Decidability

and the Undecidability of Predicate Logic

Georg Hofferek
IAIK – Graz University of Technology
georg.hofferek@iaik.tugraz.at
Motivation

- Understand Limitations of Computing
  - Undecidable Problems

- Understand Limitations of Formal Systems
  - Gödel’s Incompleteness Theorem
Outline

- Decision Problems
  - Definition
  - Examples
- Decidability
- HALT Problem
- Undecidability of Predicate Logic
  - By Problem Reduction
- Gödel’s Incompleteness Theorem
Learning Targets

- Explain what a *decision problem* is
  - Give examples (decidable & undecidable ones)
- Explain *(semi-)* *decidability*
- Sketch proof of undecidability of predicate logic
  - Using reduction of HALT problem
- Explain relation between problem reduction and decidability
- Explain Gödel’s Incompleteness Theorem
  - Sketch proof
Decision Problem

- Enumerable set $\Sigma$
  - Problem Instances
- Function $f: \Sigma \rightarrow \{\top, \bot\}$
  - Correct answer

Example:

- $\Sigma = \{\text{Propositional Formulas}\}$
- $f(\sigma) = \begin{cases} \top, & \text{if } \sigma \text{ is satisfiable} \\ \bot, & \text{otherwise} \end{cases}$
Decision Problem ≠ Problem Instance

Decision Problem

- Type of Question
  - “Given formula, decide whether it is satisfiable” (SAT Problem)

Problem Instance

- One particular question
  - “Is $p \land \neg q$ satisfiable?”
Decidability

Decision Problem \((\Sigma, f)\), Algorithm \(\mathcal{A}\):

- Decidable
  - \(\mathcal{A}\) always halts
  - \(\mathcal{A}\) correctly computes \(f(\sigma)\) for all \(\sigma \in \Sigma\)

- Semi-Decidable
  - Iff \(f(\sigma) = \top\)
    - \(\mathcal{A}\) halts
    - \(\mathcal{A}\) outputs \(\top\)
  - For \(f(\sigma) = \bot\)
    - \(\mathcal{A}\) may not halt
    - If \(\mathcal{A}\) halts, it outputs \(\bot\)
Example: Decidability

- Propositional SAT Problem
  - $\Sigma = \{Propositional\ \text{Formulas}\}$
  - $f(\sigma) = \begin{cases} \top, & \text{if } \sigma \text{ is satisfiable} \\ \bot, & \text{otherwise} \end{cases}$

- Decidable
  - e.g. $\mathcal{A} = DPLL$
    - Always halts
    - Correct answer
Halting Problem

- **Does program \( \mathcal{P} \) halt?**
  - Decision problem
  - Not decidable

- **Example programs:**
  - `while(true){ };` ❌
  - `print("Hello World");` ✓
  - `while(n!=1) {
    if(n%2==0)
      n=n/2;
    else
      n=3*n+1;
  }` ?

Undecidability of HALT

- Proof by Contradiction
  - Assume $\exists A (A$ decides HALT$)$
  
  - $\Rightarrow \forall P (A_0(P)$ outputs $\top$ iff $P$ HALTs$)$
  
  - **Weird Program:**
    - ```
      weird() {
        if($A_0$(weird))
          while(true){ }
        else
          exit();
      }
    ```
    - $A_0$: wrong answer for weird()
  
  - $\Rightarrow \neg \exists A (A$ decides HALT$)$
Problem Reduction

- **Decision Problems**
  - $f: \Sigma \rightarrow \{\top, \bot\}$
  - $g: \Delta \rightarrow \{\top, \bot\}$

- $f$ reduces to $g$ \text{ “}f \propto g\text{”}\text{ “Use } g\text{ to solve } f\text{“}

- $\exists h: \Sigma \rightarrow \Delta$
  - $h$ effectively computable
  - $f(\sigma) = g(h(\sigma))$
Problem Reduction & Decidability

- \( f \propto g \), \( g \) decidable \( \Rightarrow \) \( f \) decidable
  - Reduce it to \( g \)

- \( f \propto g \), \( f \) not decidable \( \Rightarrow \) \( g \) not decidable
  - Deciding \( f \) via \( g \) not possible
  - Contradiction if \( g \) was decidable
Decidability of Predicate Logic

**Meaning:** Satisfiability (Validity) of Formulas in Predicate Logic

**Claim:**
- Halting Problem
  - (for Turing Machines)

\[ \propto \]
- (Validity of Formulas in) Predicate Logic

**Consequence:**
- HALT not decidable

\[ \Downarrow \]
- Predicate Logic not decidable
Turing Machine

- By Alan Turing (1936)

- Model of computation
  - Formal notion of Algorithm
  - Subsumes (modern) computers
  - Infinite Memory ("Tape")
  - Universal

- Church-Turing Thesis
Formal Definition of a Turing Machine

- Set of States $Q$
- Input Alphabet $\Sigma$
- Tape Alphabet $\Gamma$
  - $\cdot \in \Gamma$
  - $\Sigma \subseteq \Gamma$
- Transition Function $\delta: Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$
- Start State $q_0 \in Q$
- Accept State $q_{acc} \in Q$
- Reject State $q_{rej} \in Q$
Configuration of a Turing Machine

- State $q$
- Position of Head
- Tape Content $uv$

$u \ q \ v$
Reduction of HALT to Predicate Logic

- For Turing Machine TM

- Construct $\Phi$ such that:

$$(\text{TM halts}) \iff (\Phi \text{ valid})$$
Model $\mathcal{M}$

- $\mathcal{A} = \{\text{strings over alphabet of TM}\}$
  - empty string $\epsilon$
- Unary function $a(s) = as$
  - For each character in alphabet
- Predicate $f_q(x, y)$
  - Configuration $\text{reverse}(x) \, q \, y$ is reachable

- Start in $q_0$, with empty tape:
  - $\mathcal{M} \models f_{q_0}(\epsilon, \epsilon)$
# Common Theories & their Decidability

<table>
<thead>
<tr>
<th>Theory</th>
<th>Full</th>
<th>Quant.-free Fragment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uninterpreted Functions and Equality</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Peano Arithmetic (0, 1, +, ·, =)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Presburger Arithmetic (0, 1, +, =)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Arrays</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Semi-Decidability of Predicate Logic

- **Algorithm**
  - “Try all possible proofs”

- **If formula valid**
  - Proof found eventually

- **If formula **not** valid**
  - No proof exists
  - Algorithm never halts
Brainteaser: Labyrinth Guards

- Fork of ways
  - One to salvation
  - One to perdition

- Two guards
  - One always lies
  - One always tells the truth
  - Can’t tell them apart

- One question

What to ask?
But Beware!

Image source: http://xkcd.com/246/
The Meta Game

- **Game:**
  1. Two players A, B
  2. A starts, turns alternate
  3. Always ends (Win or Draw)

Examples: Tic-Tac-Toe, Connect-Four, (Chess)

- **Meta-Game – One Turn:**
  - Player picks any game
  - This Game is played (same player starts)
  - Winner gets one point (both if Draw)

Ends at 5 points
Gödel’s Incompleteness Theorem

“Every sufficiently powerful formal system is either incomplete or inconsistent.”

Kurt Gödel
Notions ofCompleteness

- Theory in Predicate Logic
  - System of Axioms
  - E.g. Theory of Arithmetic

- Proof System
  - E.g. Natural Deduction
Completeness of a Proof System

- All semantically **true** sentences
  - Provable
  - $\models \Phi \rightarrow \vdash \Phi$

Natural Deduction for Predicate Logic
Completeness of a Theory

- All Sentences $\Phi$
  - Provable in theory $\vdash_T \Phi$
  - or
  - Negation provable in theory $\vdash_T \neg \Phi$

\((\text{sufficiently complex})\) Theories in Predicate Logic
Sketch: Gödel’s Sentence

“This sentence is not provable in the theory.”

- If provable:
  - Contradiction:
    - Sentence says its not provable

- If negation provable:
  - Contradiction
    - Negation of sentence says that sentence is provable
    - $\rightarrow$ Both sentence and negated sentence provable
Gödel Numbering

- **Formulas → Natural Numbers**
  - Assign number to every symbol
  - String of symbols → concatenate numbers
  - Each sentence, formula, proof has a Gödel number $G$

- **Formula $Proof(x, y)$**
  - Provable iff $x$ is Gödel number of proof for $S$, and $y$ is the Gödel number of $S$

- **Formula $Provable(y) = \exists x. Proof(x, y)$**

- **Gödel’s Sentence**: $\Phi \leftrightarrow \neg Provable(G(\Phi))$
Limitations of Formal Systems

E.g., Integer Arithmetic

- Incomplete
  - Cannot capture all aspects of “reality”
    - Some “true” statements not provable
    - Some “false” statements not refutable

- Inconsistent
  - “Too many” axioms → Contradictions
Further Reading

- Douglas Hofstadter:
  
  Gödel, Escher, Bach: An Eternal Golden Braid

- [http://en.wikipedia.org/wiki/G%C3%B6del,_Escher,_Bach](http://en.wikipedia.org/wiki/G%C3%B6del,_Escher,_Bach)
Summary

- Decision Problems
  - Mapping from instance to \{true, false\}

- Decidability
  - Terminating algorithm to compute right answer
  - Some Problems are undecidable
    - HALT
    - Satisfiability/Validity of Predicate Logic Formulas

- Problem Reduction
  - Consequences for Decidability

- Gödel’s Incompleteness Theorem
  - Limitations of Formal Systems