人工智能——基本知识体系

饶洋辉
计算机学院,
中山大学
raoyangh@mail.sysu.edu.cn
http://cse.sysu.edu.cn/node/2471

人工智能各学派的认知观

- 符号主义（Symbolicism）学派：认为人工智能源于数理逻辑。该学派将数学严格公理化，从公理出发，由逻辑推理得到引理，定理，推论。人工智能的创始人之一，John McCarthy是符号主义学派的拥护者。

- 该学派以Robinson提出的归结原理为基础，标志性应用包括：IBM 公司在1997年研发的国际象棋电脑“深蓝”（Deep Blue），及其在2011年开发的认知系统“沃森”（Watson）。

人工智能各学派的认知观

- 联结主义（Connetionism）学派：认为人工智能源于仿生学。该学派的主要理论基础为神经网络及神经网络间的连接机制与学习算法。如果说符号主义是从宏观上模拟人的思维过程的话，那么联结主义则试图从微观上解决人类的认知功能，以探索认知过程的微观结构。

- 该学派将智能理解为相互联结的神经元竞争与协作的结果，以人工神经网络为代表。近年来深度神经网络的发展与应用，掀起了联结主义学派的研究热潮，其代表性领域是计算机视觉。

人工智能各学派的认知观

- 行为主义（Actionism）学派：来源于控制论及“感知——动作”型控制系统。该学派认为智能取决于感知和行动，人工智能可以像人类智能一样逐步进化，以及智能行为只能在现实世界中与周围环境交互作用而表现出来。

- 行为主义学派的标志性应用包括：波士顿动力公司（Boston Dynamics）研发的仿生机器人，以及谷歌公司的机器狗。

知识表示和推理

- 命题/谓词逻辑：语法和语义
- 推理程序的合理性和完备性
- 将谓词公式转换成子句形式
- 合一和MGU
- 归结证明 [作业]
- 答案抽取

知识表示和推理

第一次理论课作业：1.5 任何通过了历史考试并中了彩票的人都是快乐的。任何肯学习或幸运的人都可以通过所有考试，小张不学习，但很幸运，任何人只要是幸运的，就能中彩。求证：小张是快乐的。

知识表示和推理

第一次理论课作业：1.5 任何通过了历史考试并中了彩票的人都是快乐的。任何肯学习或幸运的人都可以通过所有考试，小张不学习，但很幸运，任何人只要是幸运的，就能中彩。求证：小张是快乐的。

- 第一步：定义谓词，将待证明的问题的前提条件和逻辑结论用谓词公式表示出来。
- 第二步：将所有规则与事实及求证目标的否定化成子句集。
- 第三步：利用归结原理对子句集 S 中的子句进行归结。

搜索技术

- 搜索的性质：完备性、最优性、时间复杂度、空间复杂度
- 盲目搜索：宽度优先、深度优先、一致代价、深度受限、迭代加深
- 启发式搜索：(贪婪)最佳优先、A* [作业]
- 博弈树搜索：alpha-beta剪枝 [作业]

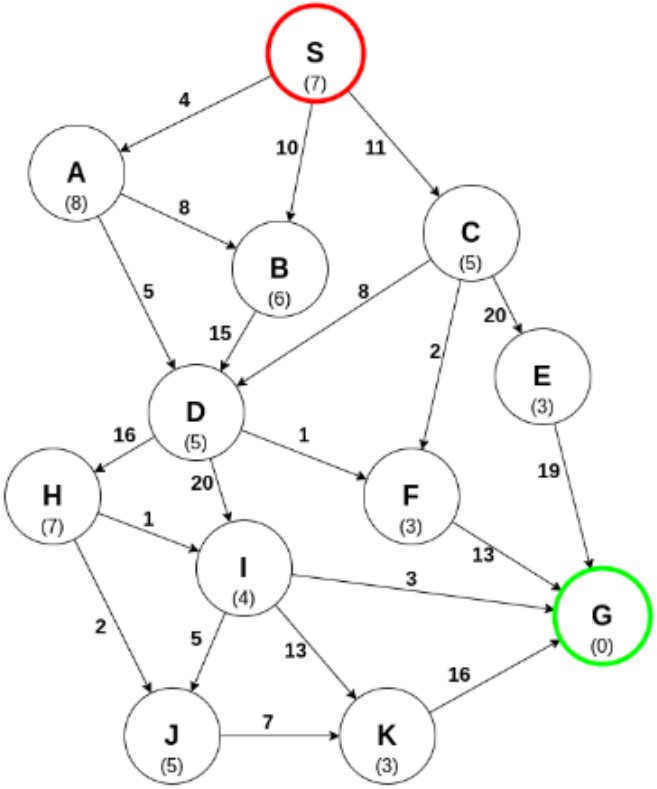
A*搜索

- A*搜索：评价函数$f(n) = g(n) + h(n)$
- $g(n)$是从初始节点到达节点n的路径成本
- $h(n)$是从n节点到达目标节点的成本的启发式估计值
- 因此，$f(n)$是经过节点n从初始节点到达目标节点的路径成本的估计值
- 利用节点对应的$f(n)$值来对边界上的节点进行排序

$$\underbrace{\boldsymbol{f(n)}}_{\text{评价函数}} = \underbrace{\boldsymbol{g(n)}}_{\substack{\text{起始节点到节点}n\text{代价}\\ \text{(当前最小代价)}}} + \underbrace{\boldsymbol{h(n)}}_{\substack{\text{节点}n\text{到目标节点代价}\\ \text{(后续估计最小代价)}}}$$

A*搜索

- 第二次理论课作业：1.1 用A*搜索算法求解最短路径，其中S表示初始节点，G表示目标节点。相邻节点之间的代价用边上的数字表示，在每个节点中括号里的数字表示此节点到目标节点的代价。

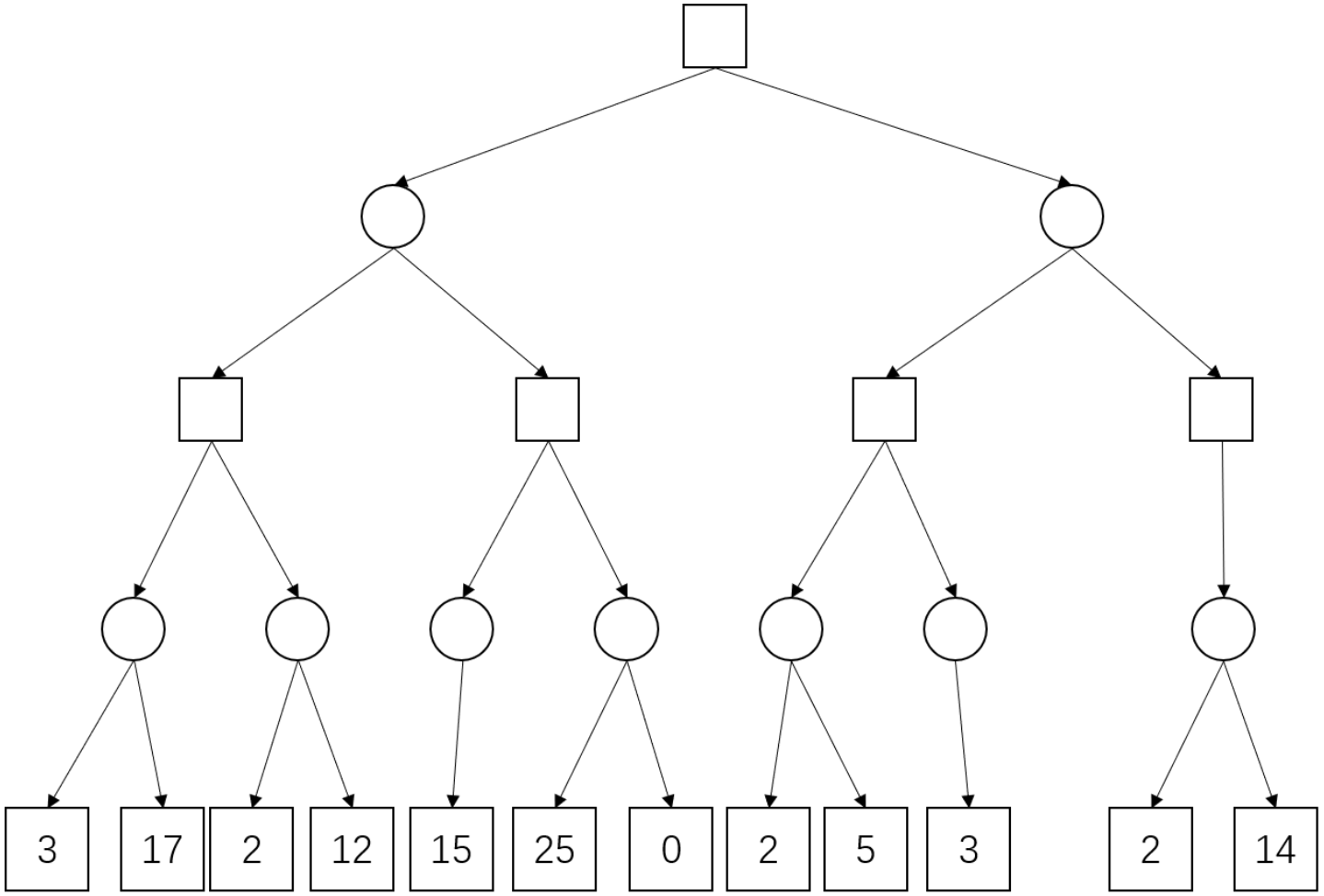


注：A*搜索算法有两种策略，一是根据边界中每个节点的 f 值，选择具有最小 f 值的节点并检测该节点是否为目标节点。如果上述节点不是目标节点，则将该节点从边界中删除，同时将它的所有后继(子)节点加入边界中；二是从初始节点出发，首先判断上述节点是否为目标节点，如果不是，则逐一获得该节点的后继(子)节点，并检测当前后继(子)节点是否为目标节点，如果不是，则将其加入边界中。

对于上述任何一种策略，如果没有给定目标节点，则都需要清空边界(即每个节点都探索一次)，算法才能结束。

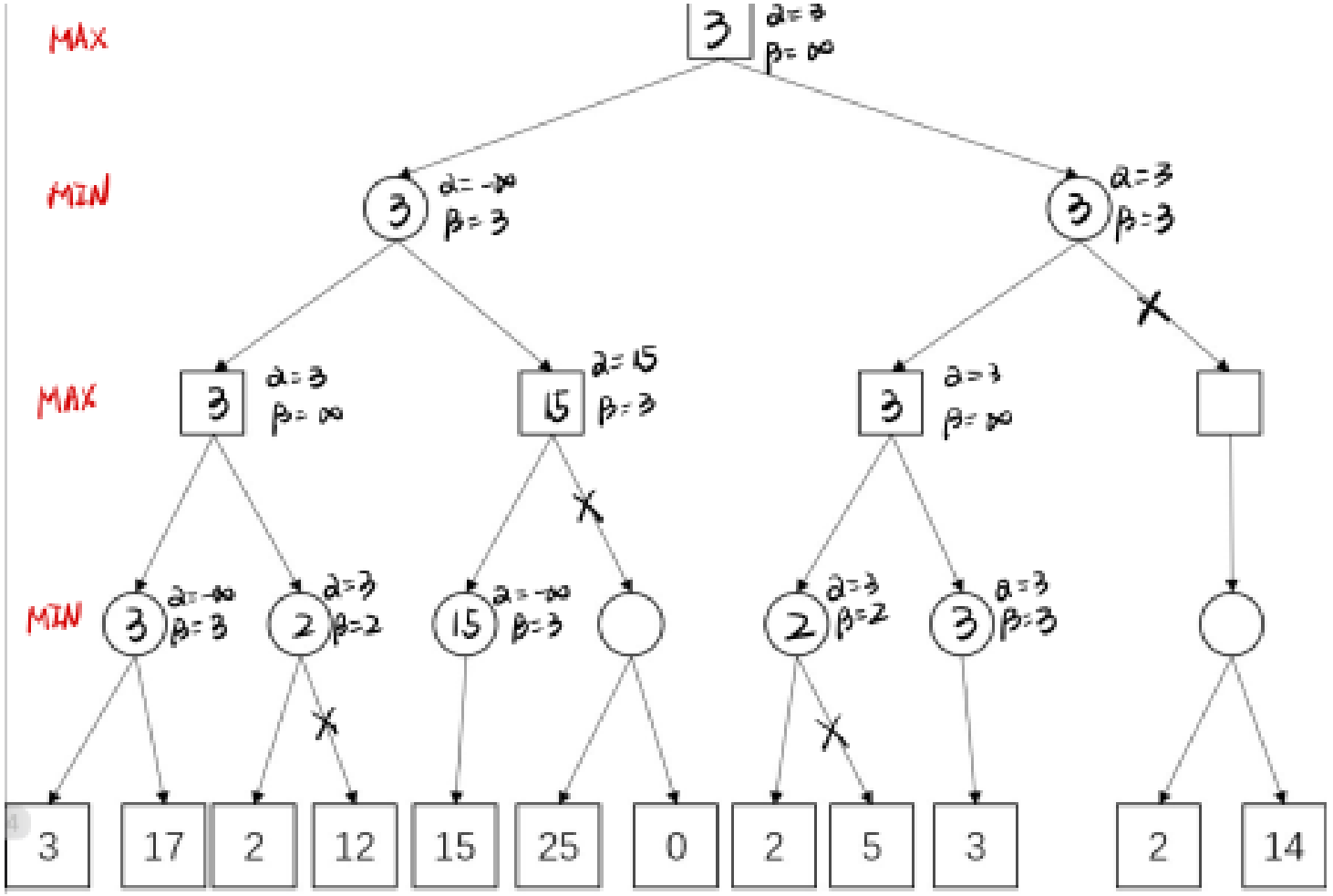
博弈树搜索

- 第二次理论课作业：1.2 在下图所示的博弈树中，进行$\alpha-\beta$剪枝搜索，写出算法过程。



博弈树搜索

- 第二次理论课作业：1.2 在下图所示的博弈树中，进行$\alpha-\beta$剪枝搜索，写出算法过程。

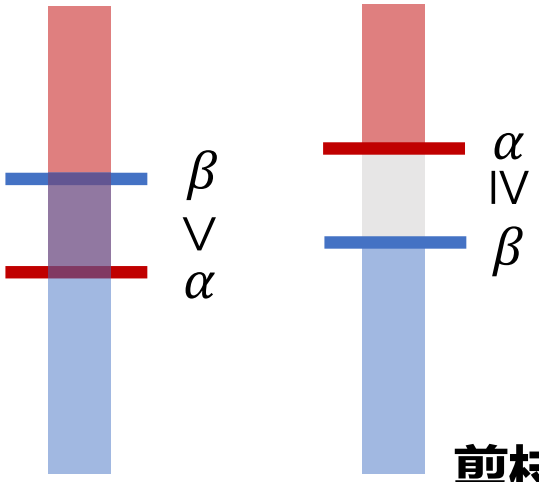


博弈树搜索

算法理解:

搜索到当前节点时，已知

- **Max**玩家的得分**下界**
- **Min**玩家的得分**上界**



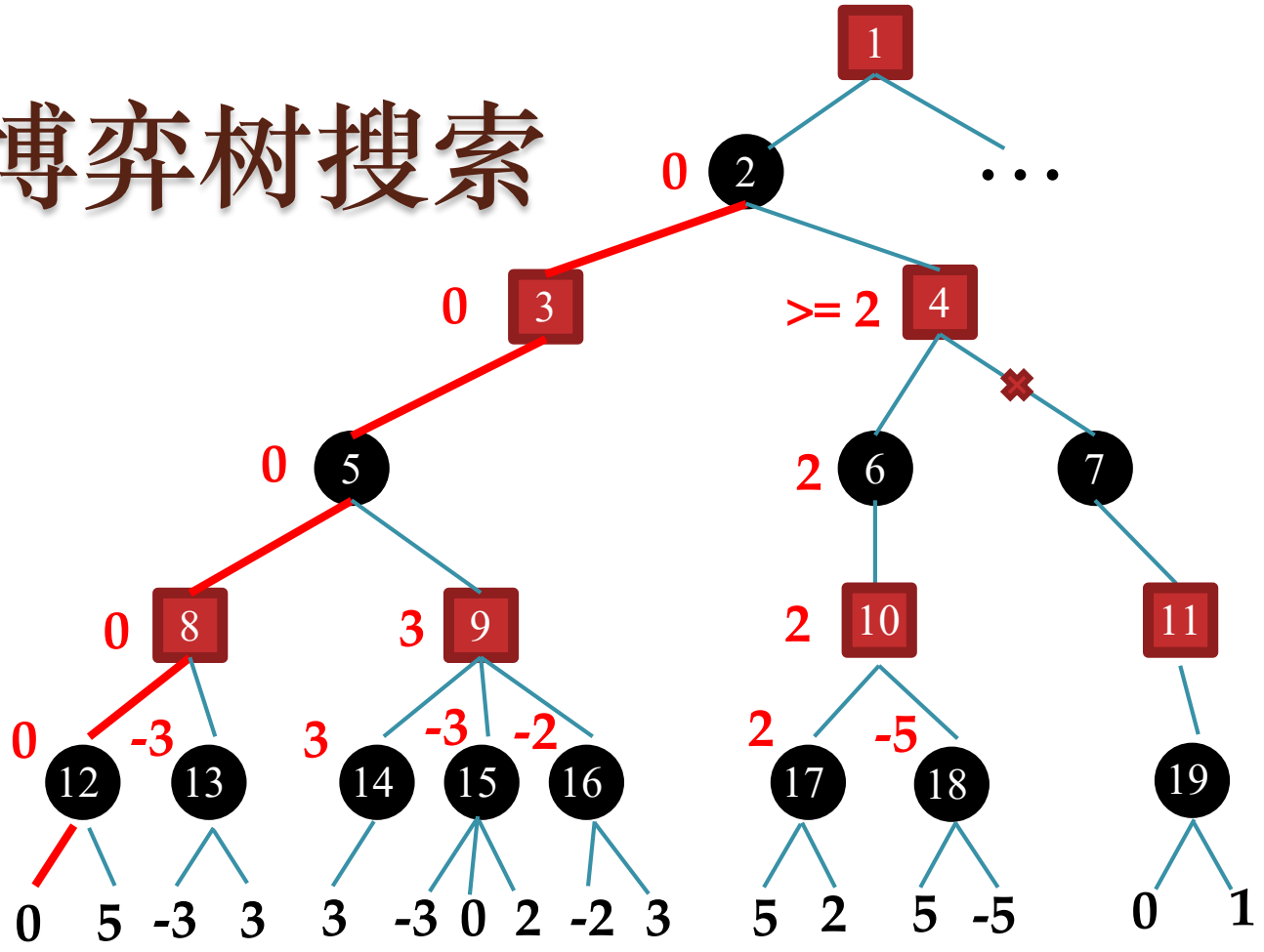
局势未确定

剪枝:
因为局势已经确定

算法关键点:

1. **深度**优先搜索
2. **向下搜索**：**传递** α **和** β 值
3. **向上回传**：**更新** α **或** β 值
4. **剪枝**：$\alpha \geq \beta$

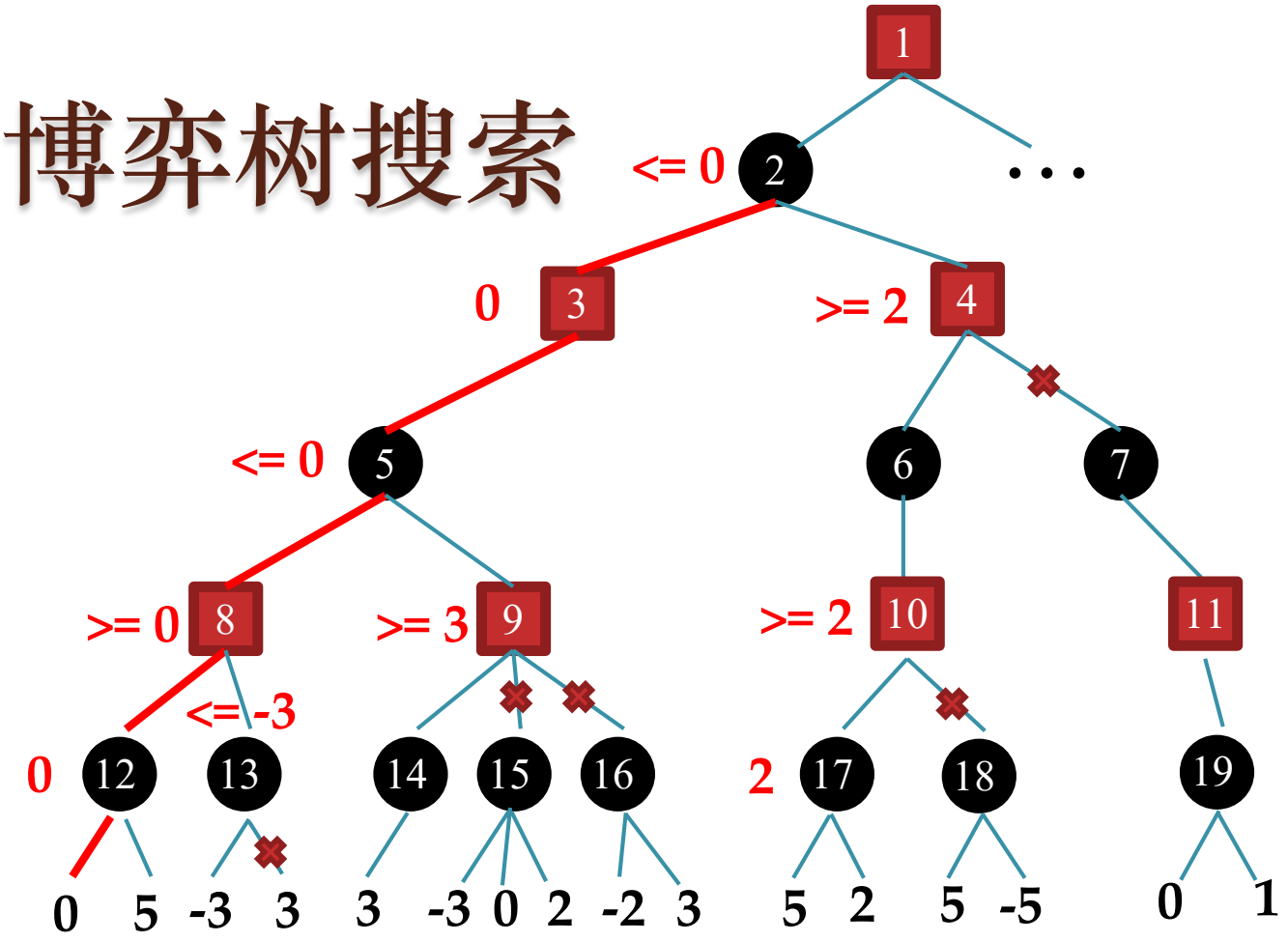
博弈树搜索



$\beta(12) = 0$ => $U(12) = 0$, $\alpha(8) = 0$ => $\beta(13) = -3$ <= $\alpha(8)$ 对Min节点13执行β剪枝
$\beta(5) = 0$ => $U(14) = 3$, $\alpha(9) = 3$ >= $\beta(5)$ 对Max节点9执行α剪枝
$\beta(2) = 0$ => $\beta(17) = 5$, $U(17) = 2$, $\alpha(10) = 2$ >= $\beta(2)$ 对Max节点10执行α剪枝
$\alpha(4) = 2$ >= $\beta(2)$ 对Max节点4执行α剪枝

（1）深度优先搜索；（2）在通过MiniMax算法向上传播效益值的过程中，更新α值或β值记录Max节点的α值和Min节点的β值，它们分别表示Max方的最小得分和Min方的最大得分

博弈树搜索



$\beta(12) = 0$ => $U(12) = 0$, $\alpha(8) = 0$ => $\beta(13) = -3$ <= $\alpha(8)$ 对Min节点13执行β剪枝

$\beta(5) = 0$ => $U(14) = 3$, $\alpha(9) = 3$ >= $\beta(5)$ 对Max节点9执行α剪枝

$\beta(2) = 0$ => $\beta(17) = 5$, $U(17) = 2$, $\alpha(10) = 2$ >= $\beta(2)$ 对Max节点10执行α剪枝

$\alpha(4) = 2$ >= $\beta(2)$ 对Max节点4执行α剪枝

（1）深度优先搜索；（2）在通过MiniMax算法向上传播效益值的过程中，更新α值或β值记录Max节点的α值和Min节点的β值，它们分别表示Max方的最小得分和Min方的最大得分

高级搜索

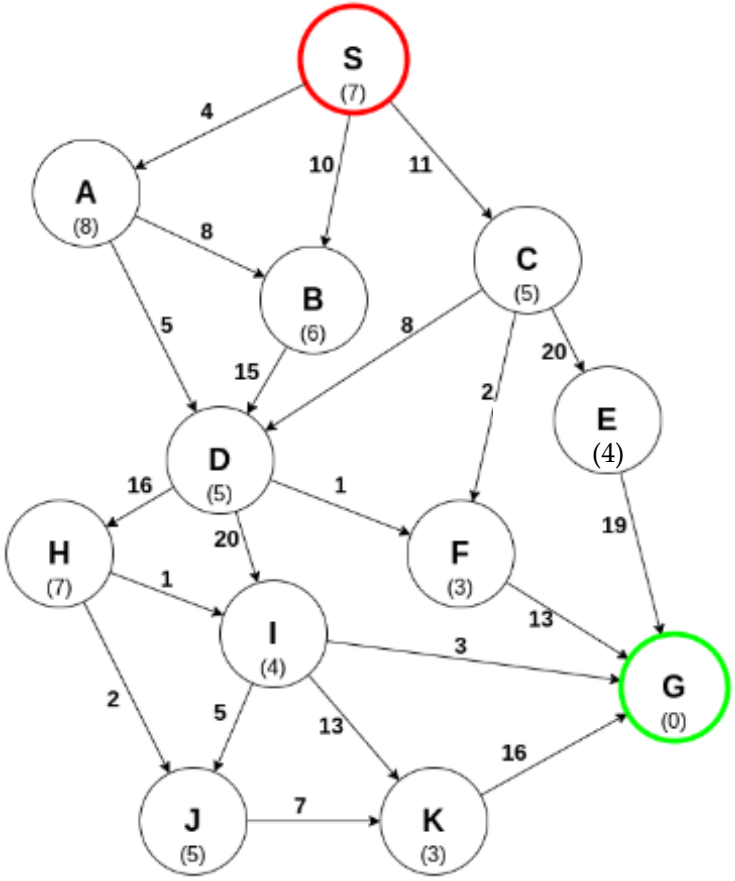
- 爬山法搜索
- 模拟退火算法
- 遗传算法

爬山法搜索

- 搜索算法在内存中保留一条或多条路径并且记录哪些是已经探索过的，哪些是还没有探索过的。当找到目标时，到达目标的路径同时也构成了这个问题的一个解。
- 爬山法搜索：贪婪局部搜索
- 登高：一直向值增加的方向持续移动，将会在到达一个“峰顶”时终止，并且在相邻状态中没有比它更高的值。这个算法不维护搜索树，因此当前节点的数据结构只需要记录当前状态和它的目标函数值。

爬山法搜索

- 用**爬山法搜索**算法求解搜索路径，其中S表示初始节点，G表示目标节点。相邻节点之间的**路径代价用边上的数字表示**，在每个节点中括号里的数字表示此节点到目标节点的**预估代价**，通常记为**h(n)**。



第一步：

	A	B	C
h(n)	8	6	5

第二步：

	D	E	F
h(n)	5	4	3

第三步：

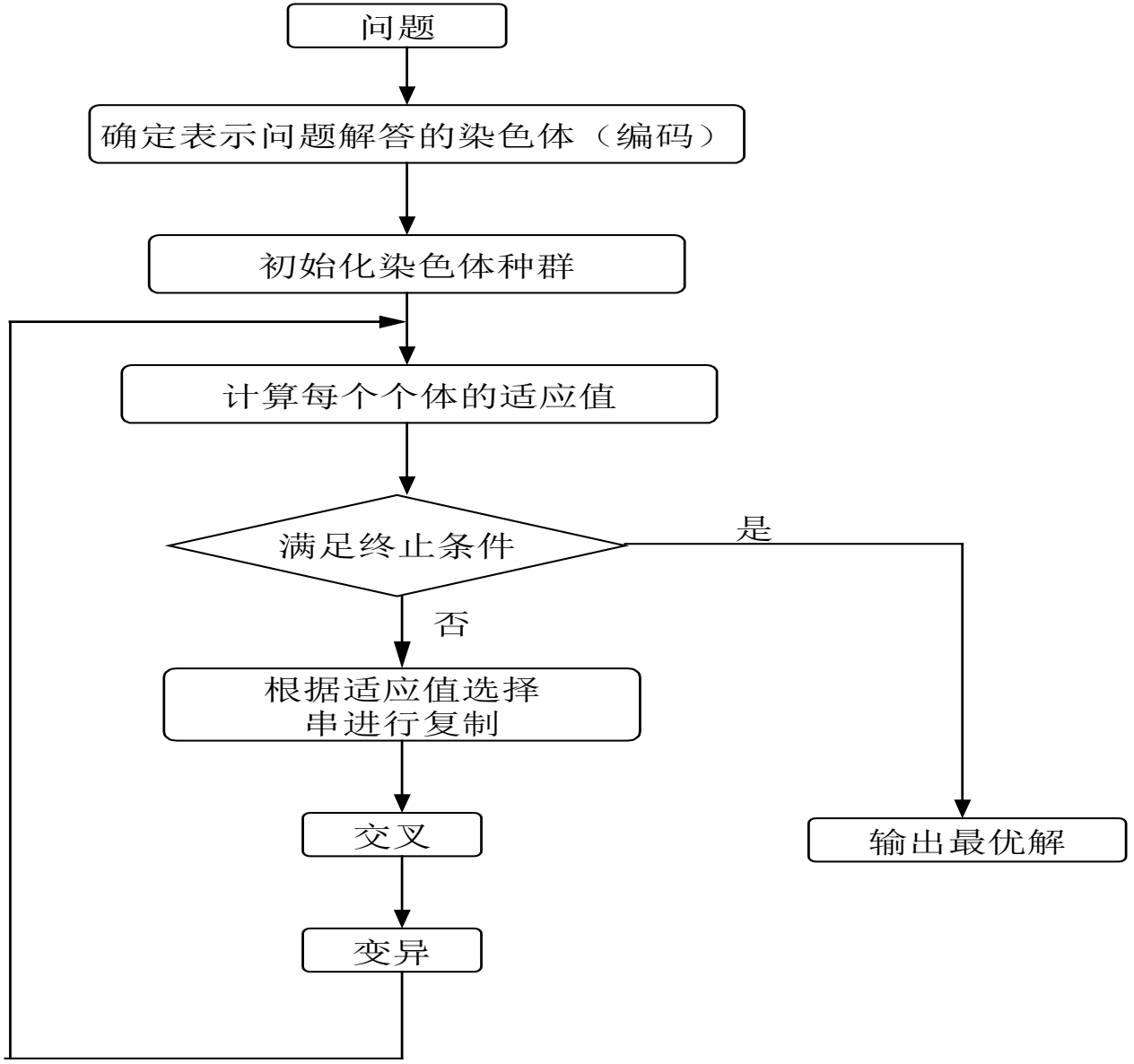
	G
h(n)	0

路径： S→C →F →G

模拟退火算法思想

- 模拟退火算法概述
  - 模拟退火算法（Simulated Annealing，SA）是一种模拟物理退火的过程而设计的优化算法。它的基本思想最早在1953年就被Metropolis提出，但直到1983年Kirkpatrick等人才设计出真正意义上的模拟退火算法并进行应用。
- 模拟退火算法思想
  - 模拟退火算法采用类似于物理退火的过程，先在一个高温状态下（相当于算法随机搜索，大概率接受劣解），然后逐渐退火，徐徐冷却（接受劣解概率变小直至为零，相当于算法局部搜索），最终达到物理基态（相当于算法找到最优解）。算法的本质是通过温度来控制算法接受劣解的概率（劣向转移是脱出局部极小的核心机制）。

遗传算法的一般步骤


问题
确定表示问题解答的染色体（编码）
初始化染色体种群
计算每个个体的适应值
满足终止条件
是
否
根据适应值选择
串进行复制
交叉
变异
输出最优解

编码

- 位串编码
  - 一维染色体编码方法：将问题空间的参数编码为一维排列的染色体的方法。
  - 例如，用若干二进制数表示一个个体，将原问题的解空间映射到位串空间 B = {0，1}上，然后在位串空间上进行遗传操作。
  - 上述二进制编码的优点：类似于生物染色体的组成，算法易于用生物遗传理论解释，遗传操作如交叉、变异等易实现。
  - 缺点：（1）相邻整数的二进制编码可能具有较大的Hamming距离，降低了遗传算子的搜索效率；（2）要先给出求解的精度；（3）求解高维优化问题的二进制编码串长，搜索效率低。

15：01111　　16：10000

编码

- 位串编码
  - 多参数映射编码的基本思想：把每个参数先进行二进制编码得到子串，再把这些子串连成一个完整的染色体。
  - 多参数映射编码中的每个子串对应各自的编码参数，所以可以有不同的串长度和参数的取值范围。
- 实数编码
  - 采用实数表达法不必进行数制转换。

群体设定

- 种群规模的确定
  - 规模太小，遗传算法的优化性能不太好，易陷入局部最优解。
  - 规模太大，计算复杂。
- 初始种群的产生
  - 随机产生种群规模数目的个体作为初始种群。
  - 随机产生一定数目的个体，从中挑选最好的个体加到初始种群中。不断迭代，直到初始群体中个体数目达到了预先确定的规模。
  - 根据问题固有知识，把握最优解所占空间在整个问题空间中的分布范围，然后，在此分布范围内设定初始种群。

适应度函数

1. 将目标函数映射成适应度函数的方法

若目标函数为最大化问题，则 $Fit(f(x)) = f(x)$

若目标函数为最小化问题，则 $Fit(f(x)) = \dfrac{1}{f(x)}$

将目标函数转换为求最大值的形式，且保证函数值非负！

若目标函数为最大化问题，则

$$Fit(f(x)) = \begin{cases} f(x) - C_{\min} & f(x) > C_{\min} \\ 0 & \text{其他情况} \end{cases}$$

若目标函数为最小化问题，则

$$Fit(f(x)) = \begin{cases} C_{\max} - f(x) & f(x) < C_{\max} \\ 0 & \text{其他情况} \end{cases}$$

适应度函数

2. 适应度函数的尺度变换

- 在遗传算法中，将所有妨碍适应度值高的个体产生，从而影响遗传算法正常工作的问题统称为欺骗问题。
- 过早收敛：缩小这些个体的适应度，以降低这些超级个体的竞争力。
- 停滞现象：改变原始适应值的比例关系，以提高个体之间的竞争力。
- 尺度变换或定标：对适应度函数值域的某种映射变换。

选择

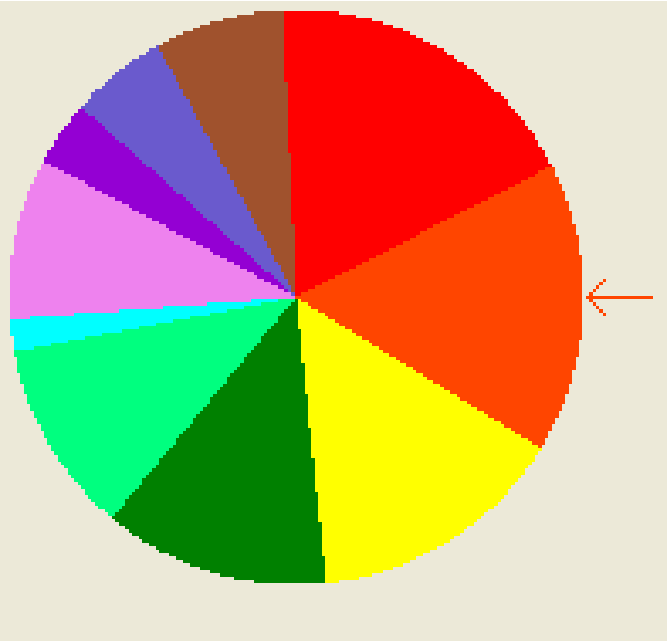
- 选择是用来确定重组或交叉个体，以及被选个体将产生多少个子代个体。首先计算适应度：
  (1) 按比例的适应度计算；
  (2) 基于排序的适应度计算。
- 适应度计算之后是实际的选择，按照适应度进行父代个体的选择。

- 可以挑选以下的算法：
  ① 轮盘赌选择；　② 锦标赛选择；
  ③ 最佳个体保存；　④ 随机遍历抽样。

选择

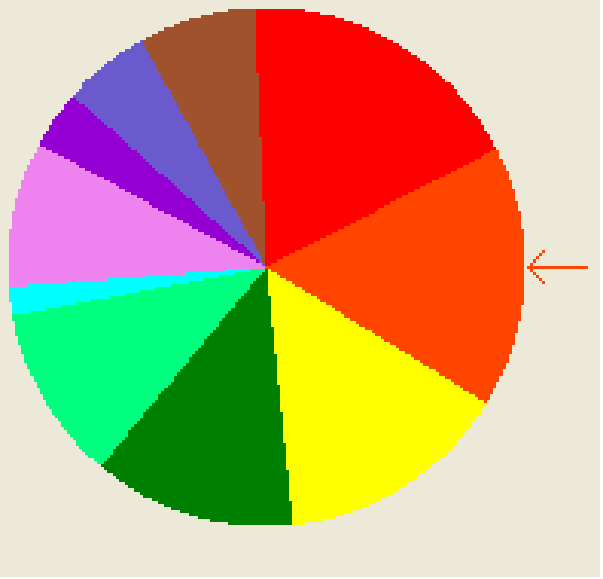
（1）轮盘赌选择

- 按个体的选择概率产生一个轮盘，轮盘每个区的角度与个体的选择概率成比例。
- 产生一个随机数，它落入轮盘的哪个区域就选择相应的个体交叉。



选择

（1）**轮盘赌选择**



个体	1	2	3	4	5	6	7	8	9	10	11
适应度	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2	0.1
选择概率	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02	0.0
累积概率	0.18	0.34	0.49	0.62	0.73	0.82	0.89	0.95	0.98	1.00	1.00

第1轮产生一个随机数：**0.81**　　第2轮产生一个随机数：**0.32**

选择

（2）**锦标赛选择**（tournament selection model）

- 锦标赛选择方法：从群体中随机选择个个体，将其中适应度最高的个体保存到下一代。这一过程反复执行，直到保存到下一代的个体数达到预先设定的数量为止。

- 随机竞争方法（stochastic tournament）：每次按轮盘赌选择方法选取一对个体，然后让这两个个体进行竞争，适应度高者获胜。如此反复，直到选满为止。

选择

（3）最佳个体保存

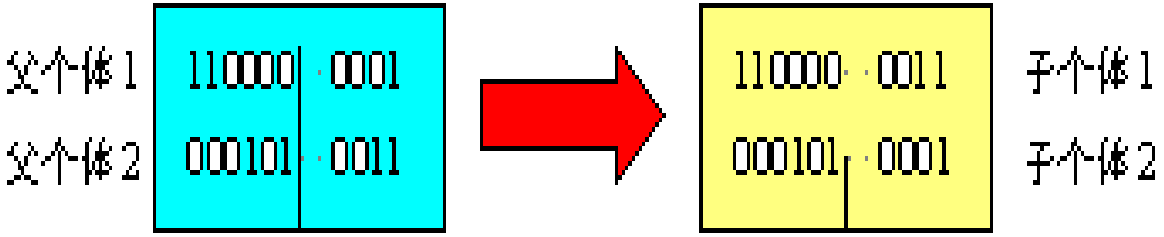
- 最佳个体（elitist model）保存方法：把群体中适应度最高的个体不进行交叉而直接复制到下一代中，保证遗传算法终止时得到的最后结果一定是历代出现过的最高适应度的个体。

交叉

1. 基本的交叉算子

（1）单点交叉（single-point crossover）

- 在个体串中随机设定一个交叉点，实行交叉时，该点前或后的两个个体的部分结构进行互换，并生成两个新的个体。



（2）两点交叉 （two-point crossover）

- 随机设置两个交叉点，将两个交叉点之间的码串相互交换。

交叉

2. 修正的交叉方法

部分匹配交叉PMX：Goldberg D. E.和R.Lingle (1985)

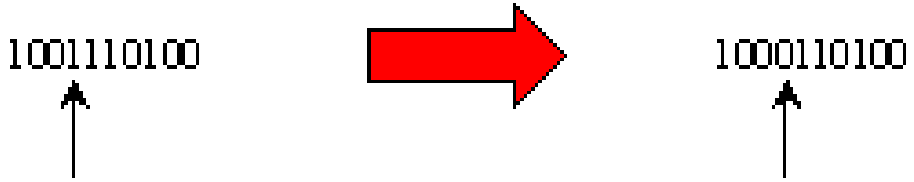
$$A = 9 \quad 8 \quad 4 \;\Big|\; 5 \quad 6 \quad 7 \;\Big|\; 1 \quad 3 \quad 2$$

$$B = 8 \quad 7 \quad 1 \;\Big|\; 2 \quad 3 \quad 9 \;\Big|\; 5 \quad 4 \quad 6$$

$$A' = 9 \quad 8 \quad 4 \;\Big|\; 2 \quad 3 \quad 9 \;\Big|\; 1 \quad 3 \quad 2$$

$$B' = 8 \quad 7 \quad 1 \;\Big|\; 5 \quad 6 \quad 7 \;\Big|\; 5 \quad 4 \quad 6$$

变异



- 位点变异：群体中的个体码串，随机挑选一个或多个基因座，并对这些基因座的基因值以变异概率作变动。
- 逆转变异：在个体码串中随机选择两点（逆转点），然后将两点之间的基因值以逆向排序插入到原位置中。
- 插入变异：在个体码串中随机选择一个码，然后将此码插入随机选择的插入点中间。
- 互换变异：随机选取染色体的两个基因进行简单互换。
- 移动变异：随机选取一个基因，向左或向右移动一个随机位数。

不确定知识表示和推理

- 给定变量$\{X_1,\ldots,X_n\}$，构建一个有向无环图的步骤如下：
  - 步骤1、在某种变量顺序下，对所有变量的联合概率应用链式法则：

    $$Pr(X_1,\ldots,X_n) = Pr(X_n|X_1,\ldots, X_{n-1})Pr(X_{n-1}|X_1,\ldots, X_{n-2})\ldots Pr(X_1)$$

  - 步骤2、对于每个变量X_i，考虑该变量的条件集合$X_1,\ldots,X_{i-1}$，采用如下方法递归地判断条件集合中的每个变量X_j是否可以删除：如果给定其余变量的集合，X_i和X_j是条件独立的，则将X_j从X_i的条件集合中删除。经过这一步骤，可以得到下式(结构化分解)：

    $$Pr(X_1,\ldots,X_n) = Pr(X_n|Par(X_n))Pr(X_{n-1}|Par(X_{n-1}))\ldots Pr(X_1)$$

  - 步骤3、基于上述公式，构建一个有向无环图。其中，对于每个用节点表示的变量X_i，其父节点为$Par(X_i)$中的变量集合。
  - 步骤4、为每个变量及其父节点集合确定条件概率表的取值。

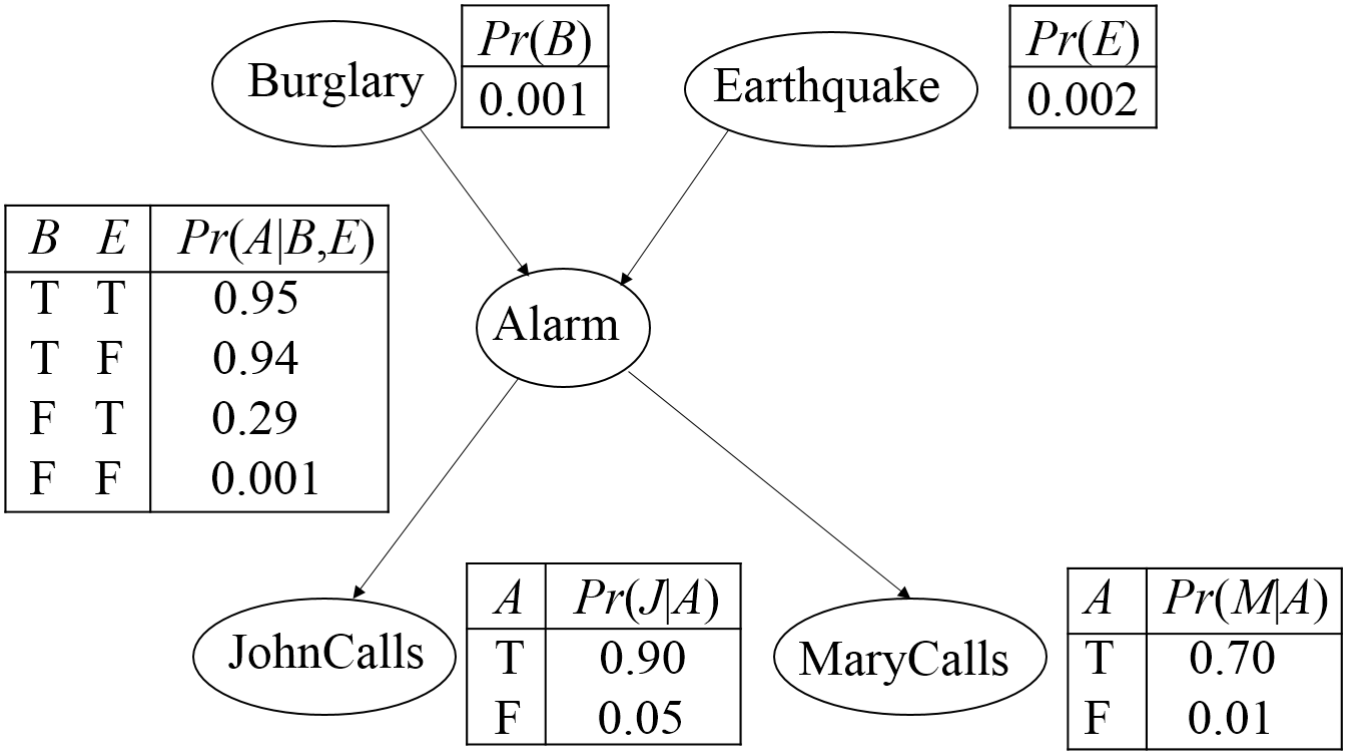
不确定知识表示和推理

- 对于一个有向无环图，D-分离（D-Separation）是一种用来判断其变量是否条件独立的图形化方法。
- 在基于贝叶斯网络的不确定性知识推理中，采用D-分离方法可以简化概率计算，提高运行速度。
- 示例：
  - 小偷（Burglar）会引发警报（Alarm）
  - 地震（Earthquake）会引发警报（Alarm）
  - 警报（Alarm）会引起邻居约翰打电话（JohnCalls）
  - 警报（Alarm）会引起邻居玛丽打电话（MaryCalls）

S. Russell, P. Norvig. *Artificial Intelligence: A Modern Approach*. 3rd edition, Prentice Hall, 2009.

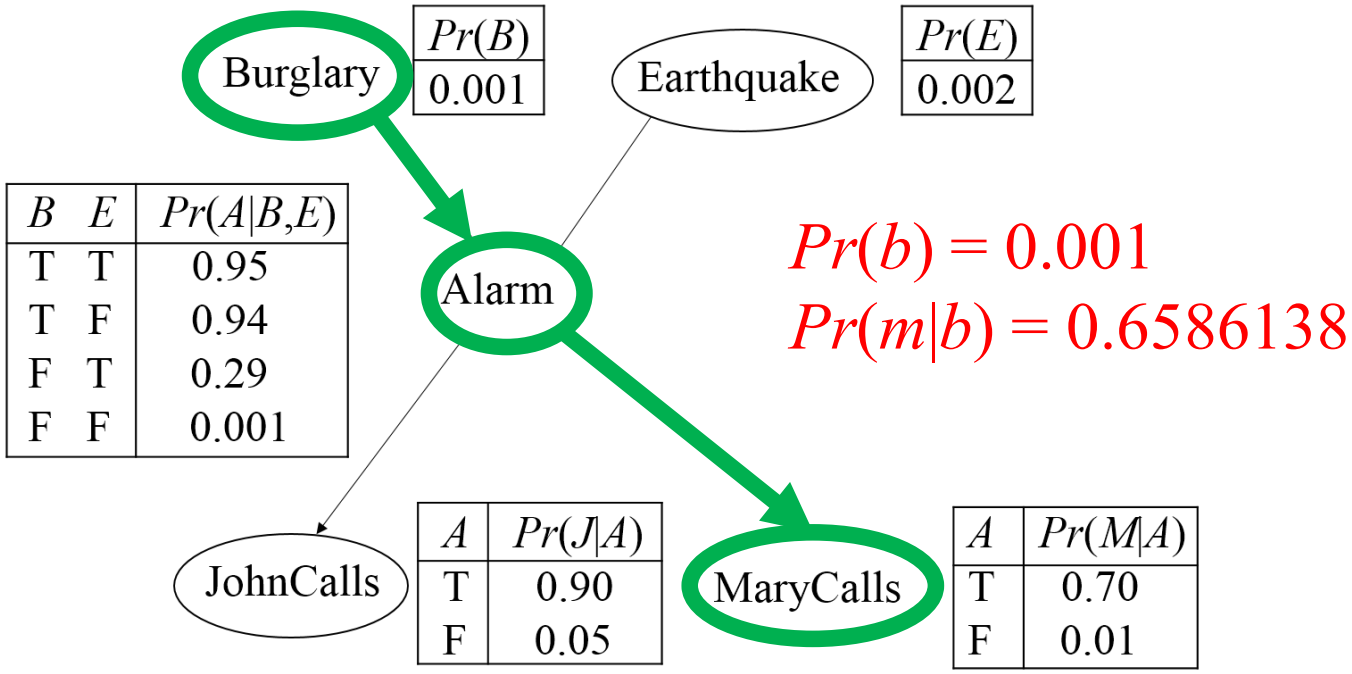
不确定知识表示和推理

- 示例：



不确定知识表示和推理

- 间接的因果/证据作用：未给定A的前提下，B与M不独立。



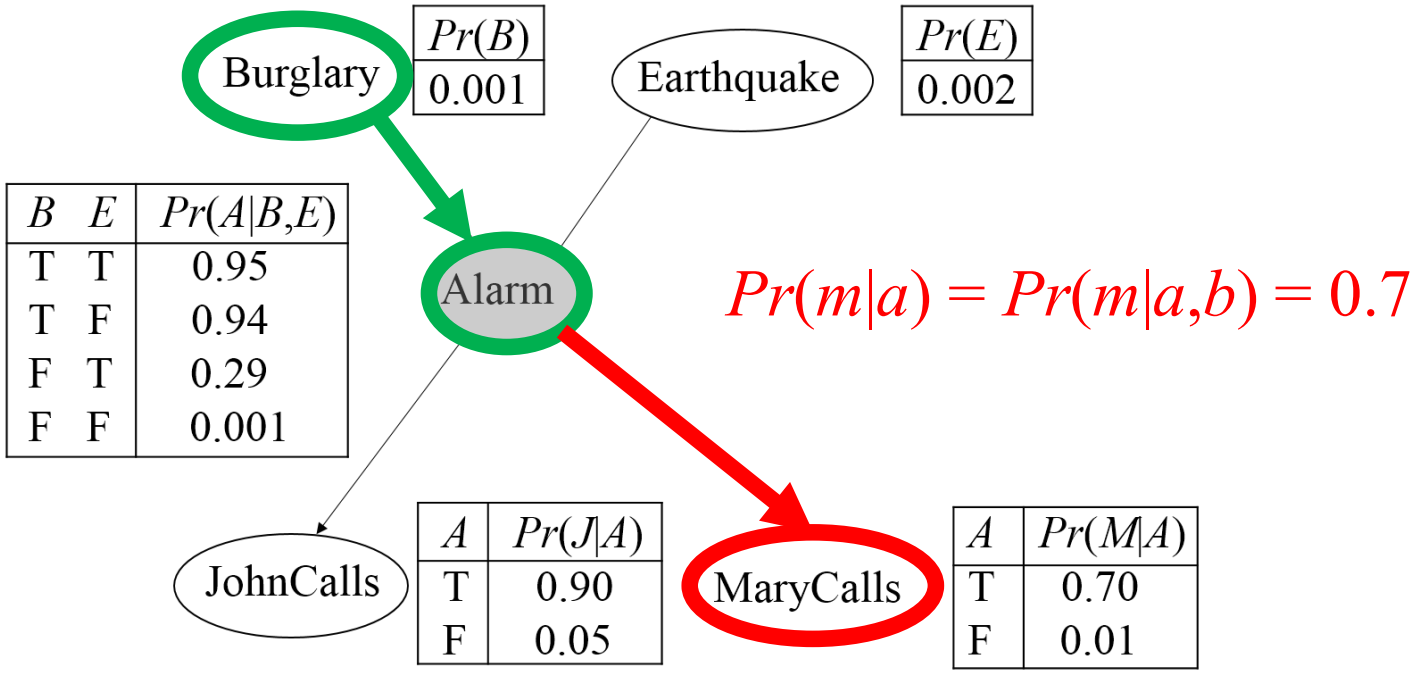
B	E	$Pr(A\|B,E)$
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

A	$Pr(J\|A)$
T	0.90
F	0.05

A	$Pr(M\|A)$
T	0.70
F	0.01

不确定知识表示和推理

- 间接的因果/证据作用：给定A的前提下，B与M条件独立。



$Pr(B)$
0.001

$Pr(E)$
0.002

B	E	$Pr(A\|B,E)$
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

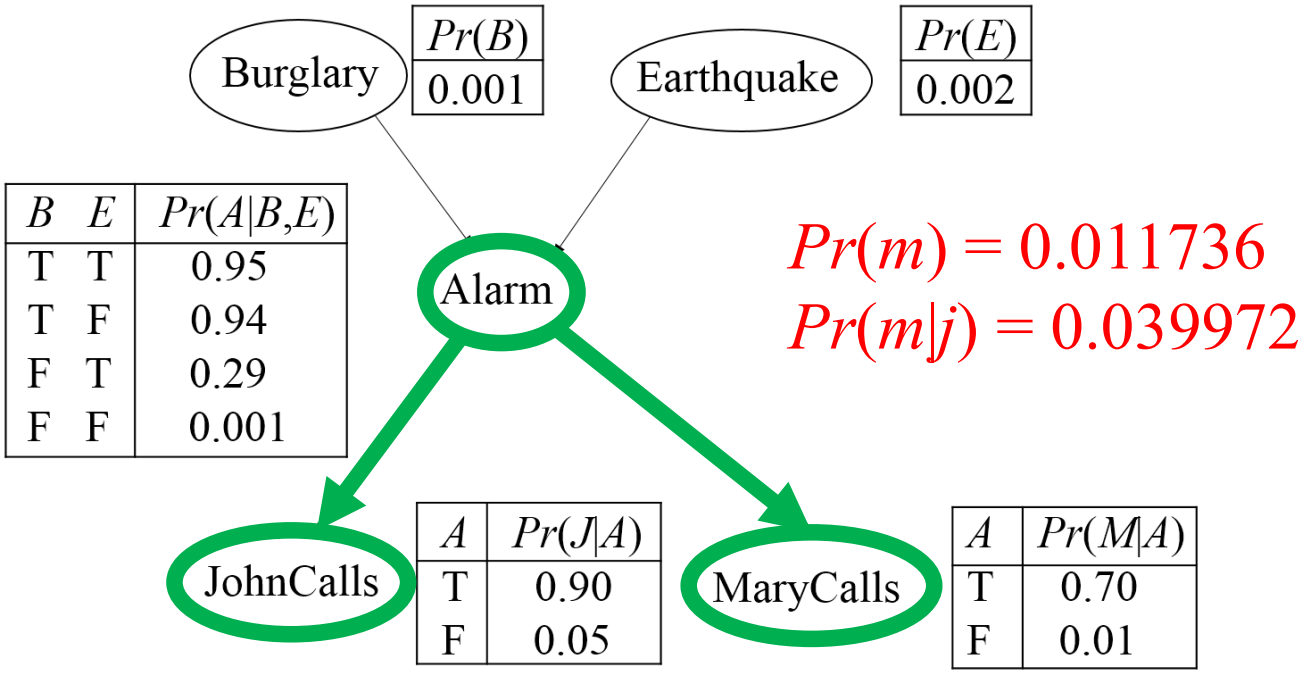
A	$Pr(J\|A)$
T	0.90
F	0.05

A	$Pr(M\|A)$
T	0.70
F	0.01

观察到中间节点时，路径阻塞

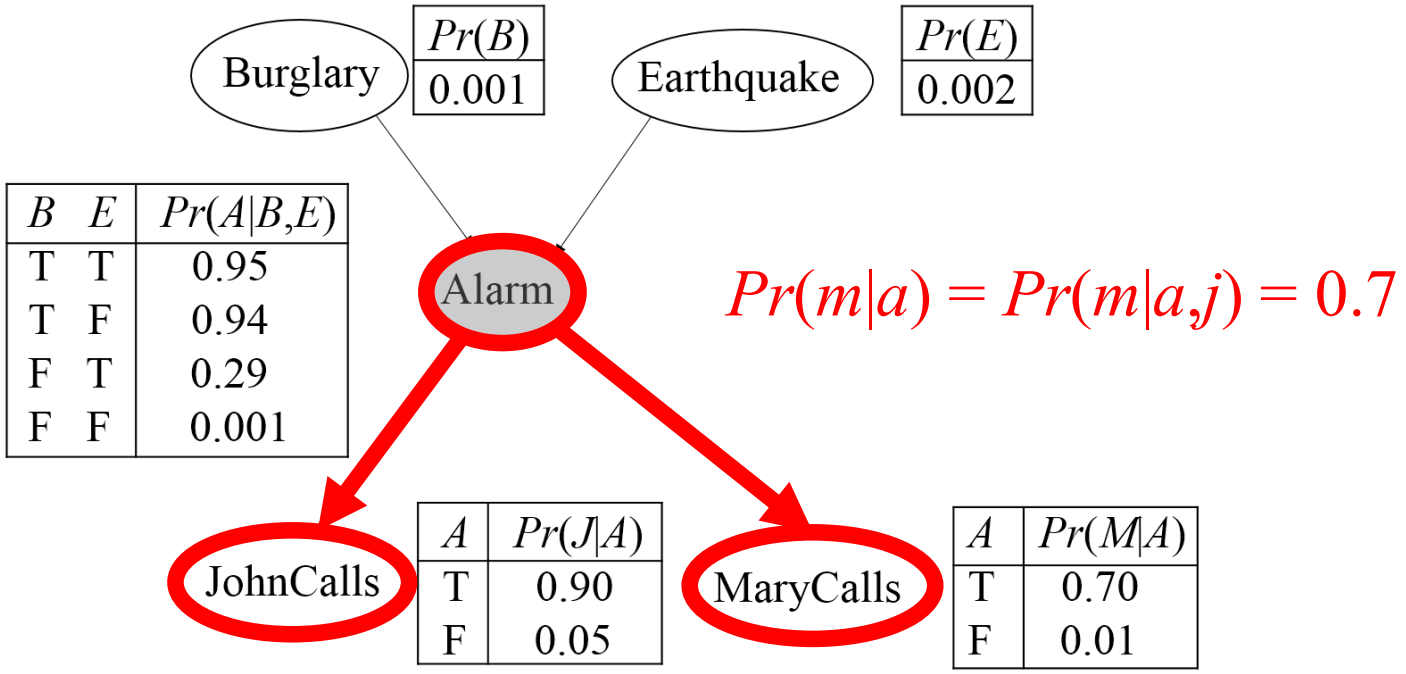
不确定知识表示和推理

- 共同的原因：未给定A的前提下，J与M不独立。



不确定知识表示和推理

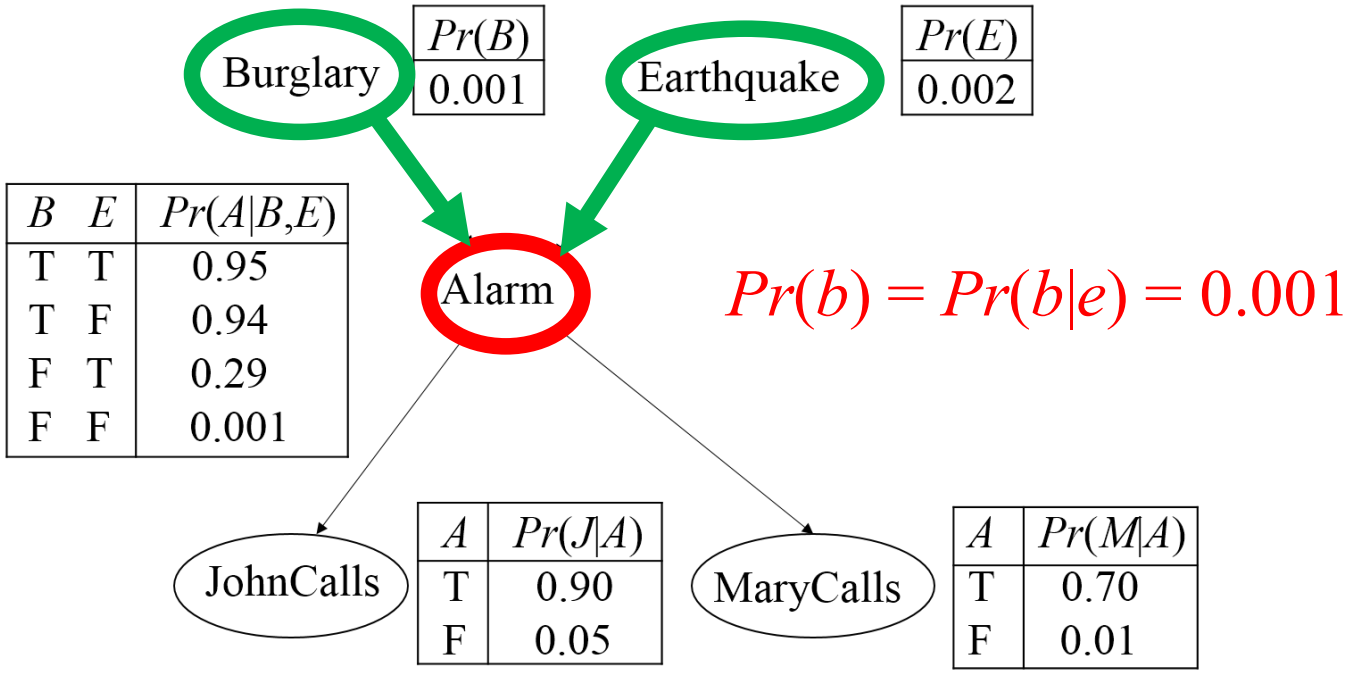
- 共同的原因：给定A的前提下，J与M条件独立。



观察到中间节点时，路径阻塞

不确定知识表示和推理

- 共同的作用：未给定A的前提下，B与E是独立的。



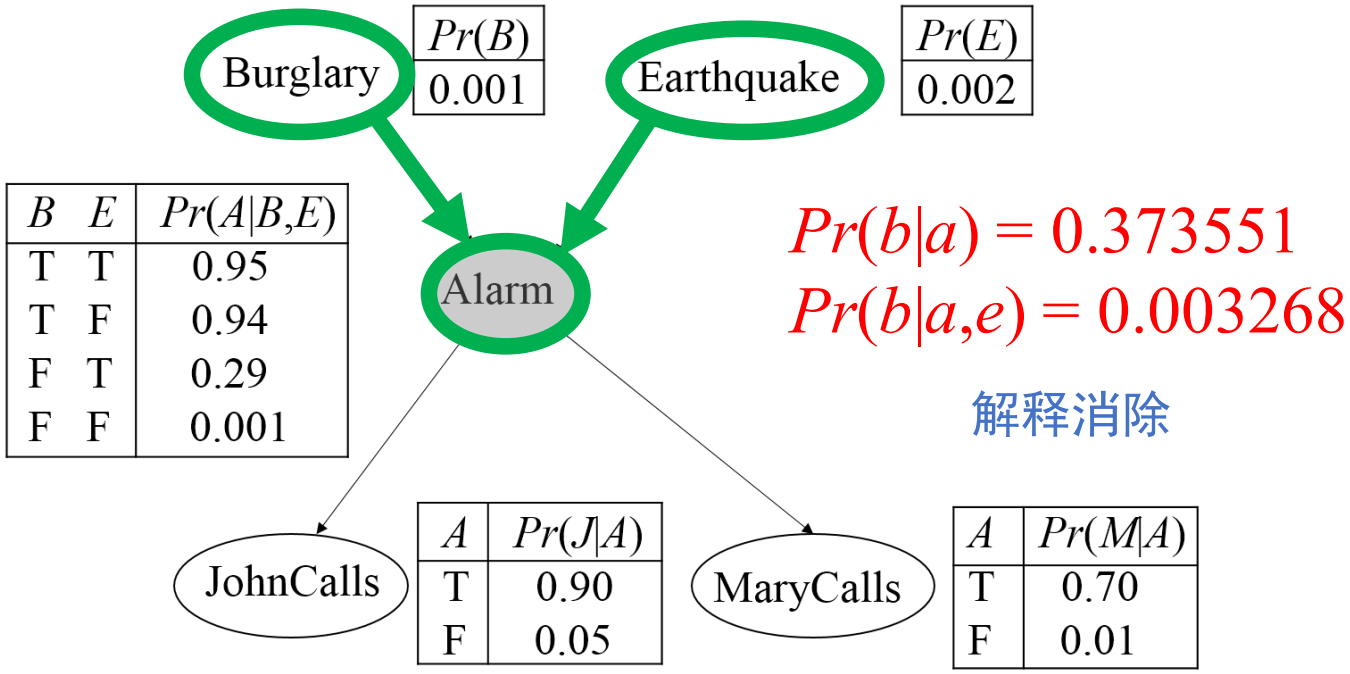
B	E	$Pr(A\|B,E)$
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

A	$Pr(J\|A)$
T	0.90
F	0.05

A	$Pr(M\|A)$
T	0.70
F	0.01

不确定知识表示和推理

- 共同的作用：给定A的前提下，B与E不是条件独立的。



B	E	$Pr(A\|B,E)$
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

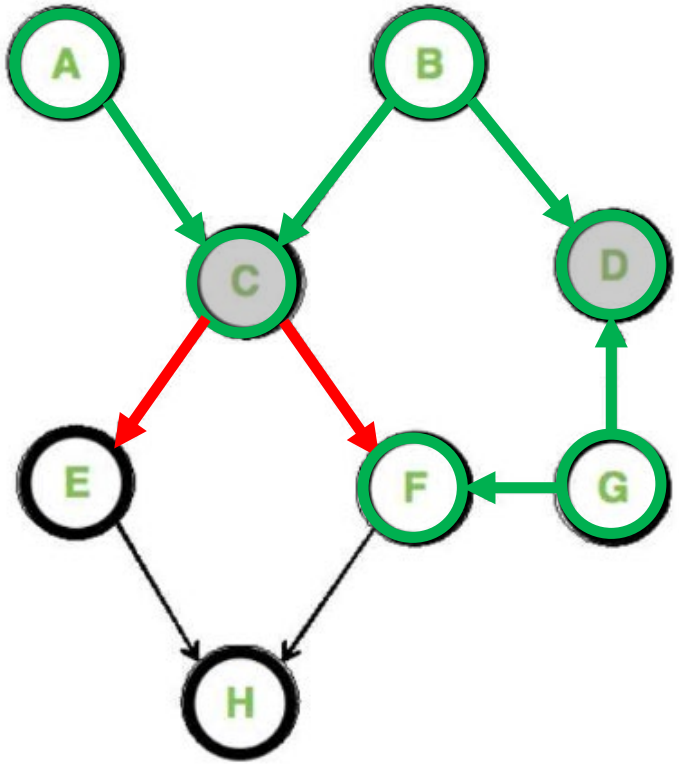
A	$Pr(J\|A)$
T	0.90
F	0.05

A	$Pr(M\|A)$
T	0.70
F	0.01

观察到中间节点时，路径连通

不确定知识表示和推理

- 第二次理论课作业：1.3 给定左下图所示的贝叶斯网络，回答问题。



(5) 给定C的前提下，E和F条件独立

(6) 给定C和D的前提下，E和F条件独立

(7) 给定C和H的前提下，A和F条件独立

(8) 给定C和D的前提下，A和F不独立

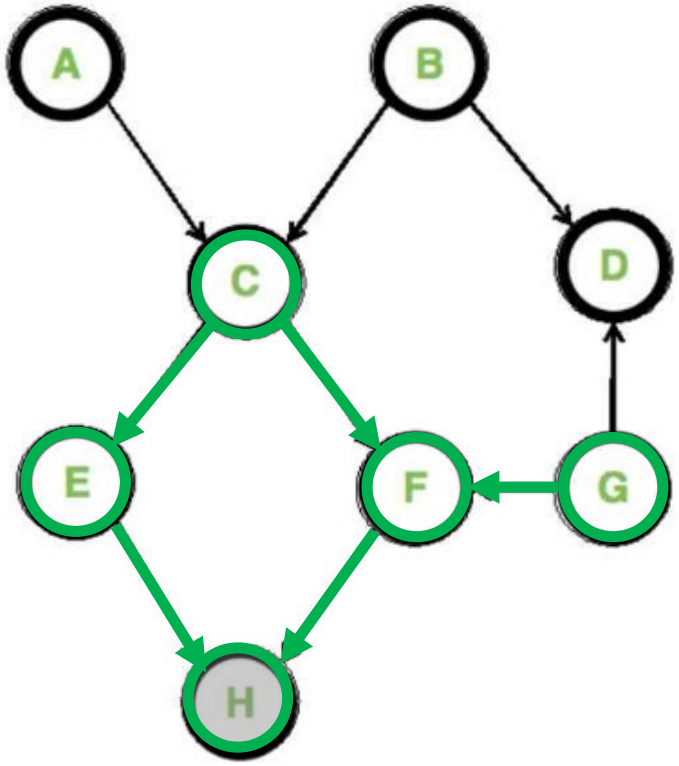
(9) 给定C和G的前提下，A和F条件独立

(10) 给定C的前提下，A和F条件独立

(11) 给定H的前提下，C和G不独立

不确定知识表示和推理

- 第二次理论课作业：1.3 给定左下图所示的贝叶斯网络，回答问题。



(5) 给定C的前提下，E和F条件独立

(6) 给定C和D的前提下，E和F条件独立

(7) 给定C和H的前提下，A和F条件独立

(8) 给定C和D的前提下，A和F不独立

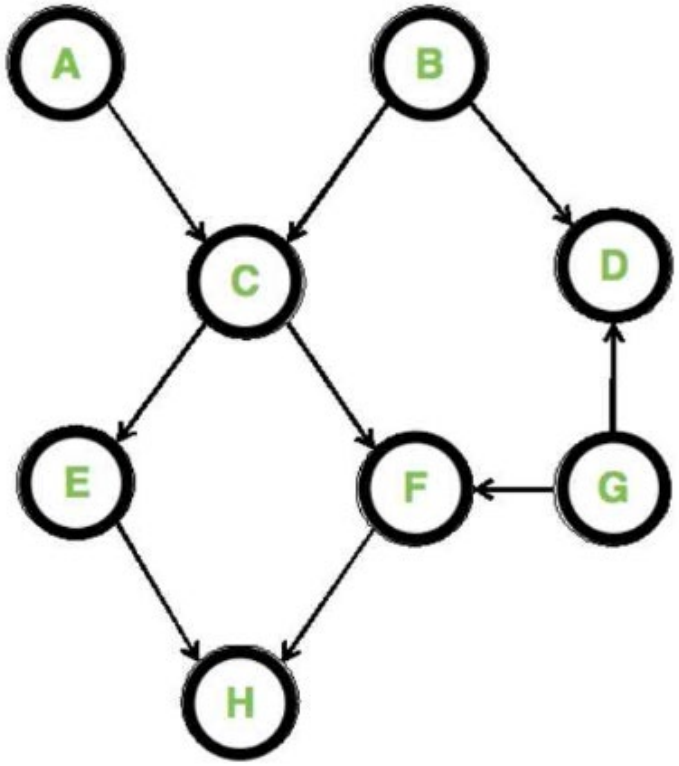
(9) 给定C和G的前提下，A和F条件独立

(10) 给定C的前提下，A和F条件独立

(11) 给定H的前提下，C和G不独立

不确定知识表示和推理

- 第二次理论课作业：1.3 给定左下图所示的贝叶斯网络，回答问题。



(5) 给定C的前提下，E和F条件独立

(6) 给定C和D的前提下，E和F条件独立

(7) 给定C和H的前提下，A和F条件独立

(8) 给定C和D的前提下，A和F不独立

(9) 给定C和G的前提下，A和F条件独立

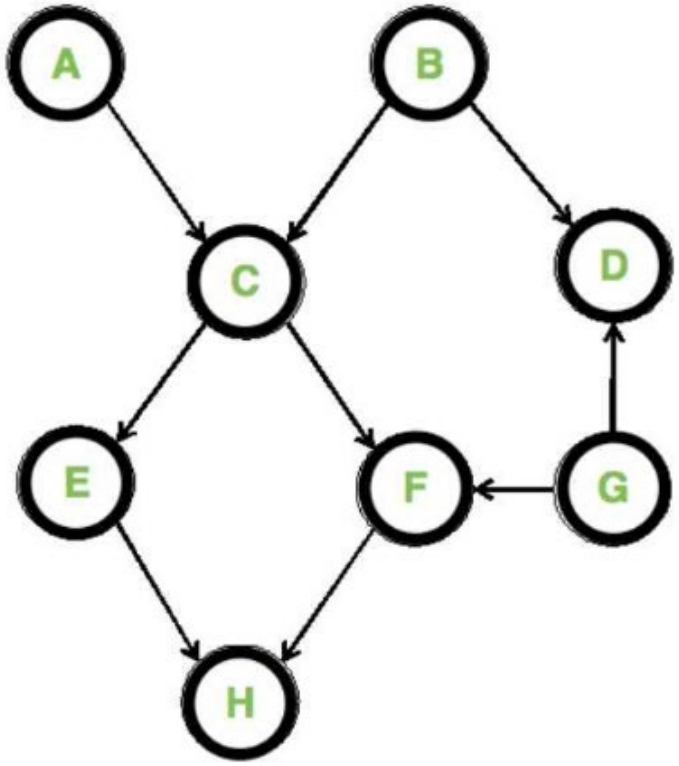
(10) 给定C的前提下，A和F条件独立

(11) 给定H的前提下，C和G不独立

- 给定贝叶斯网络，将概率分布$P(A,B,C,D)$和$P(C|A,B,D)$结构化分解

不确定知识表示和推理

- 第二次理论课作业：1.3 给定左下图所示的贝叶斯网络，回答问题。



(5) 给定C的前提下，E和F条件独立

(6) 给定C和D的前提下，E和F条件独立

(7) 给定C和H的前提下，A和F条件独立

(8) 给定C和D的前提下，A和F不独立

(9) 给定C和G的前提下，A和F条件独立

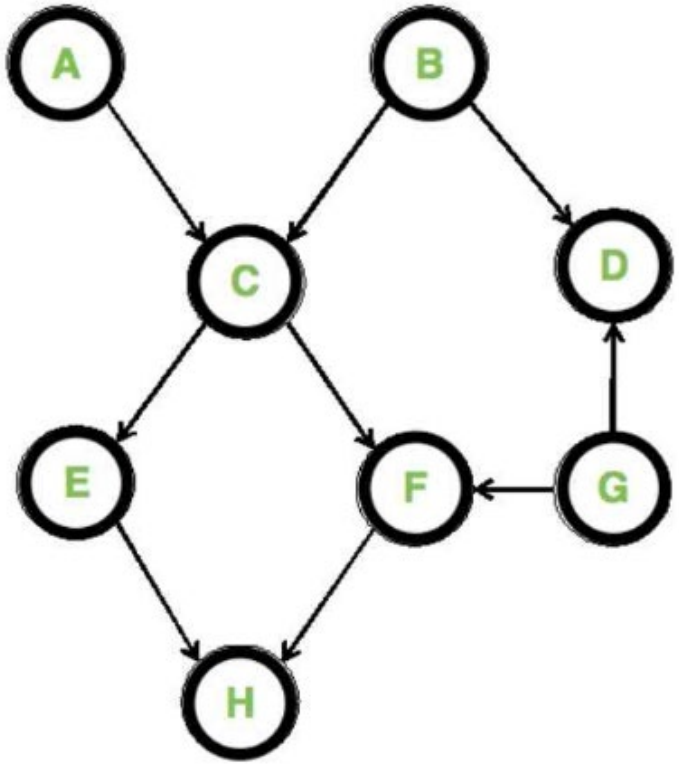
(10) 给定C的前提下，A和F条件独立

(11) 给定H的前提下，C和G不独立

- $P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)$ **简化前**

不确定知识表示和推理

- 第二次理论课作业：1.3 给定左下图所示的贝叶斯网络，回答问题。



(5) 给定C的前提下，E和F条件独立

(6) 给定C和D的前提下，E和F条件独立

(7) 给定C和H的前提下，A和F条件独立

(8) 给定C和D的前提下，A和F不独立

(9) 给定C和G的前提下，A和F条件独立

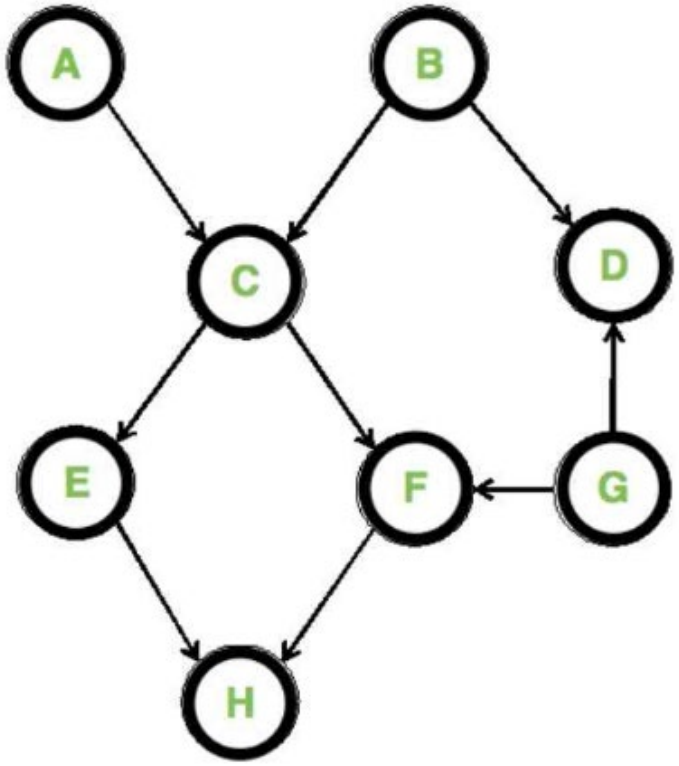
(10) 给定C的前提下，A和F条件独立

(11) 给定H的前提下，C和G不独立

- $P(A,B,C,D) = P(A)P(B)P(C|A,B)P(D|B)$ **简化后**

不确定知识表示和推理

- 第二次理论课作业：1.3 给定左下图所示的贝叶斯网络，回答问题。



(5) 给定C的前提下，E和F条件独立

(6) 给定C和D的前提下，E和F条件独立

(7) 给定C和H的前提下，A和F条件独立

(8) 给定C和D的前提下，A和F不独立

(9) 给定C和G的前提下，A和F条件独立

(10) 给定C的前提下，A和F条件独立

(11) 给定H的前提下，C和G不独立

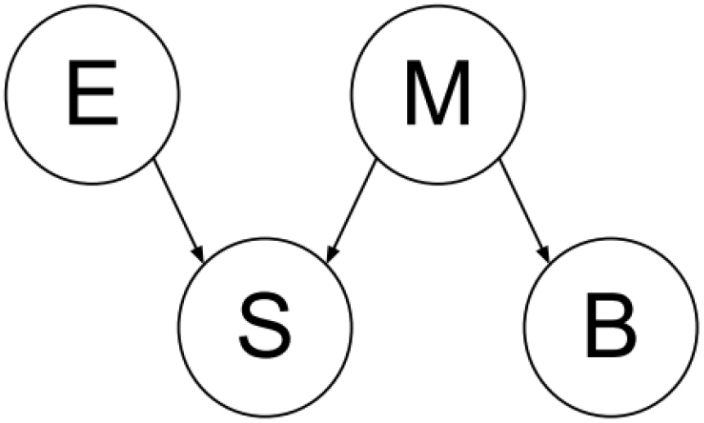
- $P(C \mid A,B,D) = P(C \mid A,B)$ **如果给定F，那么C和D还是独立的吗?**

不确定知识表示和推理

- 第二次理论课作业：1.5 计算联合/边缘/条件概率

$P(E=F,S=F,M=F,B=F)=P(E=F)P(M=F)P(S=F|E=F,M=F)P(B=F|M=F)$
$=0.6*0.9*0.9*0.9=0.4374$

$P(E=T|M=T)=P(E=T)=0.4$



E	P(E)
T	0.4
F	0.6

M	P(M)
T	0.1
F	0.9

M	B	P(B|M)
T	T	1.0
T	F	0.0
F	T	0.1
F	F	0.9

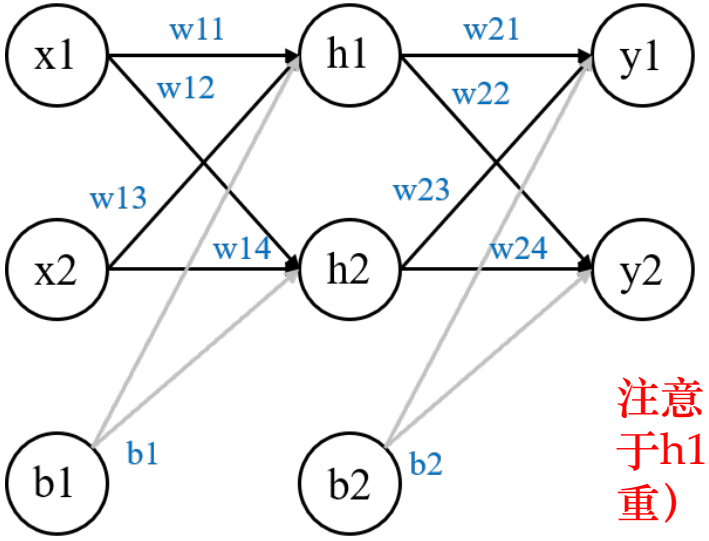
E	M	S	P(S|E,M)
T	T	T	1.0
T	T	F	0.0
T	F	T	0.8
T	F	F	0.2
F	T	T	0.3
F	T	F	0.7
F	F	T	0.1
F	F	F	0.9

机器学习

- 有监督学习
- 无监督学习
- 强化学习

有监督学习

- 如下图所示的多层感知机模型，第一层是输入层，包含两个神经元：x1=0.08，x2=0.12 和偏置b1；第二层是隐藏层，包含两个神经元：h1，h2 和偏置项b2；第三层是输出：y1，y2。每条线上标的$w_{i,j}$是第i层第j个权重参数，激活函数是sigmoid 函数（h神经元之后），Loss函数使用MSE（均方误差）函数，真实标签$Label_1 = 0.05, Label_2 = 0.95$，学习率$\alpha = 0.5$，求在经过一次反向传播后所有权重参数和偏置项参数的值（写出计算过程，最后结果保留四位小数）。



w11=0.24　w21=0.5
w12=0.3　w22=0.45
w13=0.2　w23=0.55
w14=0.16　w24=0.4
b1=0.28　b2=0.6

注意：本题中的偏置项b1同时作用于h1和h2（即h1和h2共享偏置项权重）；偏置项b2同时作用于y1和y2。

无监督学习

- 设有如下6个样本点A-F：
- A: (1, 1), B: (2, 1), C: (2, 2), D: (3, 5), E: (4, 4), F: (6, 8)
- 使用K-means算法对其进行聚类。选取A、B为初始簇中心，并且使用欧氏距离作为聚类中心和样本点之间的距离度量，写出第1次和第2次迭代后的簇中心。

第1次迭代：

	(1,1)	(2,1)
A(1,1)	0	1
B(2,1)	1	0
C(2,2)	$\sqrt{2}$	1
D(3,5)	$\sqrt{20}$	$\sqrt{17}$
E(4,4)	$\sqrt{18}$	$\sqrt{13}$
F(6,8)	$\sqrt{74}$	$\sqrt{65}$

第1个簇为{A}，簇中心为(1,1)　　第2个簇为{B,C,D,E,F}，簇中心为$(\frac{17}{5}, 4)$

无监督学习

- 设有如下6个样本点A-F：
- A: (1, 1), B: (2, 1), C: (2, 2), D: (3, 5), E: (4, 4), F: (6, 8)
- 使用K-means算法对其进行聚类。选取A、B为初始簇中心，并且使用欧氏距离作为聚类中心和样本点之间的距离度量，写出第1次和第2次迭代后的簇中心。

第2次迭代：

	(1,1)	$(\frac{17}{5}, 4)$
A(1,1)	0	$\frac{\sqrt{369}}{5}$
B(2,1)	1	$\frac{\sqrt{274}}{5}$
C(2,2)	$\sqrt{2}$	$\frac{\sqrt{149}}{5}$
D(3,5)	$\sqrt{20}$	$\frac{\sqrt{29}}{5}$
E(4,4)	$\sqrt{18}$	$\frac{3}{5}$
F(6,8)	$\sqrt{74}$	$\frac{\sqrt{569}}{5}$

第1个簇为{A,B,C}，簇中心为$(\frac{5}{3}, \frac{4}{3})$　　第2个簇为{D,E,F}，簇中心为$(\frac{13}{3}, \frac{17}{3})$

强化学习

- 强化学习（Reinforcement learning）是一种序贯决策方法，它研究如何让计算机与环境交互，从中学会最优决策。智能体在和环境的交互过程中，根据当前的状态、环境的奖惩而采取相应的动作，通过学习使环境的回报最大化。
- 强化学习的元素
  - 环境、状态S、动作A
  - 奖励（reward）R：智能体动作好坏的评价
  - 策略（policy）$\pi(a|s)$：状态到动作的映射$P(A = a \mid S = s)$
  - 值函数（value function）：动作价值函数$Q^{\pi}(s, a)$和状态价值函数$V^{\pi}(s)$，分别是在状态s采取动作a后，执行策略π的期望回报；以及从状态s起，执行策略π的期望回报

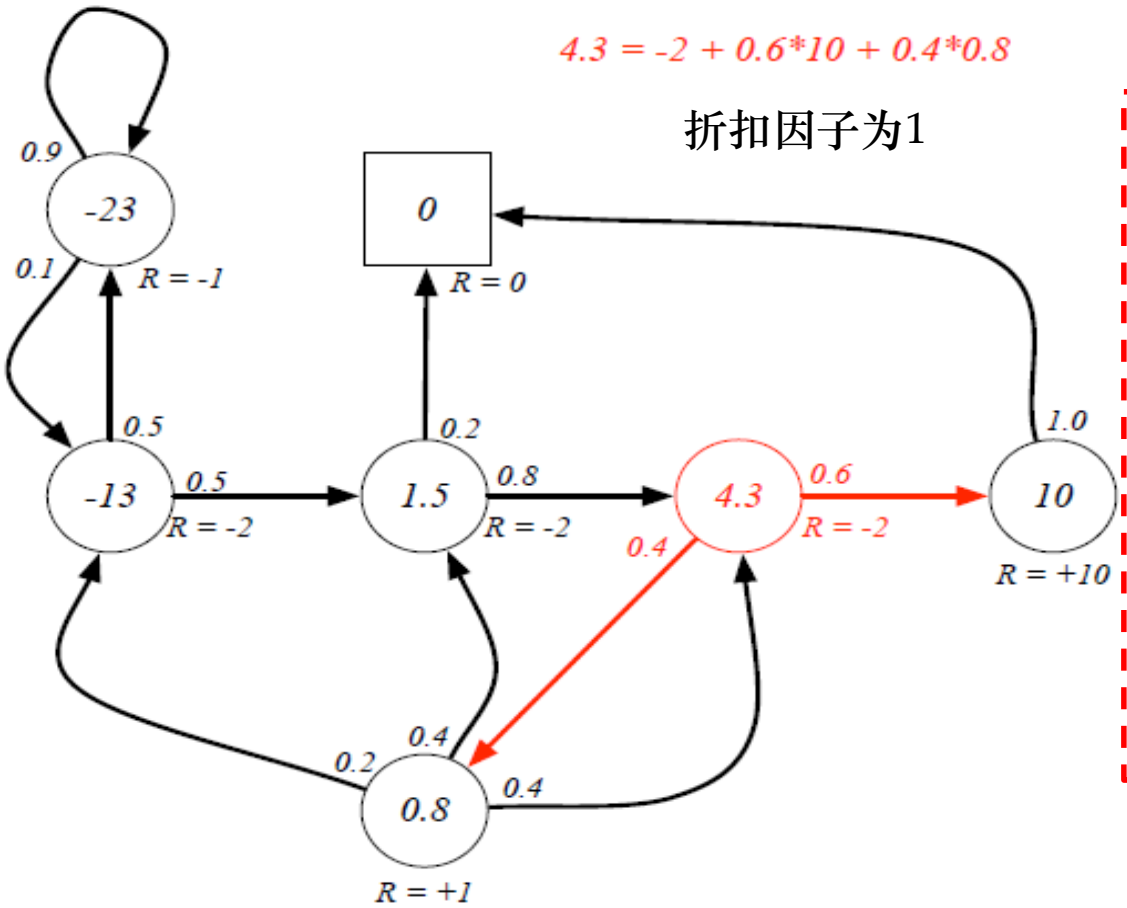
强化学习

$$U_t = R_t + \gamma \cdot U_{t+1}$$

$$V(s) = \underbrace{R(s)}_{\text{Immediate reward}} + \underbrace{\gamma \sum_{s' \in S} P(s'|s) V(s')}_{\text{Discounted sum of future rewards}}$$

4.3 = -2 + 0.6*10 + 0.4*0.8

折扣因子为1



-23 = -1 + 0.9*(-23) + 0.1*(-13)

-13 ≈ -2 + 0.5*1.5 + 0.5*(-23)

1.5 ≈ -2 + 0.8*4.3 + 0.2*0

10 = 10 + 1*0

0.8 ≈ 1 + 0.4*4.3 + 0.4*1.5 + 0.2*(-13)

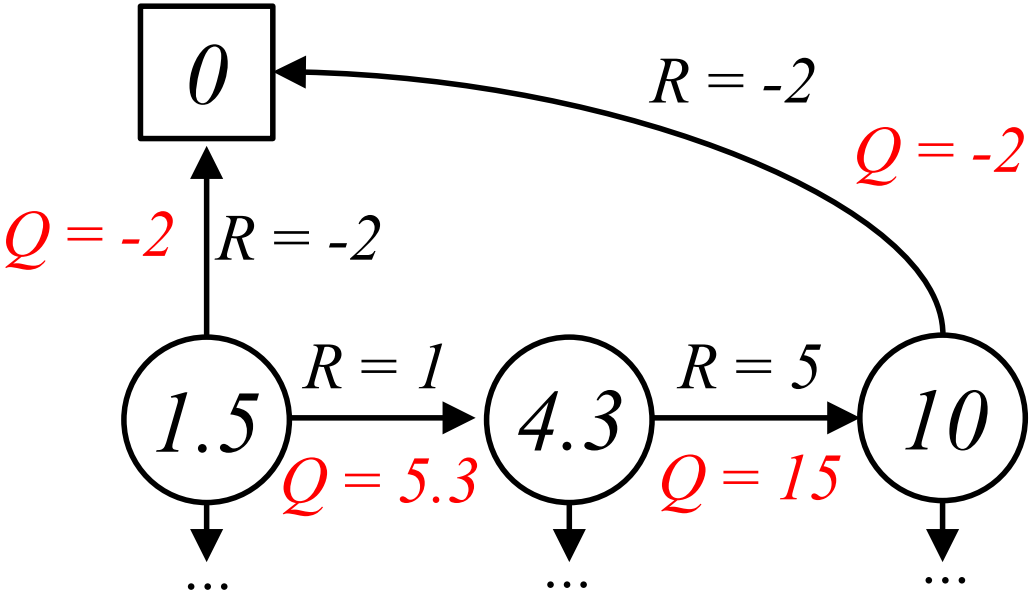
[from David Silver's course]

强化学习

- $Q_\pi(s_t, a_t) = \mathbb{E}[U_t | s_t, a_t]$

  $= \mathbb{E}[R_t + \gamma \cdot U_{t+1} | s_t, a_t]$

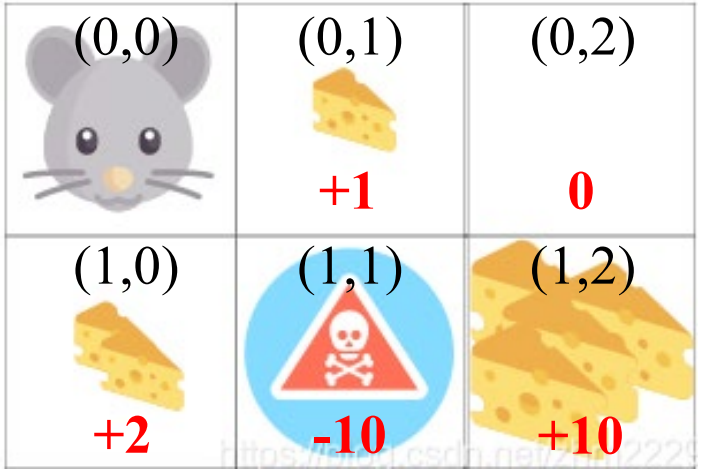
  $= \mathbb{E}[R_t | s_t, a_t] + \gamma \cdot \mathbb{E}[U_{t+1} | s_t, a_t]$



给定图中所示的MDP，其中节点（圆圈或正方形）表示状态，节点内的数值表示此状态迭代若干次后的V值。节点之间的边表示可行的动作，边上的R表示执行此动作带来的奖励值。对于每个状态而言，不同动作被选择的概率均相同。根据V值和Q值之间的关系，计算所有状态动作对的Q值（折扣因子为1）。

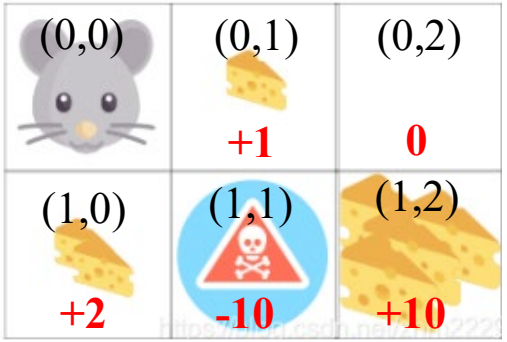
强化学习

- 左下图是一个迷宫游戏，小老鼠（Agent）最开始在(0,0)位置，分别在(0,1), (0,2), (1,0), (1,1), (1,2)处可获得+1, 0, +2, -10, +10的奖励值。当Agent位于(1,1), (1,2)时，游戏结束。Agent的动作有四个，分别是上下左右。
- 右下图是当前所有状态动作对的Q值表，其中行表示状态，列表示动作。



	up	down	left	right
(0,0)	N/A	3	N/A	2
(0,1)	N/A	0	1	1
(0,2)	N/A	4	2	N/A
(1,0)	2	N/A	N/A	2

强化学习



	up	down	left	right
(0,0)	N/A	3	N/A	2
(0,1)	N/A	0	1	2
(0,2)	N/A	4	2	N/A
(1,0)	2	N/A	N/A	2

- 假设Agent采取贪心策略(greedy)，状态转移概率为1，折扣因子γ为0.7。请根据动作价值函数定义，计算执行一次更新之后$Q\big((0,2),left\big)$的值。

动作价值函数定义$Q(s,a)=\sum_{s'}P_{ss'}^{a}[R_{ss'}^{a}+\gamma\sum_{a'}\pi(s',a')Q(s',a')]$

首先，因为状态转移概率为1，所以$P_{ss'}^{a}=P_{(0,2)\to(0,1)}^{left}=1$；

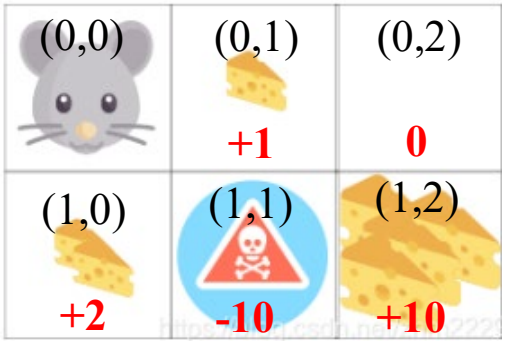
其次，因为采取贪心策略，所以$a'=\underset{a'}{\operatorname{argmax}}\,Q(s',a')=\underset{a'}{\operatorname{argmax}}\,Q((0,1),a')=right$

即，$\pi(s',a')=\pi((0,1),right)=1$，$Q(s',a')=Q((0,1),right)$

综上，在题给情况下的(最优)动作价值函数定义$Q(s,a)=R_{ss'}^{a}+\gamma\max_{a'}Q(s',a')$

因此，$Q((0,2),left)=1\cdot[1+0.7\cdot(1\cdot 2)]=\ 2.4$

强化学习



	up	down	left	right
(0,0)	N/A	3	N/A	2
(0,1)	N/A	0	1	2
(0,2)	N/A	4	2	N/A
(1,0)	2	N/A	N/A	2

- 假设Agent采取贪心策略(greedy)，学习率α为0.7，折扣因子γ为0.9，给定轨迹(0, 0) -> right -> (0, 1) -> right -> (0, 2) -> down -> (1, 2)，请使用时序差分(Temporal Difference)公式更新这条轨迹上的Q值。

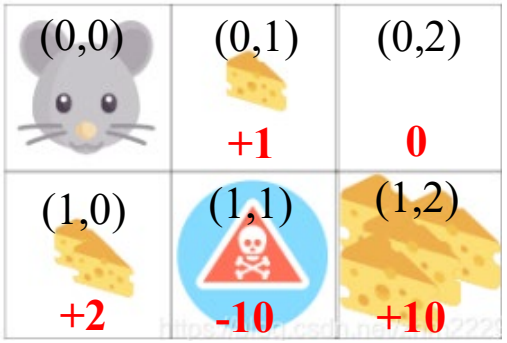
贪心策略下的时序差分公式$Q(s,a) = (1-\alpha)Q(s,a) + \alpha[R_{ss'}^{a} + \gamma \max_{a'} Q(s',a')]$

$$Q((0,0),right) = (1-0.7)\cdot Q((0,0),right) + 0.7\cdot\left[1 + 0.9\cdot \max_{a'} Q\big((0,1),a'\big)\right]$$
$$= 0.3\cdot 2 + 0.7\cdot[1 + 0.9\cdot 2] = 2.56$$

$$Q((0,1),right) = (1-0.7)\cdot Q((0,1),right) + 0.7\cdot\left[0 + 0.9\cdot \max_{a'} Q\big((0,2),a'\big)\right]$$
$$= 0.3\cdot 2 + 0.7\cdot[0 + 0.9\cdot 4] = 3.12$$

$$Q((0,2),down) = (1-0.7)\cdot Q((0,2),down) + 0.7\cdot[10 + 0.9\cdot 0]$$
$$= 0.3\cdot 4 + 0.7\cdot[10 + 0.9\cdot 0] = 8.2$$

强化学习



	up	down	left	right
(0,0)	N/A	3	N/A	2
(0,1)	N/A	0	1	2
(0,2)	N/A	4	2	N/A
(1,0)	2	N/A	N/A	2

- 假设学习率α为0.7，折扣因子γ为0.9，给定轨迹(0, 0) -> right -> (0, 1) -> right -> (0, 2) -> left -> (0, 1) -> down -> (1, 1)，请使用基于时序差分(Temporal Difference)的SARSA算法更新这条轨迹上的前三个Q值。

SARSA算法的Q值更新公式$Q(s,a) = (1-\alpha)Q(s,a) + \alpha[R_{ss'}^{a} + \gamma Q(s',a')]$

$$Q((0,0),right) = (1-0.7)\cdot Q((0,0),right) + 0.7\cdot\left[1 + 0.9\cdot Q\big((0,1),right\big)\right]$$
$$= 0.3\cdot 2 + 0.7\cdot[1 + 0.9\cdot 2] = 2.56$$

$$Q((0,1),right) = (1-0.7)\cdot Q((0,1),right) + 0.7\cdot\left[0 + 0.9\cdot Q\big((0,2),left\big)\right]$$
$$= 0.3\cdot 2 + 0.7\cdot[0 + 0.9\cdot 2] = 1.86$$

$$Q((0,2),left) = (1-0.7)\cdot Q((0,2),left) + 0.7\cdot\left[1 + 0.9\cdot Q\big((0,1),down\big)\right]$$
$$= 0.3\cdot 2 + 0.7\cdot[1 + 0.9\cdot 0] = 1.3$$