ECE750T-28: Computer-aided Reasoning for Software Engineering

Lecture 1 and 2 Review: Introduction to Logic in SE

Vijay Ganesh
(Original notes from Isil Dillig)

What are Logics?

- Precise mathematical languages with well-defined syntax and semantics

What are Logics?

- Precise mathematical languages with well-defined syntax and semantics
- Syntax: Meta-rules for defining well-formed formulas

What are Logics?

- Precise mathematical languages with well-defined syntax and semantics
- Syntax: Meta-rules for defining well-formed formulas
- Semantics: Interpretation/models

What are Logics?

- Precise mathematical languages with well-defined syntax and semantics
- Syntax: Meta-rules for defining well-formed formulas
- Semantics: Interpretation/models
- Forms of valid reasoning: Deductive, inductive, abductive,...

What are Logics?

- Precise mathematical languages with well-defined syntax and semantics
- Syntax: Meta-rules for defining well-formed formulas
- Semantics: Interpretation/models
- Forms of valid reasoning: Deductive, inductive, abductive,...
- Proof systems: Intuinistic vs. classical

What are Logics?

- Precise mathematical languages with well-defined syntax and semantics
- Syntax: Meta-rules for defining well-formed formulas
- Semantics: Interpretation/models
- Forms of valid reasoning: Deductive, inductive, abductive,...
- Proof systems: Intuinistic vs. classical
- Fields of study: proof, model, set, recursion, and type theory

What are Logics?

- Precise mathematical languages with well-defined syntax and semantics
- Syntax: Meta-rules for defining well-formed formulas
- Semantics: Interpretation/models
- Forms of valid reasoning: Deductive, inductive, abductive,...
- Proof systems: Intuinistic vs. classical
- Fields of study: proof, model, set, recursion, and type theory
- Questions studied: Proof and truth, Provability, Decidability, Complexity, Foundations,...

What are Logics?

- Precise mathematical languages with well-defined syntax and semantics
- Syntax: Meta-rules for defining well-formed formulas
- Semantics: Interpretation/models
- Forms of valid reasoning: Deductive, inductive, abductive,...
- Proof systems: Intuinistic vs. classical
- Fields of study: proof, model, set, recursion, and type theory
- Questions studied: Proof and truth, Provability, Decidability, Complexity, Foundations,...
- Properties of logics: Soundness, completeness, compactness, expressive power, decidability,...

Review of Propositional Logic: Syntax

Atom truth symbols $\top$ ("true") and $\bot$ ("false")
propositional variables $p, q, r, p_1, q_1, r_1, \cdots$

Review of Propositional Logic: Syntax

Atom	truth symbols $\top$ ("true") and $\bot$ ("false") propositional variables $p, q, r, p_1, q_1, r_1, \cdots$
Literal	atom α or its negation $\neg\alpha$

Review of Propositional Logic: Syntax

Atom truth symbols $\top$ ("true") and $\bot$ ("false")
propositional variables $p, q, r, p_1, q_1, r_1, \cdots$

Literal atom α or its negation $\neg\alpha$

Formula literal or application of a
logical connective to formulae F, F_1, F_2

Review of Propositional Logic: Syntax

Atom truth symbols $\top$ ("true") and $\bot$ ("false")
propositional variables $p, q, r, p_1, q_1, r_1, \cdots$

Literal atom α or its negation $\neg\alpha$

Formula literal or application of a
logical connective to formulae F, F_1, F_2

$\neg F$	"not"	(negation)

Review of Propositional Logic: Syntax

Atom	truth symbols $\top$ ("true") and $\bot$ ("false") propositional variables $p, q, r, p_1, q_1, r_1, \cdots$
Literal	atom α or its negation $\neg\alpha$
Formula	literal or application of a logical connective to formulae F, F_1, F_2

$\neg F$	"not"	(negation)
$F_1 \wedge F_2$	"and"	(conjunction)

Review of Propositional Logic: Syntax

Atom	truth symbols $\top$ ("true") and $\bot$ ("false") propositional variables $p, q, r, p_1, q_1, r_1, \cdots$
Literal	atom α or its negation $\neg\alpha$
Formula	literal or application of a logical connective to formulae F, F_1, F_2

$\neg F$	"not"	(negation)
$F_1 \wedge F_2$	"and"	(conjunction)
$F_1 \vee F_2$	"or"	(disjunction)

Review of Propositional Logic: Syntax

Atom truth symbols $\top$ ("true") and $\bot$ ("false")
propositional variables $p, q, r, p_1, q_1, r_1, \cdots$

Literal atom α or its negation $\neg\alpha$

Formula literal or application of a
logical connective to formulae F, F_1, F_2

$\neg F$	"not"	(negation)
$F_1 \wedge F_2$	"and"	(conjunction)
$F_1 \vee F_2$	"or"	(disjunction)
$F_1 \rightarrow F_2$	"implies"	(implication)

Review of Propositional Logic: Syntax

Atom	truth symbols $\top$ ("true") and $\perp$ ("false") propositional variables $p, q, r, p_1, q_1, r_1, \cdots$
Literal	atom α or its negation $\neg\alpha$
Formula	literal or application of a logical connective to formulae F, F_1, F_2

$\neg F$	"not"	(negation)
$F_1 \wedge F_2$	"and"	(conjunction)
$F_1 \vee F_2$	"or"	(disjunction)
$F_1 \rightarrow F_2$	"implies"	(implication)
$F_1 \leftrightarrow F_2$	"if and only if"	(iff)

Interpretations in Propositional Logic

- An interpretation I for a formula F in propositional logic is a mapping from each propositional variables in F to exactly one truth value

$$I : \{p \mapsto \top, q \mapsto \bot, \cdots\}$$

Interpretations in Propositional Logic

- An interpretation I for a formula F in propositional logic is a mapping from each propositional variables in F to exactly one truth value

$$I : \{p \mapsto \top, q \mapsto \bot, \cdots\}$$

- For a formula F with 2 propositional variables, how many interpretations are there?

Interpretations in Propositional Logic

- An interpretation I for a formula F in propositional logic is a mapping from each propositional variables in F to exactly one truth value

$$I : \{p \mapsto \top, q \mapsto \bot, \cdots\}$$

- For a formula F with 2 propositional variables, how many interpretations are there?

- In general, for formula with n propositional variables, how many interpretations?

Entailment

- Under an interpretation, every propositional formula evaluates to T or F

Formula F + Interpretation I = Truth value

Entailment

- Under an interpretation, every propositional formula evaluates to T or F

Formula F + Interpretation I = Truth value

- We write $I \models F$ if F evaluates to $\top$ under I (satisfying interpretation)

Entailment

- Under an interpretation, every propositional formula evaluates to T or F

Formula F + Interpretation I = Truth value

- We write $I \models F$ if F evaluates to $\top$ under I (satisfying interpretation)
- Similarly, $I \not\models F$ if F evaluates to $\bot$ under I (falsifying interpretation).

Inductive Definition of Propositional Semantics

<u>Base Cases</u>:

$I \models \top$

Inductive Definition of Propositional Semantics

Base Cases:

$I \models \top \qquad I \not\models \bot$

Inductive Definition of Propositional Semantics

<u>Base Cases</u>:

$$I \models \top \qquad I \not\models \bot$$

$$I \models p \quad \textsf{iff} \quad I[p] = \top$$

Inductive Definition of Propositional Semantics

<u>Base Cases</u>:

$$I \models \top \qquad I \not\models \bot$$

$$I \models p \quad \text{iff} \quad I[p] = \top$$
$$I \not\models p \quad \text{iff} \quad I[p] = \bot$$

Inductive Definition of Propositional Semantics

Base Cases:

$$I \models \top \qquad I \not\models \bot$$

$$I \models p \quad \text{iff} \quad I[p] = \top$$
$$I \not\models p \quad \text{iff} \quad I[p] = \bot$$

Inductive Cases:

Inductive Definition of Propositional Semantics

<u>Base Cases</u>:

$$I \models \top \qquad I \not\models \bot$$

$$\begin{array}{lll} I \models p & \text{iff} & I[p] = \top \\ I \not\models p & \text{iff} & I[p] = \bot \end{array}$$

<u>Inductive Cases</u>:

$$I \models \neg F \qquad \text{iff } I \not\models F$$

Inductive Definition of Propositional Semantics

Base Cases:

$$I \models \top \qquad I \not\models \bot$$

$$I \models p \quad \text{iff} \quad I[p] = \top$$
$$I \not\models p \quad \text{iff} \quad I[p] = \bot$$

Inductive Cases:

$$I \models \neg F \qquad \text{iff } I \not\models F$$
$$I \models F_1 \wedge F_2 \qquad \text{iff } I \models F_1 \text{ and } I \models F_2$$

Inductive Definition of Propositional Semantics

Base Cases:

$I \models \top \qquad I \not\models \bot$

$I \models p \quad$ iff $\quad I[p] = \top$

$I \not\models p \quad$ iff $\quad I[p] = \bot$

Inductive Cases:

$I \models \neg F \quad$ iff $I \not\models F$

$I \models F_1 \wedge F_2 \quad$ iff $I \models F_1$ and $I \models F_2$

$I \models F_1 \vee F_2 \quad$ iff $I \models F_1$ or $I \models F_2$

Inductive Definition of Propositional Semantics

Base Cases:

$I \models \top \qquad I \not\models \bot$

$I \models p \quad$ iff $\quad I[p] = \top$

$I \not\models p \quad$ iff $\quad I[p] = \bot$

Inductive Cases:

$I \models \neg F \qquad$ iff $I \not\models F$

$I \models F_1 \wedge F_2 \qquad$ iff $I \models F_1$ and $I \models F_2$

$I \models F_1 \vee F_2 \qquad$ iff $I \models F_1$ or $I \models F_2$

$I \models F_1 \rightarrow F_2$

Inductive Definition of Propositional Semantics

Base Cases:

$I \models \top \qquad I \not\models \bot$

$I \models p$ iff $I[p] = \top$

$I \not\models p$ iff $I[p] = \bot$

Inductive Cases:

$I \models \neg F$ iff $I \not\models F$

$I \models F_1 \wedge F_2$ iff $I \models F_1$ and $I \models F_2$

$I \models F_1 \vee F_2$ iff $I \models F_1$ or $I \models F_2$

$I \models F_1 \rightarrow F_2$ iff, $I \not\models F_1$ or $I \models F_2$

Inductive Definition of Propositional Semantics

Base Cases:

$I \models \top \qquad I \not\models \bot$

$I \models p \quad$ iff $\quad I[p] = \top$

$I \not\models p \quad$ iff $\quad I[p] = \bot$

Inductive Cases:

$I \models \neg F$ iff $I \not\models F$

$I \models F_1 \wedge F_2$ iff $I \models F_1$ and $I \models F_2$

$I \models F_1 \vee F_2$ iff $I \models F_1$ or $I \models F_2$

$I \models F_1 \rightarrow F_2$ iff, $I \not\models F_1$ or $I \models F_2$

$I \models F_1 \leftrightarrow F_2$

Inductive Definition of Propositional Semantics

Base Cases:

$I \models \top \qquad I \not\models \bot$

$I \models p$ iff $I[p] = \top$

$I \not\models p$ iff $I[p] = \bot$

Inductive Cases:

$I \models \neg F$ iff $I \not\models F$

$I \models F_1 \wedge F_2$ iff $I \models F_1$ and $I \models F_2$

$I \models F_1 \vee F_2$ iff $I \models F_1$ or $I \models F_2$

$I \models F_1 \rightarrow F_2$ iff, $I \not\models F_1$ or $I \models F_2$

$I \models F_1 \leftrightarrow F_2$ iff, $I \models F_1$ and $I \models F_2$

Inductive Definition of Propositional Semantics

Base Cases:

$$I \models \top \qquad I \not\models \bot$$

$$\begin{array}{lll} I \models p & \text{iff} & I[p] = \top \\ I \not\models p & \text{iff} & I[p] = \bot \end{array}$$

Inductive Cases:

$$\begin{array}{ll} I \models \neg F & \text{iff } I \not\models F \\ I \models F_1 \wedge F_2 & \text{iff } I \models F_1 \text{ and } I \models F_2 \\ I \models F_1 \vee F_2 & \text{iff } I \models F_1 \text{ or } I \models F_2 \\ I \models F_1 \rightarrow F_2 & \text{iff, } I \not\models F_1 \text{ or } I \models F_2 \\ I \models F_1 \leftrightarrow F_2 & \text{iff, } I \models F_1 \text{ and } I \models F_2 \\ & \quad\text{or } I \not\models F_1 \text{ and } I \not\models F_2 \end{array}$$

Proof Systems

- Proof Systems
- For example, Semantic argument method we studied is a proof system

Proof Systems

- Proof Systems
- For example, Semantic argument method we studied is a proof system
- Proof rules, Axioms

Proof Systems

- Proof Systems
- For example, Semantic argument method we studied is a proof system
- Proof rules, Axioms
- What is a proof?

Proof Systems

- Proof Systems
- For example, Semantic argument method we studied is a proof system
- Proof rules, Axioms
- What is a proof?
- A finite graph constructed out of proof rules

Proof Systems

- Proof Systems
- For example, Semantic argument method we studied is a proof system
- Proof rules, Axioms
- What is a proof?
- A finite graph constructed out of proof rules
- Validity and Provability

Proof Systems

- Proof Systems
- For example, Semantic argument method we studied is a proof system
- Proof rules, Axioms
- What is a proof?
- A finite graph constructed out of proof rules
- Validity and Provability
- Soundness: If the proof system says that T is provable, then it is valid

Proof Systems

- Proof Systems
- For example, Semantic argument method we studied is a proof system
- Proof rules, Axioms
- What is a proof?
- A finite graph constructed out of proof rules
- Validity and Provability
- Soundness: If the proof system says that T is provable, then it is valid
- Completeness: If T is valid, then it is provable by the proof system

Proof Systems

- Proof Systems
- For example, Semantic argument method we studied is a proof system
- Proof rules, Axioms
- What is a proof?
- A finite graph constructed out of proof rules
- Validity and Provability
- Soundness: If the proof system says that T is provable, then it is valid
- Completeness: If T is valid, then it is provable by the proof system
- Termination: The proof system terminates on all inputs