NANYANG TECHNOLOGICAL UNIVERSITY

SINGAPORE

CZ3005 Artificial Intelligence

Lab Assignment 2: Introduction to Prolog

Name: Neo Rui Xuan Berlynn

Matriculation Number: U1922712C

Group: TS3

Exercise 1: The Smart Phone Rivalry

sumsum, a competitor of appy, developed some nice smart phone technology called galacticas3, all of which was stolen by stevey, who is a boss. It is unethical for a boss to steal business from rival companies. A competitor of appy is a rival. Smart phone technology is business.

1. Translate the natural language statements above describing the dealing within the Smart Phone industry in to First Order Logic (FOL).

In simple sentences:
sumsum is a company.
appy is a company.
sumsum is a competitor of appy, so sumsum is also a rival of appy.
smart phone technology is business.
sumsum developed smart phone technology, galacticas3.
stevey is a boss.
stevey stole smart phone technology from appy.
It is unethical for boss to steal business from rival companies.

$\forall X, smartPhoneTech(X) \Rightarrow business(X)$
$\forall X, competitor(X, appy) \vee competitor(appy, X) \Rightarrow rival(X)$
$\forall X, Y, Z, boss(X) \wedge steal(X, Y) \wedge business(Z) \wedge develop(Y, Z) \wedge rival(Y) \Rightarrow unethical(X)$

2. Write these FOL statements as Prolog clauses.

To prove that Stevey is unethical, sumsum and appy must be competitors. Stevey must be the boss of appy, who steals the smart phone technology galacticas3 that must be developed by sumsum.

```
company(sumsum).
company(appy).
competitor(sumsum,appy).
smart_phone_tech(galacticas3).
develop(sumsum,galacticas3).
boss(stevey).
steal(stevey,galacticas3,sumsum).

business(X) :- smart_phone_tech(X).
rival(X) :- competitor(X,appy);competitor(appy,X).
unethical(X) :- boss(X),smart_phone_tech(Y),rival(Z),company(Z),steal(X,Y,Z).
```

3. Using Prolog, prove that Stevey is unethical. Show a trace of your proof.

```
[trace]  ?- unethical(stevey).
   Call: (10) unethical(stevey) ? creep
   Call: (11) boss(stevey) ? creep
   Exit: (11) boss(stevey) ? creep
   Call: (11) smart_phone_tech(_21138) ? creep
   Exit: (11) smart_phone_tech(galacticas3) ? creep
   Call: (11) rival(_21226) ? creep
   Call: (12) competitor(_21270, 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) steal(stevey, galacticas3, sumsum) ? creep
   Exit: (11) steal(stevey, galacticas3, sumsum) ? creep
   Exit: (10) unethical(stevey) ? creep
true .
```

As such, Stevey is unethical.

Exercise 2: The Royal Family

The old Royal succession rule states that the throne is passed down along the male line according to the order of birth before the consideration along the female line – similarly according to the order of birth. queen elizabeth, the monarch of United Kingdom, has four offsprings; namely:- prince charles, princess ann, prince andrew and prince edward – listed in the order of birth.

1. Define their relations and rules in a Prolog rule base. Hence, define the old Royal succession rule. Using this old succession rule determine the line of succession based on the information given. Do a trace to show your results.

In the old succession rule, the older male will always be nearer in line to the throne as gender has more priority over age. The age will only be considered if there are many males and/or females. Thus, the succession rule will be ordered like so: males first and in order of birth, and then females in order of birth. This is shown in the prolog code below (**old_succession_rule**).

If both children in consideration are male, the older one will be first in line followed by the younger one. If one is male and one is female, then the male will be first followed by the female. If both children in consideration are female, then the older one will be first in line followed by the younger one.

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

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

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

is_older(X, Y) :- older_than(X, Y);
                  (older_than(X, Z),is_older(Z, Y)).

/* male according to order of birth then female according to order of birth */
old_succession_rule(X, Y) :- male(X), male(Y), is_older(X, Y);
                             male(X),female(Y);
                             female(X),female(Y),is_older(X,Y).

old_insert(A, [B|C], [B|D]) :- (old_succession_rule(B, A)),!,old_insert(A, C, D).
old_insert(A, C, [A|C]).
old_order_sort([A|B], Sorted_List) :- old_order_sort(B, Partial_order),
                                      old_insert(A, Partial_order, Sorted_List).
old_order_sort([], []).

old_line_of_succession(Queen, Successors) :- findall(Y, offspring_of(Y, Queen), OffspringList),
                                             old_order_sort(OffspringList, Successors).
```

```
[trace]  ?- old_line_of_succession(queen_elizabeth, Successors).
   Call: (10) old_line_of_succession(queen_elizabeth, _20648) ? creep
^  Call: (11) findall(_21040, offspring_of(_21040, queen_elizabeth), _21102) ? creep
   Call: (16) offspring_of(_21040, queen_elizabeth) ? creep
   Exit: (16) offspring_of(prince_charles, queen_elizabeth) ? creep
   Redo: (16) offspring_of(_21040, queen_elizabeth) ? creep
   Exit: (16) offspring_of(princess_ann, queen_elizabeth) ? creep
   Redo: (16) offspring_of(_21040, queen_elizabeth) ? creep
   Exit: (16) offspring_of(prince_andrew, queen_elizabeth) ? creep
   Redo: (16) offspring_of(_21040, queen_elizabeth) ? creep
   Exit: (16) offspring_of(prince_edward, queen_elizabeth) ? creep
^  Exit: (11) findall(_21040, user:offspring_of(_21040, queen_elizabeth), [prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
   Call: (11) old_order_sort([prince_charles, princess_ann, prince_andrew, prince_edward], _20648) ? creep
   Call: (12) old_order_sort([princess_ann, prince_andrew, prince_edward], _21642) ? creep
   Call: (13) old_order_sort([prince_andrew, prince_edward], _21686) ? creep
   Call: (14) old_order_sort([prince_edward], _21730) ? creep
   Call: (15) old_order_sort([], _21774) ? creep
   Exit: (15) old_order_sort([], []) ? creep
   Call: (15) old_insert(prince_edward, [], _21864) ? creep
   Exit: (15) old_insert(prince_edward, [], [prince_edward]) ? creep
   Exit: (14) old_order_sort([prince_edward], [prince_edward]) ? creep
   Call: (14) old_insert(prince_andrew, [prince_edward], _22002) ? creep
   Call: (15) old_succession_rule(prince_edward, prince_andrew) ? creep
   Call: (16) male(prince_edward) ? creep
   Exit: (16) male(prince_edward) ? creep
   Call: (16) male(prince_andrew) ? creep
   Exit: (16) male(prince_andrew) ? creep
   Call: (16) is_older(prince_edward, prince_andrew) ? creep
   Call: (17) older_than(prince_edward, prince_andrew) ? creep
   Fail: (17) older_than(prince_edward, prince_andrew) ? creep
   Redo: (16) is_older(prince_edward, prince_andrew) ? creep
   Call: (17) older_than(prince_edward, _22446) ? creep
   Fail: (17) older_than(prince_edward, _22490) ? creep
   Fail: (16) is_older(prince_edward, prince_andrew) ? creep
   Redo: (15) old_succession_rule(prince_edward, prince_andrew) ? creep
   Call: (16) male(prince_edward) ? creep
   Exit: (16) male(prince_edward) ? creep
   Call: (16) female(prince_andrew) ? creep
   Fail: (16) female(prince_andrew) ? creep
   Redo: (15) old_succession_rule(prince_edward, prince_andrew) ? creep
   Call: (16) female(prince_edward) ? creep
   Fail: (16) female(prince_edward) ? creep
   Fail: (15) old_succession_rule(prince_edward, prince_andrew) ? creep
   Redo: (14) old_insert(prince_andrew, [prince_edward], _22976) ? creep
   Exit: (14) old_insert(prince_andrew, [prince_edward], [prince_andrew, prince_edward]) ? creep
   Exit: (13) old_order_sort([prince_andrew, prince_edward], [prince_andrew, prince_edward]) ? creep
   Call: (13) old_insert(princess_ann, [prince_andrew, prince_edward], _23114) ? creep
   Call: (14) old_succession_rule(prince_andrew, princess_ann) ? creep
   Call: (15) male(prince_andrew) ? creep
   Exit: (15) male(prince_andrew) ? creep
   Call: (15) male(princess_ann) ? creep
   Fail: (15) male(princess_ann) ? creep
   Redo: (14) old_succession_rule(prince_andrew, princess_ann) ? creep
   Call: (15) male(prince_andrew) ? creep
   Exit: (15) male(prince_andrew) ? creep
   Call: (15) female(princess_ann) ? creep
   Exit: (15) female(princess_ann) ? creep
   Exit: (14) old_succession_rule(prince_andrew, princess_ann) ? creep
   Call: (14) old_insert(princess_ann, [prince_edward], _23104) ? creep
   Call: (15) old_succession_rule(prince_edward, princess_ann) ? creep
   Call: (16) male(prince_edward) ? creep
   Exit: (16) male(prince_edward) ? creep
   Call: (16) male(princess_ann) ? creep
   Fail: (16) male(princess_ann) ? creep
   Redo: (15) old_succession_rule(prince_edward, princess_ann) ? creep
   Call: (16) male(prince_edward) ? creep
   Exit: (16) male(prince_edward) ? creep
   Call: (16) female(princess_ann) ? creep
   Exit: (16) female(princess_ann) ? creep
   Exit: (15) old_succession_rule(prince_edward, princess_ann) ? creep
   Call: (15) old_insert(princess_ann, [], _23638) ? creep
   Exit: (15) old_insert(princess_ann, [], [princess_ann]) ? creep
   Exit: (14) old_insert(princess_ann, [prince_edward], [prince_edward, princess_ann]) ? creep
   Exit: (13) old_insert(princess_ann, [prince_andrew, prince_edward], [prince_andrew, prince_edward, princess_ann]) ? creep
   Exit: (12) old_order_sort([princess_ann, prince_andrew, prince_edward], [prince_andrew, prince_edward, princess_ann]) ? creep
   Call: (12) old_insert(prince_charles, [prince_andrew, prince_edward, princess_ann], _20648) ? creep
   Call: (13) old_succession_rule(prince_andrew, prince_charles) ? creep
   Call: (14) male(prince_andrew) ? creep
   Exit: (14) male(prince_andrew) ? creep
   Call: (14) male(prince_charles) ? creep
   Exit: (14) male(prince_charles) ? creep
   Call: (14) is_older(prince_andrew, prince_charles) ? creep
   Call: (15) older_than(prince_andrew, prince_charles) ? creep
   Fail: (15) older_than(prince_andrew, prince_charles) ? creep
   Redo: (14) is_older(prince_andrew, prince_charles) ? creep
   Call: (15) older_than(prince_andrew, _24852) ? creep
   Exit: (15) older_than(prince_andrew, prince_edward) ? creep
   Call: (15) is_older(prince_edward, prince_charles) ? creep
   Call: (16) older_than(prince_edward, prince_charles) ? creep
   Fail: (16) older_than(prince_edward, prince_charles) ? creep
   Redo: (15) is_older(prince_edward, prince_charles) ? creep
   Call: (16) older_than(prince_edward, _25116) ? creep
   Fail: (16) older_than(prince_edward, _25160) ? creep
   Fail: (15) is_older(prince_edward, prince_charles) ? creep
   Fail: (14) is_older(prince_andrew, prince_charles) ? creep
   Redo: (13) old_succession_rule(prince_andrew, prince_charles) ? creep
   Call: (14) male(prince_andrew) ? creep
   Exit: (14) male(prince_andrew) ? creep
   Call: (14) female(prince_charles) ? creep
   Fail: (14) female(prince_charles) ? creep
   Redo: (13) old_succession_rule(prince_andrew, prince_charles) ? creep
   Call: (14) female(prince_andrew) ? creep
   Fail: (14) female(prince_andrew) ? creep
   Fail: (13) old_succession_rule(prince_andrew, prince_charles) ? creep
   Redo: (12) old_insert(prince_charles, [prince_andrew, prince_edward, princess_ann], _20648) ? creep
   Exit: (12) old_insert(prince_charles, [prince_andrew, prince_edward, princess_ann], [prince_charles, prince_andrew, prince_edward, princess_ann]) ? creep
   Exit: (11) old_order_sort([prince_charles, princess_ann, prince_andrew, prince_edward], [prince_charles, prince_andrew, prince_edward, princess_ann]) ? creep
   Exit: (10) old_line_of_succession(queen_elizabeth, [prince_charles, prince_andrew, prince_edward, princess_ann]) ? creep
Successors = [prince_charles, prince_andrew, prince_edward, princess_ann].
```

As such, line of succession is in this order: Prince Charles, Prince Andrew, Prince Edward, Princess Ann.

2.	Recently, the Royal succession rule has been modified. The throne is now passed down according to the order of birth irrespective of gender. Modify your rules and Prolog knowledge base to handle the new succession rule. Explain the necessary changes to the knowledge needed to represent the new information. Use this new succession rule to determine the new line of succession based on the same knowledge given. Show your results using a trace.

In the new succession rule, only the age is considered regardless of male or female. Thus, the succession rule will be ordered like so: any gender in order of birth. This is shown in the prolog code below (**new_succession_rule**).

Here, regardless of whether both children in consideration are (1) both male, (2) one male and one female, (3) one female and one male, (4), both female, the older one will be first in line followed by the younger one.

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

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

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

is_older(X, Y) :- older_than(X, Y);
                  (older_than(X, Z),is_older(Z, Y)).

/* order of birth irrespective of gender*/
new_succession_rule(X, Y) :- male(X),male(Y),is_older(X, Y);
                             male(X),female(Y),is_older(X, Y);
                             female(X),male(Y),is_older(X, Y);
                             female(X),female(Y),is_older(X, Y).

new_insert(A, [B|C], [B|D]) :- (new_succession_rule(B, A)), !, new_insert(A, C, D).
new_insert(A,C,[A|C]).

new_order_sort([A|B], Sorted_List) :- new_order_sort(B, Partial_order),
                                      new_insert(A, Partial_order, Sorted_List).
new_order_sort([], []).

new_line_of_succession(Queen, Successors) :- findall(Y, offspring_of(Y, Queen), OffspringList),
                                             new_order_sort(OffspringList, Successors).
```

```
[trace]  ?- new_line_of_succession(queen_elizabeth, Successors).
   Call: (10) new_line_of_succession(queen_elizabeth, _20648) ? creep
^  Call: (11) findall(_21040, offspring_of(_21040, queen_elizabeth), _21102) ? creep
   Call: (16) offspring_of(_21040, queen_elizabeth) ? creep
   Exit: (16) offspring_of(prince_charles, queen_elizabeth) ? creep
   Redo: (16) offspring_of(_21040, queen_elizabeth) ? creep
   Exit: (16) offspring_of(princess_ann, queen_elizabeth) ? creep
   Redo: (16) offspring_of(_21040, queen_elizabeth) ? creep
   Exit: (16) offspring_of(prince_andrew, queen_elizabeth) ? creep
   Redo: (16) offspring_of(_21040, queen_elizabeth) ? creep
   Exit: (16) offspring_of(prince_edward, queen_elizabeth) ? creep
^  Exit: (11) findall(_21040, user:offspring_of(_21040, queen_elizabeth), [prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
   Call: (11) new_order_sort([prince_charles, princess_ann, prince_andrew, prince_edward], _20648) ? creep
   Call: (12) new_order_sort([princess_ann, prince_andrew, prince_edward], _21642) ? creep
   Call: (13) new_order_sort([prince_andrew, prince_edward], _21686) ? creep
   Call: (14) new_order_sort([prince_edward], _21730) ? creep
   Call: (15) new_order_sort([], _21774) ? creep
   Exit: (15) new_order_sort([], []) ? creep
   Call: (15) new_insert(prince_edward, [], _21864) ? creep
   Exit: (15) new_insert(prince_edward, [], [prince_edward]) ? creep
   Exit: (14) new_order_sort([prince_edward], [prince_edward]) ? creep
   Call: (14) new_insert(prince_andrew, [prince_edward], _22002) ? creep
   Call: (15) new_succession_rule(prince_edward, prince_andrew) ? creep
   Call: (16) male(prince_edward) ? creep
   Exit: (16) male(prince_edward) ? creep
   Call: (16) male(prince_andrew) ? creep
   Exit: (16) male(prince_andrew) ? creep
   Call: (16) is_older(prince_edward, prince_andrew) ? creep
   Call: (17) older_than(prince_edward, prince_andrew) ? creep
   Fail: (17) older_than(prince_edward, prince_andrew) ? creep
   Redo: (16) is_older(prince_edward, prince_andrew) ? creep
   Call: (17) older_than(prince_edward, _22446) ? creep
   Fail: (17) older_than(prince_edward, _22490) ? creep
   Fail: (16) is_older(prince_edward, prince_andrew) ? creep
   Redo: (15) new_succession_rule(prince_edward, prince_andrew) ? creep
   Call: (16) male(prince_edward) ? creep
   Exit: (16) male(prince_edward) ? creep
   Call: (16) female(prince_andrew) ? creep
   Fail: (16) female(prince_andrew) ? creep
   Redo: (15) new_succession_rule(prince_edward, prince_andrew) ? creep
   Call: (16) female(prince_edward) ? creep
   Fail: (16) female(prince_edward) ? creep
   Redo: (15) new_succession_rule(prince_edward, prince_andrew) ? creep
   Call: (16) female(prince_edward) ? creep
   Fail: (16) female(prince_edward) ? creep
   Fail: (15) new_succession_rule(prince_edward, prince_andrew) ? creep
   Redo: (14) new_insert(prince_andrew, [prince_edward], _23108) ? creep
   Exit: (14) new_insert(prince_andrew, [prince_edward], [prince_andrew, prince_edward]) ? creep
   Exit: (13) new_order_sort([prince_andrew, prince_edward], [prince_andrew, prince_edward]) ? creep
   Call: (13) new_insert(princess_ann, [prince_andrew, prince_edward], _23246) ? creep
   Call: (14) new_succession_rule(prince_andrew, princess_ann) ? creep
   Call: (15) male(prince_andrew) ? creep
   Exit: (15) male(prince_andrew) ? creep
   Call: (15) male(princess_ann) ? creep
   Fail: (15) male(princess_ann) ? creep
   Redo: (14) new_succession_rule(prince_andrew, princess_ann) ? creep
   Call: (15) male(prince_andrew) ? creep
   Exit: (15) male(prince_andrew) ? creep
   Call: (15) female(princess_ann) ? creep
   Exit: (15) female(princess_ann) ? creep
   Call: (15) is_older(prince_andrew, princess_ann) ? creep
   Call: (16) older_than(prince_andrew, princess_ann) ? creep
   Fail: (16) older_than(prince_andrew, princess_ann) ? creep
   Redo: (15) is_older(prince_andrew, princess_ann) ? creep
   Call: (16) older_than(prince_andrew, _23910) ? creep
   Exit: (16) older_than(prince_andrew, prince_edward) ? creep
   Call: (16) is_older(prince_edward, princess_ann) ? creep
   Call: (17) older_than(prince_edward, princess_ann) ? creep
   Fail: (17) older_than(prince_edward, princess_ann) ? creep
   Redo: (16) is_older(prince_edward, princess_ann) ? creep
   Call: (17) older_than(prince_edward, _24174) ? creep
   Fail: (17) older_than(prince_edward, _24218) ? creep
   Fail: (16) is_older(prince_edward, princess_ann) ? creep
   Fail: (15) is_older(prince_andrew, princess_ann) ? creep
   Redo: (14) new_succession_rule(prince_andrew, princess_ann) ? creep
   Call: (15) female(prince_andrew) ? creep
   Fail: (15) female(prince_andrew) ? creep
   Redo: (14) new_succession_rule(prince_andrew, princess_ann) ? creep
   Call: (15) female(prince_andrew) ? creep
   Fail: (15) female(prince_andrew) ? creep
   Fail: (14) new_succession_rule(prince_andrew, princess_ann) ? creep
   Redo: (13) new_insert(princess_ann, [prince_andrew, prince_edward], _24660) ? creep
   Exit: (13) new_insert(princess_ann, [prince_andrew, prince_edward], [princess_ann, prince_andrew, prince_edward]) ? creep
   Exit: (12) new_order_sort([princess_ann, prince_andrew, prince_edward], [princess_ann, prince_andrew, prince_edward]) ? creep
   Call: (12) new_insert(prince_charles, [princess_ann, prince_andrew, prince_edward], _20648) ? creep
   Call: (13) new_succession_rule(princess_ann, prince_charles) ? creep
   Call: (14) male(princess_ann) ? creep
   Fail: (14) male(princess_ann) ? creep
   Redo: (13) new_succession_rule(princess_ann, prince_charles) ? creep
   Call: (14) male(princess_ann) ? creep
   Fail: (14) male(princess_ann) ? creep
   Redo: (13) new_succession_rule(princess_ann, prince_charles) ? creep
   Call: (14) female(princess_ann) ? creep
   Exit: (14) female(princess_ann) ? creep
   Call: (14) male(prince_charles) ? creep
   Exit: (14) male(prince_charles) ? creep
   Call: (14) is_older(princess_ann, prince_charles) ? creep
   Call: (15) older_than(princess_ann, prince_charles) ? creep
   Fail: (15) older_than(princess_ann, prince_charles) ? creep
   Redo: (14) is_older(princess_ann, prince_charles) ? creep
   Call: (15) older_than(princess_ann, _25506) ? creep
   Exit: (15) older_than(princess_ann, prince_andrew) ? creep
   Call: (15) is_older(prince_andrew, prince_charles) ? creep
   Call: (16) older_than(prince_andrew, prince_charles) ? creep
   Fail: (16) older_than(prince_andrew, prince_charles) ? creep
   Redo: (15) is_older(prince_andrew, prince_charles) ? creep
   Call: (16) older_than(prince_andrew, _25770) ? creep
   Exit: (16) older_than(prince_andrew, prince_edward) ? creep
   Call: (16) is_older(prince_edward, prince_charles) ? creep
   Call: (17) older_than(prince_edward, prince_charles) ? creep
   Fail: (17) older_than(prince_edward, prince_charles) ? creep
   Redo: (16) is_older(prince_edward, prince_charles) ? creep
   Call: (17) older_than(prince_edward, _26034) ? creep
   Fail: (17) older_than(prince_edward, _26078) ? creep
   Fail: (16) is_older(prince_edward, prince_charles) ? creep
   Fail: (15) is_older(prince_andrew, prince_charles) ? creep
   Fail: (14) is_older(princess_ann, prince_charles) ? creep
   Redo: (13) new_succession_rule(princess_ann, prince_charles) ? creep
   Call: (14) female(princess_ann) ? creep
   Exit: (14) female(princess_ann) ? creep
   Call: (14) female(prince_charles) ? creep
   Fail: (14) female(prince_charles) ? creep
   Fail: (13) new_succession_rule(princess_ann, prince_charles) ? creep
   Redo: (12) new_insert(prince_charles, [princess_ann, prince_andrew, prince_edward], _20648) ? creep
   Exit: (12) new_insert(prince_charles, [princess_ann, prince_andrew, prince_edward], [prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
   Exit: (11) new_order_sort([prince_charles, princess_ann, prince_andrew, prince_edward], [prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
   Exit: (10) new_line_of_succession(queen_elizabeth, [prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
Successors = [prince_charles, princess_ann, prince_andrew, prince_edward].
```

As such, the new line of succession is in this order: Prince Charles, Princess Ann, Prince Andrew, Prince Edward.