

# CLMASP(Coupling LLMs with ASP)

- ASP: 一种专门用来做逻辑推理和规划的编程语言，提供规则与现状
- TLPLAN system: 提供特定领域控制知识，加速求解
- Grounding (Process) / 实例化过程：缩小定义域（减少X的选项）
- Loops and Loop Formulas / 环与环公式：数学工具来检测和验证的“缩减”操作

## 主要工作

定义世界规则（含优化） + 定义目标 + (可选的 LLM 提示) -> 扔进求解器 -> 求解器算出步骤

添加LLM生成的骨架、求解器求解的优化

## CLMASP核心技术

- TLPLAN思想，将LLM生成的粗略规划作为提示 or 知识
- 空间压缩：利用粗略方案来缩小变量的取值范围（定义域）加速实例化
- 使用环公式来验证剪枝的可靠性
- 剪枝：admissible and safe reductions（“容许缩减”和“安全缩减”）

## ASP语法

谓词逻辑（人智导论上过的）

## ASP运行

### Grounding

likes(X, ice\_cream) 在计算机求解时，变量X替换成世界上所有可能的物体

三个人 {john, mary, bob}

规则: likes(X, ice\_cream) :- child(X)

Grounding之后生成三条规则

- likes(john, ice\_cream) :- child(john).
- likes(mary, ice\_cream) :- child(mary).
- likes(bob, ice\_cream) :- child(bob).

## Answer Set (回答集)

程序 Grounding 之后，求解器（Solver）会去寻找一种状态，让所有规则都成立。这个满足条件的状态集合，就叫 Answer Set。

## 机器人规划的专用语法

1. 时间步：ASP 里所有的动作和状态都要带上时间标签  $t$
2. 流畅性(fluent):状态
3. 惯性公理  $h(F, T) :- h(F, T-1), \text{not } -h(F, T).$ 
  - 如果状态  $F$  在上一步 ( $T-1$ ) 是真的，并且没有证据表明它在这一步 ( $T$ ) 变成了假的，那它就保持原样。

## 具体实现

## Skeleton plans(骨架计划)

1. LLM生成粗略动作列表
  - prompt:"洗衣服"
  - output: ["find(shirt)", "wash(shirt)"]
2. 文本映射为ASP逻辑对象
  - shirt->object(2)
3. subtask

```
goal(subtask(p, find(2), 1), t) :- occurs(find(2), t).
goal(subtask(p, wash(2), 2), t) :- occurs(wash(2), t).
goal(p, t) :-
    goal(subtask(p, find(2), 1), T1),
    goal(subtask(p, wash(2), 2), T2),
    T1 <= T2, % 关键: 仅约束相对顺序, 不固定绝对时间
    T2 <= t.
:- not goal(p, t), query(t). % 强制要求最终达成链条 p
```

## 实例化中缩小变量定义域

deciding whether  $D$  is an admissible reduction or a safe reduction of  $\Pi$  is coNP-hard.

在规划（Planning）任务中，我们并不需要找出所有解（并不要求变换后的解集与原来的解集完全相同），只要找到一个能用的解就行（原解集的子集）。

简单来说，验证一个简化是否“容许 (Admissible)”非常难，但验证它是否“Loop-admissible”相对容易。

Theorem 3. Given a subset D of  $HU(\Pi)$  for an ASP program  $\Pi$ , if D is a loop-admissible reduction of  $\Pi$ , then D is an admissible reduction of  $\Pi$ .

在 LLM-Assisted Planning 的背景下，这个定理是算法设计的基石：

- 场景：LLM 输出了一个骨架计划，我们根据这个骨架去删减 ASP 的搜索空间（这就是 Reduction）。
- 担忧：我们怎么确信 LLM 引导的删减不会把所有正确答案都删没了？
- 保证：算法设计者可以利用 Theorem 3 设计一套规则：只要 LLM 的骨架映射过程遵循了特定的结构约束（满足 Loop-admissible），我们就可以100% 确信

## 全流程

### 1. LLM生成骨架

- LLM输出物体类别，一个粗糙的计划 $\tau_s^0$
- 语义接地与修正：环境里只有 kitchen\_table，但 LLM 可能会输出 kitchen\_counter 通过计算词向量的相似度来修正 (SGR)

### 2. APS增强与细化

- 输入三要素：上一阶段的骨架（可选）、物理法则、环境地图
- 更新物理法则。