Prolog

School of Data and Computer Science
Sun Yat-sen University

Overview

Prolog

- Prolog, a logic programming language, plays an important role in artificial intelligence (AI).
- Unlike many other programming languages, Prolog is intended primarily as a declarative programming language.
- In Prolog, logic is expressed as relations (called as Facts and Rules).
- Formulation or Computation is carried out by running a query over these relations.

Prolog (cont'd)

Advantages

- Easy to build database.
- Pattern matching is easy. Search is recursion based.
- Easy to play with any algorithm involving lists.

Disadvantages

- LISP (another logic programming language) dominates over prolog with respect to I/O features.
- Sometimes input and output is not easy.

Applications

- Highly used in artificial intelligence(AI).
- Used for pattern matching over natural language parse trees.

Prolog Available

Several Prolog environments are available:

- SWI-Prolog
  `https://www.swi-prolog.org/`
  Installation in Linux :`sudo apt-get install swi-prolog`
- SICStus Prolog
- GNU Prolog
- YAP Prolog
- Strawberry Prolog

Facts, Rules, and Queries

- There are only three basic constructs in Prolog: facts, rules, and queries.
- A collection of facts and rules is called a Knowledge Base.
- Prolog programs simply are Knowledge Bases, collections of facts and rules which describe some collection of relationships.
- We can query against the Knowledge Base.
- We get output as affirmative if our query is already in the knowledge Base or it is implied by Knowledge Base, otherwise we get output as negative.

Prolog Facts

- Prolog facts are expressed in definite pattern.
- Facts contain entities and their relation.
  - Entities are written within the parenthesis separated by comma (,).
  - Their relation is expressed at the start and outside the parenthesis.
- Every fact/rule ends with a dot (.).

Prolog Syntax

- 4 kinds of term in Prolog: atoms, numbers, variables, and complex terms (or structures).
- Atoms and numbers are lumped together under the heading constants, and constants and variables together make up the simple terms of Prolog.
- Basic characters:
  - he upper-case letters
  - the lower-case letters are
  - the digits
  - the _ symbol, which is called underscore
  - some special characters , which include characters such as $+, -, *, /, <, >, =, :, ., \&,$.
  - The blank space is also a character, but a rather unusual one, being invisible.
  - A string is an unbroken sequence of characters.

A typical prolog fact

```
Format : relation(entity1, entity2, ....k'th entity).

Example :
friends(raju, mahesh).
singer(sonu).
odd_number(5).

Explanation :
These facts can be interpreted as :
raju and mahesh are friends.
sonu is a singer.
5 is an odd number.
```

Key Features: Unification, Backtracking and Recursion.

A typical prolog query

```
Query 1 : ?- singer(sonu).
Output : Yes.

Explanation : As our knowledge base contains
the above fact, so output was 'Yes', otherwise
it would have been 'No'.

Query 2 : ?- odd_number(7).
Output : No.

Explanation : As our knowledge base does not
contain the above fact, so output was 'No'.
```

Knowledge Base 1

Facts are used to state things that are unconditionally true of some situation of interest.

Example 1

Mia, Jody, and Yolanda are women, Jody plays air guitar, and a party is taking place.

- `woman(mia).`
- `woman(jody).`
- `woman(yolanda).`
- `playsAirGuitar(jody).`
- `party.`

`?- woman(mia)`: prolog will answer yes

`?- playsAirGuitar(mia)`: prolog will answer no

Knowledge Base 2

KB2 consists of 5 clauses:

Example 2

```
happy(vincent).
listens2Music(butch).
playsAirGuitar(vincent):- listens2Music(vincent),
happy(vincent).
playsAirGuitar(butch):- happy(butch).
playsAirGuitar(butch):- listens2Music(butch).
```

`?- playsAirGuitar(vincent)`: prolog will answer no

`?- playsAirGuitar(butch)`: prolog will answer yes

Knowledge Base 3: Eating

Example 3

```
is_digesting(X,Y) :- just_ate(X,Y).
is_digesting(X,Y) :- just_ate(X,Z), is_digesting(Z,Y).
just_ate(mosquito,blood(john)).
just_ate(frog,mosquito).
just_ate(stork,frog).
```

```
?- is_digesting(stork,mosquito).
```

Prolog goes to work as follows:

- it tries to make use of the first rule listed concerning is_digesting.
- This tells it that X is digesting Y if X just ate Y , By unifying X with stork and Y with mosquito it obtains the following goal: `just_ate(stork,mosquito).`

Family Problem

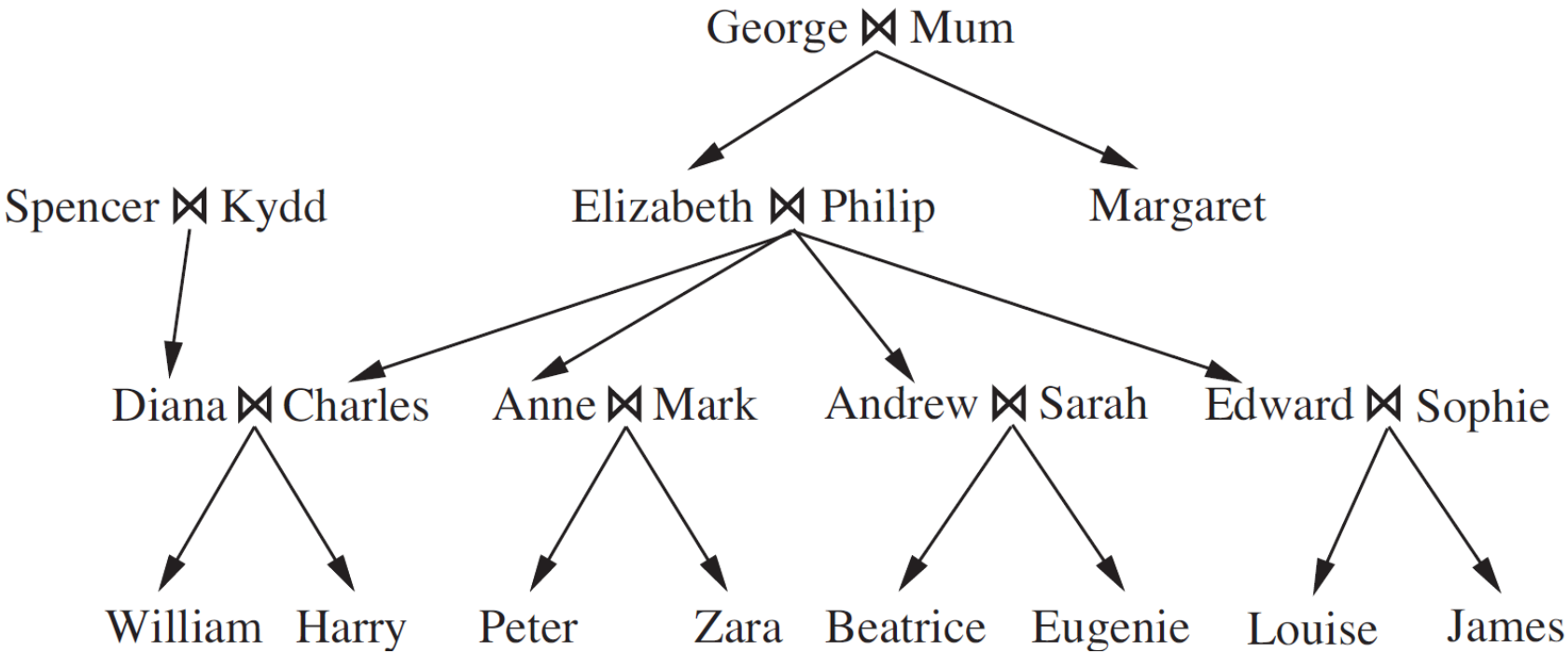

Figure 1: A typical family tree. The symbol ⋈ connects spouses and arrows point to children.

Tasks

Please fulfill the following tasks by using `Prolog`:

- Write sentences describing the predicates **Grandchild**, **Greatgrandparent**, **Ancestor**, **Brother**, **Sister**, **Daughter**, **Son**, **FirstCousin**, **BrotherInLaw**, **SisterInLaw**, **Aunt**, and **Uncle**.
  Hint: you can define these predicates by choosing child, sibling, male, female, father, mother, and so on.
- Find out the proper definition of ***m*th cousin *n* times removed**, in other words, define the predicate `mthCousinNremoved(X,Y,M,N)`.
  Hint: You'd better define the predicate `distance(X,Y,N)` by recursion (please refer to `hanoi.pl`) to show there are N generations between X and Y in advance.

Tasks (cont'd)

- Write down the basic facts depicted in the family tree in Figure 1.
- ASK it who are
  - **Elizabeth's grandchildren**,
  - **Diana's brothers-in-law**,
  - **Zara's great-grandparents**, and
  - **Eugenie's ancestors**.

About Cousin and Removed

- **Cousin (a.k.a "frst cousin")**:
  Your first cousins are the people in your family who have two of the same grandparents as you.
- **Second Cousin**:
  Your second cousins are the people in your family who have the same great-grandparents as you.,but not the same grandparents.
- **Third, Fourth, and Fifth Cousins**:
  Your third cousins have the same great great grandparents, fourth cousins have the same great-great-great-grandparents, and so on.

About Cousin and Removed (cont'd)

- "Removed" indicates that the two people are from different generations. You and your first cousins are in the same generation (two generations younger than your grandparents).
- "Once removed" means that there is a difference of one generation. For example, your mother's first cousin is your first cousin, once removed.
- "Twice removed" means that there is a two-generation difference. For example, you and your grandmother's first cousin are first cousins, twice removed.

Hints

Hint1:

Predicates

$$Grandchild(c,a) \Leftrightarrow \exists b\ Child(c,b) \wedge Child(b,a)$$
$$Greatgrandparent(a,d) \Leftrightarrow \exists b,c\ Child(d,c) \wedge Child(c,b) \wedge Child(b,a)$$
$$Ancestor(a,x) \Leftrightarrow Child(x,a) \vee \exists b\ Child(b,a) \wedge Ancestor(b,x)$$
$$Brother(x,y) \Leftrightarrow Male(x) \wedge Sibling(x,y)$$
$$Sister(x,y) \Leftrightarrow Female(x) \wedge Sibling(x,y)$$
$$Daughter(d,p) \Leftrightarrow Female(d) \wedge Child(d,p)$$
$$Son(s,p) \Leftrightarrow Male(s) \wedge Child(s,p)$$
$$FirstCousin(c,d) \Leftrightarrow \exists p_1,p_2\ Child(c,p_1) \wedge Child(d,p_2) \wedge Sibling(p_1,p_2)$$
$$BrotherInLaw(b,x) \Leftrightarrow \exists m\ Spouse(x,m) \wedge Brother(b,m)$$
$$SisterInLaw(s,x) \Leftrightarrow \exists m\ Spouse(x,m) \wedge Sister(s,m)$$
$$Aunt(a,c) \Leftrightarrow \exists p\ Child(c,p) \wedge [Sister(a,p) \vee SisterInLaw(a,p)]$$
$$Uncle(u,c) \Leftrightarrow \exists p\ Child(c,p) \wedge [Brother(a,p) \vee BrotherInLaw(a,p)]$$

Hints (cont'd)

Hint2:

Define mth cousin n times removed

Define $Distance(c, a)$ as follows:
$Distance(c, c) = 0$.
$Child(c, b) \wedge Distance(b, a) = k \Rightarrow Distance(c, a) = k + 1$.

The distance to one's grandparent is 2, great-great-grandparent is 4, and so on. Now we have:
$MthCousinNTimesRemoved(c, dm, n) \Leftrightarrow \exists a.Distance(c, a) = m + 1 \wedge Distance(d, a) = m + n + 1$.

Reference

- Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach.
- Prolog, https://en.wikipedia.org/wiki/Prolog
- Prolog programming: a do-it-yourself course for beginners, http://cs.union.edu/~striegnk/courses/esslli04prolog/index.php
- Learn Prolog Now! http://learnprolognow.org/
- https://www.swi-prolog.org/
- https://www.geeksforgeeks.org/prolog-an-introduction/

The End