



PLANET International Summer School  
On AI Planning 2002

# Planning and Execution

Martha E. Pollack

University of Michigan

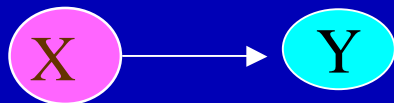
[www.eecs.umich.edu/~pollackm](http://www.eecs.umich.edu/~pollackm)

# Today's Plan

- Brief Review of Lecture 1 (no slides)
  - More Expressive Constraint-Based Temporal Problems (pages H-26 – H-29)
  - Handling Potential Plan Failures (pages H-30 – H-48)
  - *[[Deliberation Management (pages H-49 – H-59)]]*
  - Conclusions (page H-60)
- 
- Note: List of references by topic on pages H-60 – H 63

# Handling Temporal Uncertainty

- TP-u (e.g., STP-u)
- Distinguish between two kinds of events:
  - Controllable: the executing agent controls the time of occurrence
  - Uncontrollable: “nature” controls the time of occurrence



Controllable edge (Y controllable event)

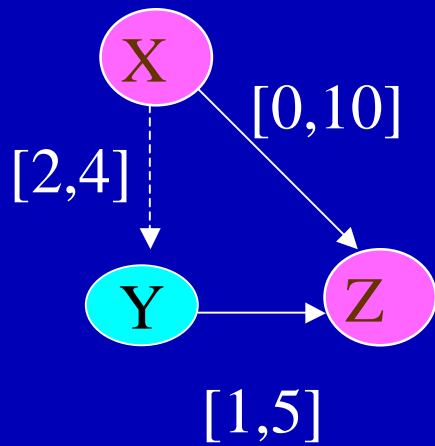


Uncontrollable edge (Y uncontrollable event)

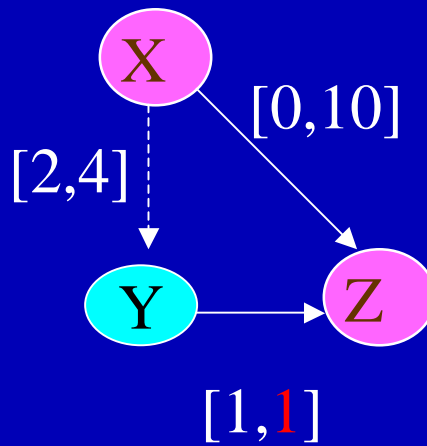
# Three Notions of “Solution”

- *Strongly Controllable*: There is an assignment of time points to the controllable events such that the constraints will be satisfied regardless of when the uncontrollables occur.
- *Weakly Controllable*: For each outcome of the uncontrollables, there is an assignment of time points to the controllables such that the constraints are satisfied.
- *Dynamically Controllable*: As time progresses and uncontrollables occur, assignments can be made to the controllables such that the constraints are satisfied.

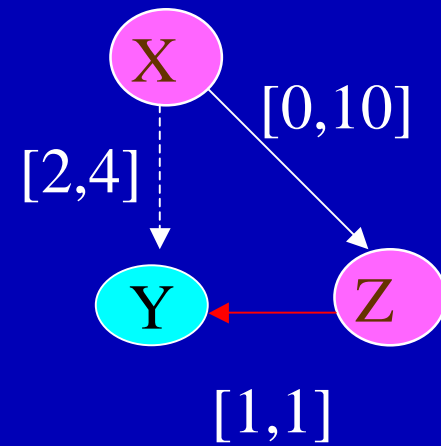
# Controllability in STP-u's



Strongly Controllable  
 $\{X=0, Z=5\}$



Dynamically Controllable  
 $\{X=0, Z=Y+1\}$

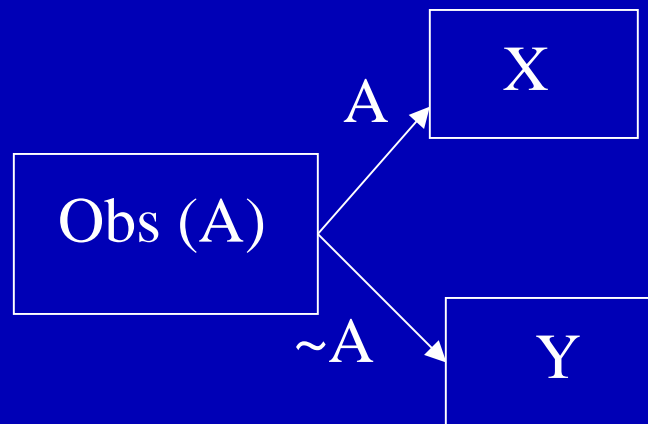


Weakly Controllable  
 $\{X=0, Z=Y-1\}$

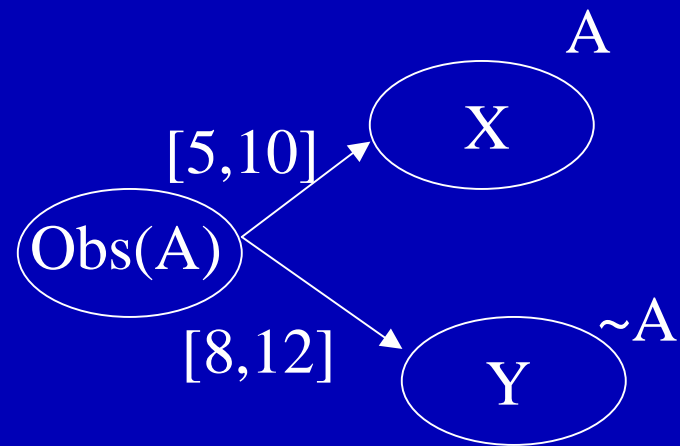
Strong  $\Rightarrow$  Dynamic  $\Rightarrow$  Weak

# Handling Causal Uncertainty

- CTP (e.g., CSTP)
- Label each node—events are executed only if their associated label is true (at a specified observation time)

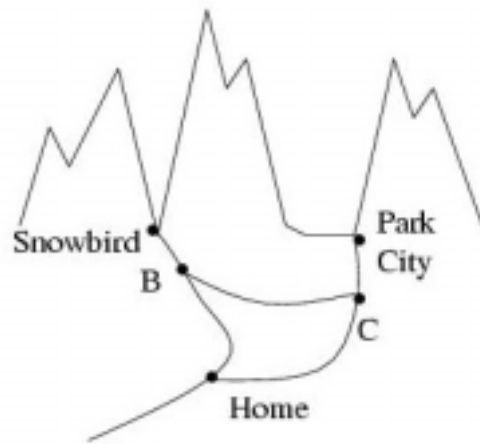


Conditional Plan



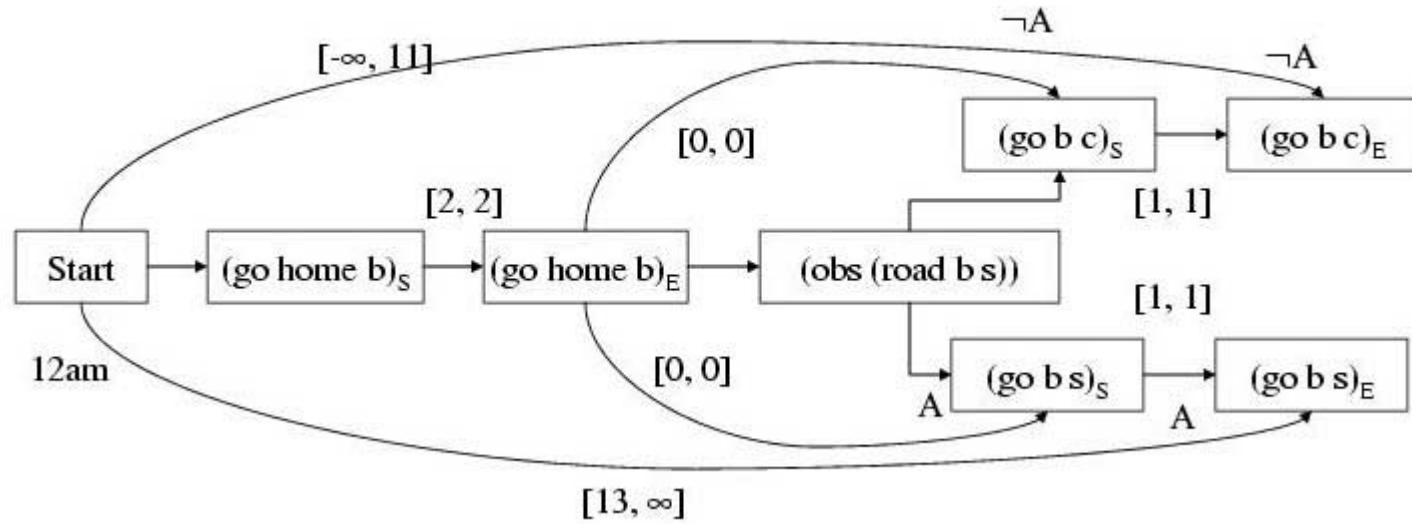
CTP

# Conditional Plan as CTP

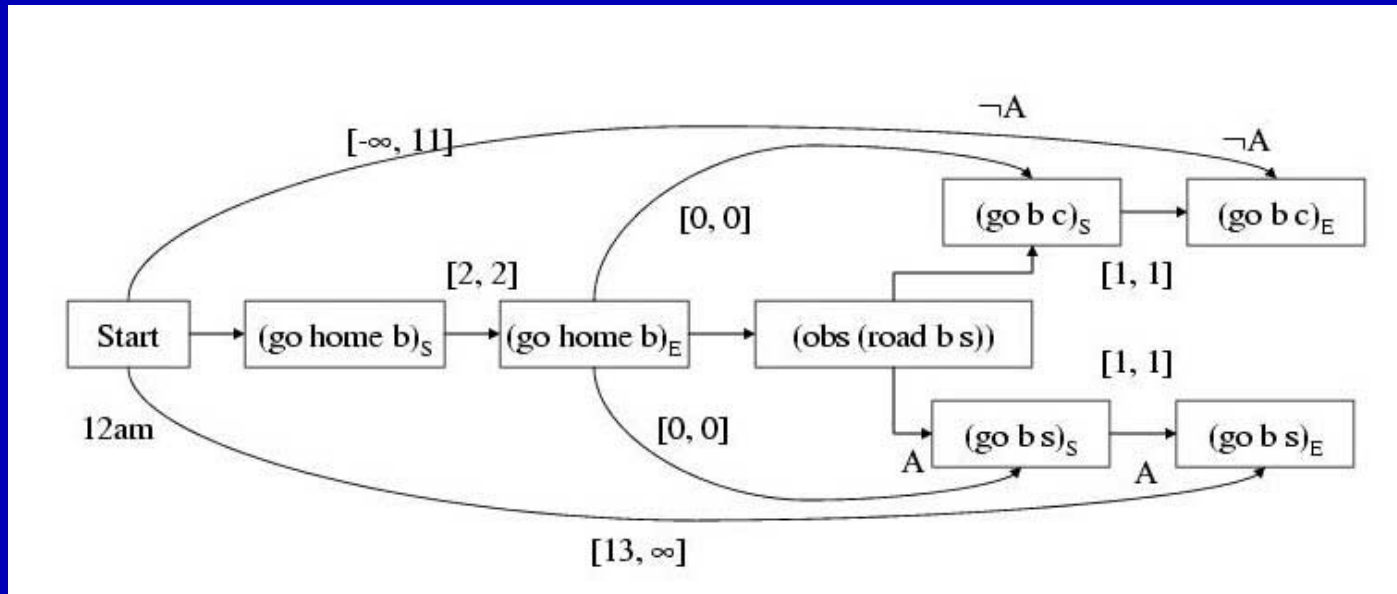


Travel from Home to S, but if the road is blocked from B to S, go to P.  
 If you go to S, arrive after 1p.m. (to take advantage of the discounts).  
 If you go to P, arrive at C by 11 a.m. (because traffic gets heavy).

B-> S  
 B-> C  
 1 hr.

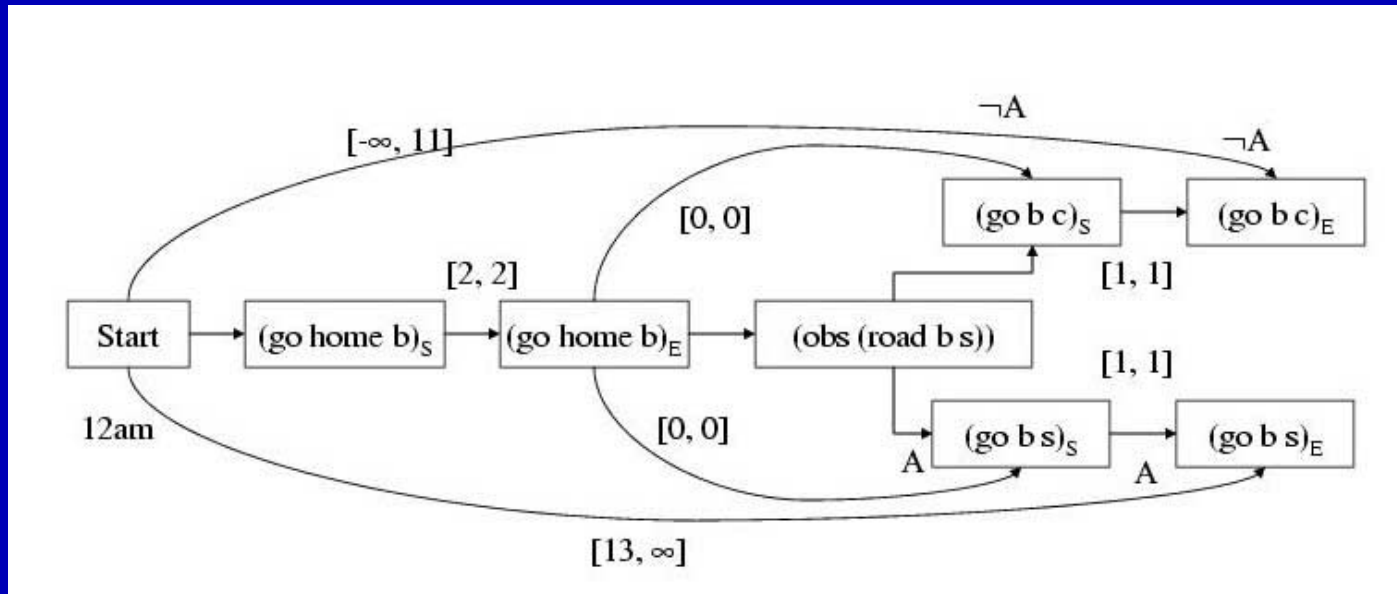


# Strong Consistency



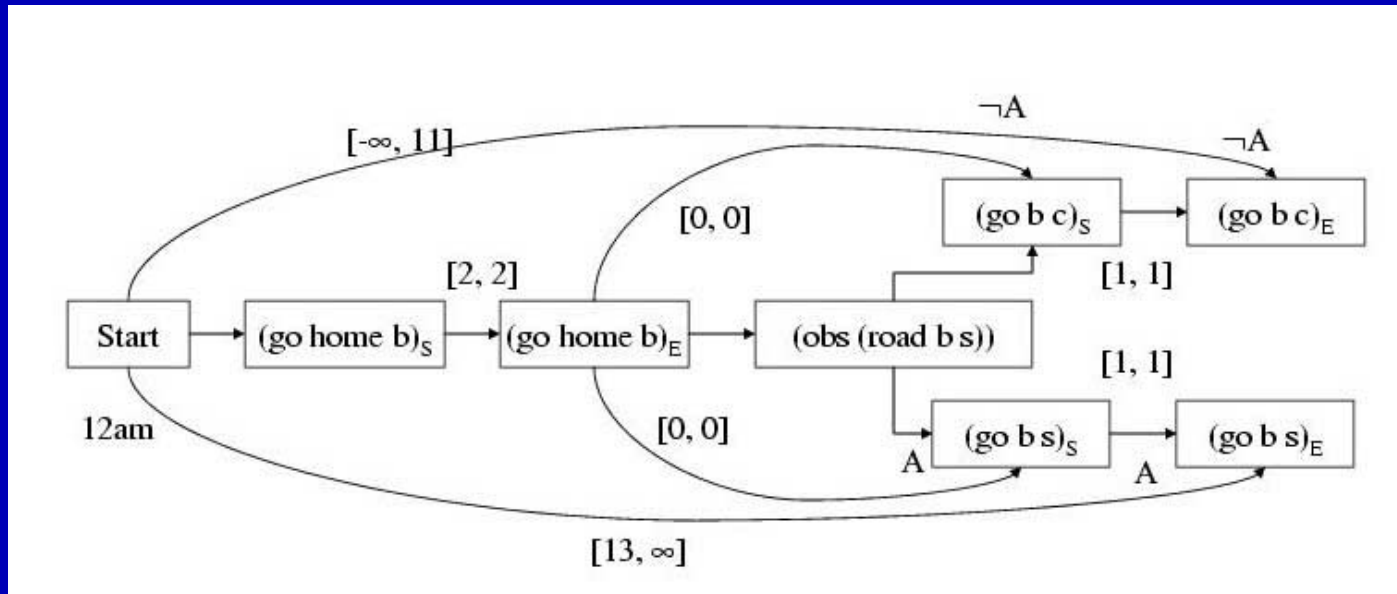
- **Not strongly consistent**: Must not be at B before 12 (if A is true); must be at B by 10 (if A is false)—and can't observe A until you're at B.

# Weak Consistency



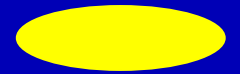
- Weakly consistent: When  $A$  is true, leave home after 10 (and all other assignments directly follow). When  $A$  is false, leave home by 9.

# Dynamic Consistency



- Not dynamically consistent: Can't tell when you need to leave home until it's too late.
- Variant that is dynamically consistent: Add a parking lot at B where you can wait.

# Planning and Execution



- So far: Execution Dispatch
  - Well-formed problems
  - Precise solutions that cohere
- This time: Planning and Execution
  - More open-ended questions
  - Partial answers
  - *Opportunity for lots of good research!*

# Problem Characteristics

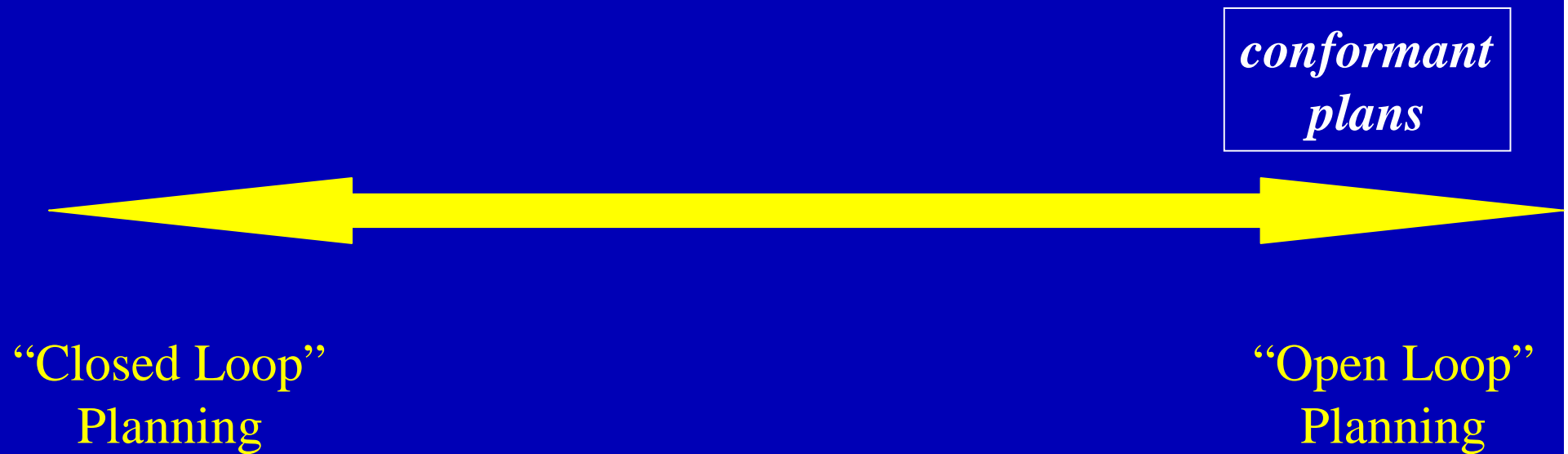
## Classical planning:

- World is static (and therefore single agent).
- Actions are deterministic.
- Planning agent is omniscient.
- All goals are known at the outset.
- *Consequently*, everything will “go as planned.”

## But in general:

- World is dynamic and multi-agent
- Actions have uncertain outcomes.
- Planning agent has incomplete knowledge.
- New planning problems arrive asynchronously
- So, things may not go as planned!

# When Plans May Fail...



# Conformant Planning

- Construct a plan that will work regardless of circumstances
  - Sweep a bar across the desk to clear it
  - Paint both the table and chair to ensure they're the same color
- Without any sensors, may be the best you can do
- In general, conformant plans may be costly or non-existent

# When Plans May Fail...

*universal  
plans*

*conformant  
plans*



“Closed Loop”  
Planning

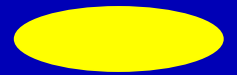
“Open Loop”  
Planning

# Universal Plans

[Schoppers]

- Construct a complete function from states to actions
- Observe state—take one step—loop
- Essentially follow a decision tree
- Assumes you can completely observe state
- May be a huge number of states!

# When Plans May Fail...



*conditional  
plans*

*MDPs*

*universal  
plans*

*probabilistic  
plans*

*conformant  
plans*

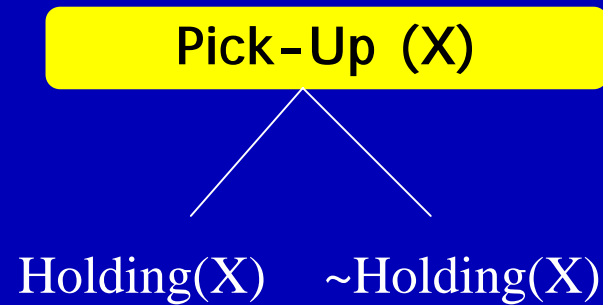


“Closed Loop”  
Planning

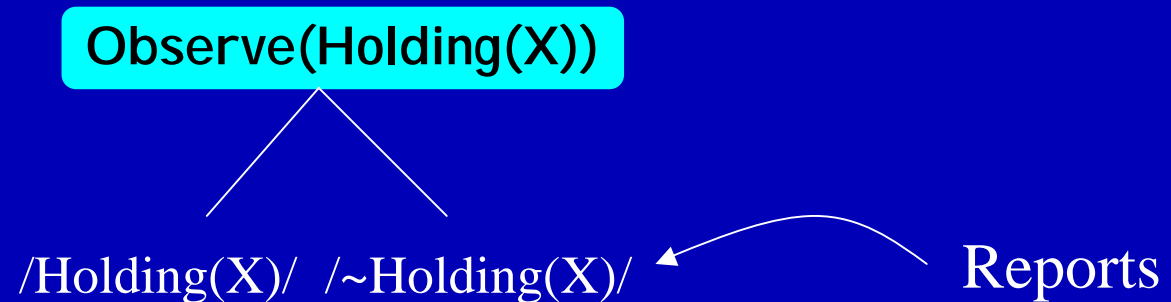
“Open Loop”  
Planning

# Conditional Planning

- Some causal actions have alternative outcomes



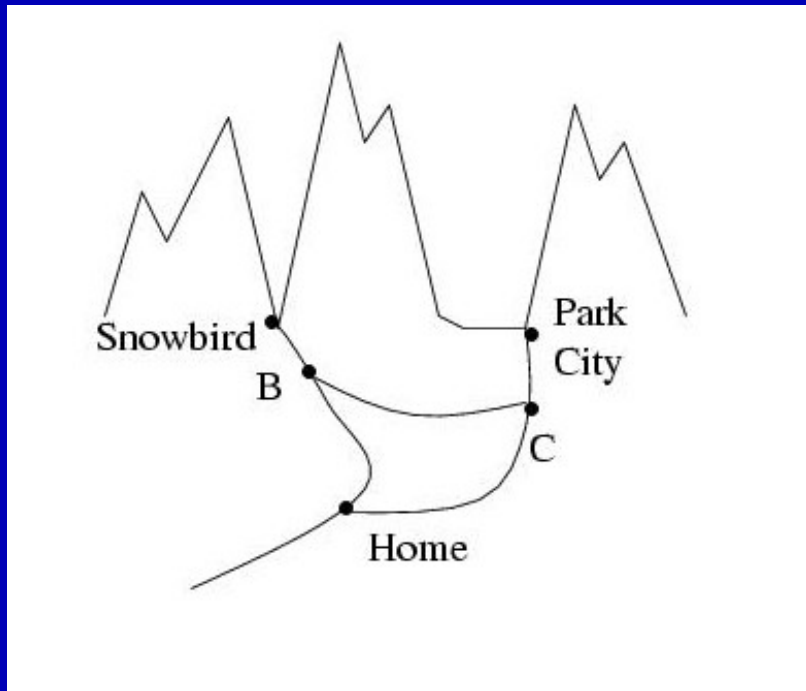
- Observational actions detect state



# Plan Generation with Contexts

- *Context* = possible outcome of conditional steps in the plan
- Generate a plan with branches for every possible outcome of conditional steps
  - Do this by creating a new goal state for the negation of the current contexts

# Conditional Planning Example



...

$\sim \text{Open}(B, S)$

$\text{At}(X), \text{Is-Resort}(X)$

© Martha E. Pollack

Init

$\text{At}(\text{Home}), \text{Resort}(P), \text{Resort}(S)$

$\sim \text{Open}(B, S)$

$\text{Open}(B, S)$

Go(Home, B)

$\text{At}(B)$

Observe(B)

$\text{At}(B), \text{Open}(B, S)$

Go(B, S)

$\text{Open}(B, S)$

~~$\text{At}(X), \text{Is-Resort}(X)$~~

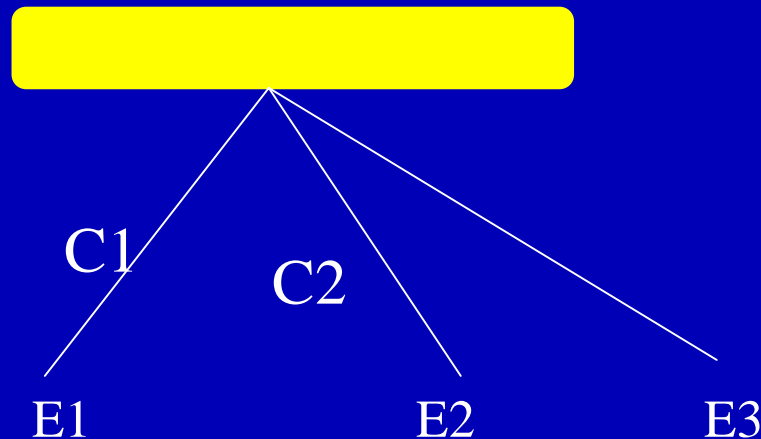
$\text{Open}(B, S)$

# Corrective Repair

- “Correct” the problems encountered, by specifying what to do in alternative contexts
- Requires observational actions, but not probabilities
- Plan for  $C1$ ;  $\sim C1 \wedge C2$ ;  $\sim C1 \wedge \sim C2 \wedge C3$ ; . . .
- Disjunction of contexts is a tautology—cover all cases!
  - In practice, may be impossible

# Preventive Repair

- Prevent problems from occurring
- “Confrontation” as a threat resolution strategy



# When Plans May Fail...

*conditional  
plans*

*MDPs*

*universal  
plans*

*probabilistic  
plans*

*conformant  
plans*

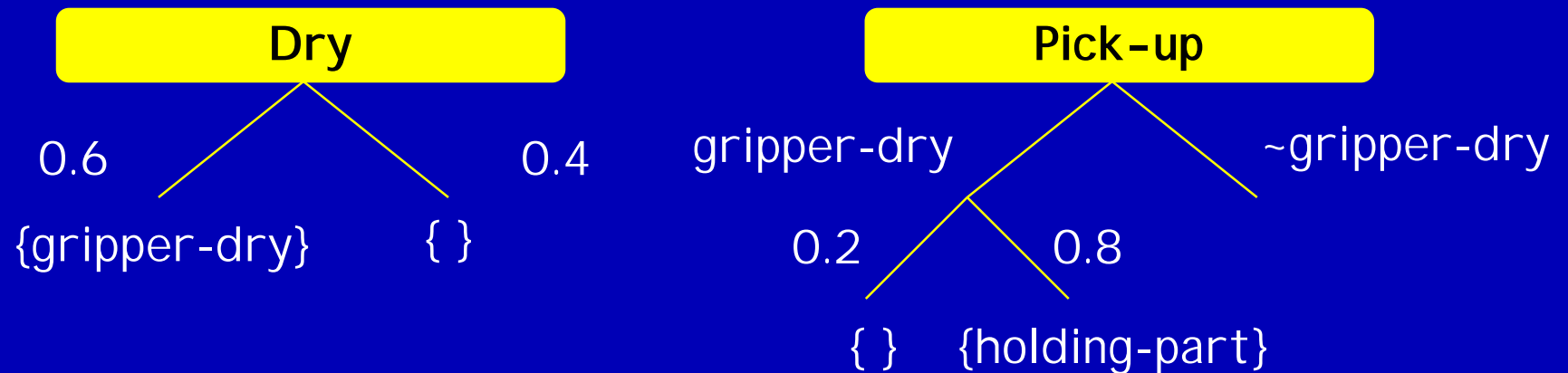


“Closed Loop”  
Planning

“Open Loop”  
Planning

# Probabilistic Planning

- Again, causal steps with alternative outcomes, but this time, know probability of each

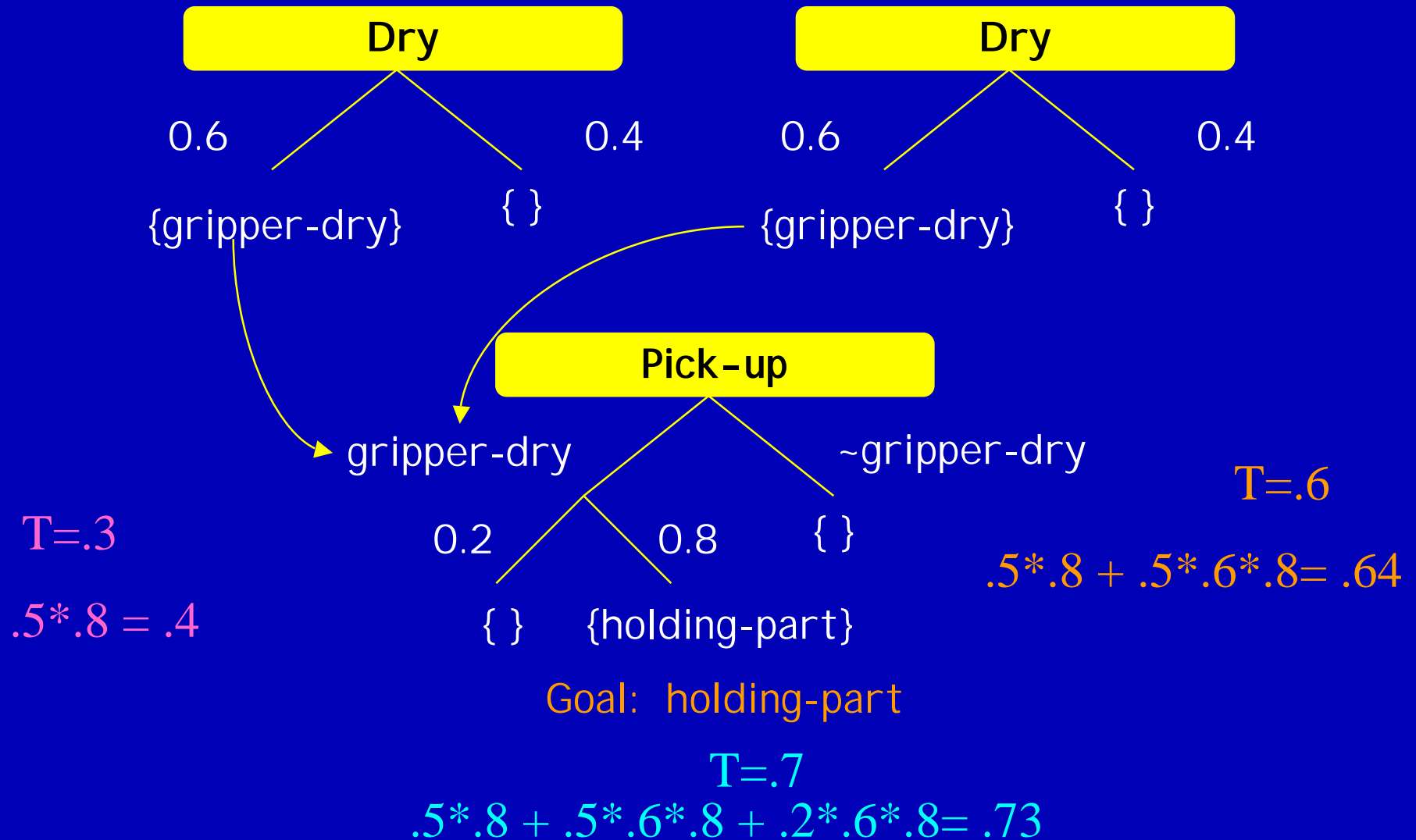


# Planning to a Guaranteed Threshold

- Generate a plan that achieves goal with probability exceeding some threshold
- Don't need observation actions

# Probabilistic Planning Example

$$P(\text{gripper-dry}) = .5$$



# Preventive Repair

- Probabilistic planning “prevents” problems from arising
- Success measured w.r.t. a threshold
- Don’t require observational actions (although in practice, may allow them)
- Exist SAT-based probabilistic planners
  - MAXPLAN

# Combining Correction and Prevention

**PLAN** (*init*, *goal*, *T*)

*plans* = {make-init-plan (*init*, *goal*)}

while *plan-time* < *T* and *plans* is not empty do

**CHOOSE** a plan *P* from *plans*

**SELECT** a flaw *f* from *P*, add all refinements of *P* to *plans*:

*plans* = *plans* ∪ new-step(*P*,*f*) ∪ step-reuse (*P*,*f*)

if *f* is an open condition

*plans* = *plans* ∪ demote(*P*,*f*) ∪ promote(*P*,*f*) ∪ confront (*P*,*f*)

∪ constrain-to-branch(*P*,*f*) if *f* is a threat

*plans* = *plans* ∪ corrective-repair(*P*,*f*) ∪ preventive-repair(*P*,*f*)

if *f* is a dangling edge

return (*plans*)

# When Plans May Fail...

*conditional  
plans*

*MDPs*

*universal  
plans*

*cond-prob plans with  
contingency selection*

*probabilistic  
plans*

*conformant  
plans*



“Closed Loop”  
Planning

“Open Loop”  
Planning

# POP QUIZ !!!

- Don't flip the page in your lectures.

# A Very Quick Decision Theory Review

	Lecture is Good	Lecture is Bad
Go to Beach		
Go to Lecture		

# A Very Quick Decision Theory Review

	Lecture is Good	Lecture is Bad
Go to Beach	+suntan ( $V=10$ ) -knowledge ( $V = -40$ )	+suntan ( $V=10$ )
Go to Lecture	-suntan ( $V=-5$ ) +knowledge ( $V=50$ )	-suntan ( $V=-5$ ) bored ( $V=-10$ )

# A Very Quick Decision Theory Review

	Lecture is Good $p$	Lecture is Bad $1-p$
Go to Beach	+suntan ( $V=10$ ) -knowledge ( $V = -40$ )	+suntan ( $V=10$ )
Go to Lecture	-suntan ( $V=-5$ ) +knowledge ( $V=50$ )	-suntan ( $V=-5$ ) bored ( $V=-10$ )

# A Very Quick Decision Theory Review

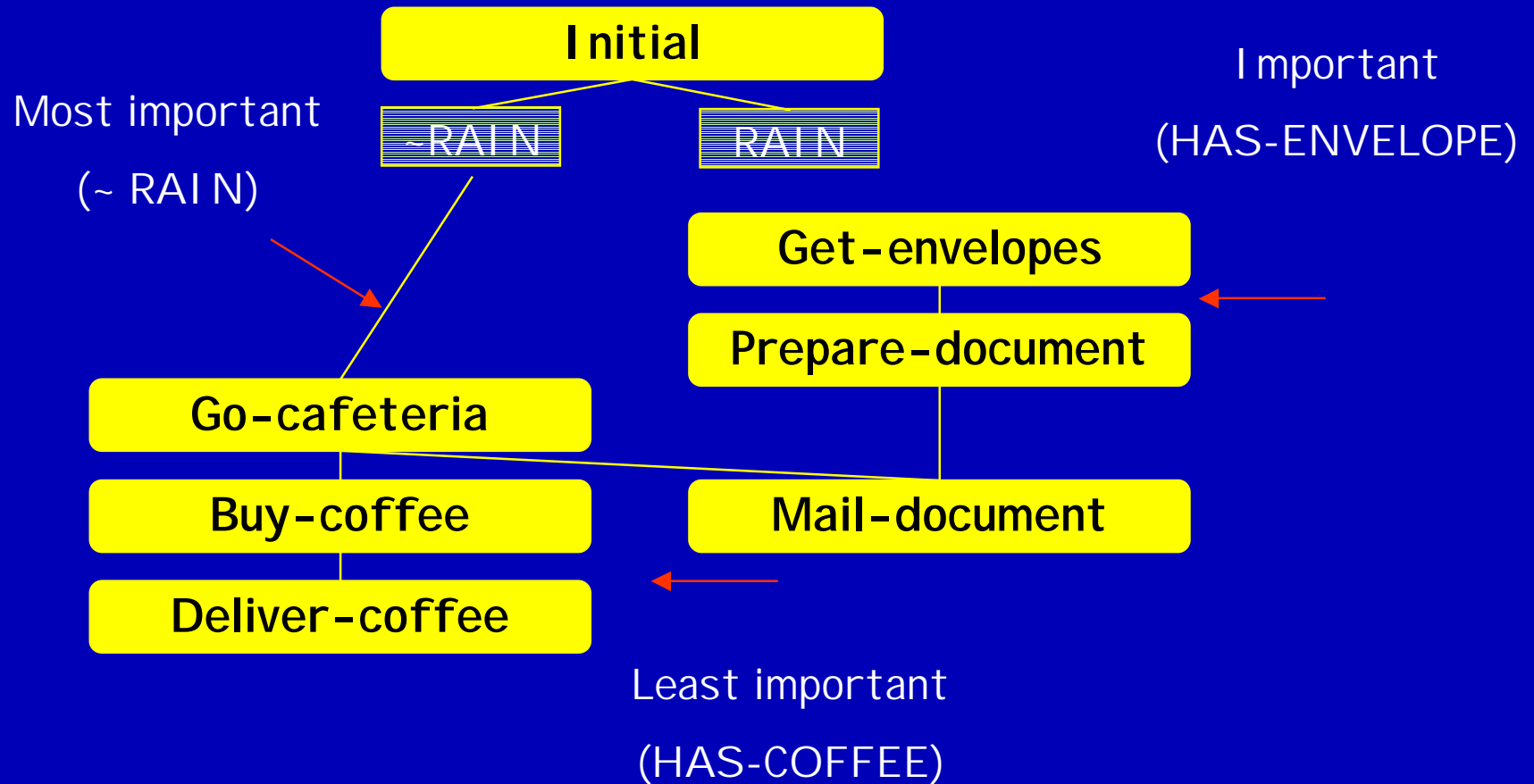
	Lecture is Good $p$	Lecture is Bad $1-p$
Go to Beach	+suntan ( $V=10$ ) -knowledge ( $V = -40$ )	+suntan ( $V=10$ )
Go to Lecture	-suntan ( $V=-5$ ) +knowledge ( $V=50$ )	-suntan ( $V=-5$ ) bored ( $V=-10$ )

$$EU(\text{Beach}) = p*(-30) + (1-p)*10 = 10-40p$$

$$EU(\text{Lecture}) = p*(45) + (1-p)*(-15) = 60p-15$$

$$EU(\text{Lecture}) \geq EU(\text{Beach}) \text{ iff } 60p-15 \geq 10-40p, \text{ i.e. } p \geq 1/4$$

# Contingency Selection Example



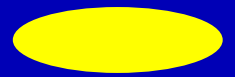
Goals: has-coffee (value=x)

document-mailed (value=y)  $y \gg x$

# Influences on Contingency Selection

Factor	Directly Available?
Expected increase in utility	YES
Expected cost of executing contingency plan	NO
Expected cost of generating contingency plan	NO
Resources available at execution time	NO

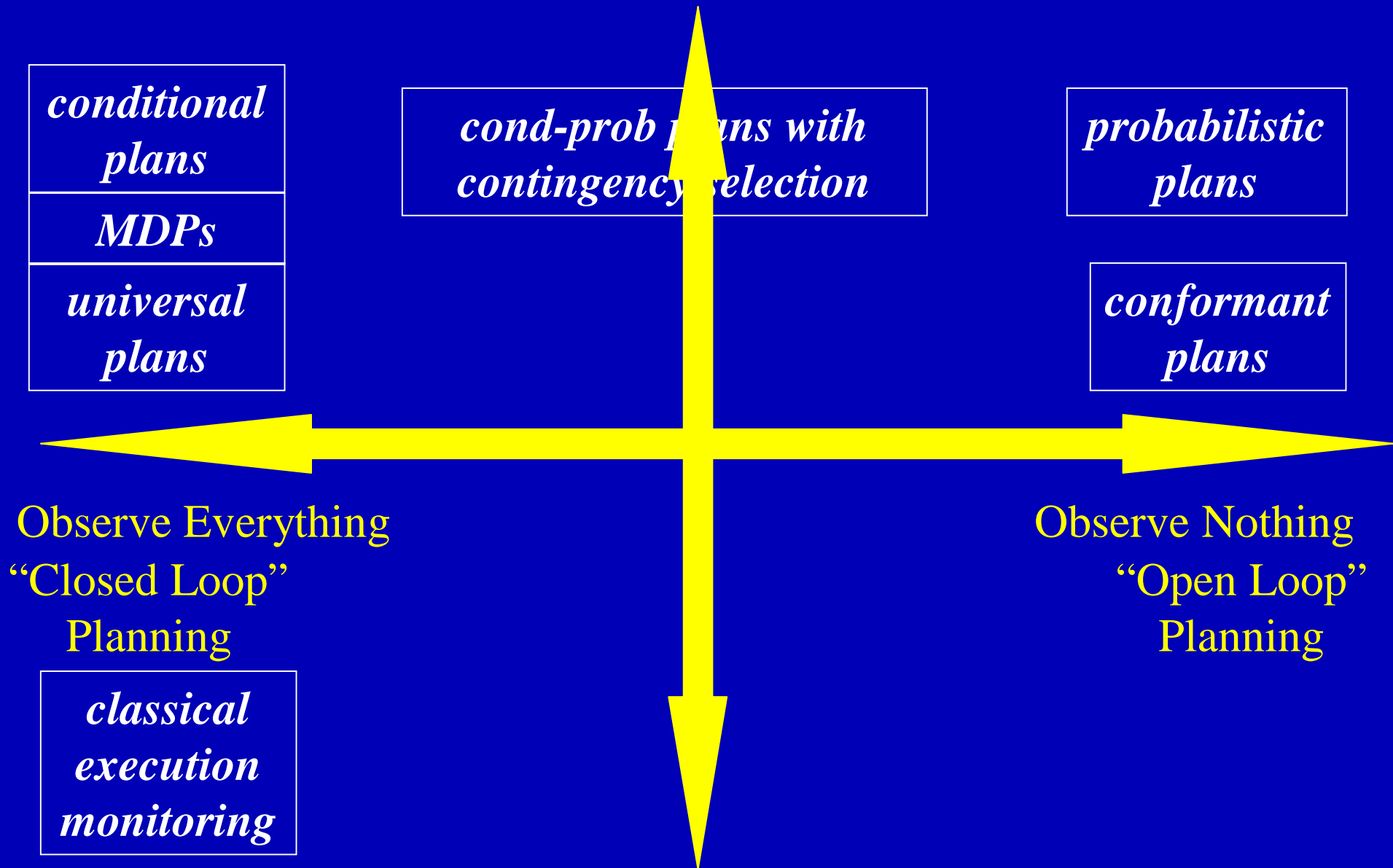
# Expected Increase in Plan's Utility



$$\sum_{g \in \text{Goals}} \{ \text{value}(g) * \text{prob}(s_i \text{ executed and } c \text{ is not true and } g \text{ is not true}) \}$$

1. Construct a plan, possibly with dangling edges.
2. For each dangling edge  $e = \langle s_i, c \rangle$ , compute expected increase in plan utility for repairing/preventing  $e$ .
3. Repair or prevent  $e$  with maximal expected utility increase.
4. If expected utility does not exceed threshold, loop.

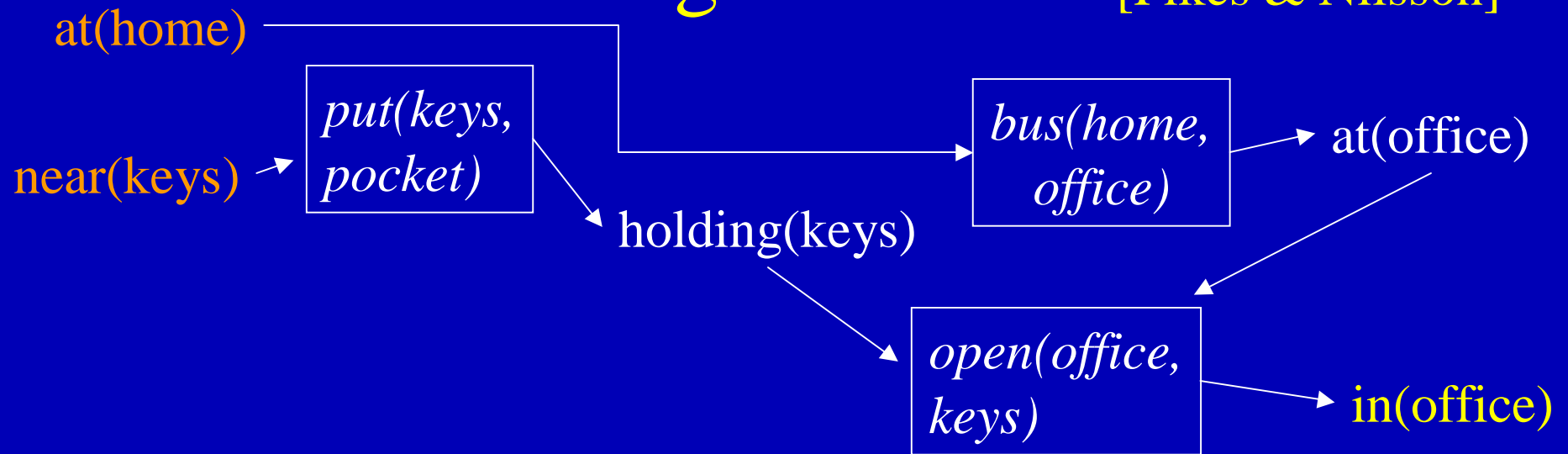
## Build Observations and Reactions into Plan



## Handle Observations and Reactions Separately

# Triangle Tables

[Fikes & Nilsson]



	<i>init</i>	<i>put(keys, pocket)</i>	<i>bus(home, office)</i>	<i>open(offce, keys)</i>
1	<i>near(keys)</i>			
2	<i>at(home)</i>			
3		<i>holding(keys)</i>	<i>at(offce)</i>	
4				<i>in(offce)</i>
	1	2	3	4

Find largest  $n$  s.t.  $n^{th}$  kernal enabled  $\rightarrow$   
Execute  $n^{th}$  action.

# Triangle Tables

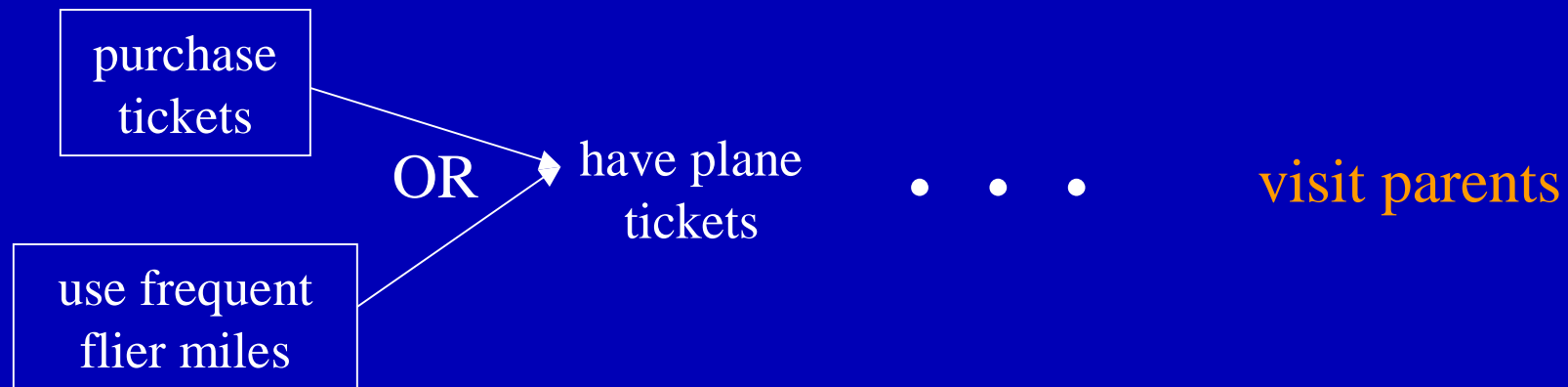
- Advantages:
  - Allow limited opportunistic reasoning
- Disadvantages:
  - Assumes a totally ordered plan
  - Expensive to check *all* preconditions before every action
  - Otherwise is silent on what preconditions to check when
  - Checks only for preconditions of actions in the plan

# Monitoring for Alternatives

[Veloso, Pollack, & Cox]

- May want to change the plan even if it can still succeed
- Monitor for conditions that caused rejection of alternatives during planning
- May be useful during planning as well as during execution

# Alternative Monitoring Example



*Preference Rule: Use frequent flier miles when cost > \$500.*

T1: Cost = \$450; Decide to purchase tickets.

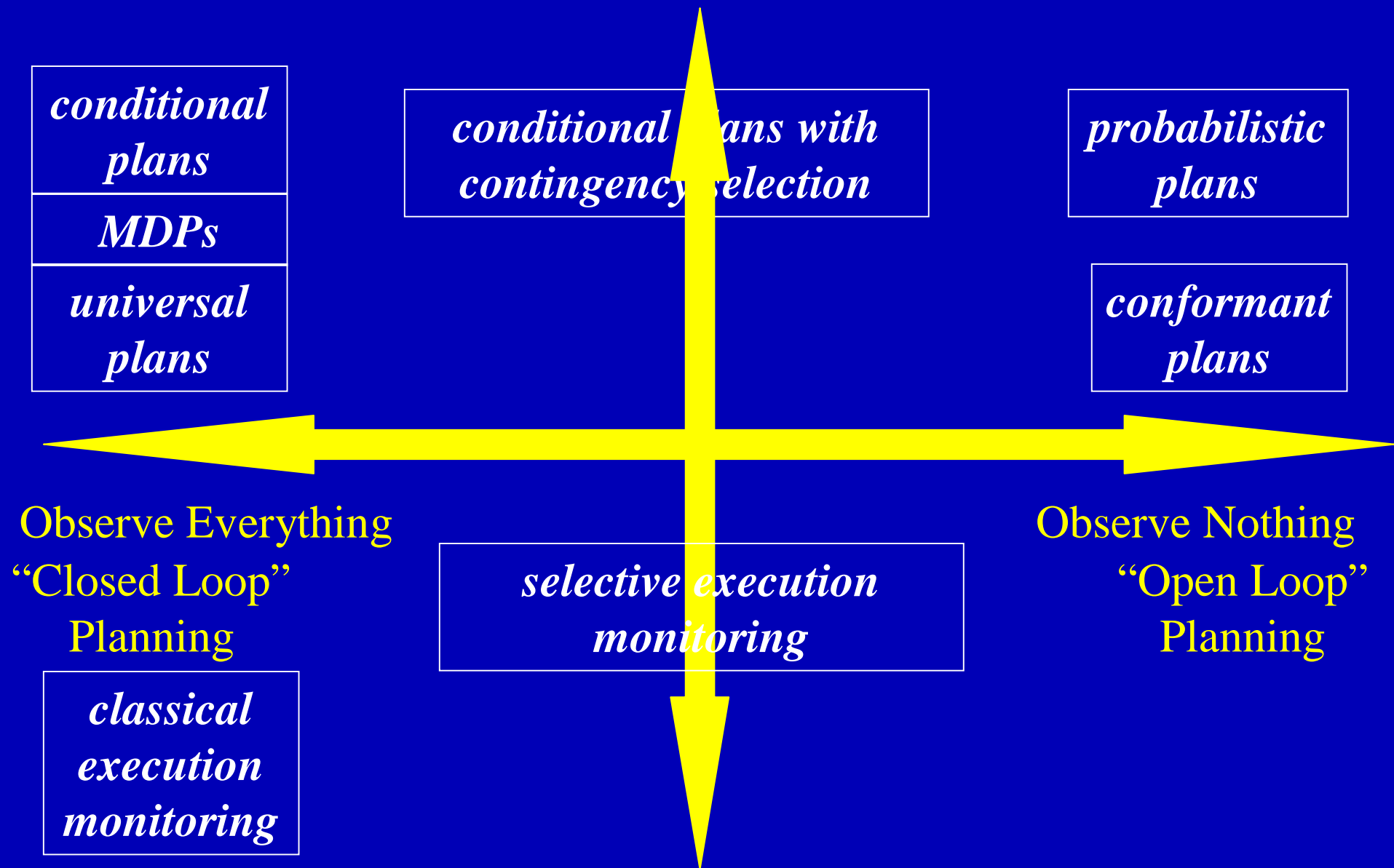
T2: Cost = \$600; Decide to use frequent flier miles???

Depends on whether execution has begun, and if so, on the cost of plan revision.

# Monitoring for Alternatives

- Classes of monitors:
  - Preconditions
  - Usability Conditions
    - take the bus (vs. bike) because of rain
  - Quantified Conditions
    - number of cars you need to move to use van goes to 0
  - Preference Conditions
- Problems
  - Oscillating conditions
  - Ignores cost of plan modification, especially after partial execution
  - Still doesn't address timing and cost of monitoring

## Build Observations and Reactions into Plan



## Handle Observations and Reactions Separately

# Decision-Theoretic Selection of Monitors

[Boutilier]

- Monitor selection is actually a sequential decision problem
- At each stage:
  - Decide what (if anything) to monitor
  - Update beliefs on the basis of monitoring results
  - Decide whether to continue or abandon the plan
  - If continue, update beliefs after acting
- Formulate as a POMDP

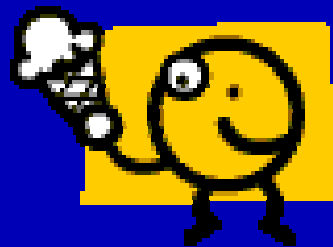
# Required Information

- Probability that any precondition may fail (or may become true) as the result of an exogenous action
- Probability that any action may fail to achieve its intended results
- Cost of attempting to execute a plan action when its preconditions have failed
- Value of the best alternative plan at any point during plan execution
- Model of the monitoring processes and their accuracy

# Heuristic Monitoring

- Solving the POMDP is computationally quite costly
- Effective alternative: Construct and solve a separate POMDP for each stage of the plan; combine results online

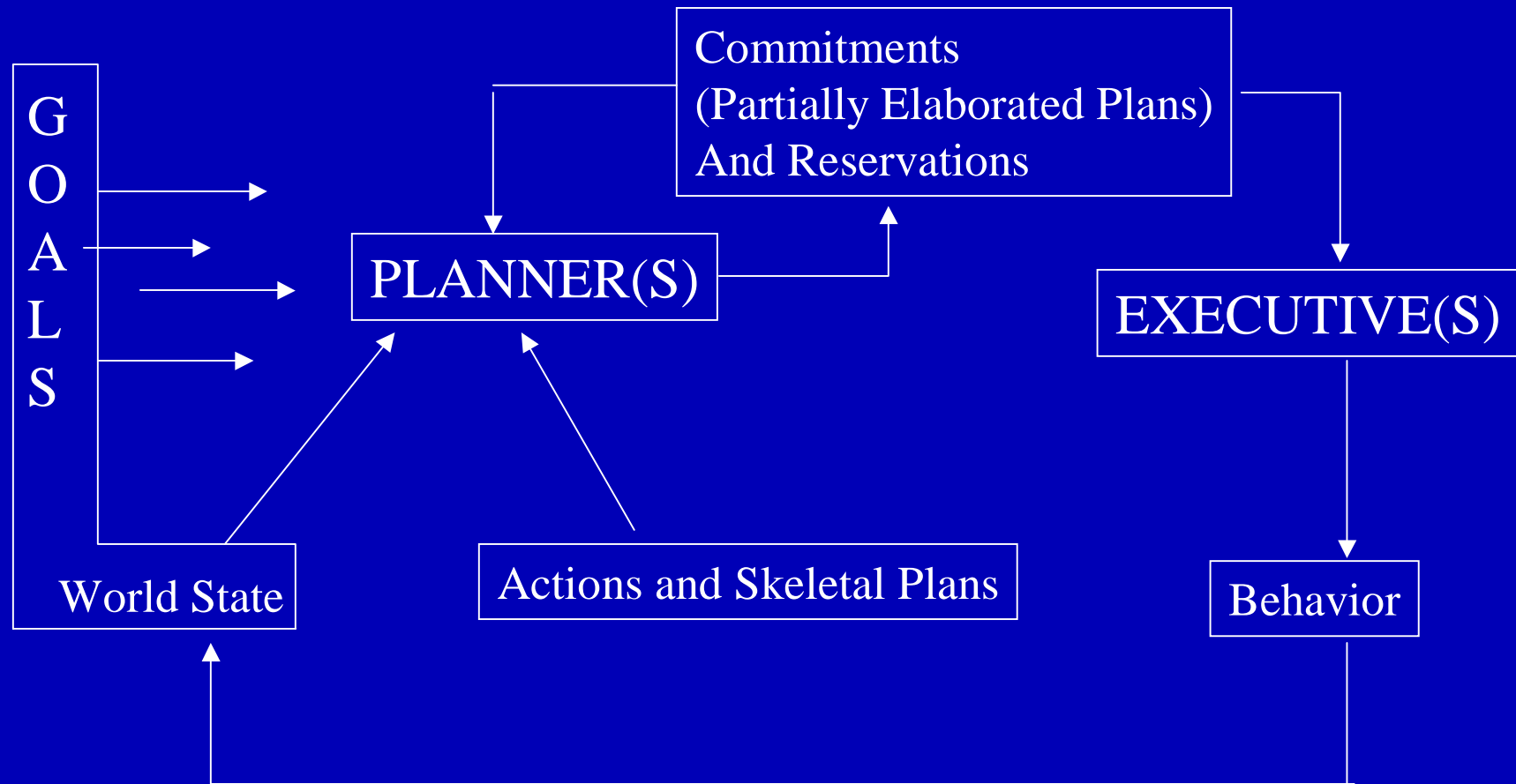
# Conclusion



# Today's Outline

- ✓ 1. Handling Potential Plan Failures
- 2. Managing Deliberation Resources

# Integrated Model of Planning and Execution



# Deliberation Management

- Have planning problems for goals  $G_1, G_2, \dots, G_n$ , and possibly competing execution step  $X$ .
- What should the agent do?
- A decision problem: can we apply decision theory?

# DT Applied to Deliberation

		PROBLEM 1. Hard to specify the conditions until the planning is complete.		
Plan for G1 now				
Plan for G2 now				
Plan for G3 now		PROBLEM 2. The DT problem takes time, during which the environment may change.		
Perform action X now				

(Not unique to DT for deliberation: Type II Rationality)

# Bounded Optimality

[Russell & Subramanian]

- Start with a method for *evaluating* agent behavior
- Basic idea:
  - Recognize that all agents have computational limits as a result of being implemented on physical architecture
  - Treat an agent as (boundedly) optimal if it performs at least as well as other agents with identical architectures

# Agent Formalism

Percepts:  $O$

Percept History:  $O^T$

Actions:  $A$

Action History:  $A^T$

Agent Function:  $f: O^t \rightarrow A$  s.t.  $A^T(t) = f(O^T)$

World States:  $X$

State History:  $X^T$

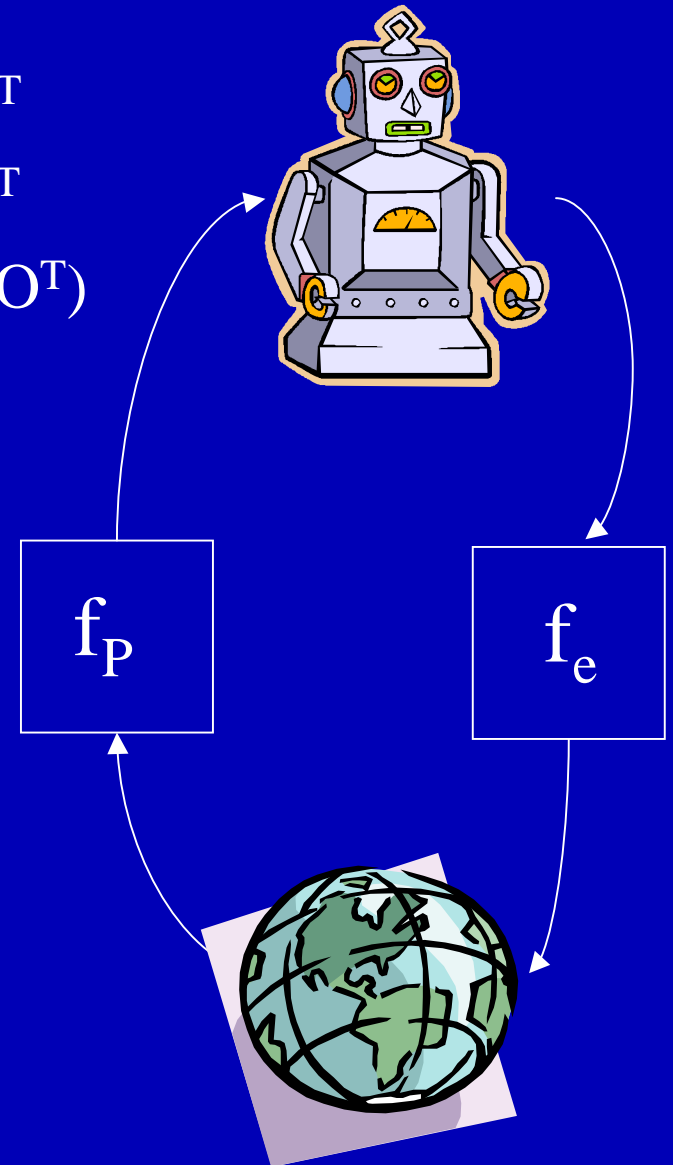
Perceptual Filtering Function:  $f_p(x)$

Action Transition Function:  $f_e(a, x)$

$X^T(0) = X_0$

$X^T(t+1) = f_e(A^T(t), X^T(t))$

$O^T(t) = f_p(X^T(t))$



# Agent Implementations

- A given architecture  $M$  can run a set of programs  $L_M$
- Every program  $l \in L_M$  implements some agent function  $f$
- But not every agent function  $f$  can be implemented on a given architecture  $M$
- So define:

$$Feasible(M) = \{f \mid \exists l \in L_M \text{ that implements } f\}$$

# Rational Programs

- Given a set of possible environments  $\mathbf{E}$ , we can compute the expected value,  $V$ , of an agent function  $f$ , or a program  $l$
- Perfectly rational agent for  $\mathbf{E}$  has *agent function*  $f_{\text{OPT}}$  such that  $f_{\text{OPT}} = \text{argmax}_f (V(f, \mathbf{E}))$
- Boundedly optimal agent for  $\mathbf{E}$  has an *agent program*  $l_{\text{OPT}} = \text{argmax}_{l \in L_M} V(l, \mathbf{M}, \mathbf{E})$
- So bounded optimality is the best you can hope for, given some fixed architecture!

# Back to Deliberation Management

“The gap between theory and practice is bigger in practice than in theory.”

Bounded Optimality not (yet?) applied to the problem of deciding amongst planning problems.

Has been applied to certain cases of deciding amongst decision procedures (planners).

# Bounded Optimality Result I

- Given an episodic real-time environment with fixed deadlines

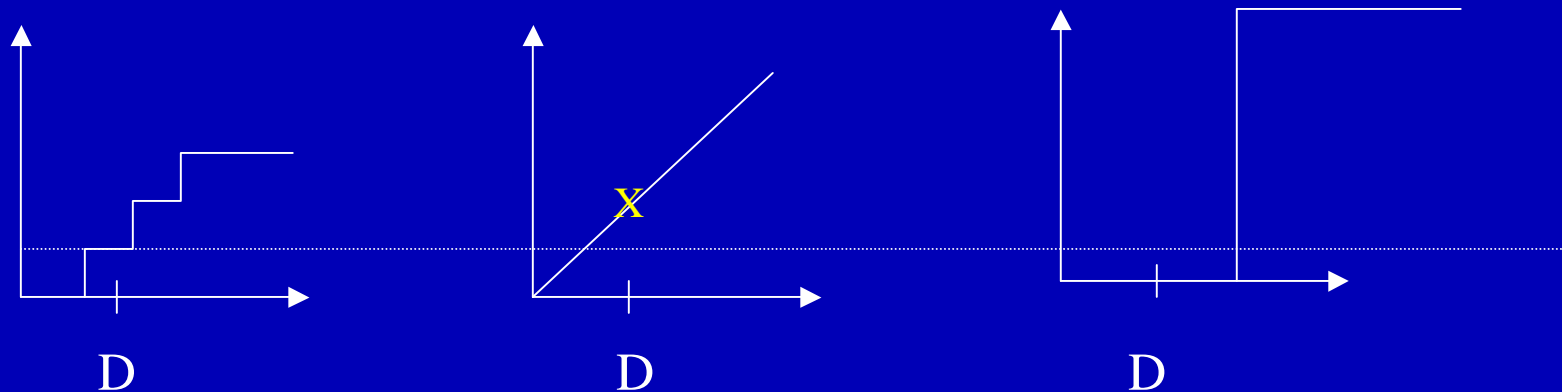
the best program is the single decision procedure of maximum quality whose runtime is less than the deadline.

An action taken any time up to the deadline gets the same value; no value after that

State history is divided into a series of episodes, each terminated by an action.

# Bounded Optimality Result I

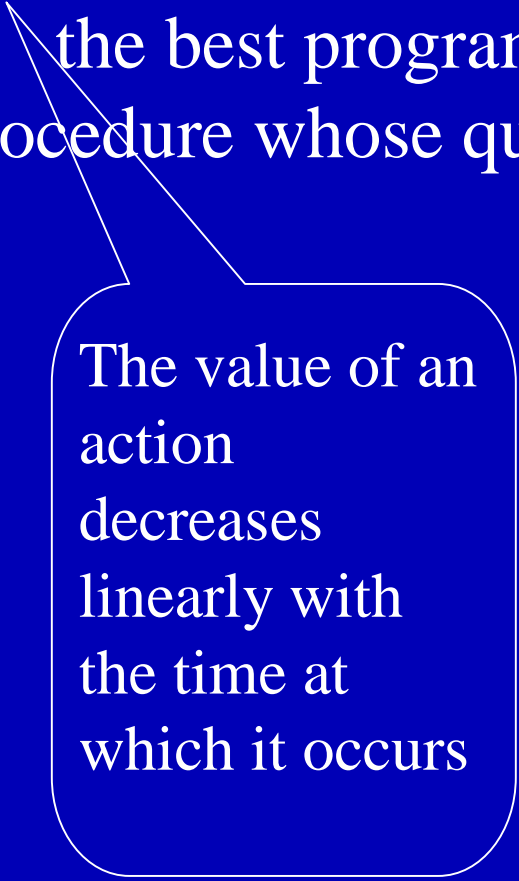
- Given an episodic real-time environment with fixed deadlines  
the best program is the single decision procedure of maximum quality whose runtime is less than the deadline.



# Bounded Optimality Result II

- Given an episodic real-time environment with fixed time costs

the best program is the single decision procedure whose quality net of time cost is highest.



The value of an action decreases linearly with the time at which it occurs

# Bounded Optimality Result III

- Given an episodic real-time environment with stochastic deadlines

can use Dynamic Programming to compute an optimal sequence of decision procedures, whose rules are in nondecreasing order of quality.

Like fixed deadlines, but the time of the deadline is given by a probability distribution

# Challenge

- Develop an account of bounded optimality for the deliberation management problem!

# An Alternative Account

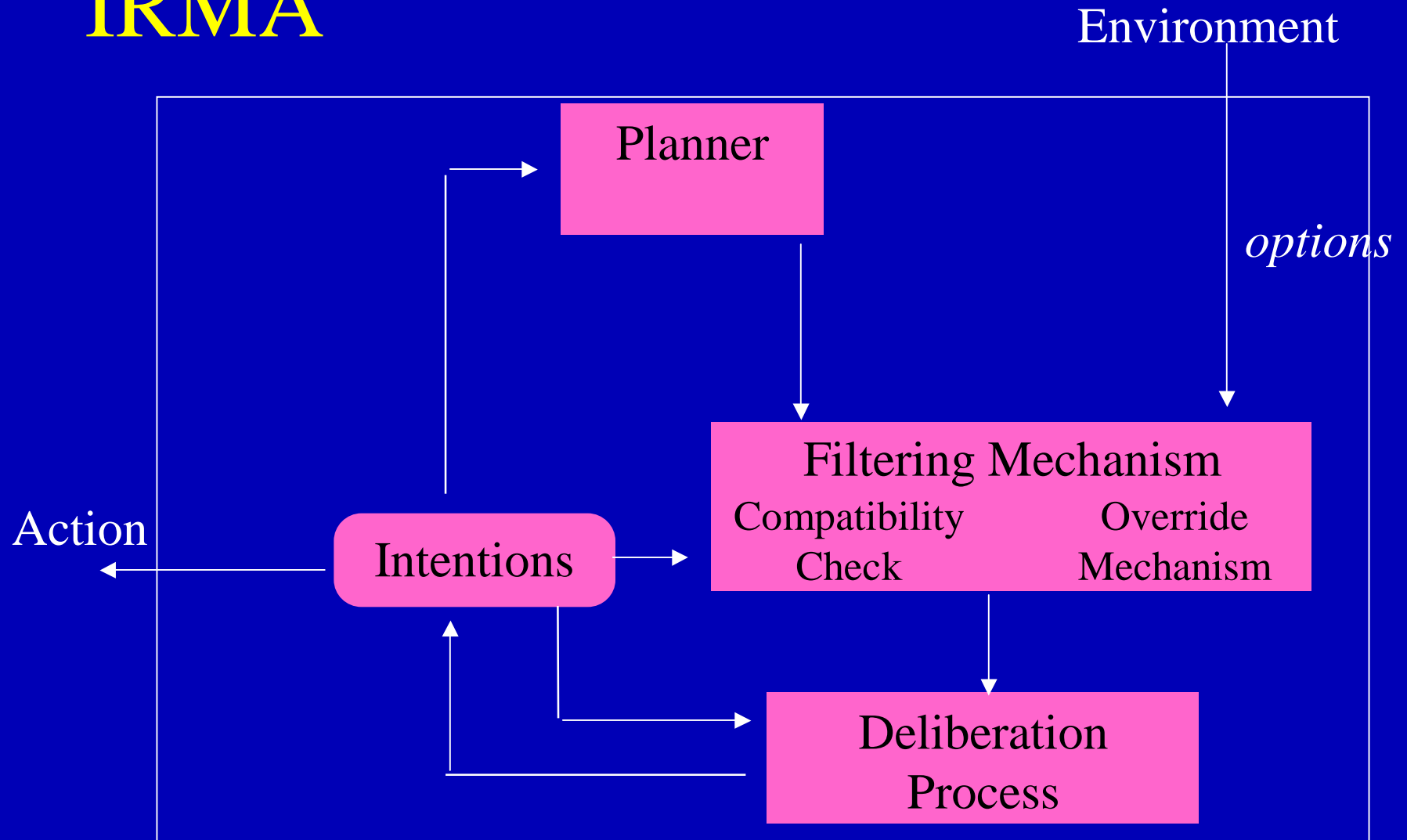
[Bratman, Pollack, & Israel]

- Heuristic approach, based on BDI (Belief-Desire-Intention) theory
- Grew out of philosophy of intention
- Was influential in the development of PRS (Procedural Reasoning System)

# The Philosophical Motivation

- Question: Why Plan (Make Commitments)?
  - Metaphysically Objectionable (action at a distance) *or*
  - Rationally Objectionable (if commitments are irrevocable) *or*
  - A Waste of Time (if you maintain commitments only when you're form the commitment anyway)
- One Answer: Plans help with deliberation management, by constraining future actions

# IRMA



# Filtering

- Mechanism for maintaining stability of intentions in order to focus reasoning
- Designer must balance appropriate sensitivity to environmental change against reasonable stability of plans
- Can't expect perfection: Need to trade occasional wasted reasoning and locally suboptimal behavior for overall effectiveness

# The Effect of Filtering

	Survives compatibility check	Triggers override	Deliberation leads to change of plan	Deliberation <i>would have</i> led to change of plan
1	N	Y	Y	
2	N	Y	N	
3	N	N		N
4	N	N		Y
5	Y			

Situations 1 & 2: Agent behaves *cautiously*

Situations 3 & 4: Agent behaves *boldly*

Situation 2: Wasted computational effort

Situation 4: Locally suboptimal behavior

# The Effect of Filtering

	Survives compatibility filter	Triggers filter override	Deliberation leads to change of plan	Deliberation <i>would have</i> led to change of plan	Deliberation worthwhile
1a		N	Y	Y	Y
1b	N	Y	Y		N
2	N	Y	N		
3	N	N		N	
4a		N	N		Y Y
4b	N	N		Y	N
5	Y				

Situations 1 & 2: Agent behaves *cautiously* (In 1a, caution pays!)

Situations 3 & 4: Agent behaves *boldly* (In 3 & 4b, boldness pays!)

Situation 1b & 2: Wasted computational effort

Situation 4a: Locally suboptimal behavior

# From Theory to Practice

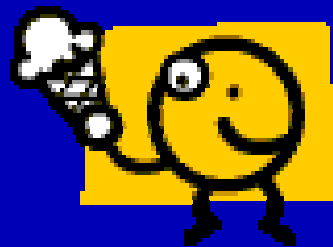
“The gap between theory and practice is bigger in practice than in theory.”

- Most results were shown in an artificial, simulated environment: The Tileworld
- More recent work:
  - Refined account in which filtering is not all-or-nothing: the greater the potential value of a new option, the more change to the background plan allowed.
  - Based on account of computing the cost of actions *in the context of other plans*.

# Planning and Execution—Other Issues

- Goal identification
- Cost/benefit assessment of plans
- Replanning techniques and priorities
- Execution Systems: PRS
- Real-Time Planning Systems: MARUTI, CIRCA

# Conclusion



# References

1. Temporal Constraint Networks

Dechter, R., I. Meiri, and J. Pearl, “Temporal Constraint Networks,” *Artificial Intelligence* 49:61-95, 1991.

2. Temporal Plan Dispatch

Muscettola, N., P. Morris, and I. Tsamardinos, “Reformulating Temporal Plans for Efficient Execution,” in *Proc. of the 6<sup>th</sup> Conf. on Principles of Knowledge Representation and Reasoning*, 1998.

Tsamardinos, I., P. Morris, and N. Muscettola, “Fast Transformation of Temporal Plans for Efficient Execution,” in *Proc. of the 15<sup>th</sup> Nat’l. Conf. on Artificial Intelligence*, pp. 254-161, 1998.

Wallace, R. J. and E. C. Freuder, “Dispatchable Execution of Schedules Involving Consumable Resources,” in *Proc. of the 5<sup>th</sup> Int’l. Conf. On AI Planning and Scheduling*, pp. 283-290, 2000.

I. Tsamardinos, M. E. Pollack, and P. Ganchev, “Flexible Dispatch of Disjunctive Plans,” in *Proc. of the 6<sup>th</sup> European Conf. on Planning*, 2001.

# References (2)

## 3. Disjunctive Temporal Problems

Oddi, A. and A. Cesta, “Incremental Forward Checking for the Disjunctive Temporal Problem,” in *Proc. of the European Conf. On Artificial Intelligence*, 2000.

Stergiou, K. and M. Koubarakis, “Backtracking Algorithms for Disjunctions of Temporal Constraints,” *Artificial Intelligence* 120:81-117, 2000.

Armando, A., C. Castellini, and E. Guinchiglia, “SAT-Based Procedures for Temporal Reasoning,” in *Proc. Of the 5<sup>th</sup> European Conf. On Planning*, 1999.

Tsamardinos, I. *Constraint-Based Temporal Reasoning Algorithms with Applications to Planning*, Univ. of Pittsburgh Ph.D. Dissertation, 2001.

## 4. CSTP

Tsamardinos, I., T. Vidal, and M. E. Pollack, “CTP: A New Constraint-Based Formalism for Conditional, Temporal Planning,” to appear in *Constraints*, 2002.

# References (3)

## 5. STP-u

Khatib, L., P. Morris, R. Morris, and F. Rossi, “Temporal Reasoning with Preferences,” in *Proc. of the 17<sup>th</sup> Int’l. Joint Conf. on Artificial Intelligence*, pp. 322-327, 2001.

Morris, P., N. Muscettola, and T. Vida, “Dynamic Control of Plans with Temporal Uncertainty,” in *Proc. of the 17<sup>th</sup> Int’l. Joint Conf. on Artificial Intelligence*, pp. 494-499, 2001.

## 6. The Nursebot Project

M. E. Pollack, “Planning Technology for Intelligent Cognitive Orthotics,” in *Proc. of the 6<sup>th</sup> Intl. Conf. on AI Planning and Scheduling*, pp. 322-331, 2002.

M. E. Pollack, S. Engberg, J. T. Matthews, S. Thrun, L. Brown, D. Colbry, C. Orosz, B. Peintner, S. Ramakrishnan, J. Dunbar-Jacob, C. McCarthy, M. Montemerlo, J. Pineau, and N. Roy, “Pearl: A Mobile Robotic Assistant for the Elderly,” in *AAAI Workshop on Automation as Caregiver*, 2002

# References (4)

## 7. Conformant Planning

Smith, D. and D. Weld, “Conformant Graphplan,” in *Proc. Of the 15<sup>th</sup> Nat’l. Conf. on Artificial Intelligence*, pp. 889-896, 1998.

Kurien, J., P. Nayak, and D. Smith, “Fragment-Based Conformant Planning,” in *Proc. of the 6<sup>th</sup> Int’l. Conf. on AI Planning and Scheduling*, pp. 153-162, 2002.

Castellini, C., E. Giunchiglia, and A. Tacchella, “Improvements to SAT-Based Conformant Planning,” in *Proc. of the 6<sup>th</sup> European Conf. on Planning*, 2001.

## 8. Universal Plans

Schoppers, M., “Universal plans for reactive robots in unpredictable environments,” in *Proc. of the 10<sup>th</sup> Int’l. Joint Conf. on Artificial Intelligence*, 1987.

Ginsberg, M., “Universal planning: an (almost) universally bad idea,” *AI Magazine*, 10:40-44, 1989.

Schoppers, M., “In defense of reaction plans as caches,” *AI Magazine*, 10:51-60, 1989.

# References (5)

## 7. Conditional and Probabilistic Planning

Peot, M. and D. Smith, “Conditional Nonlinear Planning, in *Proc. of the 1<sup>st</sup> Int’l. Conf. On AI Planning Systems*, pp. 189-197, 1992.

Kushmerick, N., S. Hanks, and D. Weld, “An Algorithm for Probabilistic Least-Commitment Planning,” in *Proc. Of the 12<sup>th</sup> Nat’l. Conf. On AI*, pp. 1073-1078, 1994.

Draper, D., S. Hanks, and D. Weld, “Probabilistic Planning with Information Gathering and Contingent Execution,” in *Proc. of the 2<sup>nd</sup> In’l. Conf. on AI Planning Systems*, p. 31-26, 1994.

Pryor, L. and G. Collins, “Planning for Contingencies: A Decision-Based Approach,” *Journal of Artificial Intelligence Research*, 4:287-339, 1996.

Blythe, J., *Planning under Uncertainty in Dynamic Domains*, Ph.D. Thesis, Carnegie Mellon Univ., 1998.

Majercik, S. and M. Littman, “MAXPLAN: A New Approach to Probabilistic Planning,” in *Proc. of 4<sup>th</sup> Int’l. Conf. On AI Planning Systems*, pp. 86-93, 1998.

Onder, N. and M. E. Pollack, “Conditional, Probabilistic Planning: A Unifying Algorithm and Effective Search Control Mechanisms,” in *Proc. Of the 16<sup>th</sup> Nat’l. Conf. On Artificial Intelligence*, pp. 577-584, 1999.

# References (6)

8. Decision Theory

Jeffrey, R. *The Logic of Decision*, 2<sup>nd</sup> Ed., Chicago: Univ. of Chicago Press, 1983.

9. Execution Monitoring

Fikes, R., P. Hart, and N. Nilsson, “Learning and Executing Generalized Robot Plans,” *Artificial Intelligence*, 3:251-288, 1972.

Veloso, M., M. E. Pollack, and M. Cox, “Rationale-Based Monitoring for Continuous Planning in Dynamic Environments,” in *Proc. of the 4<sup>th</sup> Int’l. Conf. on AI Planning Systems*, pp. 171-179, 1998.

Fernandez, J. and R. Simmons, “Robust Execution Monitoring for Navigation Plans,” in *Int’l. Conf. on Intelligent Robotic Systems*, 1998.

Boutilier, C., “Approximately Optimal Monitoring of Plan Preconditions,” in *Proc. of the 16<sup>th</sup> Conf. on Uncertainty in AI*, 2000.

# References (7)

10. Bounded Optimality

Russell, S. and D. Subramanian, “Provably Bounded-Optimal Agents,” *Journal of Artificial Intelligence Research*, 2:575-609, 1995.

11. Commitment Strategies for Deliberation Management

Bratman, M., D. Israel, and M. E. Pollack, “Plans and Resource-Bounded Practical Reasoning,” *Computational Intelligence*, 4:349-255, 1988.

Pollack, M. E., “The Uses of Plans,” *Artificial Intelligence*, 57:43-69, 1992.

Horty, J. F. and M. E. Pollack, “Evaluating New Options in the Context of Existing Plans,” *Artificial Intelligence*, 127:199-220, 2001.