

Home仿真比赛ASP决策 及与其它方法的比较

隋志强

多智能体系统实验室

中国科学技术大学

August 24, 2010

lightwingszq@gmail.com

Outline

- Home仿真ASP决策逻辑程序分析
- 与其他方法(搜索， 策略求解) 的比较

Outline

- Home仿真ASP决策逻辑程序分析
- 与其他方法(搜索， 策略求解) 的比较

任务规划ASP逻辑程序分析

- Generate, Test & Define
- Generate: 生成所有潜在可能的解(choice rules)
- Test: 用约束消除不符合的解(eliminate “bad” solutions)
- Define: 定义在约束中出现的辅助的谓词(auxillary predicates)

任务规划ASP逻辑程序分析

- Generate: $1\{\text{occurs}(A, t) : \text{action_of}(A)\}1.$
 - “:” : condition, 用冒号右边的域谓词(domain predicate)初始化左边的谓词, 比如有 $\text{action_of}(\text{move}(0))$, $\text{action_of}(\text{move}(1))$, $\text{action_of}(\text{catch}(12))$, 那么对于第一步会grounding出 $\{\text{occurs}(\text{move}(0), 1), \text{occurs}(\text{move}(1), 1), \text{occurs}(\text{catch}(12), 1)\}$
- Cardinality Constraints: 在此程序中要求 $\text{occrus}(A,t)$ 在answer set只能出现一次 (对于每一步t)

任务规划ASP逻辑程序分析

- `action_of(move(X)) :- loc(X).`
- `action_of(catch(A)) :- obj(A), small(A).`
- `action_of(putdown(A)) :- obj(A), small(A).`
- `action_of(toplate(A)) :- obj(A), small(A).`
- `action_of(fromplate(A)) :- obj(A), small(A).`

- 通过`loc(X)`, `obj(A)`和`small(A)`三个域谓词来推出机器人所有可能的动作，这三个域谓词的初始化是由规划的环境描述文件完成的

任务规划ASP逻辑程序分析

- Test:
 - :- not goal(give(B, A), t), give(B, A), obj(B), obj(A), small(A).
 - :- not goal(puton(A, B), t), puton(A, B), obj(B), obj(A), small(A).
 - :- not goal(goto(B), t), goto(B), obj(B).
 - :- not goal(putdown(A), t), putdown(A), obj(A), small(A).
 - :- not goal(catch(A), t), catch(A), obj(A), small(A).
- constraints for goals: 任务规划程序中用not和constraints的方式来满足目标

任务规划ASP逻辑程序分析

- Define:

- $\text{goal}(\text{give}(B, A), t) :- h(\text{sameplace}(A, B), t), h(\text{outagent}(A), t), \text{give}(B, A), \text{obj}(B), \text{obj}(A).$
- $\text{goal}(\text{puton}(A, B), t) :- h(\text{sameplace}(A, B), t), h(\text{outagent}(A), t), \text{puton}(A, B), \text{obj}(B), \text{obj}(A).$
- $\text{goal}(\text{goto}(B), t) :- h(\text{sameplace}(0, B), t), \text{goto}(B).$
- $\text{goal}(\text{putdown}(A), t) :- h(\text{outagent}(A), t), \text{putdown}(A).$
- $\text{goal}(\text{catch}(A), t) :- \text{not } h(\text{outagent}(A), t), \text{catch}(A).$
- $\text{sameplace}(A, B)$ 表示A和B在同一位置， $\text{outagent}(A)$ 表示A物体不在机器人手上或机器人盘子上

任务规划ASP逻辑程序分析

- $h(\text{sameplace}(A, B), t) :- h(\text{location}(A, X), t), h(\text{location}(B, X), t), \text{obj}(A)), \text{obj}(B), \text{loc}(X).$
- $h(\text{outagent}(A), t) :- \text{not } h(\text{hold}(A), t), \text{not } h(\text{plate}(A), t), \text{obj}(A), \text{small}(A).$
- 通过定义一些辅助的谓词来实现目标
- 要实现这些目标，使初始环境变换到目标环境需要机器人的对环境进行改变，既需要定义机器人原子动作的执行条件和执行效果

任务规划ASP逻辑程序分析

- move(X)
- h(location(0, X), t) :- occurs(move(X), t), loc(X).
- :- occurs(move(X), t), h(location(0, X), t-1), loc(X).
- catch(A)
- h(hold(A), t) :- occurs(catch(A), t), obj(A), small(A).
- :- occurs(catch(A), t), not h(hold(0), t-1), obj(A), small(A).
- :- occurs(catch(A), t), not h(outagent(A), t-1), obj(A), small(A).
- :- occurs(catch(A), t), not h(sameloc(A), t-1), obj(A), small(A).
- putdown(A)
- h(hold(0), t) :- occurs(putdown(A), t), obj(A), small(A).
- :- occurs(putdown(A), t), not h(hold(A), t-1), obj(A), small(A).

任务规划ASP逻辑程序分析

- `toplate(A)`
- `h(plate(A), t) :- occurs(toplate(A), t), obj(A), small(A).`
- `h(hold(0), t) :- occurs(toplate(A), t), obj(A), small(A).`
- `:- occurs(toplate(A), t), not h(hold(A), t-1), obj(A), small(A).`
- `:- occurs(toplate(A), t), not h(plate(0), t-1), obj(A), small(A).`
- `fromplate(A)`
- `h(hold(A), t) :- occurs(fromplate(A), t), obj(A), small(A).`
- `h(plate(0), t) :- occurs(fromplate(A), t), obj(A), small(A).`
- `:- occurs(fromplate(A), t), not h(plate(A), t-1), obj(A), small(A).`
- `:- occurs(fromplate(A), t), not h(hold(0), t-1), obj(A), small(A).`

任务规划ASP逻辑程序分析

- 增加因果推理知识，客观世界中的常识推理
- static casual law
- $h(\text{location}(A, X), t) :- h(\text{hold}(A), t), h(\text{location}(0, X), t), \text{obj}(A), \text{small}(A), \text{loc}(X).$
- $h(\text{location}(A, X), t) :- h(\text{plate}(A), t), h(\text{location}(0, X), t), \text{obj}(A), \text{small}(A), \text{loc}(X).$

任务规划ASP逻辑程序分析

- 对于机器人没有改变状态的物体，需要说明它们在下一步的状态依然保持当前的状态
- 框架问题： Everything is presumed to remain in the state in which it is (Leibniz, “An Introduction to a Secret Encyclopædia”, c. 1679). 也称惯性定律
- $\text{location}(A, X)$
- $\text{h}(\text{location}(A, X), t) :- \text{h}(\text{location}(A, X), t-1), \text{not } \text{h}(\text{change}(A), t), \text{obj}(A), \text{loc}(X).$
- $\text{h}(\text{location}(0, X), t) :- \text{h}(\text{location}(0, X), t-1), \text{not } \text{h}(\text{change}(0), t), \text{loc}(X).$
- $\text{h}(\text{change}(A), t) :- \text{occurs}(\text{move}(X), t), \text{h}(\text{hold}(A), t), \text{obj}(A), \text{small}(A), \text{loc}(X).$
- $\text{h}(\text{change}(A), t) :- \text{occurs}(\text{move}(X), t), \text{h}(\text{plate}(A), t), \text{obj}(A), \text{small}(A), \text{loc}(X).$
- $\text{h}(\text{change}(0), t) :- \text{occurs}(\text{move}(X), t), \text{loc}(X).$

任务规划ASP逻辑程序分析

- $\text{hold}(A)$
- $\text{h}(\text{hold}(A), t) :- \text{h}(\text{hold}(A), t-1), \text{not occurs}(\text{putdown}(A), t), \text{obj}(A), \text{small}(A).$
- $\text{h}(\text{hold}(A), t) :- \text{h}(\text{hold}(A), t-1), \text{not occurs}(\text{toplate}(A), t), \text{obj}(A), \text{small}(A).$
- $\text{h}(\text{hold}(0), t) :- \text{h}(\text{hold}(0), t-1), \text{not occurs}(\text{catch}(A), t), \text{obj}(A), \text{small}(A).$
- $\text{h}(\text{hold}(0), t) :- \text{h}(\text{hold}(0), t-1), \text{not occurs}(\text{fromplate}(A), t), \text{obj}(A), \text{small}(A).$

- $\text{plate}(A)$
- $\text{h}(\text{plate}(A), t) :- \text{h}(\text{plate}(A), t-1), \text{not occurs}(\text{fromplate}(A), t), \text{obj}(A), \text{small}(A).$
- $\text{h}(\text{plate}(0), t) :- \text{h}(\text{plate}(0), t-1), \text{not occurs}(\text{toplate}(A), t), \text{obj}(A), \text{small}(A).$

任务规划ASP逻辑程序分析

- 对于任务中含有约束的情况

- %cons(task)
- :- cons(give(B, A)), goal(give(B, A), t).
- :- cons(puton(A, B)), goal(puton(A, B), t).
- :- cons(goto(B)), goal(goto(B), t).
- :- cons(putdown(A)), goal(putdown(A), t).
- :- cons(catch(A)), goal(catch(A), t).
- %cons(info)
- :- cons(on(A, B)), info(on(A, B), t).
- :- cons(near(A, B)), info(near(A, B), t).
- :- cons(onplate(A)), info(onplate(A), t).
- %ncons(info)
- ok(on, t):- ncons(on(A, B)), info(on(A, B), t).
- ok(near, t):- ncons(near(A, B)), info(near(A, B), t).
- ok(onplate, t):- ncons(onplate(A)), info(onplate(A), t).
- :- ncons(on(A, B)), not ok(on, t).
- :- ncons(near(A, B)), not ok(near, t).
- :- ncons(onplate(A)), not ok(onplate, t).

任务规划ASP逻辑程序分析

- 用ASP求解任务规划问题可以得到最优解，但是效率是目前最大的问题
- ASP求解在grounding阶段耗费的时间最多
- 提高效率的办法
 - 优化程序代码，减少grounding中生成的规则
 - 增加宏行动，将两步或两步以上的原子动作看成一个动作
 - 提高ASP求解器的效率

Outline

- Home仿真ASP决策逻辑程序分析
- 与其他方法(搜索， 策略求解) 的比较

搜索算法求解任务规划

- 强剪枝的深度优先算法
- 程序将机器人的5个原子动作做为基本状态进行盲目搜索，进行每一步搜索之前都要判断是否满足约束，在每一步搜索之后判断是否达到任务目标要求
- 程序在求得一个解后会回溯到前一状态，继续搜索剩下的解。最后程序在求得的所有解中找到一个最优解。
- 问题
 - 搜索过程中满足目标和满足约束的判断上。因为在场景中对于同一类物体会有几个属性不同的物体，所以对于需要满足唯一性的约束和目标会增加编程的复杂度
 - 效率，算法的复杂度是指数级增长的，要求得最优解必须搜索完所有状态空间。

策略算法求解任务规划

- 对于任务中的每个目标有相应的求解策略，硬编程方式
- 求解效率高
- 问题：
 - 无法对于复杂任务解出最优解。对于相关的目标，必须考虑到所有的组合才能输出最优解
 - 对于任务中含有约束的情况较难处理

三种求解方式的比较

Stage1	不相关目标得分(14)	相关目标得分 (26)	总分
ASP	261	460	721
搜索	242	343	585
策略求解	274	410	684

Stage2	不含约束得分 (9)	包含约束得分 (31)	总分
ASP	133	700	833
搜索	112	552	664
策略求解	120	625	745