Adding Disjunction to Datalog

First, how do we represent definite clauses in Python?

```
\begin{align*}
\text{c} & \rightarrow \text{a} \\
\text{a} & \rightarrow \text{b} \lor \text{c}
\end{align*}
```

- Knowledge engineer
- Natural increasing the power of Datalog; just making it easier for
- Means that if either \( \text{b} \) or \( \text{c} \) is true, then \( \text{a} \) must be true

Wanted to allow disjunction in body of clause

Overview

- Definitional Goals
- Depth Bounded
- Disjunction

Meta-Interpreters

- How to find proofs in Datalog
  - Depth First Search
  - Breadth First Search
  - Interactive Deepening
  - A* Search

We use Python for implementing reasoning procedure

- meta-interpreter: what your knowledge engineer programs in
- base language: what your knowledge engineer programs in
- must interface: write interface and implement in Python
- textbook 
- How to find proofs in Datalog
Example

When are the new neighbors?

\[ y \rightarrow a \land b \land m \]

\[ \{f \lor h\} \rightarrow a \]

\[ g \rightarrow a \]

\[ \{f \land e\} \rightarrow a \]

\[ \{f \lor e\} \rightarrow a \]

KB

Interpreter for Disjunction

- For each rule that has a disjunction, pick all of the disjuncts.
- Need to find all rules whose head matches a.
- To find all of the neighbors of a
  - Work on the KB first
  - This is not the don't care non-determinism of picking which conjunct to use
  - More like which disjunction we pick does matter
  - Disjunction replace a with one of the disjuncts
  - Contrast with entire conjunction
- Now, replacement might be a conjunction or disjunction
  - If the rule a \rightarrow b \land c
  - Then the rule to replace a with a list of conjunctions
  - Since a proof was a conjunction
  - Before (without disjunction)

Defining Clauses in Tcl

- Need a more complex representation to handle disjunction

For simplicity, let's not allow embedded bodies. Just:

\[ \{q \land a \land s\} \]

\[ \{q \lor a \land s\} \]

\[ \{q \land a \lor s\} \]

\[ \{q \lor a \lor s\} \]

\[ \{a\} \]

Examples:

- a is a literal. match if a appears anywhere in a body
- matches if a is bound or a is a literal
- needs a more complex representation to handle disjunction

Body:

- KB is a list of length 2, first element is head, second is body
Overview

- Disjunction
- Depth Bounded

Depth Bounded Reasoning Procedure

- Similar to Iterative Deepening
- But you don't keep going to deeper and deeper depths
- Could be done for either depth-first or breadth-first

Is it sound and complete?

WILL IT ALWAYS HALT?
• Rather than always choosing the first atom in the conjunction,
  • Have rules for when you can skip over atoms
  • At each step of the proof, keep rechecking whether atoms at the front of
    the answer clause (which were previously delayed) can be proved
  • Do not move delayed atoms to end of answer clause, as you should
    respect the defined ordering as much as possible

While we are at it...

• If there are any ground atoms, you might want to prove those first
  • As no variable bindings from earlier atoms will affect their truth
  • If they can't be proved, may as well find out sooner than later

Example

• Delaying might allow a definite clause to be used in a way it
  wasn't anticipated for
  • Example: Brother, where we expect both variables to be inputs
  • When used where either is not inputs, you get wrong behavior

Example: brother/2 where we expect both vars to be inputs

Delaying might allow a definite clause to be used in a way it

Delaying Goals

• Some goals, rather than being proved, can be delayed
  • To delay assumptions, so that you can collect assumptions that are
    needed for later
  • To collect assumptions, so that you can deduce assumptions that are
    needed for later
  • To defer conclusions with variables, in the hope that substituting calls will

Parameters are uninstantiated

When used where either is not inputs, you get wrong behavior

( (X, 'sold' Z)
( 'sold' X, Z)
( not (X = Y))
brother (X, Y)

when I unprocessed for

We will see more of this later in the course

To defer assumptions, so that you can deduce assumptions that are

To defer conclusions with variables, in the hope that substituting calls will

Some goals, rather than being proved, can be delayed
Recap of Class

- Adding Disjunction
  - Did not change expressiveness of datalog
- Depth Bounded Reasoning
  - Reasoning procedure that always halts in a certain amount of time
  - Sacrifices completeness
- Delaying Goals
  - Mechanism to remove influence from knowledge engineer to order
  - Did not change expressiveness of datalog

Caveat

- Re-ordering of atoms works in datalog
- Re-ordering of atoms does not make sense in Prolog in general