lecture 2 32-60

这一章主要讲了：

pre

2.1 data object

constants 常项包括 atoms原子和数字numbers，与variables变量组成simple object。

另一种数据类型是structure

structure就是包含多个components，比如date（day, month, year），在程序中，structure被看作1个object。functor就是这个集合的名字，date。里面的东西叫argument。

语义上 所有的objects在prolog都称为term

arity：the number of arguments

2.2 matching

match if：

1they are identical 它们相同

2对于两个term里的variable，如果他们都能被instantiate to objects，并在变量替换后，两个term变得相同

instantiate：比如 data( D, M, 2001) and date(D1, may, Y1)

如果match 则把两个term都instantiate，使他们相同

variable等于说是可以变成任何东西，所以variable是可以互相instantiate的

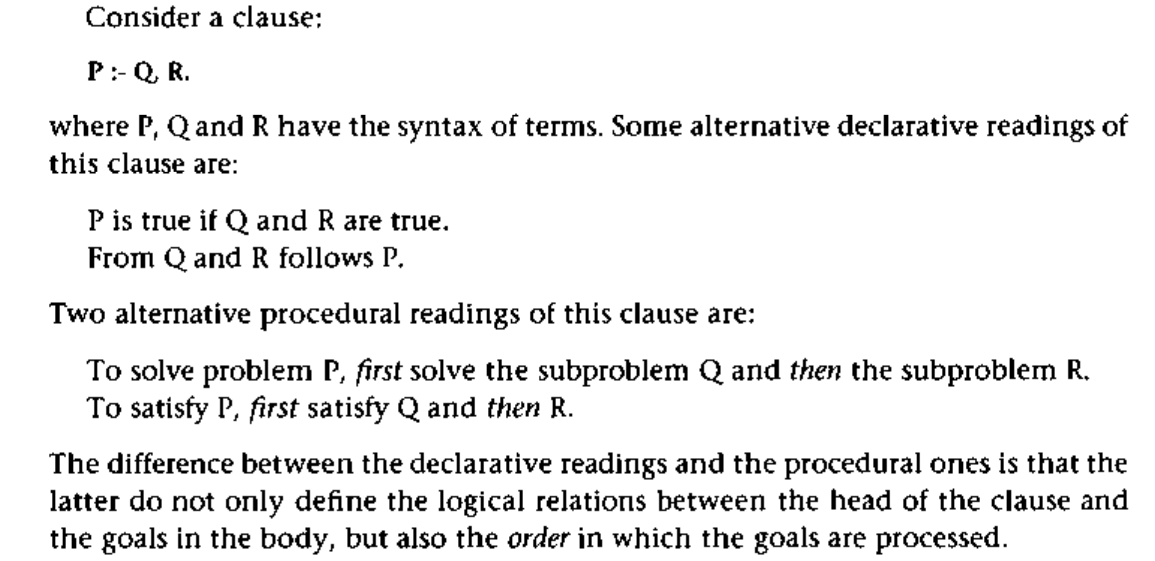
general rule 看S和T是否match：

1如果S和T是constant，only match if they are same object

2如果S是variable，T是anything，那么他们match，且S instantiate to T（替换成T），反之则T换成S

3如果S和T是structure，match only (a) S和T有同样的principal functor，且(b) all components match

2.3 Declarative meaning of prolog programs



声明式，程序式

区别在于，后者不仅表述了逻辑，还规定了处理过程的order

Clause示例

P :- Q,R.

instance：对于一个Clause C, 将其中所有variable substituted by some term，替换成term

variant变形，是指替换变量的表示

declarative决定一个given goal 是否true,如果true，variable的value是什么

conjunction 和，disjunction或

procedural决定Prolog怎么回答clause里的问题

2.4 Procedural meaning

2.5 Order of Clause and Goals

回答问题的顺序

2.6 The relation between Prolog and logic

Lecture3 Lists, operators, arithmetic 60-85

3.1 representation of lists

[ann, tennis, tom, skiing]

head, tail

3.2some operations on lists

**member(b, [a,b,c]) true** 看argument在不在list里

所以member是用于检查某元素在不在list里

member(apple, [apple | \_A)]. 找list，且第一个元素是apple

generate permutation 像排列组合

文本, 信件

描述已自动生成

dictionary用法

手机屏幕截图

描述已自动生成

逐个search

concatenation 拼接 ，decomposing分解

按规则分解

anonymous variable 匿名函数 一个下划线\_

Add an Item

[X | L]

add(X, L, [X | L]).

Delete an item

del(X, L, L1)

1if X is the head of the list, del(X, L, L1)

2if X is in the tail then

del(X, [X | Tail], Tail)

del(X, [Y | Tail], [Y | Tail1]) :- del( X, Tail, Tail1)

Insert

?- del( a, L, [1, 2, 3]).

L= [a, 1, 2, 3]; 及排列组合

Sublist

Permutation

?- permutation( [a, b, c], P).

P = [a, b, c]及其排列组合

3.3 operation notation

运算符

has( peter, information).

:- op(600, xfx, has) 表示我们要把has作为operator使用

意思precedence是600，类型是xfx，是一种infix operator 中缀、插入运算符

xfx中缀，fx前缀，xf后缀

3.4 arithmetic

lecture4 续杯

lecture5 Programming Examples 86-105, 126-142

4.1 Finding a path in a graph

自己到自己也是一个path

所以可能会卡在自己到自己的loop上？

path(Start, End, Path)

文本

描述已自动生成

解决：限制path的长度，用第三章的知识

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

慢慢增加长度，大概意思是说，如果backtrack已经有了某一路径，则在该路径上延长

使得原来的depth-first转为breadth-first，先搜索length为1的，再到2的……

list([]).

list([ \_ | L]) :-

list( L).

?- list( Path ), path( a, c, Path).

4.2 Robot task planning

4.3 Trip planning

5.1 Preventing backtracking

5.2 Examples of using cut

Lecture 6 Programming Examples, 103-115

4.4 Solving cryptarithmetic problems

4.5 The eight-queens problem

4.6 optional

solving a state space problem

135-142

5.3 Negation as failure

5.4 Closed World assumption, and problems with cut and negation

Lecture 7

9.1 Sorting lists (Note that we did not see difference lists yet (so you can skip Fig. 9.3)) 221

9.2 Representing sets by binary trees 225

9.3 Insertion and deletion in a binary dictionary 231

6.1 Testing the types of terms 143

6.3 Various kinds of equality and comparison 155

6.4 Database manipulation 157

findall/3 in 6.6 161

Lecture 8

Chapter 8 Programming Style and Technique

(for now, you can skip section 8.5.2 (3rd edition 8.5.3), namely the part related to difference lists)

Computational thinking

9.4 Displaying trees 235

9.5 Graphs 237

6.2 Constructing and decomposing terms 150

6.5 Control facilities (a.o. ifthenelse in Prolog) 160

6.7 Input and Output 164

Lecture 9

Reading assignment Lecture 9 PLPM 26/11/2021

Your questions for me

One more meta predicate

The decode problem

7.1: Constraint satisfaction and logic programming 177 – 184

Lecture 10 PLPM 3/12/2021

7.2 CLP over real numbers: CLP(R), CLP(Q) 184

7.3 Example of CLP(R) programming : blocks in boxes 188

7.4 CLP over finite domains: CLP(FD) 191

Lecture 11 PLPM 10/12/2021

19.1 Scheduling with CLP 429

19.2 A simulation program with constraints 436

8.5.2 Improving efficiency of list concatenation by difference lists 209

Reading assignment Lecture 12 PLPM 17/12/2021

Language Processing with Grammar Rules (DCG's)

23.1 Grammar rules in Prolog 554

23.2 Handling meaning 561

ex-exam