Theorem (Completeness theorem). Let $T$ be a set of propositional sentences, and $S$ be a propositional sentence. If $S$ is a tautological consequence of $T$ ($T \models S$), then $T \vdash S$.

Recall: $T \models S$ means that each valuation satisfying $T$ also satisfies $S$.

$T \vdash S$ means that $S$ is derivable from $T$ using the rules of Fitch.
Call a set of sentences $T$ formally inconsistent, if $T \vdash \bot$.
Example: $\{A \lor B, \lnot A, \lnot B\}$.
Otherwise, $T$ is called formally consistent.
Example: $\{A \lor B, A, \lnot B\}$

Lemma 1 $T \cup \{\lnot S\} \vdash \bot$ iff $T \vdash S$.

Reformulation of completeness: Every formally consistent set of sentences is tt-satisfiable.
A set $\mathcal{T}$ is formally complete, if for any sentence $S$ in the same language as $\mathcal{T}$, either $\mathcal{T} \vdash S$ or $\mathcal{T} \vdash \neg S$.

Lemma 2 For $\mathcal{T}$ formally complete and consistent,

1. $\mathcal{T} \vdash R \land S$ iff $\mathcal{T} \vdash R$ and $\mathcal{T} \vdash S$.
2. $\mathcal{T} \vdash R \lor S$ iff $\mathcal{T} \vdash R$ or $\mathcal{T} \vdash S$.
3. $\mathcal{T} \vdash \neg S$ iff not $\mathcal{T} \vdash S$.
4. $\mathcal{T} \vdash R \rightarrow S$ iff $\mathcal{T} \not\vdash R$ or $\mathcal{T} \vdash S$.
5. $\mathcal{T} \vdash R \leftrightarrow S$ iff ($\mathcal{T} \vdash R$ iff $\mathcal{T} \vdash S$).
Lemma 3 Every formally complete and consistent set is tt-satisfiable.

Lemma 4 A set of sentence is formally complete iff for every atomic sentence \( A \), either \( \mathcal{T} \vdash A \) or \( \mathcal{T} \vdash \neg A \).

Lemma 5 Every formally consistent set of sentences can be expanded to a formally consistent, formally complete set of sentences.
**Theorem** (Completeness theorem). Let $\mathcal{T}$ be a set of sentences of a first-order language $L$ and $S$ be a sentence of the same language. If $S$ is a first-order consequence of $\mathcal{T}$ ($\mathcal{T} \models S$), then $\mathcal{T} \vdash S$.

*Recall:* $\mathcal{T} \models S$ means that each first-order structure satisfying $\mathcal{T}$ also satisfies $S$.

$\mathcal{T} \vdash S$ means that $S$ is derivable from $\mathcal{T}$ using the rules of Fitch.
Adding witnessing constants  $L$ is enriched to $L_H$ by adding infinitely many constants, the *witnessing constants*.

The Henkin theory  Isolate a theory $\mathcal{H}$ in the enriched language $L_H$, which contains *Henkin witnessing axioms*.

The Elimination theorem  For any proof with sentences in $L$ or from $\mathcal{H}$ as premises, and a sentence in $L$ as conclusion, we can eliminate the premises from $\mathcal{H}$ from the proof.

The Henkin construction  For every truth assignment $h$ assigning TRUE to all formulas in $\mathcal{H}$, there is a first-order structure $\mathcal{M}_h$ such that $\mathcal{M}_h$ makes true all first-order sentence that $h$ makes true.
Assume that $\mathcal{T} \models S$.

1. Hence, there is no model of $\mathcal{T} \cup \{\neg S\}$.
2. By the Henkin construction, there is no assignment $h$ satisfying $\mathcal{T} \cup \mathcal{H} \cup \{\neg S\}$.
3. Hence, $S$ is a tautological consequence of $\mathcal{T} \cup \mathcal{H}$.
4. By completeness of propositional logic, $\mathcal{T} \cup \mathcal{H} \vdash S$.
5. By the elimination theorem, $\mathcal{T} \vdash S$. 
Adding witnessing constants

For each well-formed formula $P$ of $L$ with exactly one free variable, add a new constant $c_P$, $P$'s **witnessing constant**.

E.g. for $Small(x) \land Cube(x)$, we introduce $c_{Small(x) \land Cube(x)}$.

This process has to be iterated:

$$L = L_0 \subseteq L_1 \subseteq L_2 \subseteq \ldots$$

E.g. for $Smaller(y, c_{Small(x) \land Cube(x)})$, we introduce $c_{Smaller(y, c_{Small(x) \land Cube(x)})}$.

Each witnessing constant appears in some $L_n$ for the first time. $n$ is called its **date of birth**.
The Henkin witnessing axioms

$\exists x \ P(x) \rightarrow P(c_{P(x)})$

intuitive idea:
if there is something that satisfies $P(x)$, then the object named by $c_{P(x)}$ provides an example
The Henkin theory

H1  \( \exists x \ P(x) \rightarrow P(c_{P(x)}) \)  \( (\text{witnessing axioms, } \hat{=} \ \exists \text{Elim}) \)

H2  \( P(c) \rightarrow \exists x \ P(x) \)  \( (\hat{=} \ \exists \text{Intro}) \)

H3  \( \neg \forall x \ P(x) \leftrightarrow \exists x \ \neg P(x) \)  \( (\text{reduces } \forall \text{ to } \exists) \)

H4  \( c = c \)  \( (\hat{=} = \text{Intro}) \)

H5  \( (P(c) \land c = d) \rightarrow P(d) \)  \( (\hat{=} = \text{Elim}) \)
Independence lemma

If $c_P$ and $c_Q$ are two witnessing constants and the date of birth of $c_P$ is less than or equal to that of $c_Q$, then $c_Q$ does not appear in the witnessing axiom of $c_P$. 
Elimination Theorem

For any proof with sentences in $L$ or from $\mathcal{H}$ as premises, and a sentence in $L$ as conclusion, we can eliminate the premises from $\mathcal{H}$ from the proof.
Deduction Theorem If $\mathcal{T} \cup \{P\} \vdash Q$, then $\mathcal{T} \vdash P \rightarrow Q$.

Lemma 1 If $\mathcal{T} \vdash P \rightarrow Q$ and $\mathcal{T} \vdash \neg P \rightarrow Q$, then $\mathcal{T} \vdash Q$.

Lemma 2 If $\mathcal{T} \vdash (P \rightarrow Q) \rightarrow R$, then $\mathcal{T} \vdash \neg P \rightarrow R$, and $\mathcal{T} \vdash Q \rightarrow R$.

Lemma 3 Let $\mathcal{T}$ be a set of sentences of some first-order language $L$, and $Q$ be a sentence. Let $P(x)$ be a formula of $L$ with one free variable and which does not contain $c$. If $\mathcal{T} \vdash P(c) \rightarrow Q$ and $c$ does not appear in $\mathcal{T}$ or $Q$, then $\mathcal{T} \vdash \exists x \ P(x) \rightarrow Q$. 
Lemma 4 Let $\mathcal{T}$ be a set of sentences of some first-order language $L$, and $Q$ be a sentence of $L$. Let $P(x)$ be a formula of $L$ with one free variable and which does not contain $c$. If $\mathcal{T} \cup \{\exists x \ P(x) \rightarrow P(c)\} \vdash Q$ and $c$ does not appear in $\mathcal{T}$ or $Q$, then $\mathcal{T} \vdash Q$. 
Lemma 5 Let $\mathcal{T}$ be a set of first-order sentences, let $P(x)$ be a formula with one free variable, and let $c$ and $d$ be constant symbols. The following are provable in $\mathcal{F}$:

$$
\begin{align*}
P(c) &\rightarrow \exists x \ P(x) \\
\neg \forall x \ P(x) &\leftrightarrow \exists x \ \neg P(x) \\
&\quad \quad \quad c = c \\
(P(c) \land c = d) &\rightarrow P(d)
\end{align*}
$$
Elimination Theorem

For any proof with sentences in $L$ or from $\mathcal{H}$ as premises, and a sentence in $L$ as conclusion, we can eliminate the premises from $\mathcal{H}$ from the proof.
The Henkin construction

For every truth assignment $h$ assigning \texttt{TRUE} to all formulas in $\mathcal{H}$, there is a first-order structure $M_h$ such that $M_h$ makes true all first-order sentence that $h$ makes true.

Given $h$, construct $M$ as follows:

- $D^M$: set of constant symbols in $L_H$
- $M(c)$ is just $c$
- $M(R)$ is $\{\langle c_1, \ldots, c_n \rangle \mid h(R(c_1, \ldots, c_n)) = \texttt{TRUE}\}$
  
  $c \equiv d$ if and only if $h(c = d) = \texttt{TRUE}$

$$M_h = M/\equiv$$
The truth lemma

For any sentence $S$ of $L_H$, $M_h \models S$ iff $h(S) = true$. 
Consequences of the Completeness Theorem

Theorem (Compactness theorem) Let $\mathcal{T}$ be a set of first-order sentences in a first-order language $L$. If every finite subset of $\mathcal{T}$ is first order-satisfiable, then $\mathcal{T}$ itself is first-order satisfiable.
Let $L$ be the language of Peano arithmetic. Then there exists a first-order structure $M$ such that

1. $M$ contains all the natural numbers in its domain,
2. $M$ also contains elements greater than all the natural numbers, but
3. $M$ makes true exactly the same sentences of $L$ as are true about the natural numbers.
Exercises

- 17.4-17.16, 19.1-19.23
- deadline: 5.1.2010