ASP Modelling 2

Víctor Gutiérrez-Basulto

List of Tasks

Please complete past exercises

If you did not finish past labs, please work on that first.

1 Exercises from Lifschitz' Book

Solve the following exercises from [1]

- Exercise 7.1

  ANSWER: `:- #count{I : I = 1..n, in(I)} != m.`

  Note that the original choice rule was selecting m `in`-elements.

  `{in(1..n)}` chooses subsets of `in`-elements, but it does not specify how many. So we need to ensure that only m are chosen, which is done by the above constraint, which says that the number of `I`s in `in` cannot be different from m.

- Exercise 7.2

  ANSWER: `howmany(I,N) :- N = #count{C : where(C,I)}, I = 1..k.`

  For an arbitrary floor `I` (from 1 to k), we count the number `N` of classes `C` such that `C` is taught on floor `I`, captured by the relation `where(C,I)`.

- Exercise 7.5 (HINT: Use `#count`)

  ANSWER: `#count{X : p(X)}.`

  In the rule `#sum{1,X:p(X)}.` by summing by one for each element in `p(X)` we are counting the elements in `p(X)`, which is what the rule above is doing.

- Exercise 7.9

  ANSWER:
  ```
  howmany(I,N) :- N = #count{C : where(C,I)},
                  I = 1..K, K = #max{J : where(C,J)}.
  ```

We get the value of `K` by the taking the maximum value `J` for which is known that a class is taught on that floor.

- Exercise 7.10 (HINT: You need to add another rule)

  ANSWER: `:- #sum{Vol,I : in(I), volume(I,Vol)} > maxvolume.`

  We ensure that the total volume, that is, the sum of the volume of all items `I` in `in`, does not exceed the `maxvolume`. Note that different items might have the same volume, that is why we have `Vol, I` on the lhs of `:`.

References

[1] Vladimir Lifschitz. *Answer Set Programming*. Springer, 2019.