Five Groups of Rules for Magic Construction

Let $r$ be a typical rule

$$H : - G_1 \& G_2 \& \cdots \& G_n$$

Group I

Supplementary $\Rightarrow$ magic for next subgoal. If $G_i$ has IDB predicate $p$:

$$m_p(\text{bound args of } G_i) : - sup_{r,i-1}(\text{variables})$$

Group II

Magic $\Rightarrow$ 0th supplementary. If head has predicate $q$:

$$sup_{r,0}(\text{variables}) : - m_q(\text{bound args})$$

Group III

$i$-1st supplementary $\Rightarrow$ $i$th supplementary.

$$sup_{r,i}(\text{variables}) : - sup_{r,i-1}(\text{variables}) \& G_i$$

Group IV

Last supplementary $\Rightarrow$ last subgoal $\Rightarrow$ head.

$$H : - sup_{r,n-1}(\text{variables}) \& G_n$$

Group V

Initialize. If query has predicate $p$ we have the bodyless rule

$$m_p(\text{bound args from query})$$

- Group V is the only rule that depends on the actual query. Others depend only on the binding pattern.

Magic-Sets Beats Top-Down

- Claim: the “magic” rules implemented by seminaive evaluation only infers facts that any top-down implementation would infer.
  - I.e., magic-sets + seminaive has the advantages of both top-down and bottom-up.
  - Well not exactly. Prolog uses a “tail-recursion elimination” technique that sometimes does in $O(n)$ time what takes $O(n^2)$ by magic-sets.
  - The same trick has been used in deductive systems like LDL, Coral; it is described in Ch. 15 of PDKS-II.
Example

Binary trees constructed as terms.

- Leaves constructed by constant leaf.
- Interior nodes constructed by function symbol n. n(T₁, T₂) is a tree with left subtree T₁ and right subtree T₂.
- Predicate sub(T₁, T₂) true when T₁ is a subtree of T₂.
- Predicate eq(T₁, T₂) true when T₁ and T₂ are identical trees.

Rules:

\[ r₁ : \text{eq}(\text{leaf}, \text{leaf}) \]
\[ r₂ : \text{eq}(\text{n}(T₁, T₂), \text{n}(T₃, T₄)) : - \text{eq}(T₁, T₃) \& \text{eq}(T₂, T₄) \]
\[ r₃ : \text{sub}(T₁, T₂) : - \text{eq}(T₁, T₂) \]
\[ r₄ : \text{sub}(T₁, \text{n}(T₂, T₃)) : - \text{sub}(T₁, T₂) \]
\[ r₅ : \text{sub}(T₁, \text{n}(T₂, T₃)) : - \text{sub}(T₁, T₃) \]

Query: \text{sub}(T, \text{n}(\text{n}(\text{leaf}, \text{leaf}), \text{leaf})), \text{i.e.}, find the subtrees of a specific tree. Adorned goal: \text{sub}^{fb}.

The Rule/Goal Graph

- Interesting fact: \( r₄ \) and \( r₅ \) are not safe. however, for the \text{sub}^{fb} binding pattern, all variables of the head either appear in the
body or are present in a bound argument of the head.

- This notion of safety with respect to a binding pattern is appropriate when magic-sets is used.

The Magic Rules

Group I

\[
\begin{align*}
&m\text{eq}(T3) :- \text{sup}_{2,0}(T3,T4) \\
&m\text{eq}(T4) :- \text{sup}_{2,1}(T1,T3,T4) \\
&m\text{eq}(T2) :- \text{sup}_{3,0}(T2) \\
&m\text{sub}(T2) :- \text{sup}_{4,0}(T2,T3) \\
&m\text{sub}(T3) :- \text{sup}_{5,0}(T2,T3)
\end{align*}
\]

Group II

\[
\begin{align*}
&\text{sup}_{2,0}(T3,T4) :- m\text{eq}(n(T3,T4)) \\
&\text{sup}_{3,0}(T2) :- m\text{sub}(T2) \\
&\text{sup}_{4,0}(T2,T3) :- m\text{sub}(n(T2,T3)) \\
&\text{sup}_{5,0}(T2,T3) :- m\text{sub}(n(T2,T3))
\end{align*}
\]

Group III

\[
\begin{align*}
&\text{sup}_{2,1}(T1,T3,T4) :- \\
&\quad \text{sup}_{2,0}(T3,T4) \& eq(T1,T3)
\end{align*}
\]

Group IV

\[
\begin{align*}
&eq(leaf,leaf) \\
&eq(n(T1,T2),n(T3,T4)) :- \\
&\quad \text{sup}_{2,1}(T1,T3,T4) \& eq(T2,T4) \\
&\text{sub}(T1,T2) :- \text{sup}_{3,0}(T2) \& eq(T1,T2) \\
&\text{sub}(T1,n(T2,T3)) :- \\
&\quad \text{sup}_{4,0}(T2,T3) \& \text{sub}(T1,T2) \\
&\text{sub}(T1,n(T2,T3)) :- \\
&\quad \text{sup}_{5,0}(T2,T3) \& \text{sub}(T1,T3)
\end{align*}
\]

Group V

\[
\begin{align*}
&m\text{sub}(n(n(leaf,leaf),leaf))
\end{align*}
\]

Simplifying Magic Rules

- Use Group II rules to replace \(\text{sup}_{r,0}\)'s by magic predicates.

- If a supplementary predicate comes before an EDB subgoal, it is used only once and may be eliminated.

- However, if it is before an IDB subgoal, it is used twice, once in Group I and once in Group III or IV.
• Thus, we can omit \( sup_r, i \) if \((i + 1)\)st subgoal is EDB and \( i > 0 \).

• We use in place of Group III rules new rules of the form:

\[
\begin{align*}
    sup_r, i (\cdots) & :- \\
    sup_r, i-1 (\cdots) & \& G_i \& \cdots \& G_{j-1}
\end{align*}
\]

provided all of \( G_{i+1}, \ldots, G_{j-1} \) are EDB. Also, \( G_j \) is IDB, and either \( G_i \) is IDB or \( i = 1 \).

• Similarly, Group IV rules are replaced by

\[
H :- sup_r, i-1 (\cdots) \& G_i \& \cdots \& G_n
\]

provided all of \( G_{i+1}, \ldots, G_n \) are EDB. Also, \( G_i \) is IDB or \( i = 1 \).

• In the two rules above, use the appropriate magic predicate if \( i = 1 \).

**Example**

For the tree rules, we get:

**Group I**

\[
\begin{align*}
    m_{eq}(T3) & :- m_{eq}(n(T3,T4)) \\
    m_{eq}(T4) & :- sup_{2,1}(T1,T3,T4) \\
    m_{eq}(T2) & :- m_{sub}(T2) \\
    m_{sub}(T2) & :- m_{sub}(n(T2,T3)) \\
    m_{sub}(T3) & :- m_{sub}(n(T2,T3))
\end{align*}
\]

**Group III**

\[
\begin{align*}
    sup_{2,1}(T1,T3,T4) & :- \\
    m_{eq}(n(T3,T4)) & \& eq(T1,T3)
\end{align*}
\]

**Group IV**

\[
\begin{align*}
    eq(leaf,leaf)
    eq(n(T1,T2),n(T3,T4)) & :- \\
    sup_{2,1}(T1,T3,T4) & \& eq(T2,T4) \\
    sub(T1,T2) & :- m_{sub}(T2) \& eq(T1,T2) \\
    sub(T1,n(T2,T3)) & :- \\
    m_{sub}(n(T2,T3)) & \& sub(T1,T2) \\
    sub(T1,n(T2,T3)) & :- \\
    m_{sub}(n(T2,T3)) & \& sub(T1,T3)
\end{align*}
\]

**Group V**

\[
\begin{align*}
    m_{sub}(n(n(leaf,leaf),leaf))
\end{align*}
\]