

6.034 测验 1

2001 秋季

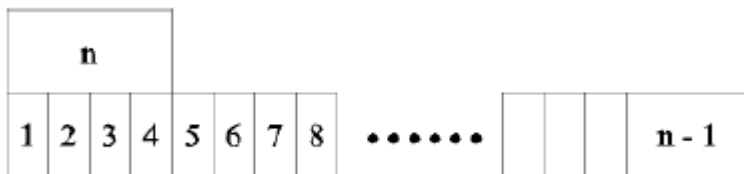
问题 1：地图中的资源分配（30 分）

A 部分（12 分）

问题中请注意以下几点：

- **保守校验**是指无论什么时候只要一个邻域的范围减少到某一数值就使用与一个变量有关的约束条件去限制该变量的范围。（注意：只要该域的范围减少到某一数值，校验就能一个邻域接着一个邻域地继续下去。）
- **激进校验**是指无论什么时候只要一个邻域的范围在减少就使用与一个变量有关的约束条件去限制该变量的范围。（注意：只要该域的范围在减少，校验就能一个邻域接着一个邻域地继续下去。）
- 无论什么时候只要校验产生了一个零值域，备份就开始。
- 没有动态排序。

一个朋友问你花多长时间给下面的地图着色：



如果你按如下方法进行你将发现只需一小会儿就可以完成着色：

- 采取我们一般图的着色限制：两个相邻的国家不能为同一种颜色。
- 从 1 到 n 按照给定的顺序给国家着色。
- 依次循环使用红、绿、蓝和黄色，以至最后图中每种颜色的出现次数都相差无几。
- 假设无论哪种校验之前，每种颜色都已经被涂过。
- 假设给一个区域着色需要 $1/100$ 秒的时间，但是换颜色不需要花费时间。
- 所有其余的计算过程不花费时间。
- 一年大约有 10^7 秒， 2^{10} 近似为 1000。
- 除了要已经着色过的地方保持一致，不使用约束校验。

哦，你对你朋友说需要花一年时间因为 n 等于 $15+4=19$ 15 从哪里来的，

$4^n = 100 \times 10^7$, 于是 $n \log 4 = \log 10^9$ 或者 $n \times 2 = \log (10^3)^3 = 3 \times \log (2^{10}) = 30$,
于是 $n = 15$

通过观察发现前四个节点并不是要搜索的指数树的一个部分, 所以有 4。

但是, 如果你使用保守校验那么必需的总时间近似为

0.2 秒

大约 20 种颜色被分配, 所需时间为 $20/100 = 0.2$ 秒

仅在给一个典型的变量比如 7 号国家分配一种颜色之后再使用保守校验, 其他还没有被分配颜色的变量正等待着进行约束校验, 这些变量的数目将:

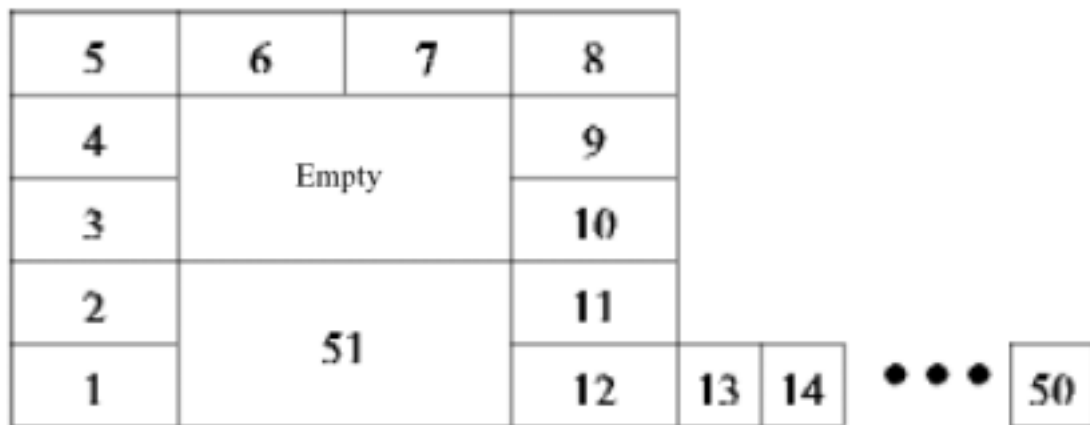
1, 因为右边邻域的范围减少到 3 种颜色, 所以将不会继续传播。

但如果仅在分配一种颜色给一个典型的变量比如 7 之后使用激进校验, 其他还没有被分配颜色的变量正等待着进行约束校验, 这些变量的数目将:

2, 因为右边邻域的范围在减少, 所以将会继续传播它的的邻域, 但是这个邻域可以是任何颜色, 所以将不会有更进一步的传播。

B 部分 (9 分)

现在你的朋友问你有关下面地图的着色问题:



你将使用 A 部分所要求的同样的策略, 除了与已经着色过的地方保持连贯性之外不需要有约束的校验。选出下面最佳的一项:

1. 除了要 and 已经着色过的地方保持一致, 不使用约束校验:
 - 最多花几秒钟。
 - 将会花一小时。
 - 将会花数年。(毕竟一棵深度为 15 树的都会花大约一年时间)

2. 接下来, 你决定使用保守校验:

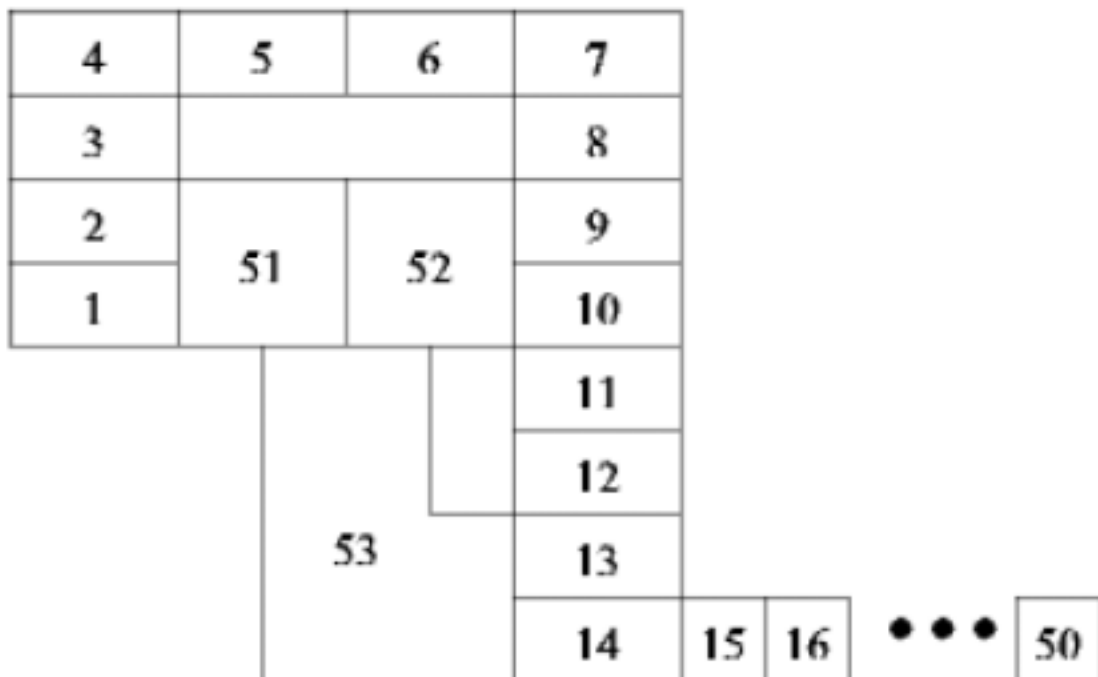
- 最多花几秒钟（实际上大约半秒钟，因为分配给 12 号国家的第一种颜色在着色之后立即被取消了。）
- 最多花几秒钟，但是比无约束校验要长一些。
- 将会花一小时。
- 将会花数年。

3. 最后，你决定使用激进校验：

- 最多花几秒钟时间。
- 最多花几秒钟时间，但是比保守校验长一些。（两种校验都可以；第一种是因为我们说只计着色时间而简化了 A 部分，第二是因为你合理地假设校验过程花费了时间。）
- 将会花一小时。
- 将会花数年。

C 部分（9 分）

现在，还有一张图：



选出下面最佳的一项：

1. 除了要和已经着色过的地方保持一致，不使用约束校验：

- 最多花几秒钟。
- 将会花一小时。
- 将会花数年。（参照 B 部分）

2. 接下来，你决定使用保守校验：

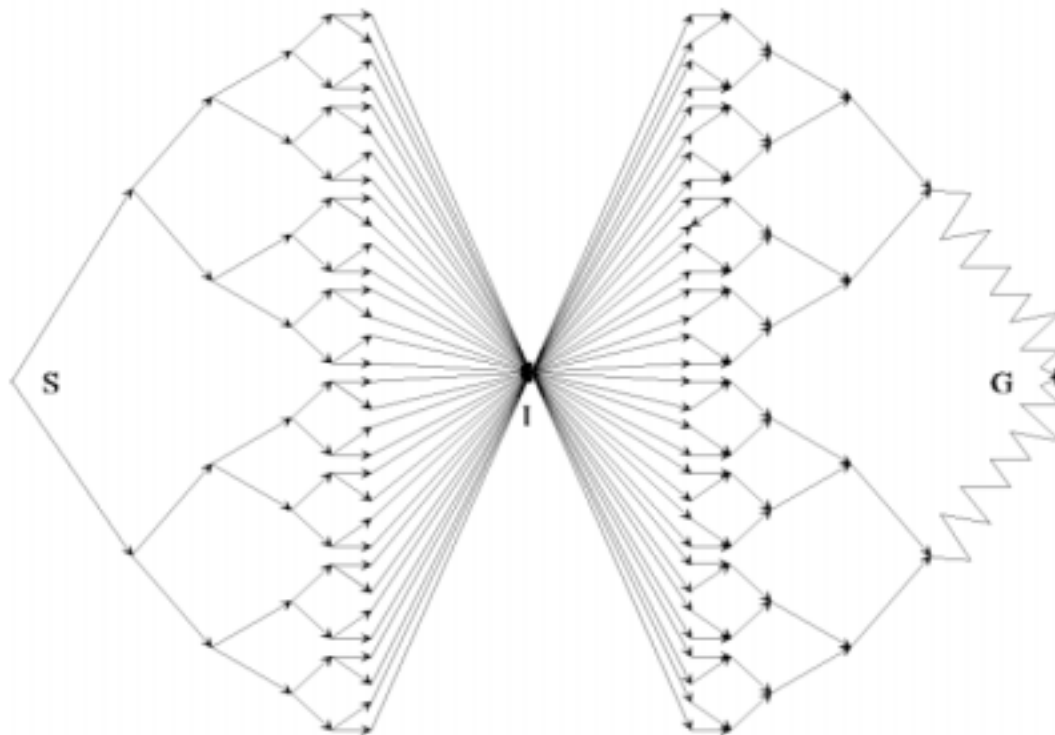
- 最多花几秒钟。
- 最多花几秒钟，但是比不校验要长一些。
- 将会花一小时。
- 将会花数年。（保守校验没有帮助；因为 51，52，53 号国家可获得的颜色不会减少到 1 种，所以当它们被校验的时候不会传播给它们的邻域。）

3. 最后，你决定使用激进校验：

- 最多花几秒钟。
- 最多花几秒钟，但是比保守校验要长一些。
- 将会花一小时。
- 将会花数年。（激进校验仍然没有帮助；虽然可以传播，但是成双的校验允许黄色和蓝色着色给 51，52 和 53 号国家，所以不可能更早停止，但在搜索的最后，三个国家只有两种颜色能分配；根据约束的编号来对国家进行排序就能解决这个问题。）

问题 2: 奇异城市中的搜索 (30 分)

你已经决定应聘驾驶出租车的工作。招聘人员是 MIT 的一名研究生, 他将要求你解决一些搜索街道的问题, 街道由图中的线段组成。在试卷的最后重印了该图, 它们可以撕下来。



地图依照比例绘制, 但是注意最后收敛于 G 的锯齿型街道长度可以认为是其他街道长度的 100 倍。所有街道都是由左至右单行的。当没有其他的条件时, 你只需遵从上述的条件即可。

你的工作是根据不同的条件, 选择从起始点 S 到终点 G 的一条路径。

答案可以使用数字(例如,8), 没有变量的表达式(例如, $2^0+2^1+2^2+2^3$) 或者两者混合。任意一个答案正确就可得满分。如果你注意到两个答案相等即使两者都是错误的, 也可以增加你的分值。

A 部分: 深度优先搜索, 带回溯 (7 分)

不使用已访问列表或者扩展列表。当搜索队列出现一个可到达目标的路径时搜索结束。指出扩展节点的个数。

12, 只是到达目标路径的节点数目, 不包括目标节点, 因为在目标节点被扩展之前搜索就已经停止了。

B 部分: 广度优先搜索 (7 分)

使用访问列表。当搜索队列出现一个可到达目标的路径时搜索结束。指出扩展节点的个数。

左边的所有节点加中间这个节点加右边的所有节点，不包括目标节点。

$$(2^6-1) + 1 + (2^6-1-1) = 126$$

C 部分：分支限界（8分）

找到最优路径，不使用扩展列表和启发式方法。指出扩展节点的个数。

左边扩展的节点数=63

有 32 条路径通往中间节点，加 32 个节点

有 32 条路径从中间节点延伸出去

从中间节点延伸出去的每条路径有 5 个节点

$$63 + 32 + (32 \times 5) = 5215$$

D 部分：A*（8分）

找到最优路径，使用扩展列表和作为启发的到目标的距离。指出扩展节点的个数。

因为锯齿状的街道太长，所以除了目标节点，每个节点将会被扩展，总数为 126，和 B 部分的一样。

问题 3：基于规则的助教跟踪（30 分）

温斯顿教授从来没有如此麻烦去找到他的助教们。今年，假定某个晚上，每个助教不是举办晚会就是去参加晚会。因为你在设计一个 UROP，所以请你帮助他设计一个基于规则的系统去预知助教们的行动。

A 部分：助教跟踪（15 分）

为了验证你对基于规则的系统理解无误，温斯顿立刻让你做一个驱动正向推理的测试。温斯顿散播出去的间谍网已经获取了一些助教们的信息。在试卷的最后重印了下面的表格，它们可以撕下来。

最初数据库：

Nevins 的 hip-hop 晚会
Nevins 的瑞格舞晚会
Crane 的狂欢晚会
Crane 的荧光棒晚会
Justin 寻找 跳舞
Jimmy 寻找 荧光棒

规则：

名字	前提	结论
R1	? 晚会上有跳舞	? 晚会上有碳酸饮料
R2	? 助教在寻找? x ? 晚会上有? x	? 助教热衷于? 晚会
R3	Justin 热衷于? 晚会 Jimmy 热衷于? 晚会 ? 晚会上有舞蹈 ? 晚会上有碳酸饮料	停止
R4	? 晚会上有狂欢	? 晚会上有跳舞
R5	? 晚会上有 hip-hop	? 晚会上有跳舞

你的工作是（在下表中）指出运行前向推理链的细节。使用规则分类作为你冲突解决的策略。

仅在添加了**停止**断言或者不再有断言时停止。只要一切都十分明确可以简化这些条款。

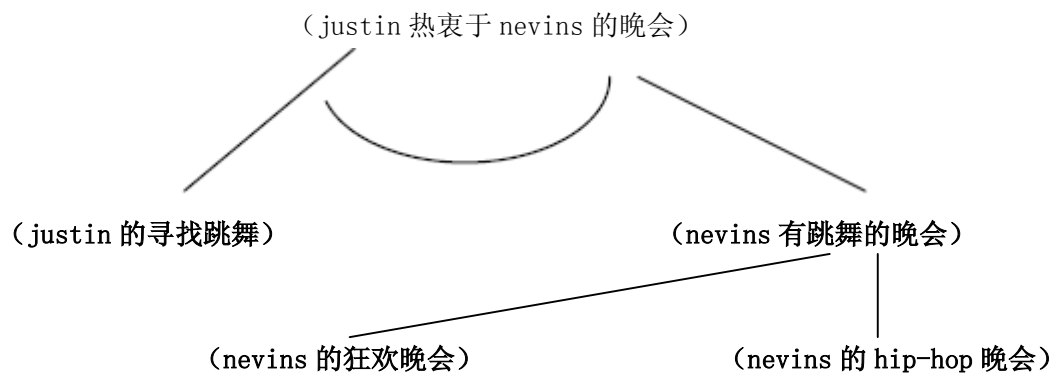
步骤	触发的规则	规则示例绑定	运用的规则	增加的断言
1	R2	? 助教: jimmy ? x: 荧光棒	R2	Jimmy 热衷于 crane 的晚会
	R4	? 晚会: crane 的晚会		
	R5	? 晚会: nevins 的晚会		
2	R4	? 晚会: crane 的晚会	R4	Crane 的晚会有跳舞
	R5	? 晚会: nevins 的晚会		
3	R1	? 晚会: crane 的晚会	R1	Crane 的晚会有碳酸饮料
	R2	? 助教: justin ? x: 舞蹈 ? 晚会: crane 的晚会		
	R5	? 晚会: nevins 的晚会		
4	R2	? 助教: justin ? x: 舞蹈 ? 晚会: crane 的晚会	R2	Justin 热衷于 crane 的晚会
	R4	? 晚会: nevins 的晚会		
5	R3	? 晚会: crane 的晚会	R3	停止
	R4	? 晚会: nevins 的晚会		

B 部分：反向推理链（15 分）

既然你已经得到了他的信任，温斯顿让你帮他编制反向推理的程序。很幸运的是附录 2 已经给你提供了代码，所以你所要做的只是找一个例子去测试你的助教跟踪程序的执行…

和 B 部分使用一样的规则，通过设计一个令人满意的目标（与/或）树，你将十分清楚地说明反向推理如何进行。根节点已经提供了，在下面空白地方画出目标树的其余部分，其所有

模式都要与数据库中的断言示例相匹配。



而且，在提供的表格中还指出了与数据库不相匹配的未示例的模式和这些模式的顺序。已经给出了第一步，表格中的空格足够你填写。

步骤	与数据库不相匹配的样式
1	(justin 热衷于 nevins 的晚会)
2	(justin 寻找 ?x)
3	(nevins 有跳舞的晚会)
4	(nevins 的狂欢晚会)
5	(nevins 的 hip-hop 晚会)
6	
7	
8	
9	

问题 4：其他（10 分）

在基于规则的正向推理系统中，一般而言（不仅仅是推导系统）：

- 所有触发的规则最终被应用（并非一般而言，仅仅是推导系统）
- 添加到数据库中的任何信息都不能被删除（只在推导系统中成立）
- 因为没有假设，所以没有办法回答为什么/如何的问题（课堂上动物园的例子则相反）
- 你的匹配函数必须能够让变量相互匹配（只适合后向连锁系统中）
- 以上所有的都是
- 以上任何一项都不是

逐步加深法，也称为反复加深法，在博弈中应用出色是因为：

- $\alpha - \beta$ 方法在给出的博弈树可以前进 2 倍深度。

(正确的, 但是和逐步加深法却无法处理)

- 最小最大法确保了分支因子在层与层之间几乎不变化 (并非最小最大的特性)
- 给定深度的博弈树中几乎没有任何一个节点是在最深的一层 (不对, 几乎所有节点都在最深的一层)
- 最后一层几乎一半的节点都趋于最大值 (并非一定正确, 但也没有什么关系)
- 以上所有的都是
- 以上任何一项都不是

$\alpha - \beta$:

- 使最小最大法搜索的深度几乎加倍
- 比最小最大法要慢 (但是比单独应用最小最大法要快)
- 和最小最大法是矛盾的 (胡说)
- 和逐步加深法是矛盾的 (不对)
- 以上所有的都是
- 以上任何一项都不是

关系数据库为 RETE 算法提供了基础, 因为:

- 选择运算允许约束传播 (胡说)
- 投影运算允许约束传播 (胡说)
- 并运算获得了规则条款间的约束 (在上课中强调过)
- 并运算给冲突解决提供了一种机制 (胡说)
- 以上所有的都是
- 以上任何一项都不是

不用搜索, 通过传播约束在连接类型处对线图进行分析:

- 不能保证产生唯一解 (在上课中强调过)
- 从不产生唯一解 (上课中举例)
- 在有多项式的时候总是产生唯一的解 (不一定产生唯一解)
- 如果一幅图包含阴影或裂纹就不能处理 (对具有阴影和裂纹的图也能很好地进行分析; 阴影通过提供更多的约束实际上还有帮助)
- 以上所有的都是
- 以上任何一项都不是