**SC3000 Prolog Traces (Assignment 2)**

## **Contributors**

| **No** | **Name** | **Contribution** |
| --- | --- | --- |
| **1** | **Luar Shui Yan (U2221566E)** | Translate english to Prolog for question 1  Researched and implemented trace logic for question 2 |
| **2** | **Tan Yong Jie (U2222722B)** | Translate english to Prolog for question 1  Researched and implemented trace logic for question 2 |

## **Question 1**

**Definitions**

| **Constants (words in blue)** | | |
| --- | --- | --- |
| **No** | **Name** | **Role** |
| **1** | sumsum | A (smartphone tech) company |
| **2** | appy | A (smartphone tech) company |
| **3** | galatica-s3 | A product of a company |
| **4** | stevey | A boss of a company |

### 

| **Predicates (self-defined)** | | |
| --- | --- | --- |
| **No** | **Name** | **Role** |
| **1** | smart\_phone\_tech(X) | X is a smart phone tech company |
| **2** | competitor(X, Y) | X is a competitor of Y |
| **3** | developed(X, Y) | X developed Y |
| **4** | stolen(X, Y, Z) | X stole Y from Z |
| **5** | boss(X, Y) | X is the boss of Y |
| **6** | unethical(X) | X is unethical |
| **7** | rival(X, Y) | X is a rival of Y |
| **8** | business(X) | X is business |
| **9** | company(X) | X is a company |

### **Q1.1 - English to FOL**

| **No** | **English** | **FOL** |
| --- | --- | --- |
| **1** |  | competitor(sumsum, appy) |
| **2** | [sumsum]  [s3] | developed(sumsum, galaticas3) |
| **3** | which = galaticas-s3 | stolen(stevey, galaticas3, sumsum) |
| **4** |  | boss(stevey, appy) |
| **5** |  | ∀a, ∀b, ∀c, ∀d,  boss(a, b) ∧ stolen(a, c, d) ∧ business(c) ∧ rival(b, d) ⇒ unethical(a) |
| **6** |  | ∀a, ∀b,  rival(a, b) ⇔ company(a) ∧ company(b) ∧ competitor(a, b) |
| **7** |  | ∀a,  smart\_phone\_tech(a) ⇒ business(a) |

### 

### **Q1.2 - FOL to Prolog**

| question1.pl contains the required Prolog statements |
| --- |

### **Q1.3 - Trace of unethical(stevey)**

|  |
| --- |

## **Question 2**

**Definitions**

| **Constants (words in blue)** | | |
| --- | --- | --- |
| **No** | **Name** | **Role** |
| **1** | queen elizabeth | People/Members of royal family |
| **2** | prince charles |
| **3** | princess ann |
| **4** | prince andrew |
| **5** | prince edward |

### 

| **Predicates (self-defined)** | | |
| --- | --- | --- |
| **No** | **Name** | **Role** |
| **1** | female(X) | X is female |
| **2** | male(X) | X is male |
| **3** | parent(X, Y) | X is the parent of Y |
| **4** | monarch(X) | X is the current monarch (king or queen) |
| **5** | older(X, Y) | X is older than Y |
| **6** | samegender(X, Y) | X and Y are both male or female |
| **7** | olderer(X, Y) | X is older than Z if:  X is older than Y and Y is older than Z |
| **8** | samegenderolder(X, Y) | X is same gender as Y and older than Y  (5) & (6) |
| **9** | maleoverfemale(X,Y) | X has precedence to the crown over Y iff:  X is male and Y is female |
| **10** | succorder(L, N) | Returns the succession order in a list L up to N characters |
| **11** | dfs functions | Recursive helper functions to find succ order  **There are 2 versions of dfs1\_helper.**  **2A accounts for gender, 2B does not.**  **Comment out 2B to run Old Succession order,**  **Comment out 2A to run New succession order.** |

### **Q2.1 - The Old Succession Order**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

## 

### 

### **Q2.2 - The New Succession Order**

|  |
| --- |
|  |

## 