Logic Programming

Outline

- Pemrograman logika
- FOL
- Logika Predikat
- Horn clause
- Prolog

What is Logic Programming (LP)?

"Say what you want, not how you want it done"

- Programming paradigm which has its foundations in mathematical logic.
- Expresses the logic of a computation without describing its control flow→declarative
- Composed of a set of axioms, or rules which define relationships between objects, not sequences of instructions.

First Order Logic (FOL)

FOL : Symbolized reasoning in which each sentence, or statement, is broken down into a subject and a predicate (relation).

Terms

- a constant: single individual or concept
- a variable that stands for different individuals

Predicates/Relations

- a relation that maps n terms to a truth value true (T) or false (F).
- a function: a mapping that maps n terms to a term

Quantifiers

- Variables are used in conjunction with quantifiers
- “there exist”(∃) and “for all”(∀)

Connectives

- ¬, →, ↔, Λ, V

Sentence in FOL

- An atomic sentence is simply a predicate applied to a set of terms.
  - Owns(John,Car1)
- The standard propositional connectives ($\vee$ ¬ $\wedge$ $\Rightarrow$ $\Leftrightarrow$) can be used to construct complex sentences:
  - Owns(John,Car1) $\vee$ Owns(Fred, Car1)
- Universal quantifier: ∀x
  Asserts that a sentence is true for all values of variable x

  $\forall x$ [Crow (x) → Black (x)]   All crows are black
- Existential quantifier: ∃
  Asserts that a sentence is true for at least one value of a variable x.

  $\exists x$ [Squirrel $(x) \wedge$ Black (x)]   Some squirrels are black

Common Mistakes to Avoid

$$\forall x \ \text{human}(x) \Rightarrow \text{mammal}(x)$$

- It's a big AND: Equivalent to the conjunction of all the instantiations of variable x:

$$(\text{human}(\text{Jerry}) \Rightarrow \text{mammal}(\text{Jerry})) \land$$
$$(\text{human}(\text{Jing}) \Rightarrow \text{mammal}(\text{Jing})) \land$$
$$(\text{human}(\text{laptop}) \Rightarrow \text{mammal}(\text{laptop})) \land \ldots$$

- Common mistake is to use $\land$ as main connective

$$\forall x \ \text{human}(x) \land \text{mammal}(x)$$

- This means everything is human and a mammal!

$$(\text{human}(\text{Jerry}) \land \text{mammal}(\text{Jerry})) \land$$
$$(\text{human}(\text{Jing}) \land \text{mammal}(\text{Jing})) \land$$
$$(\text{human}(\text{laptop}) \land \text{mammal}(\text{laptop})) \land \ldots$$

Common Mistakes to Avoid

$$\exists x \ \mathtt{human(x)} \wedge \mathtt{male(x)}$$

- It's a big OR: Equivalent to the disjunction of all the instantiations of variable x:

$$(\mathtt{human(Jerry)} \wedge \mathtt{male(Jerry)}) \vee$$
$$(\mathtt{human(Jing)} \wedge \mathtt{male(Jing)}) \vee$$
$$(\mathtt{human(laptop)} \wedge \mathtt{male(laptop)}) \vee \ldots$$

- Common mistake is to use $\Rightarrow$ as main connective
  - "Some pig can fly"

$$\exists x \ \mathtt{pig(x)} \Rightarrow \mathtt{fly(x)} \ \mathtt{(wrong)}$$

- This is true if there is something not a pig!

$$(\mathtt{pig(Jerry)} \Rightarrow \mathtt{fly(Jerry)}) \vee$$
$$(\mathtt{pig(laptop)} \Rightarrow \mathtt{fly(laptop)}) \vee \ldots$$

Nesting Quantifiers

- The order of quantifiers of the same type doesn't matter

  $\forall x \forall y(Parent(x,y) \land Male(y) \Rightarrow Son(y,x))$

  $\exists x \exists y(Loves(x,y) \land Loves(y,x))$

- The order of mixed quantifiers does matter:

  $\forall x \exists y(Loves(x,y)) \rightarrow$ everybody loves somebody,

  $\exists y \forall x(Loves(x,y)) \rightarrow$ there is someone who is loved by everyone

  $\forall y \exists x(Loves(x,y)) \rightarrow$ everyone has someone who loves them.

  $\exists x \forall y(Loves(x,y)) \rightarrow$ there is someone who loves everyone in the universe

Relation Between Quantifiers

- General Identities

$\forall x\ \neg P \Leftrightarrow \neg \exists x\ P$

$\neg \forall x\ P \Leftrightarrow \exists x\ \neg P$

$\forall x\ P \Leftrightarrow \neg \exists x\ \neg P$

$\exists x\ P \Leftrightarrow \neg \forall x\ \neg P$

Equality

- Identity relation
- Mary owns two cats. Inequality needed to insure x and y are distinct.
- $\exists x \exists y(Owns(Mary, x) \wedge Cat(x) \wedge Owns(Mary,y) \wedge Cat(y) \wedge \neg(x=y))$

Translate Sentence to FOL

"All *P*s are *Q*s."	"Some *P*s are *Q*s."
$\forall x.\ (P(x) \to Q(x))$	$\exists x.\ (P(x) \land Q(x))$
"No *P*s are *Q*s."	"Some *P*s aren't *Q*s."
$\forall x.\ (P(x) \to \neg Q(x))$	$\exists x.\ (P(x) \land \neg Q(x))$

Identifies objects, variables, predicates, function, in the sentence

Translate Sentence to FOL

Every orange cat is fluffy.

Variable : 1 variable
Predicates : orange, cat, fluffy
Quantifier : ∀

"All *P*s are *Q*s."

$$\forall x.\ (P(x) \rightarrow Q(x))$$

$$\forall x.\ (\textit{x is an orange cat} \rightarrow \textit{x is fluffy})$$

$$\forall x.\ (\textit{x is an orange cat} \rightarrow \boxed{\textit{x is fluffy})}$$

$$\forall x.\ (\textit{x is an orange cat} \rightarrow \mathit{Fluffy}(x))$$

$$\forall x.\ (\mathit{Orange}(\mathit{Cat}(x)) \rightarrow \mathit{Fluffy}(x))$$

X

$$\forall x.\ (\textit{x is orange and x is a cat} \rightarrow \mathit{Fluffy}(x))$$

$$\forall x.\ (\mathit{Orange}(x) \land \mathit{Cat}(x) \rightarrow \mathit{Fluffy}(x))$$

Translate Sentence to FOL

There's a corgi that loves everyone.

"Some *P*s are *Q*s."

$\exists x.\ (P(x) \land Q(x))$

Variable : 2 variables
Predicates : corgi, person, loves
Quantifier : ∃, ∀

$\exists x.\ (\textit{x is a corgi} \land \textit{x loves everyone})$

$\exists x.\ (Corgi(x) \land \textit{x loves everyone})$

$\exists x.\ (Corgi(x) \land \textit{x loves every person y})$

$\exists x.\ (Corgi(x) \land \textit{every person y is loved by x})$

"All *P*s are *Q*s."

$\forall x.\ (P(x) \rightarrow Q(x))$

$\exists x.\ (Corgi(x) \land \forall y.\ (\textit{y is a person} \rightarrow \textit{y is loved by x}))$

$\exists x.\ (Corgi(x) \land \forall y.\ (Person(y) \rightarrow Loves(x, y)))$

Translate Sentence to FOL

Everybody loves at least one corgi.

"All *P*s are *Q*s."

$\forall x.\ (P(x) \rightarrow Q(x))$

Variable : 2 variables
Predicates : corgi, person, loves
Quantifier : ∃, ∀

∀x. (*x is a person* → *x loves at least one corgi*)

$\forall x.\ (Person(x) \rightarrow$ *x loves at least one corgi*)

$\forall x.\ (Person(x) \rightarrow$ *x loves at least one corgi y*)

$\forall x.\ (Person(x) \rightarrow$ *there is a corgi y that is loved by x*)

"Some *P*s are *Q*s."

$\exists x.\ (P(x) \land Q(x))$

$\forall x.\ (Person(x) \rightarrow \exists y.\ ($*y is a corgi* ∧ *y is loved by x*))

$$\forall x.\ (Person(x) \rightarrow \exists y.\ (Corgi(y) \land Loves(x, y)))$$

Horn Clause

- Perhatikan pernyataan: **if (**$P1 \wedge P2 \wedge \cdots \wedge Pn$**) then Q**.
- Dapat juga ditulis sebagai: $Q \leftarrow (P1 \wedge P2 \wedge \cdots \wedge Pn)$
  dibaca *Q* hanya jika *P*1 dan *P*2 dan …dan *Pn*
- Pernyataan Q akan benar (True) jika semua pernyataan *P*1, *P*2,…, *Pn* secara simultan benar.
- Ingat: $A \rightarrow B$ setara (memiliki nilai kebenaran yang sama) dengan : $\neg A \vee B$.
- Oleh karena itu, pernyataan implikasi tadi dapat dinyatakan dalam bentuk *disjunctive normal* sebagai:
  $Q \vee \neg P1 \vee \neg P2 \cdots \vee \neg Pn$
- Ekspresi ini disebut Horn Clause

Horn Clause

- Prolog:

```
c:- a, b.
a.
b.
```

- Horn formula:

$[c \lor \neg a \lor \neg b] \land a \land b$

$[c, \neg a, \neg b] \quad [a] \quad [b]$

Proving Horn Clause

- Prolog is based on computing with Horn clauses
- Let us attempt to prove c by contradiction
- Hence, the goal clause is ¬c

[c,¬a,¬ b] [a] [b] [¬c]

[c,¬a,¬b] [a] [b] [¬c]

[c,¬b] [b] [¬c]

[c] [¬c]

[]

Resolution Inference Rule

$(A \vee B \vee C)$

$(\neg A)$

- - - - - - - - - - - - - -

$\therefore (B \vee C)$

Prolog

- Conversion FOL to Prolog
  - $(\forall x)\ (\forall y)\ (\forall z)\ (father(x,z) \wedge parent(z,y) \rightarrow grandfather(x,y))$
    grandfather(X,Y) :- father(X,Z), parent(Z,Y)
- **Atoms**: string karakter yang dimulai dengan huruf kecil, menyatakan nama dari obyek atau relasi. Misal: joko, ibu, lelaki
- **Term**:
  - **Constants**: atoms
  - **Variables**: string karakter yang dimulai dengan huruf besar. Misal: X
  - **Structure**: predikat dengan nama dan sejumlah argumen yang telah *fixed*. Misal: ibu(shinta,X).

Rules & Facts

- Technically, we don't create a Prolog program; we create a Prolog database.
- In the database are two types of clauses: Facts and Rules.

- **Facts** (fakta): single piece of information
  - blue(sky).
  - mammal(rabbit).
  - plays(john, hockey).
- **Rules** (aturan): logic expression "hanya jika", Horn clauses.
  - generate new information from facts, other rules, and even themselves
  - Rules have the form head :- body.
  - the head and the body are clauses that typically use variables instead of constants (because rules are used to define general relationships).
  - Contoh:
    - habisDibagiDua(X) :- genap(X).
  - Pernyataan tersebut setara dengan pernyataan kalkulus predikat:
    $(\forall x)\ (genap(x) \rightarrow habisDibagiDua(x))$

Conjunction & Disjunction

- *Conjunction* dari predikat direpresentasikan sebagai deretan struktur yang dipisahkan oleh tanda koma (,).

- *Disjunction* dari predikat direpresentasikan sebagai deretan struktur yang dipisahkan oleh tanda titik koma (;).
  - sibling(X,Y):- kakak(X,Y) ; adik(X,Y).
    sibling(X,Y):- kakak(X,Y).
    sibling(X,Y):- adik(X,Y).

- *Negation* predikat untuk negasi adalah not.
  - ganjil(X):- not genap(X).

How Prolog Works

- A Prolog program is initiated by a **query** - a predicate or a sequence of predicates to be proved.
- The predicate to be proved is called a **goal**.
- Prolog tries to **match** the goal to the head of some fact or some rule.
- If a match is found with a rule, Prolog continues with proving each predicate in the body of the rule.

Example

```
man(adam).
man(peter).
man(paul).
woman(marry).
woman(eve).
parent(adam,peter).     % means adam is parent of peter
parent(eve,peter).
parent(adam,paul).
parent(marry,paul).
```

Facts

```
father(F,C):-man(F),parent(F,C).
mother(M,C):-woman(M),parent(M,C).
is_father(F):-father(F,_).
is_mother(M):-mother(M,_).
```

Rules

```
?-father(X,paul).
```

Query

X=Adam

Example 2

X grandmother of Z

```
mother(pam,bob).
mother(pat,jim).
father(tom,bob).
father(tom,liz).
father(bob, ann).
father(bob,pat).
grandmother(X,Z) :- mother(X,Y), mother(Y,Z).
grandmother(X,Z) :- mother(X,Y), father(Y,Z).
grandfather(X,Z) :- father(X,Y), mother(Y,Z).
grandfather(X,Z):-father(X,Y), father(Y,Z)
```

Example 3

X professors Y

```
studies(charlie, csc135).
studies(olivia, csc135).
studies(jack, csc131).
studies(arthur, csc134).

teaches(kirke, csc135).
teaches(collins, csc131).
teaches(collins, csc171).
teaches(juniper, csc134).

professor(X, Y) :- teaches(X, C), studies(Y, C).
```

Exercise 1

- Any two pancakes x and y taste similar
- Everyone knows at least two people

Exercise 2

- Buatlah klausa Prolog untuk persoalan berikut:
  1. Hewan buas adalah hewan yang berwarna gelap, berbadan besar, dan gigi bertaring.
  2. Hewan jinak adalah hewan yang berwarna terang, gigi tak bertaring dan berbadan kecil.
  3. Faktorial dari 0 adalah 1.
  4. Faktorial dari suatu bilangan bulat positif.

     ```
     factorial(0) => 1
     factorial(n) => factorial(n-1) * n;
     ```

  5. Fungsi *f* didefinisikan sbb:

$$f(x,y) = \begin{cases} x & , x > y \\ y & , \text{selainnya} \end{cases}$$