



# 动物识别专家系统实验报告

【人工智能原理】



## 目录

一 .	CLIPS 程序实验 .....	1
1 .	实验要求 .....	1
2 .	Animal 识别专家系统实例 .....	1
3 .	动物识别专家系统, CLIPS .....	2
4 .	实例逐步分析 .....	6
二 .	附录---程序运行文本 .....	10

2018-6-18

[裴子祥 计科七班 学号 2015211921]

[指导老师：邓芳]

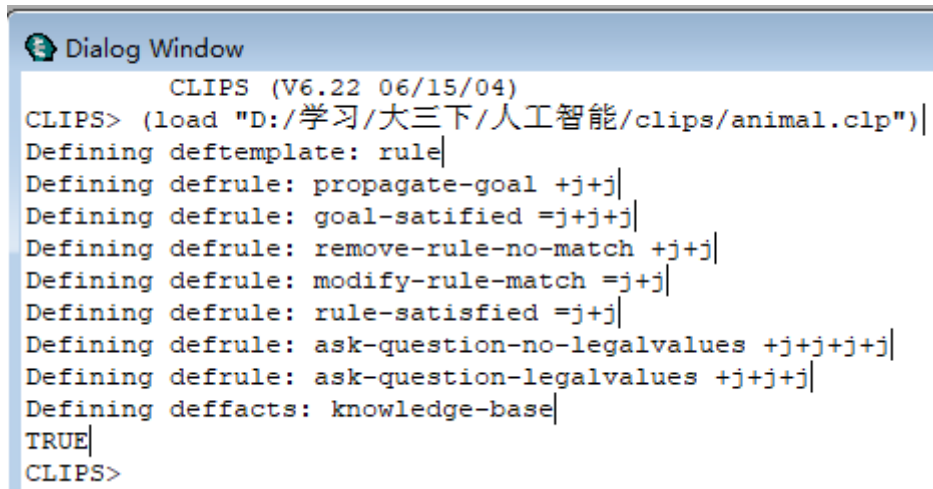
# 一 . CLIPS 程序实验

## 1 . 实验要求

- 1 . 对 CLIPS 进行熟悉。。
- 2 . 对 Animal.clp 程序进行阅读分析，完成：  
给出 Animal.clp 的一个运行实例截图，结合所给出的运行实例，对 CLIPS 和其运行及推理机制、步骤进行介绍。

## 2 . Animal 识别专家系统实例

### 2.1 load，加载编译程序，导入 animal.clp：



```
CLIPS (V6.22 06/15/04)
CLIPS> (load "D:/学习/大三下/人工智能/clips/animal.clp")
Defining deftemplate: rule
Defining defrule: propagate-goal +j+j
Defining defrule: goal-satisfied =j+j+j
Defining defrule: remove-rule-no-match +j+j
Defining defrule: modify-rule-match =j+j
Defining defrule: rule-satisfied =j+j
Defining defrule: ask-question-no-legalvalues +j+j+j+j
Defining defrule: ask-question-legalvalues +j+j+j
Defining deffacts: knowledge-base
TRUE
CLIPS>
```

可以看到，clips 解析程序中含有一个自定义模版 rule，七个自定义规则和一个自定义事实 knowledge-base，knowledge-base 在这里作为动物识别的综合数据库其中提供了所有的事实。在进行推理时，CLIPS 会将规则左端中的模式与综合数据库中的事实做匹配。

### 2.2 reset，初始化

```
CLIPS> (reset)
CLIPS> (run)
```

### 2.3 run，运行实例

Test1：

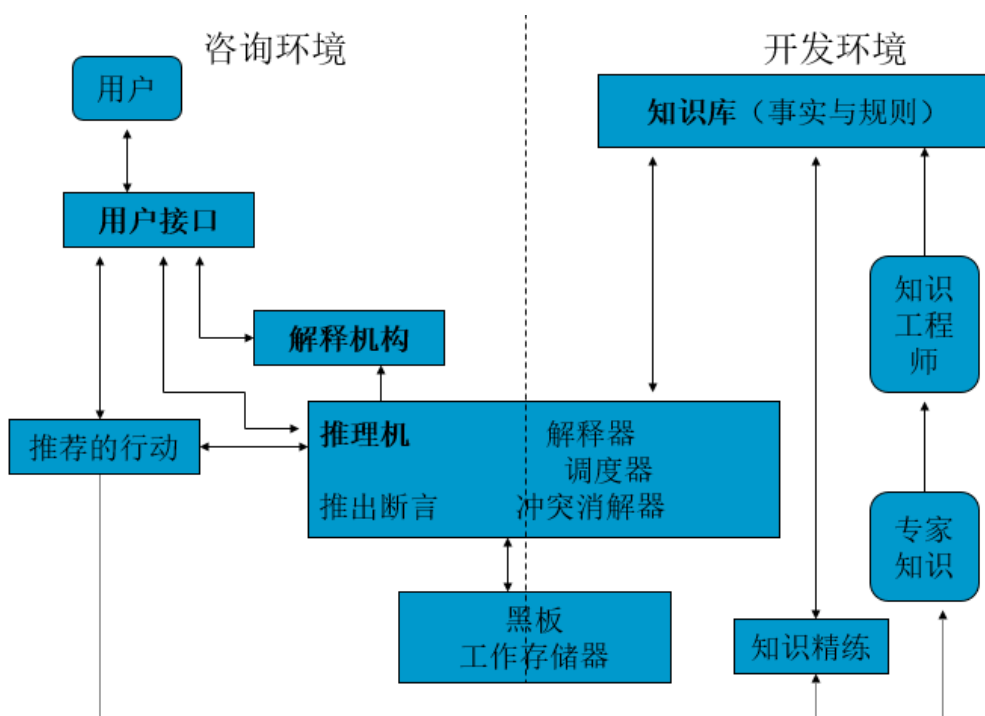
```
Does your animal have a backbone? (yes no) yes
Is the animal warm blooded? (yes no) yes
Normally, does the female of your animal nurse its young with milk? (yes no) yes
Does your animal eat red meat? (yes no) yes
Can your animal fly? (yes no) no
Does your animal have an opposing thumb? (yes no) yes
Does your animal have a prehensile tail? (yes no) yes
I think your animal is a monkey
```

Test2：

```
Does your animal have a backbone? (yes no) no
Does your animal live primarily in soil? (yes no) no
Is the animals body in segments? (yes no) yes
Does your animal have a shell? (yes no) yes
Does your animal have a tail? (yes no) yes
I think your animal is a lobster
```

### 3. 动物识别专家系统，CLIPS

专家系统 = 知识 + 推理。其核心为知识库与推理机。其一般结构如下：



“CLIPS”即 C Language Integrated Production System，一个简单的专家系统系统开发工具。“CLIPS”的核心由工作存储器（事实库）、知识库、推理机三大部分组成，采用产生式规则作为基本的知识表示方式。在这个动物识别专家系统中，同样由此三部分作为主体。

★ 事实。事实由关系名、后跟零个或多个槽（Slot）以及它们的相关值组成，用来表示已知的数据或信息。可以用自定义模板和自定义事实结构来定义事实。形如：

#### 自定义模板：

```
(deftemplate rule                                自定义模板关系名
  (multislot if)                                字段名
  (multislot then))
```

在字段名后设置的是多槽（multislot），槽可以使我们摆脱事实字段必须顺序输入的约束，我们可以在输入事实的时候指明所输入字段的槽名即可，例如：

```
(rule (if backbone is yes)
      (then superphylum is backbone))
```

If 后面跟的是多槽的值，backbone is yes

#### 自定义事实：

（deffacts）来构建我动物的知识库。自定义事实声明，必需指定一个事实名，如 knowledge-base，跟在关键字 deffacts 的后面。事实名后面便是将要被声明到事实表中的事实。

```
(deffacts knowledge-base
```

```
(goal is type.animal)
(legalanswers are yes no)
.....)
```

当 CLIPS 系统启动推理时，会把所有用 deffacts 定义的事实自动添加到工作存储器中，并提供如下命令：

```
assert 把事实添加到工作存储器中；
retract 删除指定事实；
modify 修改自定义模板事实的槽值；
duplicate 复制事实；
clear 删除所有事实。
```

★ 规则。完成一项有价值的工作，专家系统必须得有事实和规则。一条规则用来表示系统推理的有关知识。

IF certain conditions are true

THEN execute the following actions

在 animal.clp 程序代码中有，一个命名为 propagate 的规则：

```
(defrule propagate-goal ""           规则头
  (goal is ?goal)                     模式 1
  (rule (if ?variable $?)            模式 2
    (then ?goal ?value)))
=>                                     THEN,箭头表示
(assert (goal is ?variable)))         执行
```

在一条规则中加入多条模式或行为。重要的一点是，只有当规则中所有的模式都被事实表中的事实满足时，规则才能被触发。（即逻辑与条件满足）

★ 推理机

产生式系统的推理循环可分为 4 个阶段：

i. 模式匹配

从知识库中第一条规则开始，依次扫描知识库中所有规则，把规则的前件与动态存储器中的当前事实相匹配，以搜索满足条件的规则。

ii. 冲突消解

在发生冲突，即多条规则同时被匹配时，根据预先确定的冲突消解策略，确定触发规则。

iii. 激活规则

调用匹配所触发规则的所有子目标的事实。

iv. 动作

把所触发规则的结论添加到动态存储器。

CLIPS 推理机重复上述循环，不断地扫描规则的模式，并把和事实匹配的规则激活，放入议程 (Agenda)之中。议程实际上是一个堆栈，所有激活的规则被按优先级定义的次序压入堆栈。若新压入规则的优先级小于栈顶规则的优先级，则它被压入到栈的下部，直到所有比它优先级高的规则都在此规则的上面。

CLIPS 提供 2 种具体的技术来控制规则的执行：优先级(salience)和模块(module)。可以在定义规则时设置规则的优先级，使议程中的多条规则按优先级大小顺序执行。对大型的具有复杂规则库的情况，可以定义不同的模块来划分知识库，通过改变模块的焦点(focus)来控制当前被执行的模块。

★ 示例代码说明：

```
(defrule propagate-goal ""           规则头
  (goal is ?goal)                     模式 1
```

(rule (if ?variable \$?))

模式 2

(then ?goal ? ?value))

=&gt;

THEN,箭头表示

(assert (goal is ?variable)))

执行

规则 propagate-goal 的模式 1 要求匹配的事实是 goal is 和一个单字段值，并且这个字段值会被约束到变量 ?goal 上。模式 2 要求匹配的事实是使用 rule 模版，if 槽第一个字段值约束到变量?goal 后面跟零个和多个字段值，then 槽第一个字段值必须与约束到变量?goal 相同，最后一个字段值被约束到变量?value 上。当前提满足后，该规则会插入一个 goal is ?variable 这里?variable 是指之前约束到?variable 的值。

此规则反映逆向推理的过程。在综合数据库中 f-1 (goal is type.animal) 这是我们系统推理的目标。与综合数据库的 if-then 语句匹配，将匹配成功的 if-then 中前提提取出来作为我们新的目标。

(defrule goal-satisfied ""

(declare (salience 30))

?f &lt;- (goal is ?goal)

模式 1

(variable ?goal ?value)

模式 2

(answer ? ?text ?goal)

模式 3

=&gt;

(retract ?f)

(format t "%s%s\n" ?text ?value))

规则 goal-satisfied, declare 声明该规则优先值为 30。

模式 1, 要求匹配的事实是 goal is 和一个单字段值，并且这个字段值会被约束到变量 ?goal 上，该匹配事实的索引要被存在变量?f 中。

模式 2, 要求匹配的事实是 variable ?goal ?value, variable 后面的两个字段值分别约束到变量?goal ?value。?goal 的值要与模式 1 的相同。

模式 3, 要求匹配的事实是将匹配事实 answer ? ?text ?goal, ?是通配符，后面的单字段值约束到变量?text, ?goal 的值要与模式 1 的相同。

当前提满足后，该规则会撤销约束到?f 的事实，将?text 和?value 的单字段值变成字符串格式显示在屏幕上。

此规则体现的推理过程，通过回答问题，系统推理到最终目标后，打印最终结果。

(defrule ask-question-legalvalues ""

(declare (salience 10))

(legalanswers ? \$?answers)

模式 1

?f1 &lt;- (goal is ?variable)

模式 2

?f2 &lt;- (question ?variable ? ?text)

模式 3

=&gt;

(retract ?f1)

(format t "%s " ?text)

(printout t ?answers " ")

(bind ?reply (read))

(if (member (lowercase ?reply) ?answers)

then (assert (variable ?variable ?reply))

(retract ?f2)

else (assert (goal is ?variable))))

规则 ask-question-legalvalues 用 declare 声明了该规则的优先值为 10.这样可以保证在与其他低优先值的激活规则相比，可以优先执行。

模式 1, 要求匹配事实 legalanswers 跟着一个通配符 和多字段值并将这个多字段值约束到变量?answers。

模式 2 将匹配事实 goal is ?variable 的事实索引存到变量?f1, goal is 后面的单字段值约束到变量?variable。

模式 3 将匹配事实 question ?variable ? ?text 的事实索引存到变量?f2, ?variable 的字段值必须与模式 2 的相同, ? 是通配符, 后面的单字段符约束到变量?text。

当前提满足后, 该规则会撤销约束到?f1 的事实, 将?text 的单字段值变成字符串格式显示后跟 约束在?answer 上字段值。用关键字 bind 将 用关键字 read 从终端输入的单字段绑定到变量?reply 中。接下来进行条件函数推理, 用 member 检查 输入值与变量 ?answers 中约束的多字段值中的一个相同, 返回值是存在的所以只, 如果为一 则 插入事实(variable ?variable ?reply), 撤销约束到?f2 的事实。

此规则体现推理是, 根据当前的目标属性进行提问, 检测用户输入是否合法, 检测当前目标属性并打印当前动物对应的属性, 用提问的答案和当前的目标属性生成一个事实插入综合数据库中。并撤销无用事实。

```
(defrule remove-rule-no-match ""
  (declare (salience 20))
  (variable ?variable ?value)                                模式 1
  ?f <- (rule (if ?variable ? ~?value $?))                  模式 2
  =>
  (retract ?f))
```

规则 remove-rule-no-match, 优先值为 20。

模式 1, 要求匹配事实 (variable ?variable ?value), 两个单字段值分别约束到变量?variable 和 ?value。

模式 2 将匹配事实 rule (if ?variable ? ~?value \$?)的事实索引存到变量?f, ?variable 的值要与模式 1 的相同。~?value 表示 约束的字段值不能等于模式一中?value 的值, \$? 表示这里可以有零个或多个字段值。

当前提满足后, 该规则会撤销约束到?f 的事实。

此规则体现推理过程, 在 ask-question-legalvalues 规则后, 我们会得到一个动物属性值。检测与得到的属性值不匹配的事实, 并将其从综合数据库中删除。这样可以减轻之后系统做规则匹配的工作量, 提高效率。

```
(defrule modify-rule-match ""
  (declare (salience 20))
  (variable ?variable ?value)                                模式 1
  ?f <- (rule (if ?variable ? ?value and $?rest))            模式 2
  =>
  (modify ?f (if ?rest)))
```

规则 modify-rule-match, 优先值为 20。

模式 1, 要求匹配事实 (variable ?variable ?value), 两个单字段值分别约束到变量?variable 和 ?value。

模式 2 将匹配事实 rule (if ?variable ? ?value and \$?rest)的事实索引存到变量?f, ?variable 的值要与模式 1 的相同。?value 的值要与模式 1 的相同, and 后面的多字段值被约束到变量?\$rest。

当前提满足后, 该规则修改?f 中 if 的槽值, 将其替换成?rest 的值, 这个会做为新的事实被添加到综合数据库中, 撤销约束到?f 的事实索引。

此规则体现的推理过程, 在 ask-question-legalvalues 规则后, 我们会得到一个动物属性值。将需要同这个属性一起做条件的其他属性的事实做下修改。去除有关这个属性的条件语句。

```
(defrule rule-satisfied ""
  (declare (salience 20))
  (variable ?variable ?value)                                模式 1
  ?f <- (rule (if ?variable ? ?value)                        模式 2
    (then ?goal ? ?goal-value))
```

=&gt;

(retract ?f)

(assert (variable ?goal ?goal-value)))

规则 rule-satisfied, 优先值为 20。

模式 1, 要求匹配事实 (variable ?variable ?value), 两个单字段值分别约束到变量?variable 和 ?value。

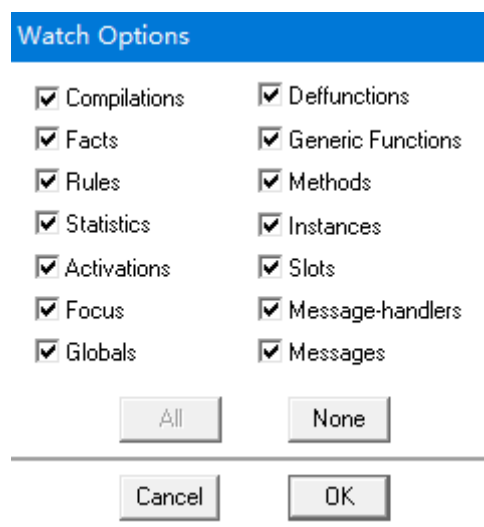
模式 2, 将匹配事实 (rule (if ?variable ? ?value) (then ?goal ? ?goal-value))的事实索引存到变量?f, ?variable 和?value 的值要与模式 1 的相同。Then 的两个槽值分别被约束到变量?goal ? ?goal-value。

当前提满足后, 该规则会撤销约束到?f 的事实并插入新的事实 variable ?goal ?goal-value。

该规则体现的推理过程, 在 ask-question-legalvalues 规则后, 我们会得到一个动物属性值。将这个属性加入综合数据库中并撤销无用事实。

## 4. 实例逐步分析

在设置框内, 将视图全选, 以便执行逐步观察。



★ Reset 命令之后, 程序运行细节如下:

```
CLIPS> (reset)
<== Focus MAIN
==> Focus MAIN
==> instance [initial-object] of INITIAL-OBJECT
MSG >> create ED:1 (<Instance-initial-object>)
HND >> create primary in class USER
      ED:1 (<Instance-initial-object>)
HND << create primary in class USER
      ED:1 (<Instance-initial-object>)
MSG << create ED:1 (<Instance-initial-object>)
MSG >> init ED:1 (<Instance-initial-object>)
HND >> init primary in class USER
      ED:1 (<Instance-initial-object>)
HND << init primary in class USER
      ED:1 (<Instance-initial-object>)
MSG << init ED:1 (<Instance-initial-object>)
==> f-0      (initial-fact)
==> f-1      (goal is type.animal)
==> f-2      (legalanswers are yes no)
==> f-3      (rule (if backbone is yes) (then superphylum is backbone))
==> f-4      (rule (if backbone is no) (then superphylum is jellyback))
==> f-5      (question backbone is "Does your animal have a backbone?")
```

```

==> f-123 (rule (if subspecies is nohorn and fleece is yes) (then type.animal is sheep/goat))
==> Activation 0 propagate-goal: f-1,f-123
==> f-124 (rule (if subspecies is nohorn and fleece is no) (then subspecies is nofleece))
==> f-125 (question fleece is "Does your animal have fleece?")
==> f-126 (rule (if subspecies is nofleece and domesticated is yes) (then type.animal is cow))
==> Activation 0 propagate-goal: f-1,f-126
==> f-127 (rule (if subspecies is nofleece and domesticated is no) (then type.animal is deer/moose/antelope))
==> Activation 0 propagate-goal: f-1,f-127
==> f-128 (question domesticated is "Is your animal domesticated?")
==> f-129 (answer is "I think your animal is a " type.animal)

```

可以看到，reset 命令之后，有三个动作

- i) 自动 focus MAIN
- ii) 预定义事实添加到 MAIN 模块的 FACTS 中。共有 130 条事实，事务列表在“程序执行附录”中能够看到。
- iii) agenda 中出现 42 条被激发的规则，当事实被加载到综合数据库中后，clips 系统会用之前的规则跟事实进行匹配，并将满足条件的规则激发加入到议程当中。激发规则在“程序执行附录”能看到。

```

CLIPS> (agenda)
0 propagate-goal: f-1,f-127
0 propagate-goal: f-1,f-126
0 propagate-goal: f-1,f-126
For a total of 42 activations.

```

系统指明被激活的规则是 propagate-goal 后面跟着激活他的事实。

★ Run 命令之后，程序运行细节如下：

```

CLIPS> (run)
FIRE 1 propagate-goal: f-1,f-127
==> f-130 (goal is subspecies)
==> Activation 0 propagate-goal: f-130,f-124
FIRE 2 propagate-goal: f-130,f-124
==> f-131 (goal is subspecies)
==> Activation 0 propagate-goal: f-131,f-100
==> Activation 0 propagate-goal: f-131,f-109
FIRE 3 propagate-goal: f-131,f-109
==> f-132 (goal is species)

```

clips 系统将在议程中的被激活的规则按顺序开始执行。

FIRE 关键字后跟着是被激活的规则。根据 propagate-goal 的代码我们会知道，事实 f-1 (goal is type.animal) 和 f-127 (rule (if subspecies is nofleece and domesticated is no) (then type.animal is deer/moose/antelope)) 会产生新的事实并将它加入到综合数据库中：f-130 (goal is subspecies)。

当规则运行完毕，综合数据库进行了更新，clips 系统再次匹配规则和事实，并激活新的规则 propagate-goal: f-130,f-124 (注意，因为议程是堆栈形式，遵循后进先出原则)，这是根据事实：

f-130 (goal is subspecies) 和 f-124 (rule (if subspecies is nohorn and fleece is no) (then subspecies is nofleece))

会产生新的事实并将它加入到综合数据库中：f-131 (goal is subspecies)

以此类推，系统运行步骤差不多，直到事实 f-139 (goal is backbone)，被加入综合数据库中后，clips 系统进行匹配规则和事实匹配时，规则 ask-question-legalvalues 被激活：

```

FIRE 10 propagate-goal: f-138,f-4
==> f-139 (goal is backbone)
==> Activation 10 ask-question-legalvalues: f-2,f-139,f-5
FIRE 11 ask-question-legalvalues: f-2,f-139,f-5
<== f-139 (goal is backbone)
Does your animal have a backbone? (yes no) |

```



根据 ask-question-legalvalue 代码我们知道 f-139 和 f-5 的事实索引约束到变量 ?f1 和 ?f2 中, ?variable=backbone, ?text = Does your animal have a backbone? 并打印出来, 我们输入的单字段会被约束到变量 ?reply 中。

### Does your animal have a backbone? 选择 yes

根据 ask-question-legalvalues 代码, 经过 member 关键字的判断和 if then 语句判断, 该规则将在综合数据库中插入新的事实 f-140(variable backbone yes) 并撤销事实 f-5 (注: 之前的 f-139 已经被撤销)

然后, 因为综合数据库的更新, 在 clips 系统进行新一轮匹配的时候, rule-satisfied: f-140,f-3 remove-rule-no-match: f-140, f-4 被激活。

规则 remove-rule-no-match 去掉原综合数据库中不符合选项的事实 f-4 (rule (if backbone is no) (then superphylum is jellyback)) 因为没有新的事实增加, 继续执行原议程中激活的规则 (注意: 规则 rule-satisfied 和 remove-rule-no-match 优先值都是 20, 根根逻辑策略, 高优先值的策略被放在议程的栈顶)

规则 rule-satisfied, 撤销原综合数据库中的 f-3 并加入新的事实 f-141(variable superphylum backbone) 因为 f-3 的撤销, 引起议程中已激活的规则 propagate-goal: f-138,f-3 失效, 所以该规则被移出议程。

新的事实 f-141 的加入, 专家系统进行新一次的匹配。又四个规则被激活并压入议程:

remove-rule-no-match: f-141,f-9 和 remove-rule-no-match: f-141,f-10 这两个规则 分别移去不满足要求的事实 f-10 f-9 和因事实撤销而移除的原激活规则 propagate-goal: f-137,f-9。

规则 modify-rule-match: f-141,f-7 将 f-7 撤销并加入新事实 f-142 (rule (if warm.blooded is no) (then phylum is cold)) 在此过程中撤销议程中原激活的规则 propagate-goal: f-137,f-142

规则 modify-rule-match: f-141,f-6 撤销 f-6 并加入新事实 f-143 (rule (if warm.blooded is yes) (then phylum is warm)) 在此过程中撤销议程中原激活的规则 propagate-goal: propagate-goal: f-137,f-6

因为新的事实加入, 规则 propagate-goal: f-137,f-143 被激活 并执行之后系统运行的方式与之前讲过相同新的事实 f-144 (goal is warm.blooded) 加入并激活规则 ask-question-legalvalues: f-2,f-144,f-8。

```
Does your animal have a backbone? (yes no) yes
==> f-140 (variable backbone yes)
==> Activation 20 rule-satisfied: f-140,f-3
==> Activation 20 remove-rule-no-match: f-140,f-4
<== f-5 (question backbone is "Does your animal have a backbone?")
FIRE 12 remove-rule-no-match: f-140,f-4
<== f-4 (rule (if backbone is no) (then superphylum is jellyback))
FIRE 13 rule-satisfied: f-140,f-3
<== f-3 (rule (if backbone is yes) (then superphylum is backbone))
<== Activation 0 propagate-goal: f-138,f-3
==> f-141 (variable superphylum backbone)
==> Activation 20 modify-rule-match: f-141,f-6
==> Activation 20 modify-rule-match: f-141,f-7
==> Activation 20 remove-rule-no-match: f-141,f-9
==> Activation 20 remove-rule-no-match: f-141,f-10
FIRE 14 remove-rule-no-match: f-141,f-10
<== f-10 (rule (if superphylum is jellyback and live.prime.in.soil is no) (then phylum is elsewhere))
FIRE 15 remove-rule-no-match: f-141,f-9
<== f-9 (rule (if superphylum is jellyback and live.prime.in.soil is yes) (then phylum is soil))
<== Activation 0 propagate-goal: f-137,f-9
FIRE 16 modify-rule-match: f-141,f-7
<== f-7 (rule (if superphylum is backbone and warm.blooded is no) (then phylum is cold))
<== Activation 0 propagate-goal: f-137,f-7
==> f-142 (rule (if warm.blooded is no) (then phylum is cold))
==> Activation 0 propagate-goal: f-137,f-142
FIRE 17 modify-rule-match: f-141,f-6
<== f-6 (rule (if superphylum is backbone and warm.blooded is yes) (then phylum is warm))
<== Activation 0 propagate-goal: f-137,f-6
==> f-143 (rule (if warm.blooded is yes) (then phylum is warm))
==> Activation 0 propagate-goal: f-137,f-143
FIRE 18 propagate-goal: f-137,f-143
==> f-144 (goal is warm.blooded)
==> Activation 10 ask-question-legalvalues: f-2,f-144,f-8
FIRE 19 ask-question-legalvalues: f-2,f-144,f-8
<== f-144 (goal is warm.blooded)
Is the animal warm blooded? (yes no)
```

### Is the animal warm blooded? 选择 yes

```

Is the animal warm blooded? (yes no) yes
==> f-145 (variable warm.blooded yes)
==> Activation 20 rule-satisfied: f-145,f-143
==> Activation 20 remove-rule-no-match: f-145,f-142
<== f-8 (question warm.blooded is "Is the animal warm blooded?")
FIRE 20 remove-rule-no-match: f-145,f-142
<== f-142 (rule (if warm.blooded is no) (then phylum is cold))
<== Activation 0 propagate-goal: f-137,f-142
FIRE 21 rule-satisfied: f-145,f-143
<== f-143 (rule (if warm.blooded is yes) (then phylum is warm))
==> f-146 (variable phylum warm)
==> Activation 20 modify-rule-match: f-146,f-12
==> Activation 20 modify-rule-match: f-146,f-13
==> Activation 20 remove-rule-no-match: f-146,f-15
==> Activation 20 remove-rule-no-match: f-146,f-16
==> Activation 20 remove-rule-no-match: f-146,f-18
==> Activation 20 remove-rule-no-match: f-146,f-19
==> Activation 20 remove-rule-no-match: f-146,f-21
==> Activation 20 remove-rule-no-match: f-146,f-22
FIRE 22 remove-rule-no-match: f-146,f-22
<== f-22 (rule (if phylum is elsewhere and body.in.segments is no) (then class is unified))
FIRE 23 remove-rule-no-match: f-146,f-21
<== f-21 (rule (if phylum is elsewhere and body.in.segments is yes) (then class is segments))
<== Activation 0 propagate-goal: f-136,f-21
FIRE 24 remove-rule-no-match: f-146,f-19
<== f-19 (rule (if phylum is soil and flat.bodied is no) (then type.animal is worm/leech))
<== Activation 0 propagate-goal: f-1,f-19
FIRE 25 remove-rule-no-match: f-146,f-18
<== f-18 (rule (if phylum is soil and flat.bodied is yes) (then type.animal is flatworm))
<== Activation 0 propagate-goal: f-1,f-18
FIRE 26 remove-rule-no-match: f-146,f-16
<== f-16 (rule (if phylum is cold and always.in.water is no) (then class is dry))
<== Activation 0 propagate-goal: f-136,f-16
FIRE 27 remove-rule-no-match: f-146,f-15
<== f-15 (rule (if phylum is cold and always.in.water is yes) (then class is water))
<== Activation 0 propagate-goal: f-136,f-15
FIRE 28 modify-rule-match: f-146,f-13
<== f-13 (rule (if phylum is warm and has.breasts is no) (then type.animal is bird/penguin))
<== Activation 0 propagate-goal: f-1,f-13
==> f-147 (rule (if has.breasts is no) (then type.animal is bird/penguin))
==> Activation 0 propagate-goal: f-1,f-147
FIRE 29 modify-rule-match: f-146,f-12
<== f-12 (rule (if phylum is warm and has.breasts is yes) (then class is breasts))
<== Activation 0 propagate-goal: f-136,f-12
==> f-148 (rule (if has.breasts is yes) (then class is breasts))
==> Activation 0 propagate-goal: f-136,f-148
FIRE 30 propagate-goal: f-136,f-148
==> f-149 (goal is has.breasts)
==> Activation 10 ask-question-legalvalues: f-2,f-149,f-14
FIRE 31 ask-question-legalvalues: f-2,f-149,f-14
<== f-149 (goal is has.breasts)
Normally, does the female of your animal nurse its young with milk? (yes no) |
    
```

与前面新添事实、激活规则方法相似，不赘述：

Does the female of your animal nurse its young with milk? 选择 yes

导出 Does your animal eat red meat? 选择 yes

导出 Can your animal fly? 选择 no

导出 Does your animal have an opposing thumb? 选择 yes

导出 Does your animal have a prehensile tail? 选择 yes

最后输出 I think your animal is a monkey, 推理完成：

```

Does your animal have a prehensile tail? (yes no) yes
==> f-170      (variable prehensile.tail yes)
==> Activation 20      rule-satisfied: f-170,f-168
==> Activation 20      remove-rule-no-match: f-170,f-167
<== f-80      (question prehensile.tail is "Does your animal have a prehensile tail?")
FIRE  98 remove-rule-no-match: f-170,f-167
<== f-167      (rule (if prehensile.tail is no) (then species is notail))
<== Activation 0      propagate-goal: f-132,f-167
FIRE  99 rule-satisfied: f-170,f-168
<== f-168      (rule (if prehensile.tail is yes) (then type.animal is monkey))
==> f-171      (variable type.animal monkey)
==> Activation 30      goal-satisfied: f-1,f-171,f-129
FIRE  100 goal-satisfied: f-1,f-171,f-129
<== f-1      (goal is type.animal)
<== Activation 0      propagate-goal: f-1,f-126
<== Activation 0      propagate-goal: f-1,f-123
<== Activation 0      propagate-goal: f-1,f-121
<== Activation 0      propagate-goal: f-1,f-120
<== Activation 0      propagate-goal: f-1,f-118
<== Activation 0      propagate-goal: f-1,f-117
<== Activation 0      propagate-goal: f-1,f-114
<== Activation 0      propagate-goal: f-1,f-112
<== Activation 0      propagate-goal: f-1,f-111
<== Activation 0      propagate-goal: f-1,f-108
<== Activation 0      propagate-goal: f-1,f-106
<== Activation 0      propagate-goal: f-1,f-105
<== Activation 0      propagate-goal: f-1,f-103
<== Activation 0      propagate-goal: f-1,f-102
<== Activation 0      propagate-goal: f-1,f-99
I think your animal is a monkey
FIRE  101 propagate-goal: f-131,f-100
<== Focus MAIN
101 rules fired      Run time is 13721.25 seconds.
0.0073608454040266 rules per second.
113 mean number of facts (140 maximum).
1 mean number of instances (1 maximum).
50 mean number of activations (77 maximum).

```

## 二 . 附录---程序运行文本

CLIPS (V6.22 06/15/04)

CLIPS> (load "D:/学习/大三下/人工智能/clips/animal.clp")

Defining deftemplate: rule

Defining defrule: propagate-goal +j+j

Defining defrule: goal-satisfied =j+j+j

Defining defrule: remove-rule-no-match +j+j

Defining defrule: modify-rule-match =j+j

Defining defrule: rule-satisfied =j+j

Defining defrule: ask-question-no-legalvalues +j+j+j+j

Defining defrule: ask-question-legalvalues +j+j+j

Defining deffacts: knowledge-base

TRUE

CLIPS> (reset)

<== Focus MAIN

==> Focus MAIN

==> instance [initial-object] of INITIAL-OBJECT

MSG >> create ED:1 (<Instance-initial-object>)

HND >> create primary in class USER

ED:1 (<Instance-initial-object>)

HND << create primary in class USER

ED:1 (<Instance-initial-object>)

MSG << create ED:1 (<Instance-initial-object>)

MSG >> init ED:1 (<Instance-initial-object>)

HND >> init primary in class USER

ED:1 (<Instance-initial-object>)

HND << init primary in class USER

ED:1 (<Instance-initial-object>)

MSG << init ED:1 (<Instance-initial-object>)

==> f-0 (initial-fact)

==> f-1 (goal is type.animal)

==> f-2 (legalanswers are yes no)

==> f-3 (rule (if backbone is yes) (then superphylum is backbone))

==> f-4 (rule (if backbone is no) (then superphylum is jellyback))

==> f-5 (question backbone is "Does your animal have a backbone?")

==> f-6 (rule (if superphylum is backbone and warm.blooded is yes) (then phylum is warm))

==> f-7 (rule (if superphylum is backbone and warm.blooded is no) (then phylum is cold))

==> f-8 (question warm.blooded is "Is the animal warm blooded?")

==> f-9 (rule (if superphylum is jellyback and live.prime.in.soil is yes) (then phylum is soil))

==> f-10 (rule (if superphylum is jellyback and live.prime.in.soil is no) (then phylum is elsewhere))

==> f-11 (question live.prime.in.soil is "Does your animal live primarily in soil?")

==> f-12 (rule (if phylum is warm and has.breasts is yes) (then class is breasts))

==> f-13 (rule (if phylum is warm and has.breasts is no) (then type.animal is bird/penguin))

==> Activation 0 propagate-goal: f-1,f-13

==> f-14 (question has.breasts is "Normally, does the female of your animal nurse its young with milk?")

==> f-15 (rule (if phylum is cold and always.in.water is yes) (then class is water))

==> f-16 (rule (if phylum is cold and always.in.water is no) (then class is dry))

==> f-17 (question always.in.water is "Is your animal always in water?")

==> f-18 (rule (if phylum is soil and flat.bodied is yes) (then type.animal is flatworm))

==> Activation 0 propagate-goal: f-1,f-18

==> f-19 (rule (if phylum is soil and flat.bodied is no) (then type.animal is worm/leech))

==> Activation 0 propagate-goal: f-1,f-19

==> f-20 (question flat.bodied is "Does your animal have a flat body?")

==> f-21 (rule (if phylum is elsewhere and body.in.segments is yes) (then class is segments))

==> f-22 (rule (if phylum is elsewhere and body.in.segments is no) (then class is unified))

==> f-23 (question body.in.segments is "Is the animals body in segments?")

==> f-24 (rule (if class is breasts and can.eat.meat is yes) (then order is meat))

```

==> f-25    (rule (if class is breasts and can.eat.meat is no) (then order is vegy))
==> f-26    (question can.eat.meat is "Does your animal eat red meat?")
==> f-27    (rule (if class is water and boney is yes) (then type.animal is fish))
==> Activation 0    propagate-goal: f-1,f-27
==> f-28    (rule (if class is water and boney is no) (then type.animal is shark/ray))
==> Activation 0    propagate-goal: f-1,f-28
==> f-29    (question boney is "Does your animal have a boney skeleton?")
==> f-30    (rule (if class is dry and scally is yes) (then order is scales))
==> f-31    (rule (if class is dry and scally is no) (then order is soft))
==> f-32    (question scally is "Is your animal covered with scaled skin?")
==> f-33    (rule (if class is segments and shell is yes) (then order is shell))
==> f-34    (rule (if class is segments and shell is no) (then type.animal is centipede/millipede/insect))
==> Activation 0    propagate-goal: f-1,f-34
==> f-35    (question shell is "Does your animal have a shell?")
==> f-36    (rule (if class is unified and digest.cells is yes) (then order is cells))
==> f-37    (rule (if class is unified and digest.cells is no) (then order is stomach))
==> f-38    (question digest.cells is "Does your animal use many cells to digest it's food instead of a stomach?")
==> f-39    (rule (if order is meat and fly is yes) (then type.animal is bat))
==> Activation 0    propagate-goal: f-1,f-39
==> f-40    (rule (if order is meat and fly is no) (then family is nowings))
==> f-41    (question fly is "Can your animal fly?")
==> f-42    (rule (if order is vegy and hooves is yes) (then family is hooves))
==> f-43    (rule (if order is vegy and hooves is no) (then family is feet))
==> f-44    (question hooves is "Does your animal have hooves?")
==> f-45    (rule (if order is scales and rounded.shell is yes) (then type.animal is turtle))
==> Activation 0    propagate-goal: f-1,f-45
==> f-46    (rule (if order is scales and rounded.shell is no) (then family is noshell))
==> f-47    (question rounded.shell is "Does the animal have a rounded shell?")
==> f-48    (rule (if order is soft and jump is yes) (then type.animal is frog))
==> Activation 0    propagate-goal: f-1,f-48
==> f-49    (rule (if order is soft and jump is no) (then type.animal is salamander))
==> Activation 0    propagate-goal: f-1,f-49
==> f-50    (question jump is "Does your animal jump?")
==> f-51    (rule (if order is shell and tail is yes) (then type.animal is lobster))
==> Activation 0    propagate-goal: f-1,f-51
==> f-52    (rule (if order is shell and tail is no) (then type.animal is crab))
==> Activation 0    propagate-goal: f-1,f-52
==> f-53    (question tail is "Does your animal have a tail?")
==> f-54    (rule (if order is cells and stationary is yes) (then family is stationary))
==> f-55    (rule (if order is cells and stationary is no) (then type.animal is jellyfish))
==> Activation 0    propagate-goal: f-1,f-55
==> f-56    (question stationary is "Is your animal attached permanently to an object?")
==> f-57    (rule (if order is stomach and multicelled is yes) (then family is multicelled))
==> f-58    (rule (if order is stomach and multicelled is no) (then type.animal is protozoa))
==> Activation 0    propagate-goal: f-1,f-58
    
```

```

==> f-59      (question multicelled is "Is your animal made up of more than one cell?")
==> f-60      (rule (if family is nowings and opposing.thumb is yes) (then genus is thumb))
==> f-61      (rule (if family is nowings and opposing.thumb is no) (then genus is nothumb))
==> f-62      (question opposing.thumb is "Does your animal have an opposing thumb?")
==> f-63      (rule (if family is hooves and two.toes is yes) (then genus is twotoes))
==> f-64      (rule (if family is hooves and two.toes is no) (then genus is onetoe))
==> f-65      (question two.toes is "Does your animal stand on two toes/hooves per foot?")
==> f-66      (rule (if family is feet and live.in.water is yes) (then genus is water))
==> f-67      (rule (if family is feet and live.in.water is no) (then genus is dry))
==> f-68      (question live.in.water is "Does your animal live in water?")
==> f-69      (rule (if family is noshell and limbs is yes) (then type.animal is crocodile/alligator))
==> Activation 0      propagate-goal: f-1,f-69
==> f-70      (rule (if family is noshell and limbs is no) (then type.animal is snake))
==> Activation 0      propagate-goal: f-1,f-70
==> f-71      (question limbs is "Does your animal have limbs?")
==> f-72      (rule (if family is stationary and spikes is yes) (then type.animal is sea.anemone))
==> Activation 0      propagate-goal: f-1,f-72
==> f-73      (rule (if family is stationary and spikes is no) (then type.animal is coral/sponge))
==> Activation 0      propagate-goal: f-1,f-73
==> f-74      (question spikes is "Does your animal normally have spikes radiating from it's body?")
==> f-75      (rule (if family is multicelled and spiral.shell is yes) (then type.animal is snail))
==> Activation 0      propagate-goal: f-1,f-75
==> f-76      (rule (if family is multicelled and spiral.shell is no) (then genus is noshell))
==> f-77      (question spiral.shell is "Does your animal have a spiral-shaped shell?")
==> f-78      (rule (if genus is thumb and prehensile.tail is yes) (then type.animal is monkey))
==> Activation 0      propagate-goal: f-1,f-78
==> f-79      (rule (if genus is thumb and prehensile.tail is no) (then species is notail))
==> f-80      (question prehensile.tail is "Does your animal have a prehensile tail?")
==> f-81      (rule (if genus is nothumb and over.400 is yes) (then species is 400))
==> f-82      (rule (if genus is nothumb and over.400 is no) (then species is under400))
==> f-83      (question over.400 is "Does an adult normally weigh over 400 pounds?")
==> f-84      (rule (if genus is twotoes and horns is yes) (then species is horns))
==> f-85      (rule (if genus is twotoes and horns is no) (then species is nohorns))
==> f-86      (question horns is "Does your animal have horns?")
==> f-87      (rule (if genus is onetoe and plating is yes) (then type.animal is rhinoceros))
==> Activation 0      propagate-goal: f-1,f-87
==> f-88      (rule (if genus is onetoe and plating is no) (then type.animal is horse/zebra))
==> Activation 0      propagate-goal: f-1,f-88
==> f-89      (question plating is "Is your animal covered with a protective plating?")
==> f-90      (rule (if genus is water and hunted is yes) (then type.animal is whale))
==> Activation 0      propagate-goal: f-1,f-90
==> f-91      (rule (if genus is water and hunted is no) (then type.animal is dolphin/porpoise))
==> Activation 0      propagate-goal: f-1,f-91
==> f-92      (question hunted is "Is your animal, unfortunately, commercially hunted?")
==> f-93      (rule (if genus is dry and front.teeth is yes) (then species is teeth))
    
```

```

==> f-94      (rule (if genus is dry and front.teeth is no) (then species is noteeth))
==> f-95      (question front.teeth is "Does your animal have large front teeth?")
==> f-96      (rule (if genus is noshell and bivalve is yes) (then type.animal is clam/oyster))
==> Activation 0      propagate-goal: f-1,f-96
==> f-97      (rule (if genus is noshell and bivalve is no) (then type.animal is squid/octopus))
==> Activation 0      propagate-goal: f-1,f-97
==> f-98      (question bivalve is "Is your animal protected by two half-shells?")
==> f-99      (rule (if species is notail and nearly.hairless is yes) (then type.animal is man))
==> Activation 0      propagate-goal: f-1,f-99
==> f-100     (rule (if species is notail and nearly.hairless is no) (then subspecies is hair))
==> f-101     (question nearly.hairless is "Is your animal nearly hairless?")
==> f-102     (rule (if species is 400 and land.based is yes) (then type.animal is bear/tiger/lion))
==> Activation 0      propagate-goal: f-1,f-102
==> f-103     (rule (if species is 400 and land.based is no) (then type.animal is walrus))
==> Activation 0      propagate-goal: f-1,f-103
==> f-104     (question land.based is "Is your animal land based?")
==> f-105     (rule (if species is under400 and thintail is yes) (then type.animal is cat))
==> Activation 0      propagate-goal: f-1,f-105
==> f-106     (rule (if species is under400 and thintail is no) (then type.animal is coyote/wolf/fox/dog))
==> Activation 0      propagate-goal: f-1,f-106
==> f-107     (question thintail is "Does your animal have a thin tail?")
==> f-108     (rule (if species is horns and one.horn is yes) (then type.animal is hippopotamus))
==> Activation 0      propagate-goal: f-1,f-108
==> f-109     (rule (if species is horns and one.horn is no) (then subspecies is nohorn))
==> f-110     (question one.horn is "Does your animal have one horn?")
==> f-111     (rule (if species is nohorns and lives.in.desert is yes) (then type.animal is camel))
==> Activation 0      propagate-goal: f-1,f-111
==> f-112     (rule (if species is nohorns and lives.in.desert is no) (then type.animal is giraffe))
==> Activation 0      propagate-goal: f-1,f-112
==> f-113     (question lives.in.desert is "Does your animal normally live in the desert?")
==> f-114     (rule (if species is teeth and large.ears is yes) (then type.animal is rabbit))
==> Activation 0      propagate-goal: f-1,f-114
==> f-115     (rule (if species is teeth and large.ears is no the type.animal is rat/mouse/squirrel/beaver/porcupine)
(then))
==> f-116     (question large.ears is "Does your animal have large ears?")
==> f-117     (rule (if species is noteeth and pouch is yes) (then type.animal is "kangaroo/koala bear"))
==> Activation 0      propagate-goal: f-1,f-117
==> f-118     (rule (if species is noteeth and pouch is no) (then type.animal is mole/shrew/elephant))
==> Activation 0      propagate-goal: f-1,f-118
==> f-119     (question pouch is "Does your animal have a pouch?")
==> f-120     (rule (if subspecies is hair and long.powerful.arms is yes) (then type.animal is
orangutan/gorilla/chimpanzie))
==> Activation 0      propagate-goal: f-1,f-120
==> f-121     (rule (if subspecies is hair and long.powerful.arms is no) (then type.animal is baboon))
==> Activation 0      propagate-goal: f-1,f-121

```

```

==> f-122 (question long.powerful.arms is "Does your animal have long, powerful arms?")
==> f-123 (rule (if subspecies is nohorn and fleece is yes) (then type.animal is sheep/goat))
==> Activation 0 propagate-goal: f-1,f-123
==> f-124 (rule (if subspecies is nohorn and fleece is no) (then subspecies is nofleece))
==> f-125 (question fleece is "Does your animal have fleece?")
==> f-126 (rule (if subspecies is nofleece and domesticated is yes) (then type.animal is cow))
==> Activation 0 propagate-goal: f-1,f-126
==> f-127 (rule (if subspecies is nofleece and domesticated is no) (then type.animal is deer/moose/antelope))
==> Activation 0 propagate-goal: f-1,f-127
==> f-128 (question domesticated is "Is your animal domesticated?")
==> f-129 (answer is "I think your animal is a " type.animal)

```

CLIPS> (agenda)

```

0 propagate-goal: f-1,f-127
0 propagate-goal: f-1,f-126
0 propagate-goal: f-1,f-123
0 propagate-goal: f-1,f-121
0 propagate-goal: f-1,f-120
0 propagate-goal: f-1,f-118
0 propagate-goal: f-1,f-117
0 propagate-goal: f-1,f-114
0 propagate-goal: f-1,f-112
0 propagate-goal: f-1,f-111
0 propagate-goal: f-1,f-108
0 propagate-goal: f-1,f-106
0 propagate-goal: f-1,f-105
0 propagate-goal: f-1,f-103
0 propagate-goal: f-1,f-102
0 propagate-goal: f-1,f-99
0 propagate-goal: f-1,f-97
0 propagate-goal: f-1,f-96
0 propagate-goal: f-1,f-91
0 propagate-goal: f-1,f-90
0 propagate-goal: f-1,f-88
0 propagate-goal: f-1,f-87
0 propagate-goal: f-1,f-78
0 propagate-goal: f-1,f-75
0 propagate-goal: f-1,f-73
0 propagate-goal: f-1,f-72
0 propagate-goal: f-1,f-70
0 propagate-goal: f-1,f-69
0 propagate-goal: f-1,f-58
0 propagate-goal: f-1,f-55
0 propagate-goal: f-1,f-52
0 propagate-goal: f-1,f-51
0 propagate-goal: f-1,f-49

```



```

0      propagate-goal: f-1,f-48
0      propagate-goal: f-1,f-45
0      propagate-goal: f-1,f-39
0      propagate-goal: f-1,f-34
0      propagate-goal: f-1,f-28
0      propagate-goal: f-1,f-27
0      propagate-goal: f-1,f-19
0      propagate-goal: f-1,f-18
0      propagate-goal: f-1,f-13
For a total of 42 activations.
CLIPS> (run)
FIRE    1 propagate-goal: f-1,f-127
==> f-130    (goal is subspecies)
==> Activation 0      propagate-goal: f-130,f-124
FIRE    2 propagate-goal: f-130,f-124
==> f-131    (goal is subspecies)
==> Activation 0      propagate-goal: f-131,f-100
==> Activation 0      propagate-goal: f-131,f-109
FIRE    3 propagate-goal: f-131,f-109
==> f-132    (goal is species)
==> Activation 0      propagate-goal: f-132,f-79
==> Activation 0      propagate-goal: f-132,f-81
==> Activation 0      propagate-goal: f-132,f-82
==> Activation 0      propagate-goal: f-132,f-84
==> Activation 0      propagate-goal: f-132,f-85
==> Activation 0      propagate-goal: f-132,f-93
==> Activation 0      propagate-goal: f-132,f-94
FIRE    4 propagate-goal: f-132,f-94
==> f-133    (goal is genus)
==> Activation 0      propagate-goal: f-133,f-60
==> Activation 0      propagate-goal: f-133,f-61
==> Activation 0      propagate-goal: f-133,f-63
==> Activation 0      propagate-goal: f-133,f-64
==> Activation 0      propagate-goal: f-133,f-66
==> Activation 0      propagate-goal: f-133,f-67
==> Activation 0      propagate-goal: f-133,f-76
FIRE    5 propagate-goal: f-133,f-76
==> f-134    (goal is family)
==> Activation 0      propagate-goal: f-134,f-40
==> Activation 0      propagate-goal: f-134,f-42
==> Activation 0      propagate-goal: f-134,f-43
==> Activation 0      propagate-goal: f-134,f-46
==> Activation 0      propagate-goal: f-134,f-54
==> Activation 0      propagate-goal: f-134,f-57
FIRE    6 propagate-goal: f-134,f-57
    
```

```

==> f-135    (goal is order)
==> Activation 0    propagate-goal: f-135,f-24
==> Activation 0    propagate-goal: f-135,f-25
==> Activation 0    propagate-goal: f-135,f-30
==> Activation 0    propagate-goal: f-135,f-31
==> Activation 0    propagate-goal: f-135,f-33
==> Activation 0    propagate-goal: f-135,f-36
==> Activation 0    propagate-goal: f-135,f-37
FIRE    7 propagate-goal: f-135,f-37
==> f-136    (goal is class)
==> Activation 0    propagate-goal: f-136,f-12
==> Activation 0    propagate-goal: f-136,f-15
==> Activation 0    propagate-goal: f-136,f-16
==> Activation 0    propagate-goal: f-136,f-21
==> Activation 0    propagate-goal: f-136,f-22
FIRE    8 propagate-goal: f-136,f-22
==> f-137    (goal is phylum)
==> Activation 0    propagate-goal: f-137,f-6
==> Activation 0    propagate-goal: f-137,f-7
==> Activation 0    propagate-goal: f-137,f-9
==> Activation 0    propagate-goal: f-137,f-10
FIRE    9 propagate-goal: f-137,f-10
==> f-138    (goal is superphylum)
==> Activation 0    propagate-goal: f-138,f-3
==> Activation 0    propagate-goal: f-138,f-4
FIRE    10 propagate-goal: f-138,f-4
==> f-139    (goal is backbone)
==> Activation 10    ask-question-legalvalues: f-2,f-139,f-5
FIRE    11 ask-question-legalvalues: f-2,f-139,f-5
<== f-139    (goal is backbone)
Does your animal have a backbone? (yes no) yes
==> f-140    (variable backbone yes)
==> Activation 20    rule-satisfied: f-140,f-3
==> Activation 20    remove-rule-no-match: f-140,f-4
<== f-5      (question backbone is "Does your animal have a backbone?")
FIRE    12 remove-rule-no-match: f-140,f-4
<== f-4      (rule (if backbone is no) (then superphylum is jellyback))
FIRE    13 rule-satisfied: f-140,f-3
<== f-3      (rule (if backbone is yes) (then superphylum is backbone))
<== Activation 0    propagate-goal: f-138,f-3
==> f-141    (variable superphylum backbone)
==> Activation 20    modify-rule-match: f-141,f-6
==> Activation 20    modify-rule-match: f-141,f-7
==> Activation 20    remove-rule-no-match: f-141,f-9
==> Activation 20    remove-rule-no-match: f-141,f-10
    
```

```

FIRE 14 remove-rule-no-match: f-141,f-10
<== f-10 (rule (if superphylum is jellyback and live.prime.in.soil is no) (then phylum is elsewhere))
FIRE 15 remove-rule-no-match: f-141,f-9
<== f-9 (rule (if superphylum is jellyback and live.prime.in.soil is yes) (then phylum is soil))
<== Activation 0 propagate-goal: f-137,f-9
FIRE 16 modify-rule-match: f-141,f-7
<== f-7 (rule (if superphylum is backbone and warm.blooded is no) (then phylum is cold))
<== Activation 0 propagate-goal: f-137,f-7
==> f-142 (rule (if warm.blooded is no) (then phylum is cold))
==> Activation 0 propagate-goal: f-137,f-142
FIRE 17 modify-rule-match: f-141,f-6
<== f-6 (rule (if superphylum is backbone and warm.blooded is yes) (then phylum is warm))
<== Activation 0 propagate-goal: f-137,f-6
==> f-143 (rule (if warm.blooded is yes) (then phylum is warm))
==> Activation 0 propagate-goal: f-137,f-143
FIRE 18 propagate-goal: f-137,f-143
==> f-144 (goal is warm.blooded)
==> Activation 10 ask-question-legalvalues: f-2,f-144,f-8
FIRE 19 ask-question-legalvalues: f-2,f-144,f-8
<== f-144 (goal is warm.blooded)
Is the animal warm blooded? (yes no) yes
==> f-145 (variable warm.blooded yes)
==> Activation 20 rule-satisfied: f-145,f-143
==> Activation 20 remove-rule-no-match: f-145,f-142
<== f-8 (question warm.blooded is "Is the animal warm blooded?")
FIRE 20 remove-rule-no-match: f-145,f-142
<== f-142 (rule (if warm.blooded is no) (then phylum is cold))
<== Activation 0 propagate-goal: f-137,f-142
FIRE 21 rule-satisfied: f-145,f-143
<== f-143 (rule (if warm.blooded is yes) (then phylum is warm))
==> f-146 (variable phylum warm)
==> Activation 20 modify-rule-match: f-146,f-12
==> Activation 20 modify-rule-match: f-146,f-13
==> Activation 20 remove-rule-no-match: f-146,f-15
==> Activation 20 remove-rule-no-match: f-146,f-16
==> Activation 20 remove-rule-no-match: f-146,f-18
==> Activation 20 remove-rule-no-match: f-146,f-19
==> Activation 20 remove-rule-no-match: f-146,f-21
==> Activation 20 remove-rule-no-match: f-146,f-22
FIRE 22 remove-rule-no-match: f-146,f-22
<== f-22 (rule (if phylum is elsewhere and body.in.segments is no) (then class is unified))
FIRE 23 remove-rule-no-match: f-146,f-21
<== f-21 (rule (if phylum is elsewhere and body.in.segments is yes) (then class is segments))
<== Activation 0 propagate-goal: f-136,f-21
FIRE 24 remove-rule-no-match: f-146,f-19

```

```

<== f-19      (rule (if phylum is soil and flat.bodied is no) (then type.animal is worm/leech))
<== Activation 0      propagate-goal: f-1,f-19
FIRE  25 remove-rule-no-match: f-146,f-18
<== f-18      (rule (if phylum is soil and flat.bodied is yes) (then type.animal is flatworm))
<== Activation 0      propagate-goal: f-1,f-18
FIRE  26 remove-rule-no-match: f-146,f-16
<== f-16      (rule (if phylum is cold and always.in.water is no) (then class is dry))
<== Activation 0      propagate-goal: f-136,f-16
FIRE  27 remove-rule-no-match: f-146,f-15
<== f-15      (rule (if phylum is cold and always.in.water is yes) (then class is water))
<== Activation 0      propagate-goal: f-136,f-15
FIRE  28 modify-rule-match: f-146,f-13
<== f-13      (rule (if phylum is warm and has.breasts is no) (then type.animal is bird/penguin))
<== Activation 0      propagate-goal: f-1,f-13
==> f-147      (rule (if has.breasts is no) (then type.animal is bird/penguin))
==> Activation 0      propagate-goal: f-1,f-147
FIRE  29 modify-rule-match: f-146,f-12
<== f-12      (rule (if phylum is warm and has.breasts is yes) (then class is breasts))
<== Activation 0      propagate-goal: f-136,f-12
==> f-148      (rule (if has.breasts is yes) (then class is breasts))
==> Activation 0      propagate-goal: f-136,f-148
FIRE  30 propagate-goal: f-136,f-148
==> f-149      (goal is has.breasts)
==> Activation 10      ask-question-legalvalues: f-2,f-149,f-14
FIRE  31 ask-question-legalvalues: f-2,f-149,f-14
<== f-149      (goal is has.breasts)
Normally, does the female of your animal nurse its young with milk? (yes no) yes
==> f-150      (variable has.breasts yes)
==> Activation 20      rule-satisfied: f-150,f-148
==> Activation 20      remove-rule-no-match: f-150,f-147
<== f-14      (question has.breasts is "Normally, does the female of your animal nurse its young with milk?")
FIRE  32 remove-rule-no-match: f-150,f-147
<== f-147      (rule (if has.breasts is no) (then type.animal is bird/penguin))
<== Activation 0      propagate-goal: f-1,f-147
FIRE  33 rule-satisfied: f-150,f-148
<== f-148      (rule (if has.breasts is yes) (then class is breasts))
==> f-151      (variable class breasts)
==> Activation 20      modify-rule-match: f-151,f-24
==> Activation 20      modify-rule-match: f-151,f-25
==> Activation 20      remove-rule-no-match: f-151,f-27
==> Activation 20      remove-rule-no-match: f-151,f-28
==> Activation 20      remove-rule-no-match: f-151,f-30
==> Activation 20      remove-rule-no-match: f-151,f-31
==> Activation 20      remove-rule-no-match: f-151,f-33
==> Activation 20      remove-rule-no-match: f-151,f-34
    
```

```
==> Activation 20      remove-rule-no-match: f-151,f-36
==> Activation 20      remove-rule-no-match: f-151,f-37
FIRE   34 remove-rule-no-match: f-151,f-37
<== f-37      (rule (if class is unified and digest.cells is no) (then order is stomach))
FIRE   35 remove-rule-no-match: f-151,f-36
<== f-36      (rule (if class is unified and digest.cells is yes) (then order is cells))
<== Activation 0      propagate-goal: f-135,f-36
FIRE   36 remove-rule-no-match: f-151,f-34
<== f-34      (rule (if class is segments and shell is no) (then type.animal is centipede/millipede/insect))
<== Activation 0      propagate-goal: f-1,f-34
FIRE   37 remove-rule-no-match: f-151,f-33
<== f-33      (rule (if class is segments and shell is yes) (then order is shell))
<== Activation 0      propagate-goal: f-135,f-33
FIRE   38 remove-rule-no-match: f-151,f-31
<== f-31      (rule (if class is dry and scally is no) (then order is soft))
<== Activation 0      propagate-goal: f-135,f-31
FIRE   39 remove-rule-no-match: f-151,f-30
<== f-30      (rule (if class is dry and scally is yes) (then order is scales))
<== Activation 0      propagate-goal: f-135,f-30
FIRE   40 remove-rule-no-match: f-151,f-28
<== f-28      (rule (if class is water and boney is no) (then type.animal is shark/ray))
<== Activation 0      propagate-goal: f-1,f-28
FIRE   41 remove-rule-no-match: f-151,f-27
<== f-27      (rule (if class is water and boney is yes) (then type.animal is fish))
<== Activation 0      propagate-goal: f-1,f-27
FIRE   42 modify-rule-match: f-151,f-25
<== f-25      (rule (if class is breasts and can.eat.meat is no) (then order is vegy))
<== Activation 0      propagate-goal: f-135,f-25
==> f-152      (rule (if can.eat.meat is no) (then order is vegy))
==> Activation 0      propagate-goal: f-135,f-152
FIRE   43 modify-rule-match: f-151,f-24
<== f-24      (rule (if class is breasts and can.eat.meat is yes) (then order is meat))
<== Activation 0      propagate-goal: f-135,f-24
==> f-153      (rule (if can.eat.meat is yes) (then order is meat))
==> Activation 0      propagate-goal: f-135,f-153
FIRE   44 propagate-goal: f-135,f-153
==> f-154      (goal is can.eat.meat)
==> Activation 10      ask-question-legalvalues: f-2,f-154,f-26
FIRE   45 ask-question-legalvalues: f-2,f-154,f-26
<== f-154      (goal is can.eat.meat)
Does your animal eat red meat? (yes no) yes
==> f-155      (variable can.eat.meat yes)
==> Activation 20      rule-satisfied: f-155,f-153
==> Activation 20      remove-rule-no-match: f-155,f-152
<== f-26      (question can.eat.meat is "Does your animal eat red meat?")
```

```

FIRE 46 remove-rule-no-match: f-155,f-152
<== f-152 (rule (if can.eat.meat is no) (then order is vegy))
<== Activation 0 propagate-goal: f-135,f-152
FIRE 47 rule-satisfied: f-155,f-153
<== f-153 (rule (if can.eat.meat is yes) (then order is meat))
==> f-156 (variable order meat)
==> Activation 20 modify-rule-match: f-156,f-39
==> Activation 20 modify-rule-match: f-156,f-40
==> Activation 20 remove-rule-no-match: f-156,f-42
==> Activation 20 remove-rule-no-match: f-156,f-43
==> Activation 20 remove-rule-no-match: f-156,f-45
==> Activation 20 remove-rule-no-match: f-156,f-46
==> Activation 20 remove-rule-no-match: f-156,f-48
==> Activation 20 remove-rule-no-match: f-156,f-49
==> Activation 20 remove-rule-no-match: f-156,f-51
==> Activation 20 remove-rule-no-match: f-156,f-52
==> Activation 20 remove-rule-no-match: f-156,f-54
==> Activation 20 remove-rule-no-match: f-156,f-55
==> Activation 20 remove-rule-no-match: f-156,f-57
==> Activation 20 remove-rule-no-match: f-156,f-58
FIRE 48 remove-rule-no-match: f-156,f-58
<== f-58 (rule (if order is stomach and multicelled is no) (then type.animal is protozoa))
<== Activation 0 propagate-goal: f-1,f-58
FIRE 49 remove-rule-no-match: f-156,f-57
<== f-57 (rule (if order is stomach and multicelled is yes) (then family is multicelled))
FIRE 50 remove-rule-no-match: f-156,f-55
<== f-55 (rule (if order is cells and stationary is no) (then type.animal is jellyfish))
<== Activation 0 propagate-goal: f-1,f-55
FIRE 51 remove-rule-no-match: f-156,f-54
<== f-54 (rule (if order is cells and stationary is yes) (then family is stationary))
<== Activation 0 propagate-goal: f-134,f-54
FIRE 52 remove-rule-no-match: f-156,f-52
<== f-52 (rule (if order is shell and tail is no) (then type.animal is crab))
<== Activation 0 propagate-goal: f-1,f-52
FIRE 53 remove-rule-no-match: f-156,f-51
<== f-51 (rule (if order is shell and tail is yes) (then type.animal is lobster))
<== Activation 0 propagate-goal: f-1,f-51
FIRE 54 remove-rule-no-match: f-156,f-49
<== f-49 (rule (if order is soft and jump is no) (then type.animal is salamander))
<== Activation 0 propagate-goal: f-1,f-49
FIRE 55 remove-rule-no-match: f-156,f-48
<== f-48 (rule (if order is soft and jump is yes) (then type.animal is frog))
<== Activation 0 propagate-goal: f-1,f-48
FIRE 56 remove-rule-no-match: f-156,f-46
<== f-46 (rule (if order is scales and rounded.shell is no) (then family is noshell))
    
```

```

<== Activation 0      propagate-goal: f-134,f-46
FIRE   57 remove-rule-no-match: f-156,f-45
<== f-45      (rule (if order is scales and rounded.shell is yes) (then type.animal is turtle))
<== Activation 0      propagate-goal: f-1,f-45
FIRE   58 remove-rule-no-match: f-156,f-43
<== f-43      (rule (if order is vegy and hooves is no) (then family is feet))
<== Activation 0      propagate-goal: f-134,f-43
FIRE   59 remove-rule-no-match: f-156,f-42
<== f-42      (rule (if order is vegy and hooves is yes) (then family is hooves))
<== Activation 0      propagate-goal: f-134,f-42
FIRE   60 modify-rule-match: f-156,f-40
<== f-40      (rule (if order is meat and fly is no) (then family is nowings))
<== Activation 0      propagate-goal: f-134,f-40
==> f-157      (rule (if fly is no) (then family is nowings))
==> Activation 0      propagate-goal: f-134,f-157
FIRE   61 modify-rule-match: f-156,f-39
<== f-39      (rule (if order is meat and fly is yes) (then type.animal is bat))
<== Activation 0      propagate-goal: f-1,f-39
==> f-158      (rule (if fly is yes) (then type.animal is bat))
==> Activation 0      propagate-goal: f-1,f-158
FIRE   62 propagate-goal: f-1,f-158
==> f-159      (goal is fly)
==> Activation 10     ask-question-legalvalues: f-2,f-159,f-41
FIRE   63 ask-question-legalvalues: f-2,f-159,f-41
<== f-159      (goal is fly)
Can your animal fly? (yes no) no
==> f-160      (variable fly no)
==> Activation 20     rule-satisfied: f-160,f-157
==> Activation 20     remove-rule-no-match: f-160,f-158
<== f-41      (question fly is "Can your animal fly?")
FIRE   64 remove-rule-no-match: f-160,f-158
<== f-158      (rule (if fly is yes) (then type.animal is bat))
FIRE   65 rule-satisfied: f-160,f-157
<== f-157      (rule (if fly is no) (then family is nowings))
<== Activation 0      propagate-goal: f-134,f-157
==> f-161      (variable family nowings)
==> Activation 20     modify-rule-match: f-161,f-60
==> Activation 20     modify-rule-match: f-161,f-61
==> Activation 20     remove-rule-no-match: f-161,f-63
==> Activation 20     remove-rule-no-match: f-161,f-64
==> Activation 20     remove-rule-no-match: f-161,f-66
==> Activation 20     remove-rule-no-match: f-161,f-67
==> Activation 20     remove-rule-no-match: f-161,f-69
==> Activation 20     remove-rule-no-match: f-161,f-70
==> Activation 20     remove-rule-no-match: f-161,f-72
    
```

```

==> Activation 20      remove-rule-no-match: f-161,f-73
==> Activation 20      remove-rule-no-match: f-161,f-75
==> Activation 20      remove-rule-no-match: f-161,f-76
FIRE 66 remove-rule-no-match: f-161,f-76
<== f-76      (rule (if family is multicelled and spiral.shell is no) (then genus is noshell))
FIRE 67 remove-rule-no-match: f-161,f-75
<== f-75      (rule (if family is multicelled and spiral.shell is yes) (then type.animal is snail))
<== Activation 0      propagate-goal: f-1,f-75
FIRE 68 remove-rule-no-match: f-161,f-73
<== f-73      (rule (if family is stationary and spikes is no) (then type.animal is coral/sponge))
<== Activation 0      propagate-goal: f-1,f-73
FIRE 69 remove-rule-no-match: f-161,f-72
<== f-72      (rule (if family is stationary and spikes is yes) (then type.animal is sea.anemone))
<== Activation 0      propagate-goal: f-1,f-72
FIRE 70 remove-rule-no-match: f-161,f-70
<== f-70      (rule (if family is noshell and limbs is no) (then type.animal is snake))
<== Activation 0      propagate-goal: f-1,f-70
FIRE 71 remove-rule-no-match: f-161,f-69
<== f-69      (rule (if family is noshell and limbs is yes) (then type.animal is crocodile/alligator))
<== Activation 0      propagate-goal: f-1,f-69
FIRE 72 remove-rule-no-match: f-161,f-67
<== f-67      (rule (if family is feet and live.in.water is no) (then genus is dry))
<== Activation 0      propagate-goal: f-133,f-67
FIRE 73 remove-rule-no-match: f-161,f-66
<== f-66      (rule (if family is feet and live.in.water is yes) (then genus is water))
<== Activation 0      propagate-goal: f-133,f-66
FIRE 74 remove-rule-no-match: f-161,f-64
<== f-64      (rule (if family is hooves and two.toes is no) (then genus is onetoe))
<== Activation 0      propagate-goal: f-133,f-64
FIRE 75 remove-rule-no-match: f-161,f-63
<== f-63      (rule (if family is hooves and two.toes is yes) (then genus is twotoes))
<== Activation 0      propagate-goal: f-133,f-63
FIRE 76 modify-rule-match: f-161,f-61
<== f-61      (rule (if family is nowings and opposing.thumb is no) (then genus is nothumb))
<== Activation 0      propagate-goal: f-133,f-61
==> f-162      (rule (if opposing.thumb is no) (then genus is nothumb))
==> Activation 0      propagate-goal: f-133,f-162
FIRE 77 modify-rule-match: f-161,f-60
<== f-60      (rule (if family is nowings and opposing.thumb is yes) (then genus is thumb))
<== Activation 0      propagate-goal: f-133,f-60
==> f-163      (rule (if opposing.thumb is yes) (then genus is thumb))
==> Activation 0      propagate-goal: f-133,f-163
FIRE 78 propagate-goal: f-133,f-163
==> f-164      (goal is opposing.thumb)
==> Activation 10      ask-question-legalvalues: f-2,f-164,f-62
    
```



```

FIRE 79 ask-question-legalvalues: f-2,f-164,f-62
<== f-164 (goal is opposing.thumb)
Does your animal have an opposing thumb? (yes no) yes
==> f-165 (variable opposing.thumb yes)
==> Activation 20 rule-satisfied: f-165,f-163
==> Activation 20 remove-rule-no-match: f-165,f-162
<== f-62 (question opposing.thumb is "Does your animal have an opposing thumb?")
FIRE 80 remove-rule-no-match: f-165,f-162
<== f-162 (rule (if opposing.thumb is no) (then genus is nothumb))
<== Activation 0 propagate-goal: f-133,f-162
FIRE 81 rule-satisfied: f-165,f-163
<== f-163 (rule (if opposing.thumb is yes) (then genus is thumb))
==> f-166 (variable genus thumb)
==> Activation 20 modify-rule-match: f-166,f-78
==> Activation 20 modify-rule-match: f-166,f-79
==> Activation 20 remove-rule-no-match: f-166,f-81
==> Activation 20 remove-rule-no-match: f-166,f-82
==> Activation 20 remove-rule-no-match: f-166,f-84
==> Activation 20 remove-rule-no-match: f-166,f-85
==> Activation 20 remove-rule-no-match: f-166,f-87
==> Activation 20 remove-rule-no-match: f-166,f-88
==> Activation 20 remove-rule-no-match: f-166,f-90
==> Activation 20 remove-rule-no-match: f-166,f-91
==> Activation 20 remove-rule-no-match: f-166,f-93
==> Activation 20 remove-rule-no-match: f-166,f-94
==> Activation 20 remove-rule-no-match: f-166,f-96
==> Activation 20 remove-rule-no-match: f-166,f-97
FIRE 82 remove-rule-no-match: f-166,f-97
<== f-97 (rule (if genus is noshell and bivalve is no) (then type.animal is squid/octopus))
<== Activation 0 propagate-goal: f-1,f-97
FIRE 83 remove-rule-no-match: f-166,f-96
<== f-96 (rule (if genus is noshell and bivalve is yes) (then type.animal is clam/oyster))
<== Activation 0 propagate-goal: f-1,f-96
FIRE 84 remove-rule-no-match: f-166,f-94
<== f-94 (rule (if genus is dry and front.teeth is no) (then species is noteeth))
FIRE 85 remove-rule-no-match: f-166,f-93
<== f-93 (rule (if genus is dry and front.teeth is yes) (then species is teeth))
<== Activation 0 propagate-goal: f-132,f-93
FIRE 86 remove-rule-no-match: f-166,f-91
<== f-91 (rule (if genus is water and hunted is no) (then type.animal is dolphin/porpoise))
<== Activation 0 propagate-goal: f-1,f-91
FIRE 87 remove-rule-no-match: f-166,f-90
<== f-90 (rule (if genus is water and hunted is yes) (then type.animal is whale))
<== Activation 0 propagate-goal: f-1,f-90
FIRE 88 remove-rule-no-match: f-166,f-88
    
```

```
<== f-88      (rule (if genus is onetoe and plating is no) (then type.animal is horse/zebra))
<== Activation 0      propagate-goal: f-1,f-88
FIRE  89 remove-rule-no-match: f-166,f-87
<== f-87      (rule (if genus is onetoe and plating is yes) (then type.animal is rhinoceros))
<== Activation 0      propagate-goal: f-1,f-87
FIRE  90 remove-rule-no-match: f-166,f-85
<== f-85      (rule (if genus is twotoes and horns is no) (then species is nohorns))
<== Activation 0      propagate-goal: f-132,f-85
FIRE  91 remove-rule-no-match: f-166,f-84
<== f-84      (rule (if genus is twotoes and horns is yes) (then species is horns))
<== Activation 0      propagate-goal: f-132,f-84
FIRE  92 remove-rule-no-match: f-166,f-82
<== f-82      (rule (if genus is nothumb and over.400 is no) (then species is under400))
<== Activation 0      propagate-goal: f-132,f-82
FIRE  93 remove-rule-no-match: f-166,f-81
<== f-81      (rule (if genus is nothumb and over.400 is yes) (then species is 400))
<== Activation 0      propagate-goal: f-132,f-81
FIRE  94 modify-rule-match: f-166,f-79
<== f-79      (rule (if genus is thumb and prehensile.tail is no) (then species is notail))
<== Activation 0      propagate-goal: f-132,f-79
==> f-167      (rule (if prehensile.tail is no) (then species is notail))
==> Activation 0      propagate-goal: f-132,f-167
FIRE  95 modify-rule-match: f-166,f-78
<== f-78      (rule (if genus is thumb and prehensile.tail is yes) (then type.animal is monkey))
<== Activation 0      propagate-goal: f-1,f-78
==> f-168      (rule (if prehensile.tail is yes) (then type.animal is monkey))
==> Activation 0      propagate-goal: f-1,f-168
FIRE  96 propagate-goal: f-1,f-168
==> f-169      (goal is prehensile.tail)
==> Activation 10      ask-question-legalvalues: f-2,f-169,f-80
FIRE  97 ask-question-legalvalues: f-2,f-169,f-80
<== f-169      (goal is prehensile.tail)
Does your animal have a prehensile tail? (yes no) yes
==> f-170      (variable prehensile.tail yes)
==> Activation 20      rule-satisfied: f-170,f-168
==> Activation 20      remove-rule-no-match: f-170,f-167
<== f-80      (question prehensile.tail is "Does your animal have a prehensile tail?")
FIRE  98 remove-rule-no-match: f-170,f-167
<== f-167      (rule (if prehensile.tail is no) (then species is notail))
<== Activation 0      propagate-goal: f-132,f-167
FIRE  99 rule-satisfied: f-170,f-168
<== f-168      (rule (if prehensile.tail is yes) (then type.animal is monkey))
==> f-171      (variable type.animal monkey)
==> Activation 30      goal-satisfied: f-1,f-171,f-129
FIRE  100 goal-satisfied: f-1,f-171,f-129
```

```

<== f-1      (goal is type.animal)
<== Activation 0      propagate-goal: f-1,f-126
<== Activation 0      propagate-goal: f-1,f-123
<== Activation 0      propagate-goal: f-1,f-121
<== Activation 0      propagate-goal: f-1,f-120
<== Activation 0      propagate-goal: f-1,f-118
<== Activation 0      propagate-goal: f-1,f-117
<== Activation 0      propagate-goal: f-1,f-114
<== Activation 0      propagate-goal: f-1,f-112
<== Activation 0      propagate-goal: f-1,f-111
<== Activation 0      propagate-goal: f-1,f-108
<== Activation 0      propagate-goal: f-1,f-106
<== Activation 0      propagate-goal: f-1,f-105
<== Activation 0      propagate-goal: f-1,f-103
<== Activation 0      propagate-goal: f-1,f-102
<== Activation 0      propagate-goal: f-1,f-99
I think your animal is a monkey
FIRE 101 propagate-goal: f-131,f-100
<== Focus MAIN
101 rules fired      Run time is 13721.25 seconds.
0.0073608454040266 rules per second.
113 mean number of facts (140 maximum).
1 mean number of instances (1 maximum).
50 mean number of activations (77 maximum).
CLIPS>
    
```