北京邮电大学 计算机学院

《人工智能原理》实验报告

姓名 王睿嘉

学号2015211906

班级2015211307

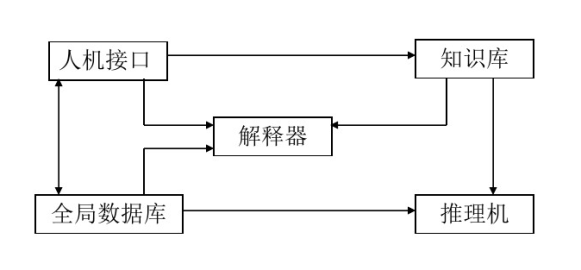
1. 实验要求和原理描述
2. 实验要求
3. 对CLIPS进行熟悉；
4. 对Animal.clp程序进行阅读分析：

结合一个运行实例写出系统完成内容和具体推理的步骤，根据代码和系统的处理方法分析其推理

求解过程和方法。

1. 实验原理
   1. 组成

知识库和推理机是专家系统的核心。随着系统的不断完善，形成了人机接口、知识库、推理机、解释器和全局数据库等5个部分，采用产生式规则作为基本的知识表示方式。如图所示：



知识库是问题求解所需要的领域知识的集合，包括基本事实、规则和其他有关信息。知识的表示形式可以是多种多样的，包括框架、规则、语义网络等。知识库中的知识源于领域专家，是决定专家系统能力的关键，即知识库中知识的质量和数量决定着专家系统的质量水平。一般来说，专家系统中的知识库与专家系统程序是相互独立的，用户可以通过改变、完善知识库中的知识内容来提高专家系统的性能。

推理机是问题求解的核心执行机构，它实际上是对知识进行解释的程序。根据知识的语义，对按一定策略找到的知识进行解释执行，并把结果记录到动态库的适当空间中。推理机的程序与知识库的具体内容无关，即推理机和知识库是分离的，这是专家系统的重要特征。优点是，对知识库的修改无需改动推理机，但是纯粹的形式推理会降低问题求解的效率。因此，将推理机和知识库相结合也不失为一种可选方法。

人机接口是系统与用户进行交流时的接口。通过该接口，用户输入基本信息，回答系统提出的相关问题，系统输出推理结果及相关的解释。

全局数据库也称为动态库或工作存储器，是反映当前问题求解状态的集合，用于存放系统运行过程中所产生的所有信息，以及所需要的原始数据，包括用户输入的信息、推理的中间结果、推理过程的记录等。全局数据库中各种事实、命题和关系组成的状态，既是推理机选用知识的依据，也是解释机制获得推理路径的来源。

解释器用于对求解过程作出说明，并回答用户的提问。两个最基本的问题是“why”和“how”。解释机制涉及程序的透明性，它让用户理解程序正在做什么和为什么这样做，向用户提供了关于系统的一个认识窗口。在很多情况下，解释机制是非常重要的。为了回答“为什么”得到某个结论的询问，系统通常需要反向跟踪动态库中保存的推理路径，并把它翻译成用户能接受的自然语言表达方式。

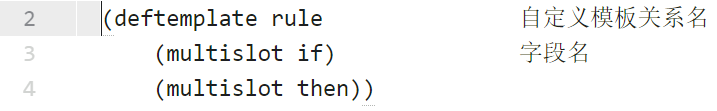
动物识别的专家系统，同样是由以上部分组成。

1. 事实

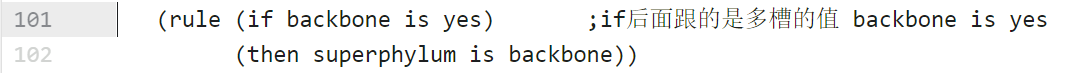
事实由关系名（relation name）后跟零或多个槽（slot）及它们的相关值组成，用来表示已知的

数据或信息，可以用自定义模板和自定义事实结构来表示事实，形如：

·自定义模板

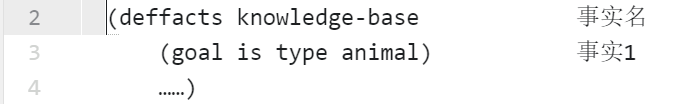


在示例系统中，字段名后设置的是多槽（multislot），槽可以使我们摆脱事实字段必须顺序输入的约束，在输入事实的时候指明所输入字段的槽名即可，例如：



·自定义事实

用自定义事实关键字（deffacts）来构建知识库。自定义事实的声明，必须指定一个事实名，如knowledge-base跟在关键字deffacts的后面。事实名之后，便是将要被声明到事实表中的事实。

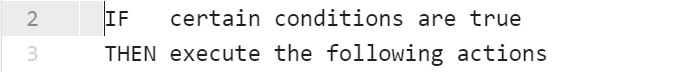


当CLIPS系统启动推理时，会把所有用deffacts定义的事实自动添加到工作存储器中，并提供如下命令：assert，把事实添加到工作存储器中；retract，删除指定事实；modify，修改自定义模板事实的槽值；duplicate，复制事实；clear，删除所有事实。

1. 规则

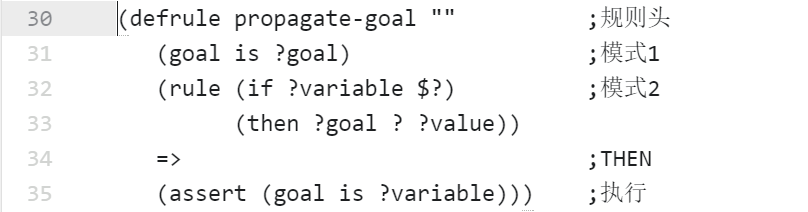
为了完成一项有价值的工作，专家系统必须得有事实和规则。规则用来表示系统推理的有关知识。

IF THEN规则可以用自然语言与计算机语言来混合表示，如下所示（伪代码）：



采用CLIPS语法将上面的伪代码写成一个事实和一个命名为propagate-goal的规则。其中，规则

名紧跟在关键字defrule后面：



可以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。这种约束类型被称为逻辑与条件元素（logical AND conditional element），是关于布尔型的“与”关系。

* 1. 推理与控制机制

产生式系统的推理循环可分为4个阶段：

1. 模式匹配

从知识库中第一条规则开始，依次扫描所有规则，把规则的前提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。

1. 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，确定触发的规则。

1. 激活规则

调用匹配所触发规则的所有子目标的事实。

1. 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）中。议程实际上是一个堆栈，所有激活的规则按优先级别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。

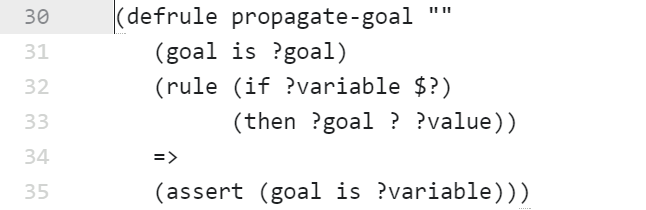
CLIPS提供两种具体的技术来控制规则的执行：优先级和模块。可以在定义规则时设置规则的优先级，使议程中的多条规则按优先级大小顺序执行。对大型的具有复杂规则库的情况，可以定义不同的模块来划分知识库，通过改变模块的焦点来控制当前被执行的模块。

* 1. 代码说明

只对推理机制的代码进行分析，如此，在之后的实例运行讲解时，可以更好地进行整体过程的说

明而不用再进行具体代码细节的剖析。（每个规则的说明顺序和系统运行调用的先后顺序一致）

在CLIPS专家系统中，规则语句进行的是正向推理，即将LHS里的所有模式和知识库中的事实匹配，均匹配上时，=>后的行动才会被执行。

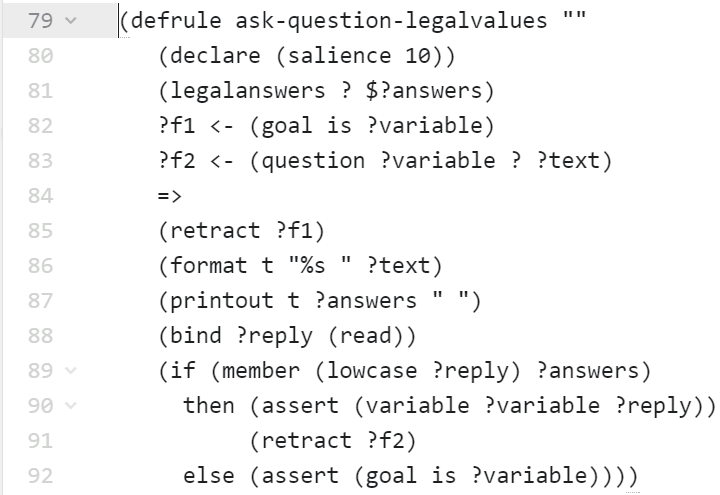


规则propagate-goal的模式1要求匹配的事实是goal is和一个单字段值，并且这个字段值会被约束到变量?goal上。

模式2要求匹配的事实使用rule模板，if槽第一个字段值需为变量?goal后跟零或多个字段值，then槽第一个字段值必须与变量?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。

这个规则反映的是逆向推理的过程。在综合数据库中的f-1（goal is type.animal），是系统推理的目标。跟if-then语句匹配，将匹配成功的if-then中的前提提取出来作为新的目标。



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，它可以优先执行。

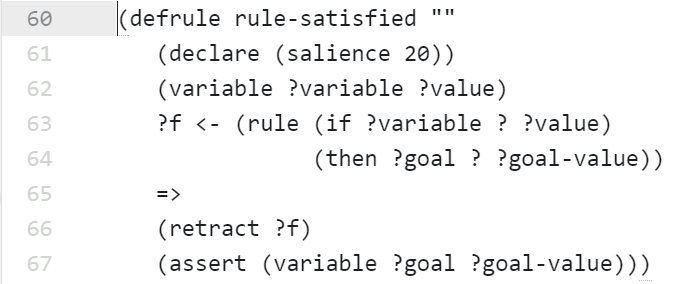
模式1要求匹配legalanswers跟着一个通配符和多字段值，并将多字段值约束到变量?answers。

模式2将匹配事实goal is ?variable的索引存入变量?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的索引存入变量?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，并与约束在变量?answer上的字段值一同显示。使用关键字bind将read操作从终端输入的单字段绑定到变量?reply中。紧接着，进行条件函数推理。用member检查输入值是否为变量?answer中约束的多字段值之一。由于存在返回值，若为1，则插入事实（variable ?variable ?reply）；否则，撤销约束到?f2的事实。

该规则的推理是根据当前的目标属性进行提问，并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。



规则rule-satisfied用declare声明了该规则的优先值为20，这样可以保证在与其他低优先值的激活规则相比，它可以优先执行。

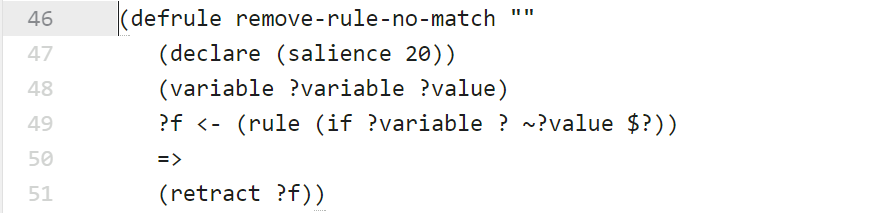
模式1要求匹配事实（variable ?variable ?value），两个单字段值分别约束到变量?variable和?value。

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

该规则的推理过程是，当得到确切的一个动物属性值（通常是在ask-question-legalvalues规则后进行）时，将它加入综合数据库中。



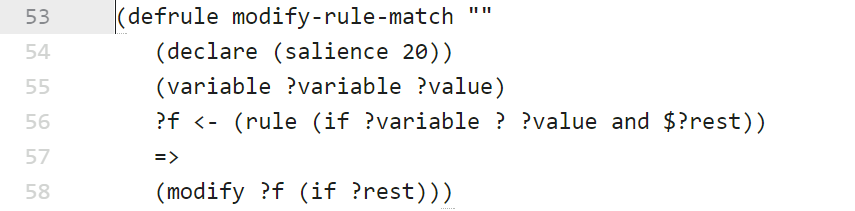
规则remove-rule-no-match用declare声明了该规则的优先值为20，这样可以保证在与其他低优先值的激活规则相比，它可以优先执行。

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 $?）的索引存入变量?f。其中，?variable的值要与模式1的相同。~?value表示约束的字段值不能等于模式1中?value的值，$?表示可以有零或多个字段值。

当前提满足后，该规则会撤销约束到?f的事实。

该规则体现的推理过程是，当得到确切的一个动物属性值（通常是在ask-question-legalvalues规则后进行）时，将那些和这个属性不匹配的事实从综合数据库中移除，这样可以减轻系统做规则匹配的工作量，提高效率。



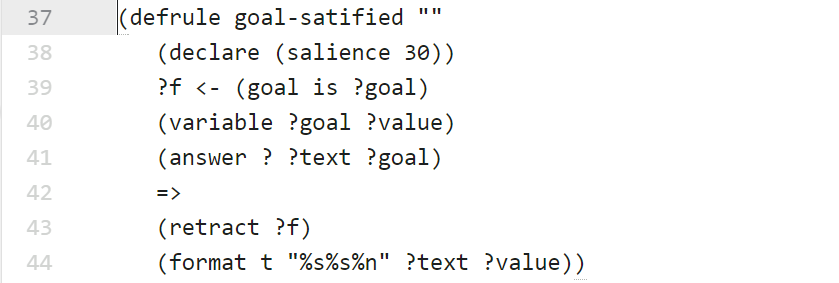
规则modify-rule-match用declare声明了该规则的优先值为20，这样可以保证在与其他低优先值的激活规则相比，它可以优先执行。

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）的索引存入变量?f。其中，?variable和?value的值要与模式1中的相同，and后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，从而作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。

该规则体现的推理过程是，当得到确切的一个动物属性值（通常是在ask-question-legalvalues规则后进行）时，对那些需要同这个属性一起做条件的其他属性的事实进行相应修改，去除有关这个属性的条件语句。



规则goal-satisfied用declare声明了该规则的优先值为20，这样可以保证在与其他低优先值的激活规则相比，它可以优先执行。

模式1要求匹配goal is和一个单字段值，且该字段值会被约束到变量?goal上，相应索引存入变量?f中。

模式2要求匹配事实（variable ?goal ?value）。variable后的两个字段值分别约束到变量?goal和?value。其中，?goal的值要与模式1中的相同。

模式3要求匹配事实（answer ? ?text ?goal）。？是通配符，后面的单字段值约束到变量?text。其中，?goal的值要与模式1中的相同。

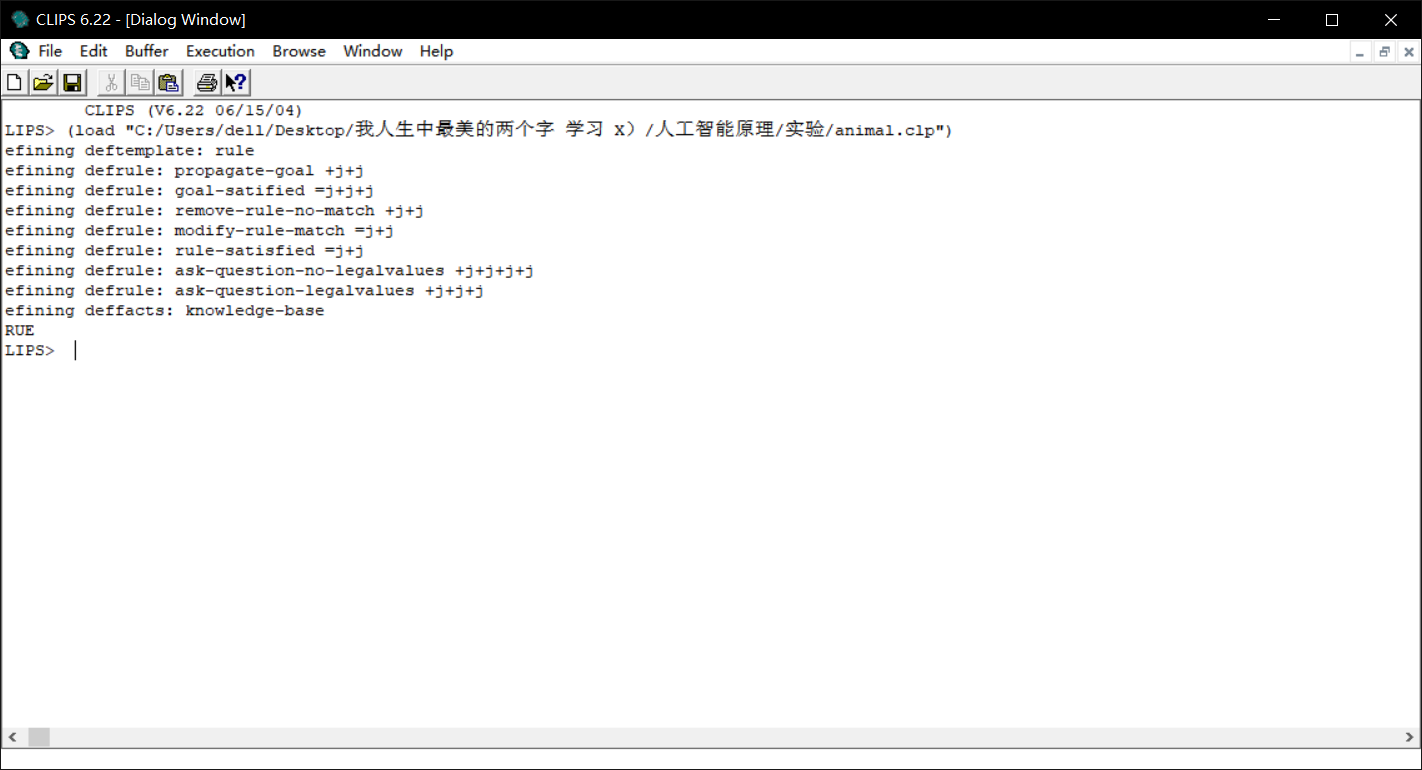
当前提满足后，该规则会撤销约束到?f的事实，并将?text和?value的单字段值以字符串格式显示在屏幕上。

该规则体现的推理过程是，用户通过回答问题，系统推理到最终目标时，将其显示到终端上。

规则ask-question-no-legalvalues不做解释，因为在实例执行中没有使用。

1. 实验内容和心得
2. 实验内容
   1. 加载

加载后，编译程序界面如下：



此步骤中，CLIPS加载了一个自定义模板，包含7个自定义规则和1个自定义事实。其中，knowledge-base作为动物识别的综合数据库提供了所有的事实。当专家系统进行推理的时候会将规则LHS中的模式与综合数据库中的事实做匹配。

* 1. Reset

有3个动作，如下：

1. 自动focus MAIN；
2. 预定义事实添加到MAIN模块的FACTS中。共有130条事实，包括intial-fact f-0。

事实列表会在报告最后以附录的形式说明，在这里简要列举下：

f-0 （initial-fact）

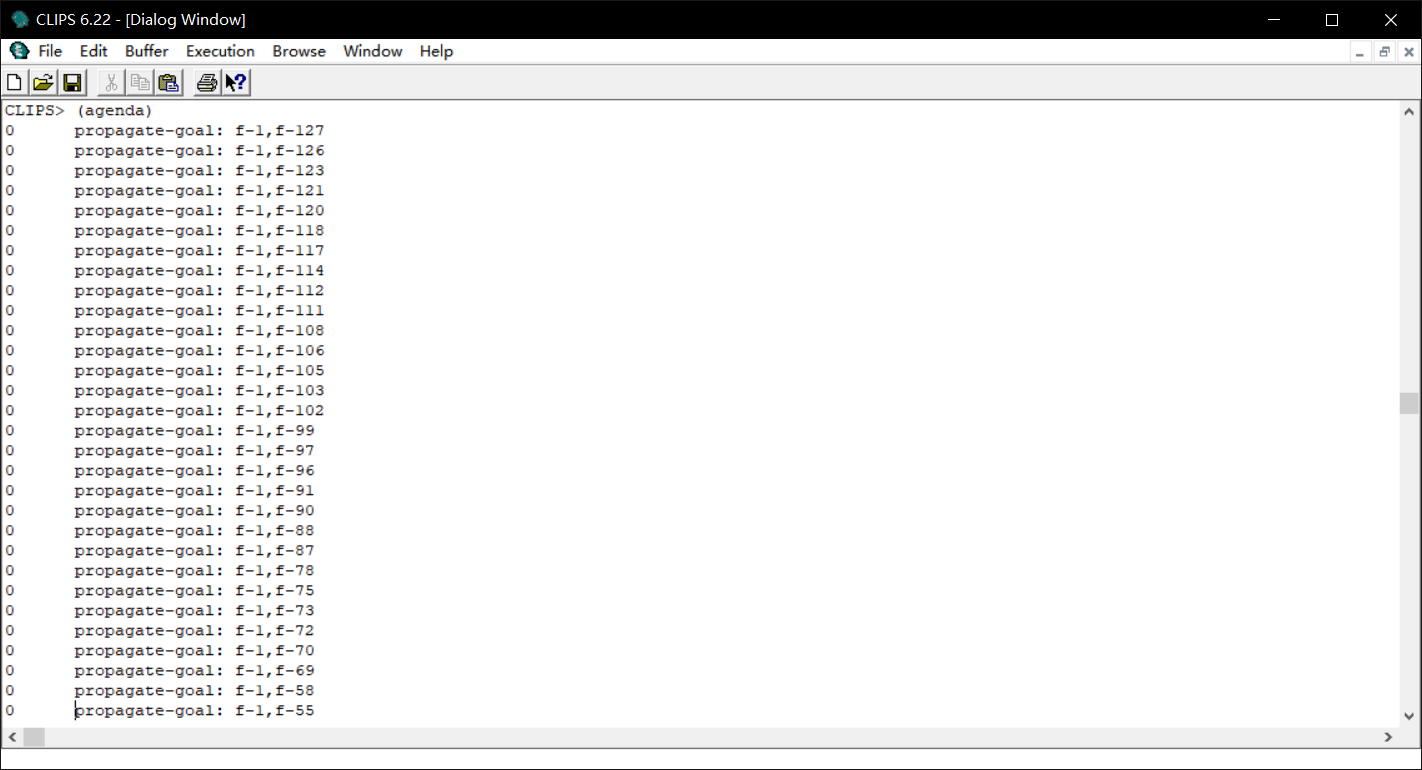
f-1 （goal is type.animal）

f-2 （legalanswers are yes no）

f-3~f-128 rule or question

f-129 （answer is "I think your animal is a " type.animal）

1. Agenda中出现42条被激发的规则：

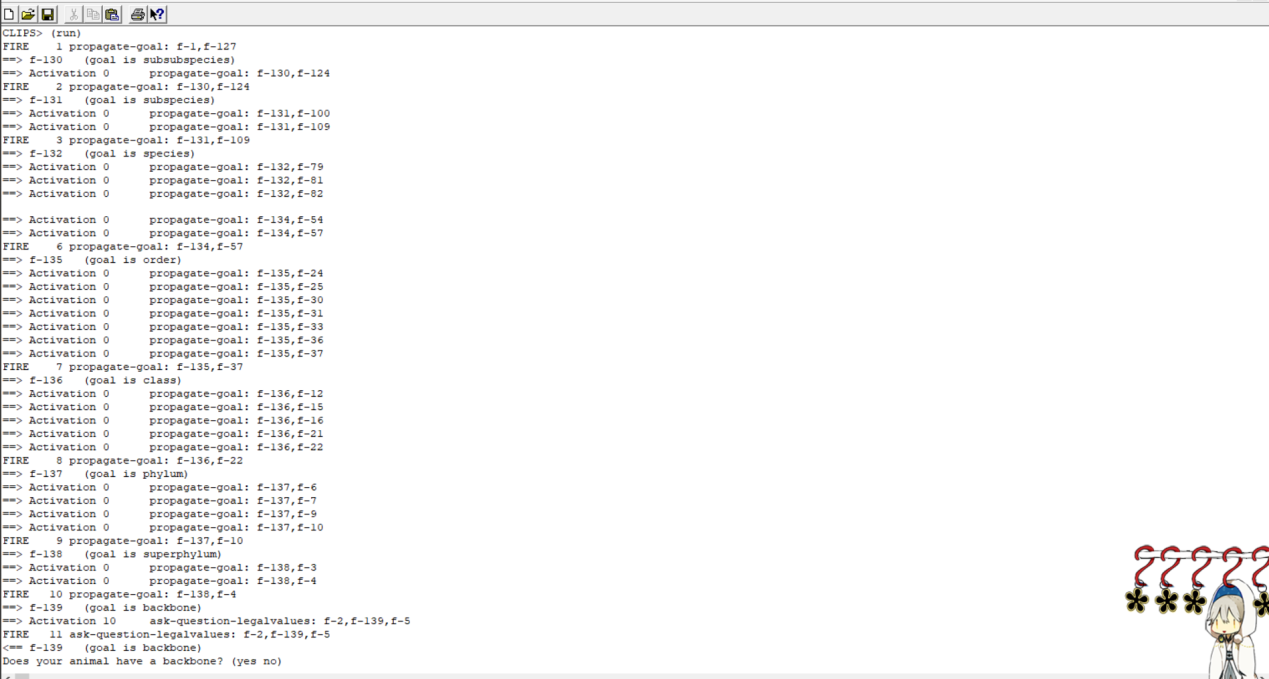


当事实被加载到综合数据库后，CLIPS系统会用之前的规则和事实进行匹配，将满足条件的规则

激发并加入议程。因为被激发的规则较多，会在报告最后以附录的形式列举，在这里仅简要截图。

* 1. Run

CLIPS系统按顺序开始执行议程中被激活的规则（注意：此时所有被激活的都是propagate-goal规则，因而优先级相同，系统顺序执行规则）。接下来，对重点的运行步骤进行讲解：



* 1. 数据库属性和参数的查询及修改

**最大连接数**

使用show variables like ‘%max\_connections%’命令，进行最大连接数的查询，结果为151；

使用set global max\_connections = 最大连接数 命令，进行最大连接数的修改。

**服务器端口**

使用show variables like ‘%port%’命令，进行服务器端口的查询，结果为3306；

服务器端口的修改，需更改配置文件。

**共享缓存数**

使用show variables like ‘%query\_cache%’命令，进行共享缓存数的查询。其中，have\_query\_cache代表是否开启，结果为YES，query\_cache\_limit代表单个查询能够使用的缓存区大小，结果为1M，query\_cache\_min\_res\_unit代表为系统分配的最小缓存块大小，结果为4KB，query\_cache\_size代表缓存大小，结果为1M；

共享缓存数的修改，需更改配置文件。

* 1. 数据库的删除

使用drop database 数据库名 命令，进行数据库的删除，且需使用反引号；

在本实验中，具体命令为drop database `Student Registration System`。

1. 实验心得

在本次实验中，遇到的主要问题有以下五点：

* 1. 根据实验指导书，先进行IP协议的抓包与分析。但在使用远程地址为[www.baidu.com](http://www.baidu.com的ping) 的ping命令时，所产生的IP数据包均传输在IPV6协议下。然而对IPV6协议仅为了解，无需细致掌握，便困惑如何才能捕获到IPV4数据包。带着疑问，先进行了DHCP协议的分析，在此过程中发现将数据发送至DHCP中继代理路由器，所产生的IP数据包为IPV4数据包，该问题得到解决；
  2. IP协议抓包结束后，擅自将结果以协议为关键字进行了排序，导致始终找不到最后一个分段数据包。随后，又按照正常的时间顺序查看结果，发现最后一个数据包为ICMP包，该问题得到解决；
  3. 验证IP分段原理时，对于最后一个ICMP包，遗忘了其封装在IP包中，还存在20字节的IP包头，导致各段数据相加非8000字节，与同学讨论后发现该问题，得到解决；
  4. TCP协议的抓包过程耗费较长时间。起初所选网址不当，导致未出现清晰的两端口数据传输，后更换网址，该问题得到解决；
  5. TCP实际的连接释放过程与课本上所讲解的有出入。书本上的释放过程大致同连接建立过程，经3次握手完成释放，然而，抓包结果通常是经4次握手，两个传输方向分别释放。

实践出真知，本次协议数据的捕获和解析实验是对课堂和书本所学知识的补充。网络中的实际情形与已了解到的原理大体一致，但又复杂许多。通过自己动手、亲力亲为捕获数据包，并对16进制数据进行分析，加深了对包头各字段功能的理解和记忆，收获颇丰。

1. 附录
2. 实验内容

出真知，本次协议数据的捕获和解析实验是对课堂和书本所学知识的补充。网络中的实际情形与已了解到的原理大体一致，但又复杂许多。通过自己动手、亲力亲为捕获数据包，并对16进制数据进行分析，加深了对包