CS389L: Automated Logical Reasoning
Lecture 1: Introduction and Review of Basics

İsıl Dillig

Course staff
- **Instructor**: İsıl Dillig
- **Research areas**: Program verification/synthesis, program analysis, decision procedures
- **E-mail**: isil@cs.utexas.edu
- **Office hours**: Tuesday-Thursday after class in GDC 5.726
- **TA**: Jessica Hoffman (hoffmann@cs.utexas.edu)
- **Office hours**: Wed 10-12 am

What is this Course About?
- This course is about computational logic.
- Explore logical theories widely used in computer science.
- Learn about decision procedures that allow us to automatically decide satisfiability and validity of logical formulas.

Why Should You Care?
Logic is a fundamental part of computer science:
- **Artificial intelligence**: planning, automated game playing
- **Program analysis**: Static analysis, software verification
- **Software engineering**: automated test generation

Overview of the Course
- **Part I**: Propositional logic and SAT solving
  - Normal forms, DPLL
  - Modern SAT solvers
  - Applications and variations
  - Binary Decision Diagrams
  - Quantified Boolean Logic, MaxSAT (guest lectures)

Overview, cont
- **Part II**: First-order theorem proving
  - Semantics of FOL and theoretical properties
  - Basics of first-order theorem proving
  - Decidable fragments of FOL
Overview, cont.

- Part III: First-order theories and SMT solving
  - Semantics of commonly-used first-order theories
  - Decision procedures (equality, LRA, Presburger arithmetic)
  - Combining theories, Nelson-Oppen method
  - DPLL(T) and practical SMT solvers

Overview, cont.

- Part IV: Applications
  - Program verification (Hoare logic)
  - Symbolic execution
  - Invariant generation

Logistics

- Class meets every Tuesday, Thursday 5-6:15 pm
- All class material (slides, relevant reading etc.) posted on the course website:
  - http://www.cs.utexas.edu/~idillig/cs389l
- Also have a Piazza page:
  - piazza.com/utexas/spring2016/cs389l
- No required books, but following textbooks may be useful

Reference #1

- The Calculus of Computation
  by Aaron Bradley and Zohar Manna

Reference #2

- Decision Procedures: An Algorithmic Point of View
  by Daniel Kroening and Ofer Strichman

Grading

- No programming assignments or big projects
- Expect problem set once every 2 weeks (30% grade)
- 2 in-class, closed-book exams (30% of grade each)
- Class attendance and participation (10% of grade)
Exams

- **Exam dates:** March 3, May 3 – put these dates on your calendar! (free during finals week)
- All exams closed-book, closed-notes, closed-laptop, closed-phone etc.
- Please introduce yourself!

Let’s get started!

- **Today:** Review of basic propositional logic
- Should already know this stuff – quick refresher!

Review of Propositional Logic: PL Syntax

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

**Literal**
- atom \( \alpha \) or its negation \( \neg \alpha \)

**Formula**
- literal or application of a logical connective to formulae \( F_1, F_2 \)

\[
\begin{align*}
\neg F & \quad \text{"not"} \quad \text{(negation)} \\
F_1 \land F_2 & \quad \text{"and"} \quad \text{(conjunction)} \\
F_1 \lor F_2 & \quad \text{"or"} \quad \text{(disjunction)} \\
F_1 \rightarrow F_2 & \quad \text{"implies"} \quad \text{(implication)} \\
F_1 \leftrightarrow F_2 & \quad \text{"if and only if"} \quad \text{(iff)}
\end{align*}
\]

PL Semantics

- **Interpretation** \( I \) : mapping from each propositional variables in \( F \) to exactly one truth value

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

- **Formula** \( F \) + Interpretation \( I = \text{Truth value} \)

- 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 PL Semantics

**Base Cases:**
\[
\begin{align*}
I \models \top & \quad \text{if } I \not\models \bot \\
I \models p & \quad \text{if } I[p] = \top \\
I \not\models p & \quad \text{if } I[p] = \bot
\end{align*}
\]

**Inductive Cases:**
\[
\begin{align*}
I \models \neg F & \quad \text{if } I \not\models F \\
I \models F_1 \land F_2 & \quad \text{if } I \models F_1 \text{ and } I \models F_2 \\
I \models F_1 \lor F_2 & \quad \text{if } I \models F_1 \text{ or } I \models F_2 \\
I \models F_1 \rightarrow F_2 & \quad \text{if } I \models F_1 \text{ and } I \not\models F_2 \\
& \quad \text{ or } I \not\models F_1 \text{ and } I \not\models F_2
\end{align*}
\]

Simple Example

- Consider formula \( F_1 : (p \land q) \rightarrow (p \lor \neg q) \)

- What is its truth value under interpretation \( F_1 : \{ p \mapsto \top, q \mapsto \bot \} \)?

- What about formula \( F_2 : (p \leftrightarrow \neg q) \rightarrow (q \rightarrow r) \) and interpretation \( F_2 = \{ p \mapsto \bot, q \mapsto \top, r \mapsto \top \}? \)
### Satisfiability and Validity

- **F** is **satisfiable** iff there exists an interpretation *I* such that *I* \( \models \) *F*.
- **F** is **valid** iff for all interpretations *I*, *I* \( \models \) *F*.
- **F** is **contingent** if it is satisfiable but not valid.
- Duality between satisfiability and validity:
  
  \[ \text{F is valid iff } \neg \text{F is unsatisfiable} \]

- Thus, if we have a procedure for checking satisfiability, this also allows us to decide validity.

### Examples

- Sat, unsat, or valid?
  - \((p \land q) \rightarrow \neg p\)
  - \((p \rightarrow q) \rightarrow (\neg(p \land \neg q))\)
  - \((p \rightarrow (q \rightarrow r)) \land \neg((p \land q) \rightarrow r)\)

### Deciding Satisfiability and Validity

- Before we talk about practical algorithms for deciding satisfiability, let’s review some simple techniques
- Two very simple techniques:
  - Truth table method: essentially a search-based technique
  - Semantic argument method: deductive way of deciding satisfiability
- Modern SAT solvers combine search and deduction!

### Method 1: Truth Tables

**Example**

\[ F : (p \land q) \rightarrow (p \lor \neg q) \]

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p \land q</th>
<th>\neg q</th>
<th>p \lor \neg q</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Thus \( F \) is valid.

### Another Example

\[ F : (p \lor q) \rightarrow (p \land q) \]

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p \lor q</th>
<th>p \land q</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Thus \( F \) is satisfiable, but invalid.

### Bad Idea!

- Truth tables are completely brute-force, impractical \( \Rightarrow \) must list all \( 2^n \) interpretations!
- Does not work for any other logic where domain is not finite (e.g., first-order logic)
Method 2: Semantic Argument

- Semantic argument method is essentially a proof by contradiction, and is also applicable for theories with non-finite domain.
- Main idea: Assume $F$ is not valid $\Rightarrow$ there exists some falsifying interpretation $I$ such that $I \not| F$
- Apply proof rules.
- If we derive a contradiction in every branch of the proof, then $F$ is valid.
- If there is some branch where we cannot derive $\bot$ (after applying all proof rules), then $F$ is not valid.

The Proof Rules (I)

- According to semantics of negation, from $I \models \neg F$, we can deduce $I \not| F$:
  $I \models \neg F$
  $I \not| F$

- Similarly, from $I \not| \neg F$, we can deduce:
  $I \not| \neg F$
  $I \models F$

The Proof Rules (II)

- According to semantics of conjunction, from $I \models F \land G$, we can deduce:
  $I \models F \land G$
  $I \models F$
  $I \models G$

- Similarly, from $I \not| F \land G$, we can deduce:
  $I \not| F \land G$
  $I \not| F$
  $I \not| G$

- The second deduction results in a branch in the proof, so each case has to be examined separately!

The Proof Rules (III)

- According to semantics of disjunction, from $I \models F \lor G$, we can deduce:
  $I \models F \lor G$
  $I \models F$
  $I \models G$

- Similarly, from $I \not| F \lor G$, we can deduce:
  $I \not| F \lor G$
  $I \not| F$
  $I \not| G$

The Proof Rules (IV)

- According to semantics of implication:
  $I \models F \rightarrow G$
  $I \not| F$ $\models G$

- And:
  $I \not| F \rightarrow G$
  $I \not| G$

The Proof Rules (V)

- According to semantics of iff:
  $I \models F \leftrightarrow G$
  $I \models F \land \neg G$

- And:
  $I \not| F \leftrightarrow G$
  $I \not| F \land \neg G$
The Proof Rules (Contradiction)

- Finally, we derive a contradiction, when $I$ both entails $F$ and does not entail $F$:

\[
\begin{align*}
I & \models F \\
I & \not\models F \\
I & \models \bot
\end{align*}
\]

Another Example

- Prove that the following formula is valid using semantic argument method:

\[
F : ((p \rightarrow q) \land (q \rightarrow r)) \rightarrow (p \rightarrow r)
\]

Implication

- Formula $F_1$ implies $F_2$ (written $F_1 \Rightarrow F_2$) iff for all interpretations $I$, $I \models F_1 \Rightarrow F_2$

\[
F_1 \Rightarrow F_2 \text{ iff } F_1 \Rightarrow F_2 \text{ is valid}
\]

An Example

- Prove $F : (p \land q) \rightarrow (p \lor \neg q)$ is valid.

Equivalence

- Formulas $F_1$ and $F_2$ are equivalent (written $F_1 \Leftrightarrow F_2$) iff for all interpretations $I$, $I \models F_1 \Leftrightarrow F_2$

\[
F_1 \Leftrightarrow F_2 \text{ iff } F_1 \Leftrightarrow F_2 \text{ is valid}
\]

Example

- Prove that $F_1 \land (\neg F_1 \lor F_2)$ implies $F_2$ using semantic argument method.

Caveat: $F_1 \Leftrightarrow F_2$ and $F_1 \Rightarrow F_2$ are not formulas (they are not part of PL syntax); they are semantic judgments!
Summary

- **Next lecture:**
  
  Normal forms and algorithms for deciding satisfiability

- **Optional reading:**
  
  Bradley & Manna textbook until Section 1.6