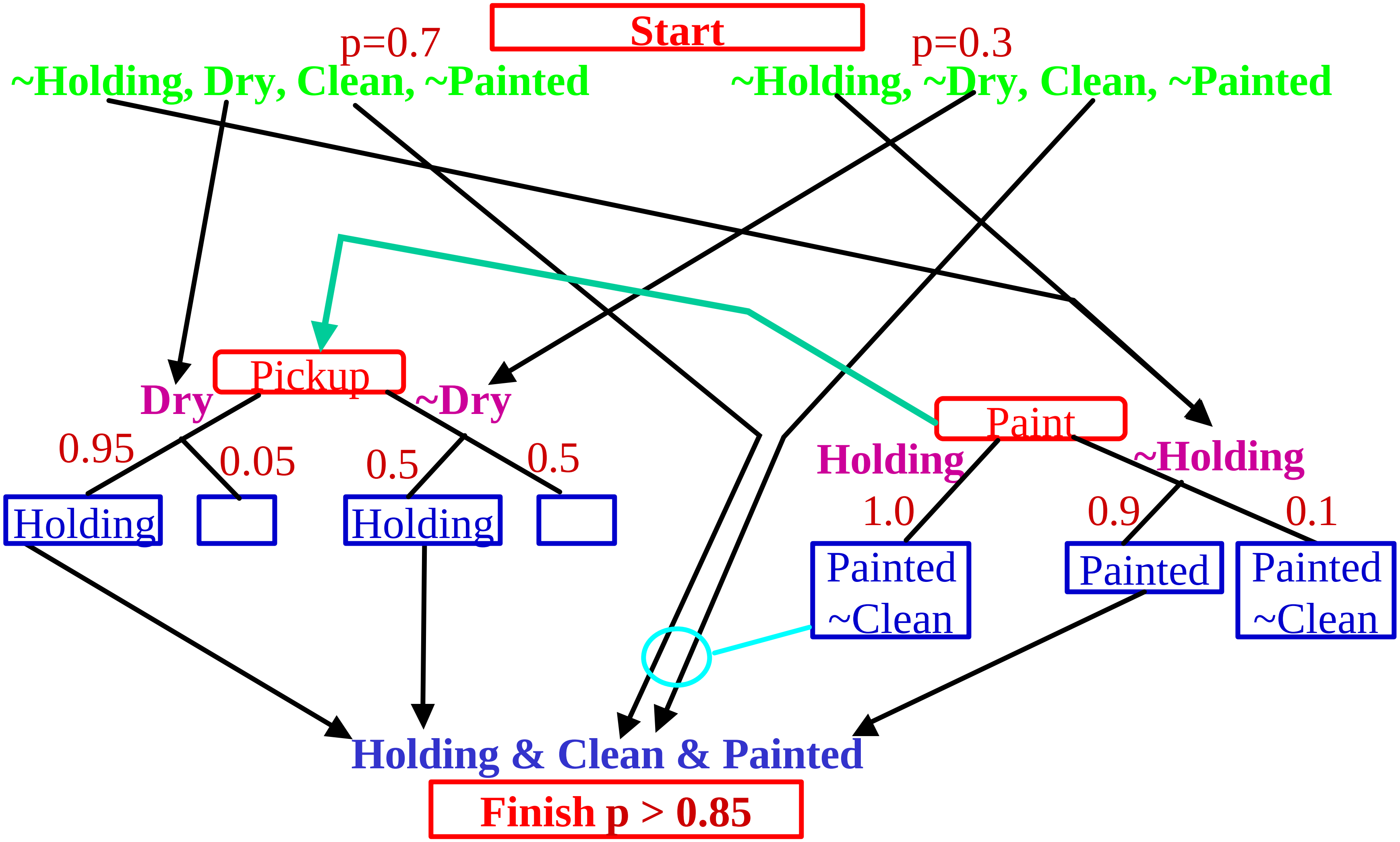
Holding & Clean & Painted

Finish p > 0.85

Increasing Probability of Success



Confronting a Threat



Assessing the Plan

Goal: pr(Holding & Clean & Painted) > 0.85

Initial: {(Dry, ~Holding, Clean, Painted), 0.7)
(~Dry, ~Holding, Clean, Painted), 0.3)}

Paint: {(Dry, ~Holding, Clean, Painted), 0.63)
(Dry, ~Holding, ~Clean, Painted), 0.07)
(~Dry, ~Holding, Clean, Painted), 0.27)
(~Dry, ~Holding, ~Clean, Painted), 0.03)}

Dry-It: {(Dry, ~Holding, Clean, Painted), 0.63)
(Dry, ~Holding, ~Clean, Painted), 0.07)
(Dry, ~Holding, Clean, Painted), 0.216)
(~Dry, ~Holding, Clean, Painted), 0.054)
(Dry, ~Holding, ~Clean, Painted), 0.024)
(~Dry, ~Holding, ~Clean, Painted), 0.006)}

{(Dry, ~Holding, Clean, Painted), 0.846)
(Dry, ~Holding, ~Clean, Painted), 0.094)
(~Dry, ~Holding, Clean, Painted), 0.054)
(~Dry, ~Holding, ~Clean, Painted), 0.006)}

Pickup: {(Dry, Holding, Clean, Painted), 0.8037)
(Dry, ~Holding, Clean, Painted), 0.0423)
(Dry, Holding, ~Clean, Painted), 0.0893)
(Dry, ~Holding, ~Clean, Painted), 0.0047)
(~Dry, Holding, Clean, Painted), 0.0513)
(~Dry, ~Holding, Clean, Painted), 0.0027)
(~Dry, Holding, ~Clean, Painted), 0.0057)
(~Dry, ~Holding, ~Clean, Painted), 0.0003)}

Buridan Summary

- Handles Probabilistic Actions
  - Outcomes conditioned on current state and random chance
- Different Notion of Plan Success
  - Probability of achieving goal greater than threshold
  - Adds multiple actions to increase probability
- No Observational Actions
  - ***Not a conditional planner***

C-Buridan (Draper, 1994)

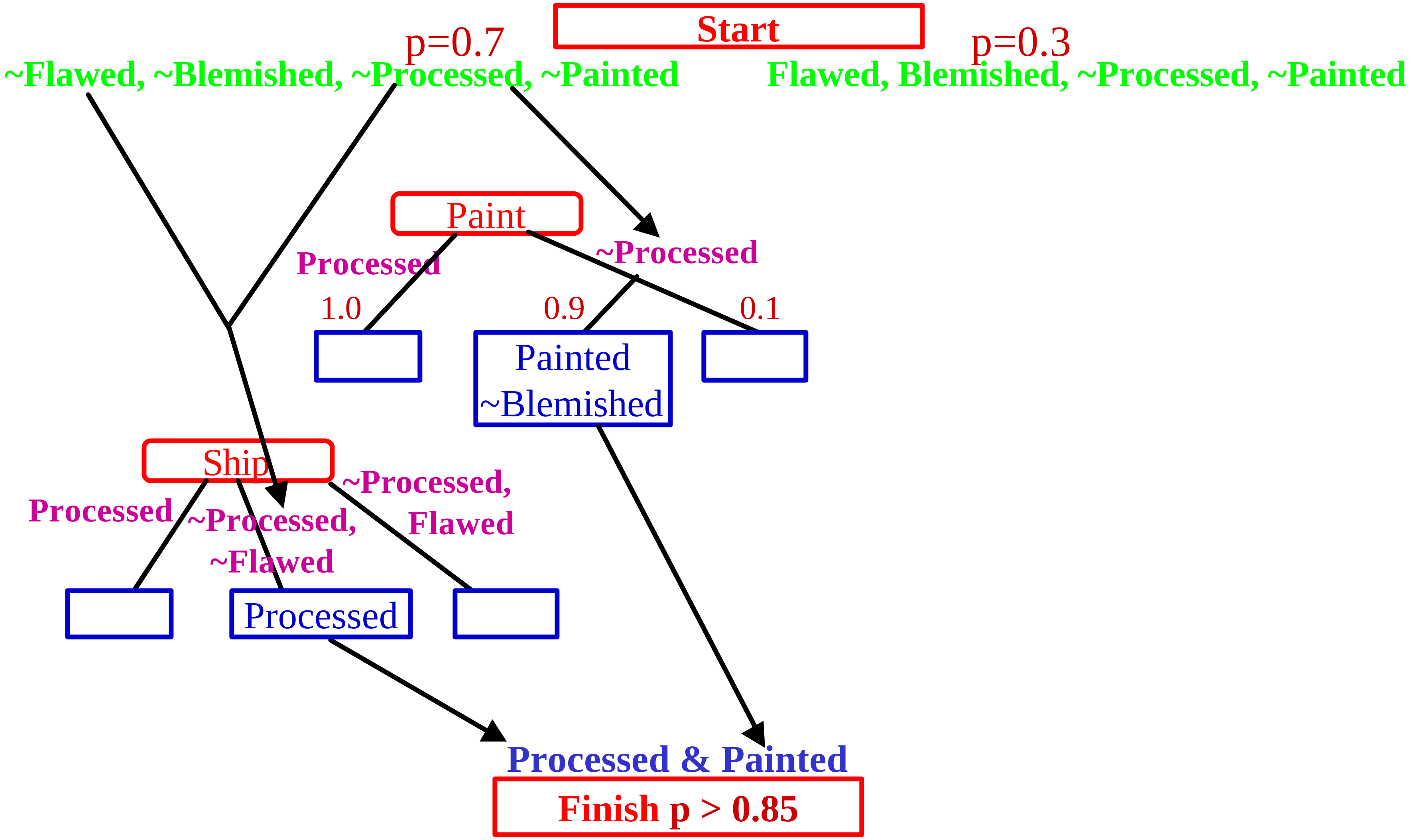
- Conditional, Partial-Order Planner

- Extensions to Buridan
  - Representation: Observation labels on actions
    - Clear distinction between effects and observations
    - Models noisy sensors
  - Algorithm: ***Conditioning*** (***branching***) to remove threats
    - Add observation actions to separate contexts
    - Propagate context labels

Differences from CNLP

- Branches can Rejoin
  - Plans are DAG's
- Branch Added Only to Remove Threat
  - Not really "planning to observe"
- No a priori Relationship Between Observation Labels and Propositions
  - Planner must "discover" correlations

Conditionally Processing Widgets (I)



Start

p=0.7

~Flawed, ~Blemished, ~Processed, ~Painted

p=0.3

Flawed, Blemished, ~Processed, ~Painted

Paint

Processed

~Processed

1.0

0.9

0.1

Painted
~Blemished

Ship

Processed

~Processed,
~Flawed

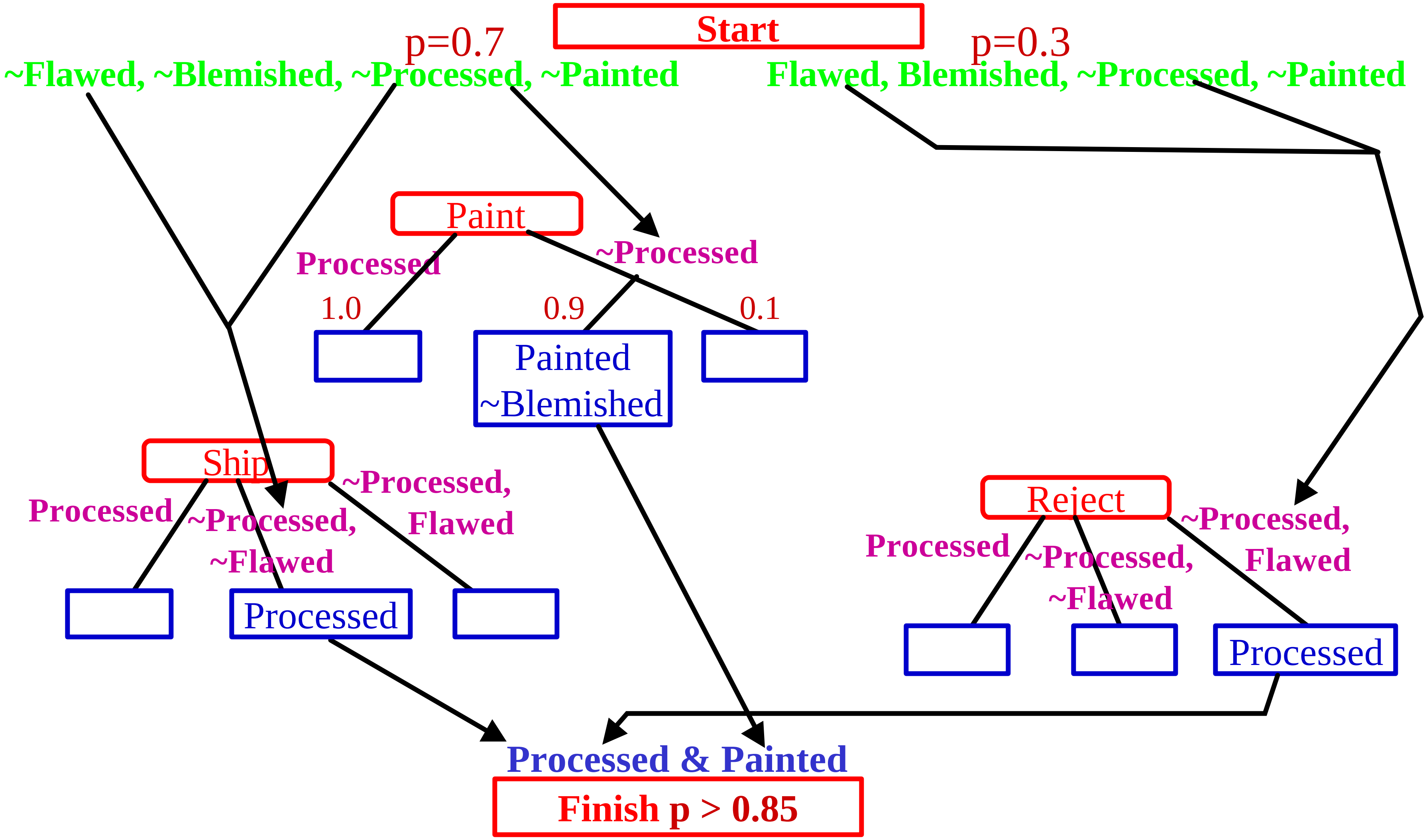
~Processed,
Flawed

Processed

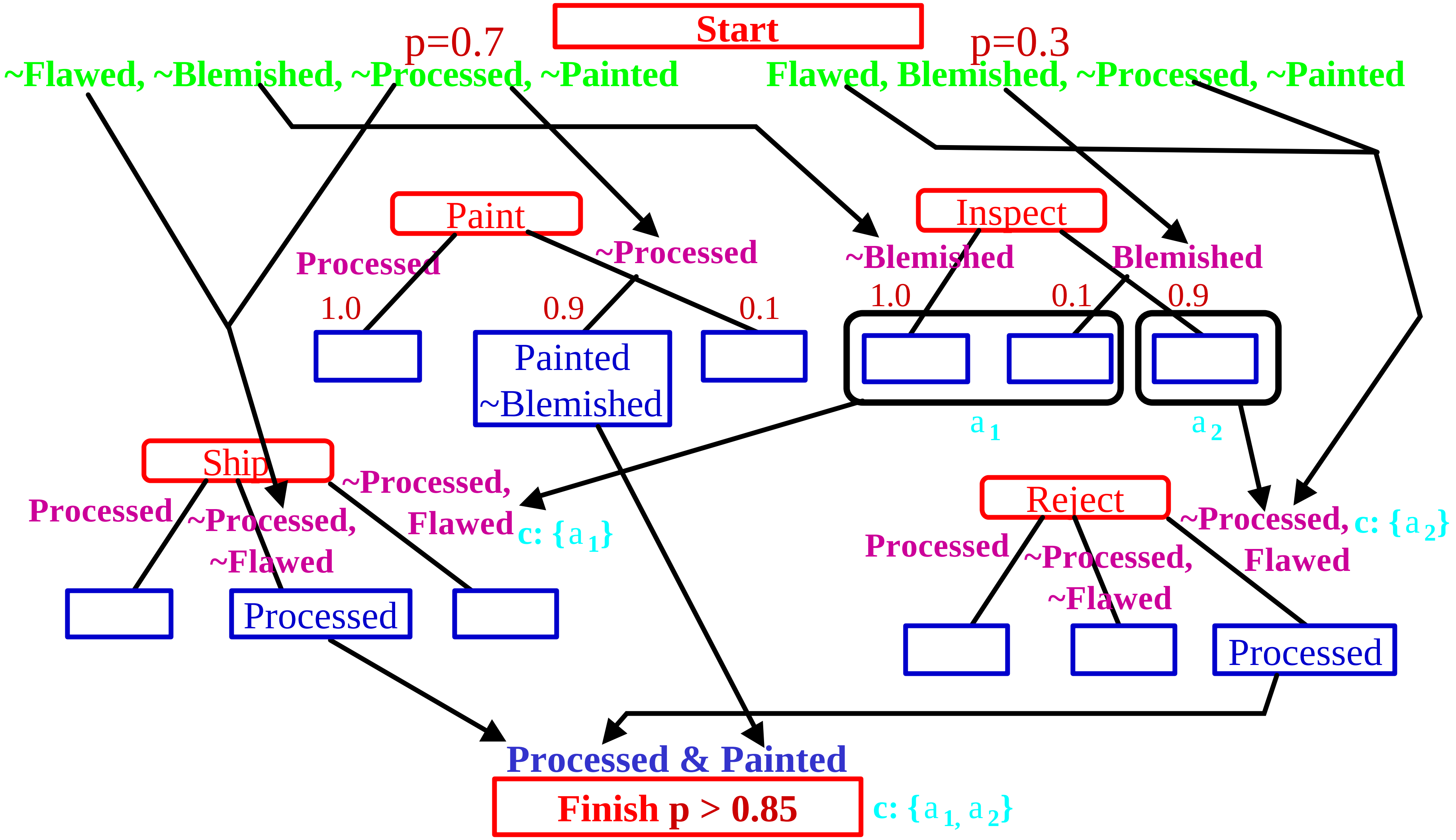
Processed & Painted

Finish p > 0.85

Conditionally Processing Widgets (II)



Conditionally Processing Widgets (III)



Start
p=0.7
~Flawed, ~Blemished, ~Processed, ~Painted
p=0.3
Flawed, Blemished, ~Processed, ~Painted
Paint
Processed
~Processed
1.0
0.9
0.1
Painted
~Blemished
Inspect
~Blemished
Blemished
1.0
0.1
0.9
a 1
a 2
Ship
Processed
~Processed,
~Flawed
~Processed,
Flawed
c: {a 1}
Processed
Reject
Processed
~Processed,
~Flawed
~Processed,
Flawed
c: {a 2}
Processed
Processed & Painted
Finish p > 0.85
c: {a 1, a 2}