

Homework 1  
TK2ICM: *Logic Programming* (CSH4Y3)  
*Chinese Lunisolar New Year*  
Second Term 2018-2019

Day, date : Tuesday, February 5, 2019

Duration : **9 hours (including submission: 12:00 a.m. – 9:00 p.m.)**

Type : ***open all, individual, cooperation is allowed***

Instruction:

1. You are allowed to discuss these problems with other class participants, but make sure that you write the script individually. Copying answers from elsewhere without understanding them will not enhance your understanding.
2. You may use any reference (books, slides, internet) as well as ask other students who are not enrolled to this class.
3. Use the predicate name as described in each of the problem. **The name of the predicate must be precisely identical.** Typographical error may lead to the cancellation of your points.
4. Submit your work to the provided slot at google classroom under the file name Hw1-<your\_name>.pl.  
For example: Hw1-Albert.pl. Please see an information regarding your nickname at google classroom.

Benjamin's family tree is depicted in Figure 1. The family tree is identical to that appeared in Problem Set 1.

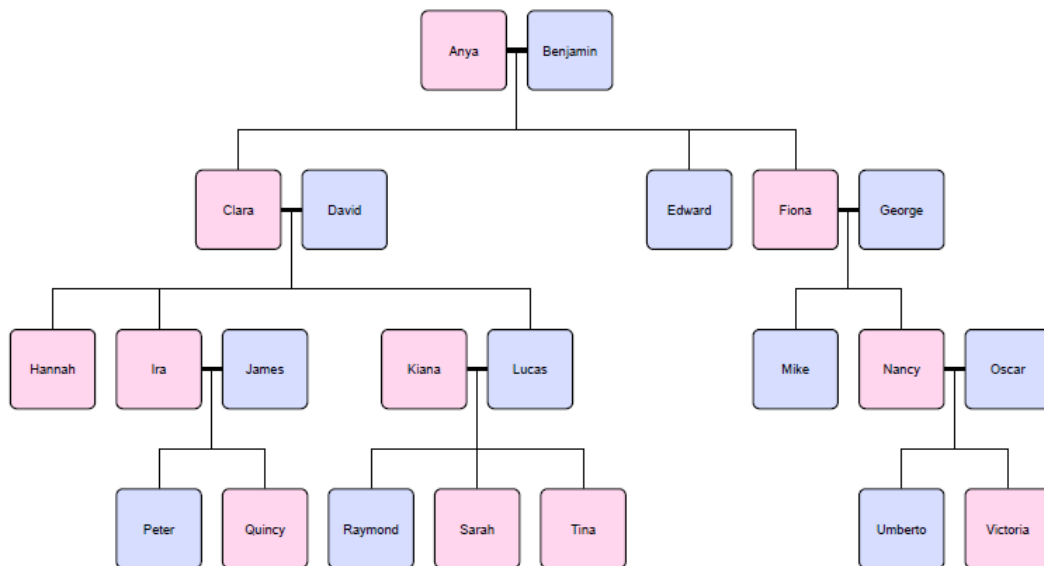


Figure 1: Benjamin's family tree.

---

We can derive a knowledge base from this tree as follows:

```
% male(X) denotes that X is a male.
male(benjamin).
male(david).  male(edward).  male(george).
male(james).  male(lucas).  male(mike).  male(oscar).
male(peter).  male(raymond).  male(umberto).

% female(X) denotes that X is female.
female(anya).
female(clara).  female(fiona).
female(hannah).  female(ira).  female(kiana).  female(nancy).
female(quincy).  female(sarah).  female(tina).  female(victoria).

% parent(X,Y) denotes that X is one of Y's parent.
parent(anya,clara).  parent(anya,edward).  parent(anya,fiona).
parent(benjamin,clara).  parent(benjamin,edward).  parent(benjamin,fiona).
parent(clara,hannah).  parent(clara,ira).  parent(clara,lucas).
parent(david,hannah).  parent(david,ira).  parent(david,lucas).
parent(fiona,mike).  parent(fiona,nancy).
parent(george,mike).  parent(george,nancy).
parent(ira,peter).  parent(ira,quincy).
parent(james,peter).  parent(james,quincy).
parent(kiana,raymond).  parent(kiana,sarah).  parent(kiana,tina).
parent(lucas,raymond).  parent(lucas,sarah).  parent(lucas,tina).
parent(nancy,umberto).  parent(nancy,victoria).
parent(oscar,umberto).  parent(oscar,victoria).

% birth(X,Y) explains the birth year of person X.
birth(anya,1938).  birth(benjamin,1929).

birth(clara,1959).  birth(david,1950).
birth(edward,1963).
birth(fiona,1965).  birth(george,1962).

birth(hannah,1980).
birth(ira,1982).  birth(james,1979).
birth(kiana,1990).  birth(lucas,1986).

birth(mike,1991).
birth(nancy,1994).  birth(oscar,1992).

birth(peter,2005).  birth(quincy,2008).
birth(raymond,2013).  birth(sarah,2015).  birth(tina,2018).
birth(umberto,2016).  birth(victoria,2019).
```

**Remark 1** You are prohibited to add any additional fact.

**Problem 1 (20 points)** Construct the rules for the following predicates:

(a). `brother_in_law(X, Y)` which means X is a brother in law of Y.

(b). `sister_in_law(X, Y)` which means X is a sister in law of Y.

Hint: X is a brother in law of Y if:

- (definition 1): X is the husband of Y's sister; or
- (definition 2): X is the brother of Y's husband or wife; or
- (definition 3): X is a man who is married to the sister or brother of Y's wife or husband.

Similar definition applies to sister in law as well.

**I/O examples**

Input: ?- `brother_in_law(james, hannah)`.

Output: **true**

Explanation: definition 1

Input: ?- `brother_in_law(lucas, james)`.

Output: **true**

Explanation: definition 2

Input: ?- `brother_in_law(david, george)`.

Output: **true**

Explanation: definition 3

Input: ?- `brother_in_law(kiana, james)`.

Output: **false**

Input: ?- `brother_in_law(lucas, hannah)`.

Output: **false**

Input: ?- `sister_in_law(kiana, ira)`.

Output: **true**

Explanation: definition 1

Input: ?- `sister_in_law(fiona, david)`.

Output: **true**

Explanation: definition 2

Input: ?- `sister_in_law(kiana, james)`.

Output: **true**

Explanation: definition 3

Input: ?- `sister_in_law(james, kiana)`.

Output: **false**

Input: ?- `sister_in_law(clara, fiona)`.

Output: **false**

---

**Problem 2 (20 points)** Construct the rules for the following predicates:

- (a). `son_in_law(X, Y)` which means X is a son in law of Y,
- (b). `daughter_in_law(X, Y)` which means X is a daughter in law of Y.

Hint: X is a son in law of Y if X is married to Y's daughter. Similarly, X is a daughter in law of Y if X is married to Y's son.

**I/O examples**

Input: ?- `son_in_law(david,anya)`.

Output: **true**

Input: ?- `son_in_law(george,benjamin)`.

Output: **true**

Input: ?- `son_in_law(edward,anya)`.

Output: **false**

Input: ?- `son_in_law(kiana,david)`.

Output: **false**

Input: ?- `son_in_law(james,clara)`.

Output: **true**

Input: ?- `daughter_in_law(kiana,david)`.

Output: **true**

Input: ?- `daughter_in_law(kiana,clara)`.

Output: **true**

Input: ?- `daughter_in_law(james,david)`.

Output: **false**

Input: ?- `daughter_in_law(clara,kiana)`.

Output: **false**

Input: ?- `daughter_in_law(quincy,ira)`.

Output: **false**

**Problem 3 (20 points)** Construct the rules for the following predicates:

- (a). `father_in_law(X,Y)` which means that X is father in law of Y,
- (b). `mother_in_law(X,Y)` which means that X is mother in law of Y.

Hint: you may use the `son_in_law(X,Y)` and `daughter_in_law(X,Y)` in Problem 2.

**I/O examples**

Input: ?- `father_in_law(benjamin,david)`.

Output: **true**

Input: ?- `father_in_law(david,kiana)`.

Output: **true**

Input: ?- `father_in_law(anya,george)`.

Output: **false**

Input: ?- `father_in_law(fiona,oscar)`.

Output: **false**

Input: ?- `father_in_law(george,oscar)`.

Output: **true**

Input: ?- `mother_in_law(fiona,oscar)`.

Output: **true**

Input: ?- `mother_in_law(clara,kiana)`.

Output: **true**

Input: ?- `mother_in_law(kiana,clara)`.

Output: **false**

Input: ?- `mother_in_law(anya,george)`.

Output: **true**

Input: ?- `mother_in_law(david,james)`.

Output: **false**

---

**Problem 4** Construct the rules for the following predicates:

- (a). `older(X, Y)` which means X is older than Y,
- (b). `younger(X, Y)` which means X is younger than Y.

Hint: use the predicate `birth/2` defined in the knowledge base. X is older than Y if X is born before Y, and X is younger than Y if X is born after Y.

**I/O examples**

Input: `?- older(benjamin,anya).`

Output: **true**

Input: `?- older(anya,benjamin).`

Output: **false**

Input: `?- younger(clara,george).`

Output: **false**

Input: `?- younger(george,clara).`

Output: **true**

Input: `?- older(mike,james).`

Output: **false**

Input: `?- younger(mike,james).`

Output: **true**

Input: `?- younger(mike,mike).`

Output: **false**

Input: `?- older(mike,mike).`

Output: **false**

**Problem 5 (20 points)** Construct the rules for the following predicates:

- (a). `xiaodidi(X, Y)` which means that X is a younger brother of Y,
- (b). `xiaomeimei(X, Y)` which means that X is a younger sister of Y,

Hint: use the predicate `older/2` and `younger/2` in Problem 4. The phrases *xiǎo dì dì* and *xiǎo mèi mèi* respectively refer to younger brother and younger sister in Mandarin.

**I/O examples**

Input: `?- xiaodidi(edward, clara).`

Output: **true**

Input: `?- xiaodidi(edward, fiona).`

Output: **false**

Input: `?- xiaodidi(lucas, hannah).`

Output: **true**

Input: `?- xiaodidi(lucas, ira).`

Output: **true**

Input: `?- xiaodidi(nancy, mike).`

Output: **false**

Input: `?- xiaomeimei(nancy, mike).`

Output: **true**

Input: `?- xiaomeimei(clara, edward).`

Output: **false**

Input: `?- xiaomeimei(fiona, edward).`

Output: **true**

Input: `?- xiaomeimei(edward, clara).`

Output: **false**

Input: `?- xiaomeimei(fiona, clara).`

Output: **true**