Knowledge Representation and Reasoning

Coursework 2

Daolin Sheng ml192ds 201305883

Part A: Sequent Calculus

Task (i).

$H \wedge Q,\ G \ \Rightarrow\ (\neg H \rightarrow P) \wedge (G \vee R)$

My answer:

$$\cfrac{\cfrac{\cfrac{\cfrac{Axiom}{H,Q,G \Rightarrow H,P}}{H,Q,G \Rightarrow H \vee P}[\Rightarrow \vee]}{H,Q,G \Rightarrow (\neg H \rightarrow P)}[\Rightarrow\rightarrow r.w.] \quad and \quad \cfrac{\cfrac{Axiom}{H,Q,G \Rightarrow G,R}}{H,Q,G \Rightarrow G \vee R}[\Rightarrow \vee]}{\cfrac{H,Q,G \Rightarrow (\neg H \rightarrow P)\wedge(G\vee R)}{H\wedge Q,G \Rightarrow (\neg H\rightarrow P)\wedge(G\vee R)}[\wedge \Rightarrow]}[\Rightarrow \wedge]$$

Task (ii).

$\forall x[\neg P(a,x)] \ \Rightarrow\ \forall x[\exists y[\neg P(y,x)]]$

My answer:

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{Axiom}{\forall x[\neg p(a,x)], \forall y[p(y,k)], p(a,k) \Rightarrow p(a,k)}[\forall \Rightarrow]}{\forall x[\neg p(a,x)], \forall y[p(y,k)] \Rightarrow p(a,k)}[\neg \Rightarrow]}{\forall x[\neg p(a,x)], \neg p(a,k), \forall y[p(y,k)] \Rightarrow}[\Rightarrow \neg]}{\forall x[\neg p(a,x)], \neg p(a,k) \Rightarrow \neg\forall y[p(y,k)]}[\Rightarrow \exists\ r.w.]}{\forall x[\neg p(a,x)], \neg p(a,k) \Rightarrow \exists y[\neg p(y,k)]}[\Rightarrow \forall]^{\dagger}}{\forall x[\neg P(a,x)], \neg p(a,k) \Rightarrow \forall x[\exists y[\neg P(y,x)]]}}{\forall x[\neg P(a,x)] \Rightarrow \forall x[\exists y[\neg P(y,x)]]}[\forall \Rightarrow]$$

Part B: Axioms and Models

Task (i).

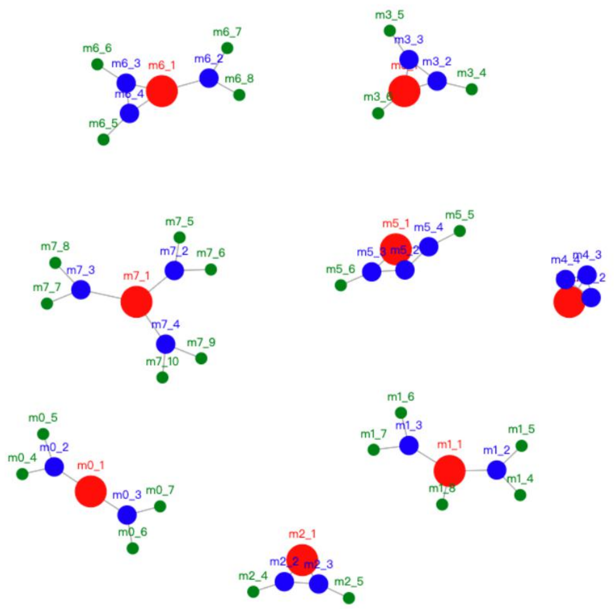
My Answers:

1. If there is a road between A locality and B locality, then there is also a road between B locality and A locality, and A and B are different localities.
2. If Any A locality is larger than any B locality, then only when A is a city and B is a town or village or A is a town and B is a village.
3. Every locality can only is a city or town or village.
4. There is at least one city in this area.
5. There is at most one city in this area.
6. This area has at least two towns.
7. Every locality except cities has a road to a larger locality.
8. Every village can be connected to at most one localitiy by one road.
9. Every town can be connected to at least three different localities.
10. Every locality can be connected to at most three different localities.

Task (ii).

My Answers:

Submit the all models in Python file ***roadmap_models.py*** on Gradescope, Draw out the graphical representation of models following:



Part C: Build Your Own Knowledge Base

Initial Experimentation

1. rule(hate, [[X, is, cat], [Y, is, dog], -[Y, is, puppy]] ==> [X, hates, Y])
2. rule(family, [[X, is_father_of, Y], [Y, is_father_of,Z]] ==> [X, is_grandfather_of,Z]).
3. rule(genesis, [[X, is, dog]] ==> [fatherof(X), is, dog]).
   (a). The purpose of this rule is that if X is a dog, then X's father is a dog.
   (b). This rule does not works because 'fatherof()' is not exits.
4. When we use the weak operator +, it will regard the uncertain conditions as True.

Task (i).

My Answers:

1. The code have submitted as separate file ***ml192ds_kb.pl*** on Gradescope.
2. The code also can see via the following url address:

   https://swish.swi-prolog.org/p/ml192ds_kb.pl

Task (ii).

My Answers:

(1). **Describe the scope of the domain, the power and limitations of the inference**

My knowledge base is base on six students who come from different countries and have different ages, gender and are good at different skills. They registered different modules which are knowledge representation and reasoning(krr), machine learning(ml), programming for data science(pds), scientific computation(sc) and advanced software engineering(ase), and these modules taught by four teachers. Besides, every module requires different skills, such as math, logic, writing or coding and also teaches different programming languages that students can be learned. Finally, I suppose the time that different students spend on different modules each week is different and store in a list.

By setting different facts and rules, it can inference they have a common country if they come from the same country. As the same time, if they have the same module and come from the same country, as a result, they are can make a study group. Besides, because every module requires different skills and every student is good at different skills and studying time on every module per week is different, we can inference he/she would get good performance if he/she gives enough time on a module. Then, the teacher who teaches a module will be happy for who has a good performance on this module.

The limitation of this Knowledge Base is that *'likes'* in the rule *'rule(default, [[X, and, Y, have_common_interest], \+(-[X, likes, Y])] ==> [X, likes, Y]).'* can cause ambiguity. By running *'infer(3), describe([likes])'*, for instance, it output *[billy, likes, cola]* and *[billy, likes, logic]*, in this case, the *'likes'* indicate liking a people and skill.

(2)**. The most interesting inferences**

(a) ***"About the Billy?"***, by running *'infer(1), describe([billy])'*, it can inference out that Billy comes from china, 28 years old, and has common country with Alice, and has common skill with Cola, and Billy can make friend with Cola.

?- infer(1), describe([billy]).
[billy, is, student]
[billy, is_age, 28]
[billy, is, male]
[billy, comes_from, china]
[billy, good_at, coding]
[billy, good_at, math]
[billy, good_at, logic]
[billy, has_module, krr]
[billy, has_module, ml]
[billy, has_module, pds]
[billy, has_module, sc]
[billy, pay_time, 7, in, krr]
[billy, pay_time, 16, in, ml]
[billy, pay_time, 8, in, pds]
[billy, pay_time, 15, in, sc]
[toni, is_teacher_of, billy]
[brandon, is_teacher_of, billy]
[billy, and, cola, makes_friend]
... and so on.

(b) ***"Who have the common interests?"***, by running *'infer(2), describe(have_common_interest)'*, it can be inference out Cola likes Billy, and Cola has common interest with Dota.

?- infer(2), describe([have_common_interest]).
[billy, and, cola, have_common_interest]
[billy, and, dota, have_common_interest]
[billy, and, echo, have_common_interest]
[billy, and, harry, have_common_interest]
[cola, and, billy, have_common_interest]
[cola, and, dota, have_common_interest]
[dota, and, billy, have_common_interest]
[dota, and, cola, have_common_interest]
[echo, and, billy, have_common_interest]
[harry, and, billy, have_common_interest]
... and so on.

(c) ***"Who are best student and teacher happy for them?"***, by running *'infer(3), describe([happy_for, is_best_student]).'*, it can be inference out Billy, Dota, Echo and Harry are the best student, then Toni, Brandon and Matteo are happy for each of them.

?- infer(3), describe([happy_for, is_best_student]).
[billy, is_best_student, _]
[dota, is_best_student, _]
[echo, is_best_student, _]
[harry, is_best_student, _]
[toni, happy_for, billy]
[matteo, happy_for, billy]
[brandon, happy_for, dota]
[brandon, happy_for, echo]
[matteo, happy_for, echo]
[matteo, happy_for, harry]