Foundations of Computing I

Some Perspective

Computer Science and Engineering

Programming
CSE 14x

Theory

Hardware

CSE 311

About the Course

We will study the theory needed for CSE:

Logic: 
How can we describe ideas precisely?

Formal Proofs:
How can we be positive we’re correct?

Number Theory:
How do we keep data secure?

Relations/Relational Algebra:
How do we store information?

Finite State Machines:
How do we design hardware and software?

Turing Machines:
Are there problems computers can’t solve?

About the Course

It's about perspective!

• Example: Sudoku
  • Given one, solve it by hand
  • Given most, solve them with a program
  • Given any, solve it with computer science

• Tools for reasoning about difficult problems
• Tools for communicating ideas, methods, objectives...
• Tools for automating difficult problems
• Fundamental structures for computer science

This is NOT a programming course!

Administrivia

Instructor: Adam Blank

Teaching Assistants:
John Armstrong
Phillip Huang
Johan Michalove
Emerson Matson
Melissa Medsker
Logan Weber
Jeniffer Van Wagenen
Ollin Boer Bohan
Jasper Hugunin
Michael Lee
Evan McCarty
Matthew Rockett

Homework: Due WED at start of class
Write up individually

Optional) Books:
Rosen, Velleman, MIT Book
Don’t buy new copies!

Grading (roughly):
50% Homework
20% Midterm
30% Final Exam

All Course Information @ cs.uw.edu/311
**Logic: The Language of Reasoning**

Why not use English?
- Turn right here...
  Does “right” mean the direction or now?
- Buffalo buffalo Buffalo buffalo buffalo Buffalo buffalo
  This means “Bison from Buffalo, that bison from Buffalo bully, themselves bully bison from Buffalo.
- We saw her duck
  Does “duck” mean the animal or crouch down?

“Language of Reasoning” like Java or English
- Words, sentences, paragraphs, arguments...
- Today is about **words** and **sentences**

**Why Learn A New Language?**

Logic, as the “language of reasoning”, will help us...
- Be more **precise**
- Be more **concise**
- Figure out what a statement means more **quickly**

**Propositions**

A **proposition** is a statement that
- has a truth value, and
- is “well-formed”

We need a way of talking about **arbitrary** ideas...

**Propositional Variables**: \( p, q, r, s, \ldots \)

**Truth Values**:
- \( T \) for **true**
- \( F \) for **false**

**Are These Propositions?**

2 + 2 = 5
- This is a proposition. It’s okay for propositions to be false.
The home page renders correctly in IE.
- This is a proposition. It’s okay for propositions to be false.
Turn in your homework on Wednesday.
- This is a “command” which means it doesn’t have a truth value.
This statement is false.
- This statement does not have a truth value! (If it’s true, it’s false, and vice versa.)
Akjsdf!
- This is not a proposition because it’s gibberish.
Who are you?
- This is a question which means it doesn’t have a truth value.
Every positive even integer can be written as the sum of two primes.
- This is a proposition. We don’t know if it’s true or false, but we know it’s one of them!

**A Proposition**

“Roger is an orange elephant who has toenails if he has tusks, and has toenails, tusks, or both.”

We’d like to **understand** what this proposition means.

This is where logic comes in. There are pieces that appear multiple times in the phrase (e.g., “Roger has tusks”).

These are called **atomic propositions**. Let’s list them:

- \( \text{RElephant} \): “Roger is an orange elephant”
- \( \text{RTusks} \): “Roger has tusks”
- \( \text{RToenails} \): “Roger has toenails”
Putting Them Together

"Roger is an orange elephant who has toenails if he has tusks, and has toenails, tusks, or both."

RElephant: "Roger is an orange elephant"
RTusks: "Roger has tusks"
RToenails: "Roger has toenails"

Now, we put these together to make the sentence:

RElephant ∧ (RToenails → RTusks) ∧ (RToenails ∨ RTusks ∨ (RToenails ∧ RTusks))

This is the general idea, but now, let's define our formal language.

Some Truth Tables

<table>
<thead>
<tr>
<th>p</th>
<th>¬p</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ∧ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ∨ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ⊕ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Logical Connectives

Negation (not) ¬p
Conjunction (and) p ∧ q
Disjunction (or) p ∨ q
Exclusive Or p ⊕ q
Implication p → q

Implication

"If it's raining, then I have my umbrella"

It's useful to think of implications as promises. That is "Did I lie?"

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p → q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

The only lie is when:
(a) It's raining AND
(b) I don't have my umbrella

Implication is not a causal relationship!

Implication

"If it's raining, then I have my umbrella"

Are these true?

2 + 2 = 4 → earth is a planet

The fact that these are unrelated doesn't make the statement false! "2 + 2 = 4" is true; "earth is a planet" is true. T → T is true. So, the statement is true.

2 + 2 = 5 → 26 is prime

Again, these statements may or may not be related. "2 + 2 = 5" is false; so, the implication is true. (Whether 26 is prime or not is irrelevant).

Implication is not a causal relationship!