

Problem Set 1
TK2ICM: *Logic Programming* (CSH4Y3)
Second Term 2018-2019

Day, date : Tuesday, January 29, 2019
Duration : **60 minutes**
Type : ***open all***, individual (no cooperation between/among class participants)

Instruction:

1. You are not allowed to discuss these problems with other class participants.
2. You may use any reference (books, slides, internet) as well as 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 PS1-<your_name>.pl. For example: PS1-Albert.pl. Please see an information regarding your nickname at google classroom.

Benjamin's family tree is depicted in Figure 1.

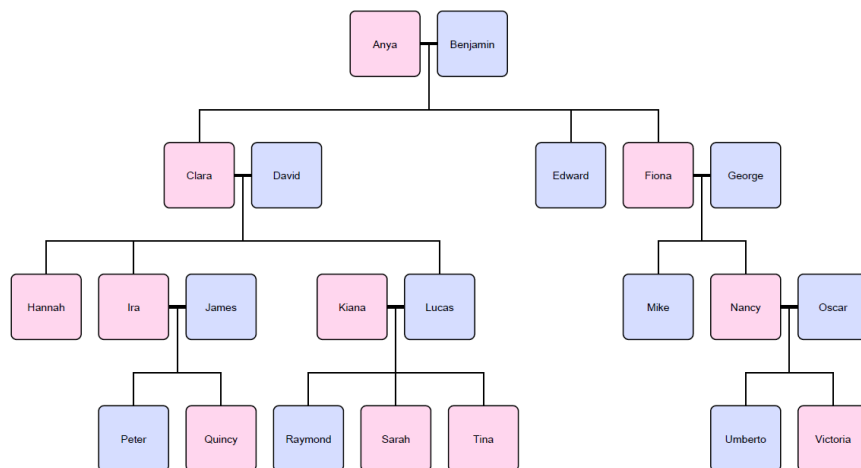


Figure 1: Benjamin's family tree.

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).
```

Remark 1 You are prohibited to add any additional fact.

Problem 1 (20 points) Assuming that all married couples have a child, construct the predicate `married(X, Y)` which means that X and Y are married couples.

I/O examples

Input: `married(anya, benjamin).`

Output: **true**

Input: `married(benjamin, anya).`

Output: **true**

Input: `married(anya, anya).`

Output: **false**

Input: `married(benjamin, benjamin).`

Output: **false**

Input: `married(anya, edward).`

Output: **false**

Input: `married(benjamin, clara).`

Output: **false**

Input: `married(kiana, david).`

Output: **false**

Input: `married(clara, david).`

Output: **true**

Input: `married(david, clara).`

Output: **true**

Input: `married(edward, fiona).`

Output: **false**

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

- (a). `is_a_grandson(X)` which means X is a grandson of someone,
- (b). `is_a_granddaughter(X)` which means X is a granddaughter of someone.

Hint: you may use the auxiliary predicate `grandchild(X,Y)` which means X is a grandchild of Y.

I/O examples

Input: `is_a_grandson(lucas).`

Output: **true**

Input: `is_a_grandson(mike).`

Output: **true**

Input: `is_a_granddaughter(hannah).`

Output: **true**

Input: `is_a_granddaughter(ira).`

Output: **true**

Input: `is_a_grandson(hannah).`

Output: **false**

Input: `is_a_grandson(ira).`

Output: **false**

Input: `is_a_granddaughter(lucas).`

Output: **false**

Input: `is_a_granddaughter(mike).`

Output: **false**

Input: `is_a_grandson(james).`

Output: **false**

Input: `is_a_granddaughter(kiana).`

Output: **false**

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

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

Hint: you may use the auxiliary predicate `sibling(X, Y)` which means X is a sibling of Y.

I/O examples

Input: `brother(edward, clara)`.

Output: **true**

Input: `brother(edward, edward)`.

Output: **false**

Input: `brother(lucas, hannah)`.

Output: **true**

Input: `brother(hannah, lucas)`.

Output: **false**

Input: `brother(james, hannah)`.

Output: **false**

Input: `sister(clara, fiona)`.

Output: **true**

Input: `sister(fiona, fiona)`.

Output: **false**

Input: `sister(ira, lucas)`.

Output: **true**

Input: `sister(lucas, hannah)`.

Output: **false**

Input: `sister(hannah, kiana)`.

Output: **false**

Problem 4 (20 points) Construct the rules for the following predicates:

- (a). `uncle(X, Y)` which means that X is an uncle of Y (X is Y's uncle),
(b). `aunt(X, Y)` which means that X is an aunt of Y (X is Y's aunt).

Hint: X is an uncle of Y (or X is Y's uncle) if:

- X is a brother of Y's parent (uncle by blood)
- X is married to the sister of Y's parent (uncle by marriage).

Similar definition applies to aunt as well.

I/O examples

Input: `uncle(edward, mike) .`

Output: **true**

Explanation: uncle by blood

Input: `uncle(mike, edward) .`

Output: **false**

Input: `uncle(james, raymond) .`

Output: **true**

Explanation: uncle by marriage

Input: `uncle(ira, raymond) .`

Output: **false**

Input: `uncle(oscar, umberto) .`

Output: **false**

Input: `aunt(clara, nancy) .`

Output: **true**

Explanation: aunt by blood

Input: `aunt(nancy, clara) .`

Output: **false**

Input: `aunt(kiana, quincy) .`

Output: **true.**

Explanation: aunt by marriage

Input: `aunt(lucas, quincy) .`

Output: **false.**

Input: `aunt(kiana, sarah) .`

Output: **false**

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

- (a). `is_a_nephew(X)` which means X a nephew of someone,
- (b). `is_a_niece(X)` which means X is a niece of someone.

Hint: you may use the result in Problem 4.

I/O examples

Input: `is_a_nephew(mike).`

Output: **true**

Input: `is_a_nephew(david).`

Output: **false**

Input: `is_a_nephew(raymond).`

Output: **true**

Input: `is_a_nephew(umberto).`

Output: **true**

Input: `is_a_nephew(sarah).`

Output: **false**

Input: `is_a_niece(ira).`

Output: **true**

Input: `is_a_niece(kiana).`

Output: **false**

Input: `is_a_niece(fiona).`

Output: **false**

Input: `is_a_niece(victoria).`

Output: **true**

Input: `is_a_niece(peter).`

Output: **false**