**1、回答集程序（Answer Set Programming）简介**

回答集程序（Answer Set Programming），简称ASP，是一个描述性编程语言（Declarative Programming Language），是人工智能知识表示领域中最重要的几个语言之一。很多知识并不适合用经典一阶逻辑（First-Order Logic）来表示，比如：“如果一个人有VIP卡，而且不知道其是否黑名单上的话，可以给折扣”。这种情况下，可以把它表示成规则如下：

Discount(x) :- VIP(x)，not BlackList(x)

其中，Discount(x)，VIP(x)以及BlackList(x)分别表示可以给x折扣，x是VIP以及x在黑名单上。“:-”是用来表示规则的符号，可以读成“如果”，含义是如果该符号右侧所有事实成立，那么该符号左侧事实也成立。“not”符号是表示缺省符号“不知道”。

回答集程序就是用来计算如果有一堆如上所述规则，那么能推出来的结果应该是什么。

回答集程序是很多经典表示和求解语言的扩充，包括命题可满足求解（SAT），约束满足求解（CSP），非单调推理（NMR），演绎数据库（Datalog）等等。简而言之，

ASP=SAT+Datatlog+NMR+CSP

**2、为什么要使用回答集程序**

a) 回答集程序简洁性。与一般过程性语言（比如C，Java，C++，Python）不同的是，描述性编程语言仅需要把问题描述出来（求解过程由机器用搜索算法自动完成），而不需要告诉机器怎么一步步去求解。因此，对于某些问题（比如约束可满足问题、逻辑求解问题等），ASP可以很简单编程及求解。例如：哈密尔顿回路（Hamiltonian Circuit）问题在一般的过程性语言（如C）中需要写很长的程序。但在ASP中，只需要以下6行:

hc(X,Y) :- edge(X,Y), not otherroute(X,Y).

otherroute(X,Y) :- edge(X,Y), edge(X,Z), hc(X,Z), not =(Y,Z).

otherroute(X,Y) :- edge(X,Y), edge(Z,Y), hc(Z,Y), not =(X,Z).

reached(Y) :- edge(X,Y), hc(X,Y), reached(X), not initialvtx(X).

reached(Y) :- edge(X,Y), hc(X,Y), initialvtx(X).

:- vtx(X), not reached(X).

b) 回答集程序的可复用性。与一般过程性语言（比如C，Java，C++，Python）相比，回答集程序可以相对容易通过简单改写，就能实现更多的功能。比如，如果把哈密尔顿回路扩展到旅行员推销（Traveling Salesman）问题，回答集程序只需要加入新的一条规则：

:- sum[hc(X,Y):edgewt(X,Y,Z) = Z] >= W + 1, maxweight(W).

c) 回答集程序表示语言和自然语言以及数据库语言相对接近。因此，能相对容易将问题表示成回答集程序中的规则。

d) 虽然效率依然是一个很大的障碍，但最新的回答集程序（比如clasp、groc）已经能够求解相对大规模的问题，例如百万级别的变量。

**3、什么时候（不）该使用回答集程序**

如同第二节所述，

因此，

回答集程序绝非

4、回答集程序Groc使用说明

5、Groc语法

6、注意事项