

Introduction & Motivation :

1. 针对一个复杂的问题，尤其是np-hard乃至pspace-complete的问题，完全依赖asp算出解非常消耗资源，“Answer set planning: a survey.” 这主要囿于grounding bottleneck, i.e., grounding before solving --> 体现在 related work (已有)
2. 通常，我们会有一些domain-dependent prior knowledge serving as some sort of heuristics。直觉上来讲，似乎这些方法起效的途径是减小每个variable的effective grounding domain。但是，这一empirical method至少会有一下两个underexplored的问题 i) encoding heuristic predicates/rules 本身也会带来 encoding&grounding overhead；ii) 数量变化后的ground atom和ground rules是否一定会加快求解（衍生的问题是：会离phase transition更远还是更近）--> 符合直觉，最好是有一些 related work 来说明这这种假设是合理的（有一些工作也基于这种假设）Zhu：有没有必要去cover phase transition 的实验？
3. 这种启发式方法的效果往往可以理解为 Herbrand universe 的缩减 --> 直觉上需要缩减 search space，映射到 HU 上的缩减？给一个小例子，比如说在寻找 最小-图着色的时候可以不用考虑那些“链”
4. ~~现有LLM的发展，使得我们可以有一种 2-stage 的框架：--> 不知道是否有必要去加实验~~
 1. ~~Logic based system 在资源（例如时效性）允许的情况下尽可能算出为真的“知识”~~
 2. ~~再由LLM补全（对于LLM而言，为真的输入的知识越多，其贴近真实解的可能性越大）~~

Introduction & Motivation :

5. 我们定义了 diminution, 我们不要求求出所有的解 (解的数量), 也不要求每个解的完整性... --> 体现在 definition 以及 property
6. 但是由于找到这样的有效的 diminution 的复杂性过高, 于是有了一种针对 Herbrand Universe 的 Guess&Test 的思路 (可以 guess 一个 constant set 作为 D, 然后再 check 其是否满足条件) 。 --> 体现在 definition 以及 property
7. 为了该方法的可迁移性, 我们给出了一种现代 grounder 通用的方法 (domain-predicate) 等价的实现了 diminution。 --> 补充证明, 从 saturation 技术出发
8. To this end, 我们定义了 a formal notion xxx to study the properties and characterize the inherent complexities, from the theory side。 In practice, making use of the existing grounders (in particular, their exploitation of domain predicates), we implement several heuristics with the aforementioned effects

实验 & 问题

现有的 pipelines 以及支持统计的数据（支持“非incmode”和“incmode”单个和批量求解以及 csv 格式数据收集，即代码部分已经基本完成，还需要实现 dlv 版，其中 ASSAT 使用 c++ 版本过老已无法编译）

1. 每个step 的 solving 和 grounding 用时
2. Constant sets 大小以及 herbrand size 大小 (herbrand size 是由 clingo 的 symbolic set 中出现的所有 constants 和 function*constants 构成的，不太准确，仅供参考)
3. 每个step Grounding 后生成的 rules 和 atoms 数量（去重后）
4. 为了使用 related 而引入的所有辅助 atom 的数量（作为fact/不作为fact）
5. 使用 aspif（官方的 grounding 中间格式）输出的 grounding 后文件的大小
6. Rss（the amount of memory a process is actively using in RAM）消耗量（难以解释，仅供参考）
7. 以及后面提到的所有 domian 的 instance 的批量生成方式

solving gw/1 (related)																		
[step	0]	G 0.151s		S 0.003s		C 51		H 1661		rules	0 atoms	0		rel 766/0		G size 194065B		mem 26.01/26.66 MB
[step	1]	G 0.230s		S 0.003s		C 52		H 2362		rules	4928 atoms	4928		rel 766/0		G size 393823B		mem 28.07/28.97 MB
[step	2]	G 0.230s		S 0.003s		C 52		H 2362		rules	9757 atoms	7245		rel 766/0		G size 589351B		mem 30.20/30.42 MB
[step	3]	G 0.231s		S 0.003s		C 52		H 2362		rules	14594 atoms	9589		rel 766/0		G size 797149B		mem 31.34/31.58 MB
[step	4]	G 0.231s		S 0.003s		C 52		H 2362		rules	19437 atoms	11965		rel 766/0		G size 1013309B		mem 32.55/32.84 MB
[step	5]	G 0.244s		S 0.003s		C 52		H 2362		rules	24286 atoms	14377		rel 766/0		G size 1225247B		mem 33.71/34.53 MB
[step	6]	G 0.241s		S 0.004s		C 52		H 2362		rules	29143 atoms	16831		rel 766/0		G size 1437182B		mem 35.30/36.31 MB
[step	7]	G 0.241s		S 0.004s		C 52		H 2362		rules	34009 atoms	19333		rel 766/0		G size 1653330B		mem 36.75/38.00 MB
[step	8]	G 0.249s		S 0.003s		C 52		H 2362		rules	38884 atoms	21881		rel 766/0		G size 1865258B		mem 39.19/40.76 MB
[step	9]	G 0.272s		S 0.004s		C 52		H 2362		rules	43766 atoms	24471		rel 766/0		G size 2081399B		mem 41.23/42.54 MB
[step	10]	G 0.261s		S 0.005s		C 52		H 2362		rules	48656 atoms	27109		rel 766/0		G size 2301684B		mem 43.59/44.68 MB
[step	11]	G 0.274s		S 0.004s		C 52		H 2362		rules	53556 atoms	29797		rel 766/0		G size 2517752B		mem 44.82/47.33 MB

实验 & 问题

现有的 domain 以及他们 domain-specific 的问题：

1. 最小颜色 Graph Coloring,

1. related：弃掉里面的链。

2. 问题：loop formula 那篇论文的实验已经说明，coding 出来 graph coloring 不会包含 loop。

3. 这个问题比较常见，根据 loop admissible 的定义(下面记 Π 和 $\Pi|_D$ 分别为 P 和 $P|_D$)，：

1. 给定一个 D ，为了判断其是否是 loop admissible 的，我们只需要检查条件 1，因为 P 中没有 loop L

2. 对于检查条件 1，实际上可以改写为：对于所有 $P|_D$ 的 answer set $I|_D$ ，检查是否存在一个 I' ，使得 $I|_D \cup I'$ 是 P 的一个 supported model (即 $\text{comp}(P)$ 的 model)。

3. 不知道这个时候在 复杂性 (效率) 上是否还有优势？

4. Check：by definition

5. 这里是否需要实现 checker 并记录消耗情况？

1. for every answer set I_D of $\Pi|_D$, there exists an interpretation I' such that $I_D \cup I'$ satisfies rules in $\Pi|_{HU(\Pi)}$ and loop formulas for every loop L' of Π with $L' \subseteq I'$, and

2. there does not exist a loop L of Π such that L is not a loop of $\Pi|_D$ and L contains a loop L' of $\Pi|_D$ with $R^-(L', \Pi|_D) \neq \emptyset$.

实验 & 问题

现有的 domain 以及他们 domain-specific 的问题：

2. Grid world (给定起点终点障碍物然后走格子)
 1. Related：贴障碍物走
 2. 问题：无 loop
3. AWS (多智能体仓储搬运货物)
 1. 扩大原 instance 的 map (用highway)，然后把所有的 robot shelf pickstation 找一个框框住，再给这个框每个边往外扩大两格。
 2. 问题：无loop,
4. Virtual home (需要 check 一下 loop 情况，需要丢掉 skeleton plan 的架构)
5. MPD (给定一个 graph，graph 上节点都是机器，机器可以给货物添加属性，目标：货物从某个机器上下流水线且具备有一些特定的属性)
 1. 拟 related：找到相关节点用广度优先剔除掉不相关路径

上述的这些 Domain 都是没有 loop 的，因此只能 by definition 的去检查

实验 & 问题

现有的 domain 以及他们 domain-specific 的问题：

6. 哈密尔顿回路

1. Related：先按序号构造一个环，然后添加一堆 redundancy 的 edge，related 的时候只需要 $1 \rightarrow 2 \rightarrow 3 \dots \rightarrow 1$ 找到这个环即可

2. 问题：有 loop 但是只支持检查的情况少。

1. for every answer set $I_{\mathcal{D}}$ of $\Pi|_{\mathcal{D}}$, there exists an interpretation I' such that $I_{\mathcal{D}} \cup I'$ satisfies rules in $\Pi|_{HU(\Pi)}$ and loop formulas for every loop L' of Π with $L' \subseteq I'$, and

2. there does not exist a loop L of Π such that L is not a loop of $\Pi|_{\mathcal{D}}$ and L contains a loop L' of $\Pi|_{\mathcal{D}}$ with $R^-(L', \Pi|_{\mathcal{D}}) \neq \emptyset$.

从给定 \mathcal{D} 出发判断是否 loop admissible 出发重述成一个 check algorithm：

1. 给定 \mathcal{D} ，计算 $\Pi|_{\mathcal{D}}$ 。

2. 对于所有的 $\Pi|_{\mathcal{D}}$ 。

1. 判断是否能基于 $\Pi|_{\mathcal{D}}$ 扩展成 P 的 classical model（计算 $\{\Pi|_{\mathcal{D}}\} \cup P$ 的 model）

2. 对于所有的 I' 中的 loop L' 判断是否符合其在 P 中的 loop formula。

3. 对于 P 中的所有非 singleton loop L ，找到 $L \cap \text{atom}(P|_{\mathcal{D}})$ 构成的 $P|_{\mathcal{D}}$ 中的 loop L' （包括 singleton），判断其 R^- 是否为空。

仔细想了一下应该是能找到 case 做 case study 的。