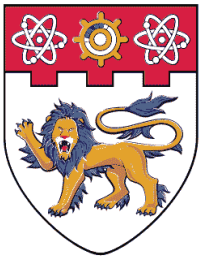

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC3000

Artificial Intelligence

Lab 2

A35

Name	Matriculation No.
Lim Dong Wan	U2122455D
Loke Yong Jian	U2120476G
Nalin Sharma	U2121904E

Table of Contents

Contributions

Loke Yong Jian	Exercise 1
Nalin Sharma, Lim Dong Wan	Exercise 2 Question 1
Nalin Sharma, Lim Dong Wan	Exercise 2 Question 2

Exercise 1

sumsum, a competitor of appy, developed some nice smart phone technology called galactica-s3, all of which was stolen by stevey, who is a boss of appy. It is unethical for a boss to steal business from rival companies. A competitor of 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).

(5 marks)

2. Write these FOL statements as Prolog clauses.

(5 marks)

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

(5 marks)

1.

Statements	FOL

Sumsum is a competitor of appy
 	
- sumsum and appy are companies
- competitor is a two-way relationship
	`company(sumsum)`
	`company(appy)`
	`competitors(sumsum, appy)`
	`∀x, ∀y, competitor(x,y) ⇔ competitor(y,x)`

Galactica-s3 is a smartphone technology	`smartphoneTech(galactica-s3)`
Sumsum developed galactica-s3	`developed(sumsum, galactica-s3)`
Galactica-s3 is stolen by stevey	`steal(stevey, galactica-s3)`
Stevey is a boss of appy	`boss(stevey, appy)`
Competitor is rival	`∀x, ∀y. (competitor(x,y) => rival(x,y))`
SmartphoneTech is a business	`∀x. (smartphoneTech(x) => business(x))`

Unethical for a boss to steal business from rival companies It is unethical if: - B and C are company - A is the boss of Company B - Company B and Company C are rival - Company C developed Technology D - A steal Technology D - Technology D is a business	∀A, ∀B, ∀C, ∀D. (company(B) ∧ company(C) ∧ boss(A,B) ∧ rival(B,C) ∧ developed(C,D) ∧ steal(A,D) ∧ business(D) => is_unethical(A))

2.

Relations

```
company(appy).
company(sumsum).
competitor(appy, sumsum).
competitor(sumsum, appy).

smartphoneTech('galactica-s3').
developed(sumsum, 'galactica-s3').

steal(stevey, 'galactica-s3').
boss(stevey, appy).
```

Rules

```
rival(X, Y) :-
      competitor(X, Y),
      competitor(Y, X).

business(X) :-
      smartphoneTech(X).

is_unethical(A) :-
      company(B),
      company(C),
      boss(A, B),
```

```
    rival(B, C),
    developed(C, D),
    steal(A, D),
    business(D).
```

3.

[Query for Tracing]

```
is_unethical(stevey).
```

Trace on:

```
is_unethical(stevey).
```

```
[trace]  ?- is_unethical(stevey).
   Call: (10) is_unethical(stevey) ? creep
   Call: (11) company(_18604) ? creep
   Exit: (11) company(appy) ? creep
   Call: (11) company(_20218) ? creep
   Exit: (11) company(appy) ? creep
   Call: (11) boss(stevey, appy) ? creep
   Exit: (11) boss(stevey, appy) ? creep
   Call: (11) rival(appy, appy) ? creep
   Call: (12) competitor(appy, appy) ? creep
   Fail: (12) competitor(appy, appy) ? creep
   Fail: (11) rival(appy, appy) ? creep
   Redo: (11) company(_20218) ? creep
   Exit: (11) company(sumsum) ? creep
   Call: (11) boss(stevey, appy) ? creep
   Exit: (11) boss(stevey, appy) ? creep
   Call: (11) rival(appy, sumsum) ? creep
   Call: (12) competitor(appy, sumsum) ? creep
   Exit: (12) competitor(appy, sumsum) ? creep
   Call: (12) competitor(sumsum, appy) ? creep
   Exit: (12) competitor(sumsum, appy) ? creep
   Exit: (11) rival(appy, sumsum) ? creep
   Call: (11) developed(sumsum, _34784) ? creep
   Exit: (11) developed(sumsum, 'galactica-s3') ? creep
   Call: (11) steal(stevey, 'galactica-s3') ? creep
   Exit: (11) steal(stevey, 'galactica-s3') ? creep
   Call: (11) business('galactica-s3') ? creep
   Call: (12) smartphoneTech('galactica-s3') ? creep
   Exit: (12) smartphoneTech('galactica-s3') ? creep
   Exit: (11) business('galactica-s3') ? creep
   Exit: (10) is_unethical(stevey) ? creep
true .
```

Exercise 2

Question 1

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.

(5 marks)

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.

(5 marks)

Relations

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

female(princess_ann).

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

older(prince_charles, princess_ann).
older(prince_charles, prince_andrew).
older(prince_charles, prince_edward).
older(princess_ann, prince_andrew).
```

```
older(princess_ann, prince_edward).
older(prince_andrew, prince_edward).
```

Old Royal Succession Rule

The Old Royal Succession rule can be defined as follows:

```
succession(X, Y) :-
      male(X),
      male(Y),
      older(X, Y). % older gets precedence

succession(X,Y) :-
      female(X),
      female(Y),
      older(X, Y). % older gets precedence

succession(X,Y) :-
      male(X),
      female(Y). % X gets precedence if male
```

- The first rule dictates that if X and Y are both males, then X is ahead of Y in the line of succession if he is older than X.
- Second, if X and Y are both females, then X is ahead of Y in the line of succession if she is older than Y.
- And third, if X is male and Y is female, then X is ahead of Y in the line of succession.

Queries for Tracing

1. `succession(prince_charles, X).`
2. `succession(prince_andrew, X).`
3. `succession(prince_edward, X).`
4. `succession(princess_ann, X).`

Trace on:

a) succession(prince_charles,X).

```
[trace]  ?- succession(prince_charles,X).
   Call: (10) succession(prince_charles, _27846) ? creep
   Call: (11) male(prince_charles) ? creep
   Exit: (11) male(prince_charles) ? creep
   Call: (11) male(_27846) ? creep
   Exit: (11) male(prince_charles) ? creep
   Call: (11) older(prince_charles, prince_charles) ? creep
   Fail: (11) older(prince_charles, prince_charles) ? creep
   Redo: (11) male(_27846) ? creep
   Exit: (11) male(prince_andrew) ? creep
   Call: (11) older(prince_charles, prince_andrew) ? creep
   Exit: (11) older(prince_charles, prince_andrew) ? creep
   Exit: (10) succession(prince_charles, prince_andrew) ? creep
X = prince_andrew ,
```

b) succession(prince_andrew,X).

```
[trace]  ?- succession(prince_andrew,X).
   Call: (10) succession(prince_andrew, _40320) ? creep
   Call: (11) male(prince_andrew) ? creep
   Exit: (11) male(prince_andrew) ? creep
   Call: (11) male(_40320) ? creep
   Exit: (11) male(prince_charles) ? creep
   Call: (11) older(prince_andrew, prince_charles) ? creep
   Fail: (11) older(prince_andrew, prince_charles) ? creep
   Redo: (11) male(_40320) ? creep
   Exit: (11) male(prince_andrew) ? creep
   Call: (11) older(prince_andrew, prince_andrew) ? creep
   Fail: (11) older(prince_andrew, prince_andrew) ? creep
   Redo: (11) male(_40320) ? creep
   Exit: (11) male(prince_edward) ? creep
   Call: (11) older(prince_andrew, prince_edward) ? creep
   Exit: (11) older(prince_andrew, prince_edward) ? creep
   Exit: (10) succession(prince_andrew, prince_edward) ? creep
X = prince_edward ,
```

c) succession(prince_edward,X).

```
[trace]  ?- succession(prince_edward,X).
   Call: (10) succession(prince_edward, _56032) ? creep
   Call: (11) male(prince_edward) ? creep
   Exit: (11) male(prince_edward) ? creep
   Call: (11) male(_56032) ? creep
   Exit: (11) male(prince_charles) ? creep
   Call: (11) older(prince_edward, prince_charles) ? creep
   Fail: (11) older(prince_edward, prince_charles) ? creep
   Redo: (11) male(_56032) ? creep
   Exit: (11) male(prince_andrew) ? creep
   Call: (11) older(prince_edward, prince_andrew) ? creep
   Fail: (11) older(prince_edward, prince_andrew) ? creep
   Redo: (11) male(_56032) ? creep
   Exit: (11) male(prince_edward) ? creep
   Call: (11) older(prince_edward, prince_edward) ? creep
   Fail: (11) older(prince_edward, prince_edward) ? creep
   Redo: (10) succession(prince_edward, _56032) ? creep
   Call: (11) female(prince_edward) ? creep
   Fail: (11) female(prince_edward) ? creep
   Redo: (10) succession(prince_edward, _56032) ? creep
   Call: (11) male(prince_edward) ? creep
   Exit: (11) male(prince_edward) ? creep
   Call: (11) female(_56032) ? creep
   Exit: (11) female(princess_ann) ? creep
   Exit: (10) succession(prince_edward, princess_ann) ? creep
X = princess_ann.
```

d) succession(princess_ann,X).

```
[trace]  ?- succession(princess_ann,X).
   Call: (10) succession(princess_ann, _17350) ? creep
   Call: (11) male(princess_ann) ? creep
   Fail: (11) male(princess_ann) ? creep
   Redo: (10) succession(princess_ann, _17350) ? creep
   Call: (11) female(princess_ann) ? creep
   Exit: (11) female(princess_ann) ? creep
   Call: (11) female(_17350) ? creep
   Exit: (11) female(princess_ann) ? creep
   Call: (11) older(princess_ann, princess_ann) ? creep
   Fail: (11) older(princess_ann, princess_ann) ? creep
   Redo: (10) succession(princess_ann, _17350) ? creep
   Call: (11) male(princess_ann) ? creep
   Fail: (11) male(princess_ann) ? creep
   Fail: (10) succession(princess_ann, _17350) ? creep
false.
```

→ This indicates that in the line of succession, the order is princess_ann < prince_edward < prince_andrew < prince_charles.

Hence, according to the old rule, the order is:

1. **Prince Charles**
2. **Prince Andrew**
3. **Prince Edward**
4. **Princess Ann**

Question 2

Relations

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

female(princess_ann).

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

older(prince_charles, princess_ann).
older(prince_charles, prince_andrew).
older(prince_charles, prince_edward).
older(princess_ann, prince_andrew).
older(princess_ann, prince_edward).
older(prince_andrew, prince_edward).
```

New Royal Succession Rule

The new Royal Succession rule states that the throne is passed down according to birth order and is no longer based on gender. Hence, regardless of whether X and Y are male or female, as long as X is older than Y, then X is ahead of Y in the line of succession.

We will modify the rule such that we merge both of our original succession rules together into a more generalized succession rule.

```
succession(X,Y) :-
     older(X, Y).
```

Queries for Tracing

```
a) succession(prince_charles, X).
b) succession(prince_andrew, X).
c) succession(prince_edward, X).
d) succession(princess_ann, X).
```

Trace on:

a) succession(prince_charles,X).

```
[trace]  ?- succession(prince_charles, X).
   Call: (10) succession(prince_charles, _38946) ? creep
   Call: (11) older(prince_charles, _38946) ? creep
   Exit: (11) older(prince_charles, princess_ann) ? creep
   Exit: (10) succession(prince_charles, princess_ann) ? creep
X = princess_ann .
```

b) succession(princess_ann,X).

```
[trace]  ?- succession(princess_ann, X).
   Call: (10) succession(princess_ann, _32922) ? creep
   Call: (11) older(princess_ann, _32922) ? creep
   Exit: (11) older(princess_ann, prince_andrew) ? creep
   Exit: (10) succession(princess_ann, prince_andrew) ? creep
X = prince_andrew .
```

c) succession(prince_andrew,X).

```
[trace]  ?- succession(prince_andrew, X).
   Call: (10) succession(prince_andrew, _44970) ? creep
   Call: (11) older(prince_andrew, _44970) ? creep
   Exit: (11) older(prince_andrew, prince_edward) ? creep
   Exit: (10) succession(prince_andrew, prince_edward) ? creep
X = prince_edward.
```

d) succession(prince_edward,X).

```
[trace]  ?- succession(prince_edward, X).
   Call: (10) succession(prince_edward, _27794) ? creep
   Call: (11) older(prince_edward, _27794) ? creep
   Fail: (11) older(prince_edward, _27794) ? creep
   Fail: (10) succession(prince_edward, _27794) ? creep
false.
```

→ This indicates that in the line of succession, the order is prince_edward < prince_andrew < princess_ann < prince_charles.

Hence, according to the new rule, the order is:

1. **Prince Charles**
2. **Princess Ann**
3. **Prince Andrew**
4. **Prince Edward**