COMP 4418 Assignment 1

z5270589
YiJie Zhao

[1.1] Consider the spanning spider from Figure 2. Indicate the center vertex and the list of edges that is included in the spanning spider.

- Referring to the spanning spider from Figure 2, we can observe that vertex d has a degree 4, and d is the only vertex in this spanning spider whose degree is 3 or 3 more. According to the definition, we know that only **vertex d** is the center of the spider and known as the center vertex.
- List of edges that is included in the spanning spider:
- **edge(d, c), edge(d, a), edge(d, b), edge(y, b), edge(d, e), edge(e, x), edge(x, z)**

[1.2] Provide ASP rules to define the center/1 predicate and ensure that in any model, exactly one vertex is selected as the center.

ASP rules:

1{centetr(V) : vertex(V)}1.

[1.3] Provide a generator for the leg/2 predicate.

Generator for the leg/2 predicate:

leg(X, Y) :- center(Y), edge(X, Y).
leg(X, Y) :- center(Y), edge(Y, X).

[2.1] One property required of spanning spiders is that every vertex should be reachable from the center through leg edges. Introduce a derived predicate reachable/1 that ranges over vertex names and is true when the corresponding vertex is reachable from the center through leg edges. Use this predicate to define a constraint that ensures every vertex of the original graph is reachable from the center through leg edges.

To check whether a certain vertex is reachable from the center, firstly we know that if the vertex has a leg with the center, this vertex should be reachable:

reachable(R) :- center(C), edge(C, R), edge(R, C), vertex(C).

Then, the vertex is also reachable if it has a edge between another reachable vertex(not a center vertex):

reachable(R) :- reachable(R2), not center(R2), edge(R, R2), edge(R2, R).

Then to define a constraint that ensures every vertex of the original graph is reachable from the center through leg edges, we accomplish this by listing total vertex number and the number of reachable vertex in the graph:

TotalVertexNum(N) :- N = #count {V : not center(X), vertex(V)}.

NumOfReachable(V) :- V = #count{R : reachable(R)}.

After we know the total vertex number and the number of reachable vertex in the graph, then we compare them and define a constraint that ensures every vertex of the original graph is reachable from the center through leg edges:

:- TotalVertexNum(N), NumOfReachable(V), N != V.

This line of code just the initial methods, the real code has some difference with these codes, for more details please refer to the **spiders.lp**.

[2.2] The reachability of every vertex from the center is a required property of spanning spiders, but it is not sufficient and we need to ensure other constraints are satisfied. Describe all the other constraints that need to be satisfied and write ASP rules to enforce these constraints.

The constraints that should be satisfied:

- If it is a spanning spider, it could only have (TotalVertexNum - 1) edges.
- If it is a spanning spider, no more than one vertex can have three or three more degrees.
- If it is a spanning spider, and there is one vertex which has three or three more degrees, it must be chosen as the center.

These are three constraints that must be satisfied to get a spanning spider, and for more details of the ASP rules, please refer to the **spiders.lp**.

[3.1] Based on your answers to the above questions, write an ASP program spiders.lp that takes an input graph and outputs all the distinct spanning spiders of the graph. How many distinct spanning spiders does the graph in Figure 1 have?

There are **74** distinct spanning spiders in Figure 1.
Note: some of the spanning spiders have exactly the same legs but only different in centers.
The actual ASP program please refer to spiders.lp.

[3.2] Write an ASP program spidershortlegs.lp that outputs a spanning spider with the shortest longest leg.

The actual ASP program please refer to spidershortlegs.lp.