1 Directed Questions

- Given a knowledge base $KB$ and a conjunction of atoms $g$, what is meant by $KB \models g$?
  \textbf{Answer:} $g$ is a logical consequence of $KB$, that is, it is true in every model of $KB$.

- Given a proof procedure $P$, a knowledge base $KB$ and a conjunction of atoms $g$, what is meant by $KB \vdash_P g$?
  \textbf{Answer:} $g$ can be derived from $KB$ by means of the proof procedure $P$.

- Define what it means for a proof procedure to be sound.
  \textbf{Answer:} $KB \vdash g$ implies that $KB \models g$.

- Define what it means for a proof procedure to be complete.
  \textbf{Answer:} $KB \models g$ implies that $KB \vdash g$.

- What is the key idea of the bottom-up proof procedure?
  \textbf{Answer:} You search forward from the given knowledge base, iteratively expanding the set of atoms implied by the knowledge base $KB$.

- How do you know when you have completed a successful derivation using the bottom-up proof procedure?
  \textbf{Answer:} If the set of atoms implied by $KB$ is a superset of the atoms in $g$, the derivation is successful.

- How can the bottom-up proof procedure show that there is no successful derivation?
  \textbf{Answer:} If the procedure finishes and the set of atoms implied by $KB$ is not a superset of $g$, then there exists no successful derivation.

- What is the key idea of the top-down proof procedure?
  \textbf{Answer:} You search backward from the query to see if it can be derived from the knowledge base.

- How do you know when you have completed a successful derivation using the top-down proof procedure?
  \textbf{Answer:} You obtain the answer clause $\text{yes} \leftarrow$ with an empty body.

- Give an example of an admissible heuristic for top-down search.
  \textbf{Answer:} The number of atoms in the clause, since it will take at least that many resolution steps, i.e. it won’t overestimate.
2 Datalog

A university has asked you to write a program to help them determine whether or not to accept students who have applied for admission. There are 3 basic pathways for a student to be accepted. If a student is returning to the university after a time away and is in good academic standing with no outstanding fees, they are accepted. Students who submit a complete application and are qualified are also accepted. Students are qualified if they have high SAT scores as well as good high-school transcripts. The university also has a legacy program, wherein children of former graduates are qualified (though these student must still submit a complete application).

For brevity, let’s only talk about 3 individuals: Sam is a former graduate and Chris is his son. Chris has good high-school transcripts and he submitted a complete application. Laura is a returning student in good academic standing.

• Give the knowledge base representing this problem, using unary predicates accepted, returning, goodStanding, clearBalance, appComplete, qualified, legacyStudent, highSAT, goodHS, and graduate, as well as the binary predicate child. The university admissions officials should be able to provide queries such as accepted(chris) and get a true or false answer.

   **Answer:** Here is a sample KB.

```%
% file: school.pl

accepted(Student) <- returning(Student) & goodStanding(Student) & clearBalance(Student).
accepted(Student) <- appComplete(Student) & qualified(Student).

qualified(Student) <- legacyStudent(Student).
qualified(Student) <- highSAT(Student) & goodHS(Student).

legacyStudent(Student) <- child(Student, Parent) & graduate(Parent).

goodHS(chris).
graduate(sam).
child(chris, sam).
appComplete(chris).

returning(laura).
goodStanding(laura).
```

• Show the top-down derivation of the query accepted(chris) applied to your KB.

   **Answer:**

```%
yes ← [accepted(chris)].
yes ← [appComplete(chris), qualified(chris)].
yes ← [qualified(chris)].
yes ← [legacyStudent(chris)].
yes ← [child(chris, Parent), graduate(Parent)].
yes ← [graduate(sam)].
yes.

"yes" - proven
```
• Show one of the failing top-down derivations of the query \textit{accepted(laura)} applied to your KB.

\textbf{Answer:}

\begin{verbatim}
yes ← \{accepted(laura)\}.
yes ← \{returning(laura), goodStanding(laura), clearBalance(laura)\}.
yes ← \{goodStanding(laura), clearBalance(laura)\}.
yes ← \{clearBalance(laura)\}.
no choice, fail
\end{verbatim}

3 Learning Goals

You can:

• Define/read/write/trace/debug the BottomUp proof procedure
• Define/read/write/trace/debug the TopDown proof procedure
• Define/read/write/trace/debug the TopDown proof procedure as a search problem
• Represent simple domains in Datalog
• Apply TopDown proof procedure in Datalog