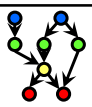


LECTURE 24 OF 42

Planning: Monitoring & Replanning, Continuous Planning

William H. Hsu
Department of Computing and Information Sciences, KSU

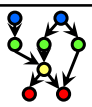
KSOL course page: http://snipurl.com/v9v3
Course web site: http://www.kddresearch.org/Courses/CIS730
Instructor home page: http://www.cis.ksu.edu/~bhsu

Reading for Next Class:

Chapter 13, p. 462– 486, Russell & Norvig 2nd edition
"The Gift of the Magi": http://en.wikipedia.org/wiki/The_Gift_of_the_Magi
Continuations: http://en.wikipedia.org/wiki/Continuation

LECTURE OUTLINE

- **Reading for Next Class: Chapter 13 (p. 462 – 486), R&N 2e**
- **Last Week: Partial-Order Planning, Blocks World, Graphplan, SATPlan**
  - **Classical planning defined: initial and goal conditions**
  - **POP: preconditions and effects, satisfying open preconditions**
  - **Algorithms: POP, Graphplan, SATPlan**
- **Last Class: Real-World Planning, 12.1 – 12.4 (p. 417 – 440), R&N 2e**
  - **Time (12.1), HTN Planning (12.2), HTN example: DWR (crane domain)**
  - **Nondeterminism and bounded indeterminacy (12.3)**
  - **Conditional planning, *aka* contingency planning (12.4)**
- **Today: Robust Planning Concluded, 12.5 – 12.8 (p. 441 – 454), R&N 2e**
  - **Monitoring and replanning (12.5)**
  - **Continuous planning (12.6)**
  - **Need for representation language for uncertainty**
- **Next Class: Intro to Probability as Language for Uncertainty**
- **Next Week: Reasoning under Uncertainty – KR, Graphical Models**

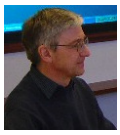


Acknowledgements

Héctor Muñoz-Avila
Associate Professor
Department of Computer Science & Engineering (CSE)
Lehigh University

Hai Hoang (not pictured)
Graduate Teaching Assistant, CSE
Lehigh University

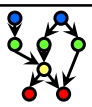
Gerhard Wickler
Research Fellow
Artificial Intelligence Applications Institute
Centre for Intelligent Systems and their Applications
University of Edinburgh



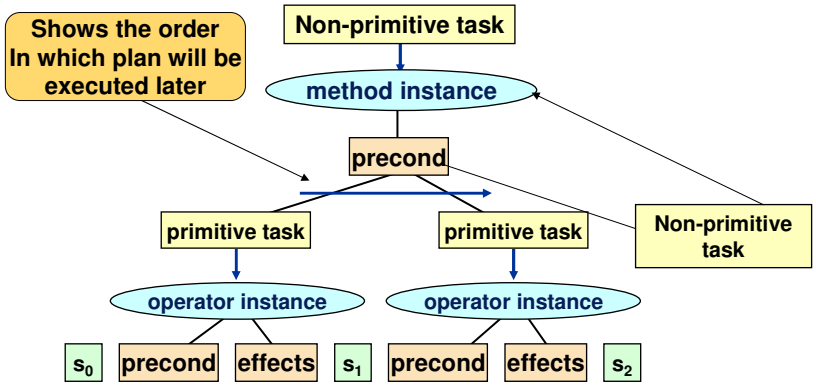
Austin Tate
Director,
Artificial Intelligence Applications Institute
University of Edinburgh

Frame, Qualification, and Ramification Problems – Review

- Frame Problem: Need to Describe and Propagate Non-Action
  - Representational – proliferation of frame axioms: *e.g.*, in Wumpus World
    - ⇨ Shoot doesn't clobber HoldingGold
    - ⇨ MoveNorth doesn't clobber HaveArrow (precondition for Shoot)
  - Inferential – copying state: HoldingGold (S_3) $\rightarrow_{Shoot}$ HoldingGold (S_4)
- Qualification Problem: Specifying All Preconditions ("Exceptions")
  - "Action A is possible *unless*..."
  - Improbable operator failures
- Ramification Problem: Specifying All Effects ("Side Effects")
  - "Action A *also* causes…"
  - Small incremental changes (*e.g.*, "wear and tear"), *aka* "butterflies in China"
- Solution Approaches
  - Representational FP: successor state axioms, graph/propositional planning
  - Inferential FP: defeasible reasoning (*e.g.*, defaults)
  - Qualification problem: abstraction; reaction; replanning
  - Ramification problem: defaults, abstraction



HTN Planning – Decomposition: Review



Adapted from slide © 2007 H. Hoang & H. Muñoz-Avila
Lehigh University
http://bit.ly/2nUghI

HTN Methods & Operators – Example: Review

Non prim

- *Method (decomposes into subtasks)*
  - *(:method (drive-truck ?truck ?loc-from ?loc-to)*
  - *((same ?loc-from ?loc-to))*
  - *((!do-nothing))*
  - *()*
  - *((!drive-truck ?truck ?loc-from ?loc-to)))*
  - *Notice the if else structure*
  - *Invoke non-primitive task: (drive-truck ?t ?x ?y)*
- *Operator (achieves PRIMITIVE TASKS)*
  - (:operator (!drive-truck ?truck ?locfrom ?locto)
  - ()
  - ((truck-at ?truck ?locfrom))
  - ((truck-at ?truck ?locto)))
  - Invoke primitive task: (!drive-truck ?t ?l1 ?l2)

Preconditions
Primitive task
Subtask list
Preconditions
Prim task
Delete list
Add list

Adapted from slide © 2007 H. Hoang & H. Muñoz-Avila
Lehigh University
http://bit.ly/2nUghI

HTN Planning Algorithm (TFD): Review

```
TFD(s, ⟨t_1, ..., t_k⟩, O, M)
    if k = 0 then return ⟨⟩ (i.e., the empty plan)
    if t_1 is primitive then
        active ← {(a, σ) | a is a ground instance of an operator in O,
                           σ is a substitution such that a is relevant for σ(t_1),
                           and a is applicable to s}
        if active = ∅ then return failure
        nondeterministically choose any (a, σ) ∈ active
        π ← TFD(γ(s, a), σ(⟨t_2, ..., t_k⟩), O, M)
        if π = failure then return failure
        else return a.π
    else if t_1 is nonprimitive then
        active ← {m | m is a ground instance of a method in M,
                      σ is a substitution such that m is relevant for σ(t_1),
                      and m is applicable to s}
        if active = ∅ then return failure
        nondeterministically choose any (m, σ) ∈ active
        w ← subtasks(m).σ(⟨t_2, ..., t_k⟩)
        return TFD(s, w, O, M)
```

Applying an operator Changing the state

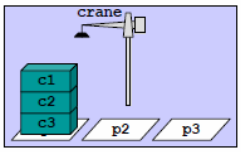
Randomly pick an applicable method

Decompose method into tasks

Adapted from slide © 2007 H. Hoang & H. Muñoz-Avila
Lehigh University
http://bit.ly/2nUghI

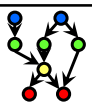
DWR – Crane Domain Example: Review

- task: move stack of containers from pallet p1 to pallet p3 in a way the preserves the order

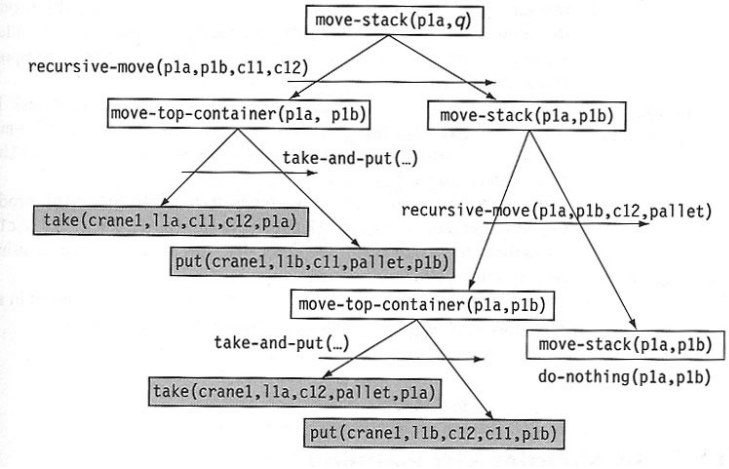


- (informal) methods:
  - move via intermediate: move stack to intermediate pile (reversing order) and then to final destination (reversing order again)
  - move stack: repeatedly move the topmost container until the stack is empty
  - move topmost: take followed by put action

Adapted from slides © 2005 G. Wickler & A. Tate
University of Edinburgh
http://bit.ly/Diir0

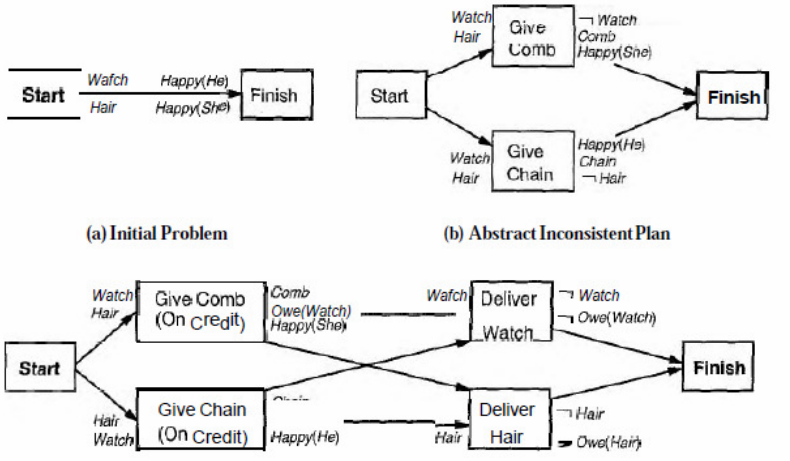


DWR – Example TFD: Review



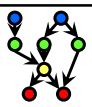
Adapted from slide © 2007 H. Hoang & H. Muñoz-Avila
Lehigh University
http://bit.ly/2nUghI

A Story: "Gift of the Magi" by O. Henry



Short story from Project Gutenberg: http://bit.ly/C89j6

Figure 12.8
p. 428 R&N 2e



Practical Planning [1]
Sensorless Planning: Review

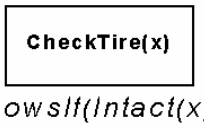
- **Problem: Bounded Indeterminacy**
  - Uncertainty in answering intelligent agent's questions (see: Lectures 0 & 1)
    - ⇨ "What world is like now"
    - ⇨ "What it will be like if I do action A"
  - Scenario for boundedly rational decision-making
- **Idea: Coerce State of World**
  - Complete plan in all possible situations
  - Example: move forward to walk through door OR push it open
- **Not Always Possible!**

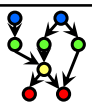
Practical Planning [2]
Conditional Planning: Review

$[\ldots, \mathbf{If}(p, [then\,plan], [else\,plan]), \ldots]$

Execution: check p against current KB, execute "then" or "else"

Conditional planning: just like POP except
if an open condition can be established by observation action
add the action to the plan
complete plan for each possible observation outcome
insert conditional step with these subplans





Sensorless & Contingency Planning in Vacuum World: Review

Single-state, start in #5. Solution??

$[Right, Suck]$

Conformant, start in $\{1,2,3,4,5,6,7,8\}$
e.g., $Right$ goes to $\{2,4,6,8\}$. Solution??

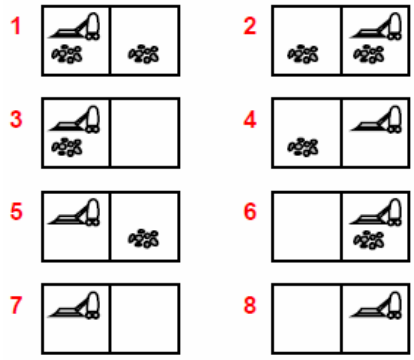
$[Right, Suck, Left, Suck]$

Sensorless

Contingency, start in #5
Murphy's Law: $Suck$ can dirty a clean carpet
Local sensing: dirt, location only.
Solution??

$[Right, \text{if } dirt \text{ then } Suck]$

Conditional

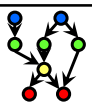


Practical Planning [3]: Monitoring and Replanning

- **Problem: Plans May Fail**
  - ✷ **Plan steps fail: *e.g.*, attempted play in sports**
  - ✷ **May be due to faulty**
    - ⇨ **sensors**
    - ⇨ **effectors**
    - ⇨ **Plans**
- **Need: Ability to Replan and Recover**

Monitoring/Replanning
Assume normal states, outcomes
Check progress *during execution*, replan if necessary
Unanticipated outcomes may lead to failure (e.g., no AAA card)

In general, some monitoring is unavoidable



Practical Planning [3]: Monitoring and Replanning

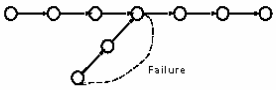
- **Failure: Preconditions of Remaining Plan Not Met**
- **Preconditions of Remaining Plan**
  - **All *preconditions* of remaining steps not achieved by remaining steps**
  - **All *causal links* crossing current time point**
- **Upon Failure**
  - **Resume POP**
  - **Achieve open conditions from current state**
- **Performs**
  - **Action monitoring**
  - **Execution monitoring (*aka* plan monitoring)**

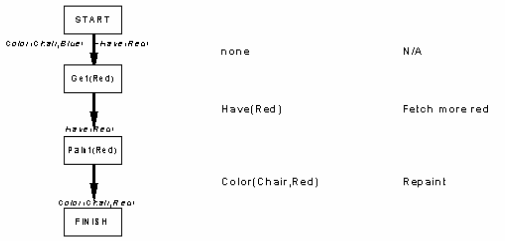
Adapted from slide © 2005 A. Cristea
http://www.dcs.warwick.ac.uk/~acristea/

Replanning and Continuations

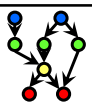
Simplest: on failure, replan from scratch

Better: plan to get back on track by reconnecting to best continuation
Generates "loop until done" behavior with no explicit loop





© 2004 S. Russell & P. Norvig. Reused with permission.



Execution *vs.* Action Monitoring

Execution monitoring

"failure" = preconditions of *remaining plan* not met

preconditions = causal links at current time

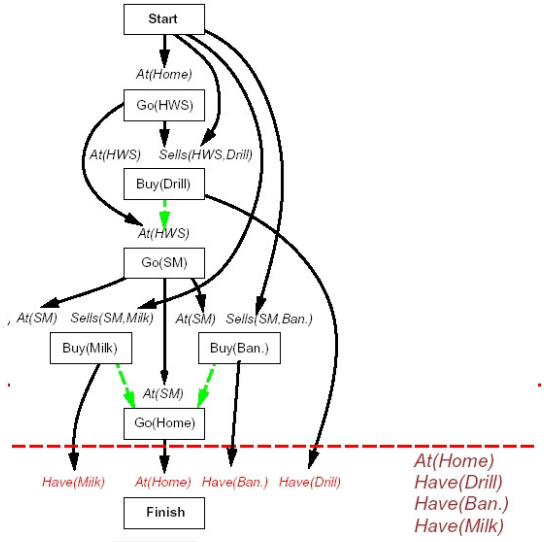
Action monitoring

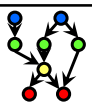
"failure" = preconditions of *next action* not met

(or action itself fails, e.g., robot bump sensor)

In both cases, need to *replan*

Practical Planning [4]: Continuous *aka* Lifelong Planning





TERMINOLOGY

- Bounded Indeterminacy: Kind of Uncertainty about Domain
  - "How world is like"
  - "How it will be if I do A"
- Robust Planning
  - Planning with plan step failures
  - Types
    - Sensorless: use coercion and reaction
    - Conditional *aka* contingency: IF statement
    - Monitoring and replanning: resume temporarily failed plans
    - Continuous: similar, but for agent that persists indefinitely
- Uncertain Reasoning
  - Ability to perform inference in presence of uncertainty about
    - premises
    - rules
  - Representations: probability, Dempster-Shafer theory, fuzzy logic

SUMMARY POINTS

- Previously: Partial-Order Planning, Blocks World, Graphplan, SATPlan
  - Classical planning defined: initial, goal conditions
  - POP: preconditions, effects, satisfying, threats and clobberings
  - Promotion, demotion
  - Planning *vs.* design, scheduling
  - Algorithms: POP, Graphplan, SATPlan
- Last Class: Real-World Planning
  - Time (12.1), HTN Planning (12.2), HTN example: DWR (crane domain)
  - Nondeterminism and bounded indeterminacy (12.3)
  - Conditional planning, *aka* contingency planning (12.4)
- Today: Robust Planning Concluded, 12.5 – 12.8 (p. 441 – 454), R&N 2e
  - Monitoring and replanning (12.5)
  - Continuous planning (12.6)
  - Need for representation language for uncertainty
- Next Class: Probability as KR for Uncertain Reasoning
- Coming Up: Bayesian Networks & Other Graphical Models