CS 4260 and CS 5260
Vanderbilt University

Planning Scenarios

This lecture assumes that you have

- Read Chapter 4, section 4.1 (6 pages)
- Read Chapter 5, section 5.1 (6 pages)
- Chapter 2, section 2.4 of ArtInt (7 pages)
- Read Chapter 6, through section 6.3 of ArtInt (15 pages)
- Watch feature-based planning under certainty Tuesday pre-class lecture at

As indicated on the Week 3 https://my.vanderbilt.edu/cs4260cs5260/schedule/

ArtInt: Poole and Mackworth, Artificial Intelligence 2E
at http://artint.info/2e/html/ArtInt2e.html

Unity Gaming Demo Website for Goal Oriented Action Planning

I discovered a website (Owens) that uses AI-planning in a small Unity game demo that involves four character classes (miner, logger, wood cutter, and blacksmith), and multiple actions for each. For example, the chopper has the MakeFirewood, ChopLog, GetAxe, CollectBranches, Move, and Rest actions. The website extensively describes the preconditions (for example, the ChopLog action requires an axe on hand, i.e. GetAxe) and effects, but the rest are discoverable. The actual code is also embedded in the website so that the viewer can see the actual algorithm of how the state-space graph is searched. Each of the four character classes also has its own goal, so in total all the parts of an AI-planner are there.

I think this demo website is a great introductory look at AI-planning, and really splits up the core concepts into easily digestible and visualizable pieces. One gripe I have is that I wish it were updated to use WebGL instead of the deprecated Unity Web player. **I also wish there was the possibility of choosing between different state-space searching algorithms and to be able to toggle between forward and regression planning for more advanced learners.**

Source: Owens, Brent. “Goal Oriented Action Planning for a Smarter AI.” Game Development Envato Tuts+, 23 Apr. 2014, https://gamedevelopment.tutsplus.com/tutorials/goal-oriented-action-planning-for-a-smarter-ai--cms-20793

Video Game NPCs
See the following URL -
http://www.primaryobjects.com/2015/11/06/artificial-intelligence-planning-with-strips-a-gentle-introduction/.

This website describes how a non-player character (NPC) can be controlled by planning AI. For example, one might want the NPC to interact with the character in certain ways as well as satisfy a list of goals, some simple and some complex. Since the world is fully virtualized and all of the actions the characters can do are completely specified by the video game developer, a representation of the goals and specific actions with their effects can be enumerated. This allows something like STRIPS to come up with a specific plan that satisfies all goals while being fairly responsive to the player and not constricting the flow of the game.

Interactive Narrative: An Intelligent Systems Approach
Mark O. Riedl and Vadim Bulitko
AI Magazine, Spring 2013
Association for the Advancement of Artificial Intelligence
https://www.aaai.org/ojs/index.php/aimagazine/article/view/2449

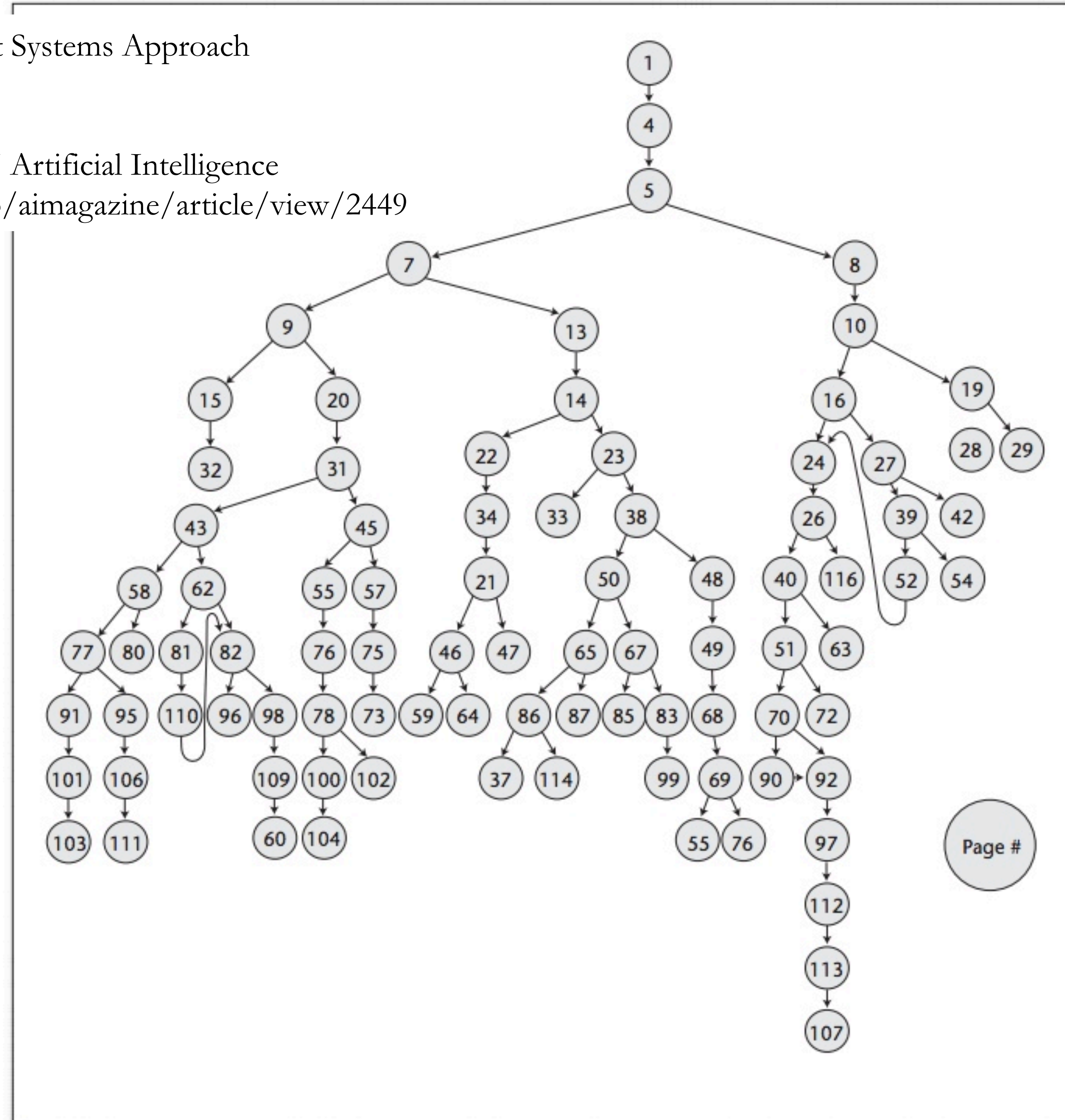


Figure 3. A Branching Story Graph from The Abominable Snowman.

Interactive Narrative: An Intelligent Systems Approach
Mark O. Riedl and Vadim Bulitko
AI Magazine, Spring 2013
Association for the Advancement of Artificial Intelligence
https://www.aaai.org/ojs/index.php/aimagazine/article/view/2449

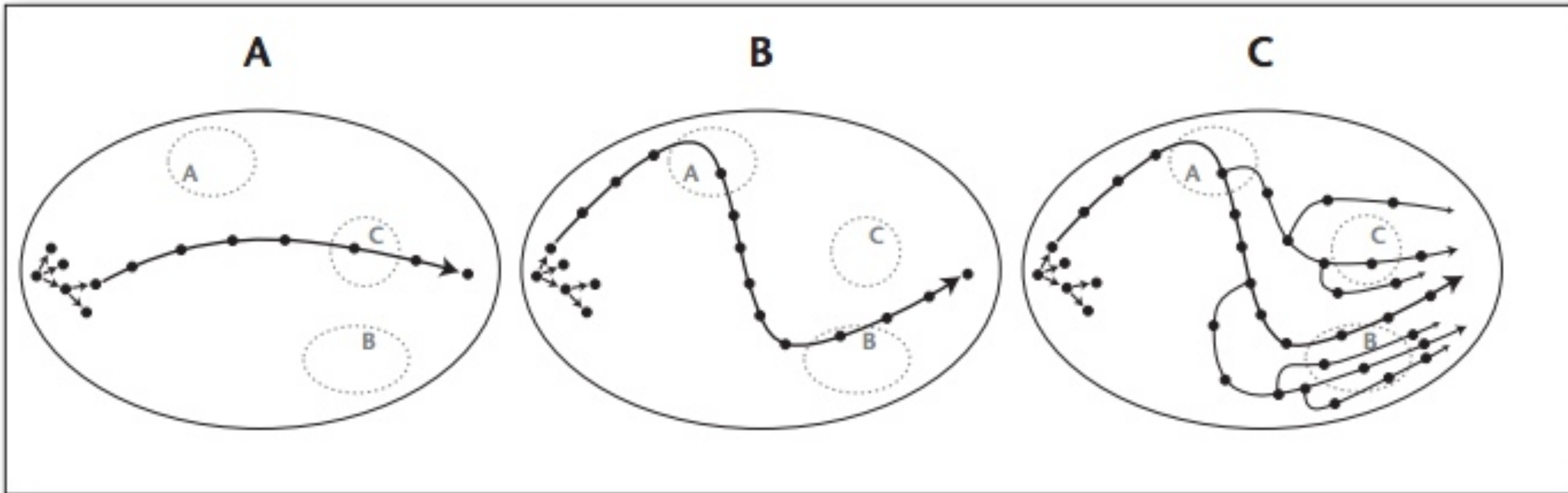


Figure 1. The Experience Management Problem Is to Compute Trajectories through State Space.

a. A possible narrative trajectory through state space. b. A possible narrative trajectory that visits states deemed favorable and avoids states deemed unfavorable. c. Accounting for player interaction.

Also see http://www.interactivestory.net/ for a neat demo of interactive narrative

AI Assistant for Visitors

Reference: Nadella, Satya. Hit Refresh. HarperCollins, 2017.

The book excerpt my synopsis refers to:

(about Microsoft's researcher, Eric Horvitz) "*His experiment was to make it easier for a visitor to find him, and to free up his human assistant for more critical work than the mundane task of constantly giving directions. So, to visit his office, you enter the ground floor lobby whee a camera and computer immediately notices you, calculates your direction, pace, and distance and then makes a prediction so that an elevator is suddenly waiting for you. Getting off the elevator a robot says hello and asks if you need help finding Eric's desk among the confusing corridors and warren of surrounding offices. Once there, a virtual assistant has already anticipated your arrival, knows Eric is finishing up a phone call, and asks if you'd like to be seated until Eric is available. The system received some basic training but, over time, learned to learn on its own so that programmers were not needed. It was trained, for example, to know what to do if someone in the lobby pauses to answer a call or stops to pick up a pen that's fallen on the floor. It begins to infer, to learn, and to program itself.*"

This excerpt from Hit Refresh highlights an example of AI planning in the limited scope of recognizing and welcoming visitors who want to meet Horvitz. When thinking about AI planning, a key factor that comes to mind is the intelligent system's ability to be flexible while charting the sequence of steps or actions it needs to undertake to achieve its goal (which in this case is delivering the visitor to Horvitz). The steps that need to be undertaken are detailed in the excerpt itself (calling the elevator, directing the visitor to Horvitz's desk), and we know that the system is an example of AI planning because it has trained itself to deal with new, uncertain situations that may arise (like the visitor temporarily pausing on their way to the elevator).

Interactivity, communication, stepping stones

AI Planning in Outer Space

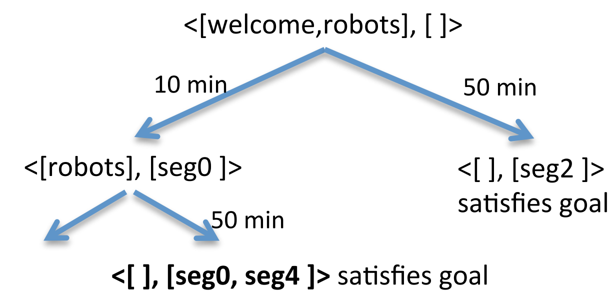
The Hubble Space Telescope, which has been in orbit since 1990, uses a planning and scheduling software called Spike. Spike was developed for the Hubble Telescope specifically but was also designed as a general planner for astronomy problems. Specifically, the Hubble Telescope is scheduling tens of thousands of observations per year, and must do this scheduling around various “operational and scientific constraints.” Thus Spike is a program installed within the telescope that takes into account these constraints in order to plan where the telescope should aim at what time. I think that this is a very cool application, especially considering that it has been in use since 1990 and will be in use until the Hubble Telescope is replaced in 2021. The constraints which this program are taking into account are very grand phenomenon, such as the movement of objects within the universe, and technical issues such as lighting that I would think are difficult to know from Earth without seeing it firsthand. In all, I think that it is very impressive that the Hubble Telescope can use Spike to make decisions on its own, and so we can be fairly confident that it is capturing the best images that it can (with little direct interaction and manipulation from Earth).

"SPIKE: Intelligent Scheduling of Hubble Space Telescope Observations", by Mark Johnston and Glenn Miller. (in Intelligent Scheduling, ed. M. Fox and M. Zweben, San Francisco: Morgan-Kaufmann, ISBN 1-55860-260-7, 1994, pp 391-422.) http://www.stsci.edu/institute/software_hardware/spike/documents/spike-chapter3.pdf.

"Spike Planning And Scheduling System". Space Telescope Science Institute, 2018, http://www.stsci.edu/institute/software_hardware/spike/. Accessed 4 Sept 2018

4. This question investigates using graph searching to design video presentations. Suppose there exists a database of video segments, together with their length in seconds and the topics covered, set up as follows:

Segment	Length	Topics covered
seg0	10	[welcome]
seg1	30	[skiing, views]
seg2	50	[welcome, artificial_intelligence, robots]
seg3	40	[graphics, dragons]
seg4	50	[skiing, robots]



Represent a node as a pair:

$$\langle To\_Cover, Segs \rangle ,$$

Taken from Poole and Mackworth, https://artint.info/2e/html/ArtInt2e.Ch3.S11.html

where $Segs$ is a list of segments that must be in the presentation, and To_Cover is a list of topics that also must be covered. Assume that none of the segments in $Segs$ cover any of the topics in To_Cover.

The neighbors of a node are obtained by first selecting a topic from To_Cover. There is a neighbor for each segment that covers the selected topic. [Part of this exercise is to think about the exact structure of these neighbors.]

For example, given the aforementioned database of segments, the neighbors of the node $\langle [welcome, robots], [] \rangle$, assuming that $welcome$ was selected, are $\langle [], [seg2] \rangle$ and $\langle [robots], [seg0] \rangle$.

Thus, each arc adds exactly one segment but can cover one or more topics. Suppose that the cost of the arc is equal to the time of the segment added.

The goal is to design a presentation that covers all of the topics in $MustCover$. The starting node is $\langle MustCover, [] \rangle$, and the goal nodes are of the form $\langle [], Presentation \rangle$. The cost of the path from a start node to a goal node is the time of the presentation. Thus, an optimal presentation is a shortest presentation that covers all of the topics in $MustCover$.

(a) Suppose that the goal is to cover the topics $[welcome, skiing, robots]$ and the algorithm always select the leftmost topic to find the neighbors for each node. Draw the search space expanded for a lowest-cost-first search until the first solution is found. This should show all nodes expanded, which node is a goal node, and the frontier when the goal was found.

Deep Space 1

Deep Space 1 (DS1) was a NASA spacecraft launched in the late 90s. Its primary mission was to test 12 recently developed technologies in space. This spacecraft was able to initiate, generate, execute and achieve high-level goals expanding these into flexible plans on board. It was able to demonstrate and detect failure, and act in respond to it, generating another plan after this encounter. Another of its capabilities is to update the plans and goals sent from Ground (NASA center) and its state.

DS1 project was very important, as it was the first mission of the New Millennium program (NMP), and it tested the capabilities of 12 new technologies. A special mention goes to the ion engine, as it was the first interplanetary spacecraft that successfully used it. After DS1, it is extended to other spacecrafts built for complex flights that involve collision avoidance like exploring comets and asteroids.

References:

http://www.inf.ed.ac.uk/teaching/courses/plan/slides/Applications.ppt

https://www.jpl.nasa.gov/nmp/ds1/DS1_Primary_Mission.pdf

https://www.jpl.nasa.gov/nmp/ds1/DS1_conclusion.pdf

AI Planning with Udacity's Self-Driving Car

Udacity is working on a self-driving car, which presents a path planning problem for getting from point A to point B. The goal state of the problem would be the when the vehicle has arrived at the destination. A procedural course of action (a traversal of a geographical path) to get the vehicle from its start state to its goal state must be determined.

Another goal would be to avoid collisions with other objects at all costs. This involves is a prediction component that estimates the trajectories of other objects in view. This prediction forms the belief state used to determine the next steps of the vehicle (i.e. whether to slow down, speed up, change lanes, etc.), and is calculated using both senor input and previous predictions in the knowledge base.

It seems to me that the goal of avoiding collisions while getting into the correct lanes and making the correct turns is much harder than the goal of finding a path from the start location to the end location, because it involves constant input and reinforcement from sensor data as opposed to just calculating the path based on an existing knowledge base (i.e. a map of the roads in the area).

Source: https://medium.com/intro-to-artificial-intelligence/path-planning-project-udacitys-self-driving-car-nanodegree-be1f531cc4f7

Also see http://mchrbn.net/

Introduction on a fielded AI-planning application 'Waymo'

'Waymo is a self-driving technology development company. And driveless car service is announced to be available for public use by the end of 2018.' [1] It uses on-board sensors such as cameras, rader sensors, and laser rangefinders to sense the surrounding environment of the vehicle, controls the steering and speed according to the road condition, and uses detailed maps to navigate, which allows the Waymo car driving on the road and arriving destination safely.

In my opinion, human drivers often misbehave when it comes to emergency situations. The sophisticate control system of the product plan and conduct the method dealing with emergency, which is more responsive than human drivers and can coordinate inner systems to prevent the vehicle from getting out of control. However, the 'Waymo' car cannot perform as well as normal when it runs on roads with blurred road lines. So I think manual intervention is necessary at the beginning period of the use by public. And with the facilities of roads improving, manual intervention can descend.

References:

[1] https://en.wikipedia.org/wiki/Waymo

[2] John Markoff (October 9, 2010). "Google Cars Drive Themselves, in Traffic". The New York Times. Retrieved October 11, 2010.

[3] https://www.bloomberg.com/news/articles/2014-04-03/the-problem-with-self-driving-cars-they-dont-cry#r=hpt-ls

A number of you found the ICAPS conference

https://icaps19.icaps-conference.org/

ICAPS
http://www.icaps-conference.org/

The International Conference on Automated Planning and Scheduling (ICAPS) is one conference/organization attempting to further research in the automated planning field. A large part of how they set out to accomplish this is by supplementing the annual conference with teaching endeavors throughout the rest of the year. The conference organizes various training, plans competitions, and awards scholarships in an attempt to get younger people involved in the field. The focus that ICAPS puts on attracting young talent to the field of automated planning is what really sets them apart from some of the other organizations I found. ICAPS is very focused on not just consolidating information among the top researchers in the field, but also disseminating that information and making an effort to expand their community.

AI Planning – NASA

In this paper, https://www.aaai.org/Papers/ICAPS/2005/ICAPS05-005.pdf, NASA details the AI Planning in its Mars Exploration Rovers. The rovers are tasked with achieving "the maximum number of highest-priority science goals" while considering their limited physical resources, their largely unstudied environment, and their desire to be efficient with their time. NASA's AI Planning system uses mixed-initiative planning to accomplish this by allowing for flexible plan editing, enforcing active constraints while plan editing, implementing a time-out if searches take too long, and automatically adapting for user preferences that satisfy constraints. I think that this planning system is good for situations where a robot needs to be risk-averse, since it has built-in safeguards but is still flexible. However, I think this system is highly dependent on well-defined science priorities communicated to NASA by scientists, so it would be difficult to replicate NASA's success in AI Planning without involving domain experts, which is a barrier to adaption.

AI-planning scheduler

An example of a commercial AI-planning application is Stottler Henke's Aurora software system (https://www.stottlerhenke.com/products/aurora/).

The Aurora system is an "intelligent planning and scheduling software solution" designed for complex manufacturing processes. Stottle Henke claims the software uses a large knowledge base in conjunction with advanced heuristics to more intelligently develop efficient schedules when compared to simpler rule based systems. It seems as if the system utilizes propositional calculus to merge real-world constraints and possible solutions in order to identify the optimal path. The website claims the system can group atomic operations and constraints for efficiency.

According to the website, the system was developed for NASA and is currently used by Boeing, among others.

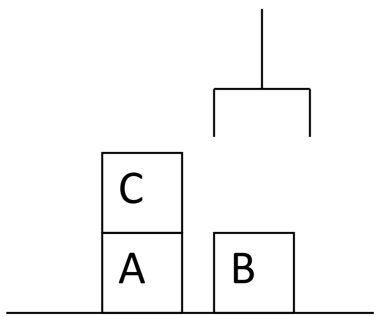
Reference: https://formal.iti.kit.edu/~beckert/teaching/KI-fuer-IM-WS0405/09Planning.pdf

This article gives a detailed tutorial of how AI planning works as well as some real world examples of how AI planning is used in the real world. This articles starts from the difference of Searching and planning and then makes a detailed tutorial on how to used STRIPS operators and Partial-Order planning. In my opinion, this article is significant because after reading this article, I not only had a big picture of how AI planning is working but I could also use AI planning to solve real-world problems. This can be a lot of help for my future study in the field of AI.

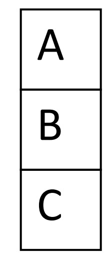
Total order plan

UNSTACK(C,A) → PUTDOWN(C) → PICKUP(B) → STACK(B,C) → PICKUP(A) →STACK(A,B)

ON(C,A)
ONTAB(A)
ONTAB(B)
CLEAR(B)
CLEAR(C)
HANDEMPTY

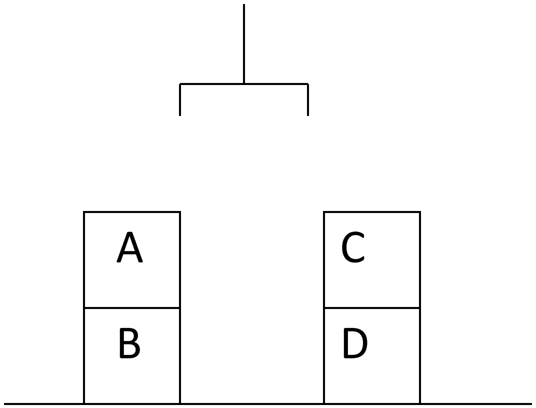


ON(A,B)
ON(B,C)

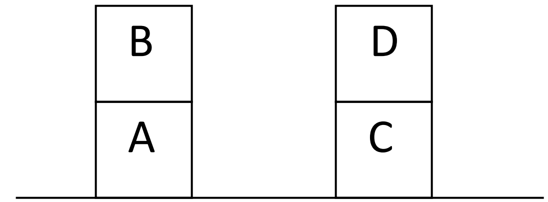


First order representations are more powerful than propositional representations

ON(?X, ?Y), where variables ?X and ?Y can be instantiated by different constants, such as A, B, C instead of ON-A-B, ON-B-C, ON-B-A



Initial State

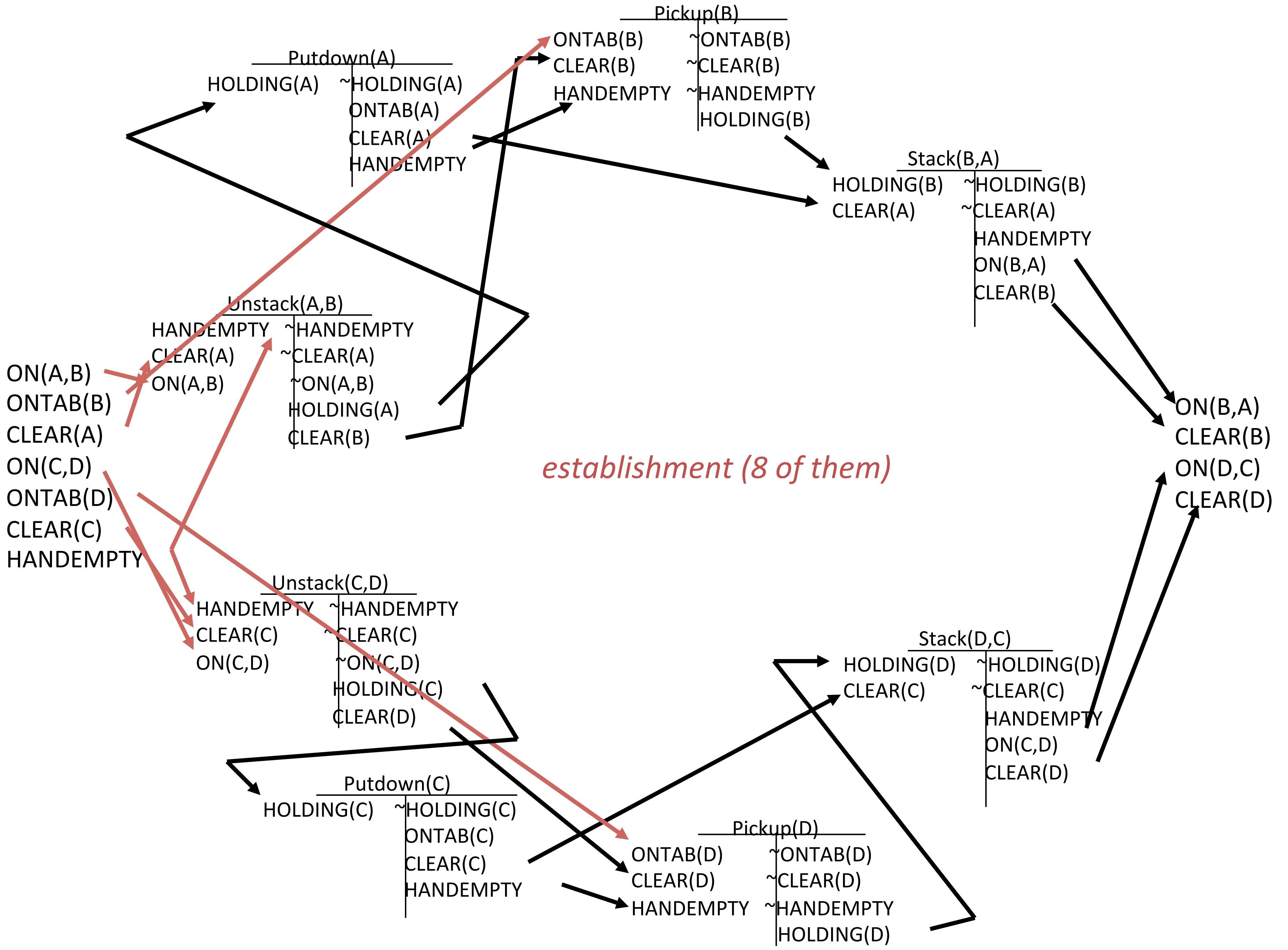


Goal spec

ON(A,B)
ONTAB(B)
CLEAR(A)
ON(C,D)
ONTAB(D)
CLEAR(C)
HANDEMPTY

ON(B,A)
CLEAR(B)
ON(D,C)
CLEAR(D)

This does not require a total order plan, but rather a more flexible partial order plan
Poole and Mackworth 6.5



Putdown(A)
HOLDING(A)
~HOLDING(A)
ONTAB(A)
CLEAR(A)
HANDEMPTY
Pickup(B)
ONTAB(B)
CLEAR(B)
HANDEMPTY
~ONTAB(B)
~CLEAR(B)
~HANDEMPTY
HOLDING(B)
Stack(B,A)
HOLDING(B)
CLEAR(A)
~HOLDING(B)
~CLEAR(A)
HANDEMPTY
ON(B,A)
CLEAR(B)
Unstack(A,B)
HANDEMPTY
CLEAR(A)
ON(A,B)
~HANDEMPTY
~CLEAR(A)
~ON(A,B)
HOLDING(A)
CLEAR(B)
ON(A,B)
ONTAB(B)
CLEAR(A)
ON(C,D)
ONTAB(D)
CLEAR(C)
HANDEMPTY
establishment (8 of them)
ON(B,A)
CLEAR(B)
ON(D,C)
CLEAR(D)
Unstack(C,D)
HANDEMPTY
CLEAR(C)
ON(C,D)
~HANDEMPTY
~CLEAR(C)
~ON(C,D)
HOLDING(C)
CLEAR(D)
Stack(D,C)
HOLDING(D)
CLEAR(C)
~HOLDING(D)
~CLEAR(C)
HANDEMPTY
ON(C,D)
CLEAR(D)
Putdown(C)
HOLDING(C)
~HOLDING(C)
ONTAB(C)
CLEAR(C)
HANDEMPTY
Pickup(D)
ONTAB(D)
CLEAR(D)
HANDEMPTY
~ONTAB(D)
~CLEAR(D)
~HANDEMPTY
HOLDING(D)

https://pdfs.semanticscholar.org/bbb5/b2456ff1b6e493d9b162e17817e70005877f.pdf

One of the most notable and disruptive appearances of automated planning has been the increased usage of quantitative automatic trading. Many large investment firms and banks are becoming increasingly dependent on strong technology divisions to make their systems as 'smart' and fast as possible. Many quant trading systems can determine complex states through instantaneous aggregation of large quantities of data. Thus, they can often better predict the movements of stocks and react accordingly. This is changing much of the workforce in the financial industry, particularly in areas near my hometown in a NY suburb. Many large corporations are looking for traders with strong quantitative and programming skills to best manage their systems.

Asset (?Stock, ?Shares), Trade(?Stock, ?X) ➔ Asset (Stock, ?Shares - ?X)

Strategic planning, rather than impulsive response

A number of you found AI Planning MOOC(s)

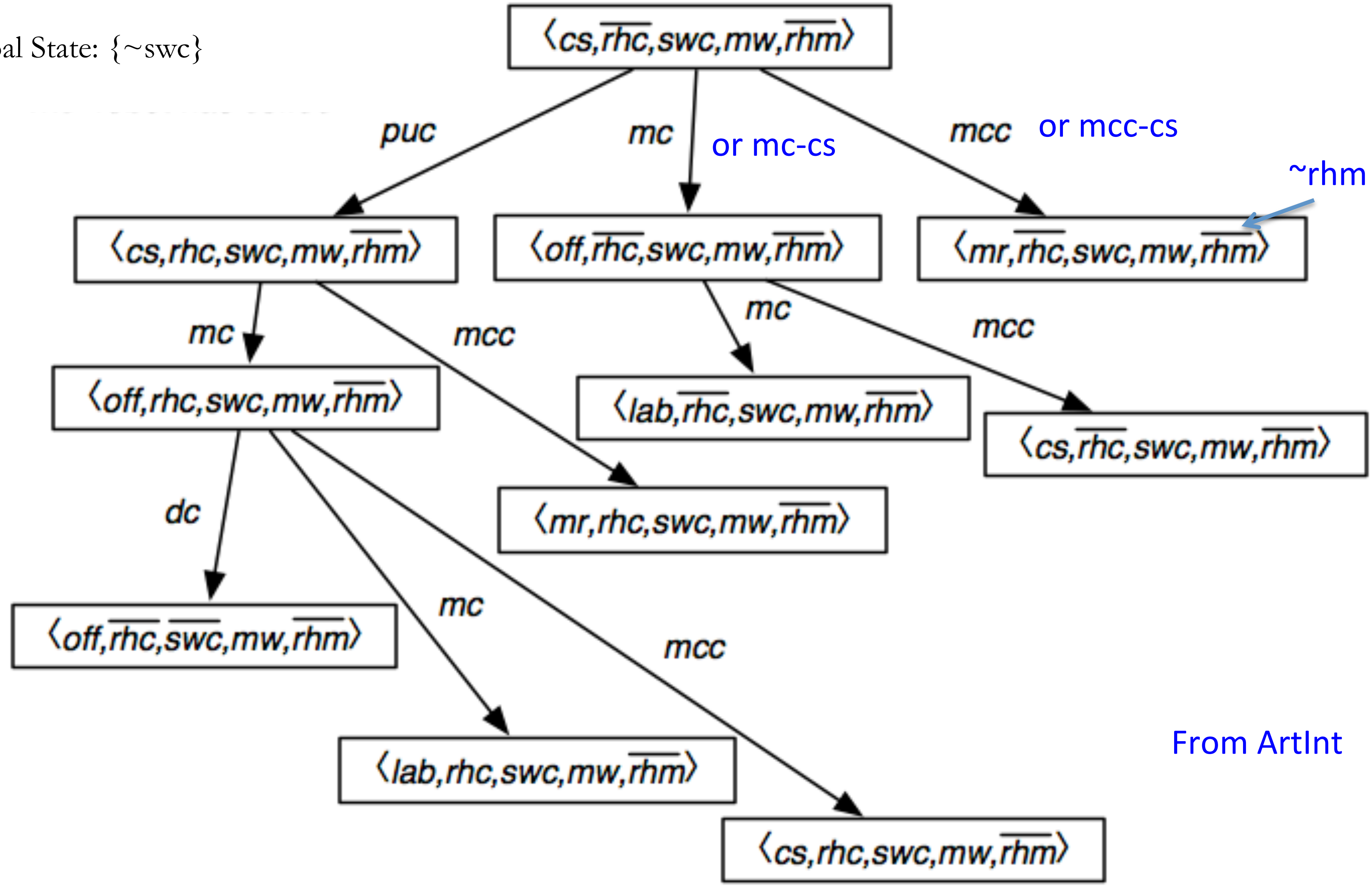
AI Planning MOOC

This MOOC (http://media.aiai.ed.ac.uk/Project/AIPLAN/) provides an overview of AI planning and specifically addresses **plan-space search**. In this kind of search, the states are partial plans. By using plan refinement operations we can find arcs through the graph to arrive at a solution. Overall, plan space search problem is composed of two sub-problems: identification of actions that need to be performed and organization of these actions. A concept I have found particularly interesting is specifying constraints that will be used to find the solution. These include clobbering (eliminating condition needing for goal state) and flaws/threats (unsatisfied or unlinked sub goals). The Plan Space Planning Algorithm defined in (http://media.aiai.ed.ac.uk/Project/AIPLAN/resource/3.1-3.7-Plan-Space-Search.pdf) is predicated on finding a solution by navigating against these specified constraints. So as opposed to searching based on goal criteria, we are also relying more on constraint criteria to solve the problem for us.

Planning under constraints – Poole and Mackworth 6.4

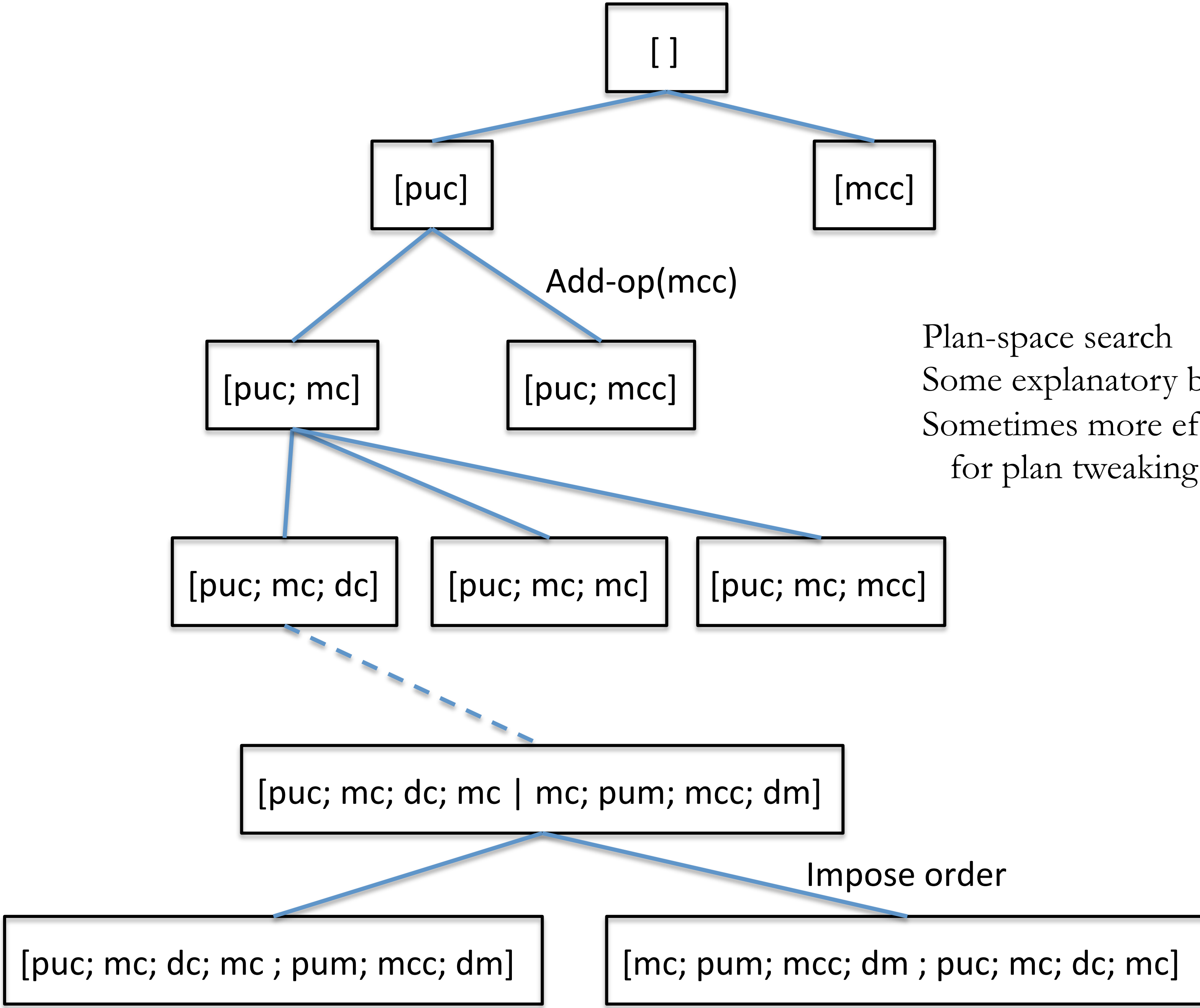
Initial State: {cs, ~rhc, swc, mw, ~rhm}

Goal State: {~swc}



From ArtInt

Figure 6.2 Part of the search space for a state-space planner



Plan-space search
Some explanatory benefit
Sometimes more efficient
for plan tweaking

Games (Chess, AlphaGo, Contract Bridge…)
Retail shopping
Roomba
Trip planning (including car pooling)
Undersea exploration
Science fiction (Her, Asimov, …)
Shakey and other history