Logic Programming

Query Optimization

Michael Genesereth
Computer Science Department
Stanford University
Two queries are semantically equivalent if and only if they produce identical results for every dataset.

Query 1:
\[
goal(X,Y) :- \ p(X) \ & \ r(X,Y) \ & \ q(X)
\]

Query 2:
\[
goal(X,Y) :- \ p(X) \ & \ \ q(X) \ & \ r(X,Y)
\]
Syntactically different but semantically equivalent queries may have different computational properties.

Query 1: $O(n^4)$
\[
\text{goal}(X,Y) :- p(X) \land r(X,Y) \land q(X)
\]

Query 2: $O(n^3)$
\[
\text{goal}(X,Y) :- p(X) \land q(X) \land r(X,Y)
\]
Types of Reformulation
- Logical - deleting and/or rearranging subgoals and rules
- Conceptual - changing vocabulary

Types of Logical Reformulation
- Rule Removal
- Subgoal Removal
- Subgoal Ordering

Types of Conceptual Reformulation
- Triples vs Wide Relations
- Minimal Spanning Trees
- Reification and Relationalization
Rule Removal
Useless Rules

Example:
\[
\text{goal}(X) :\neg p(X, Y) \land q(Y) \land \text{false}
\]

Example:
\[
\text{goal}(X) :\neg p(X, Y) \land q(Y) \land \lnot q(Y)
\]

*Useless rules produce no results.*
Redundant Rules

Example:

\[
\text{goal}(X) :- \ p(X,Y) \ & \ q(Y) \ & \ r(Y)
\]
\[
\text{goal}(X) :- \ p(X,Y) \ & \ q(Y)
\]

*Redundant rules produce only results produced by other rules, i.e. answers to one rule are a subset of answers to the other.*
Redundant Rules:

\[
\text{goal}(X) \leftarrow p(X,b) \land q(b) \land r(Z) \\
\text{goal}(X) \leftarrow p(X,Y) \land q(Y) \land r(Z)
\]

Non-Redundant Rules:

\[
\text{goal}(X) \leftarrow p(X,b) \land q(b) \land r(Z) \\
\text{goal}(X) \leftarrow p(X,Y) \land q(Y) \land r(c)
\]
A rule $r_1$ subsumes a rule $r_2$ if and only if it is possible to replace some or all of the variables of $r_1$ in such a way that the heads are the same and all of subgoals of $r_1$ are members of the body of $r_2$.

$$\text{goal}(X) :- \ p(X,Y) \ & \ q(Y)$$
$$\text{goal}(X) :- \ p(X,b) \ & \ q(b) \ & \ r(Z)$$

Here, the first rule subsumes the second. We just replace $X$ in the first rule by itself and replace $Y$ by $b$, with the following result.

$$\text{goal}(X) :- \ p(X,b) \ & \ q(b)$$
Start with rule 1 and rule 2 as inputs where (a) the heads are identical and (b) neither rule contains any negations.

(1) Create a substitution in which each variable in rule 2 is bound to a distinct, new symbol.

(2) Create a **canonical dataset** consisting of the subgoals of rule 2 where all variables are replaced by these bindings.

(3) Substitute the bindings for the head variables in rule 1.

(4) Evaluate this modified rule on the dataset created in (2). If there are answers, then rule 1 subsumes rule 2. If not, then rule 1 does not subsume rule 2.
Example

Inputs

\[ \text{goal}(X) := p(X,Y) \land q(Y) \]
\[ \text{goal}(X) := p(X,b) \land q(b) \land r(Z) \]

Substitution: \{x\leftarrow c_1, z\leftarrow c_3\}

Canonical Dataset: \{p(c_1,b), q(b), r(c_3)\}

Evaluate: \(p(c_1,Y) \land q(Y)\)
Result: \{Y\leftarrow b\}

The first rule does subsume the second.
Example

Inputs

\[
\text{goal}(X) :- p(X,b) & q(b) & r(Z) \\
\text{goal}(X) :- p(X,Y) & q(Y)
\]

Substitution: \{x\leftarrow c_1, y\leftarrow c_2\}

Canonical Dataset: \{p(c_1,c_2), q(c_2)\}

Evaluate: \ p(c_1,b) & q(b) & r(Z) \\
Result: failure

The first rule \textit{does not} subsume the second.
Rule Removal Technique

Compare every rule to every other rule (quadratic). If one rule subsumes another, it is okay to drop the subsumed rule.

NB: Applies only to rules with no negative subgoals and no predefined relations.

NB: The technique is sound in that it is guaranteed to produce an equivalent query.

NB: In the absence of any constraints on datasets to which the rules are applied, it is also guaranteed to be complete in that all surviving rules are needed for some dataset.

NB: In the face of constraints, it may be possible to drop rules that are not detected by this method, i.e. not complete.
If heads are not identical, they can sometimes be made identical by consistently replacing variables while avoiding clashes.

Original rules:

\[
\text{goal}(X) :- p(X,b) & q(b) & r(Z) \\
\text{goal}(U) :- p(U,V) & q(V)
\]

Equivalent rules:

\[
\text{goal}(X) :- p(X,b) & q(b) & r(Z) \\
\text{goal}(X) :- p(X,V) & q(V)
\]

There are other extensions for dealing with rules involving *negations* and *built-ins* and *constraints*. 
Subgoal Removal
Original Rule:

\[
\text{goal}(X,Y) :- p(X,Y) \& q(Y) \& q(Z)
\]

Equivalent Reformulation:

\[
\text{goal}(X,Y) :- p(X,Y) \& q(Y)
\]
Accept query rule as input.

(1) Create new query with rule head and empty body.

(2) Create a substitution with an entry for each head variable and a distinct constant.

(3) For each subgoal, (a) augment the substitution with bindings for the remaining variables in the other subgoals. (b) Create a dataset by replacing variables in those subgoals with their bindings. (c) Replace variables in chosen subgoal with bindings, leaving unbound variables as is. (d) Evaluate query. If fail, add to body of new query. Otherwise, drop.

Output the new query.
Query: \texttt{goal(X,Y) :- p(X,Y) \& q(Y) \& q(Z)}
Initial binding list - \{X\leftarrow c1, Y\leftarrow c2\}
Query: \texttt{goal(X,Y) :- p(X,Y) \& q(Y) \& q(Z)}

Initial binding list - \{X\leftarrow c1, Y\leftarrow c2\}

Subgoal: \texttt{p(X,Y)}. Binding list: \{X\leftarrow c1, Y\leftarrow c2, Z\leftarrow c3\}. Create dataset \{q(c2), q(c3)\}.

Evaluate \texttt{p(c1,c2)}. \textit{Fail}. So add \texttt{p(X,Y)} to new query.
Query: \( \text{goal}(X,Y) :- \ p(X,Y) \ & \ q(Y) \ & \ q(Z) \)

Initial binding list - \( \{X\leftarrow c1, Y\leftarrow c2\} \)

Subgoal: \( p(X,Y) \). Binding list: \( \{X\leftarrow c1, Y\leftarrow c2, Z\leftarrow c3\} \).
Create dataset \( \{q(c2), q(c3)\} \).
Evaluate \( p(c1,c2) \). \( \text{Fail} \). So add \( p(X,Y) \) to new query.

Subgoal: \( q(Y) \). Binding list: \( \{X\leftarrow c1, Y\leftarrow c2, Z\leftarrow c3\} \).
Create dataset \( \{p(c1,c2), q(c3)\} \).
Evaluate \( q(c2) \). \( \text{Fail} \). So add \( q(Y) \) to new query.
Query: \( \text{goal}(X,Y) :\leftarrow \ p(X,Y) \ & \ q(Y) \ & \ q(Z) \)
Initial binding list - \{X←c1, Y←c2\}

Subgoal: \( p(X,Y) \). Binding list: \{X←c1, Y←c2, Z←c3\}. Create dataset \{q(c2), q(c3)\}. Evaluate \( p(c1,c2) \). \text{Fail}. So add \( p(X,Y) \) to new query.

Subgoal: \( q(Y) \). Binding list: \{X←c1, Y←c2, Z←c3\}. Create dataset \{p(c1,c2), q(c3)\}. Evaluate \( q(c2) \). \text{Fail}. So add \( q(Y) \) to new query.

Subgoal: \( q(Z) \). Binding list: \{X←c1, Y←c2\}. Create dataset \{p(c1,c2), q(c2)\}. Evaluate \( q(Z) \). \text{Succeed}. So drop.
Query: \( \text{goal}(X,Y) :- p(X,Y) \& q(Y) \& q(Z) \)

Initial binding list - \( \{X \leftarrow c_1, Y \leftarrow c_2\} \)

Subgoal: \( p(X,Y) \). Binding list: \( \{X \leftarrow c_1, Y \leftarrow c_2, Z \leftarrow c_3\} \). Create dataset \( \{q(c_2), q(c_3)\} \).
Evaluate \( p(c_1,c_2) \). \text{Fail}. So add \( p(X,Y) \) to new query.

Subgoal: \( q(Y) \). Binding list: \( \{X \leftarrow c_1, Y \leftarrow c_2, Z \leftarrow c_3\} \). Create dataset \( \{p(c_1,c_2), q(c_3)\} \).
Evaluate \( q(c_2) \). \text{Fail}. So add \( q(Y) \) to new query.

Subgoal: \( q(z) \). Binding list: \( \{X \leftarrow c_1, Y \leftarrow c_2\} \). Create dataset \( \{p(c_1,c_2), q(c_2)\} \).
Evaluate \( q(z) \). \text{Succeed}. So drop \( q(Z) \).

Output: \( \text{goal}(X,Y) :- p(X,Y) \& q(Y) \)
Original Rule

\[
\text{goal}(X,Y) \ :- \ p(X,Y) \ & \ q(Y) \ & \ q(Z)
\]

Reformulation

\[
\text{goal}(X,Y) \ :- \ p(X,Y) \ & \ q(Y)
\]
Original Rule

\[ \text{goal}(X,Y) :- \text{p}(X,Y) \& \text{q}(X,Y) \& \text{p}(X,Z) \& \text{q}(X,Z) \]

Reformulation

\[ \text{goal}(X,Y) :- \text{p}(X,Y) \& \text{q}(X,Y) \]
Subgoal Ordering
Subgoal Ordering

Original Rule
\[
\text{goal}(X,Y) :- p(X) \land r(X,Y) \land q(X)
\]

Reformulation
\[
\text{goal}(X,Y) :- p(X) \land q(X) \land r(X,Y)
\]
Analysis

Original Rule

\[
\text{goal}(X,Y) :- \ p(X) \ & \ r(X,Y) \ & \ q(X)
\]

\[
(n^2 + 2n) + n*((n^2 + 2n) + n*(n^2 + 2n)) = n^4 + 3n^3 + 3n^2 + 2n
\]

Reformulation

\[
\text{goal}(X,Y) :- \ p(X) \ & \ q(X) \ & \ r(X,Y)
\]
Analysis

Original Rule
\[ \text{goal}(X, Y) \leftarrow \text{p}(X) \& \text{r}(X, Y) \& \text{q}(X) \]
\[
(n^2 + 2n) + n*((n^2 + 2n) + n*(n^2 + 2n)) = n^4 + 3n^3 + 3n^2 + 2n
\]

Reformulation
\[ \text{goal}(X, Y) \leftarrow \text{p}(X) \& \text{q}(X) \& \text{r}(X, Y) \]
\[
(n^2 + 2n) + n*((n^2 + 2n) + 1*(n^2 + 2n)) = 2n^3 + 5n^2 + 2n
\]
Accept query rule as input.

(1) Create new query with head of input and empty body.

(2) Iterate through subgoals. On encountering one with all variables bound in subgoals of new query, add to new query and remove from original query. If none found, remove first subgoal, add to new query, and repeat.

Output the new query.

Example:

\[
\text{goal}(X,Y) :\neg \ p(X) \ & \ r(X,Y) \ & \ q(X) \\
\text{goal}(X,Y) :\neg
\]
Accept query rule as input.

(1) Create new query with head of input and empty body.

(2) Iterate through subgoals. On encountering one with all variables bound in subgoals of new query, add to new query and remove from original query. If none found, remove first subgoal, add to new query, and repeat.

Output the new query.

Example:

\[
\begin{align*}
goal(X,Y) & \ :- \ p(X) \ & \ r(X,Y) \ & \ q(X) \\
goal(X,Y) & \ :- \ p(X)
\end{align*}
\]
Accept query rule as input.

(1) Create new query with head of input and empty body.

(2) Iterate through subgoals. On encountering one with all variables bound in subgoals of new query, add to new query and remove from original query. If none found, remove first subgoal, add to new query, and repeat.

Output the new query.

Example:

\[
\begin{align*}
\text{goal}(X,Y) & : \ p(X) \ & r(X,Y) \ & q(X) \\
\text{goal}(X,Y) & : \ p(X) \ & q(X)
\end{align*}
\]
Accept query rule as input.

(1) Create new query with head of input and empty body.

(2) Iterate through subgoals. On encountering one with all variables bound in subgoals of new query, add to new query and remove from original query. If none found, remove first subgoal, add to new query, and repeat.

Output the new query.

Example:

\[
\text{goal}(X,Y) :- \ p(X) \ & \ r(X,Y) \ & \ q(X) \\
\text{goal}(X,Y) :- \ p(X) \ & \ q(X) \ & \ r(X,Y)
\]
Example
Example

SEND
+MORE
-----
MONEY
goal(S,E,N,D,M,O,R,Y) :-
  digit(S) & digit(E) & digit(N) & digit(D) &
  digit(M) & digit(O) & digit(R) & digit(Y) &
  M!=0 & M!=S & M!=E & M!=N & M!=D &
  O!=S & O!=E & O!=N & O!=D & O!=M &
evaluate(S*1000+E*100+N*10+D,U) &
evaluate(M*1000+O*100+R*10+E,V) &
evaluate(M*10000+O*1000+N*100+E*10+Y,W) &
evaluate(plus(U,V),W)
Computational Analysis

Data

\begin{align*}
\text{digit}(1) & \quad \text{digit}(6) \\
\text{digit}(2) & \quad \text{digit}(7) \\
\text{digit}(3) & \quad \text{digit}(8) \\
\text{digit}(4) & \quad \text{digit}(9) \\
\text{digit}(5) & \quad \text{digit}(0)
\end{align*}

Rule

\[
\text{goal}(S, E, N, D, M, O, R, Y) :- \\
\quad \text{digit}(S) \land \text{digit}(E) \land \text{digit}(N) \land \text{digit}(D) \land \\
\quad \text{digit}(M) \land \text{digit}(O) \land \text{digit}(R) \land \text{digit}(Y) \land \ldots
\]

Analysis

\[
10 \times 10 \times 10 \times 10 \times 10 \times 10 \times 10 \times 10 = 10^8 = 100,000,000 \text{ cases}
\]

\[
111,111,110 \text{ unifications}
\]

Running time \(\sim\) minutes
Another Solution

goal(S,E,N,D,M,O,R,Y) :-
    digit(S) & S!=0 &
    digit(E) & E!=S &
    digit(N) & N!=S & N!=E &
    digit(D) & D!=S & D!=E & D!=N &
    digit(M) & M!=0 & M!=S & M!=E & M!=N & M!=D &
    digit(O) & O!=S & O!=E & O!=N & O!=D & O!=M &
    digit(R) & R!=S & R!=E & R!=N & R!=D &
        R!=M & R!=O &
    digit(Y) & Y!=S & Y!=E & Y!=N & Y!=D &
        Y!=M & Y!=O & Y!=R &
    evaluate(S*1000+E*100+N*10+D,U) &
    evaluate(M*1000+O*100+R*10+E,V) &
    evaluate(M*10000+O*1000+N*100+E*10+Y,W) &
    evaluate(plus(U,V),W)
goal(S,E,N,D,M,O,R,Y) :-
  digit(S) & mutex(S,0) &
  digit(E) & mutex(S,E) &
  digit(N) & mutex(S,E,N) &
  digit(D) & mutex(S,E,N,D) &
  digit(M) & distinct(M,0) & mutex(S,E,N,D,M) &
  digit(O) & mutex(S,E,N,D,M,O) &
  digit(R) & mutex(S,E,N,D,M,O,R) &
  digit(Y) & mutex(S,E,N,D,M,O,R,Y) &
  evaluate(S*1000+E*100+N*10+D,XX) &
  evaluate(M*1000+O*100+R*10+E,YY) &
  evaluate(M*10000+O*1000+N*100+E*10+Y),ZZ) &
  evaluate(XX+YY,ZZ)
Goal

\[ \text{goal}(S,E,N,D,M,O,R,Y) :\ - \]
\[ \text{digit}(S) & \text{mutex}(S,0) & \]
\[ \text{digit}(E) & \text{mutex}(S,E) & \]
\[ \text{digit}(N) & \text{mutex}(S,E,N) & \]
\[ \text{digit}(D) & \text{mutex}(S,E,N,D) & \]
\[ \text{digit}(M) & \text{distinct}(M,0) & \text{mutex}(S,E,N,D,M) & \]
\[ \text{digit}(O) & \text{mutex}(S,E,N,D,M,O) & \]
\[ \text{digit}(R) & \text{mutex}(S,E,N,D,M,O,R) & \]
\[ \text{digit}(Y) & \text{mutex}(S,E,N,D,M,O,R,Y) & \ldots \]

Analysis

\[ 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 = 1,814,400 \text{ cases} \]
\[ 5,989,558 \text{ unifications} \]
\[ \text{Running time } \sim 20 \text{ seconds} \]
Goal

\[ \text{goal}(S,E,N,D,M,O,R,Y) :- \]
\[ \text{digit}(S) \& \text{mutex}(S,0) \& \]
\[ \text{digit}(E) \& \text{mutex}(S,E) \& \]
\[ \text{digit}(N) \& \text{mutex}(S,E,N) \& \]
\[ \text{digit}(D) \& \text{mutex}(S,E,N,D) \& \]
\[ \text{digit}(M) \& \text{same}(M,1) \& \text{mutex}(S,E,N,D,M) \& \]
\[ \text{digit}(O) \& \text{mutex}(S,E,N,D,M,O) \& \]
\[ \text{digit}(R) \& \text{mutex}(S,E,N,D,M,O,R) \& \]
\[ \text{digit}(Y) \& \text{mutex}(S,E,N,D,M,O,R,Y) \& \ldots \]

Analysis

\[ 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 = 320,400 \text{ cases} \]

\[ 699,858 \text{ unifications} \]

Running time < 2 seconds
Computational Analysis

Data
digit(9)
digit(5)
digit(6)
digit(7)
digit(7)
digit(1)
digit(0)
digit(0)
digit(8)
digit(2)
digit(3)
digit(4)

Analysis

\[ 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36 \text{ cases} \]
Interpreted \sim 0 \text{ seconds}
Computational Analysis

Data

digit(9)
digit(5)
digit(6)
digit(7)
digit(1)
digit(0)
digit(8)
digit(2)
digit(3)
digit(4)

Analysis

1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = \textbf{36} unifications

Interpreted \sim 0 seconds
Narrow and Wide Relations
Represent wide relations as collections of binary relations.

Wide Relation:
\[
\text{student}(\text{Student, Department, Advisor, Year})
\]

Binary Relations:
\[
\text{student.major}(\text{Student, Department}) \\
\text{student.advisor}(\text{Student, Faculty}) \\
\text{student.year}(\text{Student, Year})
\]

Always works when there is a field of the wide relation (called the **key**) that uniquely specifies the values of the other elements. If none exists, possible to create one.
Wide Relation:

\[ p(a, d, e) \]
\[ p(b, d, e) \]
\[ p(c, d, e) \]

Triples:

\[ p1(a, d) \quad p2(a, e) \]
\[ p1(b, d) \quad p2(b, e) \]
\[ p1(c, d) \quad p2(c, e) \]
Wide Relation:
\[ p(a,d,e) \]
\[ p(b,d,e) \]
\[ p(c,d,e) \]

Query: \[ \text{goal}(X) :- p(X,d,e) \]
Cost without indexing: 3  
Cost with indexing: 3

Triples:
\[ p_1(a,d) \quad p_2(a,e) \]
\[ p_1(b,d) \quad p_2(b,e) \]
\[ p_1(c,d) \quad p_2(c,e) \]

Query: \[ \text{goal}(X) :- p_1(X,d) \land p_2(X,e) \]
Cost without indexing: 24  
Cost with indexing: 9
Minimal Spanning Trees
Social Isolation Cells
Vocabulary:
People - a, b, c, d, e, f, g, h, i, j, ...
Interaction - r/2

Example:
\( r(a,b) \)
\( r(a,e) \)
\( r(b,a) \)
\( r(b,c) \)
\( r(c,b) \)
\( r(d,e) \)
\( r(e,a) \)
\( r(e,d) \)

NB: Possible to represent undirected with only one factoid per arc rather than two, but we will ignore that for now.
Are two people are in the same cell?

goal(a,e) :- r(a,e)
goal(a,e) :- r(a,Y) & r(Y,e)
goal(a,e) :- r(a,Y1) & r(Y1,Y2) & r(Y2,e)
goal(a,e) :- r(a,Y1) & r(Y1,Y2) & r(Y2,Y3) & r(Y3,e)

Number of unifications for goal(a,e) (with indexing):

\[
\begin{align*}
8 \\
8 + 4*8 &= 40 \\
8 + 4*(8 + 4*8) &= 168 \\
8 + 4*(8 + 4*(8 + 4*8)) &= 680
\end{align*}
\]

Total: 896
Precompute and store the transitive closure of \( r \)

\[
\begin{array}{llllll}
  r(a,b) & r(b,a) & r(c,a) & r(d,a) & r(e,a) \\
  r(a,c) & r(b,c) & r(c,b) & r(d,b) & r(e,b) \\
  r(a,d) & r(b,d) & r(c,d) & r(d,c) & r(e,c) \\
  r(a,e) & r(b,e) & r(c,e) & r(d,e) & r(e,d) \\
\end{array}
\]

Are two people are in the same cell?

\[ \text{goal}(a,e) :- r(a,e) \]

Number of unifications for \( \text{goal}(a,e) \) (with indexing):

8

Number of factoids for \( n \) objects:

\( 8^n \)
Assign a number for each group and store with people

\[
\begin{align*}
  r(&a,1) & \quad r(&f,2) & \quad \ldots \\
  r(&b,1) & \quad r(&g,2) & \quad \ldots \\
  r(&c,1) & \quad r(&h,2) & \quad \ldots \\
  r(&d,1) & \quad r(&i,2) & \quad \ldots \\
  r(&e,1) & \quad r(&j,2) & \quad \ldots \\
\end{align*}
\]

Are two people are in the same cell?

\[
\text{goal}(a,e) :- r(a,N) \land r(e,N)
\]

Number of unifications for \text{goal}(a,e) (with indexing):

\[
2
\]

Number of factoids for \( n \) objects:

\[
n
\]
Lambda

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>goal(a,e)</code></td>
<td><code>p(a,Y1) &amp; p(Y1,Y2) &amp; p(Y2,Y3) &amp; p(Y3,e) &amp; false</code></td>
</tr>
</tbody>
</table>

680 unification(s)
Lambda

Save Revert Sort

\[ p(a,1) \]
\[ p(b,1) \]
\[ p(c,1) \]
\[ p(d,1) \]
\[ p(e,1) \]

Query

Pattern \( \text{goal}(a, e) \)

Query \( p(a, N) \land p(e, N) \)

Show Next 100 result(s) Autorefresh

2 unification(s)

goal(a, e)