Logic: Bottom-up & Top-down proof procedures

Alan Mackworth

UBC CS 322 – Logic 3

March 4, 2013

P & M Textbook §5.2

Lecture Overview

Recap: Soundness, Completeness,
Bottom-up proof procedure

- Bottom-up Proof Procedure
  - Soundness proof
  - Completeness proof

- Top-down Proof Procedure

(Propositional) Logic: Review of Key ideas

- Given a domain that can be represented with n propositions, how many interpretations are there?
  - 2^n interpretations (similar to possible worlds)
- If you do not know anything about the domain you could be in any of those interpretations
- If you know that some logical formulae are true (your KB), you know that you can only be in interpretations in which those formulae hold (i.e. in …………… of KB)

(Propositional) Logic: Review of Key ideas

- Given a domain that can be represented with n propositions, how many interpretations are there?
  - 2^n interpretations (similar to possible worlds)
- If you do not know anything about the domain you could be in any of those interpretations
- If you know that some logical formulae are true (your KB), you know that you can only be in interpretations in which those formulae hold (i.e. in models of KB)
- It is useful to know what else is true in all those models

Definition (logical consequence)
If KB is a set of clauses and g is a conjunction of atoms,
g is a logical consequence of KB, written KB $\models$ g,
if g is true in every model of KB

- Example: KB = $\{h \leftarrow a, a, d \leftarrow c\}$. For which g is KB $\models$ g true?

(Propositional) Logic: Review of Key ideas

- Given a domain that can be represented with n propositions, how many interpretations are there?
  - 2^n interpretations (similar to possible worlds)
- If you do not know anything about the domain you could be in any of those interpretations
- If you know that some logical formulae are true (your KB), you know that you can only be in interpretations in which those formulae hold (i.e. in models of KB)
- It would be nice to know what else is true in all those models

Definition (logical consequence)
If KB is a set of clauses and g is a conjunction of atoms,
g is a logical consequence of KB, written KB $\models$ g,
if g is true in every model of KB

- Example: KB = {h $\leftarrow$ a, a, d $\leftarrow$ c}. Then KB $\models$ a and KB $\models$ h.

Intended interpretation

- User chooses task domain: intended interpretation.
  - This is the interpretation of the symbols the user has in mind
- User tells the system clauses (the knowledge base KB)
  - Each clause is true in the user's intended interpretation
  - Thus, the intended interpretation is a model
- The computer does not know the intended interpretation
  - But if it can derive something that's true in all models, then it is true in the intended interpretation
  - Once more, we want to derive logical consequences

Logical consequence

Definition (logical consequence)
If KB is a set of clauses and g is a conjunction of atoms,
g is a logical consequence of KB, written KB ⊧ g,
if g is true in every model of KB

- If KB ⊧ g, then …   (multiple answers correct)

g is true in the intended interpretation

There is at least one model of KB in which g is true

g is true in every model of KB

g is true in some models of KB,
but not necessarily the intended interpretation

Logical consequence

> **Definition (logical consequence)**
> If KB is a set of clauses and g is a conjunction of atoms,
> g is a logical consequence of KB, written KB $\models$ g,
> if g is true in every model of KB

- If KB $\models$ g, then …
  - g is true in every model of KB (by definition)
  - The intended interpretation is one of these models, so g is also true in it
- Notice there is always at least one model of any PDCL theory! Q: What is it?

Logical consequence

Definition (logical consequence)
If KB is a set of clauses and g is a conjunction of atoms,
g is a logical consequence of KB, written KB $\models$ g,
if g is true in every model of KB

- If KB $\models$ g, then …
  - g is true in every model of KB (by definition)
  - The intended interpretation is one of these models, so g is also true in it
- Notice there is always at least one model of any PDCL theory! Q: What is it?
- A: The interpretation with every atom true is always a model. Why? Because each clause is true in that int'n.

Recap: proofs, soundness, completeness

- A proof is a mechanically derivable demonstration that a formula logically follows from a knowledge base.

Definition (derivability with a proof procedure)
Given a proof procedure P, KB $\vdash_P$ g means g can be derived from knowledge base KB with proof procedure P.

- We want our proof procedures to be sound and complete

Definition (soundness)
A proof procedure P is sound if KB $\vdash_P$ g implies KB $\models$ g.

sound: every atom g that P derives follows logically from KB

Definition (completeness)
A proof procedure P is complete if KB $\models$ g implies KB $\vdash_P$ g.

complete: every atom g that logically follows from KB is derived by P

Example: an unsound proof procedure

- Unsound proof procedure U:
  - U derives every atom in KB: for any g that appears in KB, KB $\vdash_U$ g
- Proof procedure U is unsound:
  - There are atoms it derives that do not logically follow from KB
  - E.g. KB = {a $\leftarrow$ b}.

    It will derive a and b, but neither of them logically follows from KB
  - Thus KB $\vdash_U$ g does not imply KB $\models$ g $\rightarrow$ unsound
- Proof procedure U is complete:
  - It will not miss any atoms since it derives every atom g
  - Thus KB $\models$ g implies KB $\vdash_U$ g $\rightarrow$ complete

Example: an incomplete proof procedure

- Incomplete proof procedure I:
  - I derives nothing: there is no atom g such that KB $\vdash_I$ g

- Proof procedure I is sound:
  - It does not derive any atom at all,
    so every atom it derives follows from KB
  - Thus KB $\vdash_I$ g implies KB $\models$ g $\rightarrow$ sound

- Proof procedure I is incomplete:
  - It will miss atoms that logically follow from KB
  - E.g. KB = {a}: KB $\models$ a, but not KB $\vdash_I$ a
  - Thus KB $\models$ g does not imply KB $\vdash_I$ g $\rightarrow$ incomplete

Recap: Bottom-up proof procedure

KB $\vdash_{BU}$ g if and only if g $\in$ C at the end of the following procedure.

C := {};

repeat

select clause $h \leftarrow b_1 \wedge \ldots \wedge b_m$ in KB such that $b_i \in C$ for all i, and $h \notin C$;

C := C $\cup$ {h}

until no more clauses can be selected.

Bottom-up proof procedure: example

C := {};
repeat
    select clause $h \leftarrow b_1 \wedge \ldots \wedge b_m$ in KB
        such that $b_i \in C$ for all i, and $h \notin C$;
    $C := C \cup \{h\}$
until no more clauses can be selected. KB $\vdash_{BU}$ g if and only if $g \in C$

$a \leftarrow b \wedge c$
$a \leftarrow e \wedge f$
$b \leftarrow f \wedge k$
$c \leftarrow e$
$d \leftarrow k$
e.
$f \leftarrow j \wedge e$
$f \leftarrow c$
$j \leftarrow c$

{}
{e}
{c,e}
{c,e,f}
{c,e,f,j}
{a,c,e,f,j}

Done.

Lecture Overview

- Recap: Soundness, Completeness,
  Bottom-up proof procedure

- Bottom-up Proof Procedure
  - Soundness proof
  - Completeness proof

- Top-down Proof Procedure

Soundness of bottom-up proof procedure BU

Definition (soundness)

A proof procedure P is sound if KB $\vdash_P$ g implies KB $\models$ g.

sound: every atom g that P derives follows logically from KB

C := {};
repeat
select clause $h \leftarrow b_1 \wedge \ldots \wedge b_m$ in KB
such that $b_i \in C$ for all i, and $h \notin C$;
C := C $\cup$ {h}
until no more clauses can be selected. KB $\vdash_{BU}$ g if and only if $g \in C$

What do we need to prove to show that BU is sound ?

Soundness of bottom-up proof procedure BU

Definition (soundness)

A proof procedure P is sound if KB $\vdash_P$ g implies KB $\models$ g.

sound: every atom g that P derives follows logically from KB

C := {};
repeat
    select clause $h \leftarrow b_1 \wedge \ldots \wedge b_m$ in KB
        such that $b_i \in C$ for all i, and $h \notin C$;
    C := C $\cup$ {h}
until no more clauses can be selected. KB $\vdash_{BU}$ g if and only if $g \in C$

What do we need to prove to show that BU is sound ?

If $g \in C$ at the end of BU procedure,
then g is true in all models of KB (KB $\models$ g)

Soundness of bottom-up proof procedure BU

```
C := {};
repeat
        select clause h ← b1 ∧ … ∧ bm in KB
              such that bi ∈ C for all i, and h ∉ C;
        C := C ∪ {h}
until no more clauses can be selected.
```

For soundness of bottom-up proof procedure BU: prove

If $g \in C$ at the end of BU procedure,
then g is true in all models of KB (KB $\models$ g)

By contradiction: Suppose there is a g such that KB $\vdash$ g but not KB $\models$ g.

- Let h be first atom added to C that's not true in every model of KB
  - In particular, suppose I is a model of KB in which h isn't true.
- There must be a clause in KB of form $h \leftarrow b_1 \wedge \ldots \wedge b_m$
- Each b_i is true in I. h is false in I. So this clause is false in I.
- Thus, I is not a model of KB. Contradiction: thus no such g exists

Soundness of bottom-up proof procedure BU

- The previous proof of soundness of BU proof procedure was by contradiction.
- Now we will give a direct proof by induction….

Soundness of bottom-up proof procedure BU

C := {};
repeat
    select clause $h \leftarrow b_1 \wedge \ldots \wedge b_m$ in KB
        such that $b_i \in C$ for all i, and $h \notin C$;
    $C := C \cup \{h\}$
until no more clauses can be selected. KB $\vdash_{BU}$ g if and only if $g \in C$

Inductive proof using inductive hypothesis IH:

IH: if $g \in C$ at loop iteration n, **then** g is true in all models of KB (KB $\models$ g)

Base case: "IH holds for n=0". C = {}, so IH holds trivially

Inductive case: "if IH holds for n, it holds for n+1".

- Here: "if IH held before a loop iteration, it holds afterwards"
- The only new element in C is h, so we only need to prove KB $\models$ h
- $b_1, \ldots, b_m$ were in C before, so by IH we know KB $\models b_1 \wedge \ldots \wedge b_m$
- In every model, "$b_1 \wedge \ldots \wedge b_m$" is true and "$h \leftarrow b_1 \wedge \ldots \wedge b_m$" is true
    - Thus, in every model, h is true. Done. KB $\vdash_{BU}$ g implies KB $\models$ g

Lecture Overview

- Recap: Soundness, Correctness,
  Bottom-up proof procedure

- Bottom-up Proof Procedure
  - Soundness proof
  - Completeness proof

- Top-down Proof Procedure

Minimal Model

C := {};
repeat
  select clause $h \leftarrow b_1 \wedge \ldots \wedge b_m$ in KB
    such that $b_i \in C$ for all i, and $h \notin C$;
  $C := C \cup \{h\}$
until no more clauses can be selected. KB $\vdash_{BU}$ g if and only if $g \in C$

The C at the end of BU procedure is a fixed point:

- Further applications of our rule of derivation will not change C!

Definition (minimal model)
The minimal model MM is the interpretation in which every element of BU's fixed point C is true and every other atom is false.

Lemma: minimal model MM is a model of KB

Definition (minimal model)
The minimal model MM is the interpretation in which every element of BU's fixed point C is true and every other atom is false.

Definition (model)
A **model** of a knowledge base KB is an interpretation in which every clause in KB is true.

Proof by contradiction.

Assume (for contradiction) that MM is not a model of KB.

- Then there must exist some clause in KB which is false in MM
  - Like every clause in KB, it is of the form $h \leftarrow b_1 \wedge \ldots \wedge b_m$ (with $m \geq 0$).
- $h \leftarrow b_1 \wedge \ldots \wedge b_m$ can only be false in MM if each b_i is true in MM and h is false in MM.
  - Since each b_i is true in MM, each b_i must be in C as well.
  - BU would add h to C, so h would be true in MM
  - Contradiction! Thus, MM is a model of KB

Completeness of bottom-up procedure

Definition (completeness)

A proof procedure P is complete if KB $\models$ g implies KB $\vdash_P$ g.

complete: everything that logically follows from KB is derived

What do we need to prove to show that BU is complete?

Completeness of bottom-up procedure

Definition (completeness)

A proof procedure P is complete if $KB \models g$ implies $KB \vdash_P g$.

complete: everything that logically follows from KB is derived

What do we need to prove to show that BU is complete?

If g is true in all models of KB ($KB \models g$)
then $g \in C$ at the end of BU procedure ($KB \vdash_{BU} g$)

Direct proof based on Lemma about minimal model:

- Suppose $KB \models g$. Then g is true in all models of KB.
- Thus g is true in the minimal model.
- Thus $g \in C$ at the end of BU procedure.
- Thus $KB \vdash_{BU} g$. Done. $KB \models g$ implies $KB \vdash_{BU} g$

Summary for bottom-up proof procedure BU

- BU is sound:
  it derives only atoms that logically follow from KB

- BU is complete:
  it derives all atoms that logically follow from KB

- Together:
  it derives exactly the atoms that logically follow from KB

- And, it is efficient!
  - Linear in the number of clauses in KB
    - Each clause is used maximally once by BU

Learning Goals Up To Here

- PDCL syntax & semantics
  - Verify whether a logical statement belongs to the language of propositional definite clauses
  - Verify whether an interpretation is a model of a PDCL KB.
  - Verify when a conjunction of atoms is a logical consequence of a knowledge base
- Bottom-up proof procedure
  - Define/read/write/trace/debug the Bottom Up (**BU**) proof procedure
  - Prove that the BU proof procedure is sound and complete

Lecture Overview

- Recap: Soundness, Correctness,
  Bottom-up proof procedure

- Bottom-up Proof Procedure
  - Soundness proof
  - Completeness proof

- Top-down Proof Procedure

Bottom-up vs. Top-down

Bottom-up

KB $\Rightarrow$ C

g is proved if $g \in C$

When does BU look at the query g?

In every loop iteration

Never

At the end

At the beginning

Bottom-up vs. Top-down

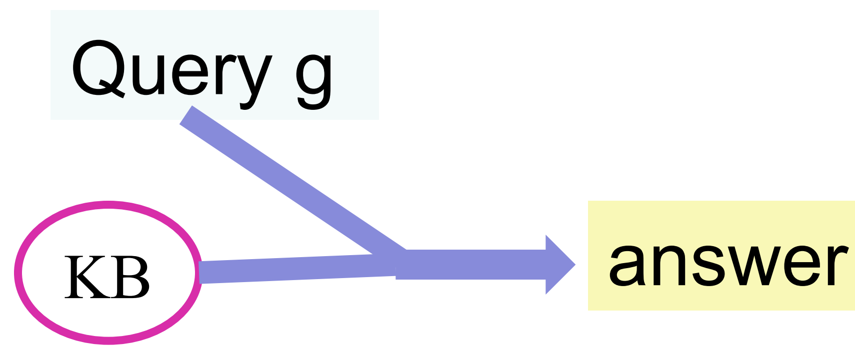
- **Key Idea of top-down:** search backward from a query g to determine if it can be derived from *KB*.

Bottom-up

g is proved if $g \in C$

When does BU look at the query g?
- Never
- It derives the same C regardless of the query

Top-down



We'll see how g is proved

TD performs a backward search starting at g

Top-down Ground Proof Procedure

Idea: search backward from a query to determine if it is a logical consequence of KB

An answer clause is of the form: $yes \leftarrow a_1 \wedge \ldots \wedge a_m$ where $a_1, \ldots, a_m$ are atoms

We express the query as an answer clause

– E.g. query $q_1 \wedge \ldots \wedge q_k$ is expressed as $yes \leftarrow q_1 \wedge \ldots \wedge q_k$

Basic operation: SLD Resolution of an answer clause

$$yes \leftarrow c_1 \wedge \ldots \wedge c_{i-1} \wedge c_i \wedge c_{i+1} \ldots \wedge c_m$$

on an atom c_i with another clause

$$c_i \leftarrow b_1 \wedge \ldots \wedge b_p$$

yields the clause

$$yes \leftarrow c_1 \wedge \ldots \wedge c_{i-1} \wedge b_1 \wedge \ldots \wedge b_p \wedge c_{i+1} \ldots \wedge c_m$$

Rules of derivation in top-down and bottom-up

Top-down:

SLD Resolution

$$\frac{yes \leftarrow c_1 \wedge \ldots \wedge c_m \qquad\qquad c_i \leftarrow b_1 \wedge \ldots \wedge b_p}{yes \leftarrow c_1 \wedge \ldots \wedge c_{i-1} \wedge b_1 \wedge \ldots \wedge b_p \wedge c_{i+1} \ldots \wedge c_m}$$

Bottom-up:

Generalized modus ponens

$$\frac{h \leftarrow b_1 \wedge \ \ldots \wedge b_m \qquad\qquad b_1 \wedge \ \ldots \wedge b_m}{h}$$

SLD Derivations

- An answer is an answer clause with m = 0.

  yes $\leftarrow$.

- A successful derivation from KB of query $?q_1 \wedge \dots \wedge q_k$ is a sequence of answer clauses $\gamma_0, \gamma_1, \dots, \gamma_n$ such that
  - γ_0 is the answer clause yes $\leftarrow q_1 \wedge \dots \wedge q_k$.
  - γ_i is obtained by resolving γ_{i-1} with a clause in KB, and
  - γ_n is an answer yes $\leftarrow$

- An unsuccessful derivation from KB of query $?q_1 \wedge \dots \wedge q_k$
  - We get to something like yes $\leftarrow b_1 \wedge \dots \wedge b_k$.
  - There is no clause in KB with any of the b_i as its head

Top-down Proof Procedure for PDCL

To solve the query $? \; q_1 \wedge \ldots \wedge q_k$:

ac:= yes $\leftarrow$ body, where body is $q_1 \wedge \ldots \wedge q_k$

repeat

select $q_i \in$ body;

choose clause $C \in$ KB, C is $q_i \leftarrow b_c$;

replace q_i in body by b_c

until ac is an answer (fail if no clause with q_i as head)

select: any choice will work

("don't care" non-determinism)

choose: non-deterministic, have to pick the right one

("don't know" non-determinism)

Example: successful derivation

$a \leftarrow b \wedge c.$ 1 $\boxed{a \leftarrow e \wedge f.}$ $b \leftarrow f \wedge k.$

4 $\boxed{c \leftarrow e.}$ $d \leftarrow k$ 3 5 $\boxed{e.}$

$f \leftarrow j \wedge e.$ 2 $\boxed{f \leftarrow c.}$ $j \leftarrow c.$

Query: ?a

γ_0: yes $\leftarrow$ a

γ_1: yes $\leftarrow$ e $\wedge$ f

γ_2: yes $\leftarrow$ e $\wedge$ c

γ_3: yes $\leftarrow$ c

γ_4: yes $\leftarrow$ e

γ_5: yes $\leftarrow$

Example: failing derivation

1 $\boxed{a \leftarrow b \wedge c.}$ $a \leftarrow e \wedge f.$ 2 $\boxed{b \leftarrow f \wedge k.}$

4,6 $\boxed{c \leftarrow e.}$ $d \leftarrow k$ 5, 7 $\boxed{e.}$

$f \leftarrow j \wedge e.$ 3 $\boxed{f \leftarrow c.}$ $j \leftarrow c.$

Query: ?a

γ_0: yes $\leftarrow$ a

γ_1: yes $\leftarrow$ b $\wedge$ c

γ_2: yes $\leftarrow$ f $\wedge$ k $\wedge$ c

γ_3: yes $\leftarrow$ c $\wedge$ k $\wedge$ c

γ_4: yes $\leftarrow$ e $\wedge$ k $\wedge$ c

γ_5: yes $\leftarrow$ k $\wedge$ c

γ_6: yes $\leftarrow$ k $\wedge$ e

γ_7: yes $\leftarrow$ k

There is no rule with k as its head, thus … fail

Correspondence between BU and TD proofs

If the following is a top-down derivation in a given KB, what would be the bottom-up derivation of the same query?

yes ← a.	{}
yes ← b ∧ f.	{h}
yes ← b ∧ g ∧ h.	{g,h}
yes ← c ∧ d ∧ g ∧ h.	{d,g,h}
yes ← d ∧ g ∧ h.	{c,d,g,h}
yes ← g ∧ h.	{b,c,d,g,h}
yes ← h.	{b,c,d,f,g,h}
yes ← .	{a,b,c,d,f,g,h}

Is the Top-down procedure sound and complete?

- Yes, since there is a 1:1 correspondence between top-down and bottom-up proofs.
  - The two methods derive exactly the same atoms (if the SLD resolution picks the successful derivations)
  - And the bottom-up procedure is sound and complete
  - Therefore the top-down procedure is sound and complete