



Chương 5

QUI TRÌNH VÀ CÔNG CỤ XÂY DỰNG

HỆ HỖ TRỢ RA QUYẾT ĐỊNH

DSS Implementation & Tools

- **TS. Lê Đức Nhường**
- Khoa Công nghệ thông tin - Trường Đại học Hải Phòng
- Phone: (+84)987394900. Email: Nhuongld@dhhp.edu.vn
- Website: www.dhhp.edu.vn/~nhuongld



Nội dung Chương 5

1. Qui trình xây dựng hệ hỗ trợ ra quyết định

2 Ngôn ngữ lập trình Prolog

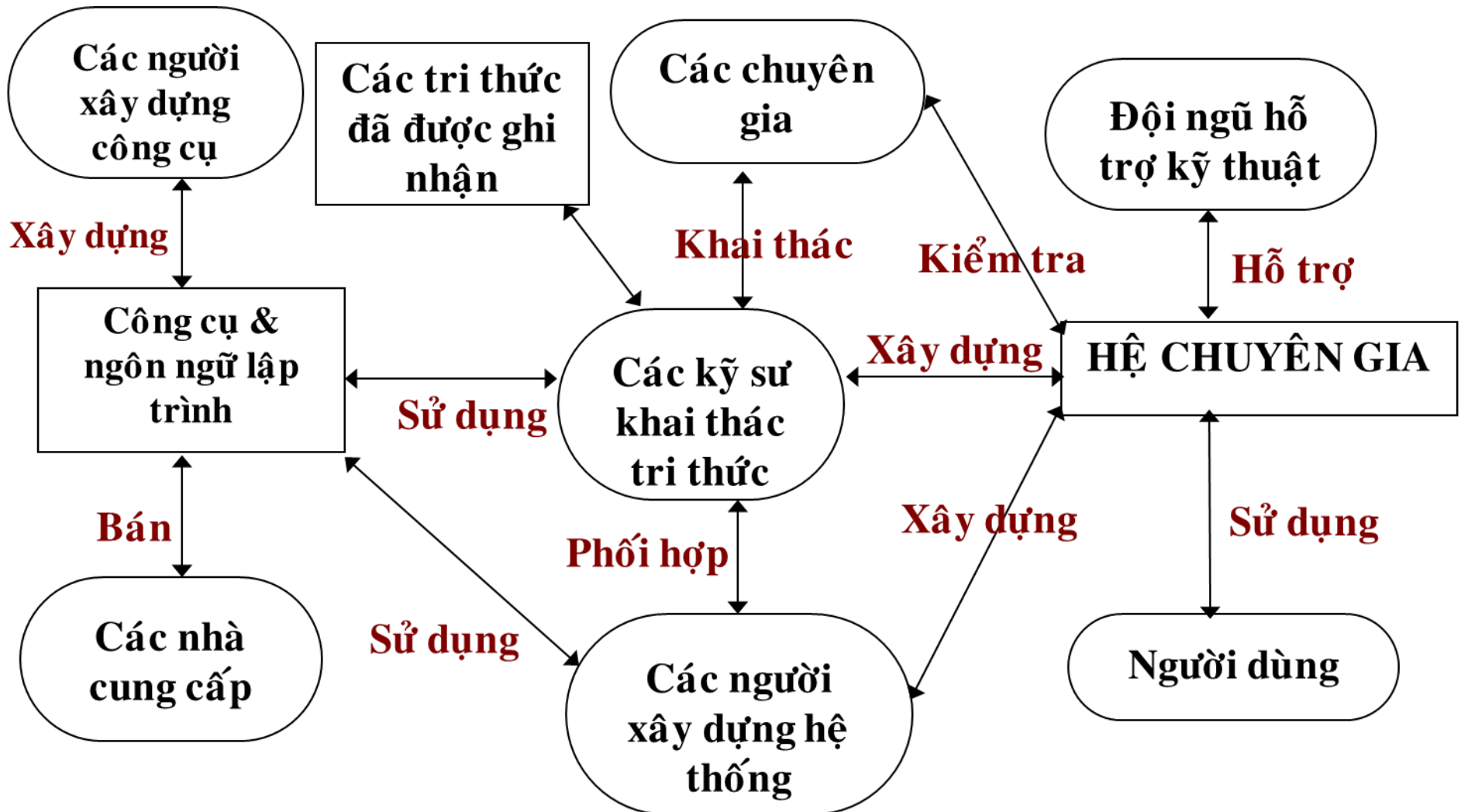
3. Ngôn ngữ lập trình LISP

4. Ngôn ngữ lập trình thông thường

5. Một số công cụ khác

1. Quy trình xây dựng hệ hỗ trợ ra quyết định

■ Quá trình xây dựng hệ hỗ trợ ra quyết định

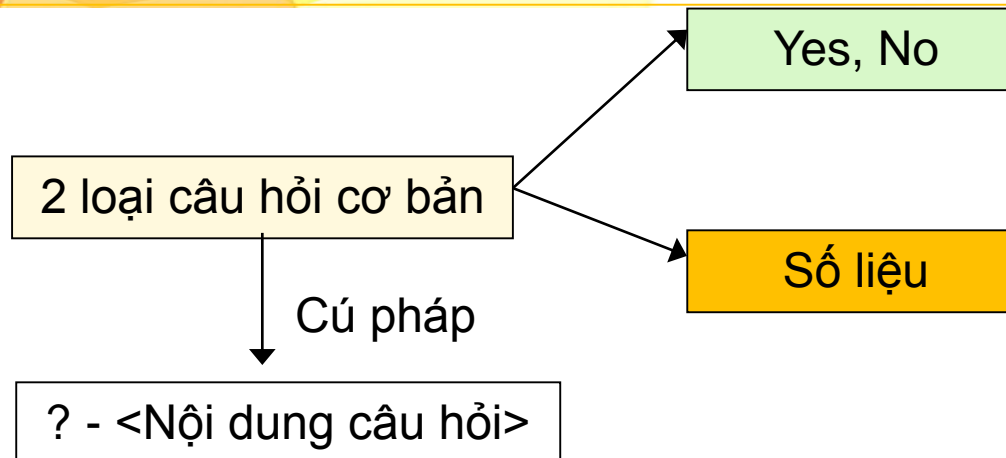


1. Quy trình xây dựng hệ hỗ trợ ra quyết định

Một số bước cơ bản để xây dựng hệ hỗ trợ ra quyết định

- Tiếp cận chuyên gia
- Tổ chức thu thập tri thức
- Chọn lựa công cụ phát triển hệ cơ sở tri thức
 - Chọn ngôn ngữ lập trình trí tuệ nhân tạo (LISP, PROLOG, ...)
 - Các ngôn ngữ lập trình thông dụng (C#, Matlab, Python,...)
 - Các hệ cơ sở tri thức rỗng (shell): là một công cụ lai giữa hai loại trên
- Cài đặt hệ hỗ trợ ra quyết định.

Truy vấn cơ sở tri thức



Ví dụ 1: Quả chanh có màu xanh là đúng hay sai ?

→ ?- Xanh(Chanh)

Ví dụ 2: Nếu ta có khai báo hai vị từ là : Yeu(An, Binh), Yeu(An, Chau)

An yêu ai ? → Yeu(An, X)

Hệ thống sẽ trả lời là :

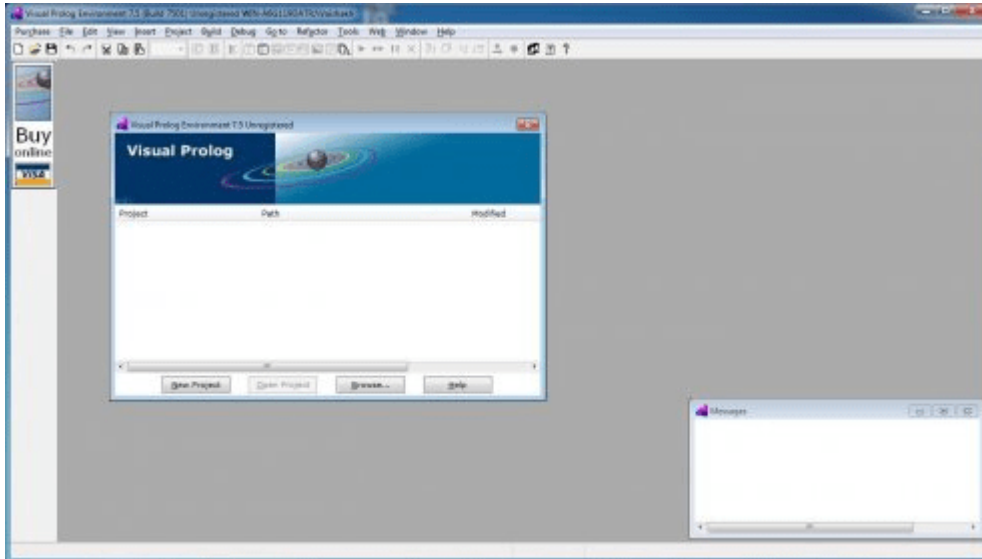
X → Binh

X → Chau

2 Solution(s)

2 Ngôn ngữ lập trình Prolog

- Prolog (**PRO**gramming in **LOG**ic) là ngôn ngữ lập trình dạng khai báo



<https://visual-prolog.software.informer.com/7.1/>



```
D:\VP_Vip6\demo\pie\Exe\FILE0.PRO
parent(person("Bill", "male"), person("John", "male")).
parent(person("Pam", "female"), person("Bill", "male")).
parent(person("Pam", "female"), person("Jane", "female")).
parent(person("Jane", "female"), person("Joe", "male")).

grandFather(Person, TheGrandFather) :-
    parent(Person, ParentOfPerson),
    father(ParentOfPerson, TheGrandFather).

father(P, person(Name, "male")) :-
    parent(P, person(Name, "male")).
```

```
Dialog
grandFather(person("Pam", "female"), W)
grandFather(person("Pam", "female"), W)
W = person("John", "male").
W = person("Joe", "male").
2 Solutions
```



2 Ngôn ngữ lập trình Prolog

- Cơ sở tri thức của Prolog bao gồm các vị từ mô tả khái niệm.

- Sự kiện:

Cú pháp: <thuộc tính>(<đối tượng>)

Quả chanh có màu xanh → Xanh(Chanh)

- Mối liên hệ giữa các đối tượng

Cú pháp: <quan hệ> (<đối tượng 1>, ..., <đối tượng n>)

An yêu Bình → Yêu(An, Bình)

2 Ngôn ngữ lập trình Prolog

- Cấu trúc giữa các đối tượng

Cú pháp: <đối tượng> (<thành phần 1>, ..., <thành phần n>)

Ví dụ: Xe máy hiệu Dream, 110 phân khối, màu nâu, 4 số, giá 30 triệu.

→ Xe máy(Dream, 110, nâu, 4, 30)

- Các luật

Cú pháp: <luật>(<đối tượng 1>, ..., <đối tượng n>) :- <điều kiện 1>, ..., <điều kiện n>

Ví dụ: A là chim nếu A có cánh và A biết bay

→ Chim(A) :- CóCánh(A), BiếtBay(A).

Dùng dấu phẩy (,) để biểu diễn toán tử AND, dấu chấm phẩy (;) để biểu diễn toán tử OR và toán tử không bằng là \=



2 Ngôn ngữ lập trình Prolog

- Ví dụ:

- A là tổ tiên của B nếu:

A là cha mẹ của B (phần kết thúc)

A là cha mẹ của C và C là tổ tiên của B.

Ta định nghĩa luật như sau :

$\text{ToTien}(A,B) \text{ :- } \text{ChaMe}(A,B).$

$\text{ToTien}(A,B) \text{ :- } \text{ChaMe}(A,C), \text{ToTien}(C,B).$

2 Ngôn ngữ lập trình Prolog

■ Ví dụ:

Prolog Search Tree

```
/* program P clause # */
```

```
p(a). /* #1 */
```

```
p(X) :- q(X), r(X). /* #2 */
```

```
p(X) :- u(X). /* #3 */
```

```
q(X) :- s(X). /* #4 */
```

```
r(a). /* #5 */
```

```
r(b). /* #6 */
```

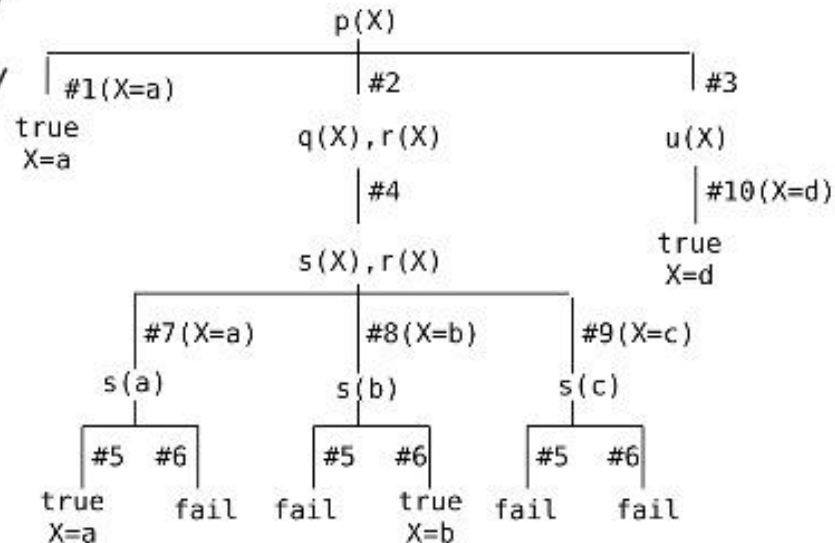
```
s(a). /* #7 */
```

```
s(b). /* #8 */
```

```
s(c). /* #9 */
```

```
u(d). /* #10 */
```

?- p(X)



2 Ngôn ngữ lập trình Prolog

- Ví dụ: Hệ hỗ trợ ra quyết định về tình trạng gia đình trên Prolog
 - Bước 1: Mô tả các sự kiện trong quan hệ gia đình

married(philip, elizabeth).

married(mark, anne).

married(charles, diana).

married(tim, anne).

divorced(mark, anne).

parents(philip, elizabeth, charles).

parents(philip, elizabeth, anne).

parents(philip, elizabeth, andrew).

parents(philip, elizabeth, edward).

parents(mark, anne, zara).

parents(mark, anne, peter).

parents(charles, diana, william).

parents(charles, diana, harry).

parents(andrew, sarah, eugene).

parents(andrew, sarah, beatrice).

gender(philip, male).

gender(elizabeth, female).

gender(charles, male).

gender(andrew, male).

gender(edward, male).

gender(mark, male).

gender(tim, male).

gender(diana, female).

gender(sarah, female).

gender(zarah, female).

gender(peter, male).

gender(william, male).

gender(harry, male).

gender(eugene, female).

gender(beatrice, female).

2 Ngôn ngữ lập trình Prolog

- Hệ hỗ trợ ra quyết định về tình trạng gia đình trên Prolog
 - Bước 2: Định nghĩa các quan hệ gia đình khác dựa trên các sự kiện đã nêu

```
parent_of(Parent, Child) :- parents(Parent, _, Child).
```

```
parent_of(Parent, Child) :- parents(_, Parent, Child).
```

```
sibling_of(S, Person) :- parents(F, M, S), parents(F, M, Person), S\=Person.
```

```
father_of(Father, Child) :- parents(Father, _, Child).
```

```
sister_of(S, Person) :- gender(S, female), sibling_of(S, Person).
```

```
grand_parent_of(GP, GC) :- parent_of(GP, Temp), parent_of(Temp, GC).
```

```
ancesstor_of(Ancesor, Person) :- parent_of(Ancesor, Person).
```

```
ancesstor_of(Ancesor, Person) :-
```

```
    ancesstor_of(Ancesor, Temp), parent_of(Temp, Person).
```

```
cousin_of(A, B) :- parent_of(P1, A), parent_of(P2, B), sibling_of(P1, P2).
```

2 Ngôn ngữ lập trình Prolog

■ Bước 3: Suy luận

- Muốn đặt ra câu hỏi "**Ai là chị của andrew**" và bật chức năng **TRACE** theo quá trình suy luận của PROLOG ta sẽ có hiển thị các thông tin sau:

?- sister_of(S, andrew)

- Đầu tiên, hệ thống sẽ tìm giá trị S thỏa điều kiện **gender(S, female)**. Quá trình tìm kiếm sẽ dừng lại ở sự kiện **gender(elizabeth, female)**.

CALL gender(S, female) ... succeeds; S → elizabeth

- Do đó, sự kiện **sibling_of(elizabeth, andrew)** được đánh giá là sai.

FAIL sibling_of(elizabeth, andrew)

2 Ngôn ngữ lập trình Prolog

■ Bước 3: Suy luận

- Muốn tìm một giá trị S khác thỏa điều kiện **gender(S, female)**. Quá trình tìm kiếm sẽ dừng lại ở sự kiện **gender(anne, female)**.

REDO gender(S, female) ... succeeds; S → anne

- Muốn tìm A, B thỏa điều kiện tiếp theo là **parents(A, B, elizabeth)**. Quá trình tìm kiếm sẽ dừng lại ở sự kiện **parents(philip, elizabeth, anne)**

CALL sibling_of(anne, andrew)

CALL parents(A, B, anne)

... succeeds; A → philip, B → elizabeth

- Hệ thống kiểm tra điều kiện cuối cùng **S \= andrew**

CALL anne \= andrew ... succeeds



2 Ngôn ngữ lập trình Prolog

▪ Bước 3: Suy luận

- Như vậy là vị từ `subling_of(anne, andrew)` có giá trị đúng.

EXIT subling_of(anne, andrew)

- Kết luận là anne là chị của andrew.

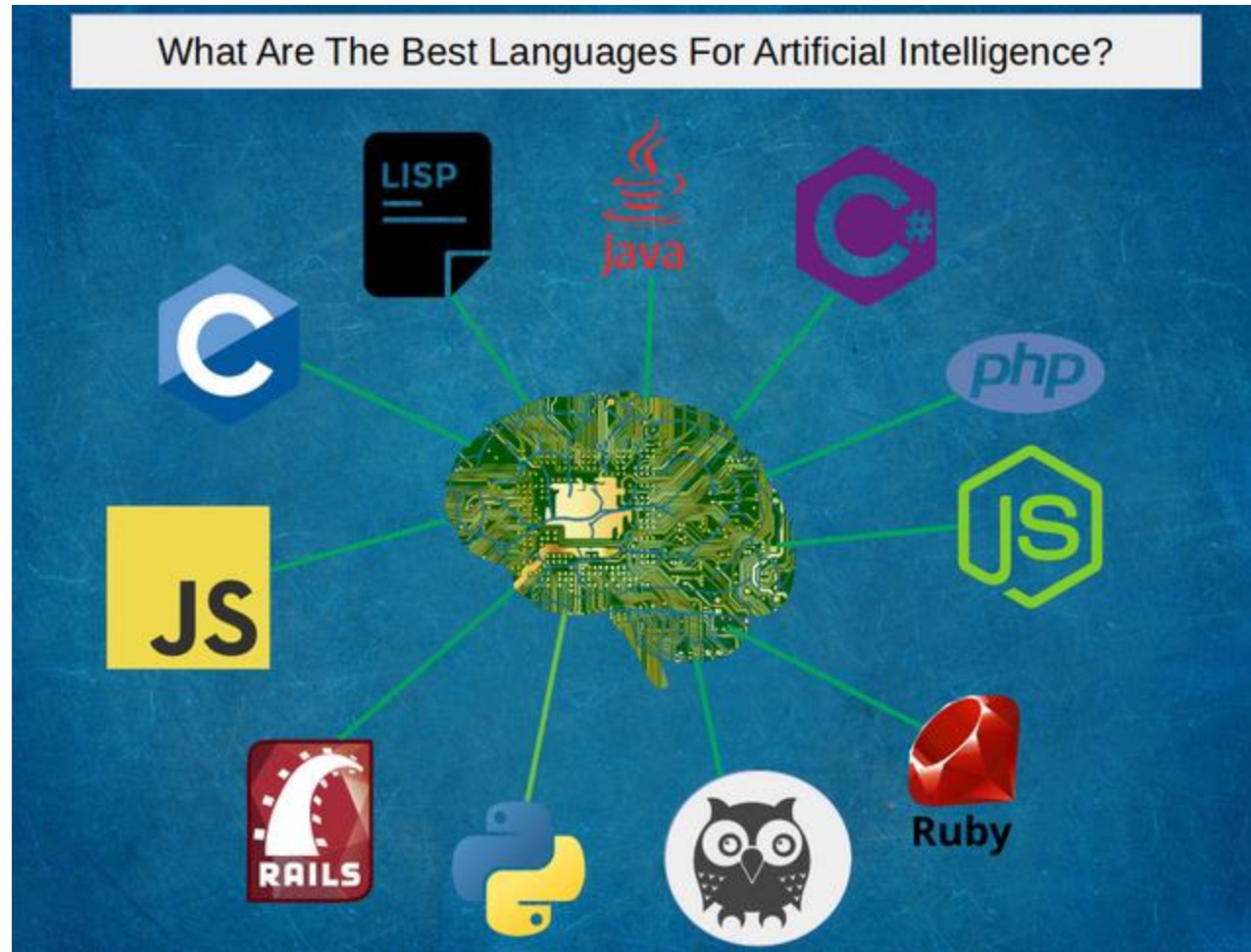
EXIT sister_of(anne, andrew)




3. Ngôn ngữ lập trình LISP

- Lisp – List Processing language
- Lisp là hiện thân của 9 ý tưởng mới:
 - 1. Conditionals (*Điều kiện*)
 - 2. A function type (*Kiểu hàm*)
 - 3. Recursion (*Đệ quy*)
 - 4. A new concept of variables (*Khái niệm mới về biến*)
 - 5. Garbage-collection (*Thu dọn rác (tự động quản lí bộ nhớ)*).
 - 6. Programs composed of expressions (*Chương trình được kết hợp bởi các biểu thức*).
 - 7. A symbol type (*Kiểu ký hiệu*).
 - 8. A notation for code (*Cách viết mã sử dụng cây kí hiệu*).
 - 9. The whole language always available (*Ngôn ngữ có thể sử dụng mọi lúc mọi nơi*).

4. Ngôn ngữ lập trình thông thường



4. Ngôn ngữ lập trình thông thường

Prolog

Great For Prototyping


Easy Rule Implementation

Very Old Language

Logic-based

Tree-based


Powerful



Hello World Example in Prolog

```
write('Hello, World!'), nl.
```

This outputs the text Hello, World!
to the screen.

Lisp


Fast

Few Developers

Garbage Collection

Symbolic Programming

Fast Prototyping




Hello World Example in Lisp

```
(print "Hello, World!")
```

This outputs the text Hello, World!
to the screen.

4. Ngôn ngữ lập trình thông thường


 C++

Very Fast

Powerful

Difficult To Learn

Multi-paradigm Language




Hello World Example in C++

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello, World!";
    return 0;
}
```

This outputs the text Hello, World! to the screen.

 Java


Strong Libraries

Simple

Object Oriented

Slower Than C++

Portable & Scalable



Hello World Example in Java

```
public class Main {


    public static void main(String[] args) {

        System.out.println("Hello, World!");

    }

}
```

This outputs the text Hello, World! to the screen.

 Python


Easy To Learn

Powerful

Tons of Libraries

Fast Testing

Open Source




Hello World Example in Python

```
print("Hello, World!")
```

This outputs the text Hello, World! to the screen.

Both Languages are amazing, but which one looks easier to learn? Hint...



Hello World Example in Java

```
public class Main {

    public static void main(String[] args) {

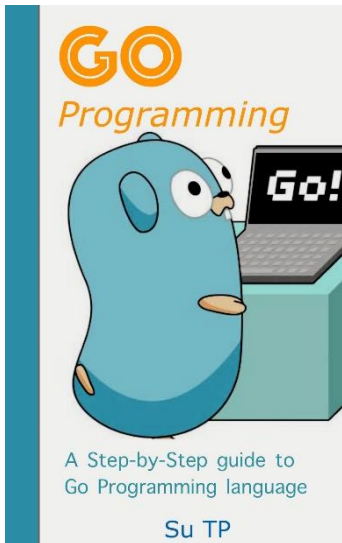
        System.out.println("Hello, World!");

    }

}
```

This outputs the text Hello, World! to the screen.

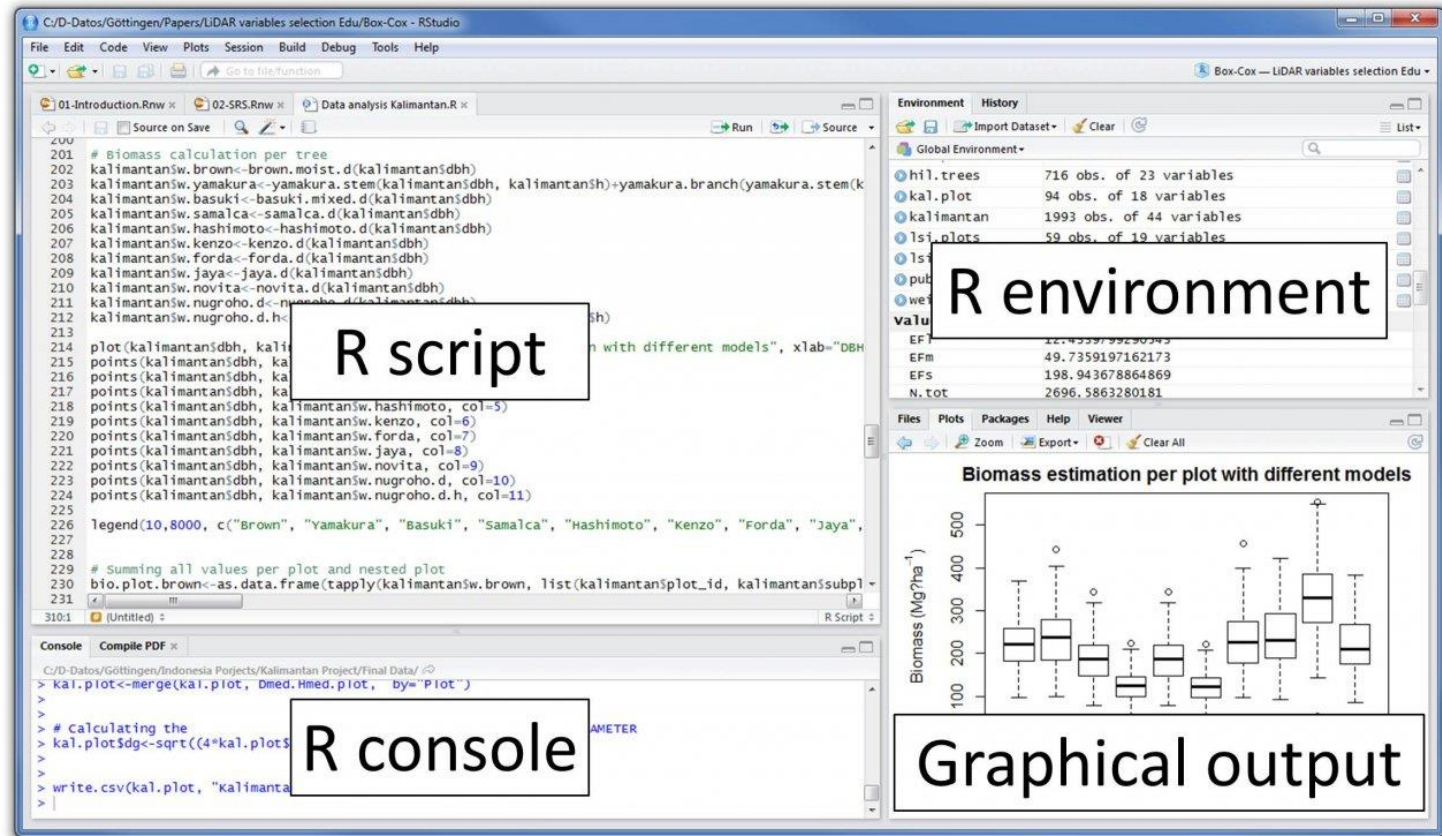
4. Ngôn ngữ lập trình thông thường



```
0 package main
1
2
3 import "fmt"
4
5 // Book to study go lang
6 type Book struct {
7     title string
8     author string
9 }
10
11 func (b Book) getTitleAndAuthor() string {
12     return fmt.Sprintf("%s by %s to read", b.title, b.author)
13 }
14
15 //WithTitleAndAuthor is something that has a title and author
16 type WithTitleAndAuthor interface {
17     getTitleAndAuthor() string
18 }
19
20 func main() {
21     fmt.Println("Welcome to go books.")
22 }
```

<https://golang.org/dl/>

4. Ngôn ngữ lập trình thông thường

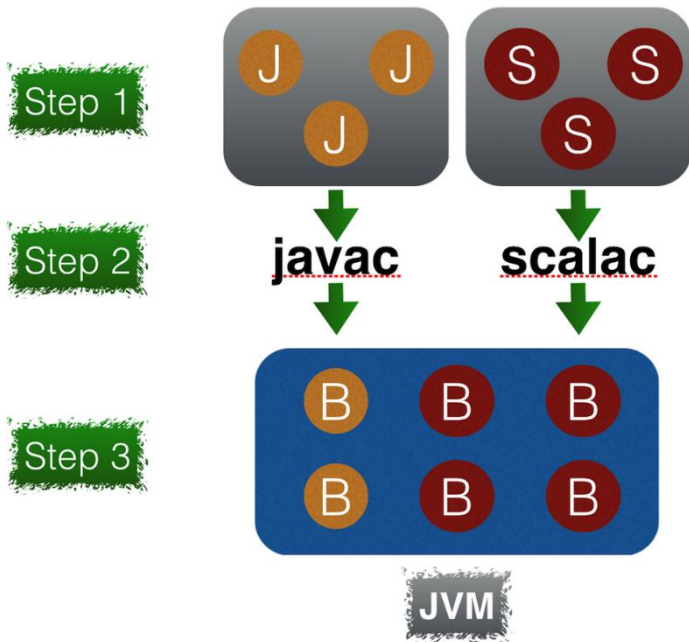


The screenshot displays the RStudio environment with three main panels:

- R script:** The editor window shows R code for biomass calculation and plotting. The code includes data loading, variable assignment, and plotting functions. A label "R script" is overlaid on the code.
- R console:** The console window shows the execution of R commands, including data merging and plotting. A label "R console" is overlaid on the console output.
- Graphical output:** The plot window displays a box plot titled "Biomass estimation per plot with different models". The y-axis is labeled "Biomass (Mg/ha)" and ranges from 100 to 500. The plot shows the distribution of biomass for different models. A label "Graphical output" is overlaid on the plot.

<https://cran.r-project.org/bin/windows/base/>

4. Ngôn ngữ lập trình thông thường

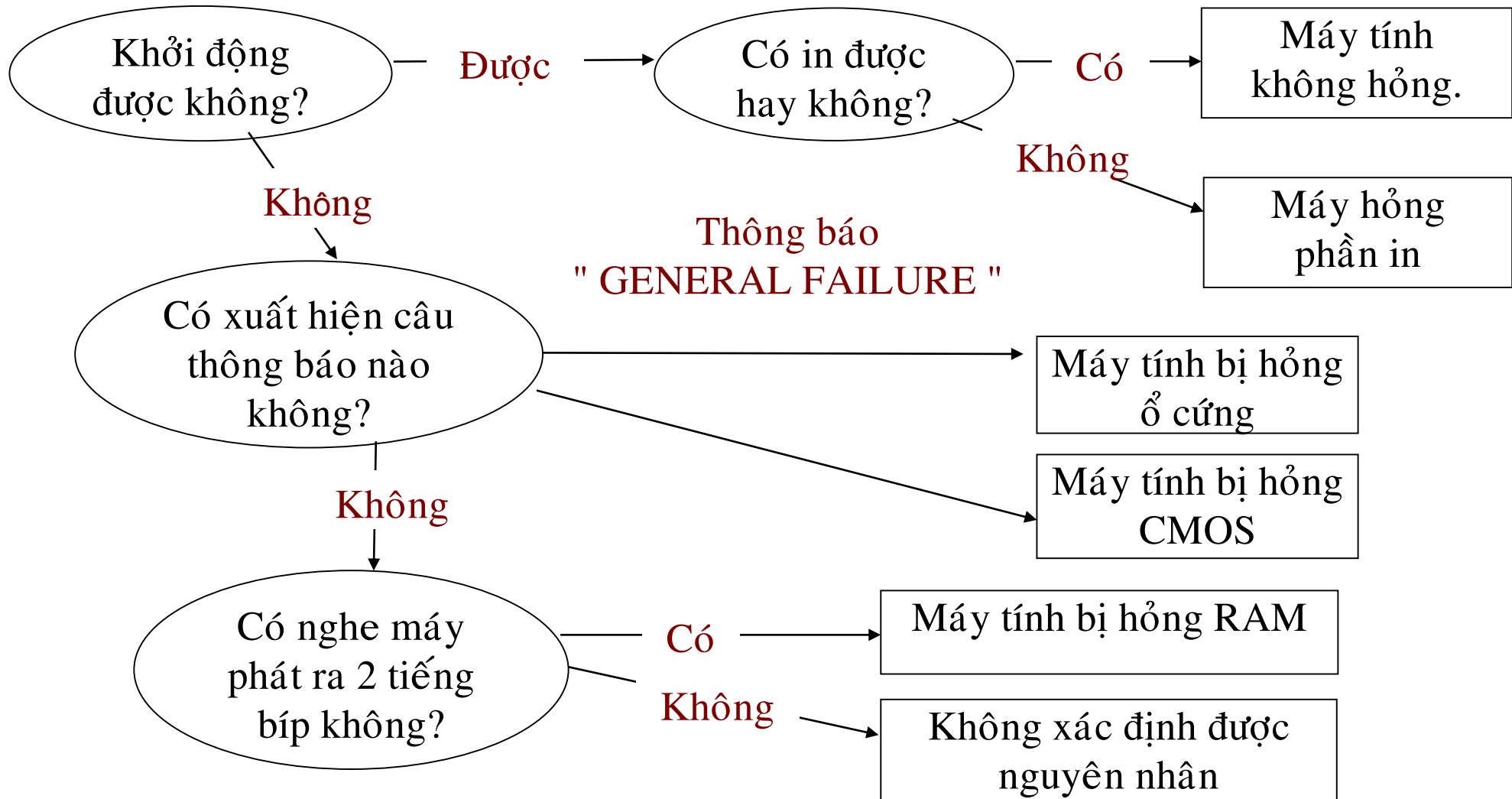


```
11 import java.io.File
12 import scala.io.Source
13 import scala.collection.mutable.Map
14
15 class SpellCorrector {
16   var wordCounts : Map[String, Int] = Map()
17   val alphabets = ('a' to 'z').toSet
18
19   def train(trainFile : File) = {
20     val lines = Source.fromFile(trainFile) mkString
21     val wordREPattern = "[A-Za-z]+"
22     wordREPattern.r.findAllIn(lines).foreach( txtWord => {
23       val word = txtWord.toLowerCase
24       if (wordCounts.keySet contains(word)) {
25         wordCounts(word) = wordCounts(word)+1
26       } else {
27         wordCounts += (word -> 1)
28       }
29     })
30   }
31
32   def getSplittedCombinations(word : String) : Set[(String, String)] =
33     (0 to word.length).map( idx => (word.substring(0, idx), word.su
34
35   def getEditOneSpellings(word: String) : Set[String] = {
36     val splits = getSplittedCombinations(word)
37     val deletes = splits.map( s => if (s._2.length>0) {s._1+s._2.su
```

<https://www.scala-lang.org/download/>

4. Ngôn ngữ lập trình thông thường

- Cài đặt hệ ra quyết định bằng ngôn ngữ lập trình thông thường





4. Ngôn ngữ lập trình thông thường

- Biểu diễn tri thức dưới dạng luật dẫn

Tên biến	Miền giá trị
KHOIDONG	DUOC, KHONG
IN	DUOC, KHONG
THONGBAO	HDD, GENERAL, KHONG
AMTHANH	CO, KHONG
HONG	KHONG, IN, HDD, CMOS, RAM, UNKNOWN

4. Ngôn ngữ lập trình thông thường

▪ Tập luật dẫn ban đầu có được từ cây quyết định

1. IF (KHOIDONG = DUOC) AND (IN = DUOC) THEN HONG = KHONG.
2. IF (KHOIDONG = DUOC) AND (IN = KHONG) THEN HONG = IN
3. IF (KHOIDONG = KHONG) AND (THONGBAO = HDD) THEN HONG = HDD
4. IF (KHOIDONG = KHONG) AND (THONGBAO = GENERAL) THEN HONG = CMOS
5. IF (KHOIDONG = KHONG) AND (THONGBAO = KHONG) AND (AMTHANH = CO) THEN HONG = RAM
6. IF (KHOIDONG = KHONG) AND (THONGBAO = KHONG) AND (AMTHANH = KHONG) THEN HONG = UNKNOWN

4. Ngôn ngữ lập trình thông thường

▪ Tập luật có thể viết lại như sau:

- IF (KHOIDONG = DUOC) AND (IN = DUOC) THEN HONG = KHONG.
- IF (KHOIDONG = DUOC) AND (IN = KHONG) THEN HONG = IN
- IF (KHOIDONG = KHONG) AND (THONGBAO = HDD) THEN HONG = HDD
- IF (KHOIDONG = KHONG) AND (THONGBAO = GENERAL) THEN HONG = CMOS
- IF (KHOIDONG = KHONG) AND (THONGBAO=KHONG) THEN KH_KDTB = DUNG
- IF (KH_KDTB = DUNG) AND (AMTHANH = CO) THEN HONG = RAM
- IF (KH_KDTB = DUNG) AND (AMTHANH = KHONG) THEN HONG = UNKNOWN

4. Ngôn ngữ lập trình thông thường

Lưu trữ và phân loại biến:

- **Biến nhập:** Là các biến chỉ xuất hiện ở vế trái của các luật
- **Biến trung gian:** Là các biến xuất hiện ở cả vế trái lẫn vế phải ở các luật
- **Biến xuất:** Các biến chỉ xuất hiện ở vế phải ở các luật

Tên biến	Khởi tạo	Giá trị	Loại	Câu thông báo
KHOIDONG	FALSE		INPUT	Máy tính có khởi động được không?
IN	FALSE		INPUT	Máy tính có in được không?
THONGBAO	FALSE		INPUT	Máy tính có thông báo gì không?
AMTHANH	FALSE		INPUT	Máy tính có phát ra âm thanh gì không?
HONG	FALSE		OUTPUT	Máy bị hỏng ở phần
KH_KDTB	FALSE		TEMP	

4. Ngôn ngữ lập trình thông thường

Lưu trữ luật:

- Để lưu trữ một luật, ta cần lưu trữ các biến tham gia vào vế trái cùng với giá trị của các biến đó (để kích hoạt luật).
- Vế phải của luật chỉ bao gồm một biến nên khá đơn giản ta chỉ việc thêm một cột tên biến và giá trị của biến sẽ được đặt khi luật cháy gọi là giá trị dừng vào bảng VếPhải sau:

Luật	Biến	Giá trị dừng
1	HONG	KHONG
2	HONG	IN
3	HONG	HDD
4	HONG	CMOS
5	KH_KDTB	DUNG
6	HONG	RAM
7	HONG	UNKNOWN

4. Ngôn ngữ lập trình thông thường

Để mô tả về trái của luật, ta dùng bản Vế Trái với 3 cột như sau:

Luật	Biến	Giá Trị dùng
1	KHOIDONG	DUOC
1	IN	DUOC
2	KHOIDONG	DUOC
2	IN	KHONG
3	KHOIDONG	KHONG
3	THONGBAO	HDD
4	KHOIDONG	KHONG
4	THONGBAO	GENERAL
5	KHOIDONG	KHONG
5	THONGBAO	KHONG
6	KH_TDTB	DUNG
6	AMTHANH	CO
7	KH_TDTB	DUNG
7	AMTHANH	KHONG



4. Ngôn ngữ lập trình thông thường

- Với các cấu trúc trên, tại mọi thời điểm, ta đều có thể truy xuất đến mọi thuộc tính của các luật

Sau đây là các ký hiệu :

<luật>.Chay : Cho biết luật có dừng (Cháy) hay chưa.

<luật>.VePhai.Bien : Biến ở vế phải của luật.

<luật>.VePhai.GiaTriChay: Giá trị cháy ứng với biến ở vế phải của luật.

<luật>.VeTrai.SoBien : số lượng biến trong vế trái của luật.

<luật>.VeTrai.Bien[i] : biến thứ i ở vế trái của luật.

<luật>.VeTrai.GiaTriChay[i] : giá trị cháy ứng với biến thứ i ở vế trái của luật.



4. Ngôn ngữ lập trình thông thường

- Hàm kích hoạt luật

FUNCTION KichHoatLuat(L : Luat) : BOOLEAN

BEGIN

IF L.Chay = TRUE **THEN**

 RETURN FALSE; { Luật đã cháy rồi, không kích hoạt được }

 Fire = TRUE;

FOR i = 1 **TO** L.VeTrai.SoBien **DO**

BEGIN

 v = L.VeTrai.Bien[i];

 { có một biến không thỏa điều kiện cháy }

IF (v.KhoiTao = FALSE) **OR** (v.GiaTri \neq L.VeTrai.GiaTriChay[i]) **THEN**

BEGIN

 Fire = FALSE; EXIT FOR;

END;

END;

If Fire = TRUE **THEN**

 L.VePhai.Bien.ThuocTinh.GiaTri = L.VePhai.Bien.GiaTriChay;

 RETRUN Fire;

END;



4. Ngôn ngữ lập trình thông thường

- Cài đặt thuật toán suy diễn lùi

```
FUNCTION TinhGiaTriBien(V : Bien, L : Luat) { Tính giá trị của biến V trong trái của luật L}
BEGIN
    IF (V.KhoiTao = TRUE) THEN RETURN;
    ELSE
        BEGIN
            IF V.Loai = INPUT THEN
                BEGIN <Hỏi người dùng giá trị biến V>; RETURN; END;
            ELSE BEGIN
                FOR EACH LT IN TapLuat DO
                    IF (LT.VePhai = V) THEN BEGIN
                        FOR i = 1 TO LT.VeTrai.SoBien DO BEGIN
                            TinhGiaTriBien(LT.VeTrai.Bien[i], LT);
                        END;
                        IF KichHoatLuat(LT) THEN RETURN;
                    END;
                END;
            END;
        END;
    END;
END
```



4. Ngôn ngữ lập trình thông thường

- Để biết giá trị biến HONG, ta có thể thực hiện như sau :

{ Khởi động trạng thái ban đầu cho tập biến và tập luật. }

FOR EACH $v \in \text{TapBien}$ $v.\text{KhoiTao} = \text{FALSE}$;

FOR EACH $LT \in \text{TapLuot}$ $LT.\text{Chay} = \text{FALSE}$

$V = \text{HONG}$;

{ Luật 0 là một luật rỗng, dùng để "đệm" cho lần đệ quy đầu tiên, luôn cháy }

$\text{TinhGiaTriBien}(V, 0)$;

IF $V.\text{KhoiTao} = \text{FALSE}$ **THEN**

 <Không tính được giá trị biến>;

ELSE

 <Hiển thị giá trị V >;

END

4. Ngôn ngữ lập trình thông thường

- Cài đặt thuật toán suy diễn tiến

Xuất phát từ ba trạng thái là KHOIDONG = KHONG, THONGBAO = KHONG, AMTHANH = CO ta có thể kết luận được điều gì ?

<Cập nhật các trạng thái khởi tạo là TRUE và giá trị của các biến đã biết>;

CapNhat = TRUE;

LSET = TapLuot;

WHILE CapNhat **DO**

BEGIN

CapNhap = FALSE;

FOR EACH LT \in LSET **DO**

IF KichHoatLuot(LT) = TRUE **THEN BEGIN**

{Bỏ những luật đã cháy ra khỏi tập luật.}

LSET = LSET \ LT;

CapNhap = TRUE;

END;

END

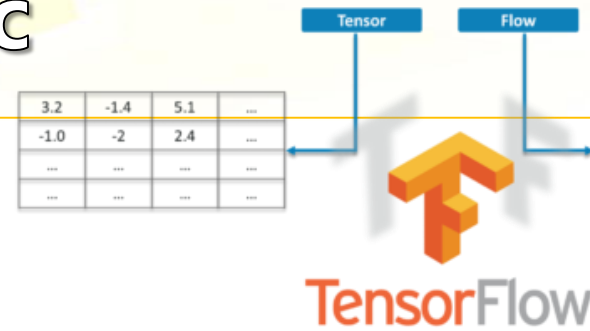


4. Ngôn ngữ lập trình thông thường

- Mục đích thường gặp của quá trình suy diễn tiến là xác định giá trị của tất cả biến xuất. Do đó, ta có thể xem đây là một trường hợp đặc biệt của suy diễn lùi → **Chỉ cần bỏ dòng**
<Hỏi người dùng giá trị biến V>;
- Trong cài đặt ở phần suy diễn lùi là ta có thể dùng lại hàm `TínhGiaTriBien` để cài đặt thuật toán suy diễn tiến một cách vô cùng đơn giản (!) như sau:
<Cập nhật các trạng thái khởi tạo là TRUE và giá trị của các biến nhập đã biết>;
FOR EACH `LT` \in `TapLuot` **DO**
`TínhGiaTriBien(LT.VePhai.Bien,LT);`

5. Một số công cụ khác

<https://www.tensorflow.org/install>

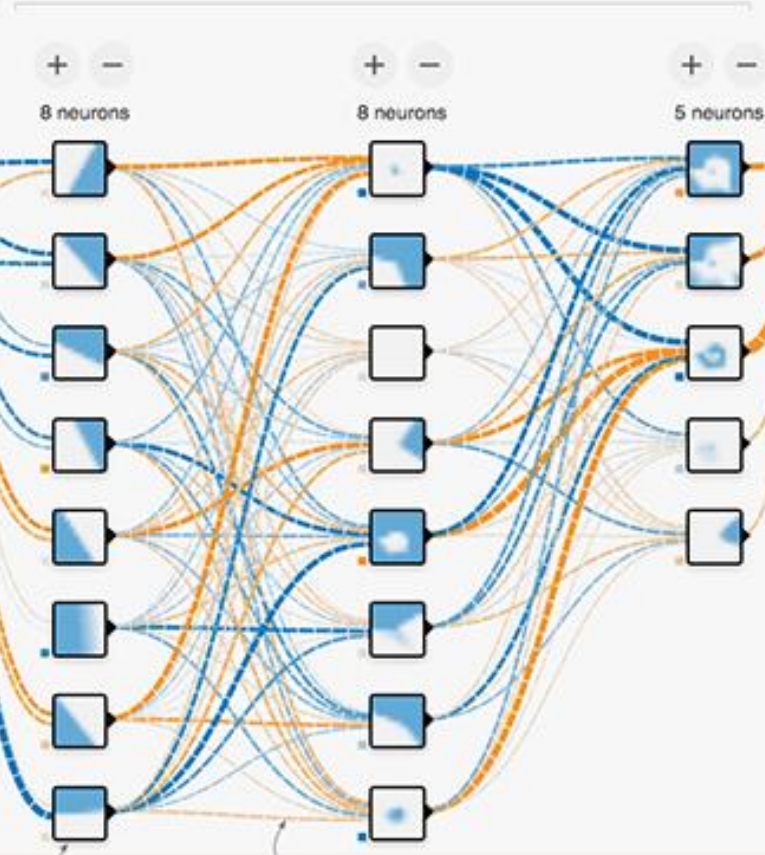


FEATURES

Which properties do you want to feed in?

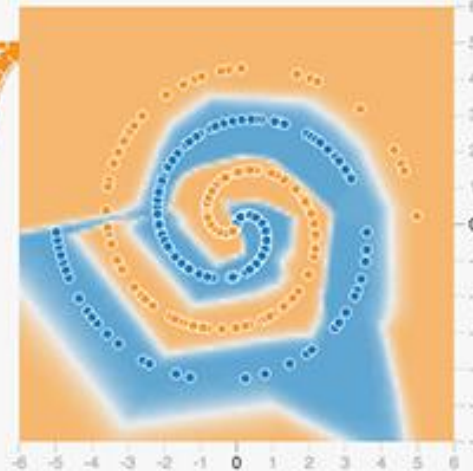
- X_1
- X_2
- X_1^2
- X_2^2
- $X_1 X_2$
- $\sin(X_1)$
- $\sin(X_2)$

3 HIDDEN LAYERS



OUTPUT

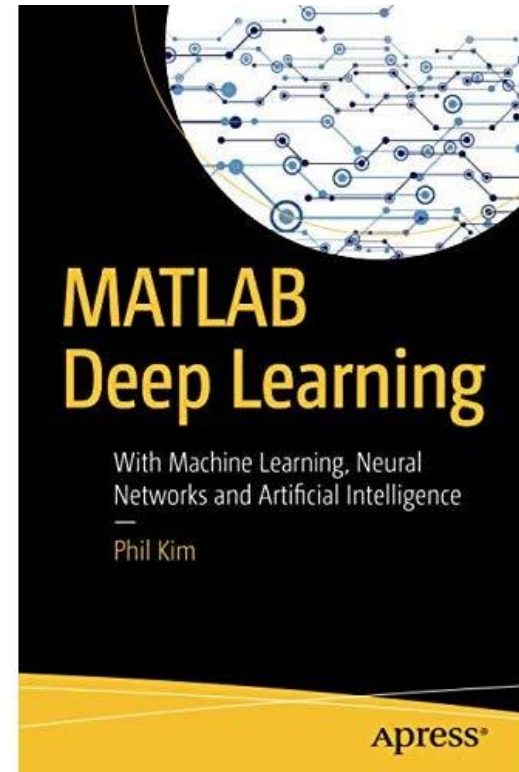
Training loss 0.001



Colors shows data, neuron and weight values.



Neural Network and Fuzzy Logic Fundamental



5. Một số công cụ khác

■ Expert Choice

Decision Support System for Solar Power Plant Site Selection - [Weighting]

Working Data Security About

Weighting Report Site Data Total Weight Setting Specification Log Off

Input Score

	A	B	C	D	E
A	1,1,1	2,3,4	1/4,1/3	3,1/2,1	1/3,1/2
B	1/4,1/3,1	1,1,1	1/6,1/5	1/4,1/3	1/4,1/3
C	3,4,5	4,5,6	1,1,1	1,2,3	1,2,3
D	1,2,3	3,4,5	1/3,1/2,1	1,1,1	3,1/2,1
E	2,3,4	3,4,5	1/3,1/2,1	1,2,3	1,1,1

Result Criteria Weight

Climate (A) 0.070

Geographical (B) 0.000

Transportation (C) 0.399

Environment (D) 0.224

Cost (E) 0.308

Consistency Ratio (CR)= 0.030

Consistency Index (CI)= 0.034

The Hierarchy (Main Level)

```

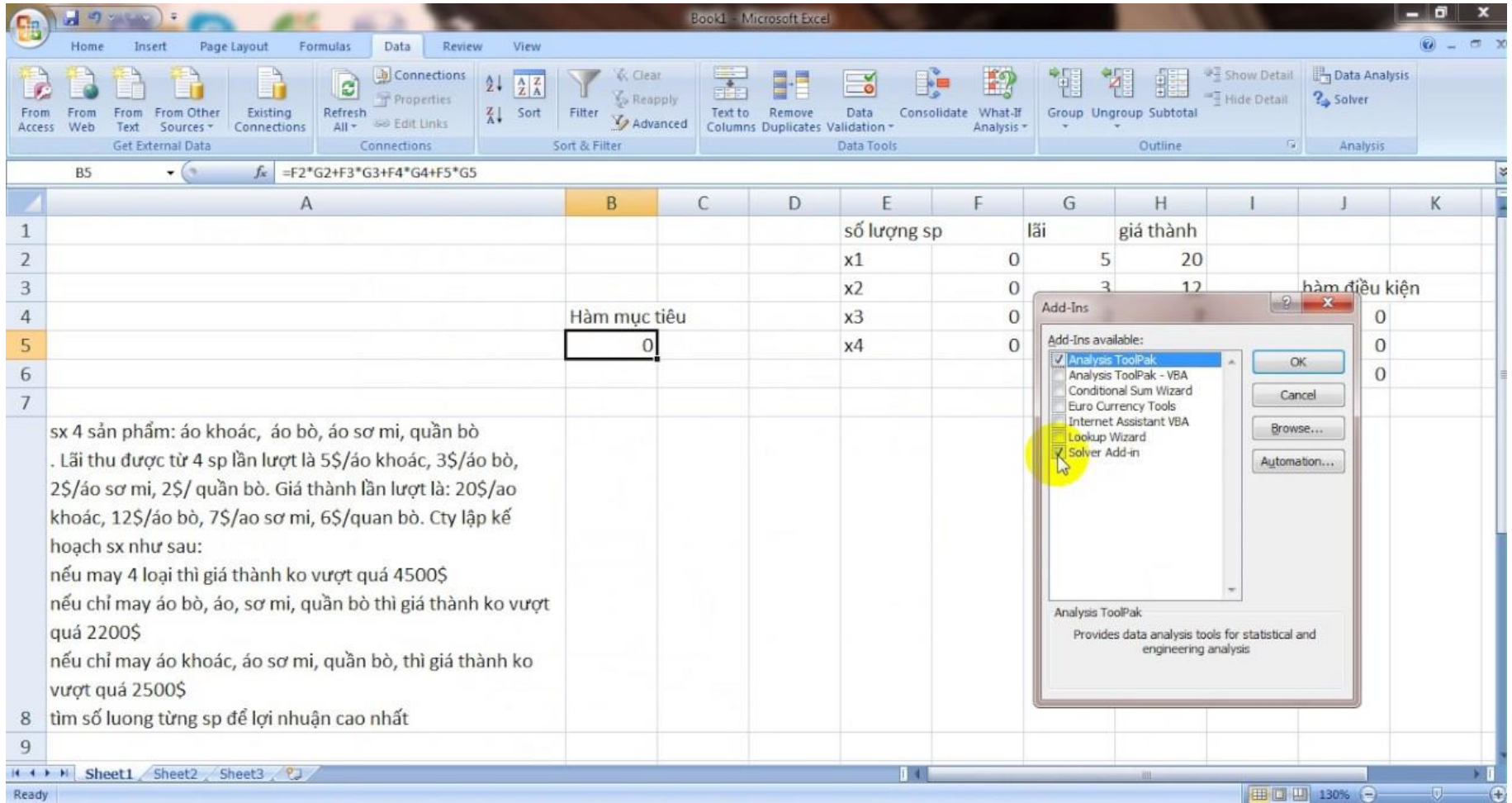
graph LR
    OS[Optimal Site] --- C[Climate (A)]
    OS --- G[Geographical (B)]
    OS --- T[Transportation (C)]
    OS --- E[Environment (D)]
    OS --- Co[Cost (E)]
  
```

Score Description

Equal	(1,1,1)	Cal Weight
Weak	(1,2,3)	Clear
Not bad	(2,3,4)	Save
Preferable	(3,4,5)	Next
Good	(4,5,6)	Back
Fairly good	(5,6,7)	
Very Good	(6,7,8)	
Absolut	(7,8,9)	
Perfect	(8,9,10)	

5. Một số công cụ khác

■ Excel: Công cụ Solve



The screenshot shows the Microsoft Excel interface with the Solver tool being configured. The Solver Parameters dialog box is open, showing the target cell B5 with the formula $=F2*G2+F3*G3+F4*G4+F5*G5$. The Solver is set to maximize the value of the target cell by changing the variable cells E2:E5 (labeled x1, x2, x3, x4). The constraints are set to E2:E5 >= 0. The Solver Load/Save options are set to 'Load/Save' and 'Load Solver Model' is checked. The Solver Options dialog box is also open, showing the 'Add-Ins' tab with the 'Analysis ToolPak' add-in selected.

	A	B	C	D	E	F	G	H	I	J	K
1					số lượng sp		lãi	giá thành			
2					x1	0	5	20			
3					x2	0	3	12			
4					x3	0					
5					x4	0					
6											
7											
8	<p>hàm mục tiêu</p> <p>0</p>										
9	<p>hàng điều kiện</p> <p>0</p> <p>0</p> <p>0</p>										

Text in cell A8:

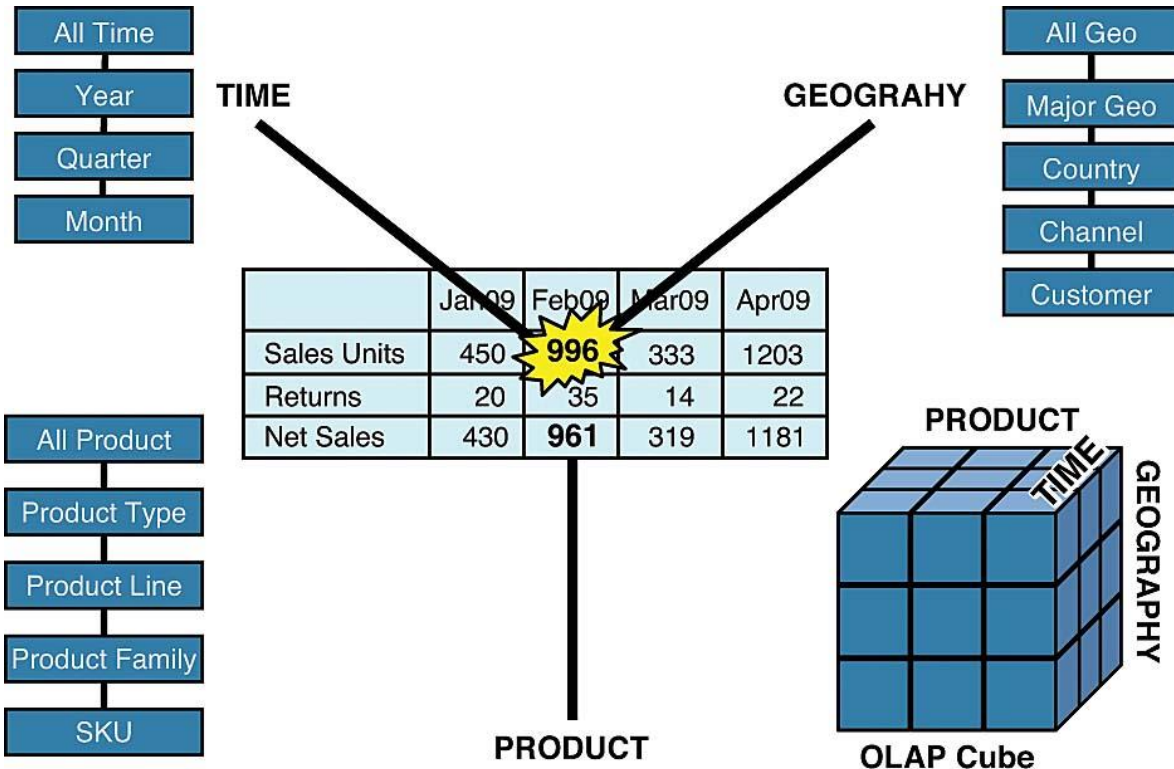
sx 4 sản phẩm: áo khoác, áo bò, áo sơ mi, quần bò
 . Lãi thu được từ 4 sp lần lượt là 5\$/áo khoác, 3\$/áo bò, 2\$/áo sơ mi, 2\$/ quần bò. Giá thành lần lượt là: 20\$/áo khoác, 12\$/áo bò, 7\$/áo sơ mi, 6\$/quan bò. Cty lập kế hoạch sx như sau:
 nếu may 4 loại thì giá thành ko vượt quá 4500\$
 nếu chỉ may áo bò, áo, sơ mi, quần bò thì giá thành ko vượt quá 2200\$
 nếu chỉ may áo khoác, áo sơ mi, quần bò, thì giá thành ko vượt quá 2500\$

Text in cell A9:

8 tìm số luong từng sp để lợi nhuận cao nhất

5. Một số công cụ khác

■ SQL Server Analysis Services



Microsoft®
SQL Server®
Analysis Services

<https://docs.microsoft.com › analysis-services › instances › install-windows>

5. Một số công cụ khác

■ LINGO

