CSE 579: Knowledge Representation & Reasoning

Module 4: Practice of Answer Set Programming

Ali Altunkaya
Spring 2023

Outline

1. Choice Rules
2. Constraints
3. Methodology of ASP
4. Representing Functions in ASP
5. Solving Problems in ASP using Clingo
    1. Combinatorial Search
    2. Graph Problems
    3. Sudoku Puzzle and Elaboration Tolerance
6. Aggregates and Optimization
    1. Solving optimization problems

1. Choice Rule

Choice Rule:

- We abbreviate the formula $(p_1 \vee \neg p_1) \wedge (p_2 \vee \neg p_2) \wedge$ ….. $(p_n \vee \neg p_n)$ as $\{p_1; p_2; \ldots p_n\}$ and call it as **choice rule**.
- The special thing about this formula is, there are 2^n possible stable models, which assume all possible interpretations are stable models.
- So we can write succinct Clingo programs.
- Choice rules with Intervals and Pooling
  - { p(1..3) }
  - { p(1); p(2); p(3) }
- Choice rules with Cardinality Bounds, puts a lower bound and upper bound on the number of elements in a stable model.
  - 1 { p(1..3) } 2 describes the interpretation that consists of
    - At least 1 elements and
    - At most 2 elements
  - { p(1..3) } = 2 describes the subsets of {1,2,3} that consists exactly 2 elements

1. Choice Rule

Choice Rule:

- Choice rules with variables: We use variables to write more complex formula
  - A **local variable** is a variable such that all its occurences in the rule are between {…}. All other variables are **global variables**.
    - { p(i) :- i = 1..3 }   i is a local variable
    - { p(i) } :- i = 1..3   i is a global variable
- When we apply **local variable** to a formula, still we will have one formula.
    - { p(i) :- i = 1..3 }   equals to below:
    - {p(1), p(2), p(3) :- T}
- When we apply **global variable** to a formula, we will have many formulas.
    - { p(i) } :- i = 1..3   equals to following three formulas:
    - {p1} :- T
    - {p2} :- T
    - {p3} :- T

2. Constraints

Constraints:

- A constraint is a rule that has no head.
- Example

  :- p(1)   also equals to below

  $\perp \leftarrow$ p(1)

  which means rule is FALSE.

- Constraints are often used with choice rules to remove “undesirable” stable models, for which constraint is “violated”.
- Used to decrease the search space.
- Cardinality bounds in a choice rule can be sometimes replaced by constraints.

3. Methodology of ASP

Methodology of ASP in Clingo:

- A way to organize rules and solve a problem, in declarative programming

  1-) Generate part: Generates a “search space”, a set of potential solutions. (using choice rules)

  2-) Define part: If needed, defines new atoms in terms of other atoms. (this step is optional, define new atoms if needed)

  3-) Test part: Remove (weed out) the elements of the search space that do not represent solutions. (using constraints)

- N-Queens puzzle and Schur numbers can be solved just using the Generate and Test part.
  - Without using Define part, without defining new atoms or functions.

4. Representing Functions in ASP

A **function** from set A to set B is a relation f() such that for every element x in A, there is exactly one element y in B. But there is no other restriction.

- A = {1, 2, 3}   B = {a, b}
- How many functions from A to B? 2 * 2 * 2 = 8

Three types of functions:

- One-to-One functions: Mapping from set A to set B, but each element in A should map to different elements in B.
  - A = {1, 2, 3} B={a,b,c,d,e}
  - How many 1-1 functions from A to B? 5 * 4 * 3 =60
- Onto functions: Mapping from set A to set B, but each element in B must be mapped.
  - A = {1, 2, 3}   B = {a, b}
  - How many onto functions from A to B? 2^3 - 2 = 6
- One-to-One Correspondence functions: It should be both 1-1 and Onto.
  - A = {1, 2, 3}   B = {a, b, c}
  - How many 1-1 correspondence functions from A to B? 3 * 2 * 1 = 6

5. Solving Problems in ASP

Combinatorial Search Problems:

- Seating Arrangement
- Logic Puzzle

Graph Problems:

- Graph Coloring problem
- Finding Maximum Clique in a Graph
- Vertex Cover
- Hamiltonian Cycle (aka Travelling Salesman Problem)
- Sudoku Puzzle and its variations (using Elaboration Tolerance)

6. Aggregates and Optimization

An **aggregate** is a function that can be applied to sets, such as:

- #count
- #sum
- #minimize
- #maximize

  Since it is set, ignore the duplicate elements!!!

We can use these aggregates (clingo directives) for **optimization** problems.

- Usually optimization problems have many solutions, where each solution associated with a score, but we want to find the optimal (min or max) solution.
- Clingo finds possible solutions (answer sets) with associated scores, then we can find the best/optimal one (using min or max).
- Ex: Max Clique in a graph
- Ex: Hamiltonian Cycle with minimum distance
- Ex: Assigning Papers to Referees

Conclusions

Probably you have noticed that most of these problems that we have solved in Clingo are NP-complete or NP-hard.

- N-Queens puzzle
- Max Clique in a graph
- Vertex Cover
- Graph Coloring
- Hamiltonian Cycle or (Traveling Salesman Problem)
- Etc.

We solved them in Clingo using a Declarative Programming approach. A different perspective.

Be aware that, Clingo or ASP is not magic. These problems are still NP-complete or NP-hard, which means we can find the solution when problem size = n is small, but we cannot find the solution in a feasible time when n is big (even with Clingo).

Thanks

&

Questions