CS 511, Fall 2024, Lecture Slides 23

First-Order Logic: Soundness and Completeness

Assaf Kfoury

7 October 2024

consistency

Γ is a set of WFF's.

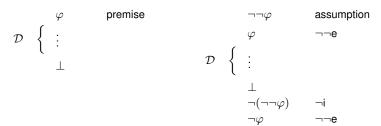
- ightharpoonup Γ is consistent iff $\Gamma \not\vdash \bot$.
 - **FACT.** The following three conditions are equivalent:
 - 1. Γ is consistent.
 - 2. For no WFF φ is it the case that both $\Gamma \vdash \varphi$ and $\Gamma \vdash \neg \varphi$.
 - 3. There is at least one WFF φ such that $\Gamma \not\vdash \varphi$.
 - ► Contrapositive FACT. The following conditions are equivalent:
 - 4. Γ is inconsistent.
 - 5. There is a WFF φ such that both $\Gamma \vdash \varphi$ and $\Gamma \vdash \neg \varphi$.
 - 6. For every WFF φ , it holds that $\Gamma \vdash \varphi$.

Proof.

- (4) \Rightarrow (6): Let $\Gamma \vdash \bot$. By the rule " \bot elimination", we add one more step in the proof to obtain $\Gamma \vdash \varphi$, which holds for every φ .
- (6) \Rightarrow (5): Immediate.
- (5) \Rightarrow (4): By the rule " \neg elimination", from the derivations $\Gamma \vdash \varphi$ and $\Gamma \vdash \neg \varphi$, we get $\Gamma \vdash \bot$.

consistency (continued)

- ▶ Theorem. Let Γ be a set of WFF's and φ a WFF. We then have the two following (equivalent) statements:
- 1. $\Gamma \cup \{\varphi\}$ is inconsistent iff $\Gamma \vdash \neg \varphi$.
- 2. $\Gamma \cup \{\varphi\}$ is consistent iff $\Gamma \not\vdash \neg \varphi$.
- ▶ **Proof.** It suffices to prove part 1 only. The simple right-to-left implication is left to you. For the left-to-right, suppose $\Gamma \cup \{\varphi\}$ is inconsistent. Hence we are given a formal derivation of the form on the left, and we build a new one on the right. The new one starts by opening a box with assumption $\neg\neg\varphi$, then uses rule " $\neg\neg e$ " and copies the given derivation with no change, and closes the initial box with rule " \neg i":



The new formal derivation on the right shows that $\Gamma \vdash \neg \varphi$ is a derivable sequent.

soundness

Theorem. Let Γ be a set of WFF's and φ a WFF.

```
If \Gamma \vdash \varphi then \Gamma \models \varphi. (most common form for "soundness")
```

Proof. Left to you. Consult also how each deduction rule is justified [LCS, Section 2.3].

soundness

Theorem. Let Γ be a set of WFF's and φ a WFF.

```
If \Gamma \vdash \varphi then \Gamma \models \varphi. (most common form for "soundness")
```

▶ Proof. Left to you. Consult also how each deduction rule is justified [LCS, Section 2.3].

Another form for "soundness" is the following:

- **Corollary.** If Γ is satisfiable, then Γ is consistent.
- ▶ **Proof.** Suppose Γ is inconsistent. Then there is a WFF φ such that both $\Gamma \vdash \varphi$ and $\Gamma \vdash \neg \varphi$, by part (5) on slide 3. By the previous theorem, both $\Gamma \models \varphi$ and $\Gamma \models \neg \varphi$, which is a contradiction.

completeness

One form of "completeness" is the following:

Theorem. Let Γ be a set of sentences (closed WFF's). If Γ is consistent, then Γ is satisfiable.

Proof. By the Model-Existence Lemma (not in the book [LCS], and not included in these notes, look up "model-existence" lemma or theorem on the Web).

completeness

One form of "completeness" is the following:

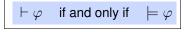
- **Theorem.** Let Γ be a set of sentences (closed WFF's). If Γ is consistent, then Γ is satisfiable.
- Proof. By the Model-Existence Lemma (not in the book [LCS], and not included in these notes, look up "model-existence" lemma or theorem on the Web).

Another form of "completeness", which is the most common:

- **Corollary.** If $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$.
- ▶ **Proof.** Suppose $\Gamma \not\vdash \varphi$. Then $\Gamma \not\vdash \neg \neg \varphi$. So that $\Gamma \cup \{\neg \varphi\}$ is consistent, by part 2 of theorem on slide 4. By the theorem on this slide, there is a model \mathcal{M} of $\Gamma \cup \{\neg \varphi\}$. Hence, \mathcal{M} is a model of Γ but not of φ . Hence, $\Gamma \not\models \varphi$.

soundness and completeness - short form

For all WFF φ



(THIS PAGE INTENTIONALLY LEFT BLANK)