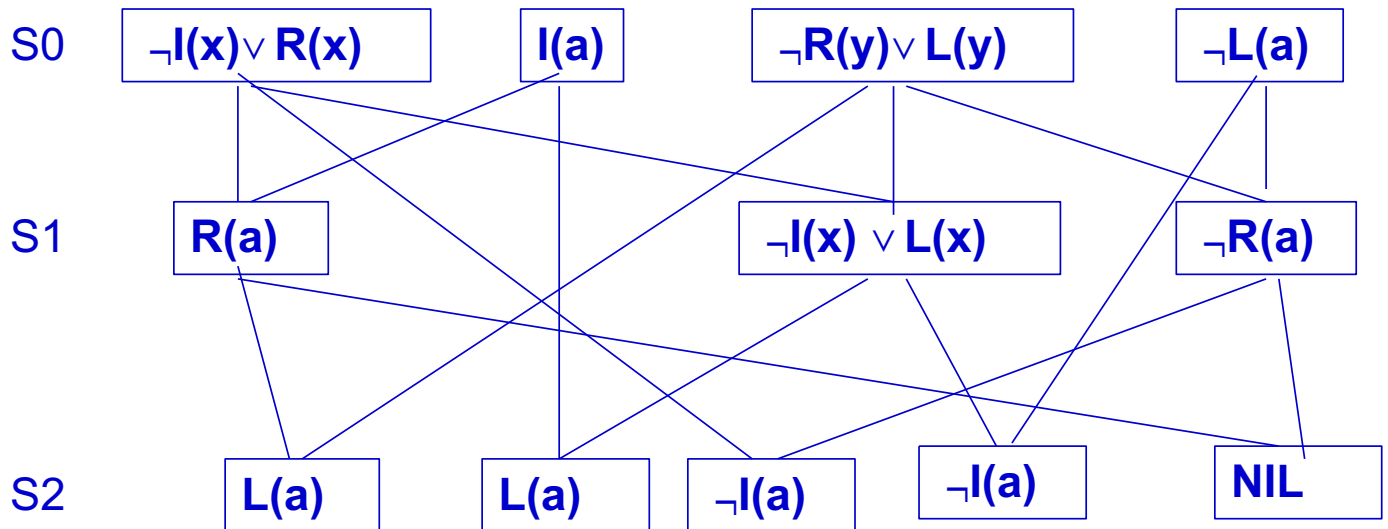


归结策略：广度优先

定义：状态、目标状态

例设有如下子句集： $S = \{ \neg I(x) \vee R(x), I(a), \neg R(y) \vee L(y), \neg L(a) \}$
用广度优先策略证明 S 为不可满足。

广度优先策略的归结树如下：



归结策略：广度优先

- 广度优先策略的优点：
 - 当问题有解时保证能找到最短归结路径。
 - 是一种完备的归结策略。
- 广度优先策略的缺点：
 - 归结出了许多无用的子句
 - 既浪费时间，又浪费空间
- 广度优先对大问题的归结容易产生组合爆炸，但对小问题却仍是一种比较好的归结策略。

归结策略

- 常用的归结策略可分为两大类：
 - **删除策略**是通过删除某些无用的子句来缩小归结范围
 - **限制策略**
是通过对参加归结的子句进行某些限制，来减少归结的盲目性，以尽快得到空子句。

归结策略

删除策略

归结策略：删除纯文字

- 删除法主要想法是：把子句集中无用的子句删除掉，这就会缩小搜索范围，减少比较次数，从而提高归结效率。

- 纯文字删除法

- 如果某文字 L 在子句集中不存在可与其互补的文字 $\neg L$ ，则称该文字为纯文字。
- 在归结过程中，纯文字不可能被消除，用包含纯文字的子句进行归结也不可能得到空子句
- 对子句集而言，删除包含纯文字的子句，是不影响其不可满足性的。例如，对子句集 $S = \{P \vee Q \vee R, \neg Q \vee R, Q, \neg R\}$ ，其中 P 是纯文字，因此可以将子句 $P \vee Q \vee R$ 从子句集 S 中删除。

归结策略：删除重言式

- 重言式删除法

- 如果一个子句中包含有互补的文字对，则称该子句为重言式。

例如 $P(x) \vee \neg P(x)$, $P(x) \vee Q(x) \vee \neg P(x)$

都是重言式，不管 $P(x)$ 的真值为真还是为假， $P(x) \vee \neg P(x)$ 和 $P(x) \vee Q(x) \vee \neg P(x)$ 都均为真。

- 重言式(**valid sentences**)是真值为真的子句。
对于一个子句集来说，不管是增加还是删除一个真值为真的子句，都不会影响该子句集的不可满足性。因此，可从子句集中删去重言式。

归结策略

限制策略

归结策略：支持集策略

- 支持集策略 (**Set of support**) :

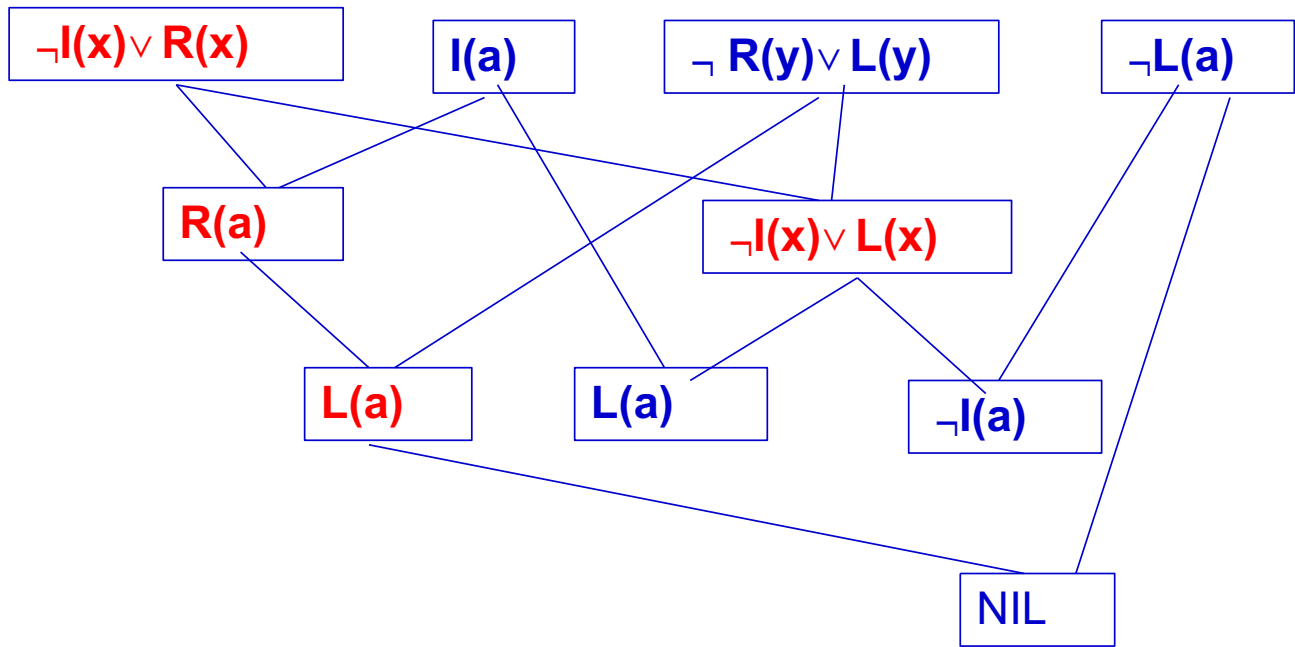
每一次参加归结的两个亲本子句中,

至少应该有一个是由目标公式的否定所得到的子句或它们的后裔。

- 支持集策略是完备的(?)
，即当子句集为不可满足时，则由支持集策略一定能够归结出一个空子句。
- 也可以把支持集策略看成是在广度优先策略中引入了某种限制条件
，这种限制条件代表一种启发信息，因而有较高的效率

归结策略：支持集策略

目标公式的否定



归结策略：支持集策略

- 支持集策略限制了子句集元素的剧增，但会增加空子句所在的深度（结果可能不是最优）。
- 支持集策略具有逆向推理的含义
，由于进行归结的亲本子句中至少有一个与目标子句有关，因此推理过程可以看作是沿目标、子目标的方向前进的。

归结策略:单文字子句策略

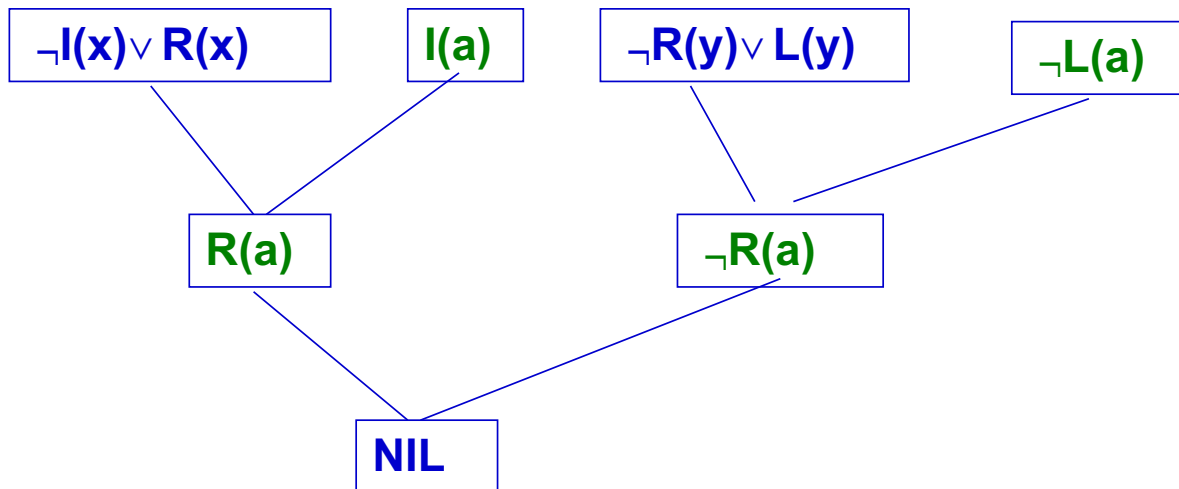
- 如果一个子句只包含一个文字，则称此子句为单文字子句。单文字子句策略是对支持集策略的进一步改进，它要求每次参加归结的两个亲本子句中至少有一个子句是单文字子句。
- 采用单文字子句策略，归结式包含的文字数将少于其非单文字亲本子句中的文字数，这将有利于向空子句的方向发展，因此会有较高的归结效率。

归结策略:单文字子句策略

例: 设有如下子句集:

$$S = \{ \neg I(x) \vee R(x), I(a), \neg R(y) \vee L(y), \neg L(a) \}$$

用单文字子句策略证明S为不可满足。

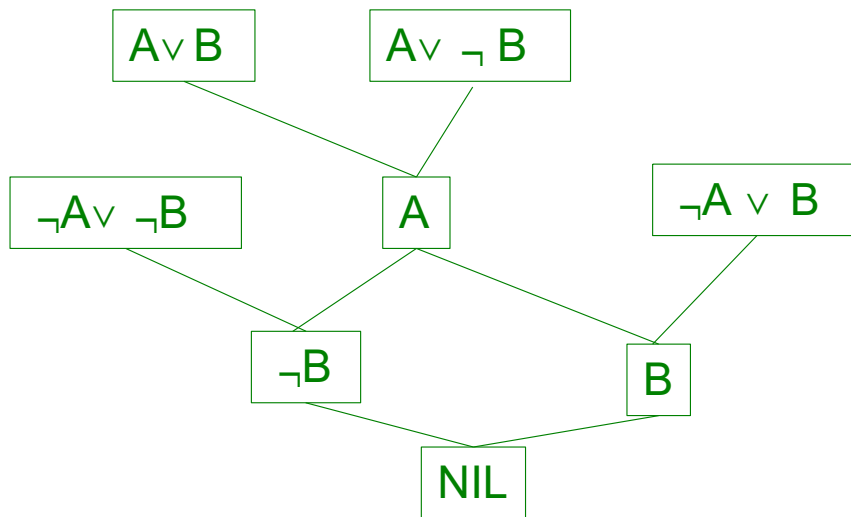


归结策略:单文字子句策略

- 单文字子句策略是**不完备**的, 即当子句集为不可满足时, 用这种策略**不一定能归结出空子句**。
- 原因: 没有可用的单文字字句

例如: 已知: $A \vee B$, $A \vee \neg B$, $\neg A \vee B$, 求证: $A \wedge B$

化为字句集后为: $A \vee B$, $A \vee \neg B$, $\neg A \vee B$, $\neg A \vee \neg B$, 不存在单文字的字句。但是可以消解出空。



归结策略:祖先过滤策略

- 祖先过滤策略(**Ancestry Filtering**):
满足以下两个条件中的任意一个就可进行归结:
 - 两个亲本子句中至少有一个是初始子句集中的子句。
 - 如果两个亲本子句都不是初始子句集中的子句, 则一个子句应该是另一个子句的先辈子句。
- 祖先过滤策略是完备的

归结策略:祖先过滤策略

例：设有如下子句集：

S={ $\neg Q(x) \vee \neg P(x)$, $Q(y) \vee \neg P(y)$, $\neg Q(w) \vee P(w)$, $Q(a) \vee P(a)$ }
用祖先过滤策略证明**S**为不可满足

