Applying model based techniques for view deletion in databases

Chandrabose Aravindan, Peter Baumgartner
Universität Koblenz-Landau, Fachbereich Informatik
Rheinau 1, D-56075 Koblenz, Germany.
`{arvind,peter}@informatik.uni-koblenz.de`

Funded by DFG

Background: Building models using Hyper Tableuax

What is view deletion problem?

Given:

- Base facts (EDB)
- View definitions (IDB)
- A view atom to be deleted

Determine

new EDB that achieves the required view update

Rationality of View Updates

$$SD(X, Y) \leftarrow SSc(X, Z), ScD(Z, Y)$$
$$ScD(cs, uli) \leftarrow$$
$$SSc(peter, cs) \leftarrow$$
$$SSc(doro, cs) \leftarrow$$

Delete: $SD(peter, uli)$

General Guidelines: View definitions are immutable and deleted atom should not follow anymore

$\boxed{\text{Are these two the } \textit{only} \text{ guidelines ???}}$

What if I delete both $SSc(peter, cs)$ and $SSc(doro, cs)$?

Rationality Postulates

(KB—1) (Inclusion)	$KB—\alpha \subseteq KB$
(KB—2) (Immutable-inclusion)	$KB_I \subseteq KB—\alpha$
(KB—3) (Vacuity)	If $\alpha \notin$ Cn(KB), then $KB—\alpha = KB$
(KB—4) (Immutable-success)	If $KB_I \nvdash \alpha$, then $\alpha \notin \mathrm{Cn}(KB—\alpha)$
(KB—5) (Preservation)	If α and β are KB-equivalent, then $KB—\alpha = KB—\beta$
(KB—6.1) (Strong relevance)	If $\beta \in KB \setminus KB—\alpha$, then $\alpha \in \mathrm{Cn}(KB—\alpha \cup \{\beta\})$
(KB—6.2) (Relevance)	If $\beta \in KB \setminus KB—\alpha$, then $\exists KB'$ with $KB—\alpha \subseteq KB' \subseteq KB$ s.t. $\alpha \notin \mathrm{Cn}(KB')$ and $\alpha \in \mathrm{Cn}(KB' \cup \{\beta\})$
(KB—6.3) (Weak relevance)	If $\beta \in KB \setminus KB—\alpha$, then $\exists KB'$ with $KB' \subseteq KB$ s.t. $\alpha \notin \mathrm{Cn}(KB')$ and $\alpha \in \mathrm{Cn}(KB' \cup \{\beta\})$

Previous approaches to view deletion

Algorithms by Aravindan & Dung, Decker, Guessoum & Lloyd, Tomasic etc. are based on the following approach:

- Construct the set of all (locally minimal) explanations for the view atom to be deleted
- Determine a hitting set of this
- Remove this hitting set from EDB

- **Note:** Since all explanations are generated, this approach is of exponential space complexity

The transformation idea

$$
\begin{array}{llll}
IDB: & p & \leftarrow & t \\
 & p & \leftarrow & q \wedge u \\
 & q & \leftarrow & s \\
 & u & \leftarrow & r
\end{array}
\qquad\qquad
\begin{array}{lll}
EDB: & t & \leftarrow \\
 & r & \leftarrow
\end{array}
$$

The set S_0 is determined by all the IDB atoms and the current EDB atoms and in our case it is $\{p, q, u, t, r\}$.

$$
\begin{array}{rrcl}
IDB^*: & \neg t & \leftarrow & \neg p \\
 & \neg q \vee \neg u & \leftarrow & \neg p \\
 & & \leftarrow & s \wedge \neg q \\
 & \neg r & \leftarrow & \neg u
\end{array}
$$

The Transformation (Contd.)

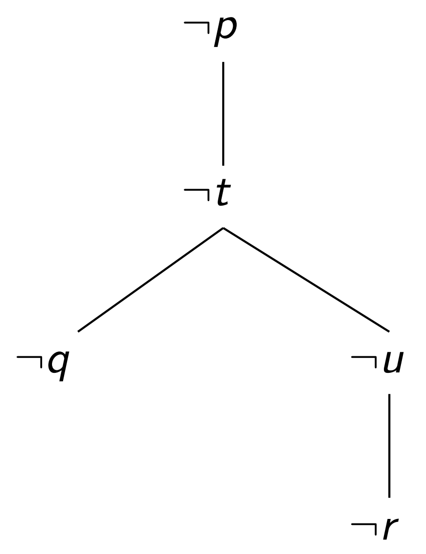
- The transformation results in a disjunctive logic program (in general)
- Negative literal such as $\neg A$ is treated as an atom that stands for deletion of A.
- There are no facts in the transformed program. So, when a delete request such as $\neg A$ is added to the program, any bottom-up strategy is fully focussed on the goal.
- **The Intuition:** To delete a view atom A, delete the EDB atoms appearing in a model of $IDB^{\star} \cup \{\neg A\}$.
- This set of EDB atoms is referred to as a *hitting set*.

An Example

$$\begin{array}{lrcl} IDB^{*}: & \neg t & \leftarrow & \neg p \\ & \neg q \vee \neg u & \leftarrow & \neg p \\ & & \leftarrow & s \wedge \neg q \\ & \neg r & \leftarrow & \neg u \end{array}$$

Update Request: Delete p

Hyper Tableau:



Hitting sets: $\{t\}$ $\{t, r\}$

Problem: The solution $\{t, r\}$ is not rational! r does not currently contribute to the derivation of p.

Strong Minimality Test

Idea: Use a rationality postulate as a test to filter out non-rational solutions!!!

(KB—6.1) (Strong relevance) If $\beta \in KB \backslash KB—\alpha$, then $\alpha \in \mathrm{Cn}(KB—\alpha \cup \{\beta\})$

Important Observation: This strong relevance test is equivalent to groundedness test used by Ilkka Niemelä for generating minimal models of a disjunctive logic program.

Consequence: Generating only the minimal models of $IDB^{\star} \cup \{\neg A\}$ ensures rational deletion of A. Also, the solution obtained is minimal wrt set inclusion.

Another important consequence: Since only one branch of hyper tableaux need to be kept in memory at a time, we achieve polynomial space complexity!!!

The Algorithm

Input:	A database $IDB \cup EDB$ and a ground view atom *A* to be deleted.
Output:	A new database $IDB \cup EDB'$

begin

1. Construct a branch *b* of an update tableau satisfying strong minimality for $IDB \cup EDB$ and delete request $\neg$*A*.
2. Produce $IDB \cup EDB \backslash HS(b)$ as a result.

end.

Main Results

- All minimal explanations are generated and every generated explanation contains a minimal explanation
- Since a branch cuts through all the explanations, it constitutes a hitting set, minimal in this case
- Hence the solutions generated by our algorithm are rational
- In particular, the strong relevance policy is satisfied and all solutions are minimal

Concluding Remarks

- We have presented a polynomial space algorithm for view deletion in databases
- This demonstrates successful application of model based techniques in real world problems
- This approach is very close to our diagnosis framework. Hence, our existing implementation for diagnosis can be adopted for view deletion as well.
- Our algorithm as presented assumes that the complete EDB is available. How to cope up with the situations where EDB is very huge or not completely available?
- Can this be extended for disjunctive databases?