

规划 (AI Planning)

University of Science and Technology of China

April 16, 2014

1 经典规划 (Classical Planning)

- 规划问题
- 状态空间规划 (State-Space Planning)
- 规划空间规划 (Plan-Space Planning)
- 规划图技术 (Planning-Graph Techniques)
- Planning as { SAT, CSP, ILP, ... }

2 现实世界的规划与行动

1 经典规划 (Classical Planning)

- 规划问题
- 状态空间规划 (State-Space Planning)
- 规划空间规划 (Plan-Space Planning)
- 规划图技术 (Planning-Graph Techniques)
- Planning as { SAT, CSP, ILP, ... }

2 现实世界的规划与行动

1 经典规划 (Classical Planning)

- 规划问题
- 状态空间规划 (State-Space Planning)
- 规划空间规划 (Plan-Space Planning)
- 规划图技术 (Planning-Graph Techniques)
- Planning as { SAT, CSP, ILP, ... }

2 现实世界的规划与行动

- 规划（智能规划，AI Planning）是人工智能一个重要领域，起源于 60 年代
- Planning in AI is *decision making* about the *actions* to be taken
- 规划器分为：
 - Domain-specific: 一般针对特定领域专门设计，高效，已大量应用
 - Domain-independent: 通用的求解器，效率相对不高，目前还很难实际应用。但仍然有重要意义：
 - 领域无关是对规划问题共性的研究，其成果可以用来提高领域相关规划器的效率；
 - 研究一般规划行为所反映的理性行为；
 - 便宜的规划器（针对领域定制，一般需花费更多）；
 - 研究和设计自治智能体（如，智能机器人）就需要领域无关
 - Configurable: 在领域无关基础上，根据领域特点增加一些控制信息
- 在 AI 领域，规划 (Planning) 一般指 Domain-independent 和 Configurable Planning

智能规划

- 规划领域主要两个工作：如何方便的（紧凑并且方便求解）表达和如何高效求解（搜索）

- 表示：状态、行动

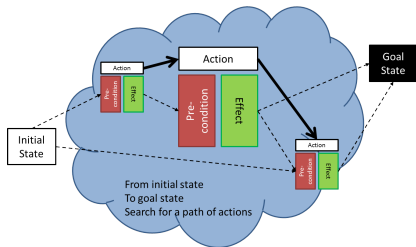
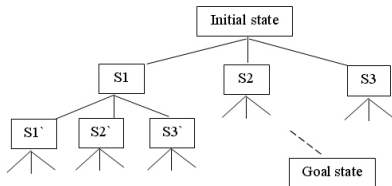
- **状态**：（动态）系统在某时刻的情况，问题的状态描述

初始状态、目标状态

- **行动**：状态空间上的部分映射，对状态描述进行变换的一组操作

- 求解：计算（搜索）出从初始状态变化到目标状态的一个操作序列

- 状态空间搜索，偏序规划，规划图，Planning as {SAT, CSP, ILP, ... }, 等



经典规划基本假设

- 规划问题非常复杂，为简化问题而提出一些简化的假设（经典规划基本假设）：
 - (A0) Finite system: finitely many states, action, events
 - (A1) Fully observable: always know current state
 - (A2) Deterministic: each action has only one outcome.
 - (A3) Static (no exogenous events): no changes but the controller's actions
 - (A4) Attainment goals: a set of goal states
 - (A5) Sequential plans: a plan is a linearly ordered sequence
 - (A6) Implicit time: no time durations
 - (A7) Off-line planning: planner doesn't know the execution status
- 经典规划的任务，简单说就是：Computing paths from an initial state to a goal state in the transition graph.
 - 已知 transition graph，用 Dijkstra 算法（时间复杂度为 $O(n \log n)$ ）
 - 但状态空间一般非常大 ($10^9, 10^{12}, 10^{15}, \dots$)，所以无法构造出整个 transition graph
 - 规划算法要避免构造出整个 transition graph

- 经典规划只考虑确定的、静态的、有限的、完全可观察的、离散环境的、目标受限和忽略时间的状态转移系统
- 经典规划的主要问题包括：
 - 如何在不显示枚举的情况下，表达状态和动作: compact and easy searched
 - 如何有效的进行解的搜索
- 即使在受限条件下，规划问题的求解仍然是非常困难的，奢求用经典规划技术来解决实际规划问题是不现实的
- 需要用一种通用的表达方式，compact 表达状态和动作，并且便于搜索求解，一般的思路是：
 - 用 features 来表达单个状态。状态为 features 特定值的集合
 - 用 operators 来计算状态转移。operators 为 features 之间的转换关系
 - 不需显示表达出所有状态，只给出初始状态，然后通过 operators 计算出需要的状态

经典规划的集合论表达 (Set-Theoretic Representation)

- 用有限的命题符号集 (L) 来表达状态转移系统:
 - $S \subseteq 2^L$, 状态 s 为 L 的子集, 命题的集合, 表示在 s 上真的命题
 - action 是三元组 $\langle \text{precond}, \text{effect}^-, \text{effect}^+ \rangle$
 - S 的性质: 对于任意状态 s 和可应用于 s 的行动 a , 集合 $(s - \text{effect}^-(a)) \cup \text{effect}^+(a) \in S$
 - 如果 a 可应用于 s , 则状态转移函数为 $\gamma(s, a) = (s - \text{effect}^-(a)) \cup \text{effect}^+(a)$, 否则无定义
- 规划问题是在状态转移系统的基础上增加初始状态和目标状态集
 - $s_0 \in S$
 - $g \subseteq L$, 目标状态集合为 $S_g = \{s \in S \mid g \subseteq s\}$
- 规划是一个动作序列 $\pi = \langle a_1, \dots, a_k \rangle$, 如果 $g \subseteq \gamma(s_0, \pi)$, 则规划 π 是此规划问题的一个解
- 并不是每一个状态转移系统都可以用集合论表达, 但可以构造一个与其等价的系统, 而此系统可以用集合论表达

经典规划的经典表达 (Classical Representation)

- 经典表达是对集合论表达的推广，使用一阶逻辑符号，用公式来表达状态集和行动，通过语义解释来确定具体的状态和行动
 - 用一阶逻辑语言（有限多谓词，变元，常元，没有函数符号）来描述系统。一个状态是一个 **grounded atoms** 集合。谓词又分为状态谓词 (**fluent**) 和刚性关系 (**rigid relation**)，前者是状态集的函数，后者不随状态变化而变化
 - 规划操作是一个三元组 $o = (name(o), precondition(o), effects(o))$ ，其中：
 - $name(o)$ ，操作的名字，形如 $n(x_1, \dots, x_k)$
 - $precond(o)$ 和 $effects(o)$ 分别是 o 的前提和效果，都是文字集。刚性关系不能出现在任何操作 o 的效果中。action 是 ground instance of an operator
- 例如： $move(r, l, m)$ 表示机器人 r 从位置 l 移动到位置 m ：
 - $precond$: $adjacent(l, m), at(r, l), \neg occupied(m)$
 - $effect$: $at(r, m), occupied(m), \neg occupied(l), \neg at(r, l)$
- 经典表达 **grounding** 后与集合论表达等价，不过 **grounding** 结果可能指数增大
- 状态变量表达 (State-Variable Representation):
 - 状态为向量值，动作为函数映射。经典表达与状态变量方法在表达能力上是等价的

发展历史

- 自动规划最初受自动推理证明很大的影响，用 **Situation Calculus** 的方式对初始状态、目标状态和行动做公理化描述，使用归结定理证明来构造求解
- 然而，这种方式遇到框架问题的困难，从而引入对经典规划的描述问题，其目的之一就是为框架问题提供一个简单的解法：
 - **STRIPS** 假设：在效果中没提及的每个文字保持不变
- **STRIPS** 就是这方面早期的工作。这里介绍的经典表达和 **STRIPS** 有同样的表达能力（**STRIPS** 不同与自动定理证明不同的是，将解的搜索和对系统的逻辑描述分离了）
- **ADL** 权衡了一阶表达能力和推理的复杂性，之后扩展的 **PDDL**，为规划问题的标准描述语言
- **PDDL** (The Planning Domain Definition Language) 是由 **IPC** (International Planning Competition 两年一届) 定义的标准语言。已有一大批基于 **PDDL** 的通用规划器
- **STRIPS** 或 **PDDL** 方式与 **Situation Calculus** 方式都可以刻画动态系统，两者在规划、预测方面的效果相同，但如果需要更复杂的推理（如诊断），则只能使用 **Situation Calculus**

经典规划的复杂性

经典规划存在解 (PLAN-EXT), 存在固定长度解 (PLAN-LEN) 问题的计算复杂性

Kind of representation	How the operators are given	Allow negative effects?	Allow negative preconditions?	Complexity of PLAN-EXISTENCE	Complexity of PLAN-LENGTH
Classical rep.	In the input	Yes	Yes/no	EXPSpace-complete	NEXPTIME-complete
		No	Yes	NEXPTIME-complete	NEXPTIME-complete
			No	EXPTIME-complete	NEXPTIME-complete
			No ^d	PSPACE-complete	PSPACE-complete
	In advance	Yes	Yes/no	PSPACE ^b	PSPACE ^b
		No	Yes No No ^d	NP ^b P NLOGSPACE	NP ^b NP ^b NP
Set-theoretic or ground classical rep.	In the input	Yes	Yes/no	PSPACE-complete	PSPACE-complete
		No	Yes No No ^d /no ^c	NP-complete P NLOGSPACE-complete	NP-complete NP-complete NP-complete
	In advance	Yes/no	Yes/no	Constant time	Constant time
State-variable rep.	In the input	Yes ^d	Yes/no	EXPSpace-complete	NEXPTIME-complete
	In advance	Yes ^d	Yes/no	PSPACE ^b	PSPACE ^b
Ground state-variable rep.	In the input	Yes ^d	Yes/no	PSPACE-complete	PSPACE-complete
	In advance	Yes ^d	Yes/no	Constant time	Constant time

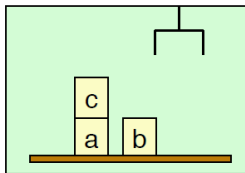
^a No operator has > 1 precondition.

^b With PSPACE- or NP-completeness for some sets of operators.

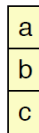
^c Each operator with > 1 precondition is the composition of other operators.

^d There is no way to keep the operators from having negative effects.

Example: blocks world



Initial state



goal

初始状态:

$onTable(A) \wedge on(C, A) \wedge onTable(B) \wedge clear(B) \wedge clear(C) \wedge handempty$

目标: $on(A, B) \wedge on(B, C)$

Example: blocks world (con't)

操作:

unstack(x,y)

Pre: $\text{on}(x,y)$, $\text{clear}(x)$, handempty

Eff: $\sim\text{on}(x,y)$, $\sim\text{clear}(x)$, $\sim\text{handempty}$,
 $\text{holding}(x)$, $\text{clear}(y)$

stack(x,y)

Pre: $\text{holding}(x)$, $\text{clear}(y)$

Eff: $\sim\text{holding}(x)$, $\sim\text{clear}(y)$,
 $\text{on}(x,y)$, $\text{clear}(x)$, handempty

pickup(x)

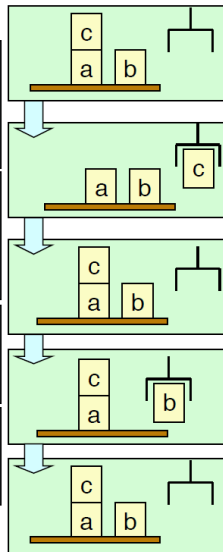
Pre: $\text{ontable}(x)$, $\text{clear}(x)$, handempty

Eff: $\sim\text{ontable}(x)$, $\sim\text{clear}(x)$, $\sim\text{handempty}$, $\text{holding}(x)$

putdown(x)

Pre: $\text{holding}(x)$

Eff: $\sim\text{holding}(x)$, $\text{ontable}(x)$, $\text{clear}(?x)$, handempty



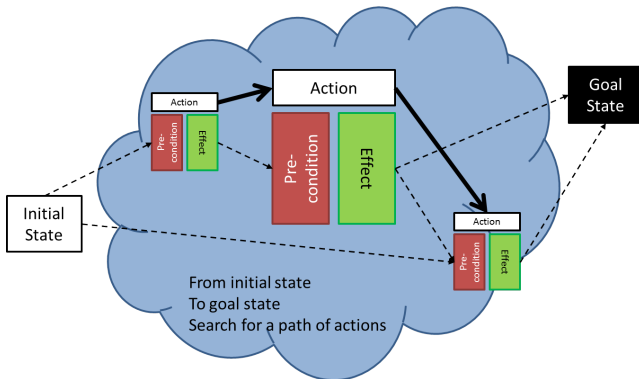
1 经典规划 (Classical Planning)

- 规划问题
- 状态空间规划 (State-Space Planning)
- 规划空间规划 (Plan-Space Planning)
- 规划图技术 (Planning-Graph Techniques)
- Planning as { SAT, CSP, ILP, ... }

2 现实世界的规划与行动

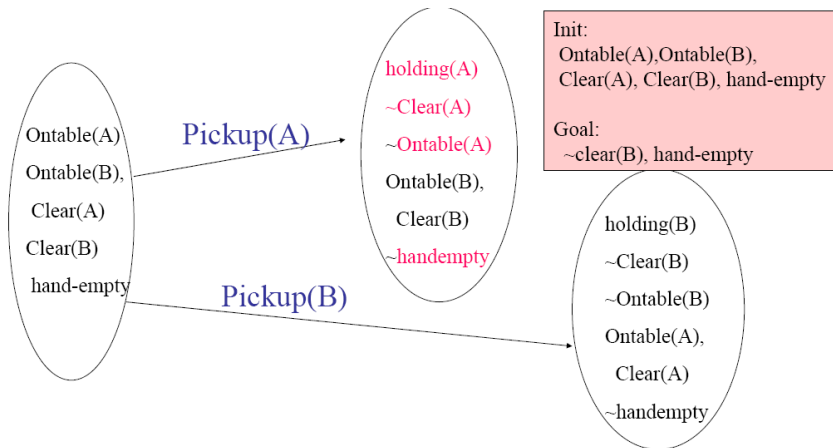
状态空间搜索方式

- 在状态转移图中搜索从初始状态到目标状态的一条路径
 - 前向搜索 (Forward Search): 从初始状态出发向前搜索
 - 后向搜索 (Backward Search): 从目标状态向后搜索
 - 启发式搜索: 利用启发式函数 (例如, 估计从状态到目标的距离) 进行前向或后向搜索



前进规划 (progression)

- 从初始状态开始，考虑所有可行的行动，进行深度或广度搜索， $\gamma(s, a)$
- 可靠并完全

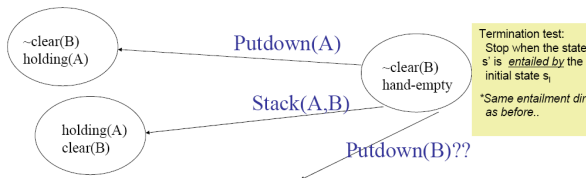


回归规划 (regression)

- 从目标出发，将当前目标还原为回归子目标， $\gamma^-(g, a)$
- 与 Forward 相比，一般有较小的分支数
- An action a is relevant for a goal g if
 - $g \cap effects^+(a) \neq \emptyset$
 - $g^+ \cap effects^-(a) = \emptyset$ and $g^- \cap effects^+(a) = \emptyset$

Init:
Ontable(A), Ontable(B),
Clear(A), Clear(B), hand-empty

Goal:
 $\sim clear(B)$, hand-empty



- 类似 backward, 每次选择一个相关的 action, 但只将 action 的 precondition 作为下一步迭代的目标
- 当状态满足 precondition 后执行此动作, 并不再回溯
- STRIPS 是不完备的
- 算法过程: return a sequence of actions that transforms s into g
 - ① Calculate the difference set $d = g - s$
 - ② If d is empty, return an empty plan
 - ③ Choose action a whose add-list has most formulas contained in g
 - ④ $p' = STRIPS(s, precondition\ of\ a)$
 - ⑤ Compute the new state s' by applying p' and a to s
 - ⑥ $p = STRIPS(s', g)$
 - ⑦ return $p'; a; p$

1 经典规划 (Classical Planning)

- 规划问题
- 状态空间规划 (State-Space Planning)
- 规划空间规划 (Plan-Space Planning)
- 规划图技术 (Planning-Graph Techniques)
- Planning as { SAT, CSP, ILP, ... }

2 现实世界的规划与行动

- 状态空间是最直接的搜索方式，规划是状态转移图中的一条路径，规划求解自然也就是状态空间上的搜索问题
- 但很多时候，状态空间上的搜索是在以不同的顺序不断说明一组行动不可行
- 因此提出规划空间：节点为局部具体化的规划，弧为规划的求精操作（refinement operation）
- 求解操作基于极小承诺原则 (least commitment principle): don't commit to orderings, instantiations, etc., until necessary
- PSP (Plan-Space Planning) 算法：找到 plan 的缺陷，选择一个缺陷，找到解决缺陷的所有方法，选择一个方法，按此方法对 plan 求精。PSP 算法是可靠完全的

偏序规划 (Partial-Order Planning)

- 规划问题：给定 s, g ，要找到行动序列 a_1, \dots, a_n 使得：

- ① a_1 在 s 上可执行
- ② 执行 a_i 后 a_{i+1} 可执行
- ③ 执行 a_n 后 g 为真

前行，回归规划都是在过程中生成“部分计划”，满足 (2)，但只满足 (1) 或 (3) 之一；而偏序规划在过程中满足 (1) 和 (3)，可以不满足 (2)

- POP 算法：以伪计划 $P = (s, e)$ 为起点，在保持 P 为一个 POP 部分计划的前提下，不断向 P 中添加新的因果链，直至得到一个 POP 计划
- 因果链形如： $a_p \xrightarrow{Q} a_c$, $Q \in \text{eff}(a_p) \cap \text{pre}(a_c)$ 是 a_p 对 a_c 前提条件的贡献
- POP 算法，类似 PSP，不过对两类不同的缺陷（子目标和威胁）采取不同的处理方式。先处理子目标，再处理相应的威胁。

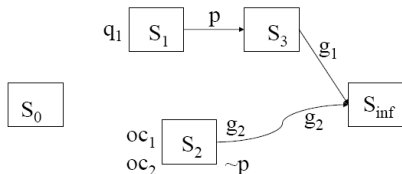
偏序规划

1. **Plan Selection**: Select a plan P from the search queue
2. **Flaw Selection**: Choose a flaw f (open cond or unsafe link)
3. **Flaw resolution**:
 - If f is an open condition,
choose an action S that achieves f
 - If f is an unsafe link,
choose promotion or demotion
 - Update P
 - Return NULL if no resolution exist
4. If there is no flaw left, return P

1. Initial plan:



2. Plan refinement (flaw selection and resolution):



状态空间和规划空间规划的比较

- 状态空间有限，而规划空间无限。但规划空间规划通常有较小的搜索空间
- 状态空间有显式的中间的状态，而规划空间中没有，没有明确的状态的概念。如果状态的概念清晰，可以有效利用领域知识和控制知识
- 规划空间，将对动作的选择和其顺序分离

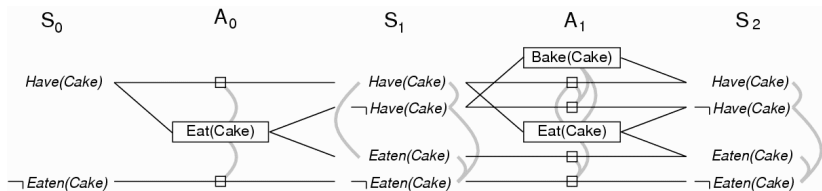
1 经典规划 (Classical Planning)

- 规划问题
- 状态空间规划 (State-Space Planning)
- 规划空间规划 (Plan-Space Planning)
- 规划图技术 (Planning-Graph Techniques)
- Planning as { SAT, CSP, ILP, ... }

2 现实世界的规划与行动

规划图 (Planning Graph)

- Planning Graph 是对 possible plans 约束的描述。“If valid plan exists, it's a subgraph of the planning graph.”，并且可以在多项式时间构造
- 规划图是有向分层图：
 - 两种节点：Proposition P , Action A
 - 三种边：Precondition: $P \rightarrow A$, Add: $A \rightarrow P$, Delete: $A \rightarrow P$
 - 两层：Proposition 和 Action
- Action level: 前提条件被上一层满足的 actions + no-op actions (for frame problem)



规划图方法

- 规划图方法是两个阶段交替执行：图扩展（graph expansion）阶段和解提取（solution extraction）阶段
 - 图扩展阶：正向扩展规划图直到目标状态的所有命题都出现为止
 - 解提取阶段：反向搜索规划图以求出规划解
- 规划图的构造过程：
 - ① 用初始状态真的所有文字构造 P_1 层；
 - ② 用前提被满足的行动构造 A_1 层；
 - ③ 再根据 A_1 层的 effects 构造 P_2 层（包括 no-ops，惯性）；
 - ④
- 同时还要维护一组互斥关系 (Mutual Exclusion relations)，来删除 incompatible propositions and actions：
 - Two actions (or literals) are mutually exclusive at some stage if no valid plan could contain both.
 - Two actions are mutex if:
 - Interference: one clobbers others' effect or precondition.
 - Competing needs: mutex preconditions.
 - Two propositions are mutex if: All ways of achieving them are mutex.

规划图方法 (con't)

- 规划图的特点：
 - Propositions and actions monotonically increase
 - Proposition mutex and action mutex relationships monotonically decrease
 - After some time k all levels are identical
- 规划图的构造时间是规划问题大小的多项式倍
- 规划解 (valid plan) 是规划图的一个子图，满足如下条件：
 - Actions at the same level don't interfere;
 - Each action's preconditions are made true by the plan;
 - Goals are satisfied.
- 规划图技术的算法：
 - 1 先构造规划图 (PG) 直到所有的目标都可达且 not mutex (If PG levels off first, fail),
 - 2 Search the PG for a valid plan,
 - 3 If not found, add a level to the PG and try again.

1 经典规划 (Classical Planning)

- 规划问题
- 状态空间规划 (State-Space Planning)
- 规划空间规划 (Plan-Space Planning)
- 规划图技术 (Planning-Graph Techniques)
- Planning as {SAT, CSP, ILP, ...}

2 现实世界的规划与行动

Planning as {SAT, CSP, ILP, ...}

- **Grounding** 以后的规划问题复杂度是 PSPACE-complete, 因为规划本身的长度有可能是指数的
- 只在有限长度 k 内计算 plan, 则复杂度为 NP-complete
- 所以将规划问题翻译为其他解决 NP-hard 的经典问题, Planning as:
 - **SAT: Propositional Satisfiability**
 - Situation Calculus 方式。将规划问题用命题逻辑方式表达, 使得每个 model 对应一个 plan
 - 结合规划图和 SAT。先构造 k 层的规划图, 将此图转化为 SAT 问题, 再用 SAT 求解器来计算里面的 plan。比较成功, 代表有 Black-Box (1998 年规划竞赛优秀者之一), SatPlan (2004, 2006 年冠军)
 - **CSP: Constraint Satisfaction**
 - 将规划问题翻译为一个 CSP 问题, 用其求解器求解。也可以通过规划图来翻译 GP-CSP。CSP encodings can be more compact.
 - 将 CSP 技术应用到规划中
 - **ILP: Integer Linear Programming**

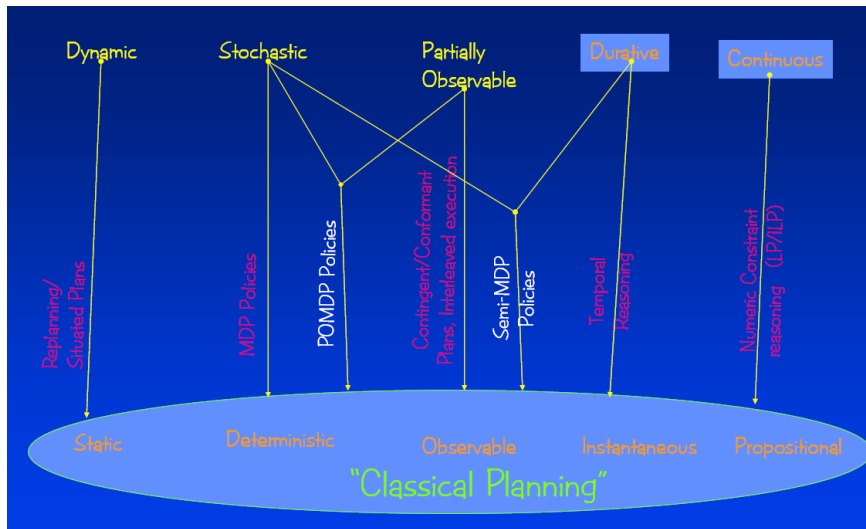
- 规划的核心问题是表示和求解（搜索）
- 经典规划用命题（或一阶）表示状态和行动
- 状态空间规划一般分为前进（**progression**）和回归（**regression**）规划
- 偏序规划为规划空间规划
- 规划图可以多项式时间构造出来，规划解是规划图的一个子图。规划图算法就是反复扩展规划图和尝试提取解的过程
- 一般规划问题（**Grounding** 以后）如果不限定规划长度，则计算复杂性为 **PSPACE-complete**；若限定长度，则为 **NP-complete**，可以等价翻译为其他解决 **NP-hard** 的经典问题，如 **SAT, CSP, ILP**，等

1 经典规划 (Classical Planning)

- 规划问题
- 状态空间规划 (State-Space Planning)
- 规划空间规划 (Plan-Space Planning)
- 规划图技术 (Planning-Graph Techniques)
- Planning as { SAT, CSP, ILP, ... }

2 现实世界的规划与行动

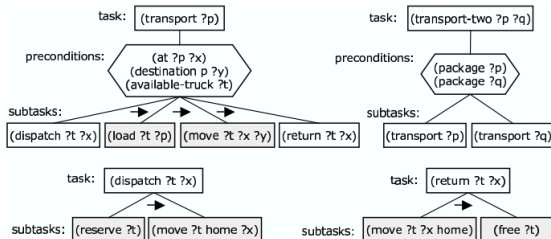
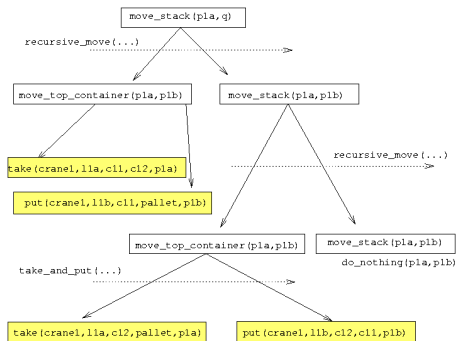
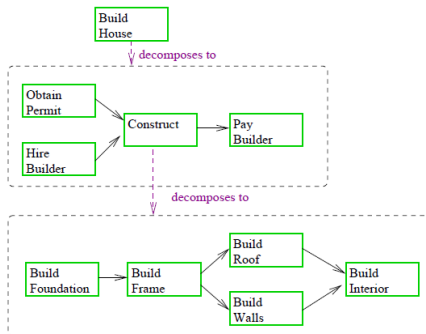
经典规划的扩展



分层任务网络规划 (Hierarchical Task Network (HTN) Planning)

- 分层任务网规划和经典规划类似，但增加一个方法集合 (method)，告诉系统，如何将一类任务分解为更小的子任务（可能有偏序约束），规划过程就是递归的将那些非原子任务分解到原子任务。
- HTN 规划基本过程：
 - ① Input a planning problem P
 - ② If P contains only primitive tasks, then resolve the conflicts and return the result. If the conflicts cannot be resolved, return failure
 - ③ Choose a non-primitive task t in P
 - ④ Choose an expansion for t
 - ⑤ Replace t with the expansion
 - ⑥ Find interactions among tasks in P and suggest ways to handle them. Choose one
 - ⑦ Goto 2
- 因为 HTN 可以方便的根据领域知识加入如何分解 task 的方法，HTN 应用很广，而且效果很好，在规划大赛中总是前几名

HTN Examples



不确定性领域中进行规划 (Planning under Uncertainty)

- 经典规划有如下三个限制：
 - 确定性 (determinism), 但现实情况更多是不确定的：
 - 由于信息的不完全, 我们不能确定具体在哪个状态。Uncertainty about the state of the world.
 - 行动的结果可能是不确定的。Uncertain effects for actions.
 - 在执行决策的过程中, 外部事件可能改换世界。External events.
 - 完全可观察性 (full observability), 但实际情况更多的是部分可见
 - 可达性目标 (reachability goals), 由于动作结果的不确定和可能的执行失败, 目标也相应扩展的更灵活, 主要有两种:
 - 效用函数, 规划求解的任务是使得效用函数的值最大化
 - 用时态逻辑公式表达目标, 规划求解的任务是使得, 规划中行动令此时态公式为真

Planning under Uncertainty (con't)

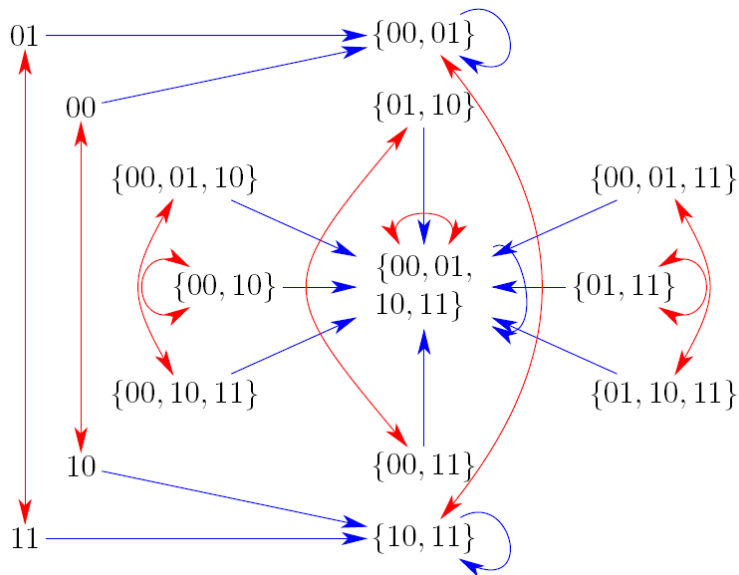
- 由于不确定性，规划的执行可能对应于多条不同的执行路径，需要规划算法能高效的分析所有动作各种可能的执行结果。可以有如下方式：
 - Re-planning: make a plan assuming nothing bad will happen, monitor for problems during execution, build a new plan if a problem is found, either re-plan to the goal state or try to patch the existing plan.
 - Conditional planning: deal with contingencies at planning time before they occur, every possible contingency is covered in the policy.
- 两种方式，前者没办法未雨绸缪，后者又考虑过甚（不可能真正处理完所有的可能），需要平衡
- 用概率来表达 uncertainty 能实现较好的平衡，Probabilistic planning: Plan ahead for likely contingencies that may need steps taken before they occur.

- 使用 MDP/POMDP 可以数学的刻画不确定性条件下的规划规划, 相关研究是持续多年的热点
- 对于使用 MDP/POMDP 也有不同声音:
 - MDP 需要考虑state space explosion 问题
 - require structured representations, exploit regularities in probabilities, rewards. 而 AI 的技术一般更natural, concise (STRIPS, Bayesian networks)
 - require structured computation, exploit regularities in policies, value functions, can aid in approximation.

非确定规划 (Nondeterministic Planning)

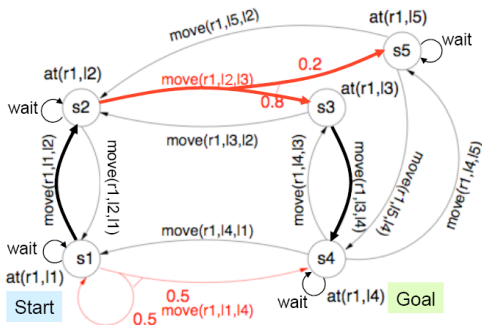
- 根据规划结果的质量可以有如下分类:
 - weak plan: 某种执行方式可能达到目标
 - strong cyclic plan: 最终都有可能达到终止状态, 且都为目标状态
 - strong plan: 是strong cyclic plan, 且没有cycles
 - conformant plan: 是strong plan, 且没有观察
- 在不同的要求下, 非确定规划问题又可以分类为:
 - Strong nondeterministic planning with full observability:
 - 由于有 full observability, 所以可以用 memoryless strategy $\pi : S \rightarrow O$.
 - 具体算法有两种: 1. 看作是 AND-OR search. 2. Dynamic programming (backward).
 - Conformant planning: planning without observability, EXPSPACE-complete
 - plans are sequences of actions
 - 不知道具体的 state, 用 belief state。具体的算法也只在 belief state 上处理。但 belief state 的空间非常大
 - Nondeterministic planning with partial observability, 有三种方法:
 - Reduction to full observability by viewing belief states as states.
 - Forward search in AND/OR trees.
 - Dynamic-programming style backward construction of solvable belief states, starting from goal belief states.

The Belief Space: An Example



概率规划 (Probabilistic Planning)

- 如果将不确定信息用概率的方式表达出来，就是 Probabilistic planning
- PDDL 也已扩展出 PPDDL (Probabilistic PDDL) 用概率表达不确定性 (2004 年)
- 可以直接使用 MDP/POMDP 方法，也可以使用其他方法，如 Planning based on Markov Decision Processes
 - 将规划问题表示称为一个优化问题来解决



机器人上的规划 (Planning in Robotics)

- 机器人涉及的规划问题：
 - Task planning
 - Path and motion planning, 包括: Navigation planning, Manipulation planning
 - 本质上是在参数空间中, 找一条路径
 - Perception planning
- 经典规划框架对于机器人应用来说: too hard and too easy
 - too hard: it is intractable;
 - too easy: action sequences are not adequate as a representation of a real robot's program.

- 分层任务网络（HTN）规划可以方便的根据领域知识加入如何分解任务的方法，从而可以解决实际问题
- 经典规划算法假设有完备的和正确的信息、确定性的和完全可观察的环境，实际问题通常违反这些假设
- MDP/POMDP 为不确定规划问题采用概率手段给出了统一的数学描述框架
- 经典规划框架对机器人应用来说：too hard and too easy