

NANYANG TECHNOLOGICAL UNIVERSITY

CZ3005 ARTIFICIAL INTELLIGENCE

Assignment 3 Report

Introduction to Prolog

Written by: MOCK JIN WEI

Tutorial Group: TSP1

Matriculation Number: 1721242F

Email: JMOCK001@e.ntu.edu.sg

Exercise 1: The Smart Phone Rivalry

Q1.

The First Order Logic will be:

$\forall X \text{ smartPhoneTech}(X) \Rightarrow \text{business}(X)$

$\forall X \text{ competitor}(X, \text{appy}) \vee \text{competitor}(\text{appy}, X) \Rightarrow \text{rival}(X)$

$\forall X, Y, Z \text{ boss}(X) \wedge \text{stole}(X, Y) \wedge \text{business}(Z) \wedge \text{develop}(Y, Z) \wedge \text{rival}(Y) \Rightarrow \text{unethical}(A)$

Q2.

Prolog Clauses:

```
company(sumsum).
company(appy).
competitor(sumsum, appy).
smartPhoneTech(galacticaS3).
develop(galacticaS3, sumsum).
boss(stevey).
stole(stevey, galacticaS3, sumsum).

business(X) :- smartPhoneTech(X).
rival(X) :- competitor(X, appy).
unethical(X) :- boss(X), smartPhoneTech(Y), rival(Z), company(Z), stole(X, Y, Z).
```

Q3.

Tracing proof that stevey is unethical:

```
[trace] ?- unethical(stevey).
Call: (10) unethical(stevey) ? Unknown option (h for help)
Call: (10) unethical(stevey) ? creep
Call: (11) boss(stevey) ? creep
Exit: (11) boss(stevey) ? creep
Call: (11) smartPhoneTech(_11212) ? creep
Exit: (11) smartPhoneTech(galacticaS3) ? creep
Call: (11) rival(_11300) ? creep
Call: (12) competitor(_11344, appy) ? creep
Exit: (12) competitor(sumsum, appy) ? creep
Exit: (11) rival(sumsum) ? creep
Call: (11) company(sumsum) ? creep
Exit: (11) company(sumsum) ? creep
Call: (11) stole(stevey, galacticaS3, sumsum) ? creep
Exit: (11) stole(stevey, galacticaS3, sumsum) ? creep
Exit: (10) unethical(stevey) ? creep
true.
```

---End of Question 1---

Exercise 2: The Royal Family

Q1.

Relations and Rules in Prolog for old succession:

```

offspring(prince_charles,queen_elizabeth).
offspring(princess_ann,queen_elizabeth).
offspring(prince_andrew,queen_elizabeth).
offspring(prince_edward,queen_elizabeth).

male(prince_charles).
male(prince_andrew).
male(prince_edward).
female(princess_ann).

olderThan(prince_charles,princess_ann).
olderThan(prince_charles,prince_andrew).
olderThan(prince_charles,prince_edward).
olderThan(princess_ann,prince_andrew).
olderThan(princess_ann,prince_edward).
olderThan(prince_andrew,prince_edward).

isOlder(X,Y) :- olderThan(X,Y).
isOlder(X,Y) :- olderThan(X,Z),isOlder(Z,Y).

successionOrder(X,Y) :- male(X),male(Y),isOlder(X,Y).
successionOrder(X,Y) :- male(X),female(Y).

insert(A,[B|C],[B|D]) :- not(successionOrder(A,B)),!,insert(A,C,D).
insert(A,C,[A|C]).

successionSort([A|B],SortList) :- successionSort(B,Tail), insert(A,Tail,SortList).
successionSort([],[]).

successionList(X,SuccessionList) :- findall(Y,offspring(Y,X),ChildNodes),successionSort(ChildNodes,SuccessionList).

```

Tracing Proof of old succession:

```

[trace] ?- successionList(X,SuccessionList).
Correct to: "successionList(X,SuccessionList)"?
Please answer 'y' or 'n'? yes
^ Call: (10) successionList(_13892,_13894) ? creep
Call: (11) findall(_14992,offspring(_14992,_13892),_15054) ? creep
Call: (16) offspring(_14992,_13892) ? creep
Exit: (16) offspring(prince_charles,queen_elizabeth) ? creep
Redo: (16) offspring(_14992,_13892) ? creep
Exit: (16) offspring(princess_ann,queen_elizabeth) ? creep
Redo: (16) offspring(_14992,_13892) ? creep
Exit: (16) offspring(prince_andrew,queen_elizabeth) ? creep
Redo: (16) offspring(_14992,_13892) ? creep
Exit: (16) offspring(prince_edward,queen_elizabeth) ? creep
Exit: (11) findall(_14992,user:offspring(_14992,_13892),[prince_charles,princess_ann,prince_andrew,prince_edward]) ? creep
Call: (11) successionSort([prince_charles,princess_ann,prince_andrew,prince_edward],_13894) ? creep
Call: (12) successionSort([princess_ann,prince_andrew,prince_edward],_15594) ? creep
Call: (13) successionSort([prince_andrew,prince_edward],_15638) ? creep
Call: (14) successionSort([prince_edward],_15682) ? creep
Call: (15) successionSort([],_15726) ? creep
Exit: (15) successionSort([],[]) ? creep
Call: (15) insert(prince_edward,[],_15816) ? creep
Exit: (15) insert(prince_edward,[],[prince_edward]) ? creep
Call: (14) successionSort([prince_edward],[prince_edward]) ? creep
Call: (14) insert(prince_andrew,[prince_edward],_15954) ? creep
Call: (15) not(successionOrder(prince_andrew,prince_edward)) ? creep
Call: (16) successionOrder(prince_andrew,prince_edward) ? creep
Call: (17) male(prince_andrew) ? creep
Exit: (17) male(prince_andrew) ? creep
Call: (17) male(prince_edward) ? creep
Exit: (17) male(prince_edward) ? creep
Call: (17) isOlder(prince_andrew,prince_edward) ? creep
Call: (18) olderThan(prince_andrew,prince_edward) ? creep
Exit: (18) olderThan(prince_andrew,prince_edward) ? creep
Exit: (17) isOlder(prince_andrew,prince_edward) ? creep
Exit: (16) successionOrder(prince_andrew,prince_edward) ? creep
^ Fail: (15) not(user:successionOrder(prince_andrew,prince_edward)) ? creep
Redo: (14) insert(prince_andrew,[prince_edward],_16544) ? creep
Exit: (14) insert(prince_andrew,[prince_edward],[prince_andrew,prince_edward]) ? creep
Exit: (13) successionSort([prince_andrew,prince_edward],[prince_andrew,prince_edward]) ? creep
^ Call: (13) insert(princess_ann,[prince_andrew,prince_edward],_16682) ? creep
Call: (14) not(successionOrder(princess_ann,prince_andrew)) ? creep
Call: (15) successionOrder(princess_ann,prince_andrew) ? creep
Call: (16) male(princess_ann) ? creep
Fail: (16) male(princess_ann) ? creep
Redo: (15) successionOrder(princess_ann,prince_andrew) ? creep
Call: (16) male(princess_ann) ? creep
Fail: (16) male(princess_ann) ? creep
Fail: (15) successionOrder(princess_ann,prince_andrew) ? creep

```

---Continue on the next page---

```

^ Exit: (14) not(user:successionOrder(princess_ann, prince_andrew)) ? creep
Call: (14) insert(princess_ann, [prince_edward], _16672) ? creep
Call: (15) not(successionOrder(princess_ann, prince_edward)) ? creep
Call: (16) successionOrder(princess_ann, prince_edward) ? creep
Call: (17) male(princess_ann) ? creep
Fail: (17) male(princess_ann) ? creep
Redo: (16) successionOrder(princess_ann, prince_edward) ? creep
Call: (17) male(princess_ann) ? creep
Fail: (17) male(princess_ann) ? creep
Fail: (16) successionOrder(princess_ann, prince_edward) ? creep
^ Exit: (15) not(user:successionOrder(princess_ann, prince_edward)) ? creep
Call: (15) insert(princess_ann, [], _17130) ? creep
Exit: (15) insert(princess_ann, [], [princess_ann]) ? creep
Exit: (14) insert(princess_ann, [prince_edward], [prince_edward, princess_ann]) ? creep
Exit: (13) insert(princess_ann, [prince_andrew, prince_edward], [prince_andrew, prince_edward, princess_ann]) ? creep
Exit: (12) successionSort([princess_ann, prince_andrew, prince_edward], [prince_andrew, prince_edward, princess_ann]) ? creep
Call: (12) insert(prince_charles, [prince_andrew, prince_edward, princess_ann], [prince_andrew, prince_edward, princess_ann, _13894]) ? creep
Call: (13) not(successionOrder(prince_charles, prince_andrew)) ? creep
Call: (14) successionOrder(prince_charles, prince_andrew) ? creep
Call: (15) male(prince_charles) ? creep
Exit: (15) male(prince_charles) ? creep
Call: (15) male(prince_andrew) ? creep
Exit: (15) male(prince_andrew) ? creep
Call: (15) isOlder(prince_charles, prince_andrew) ? creep
Call: (16) olderThan(prince_charles, prince_andrew) ? creep
Exit: (16) olderThan(prince_charles, prince_andrew) ? creep
Exit: (15) isOlder(prince_charles, prince_andrew) ? creep
Exit: (14) successionOrder(prince_charles, prince_andrew) ? creep
^ Fail: (13) not(user:successionOrder(prince_charles, prince_andrew)) ? creep
Redo: (12) insert(prince_charles, [prince_andrew, prince_edward, princess_ann], [prince_charles, prince_andrew, prince_edward, princess_ann]) ? creep
Exit: (12) insert(prince_charles, [prince_andrew, prince_edward, princess_ann], [prince_charles, prince_andrew, prince_edward, princess_ann]) ? creep
Exit: (11) successionSort([prince_charles, prince_andrew, prince_edward], [prince_charles, prince_andrew, prince_edward, princess_ann]) ? creep
creep
Exit: (10) successionList(_13892, [prince_charles, prince_andrew, prince_edward, princess_ann]) ? creep
SuccessionList = [prince_charles, prince_andrew, prince_edward, princess_ann].

```

Q2.

Relations and Rules in Prolog for new succession:

```

offspring(prince_charles, queen_elizabeth).
offspring(princess_ann, queen_elizabeth).
offspring(prince_andrew, queen_elizabeth).
offspring(prince_edward, queen_elizabeth).

male(prince_charles).
male(prince_andrew).
male(prince_edward).
female(princess_ann).

olderThan(prince_charles, princess_ann).
olderThan(prince_charles, prince_andrew).
olderThan(prince_charles, prince_edward).
olderThan(princess_ann, prince_andrew).
olderThan(princess_ann, prince_edward).
olderThan(prince_andrew, prince_edward).

isOlder(X, Y) :- olderThan(X, Y).
isOlder(X, Y) :- olderThan(X, Z), isOlder(Z, Y).

insert(A, [B|C], [B|D]) :- not(isOlder(A, B)), !, insert(A, C, D).
insert(A, C, [A|C]).

successionSort([A|B], SortList) :- successionSort(B, Tail), insert(A, Tail, SortList).
successionSort([], []).

successionList(X, SuccessionList) :- findall(Y, offspring(Y, X), ChildNodes), successionSort(ChildNodes, SuccessionList).

```

The red box that is shown in the image above highlights the modification made to the Prolog clause to reflect the new succession rule. In the previous old succession rule, 2 “successionOrder” rules is declared in the Prolog and these will help to generate the list of succession line according to the gender and the order of birth of the princes and princess.

For this new succession rule, since the gender factor is no longer a rule in the line of succession, the “successionOrder” rule is removed. Instead, the Prolog will make use of the “olderThan” rule, which order the princes and princess according to their birth order, to generate the list of succession line.

---Continue on the next page---

Tracing Proof of new succession:

```

[trace] ?- successionList(X, SuccessionList).
Correct to: "successionList(X, SuccessionList)"
Please answer 'y' or 'n' ? yes
^ Call: (10) successionList(_24504, _24506) ? creep
^ Call: (11) findall(_25606, offspring(_25606, _24504), _25668) ? creep
^ Call: (16) offspring(_25606, _24504) ? creep
^ Exit: (16) offspring(prince_charles, queen_elizabeth) ? creep
^ Redo: (16) offspring(_25606, _24504) ? creep
^ Exit: (16) offspring(princess_ann, queen_elizabeth) ? creep
^ Redo: (16) offspring(_25606, _24504) ? creep
^ Exit: (16) offspring(prince_andrew, queen_elizabeth) ? creep
^ Redo: (16) offspring(_25606, _24504) ? creep
^ Exit: (16) offspring(prince_edward, queen_elizabeth) ? creep
^ Call: (11) findall(_25606, user:offspring(_25606, _24504), [prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
^ Call: (11) successionSort([prince_charles, princess_ann, prince_andrew, prince_edward], _24506) ? creep
^ Call: (12) successionSort([princess_ann, prince_andrew, prince_edward], _26208) ? creep
^ Call: (13) successionSort([prince_andrew, prince_edward], _26252) ? creep
^ Call: (14) successionSort([prince_edward], _26296) ? creep
^ Call: (15) successionSort([], _26340) ? creep
^ Exit: (15) successionSort([], []) ? creep
^ Call: (15) insert(prince_edward, [], _26430) ? creep
^ Exit: (15) insert(prince_edward, [], [prince_edward]) ? creep
^ Exit: (14) successionSort([prince_edward], [prince_edward]) ? creep
^ Call: (14) insert(prince_andrew, [prince_edward], _26568) ? creep
^ Call: (15) not(isOlder(prince_andrew, prince_edward)) ? creep
^ Call: (16) isOlder(prince_andrew, prince_edward) ? creep
^ Call: (17) olderThan(prince_andrew, prince_edward) ? creep
^ Exit: (17) olderThan(prince_andrew, prince_edward) ? creep
^ Exit: (16) isOlder(prince_andrew, prince_edward) ? creep
^ Call: (15) not(user:isOlder(prince_andrew, prince_edward)) ? creep
^ Redo: (15) insert(prince_andrew, [prince_edward], _26894) ? creep
^ Exit: (14) insert(prince_andrew, [prince_edward], [prince_andrew, prince_edward]) ? creep
^ Exit: (13) successionSort([prince_andrew, prince_edward], [prince_andrew, prince_edward]) ? creep
^ Call: (13) insert(princess_ann, [prince_andrew, prince_edward], _27032) ? creep
^ Call: (14) not(isOlder(princess_ann, prince_andrew)) ? creep
^ Call: (15) isOlder(princess_ann, prince_andrew) ? creep
^ Call: (16) olderThan(princess_ann, prince_andrew) ? creep
^ Exit: (16) olderThan(princess_ann, prince_andrew) ? creep
^ Exit: (15) isOlder(princess_ann, prince_andrew) ? creep
^ Fail: (14) not(user:isOlder(princess_ann, prince_andrew)) ? creep

^ Redo: (13) insert(princess_ann, [prince_andrew, prince_edward], _27358) ? creep
^ Exit: (13) insert(princess_ann, [prince_andrew, prince_edward], [princess_ann, prince_andrew, prince_edward]) ? creep
^ Exit: (12) successionSort([princess_ann, prince_andrew, prince_edward], [princess_ann, prince_andrew, prince_edward]) ? creep
^ Call: (12) insert(prince_charles, [princess_ann, prince_andrew, prince_edward], _24506) ? creep
^ Call: (13) not(isOlder(prince_charles, princess_ann)) ? creep
^ Call: (14) isOlder(prince_charles, princess_ann) ? creep
^ Call: (15) olderThan(prince_charles, princess_ann) ? creep
^ Exit: (15) olderThan(prince_charles, princess_ann) ? creep
^ Exit: (14) isOlder(prince_charles, princess_ann) ? creep
^ Fail: (13) not(user:isOlder(prince_charles, princess_ann)) ? creep
^ Redo: (12) insert(prince_charles, [princess_ann, prince_andrew, prince_edward], _24506) ? creep
^ Exit: (12) insert(prince_charles, [princess_ann, prince_andrew, prince_edward], [prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
^ Exit: (11) successionSort([prince_charles, princess_ann, prince_andrew, prince_edward], [prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
creep
^ Exit: (10) successionList(_24504, [prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
SuccessionList = [prince_charles, princess_ann, prince_andrew, prince_edward].

```

---End of Question 2---

--End of Assignment 3 Report--