soundness and completeness

Recall that $g$ is a logical consequence of $KB$, $KB \models g$, precisely if $g$ is true in all models of $KB$.

Let $\vdash$ be a mechanical procedure for deriving a formula $g$ from a knowledge base $KB$, written $KB \vdash g$.

$\vdash$ is sound if $KB \models g$ whenever $KB \vdash g$.

$\vdash$ is complete if $KB \vdash g$ whenever $KB \models g$. 
Soundness and completeness

Recall that $g$ is a logical consequence of $KB$, $KB \models g$, precisely if $g$ is true in all models of $KB$.

Let $\vdash$ be a mechanical procedure for deriving a formula $g$ from a knowledge base $KB$, written $KB \vdash g$.

$\vdash$ is sound if $KB \models g$ whenever $KB \vdash g$.

$\vdash$ is complete if $KB \vdash g$ whenever $KB \models g$.

Two extreme examples:

(1) $KB \vdash g$ for no $g$ sound
Soundness and completeness

Recall that $g$ is a *logical consequence of* $KB$, $KB \models g$, precisely if $g$ is true in all models of $KB$.

Let $\vdash$ be a mechanical procedure for deriving a formula $g$ from a knowledge base $KB$, written $KB \vdash g$.

$\vdash$ is *sound* if $KB \models g$ whenever $KB \vdash g$.

$\vdash$ is *complete* if $KB \vdash g$ whenever $KB \models g$.

Two extreme examples:

(1) $KB \vdash g$ for no $g$

sound

(2) $KB \vdash g$ for all $g$

complete
Soundness and completeness

Recall that $g$ is a logical consequence of $KB$, $KB \models g$, precisely if $g$ is true in all models of $KB$.

Let $\vdash$ be a mechanical procedure for deriving a formula $g$ from a knowledge base $KB$, written $KB \vdash g$.

$\vdash$ is sound if $KB \models g$ whenever $KB \vdash g$.

$\vdash$ is complete if $KB \vdash g$ whenever $KB \models g$.

Two extreme examples:

(1) $KB \vdash g$ for no $g$  \hspace{1cm} ‘say nothing’ undergenerates sound
(2) $KB \vdash g$ for all $g$ \hspace{1cm} ‘say everything’ overgenerates complete
Propositional KBs

Recall

\[
\begin{align*}
i & : - p, q. \\
i & : - r. \\
p. \\
r.
\end{align*}
\]

Let \( \text{KB} \vdash G \iff \text{prove([G],KB)} \)

Claim

(1) \( \vdash \) is sound (proved by induction)

(2) \( \vdash \) is not complete (homework question)
Propositional KBs

Recall

\[
i :- p,q.
\]

\[
i :- r.
\]

KB = \[[i,p,q],[i,r],[p],[r]\]

arc([H|T],N,KB) :- member([H|B],KB), append(B,T,N).

prove([],KB).

prove(Node,KB) :- arc(Node,Next,KB), prove(Next,KB).
Propositional KBs

Recall

\[ i : - p,q. \]
\[ i : - r. \]
\[ KB = \{[i,p,q],[i,r],[p],[r]\} \]

arc([H|T],N,KB) :- member([H|B],KB), append(B,T,N).

\[ p. \]
\[ r. \]

prove([],KB).

prove(Node,KB) :- arc(Node,Next,KB), prove(Next,KB).

Let

\[ KB \vdash G \iff \text{prove}([G],KB) \]

Claim

(1) \vdash \text{is sound} \quad \text{(proved by induction)}

(2) \vdash \text{is } not \text{ complete} \quad \text{(homework question)}
Logical consequences bottom-up

\[ C_0 := \emptyset \]

\[ C_{n+1} := \{ H \mid (\text{for some } B \subseteq C_n) \text{ member } ([H|B], KB) \} \]

\[ C := \bigcup_{n \geq 0} C_n \]

\[ = \bigcup_{n \leq k} C_n \quad \text{where } k = \text{number of clauses in } KB \]
Logical consequences bottom-up

\[ C_0 := \emptyset \]
\[ C_{n+1} := \{ H \mid (\text{for some } B \subseteq C_n) \text{ member}([H|B], KB) \} \]
\[ C := \bigcup_{n \geq 0} C_n \]
\[ = \bigcup_{n \leq k} C_n \quad \text{where } k = \text{number of clauses in } KB \]

\[
\begin{align*}
 i & :- p, q. \\
 i & :- r. \\
p & . \\
r & .
\end{align*}
\]

\[ KB = [[i,p,q],[i,r],[p],[r]] \]

\[ \text{arc}([H|T], N, KB) :- \text{member}([H|B], KB), \]
\[ \text{append}(B, T, N). \]

\[ C_1 = \{p, r\} \]
\[ C_2 = \{p, r, i\} = C_n \text{ for } n \geq 2 \]
A 0-ary predicate $p$ is interpreted by $I = \langle D, \phi, \pi \rangle$ as

\[ \pi(p) : D^0 \rightarrow \{\text{true, false}\}. \]
Substitutions and instances

A 0-ary predicate $p$ is interpreted by $I = \langle D, \phi, \pi \rangle$ as

$$\pi(p) : D^0 \to \{\text{true}, \text{false}\}.$$ 

Let $K$ be a set of constants.

A $K$-substitution is a function from a finite set of variables to $K$ — i.e. a set $\{V_1/c_1, \ldots, V_n/c_n\}$ of $c_i \in K$ and distinct variables $V_i$.

The application $e\theta$ of a $K$-substitution $\theta = \{V_1/c_1, \ldots, V_n/c_n\}$ to a clause $e$ is $e$ with each $V_i$ replaced by $c_i$.

e.g. $p(Z, U, Y, a, X)\{X/b, U/a, Z/b\} = p(b, a, Y, a, b)$.

A $K$-instance of $e$ is $e\theta$ for some $K$-substitution $\theta$.

Given a set $B$ of clauses and a $K$-substitution $\theta$, let

$$B\theta := \{e\theta \mid e \in B\}.$$
Bottom-up with substitutions

If $KB$ has constants from some non-empty finite set $K$, let

$$C^K_0 := \emptyset$$

$$C^K_{n+1} := \{ H\theta \mid \theta \text{ is a } K\text{-substitution s.t. } B\theta \subseteq C^K_n \text{ for some } B \text{ s.t. } \text{member}([H|B], KB) \}$$

$$C^K := \bigcup_{n \geq 0} C^K_n$$
Bottom-up with substitutions

If $KB$ has constants from some non-empty finite set $K$, let

$$
\begin{align*}
C_0^K & := \emptyset \\
C_{n+1}^K & := \{ H\theta \mid \theta \text{ is a } K\text{-substitution s.t. } B\theta \subseteq C_n^K \\
& \quad \text{for some } B \text{ s.t. } \text{member}([H|B], KB) \}
\end{align*}
$$

$$
C^K := \bigcup_{n \geq 0} C_n^K
$$

E.g. for $KB = [[[p(a, b)]], [q(X), p(X, Y)]]$ and $K = \{a, b\}$,

$$
\begin{align*}
C_1^K & = \{p(a, b)\} \\
C_2^K & = \{p(a, b), q(a)\} = C^K
\end{align*}
$$
Soundness & completeness via Herbrand

The Herbrand interpretation of a set $KB$ of clauses with constants from a non-empty set $K$ is the triple $I = \langle D, \phi, \pi \rangle$ where

- the domain $D$ is the set $K$ of constants
- $\phi$ is the identity function on $K$ (each constant in $K$ refers to itself)
- for each $n$-ary $p$ and $n$-tuple $c_1 \ldots c_n$ from $K$,

$$\pi(p)(c_1 \ldots c_n) = \text{true} \iff p(c_1 \ldots c_n) \in C^K$$
Soundness & completeness via Herbrand

The Herbrand interpretation of a set $KB$ of clauses with constants from a non-empty set $K$ is the triple $I = \langle D, \phi, \pi \rangle$ where

- the domain $D$ is the set $K$ of constants
- $\phi$ is the identity function on $K$ (each constant in $K$ refers to itself)
- for each $n$-ary $p$ and $n$-tuple $c_1 \ldots c_n$ from $K$,

$$\pi(p)(c_1 \ldots c_n) = \text{true} \iff p(c_1 \ldots c_n) \in C^K$$

**Fact.** $I$ is a model of $KB$, and every clause true in $I$ is true in every model of $KB$ (interpreting constants in $K$).
Soundness & completeness via Herbrand

The **Herbrand interpretation** of a set $KB$ of clauses with constants from a non-empty set $K$ is the triple $I = \langle D, \phi, \pi \rangle$ where

- the domain $D$ is the set $K$ of constants
- $\phi$ is the identity function on $K$ (each constant in $K$ refers to itself)
- for each $n$-ary $p$ and $n$-tuple $c_1 \ldots c_n$ from $K$,

\[ \pi(p)(c_1 \ldots c_n) = \text{true} \iff p(c_1 \ldots c_n) \in C^K \]

**Fact.** $I$ is a model of $KB$, and every clause true in $I$ is true in every model of $KB$ (interpreting constants in $K$).

**Corollary.** The bottom-up procedure with substitutions is sound and complete (for Datalog).