WHAT IS AN EXPLANATION IN ANSPROLOG

Chitta Baral, and Jicheng Zhao

February 10, 2005
Explanation in Prolog, Datalog

Explanation of \{e\}: Proof tree

\[ e \leftarrow a \]
\[ a \leftarrow \text{not } b \]
\[ b \leftarrow c \]

There is no loop in the proof process
Explanation in Stratified Logic Program

A stratified logic program (A logic program without negative loops)

There are loops, we need level-by-level proof tree

\[
\begin{align*}
e & \leftarrow a \\
a & \leftarrow \text{not } b \\
b & \leftarrow c \\
c & \leftarrow b \\
\end{align*}
\]
What is the Explanation in Normal Logic Program (AnsProlog)?

- In the proof tree, we can always rank the atoms

\[
\begin{align*}
a & \leftarrow \text{not } b \\
b & \leftarrow \text{not } a \\
a & \leftarrow a \\
c & \leftarrow a \\
c & \leftarrow \text{not } c
\end{align*}
\]
• Stable model of the program: \( \{a, c\} \).

• We want the explanation of \( \{a, c\} \) to be

\[
\begin{align*}
a & \leftarrow \text{not } b \\
c & \leftarrow a
\end{align*}
\]
Definition: The Explanation of a Stable Model

A set of rules $Q$ is a explanation of a model $M$ in normal logic program $P$ if:

- There is a mapping $r : M \rightarrow \text{Integer}$ For each atom $a \in M$, there is a rule $r : a \leftarrow b_1, \ldots, b_m, \text{not } c_1, \ldots \text{ not } c_n$ in $Q$ such that $r$ is satisfied by $M$ and $r(b_i) < r(a) \ (1 \leq i \leq m)$

Proposition: $Q$ is an order-consistent program
Finding Explanations of a Stable Model

- A explanation of a model $M$ can be found in polynomial time:

- Algorithm:
− Input: 1. \( M = \{a, c\} \) 2.

\[
\begin{align*}
  a & \leftarrow \text{not } b \\
  b & \leftarrow \text{not } a \\
  a & \leftarrow a \\
  c & \leftarrow a \\
  c & \leftarrow \text{not } c
\end{align*}
\]

− Step 1: Get a set of rules that are satisfied by \( M \), and
with head in $M$.

\[
\begin{align*}
    a & \leftarrow \text{not } b \\
    a & \leftarrow a \\
    c & \leftarrow a
\end{align*}
\]

– Step 2: It is a stratified logic program. Finding the fixpoint according to the level-by-level proof tree and assign the
ranking

\[ a \leftarrow \text{not } b \]
\[ c \leftarrow a \]
More related Problems

- Finding the explanation of any set of atoms
- Finding a partial explanation of a set of atoms if the program is inconsistent
Explanation of a Set of Atoms

- Finding the explanation of $M = \{a\}$ in program

\[
\begin{align*}
    a & \leftarrow \text{not } b \\
    b & \leftarrow \text{not } a \\
    a & \leftarrow a \\
    c & \leftarrow a \\
    c & \leftarrow \text{not } c
\end{align*}
\]
• We want the explanation to be \( \{ a \leftarrow \text{not } b \} \).

• We can treat it equally as finding the explanation of \( \{ a, c \} \): Finding the explanation of an answer set \( N \), such that \( N \supseteq M \).

• We may not need rules about \( \{ c \} \)

• Approach:
  – Find the explanation of \( N \) and postprocessing by removing nodes in the bottom part of the proof tree [The complexity of finding an answer set]
  – directly
• Definition: Program $Q$ is an explanation of a set of atoms $M$ in a normal logic program $P$ if $Q \subseteq P$, $M$ is an answer set of $P$, each answer set of $Q$ is a subset of an answer set of $P$, and $Q$ is a minimal program satisfying these conditions.

• More difficult than finding an explanation to an answer set

• The explanation is an order-consistent program
Partial Explanation and Partial Stable Models

• Finding the explanation of $M = \{a\}$ in program

  
  \[
  \begin{align*}
  a & \leftarrow \text{not } b \\
  b & \leftarrow \text{not } a \\
  a & \leftarrow a \\
  c & \leftarrow \text{not } c
  \end{align*}
  \]

• The program has no stable models
• The inconsistency is not related to the set $M$.

• Partial explanation
Conclusions

- Explanation and partial explanation is useful in debugging a logic program.

- They are useful in answering queries in a larger logic program. A larger program may be inconsistent in one part. We can still answer queries that are not related to the inconsistencies.