Introduction to “Introduction to Theory of Computing”

Mark Greenstreet

Lecture Outline

CpSc 421: “Introduction to the Theory of Computing”

- What’s the “Theory of Computing”
- Course Mechanics
- Machines and Languages

What's the "Theory of Computing"?

Here's the kinds of questions we consider:

1. What is a computer?
2. What problems are possible to solve with a computer?
3. What problems are impossible to solve with a computer?
4. What problems are easy/hard to solve with a computer?
5. Do do the answers to 2-4 depend on the answer to 1?

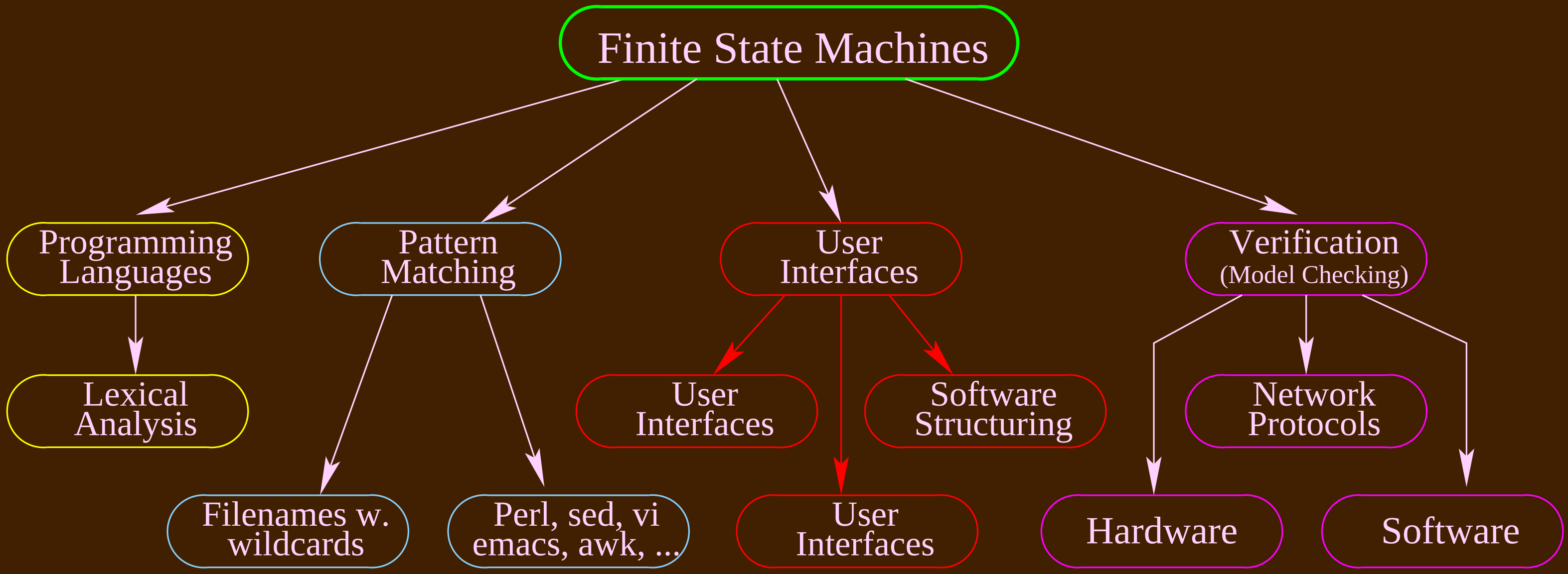
[Outline section I.A]

What is a computer?

- Finite state machines:
  A fixed amount of memory.
- Pushdown automata:
  An infinite amount of memory, arranged as a stack.
- Turing machines:
  An infinite amount of memory, arranged as a tape with a “head” that can read, write, and move left or right.
  A Turing machine is very simple but can perform any computation that a conventional comptuter can do. In fact, we don’t know of anything that can compute something that a Turing machine cannot.

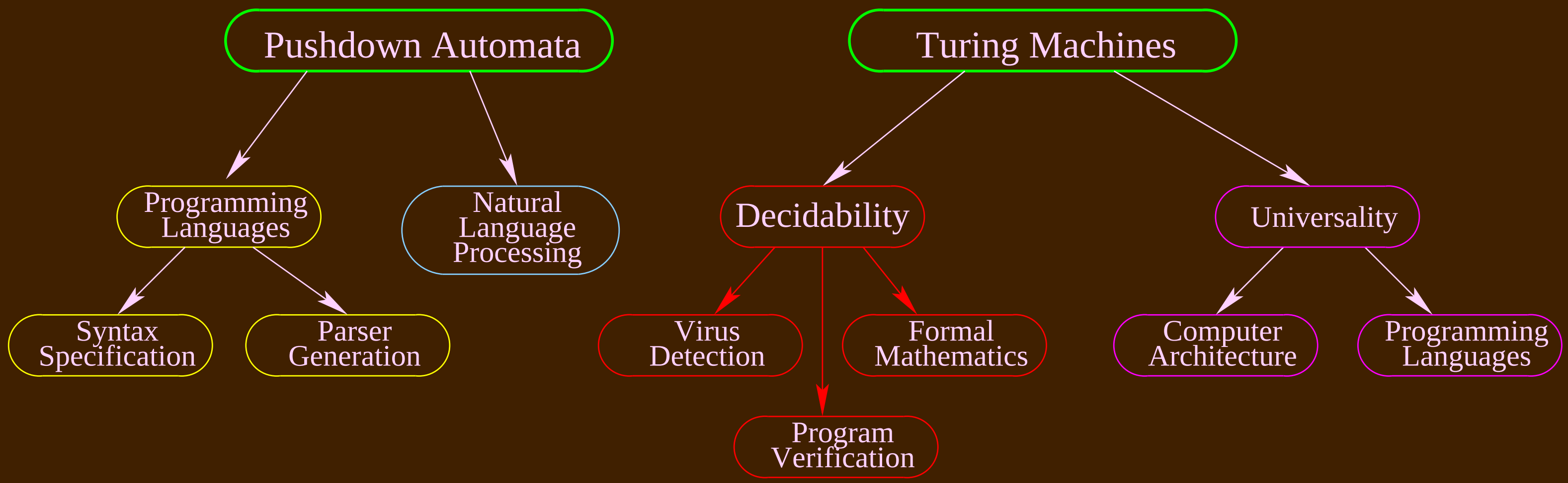
[Outline section I.B]

Connections



[Outline section I.C]

Connections



[Outline section I.C]

Course Mechanics

- Who
  - Instructor: Mark Greenstreet, mrg@cs.ubc.ca.
    Office hours: Thursdays 11am-12noon, Fridays 9-10am.
  - TAs: Mohammad Ali Safari and Jan Ulrich
    Office hours: To be announced.
- Web: http://www.ugrad.cs.ubc.ca/ cs421
- Midterms:

First Midterm	October 11, in class
Second Midterm	November 15, in class

Contact me by September 20 if you cannot attend for one or both of these dates.

[Outline section II.B.1 & II.B.2]

Grading

- Grading

Homework	25%
Midterm	30%
Final Exam	45%

- Homework:
  Roughly one assignment per week, with a week skipped for each midterm.
  Late homework penalty 10% per day, not accepted after two days late.
  See the lecture notes for more details.

[Outline section II.B.3]

Plagiarism

Submitting the work of another person, whether that be another student, something from a book, or something off the web and representing it as your own is plagiarism and constitutes academic misconduct. If the source is clearly cited, then it is not academic misconduct.

See the lecture notes for more details (section II.B.5).

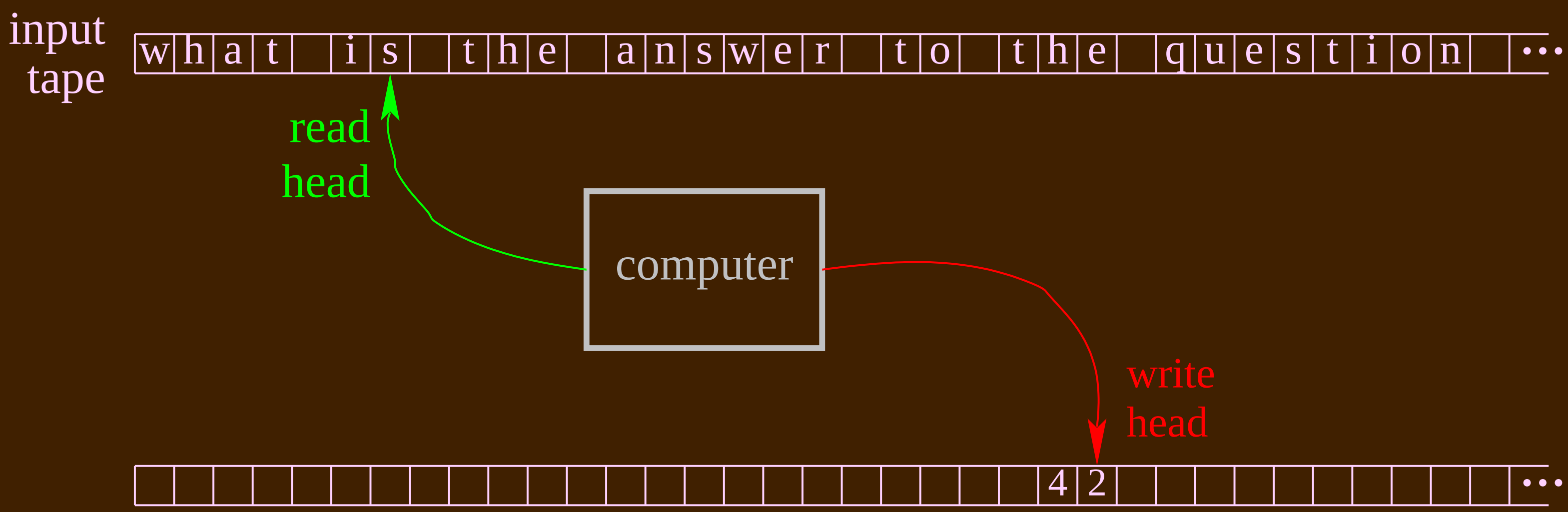
Machine Models

Today's computer:

- Input from keyboard, mouse, disk(s), network, . . .
- Output to screen, speakers, disk, networks, . . .
- To prove properties of such a machine, we would need to model the computer, all of its peripherals, the timing of events, and lots of other stuff.
- This motivates finding a simpler model.

[Outline section III.A.1]

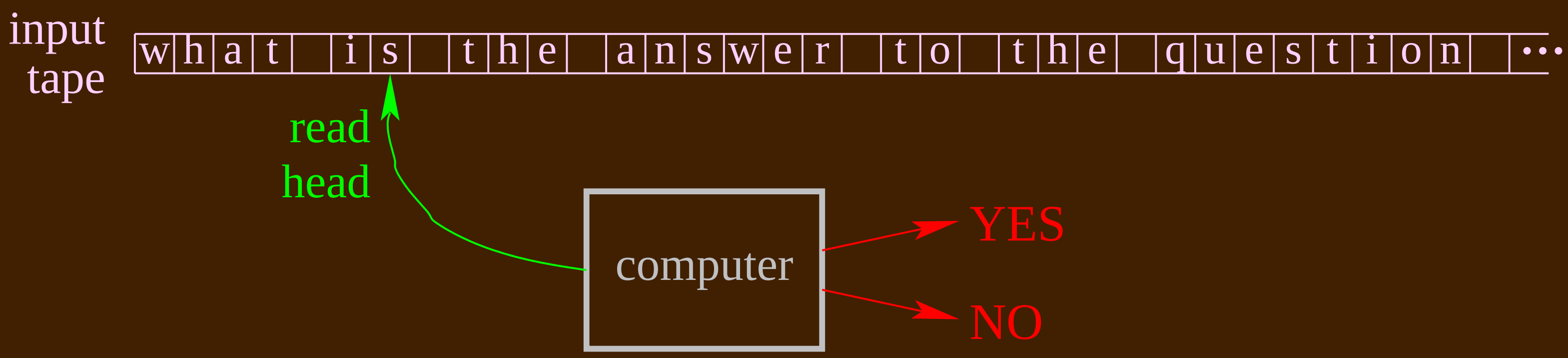
A simpler computer:



- Input from a tape.
- Output to a tape.
- Similar to a simple UNIX filter that reads from `stdin` and writes to `stdout`.
- Now, we could study the transformations that the computer can perform from its input to its output.

[Outline section III.A.2.a]

An even simpler computer:



- Input from a tape.
- Output either “YES” or “NO”.
- It’s a machine that answers “yes or no” questions.

[Outline section III.A.2.b]

Some definitions

M: The machine.

Σ: The alphabet of M, i.e. the set of symbols that can appear on squares of the input tape of M. Examples: $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots, \mathtt{z}\}$, $\{\mathtt{0}, \mathtt{1}\}$.

Strings: A string is a sequence of zero or more elements of Σ. We write ϵ to indicate the string of length 0. For example, ϵ, `abc`, `cat`, `dog`, `computer`, and `xrqmbjy` are strings of elements of the alphabet $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots, \mathtt{z}\}$.

Σ^*: The set of all strings of elements of Σ.

$L(M)$: The language of M. This is the set of all input strings for which M outputs a "YES" answer.

[Outline section III.B.1]

The English Language

Which of the following are valid English sentences:

[Outline section III.B.2.a]

The English Language

Which of the following are valid English sentences:

- vgrlum qp#d*n aoiuiui brubrubrubru 3jc6r

[Outline section III.B.2.a]

The English Language

Which of the following are valid English sentences:

- vgrlum qp#d*n aoiuiui brubrubrubru 3jc6r
- dog homework ate my. My

[Outline section III.B.2.a]

The English Language

Which of the following are valid English sentences:

- vgrlum qp#d*n aoiuiui brubrubrubru 3jc6r
- dog homework ate my. My
- Erpa shumblers groffed dulky brubrus.

[Outline section III.B.2.a]

The English Language

Which of the following are valid English sentences:

- vgrlum qp#d*n aoiuiui brubrubrubru 3jc6r
- dog homework ate my. My
- Erpa shumblers groffed dulky brubrus.
- Iron is denser than styrofoam.

[Outline section III.B.2.a]

The English Language

Which of the following are valid English sentences:

- vgrlum qp#d*n aoiuiui brubrubrubru 3jc6r
- dog homework ate my. My
- Erpa shumblers groffed dulky brubrus.
- Iron is denser than styrofoam.
- The textbook for this class has exactly ten pages.

[Outline section III.B.2.a]

The English Language

Which of the following are valid English sentences:

- vgrlum qp#d*n aoiuiui brubrubrubru 3jc6r
- dog homework ate my. My
- Erpa shumblers groffed dulky brubrus.
- Iron is denser than styrofoam.
- The textbook for this class has exactly ten pages.
- The loneliness sat of cast iron subtraction.

[Outline section III.B.2.a]

The English Language

Which of the following are valid English sentences:

- vgrlum qp#d*n aoiuiui brubrubrubru 3jc6r
- dog homework ate my. My
- Erpa shumblers groffed dulky brubrus.
- Iron is denser than styrofoam.
- The textbook for this class has exactly ten pages.
- The loneliness sat of cast iron subtraction.
- Geoge W. Bush is smarter than a dead slug.

[Outline section III.B.2.a]

The English Language

Which of the following are valid English sentences:

- vgrlum qp#d*n aoiuiui brubrubrubru 3jc6r
- dog homework ate my. My
- Erpa shumblers groffed dulky brubrus.
- Iron is denser than styrofoam.
- The textbook for this class has exactly ten pages.
- The loneliness sat of cast iron subtraction.
- Geoge W. Bush is smarter than a dead slug.

Can we make a machine that answers “YES” iff

- Each word in the sentence is a valid, English word,
- The sentence is grammatically correct,
- The sentence is true?

[Outline section III.B.2.a]

Mathematics as a Language

Which of the following are valid mathematical statements:

[Outline section III.B.2.b]

Mathematics as a Language

Which of the following are valid mathematical statements:

- $2 + 2 = 4$.

[Outline section III.B.2.b]

Mathematics as a Language

Which of the following are valid mathematical statements:

- $2 + 2 = 4.$
- $2 + 2 = 5.$

[Outline section III.B.2.b]

Mathematics as a Language

Which of the following are valid mathematical statements:

- $2 + 2 = 4$.
- $2 + 2 = 5$.
- $/7 *^4 3 - *5$.

[Outline section III.B.2.b]

Mathematics as a Language

Which of the following are valid mathematical statements:

- $2 + 2 = 4$.
- $2 + 2 = 5$.
- $/7 *^4 3 - *5$.
- $\forall n \in \mathbb{Z}.\ n^2 \geq n$.

[Outline section III.B.2.b]

Mathematics as a Language

Which of the following are valid mathematical statements:

- $2 + 2 = 4$.
- $2 + 2 = 5$.
- $/7 *^4 3 - *5$.
- $\forall n \in \mathbb{Z}.\ n^2 \geq n$.
- $\exists a, b, c, n \in \mathbb{Z}.\ (a > 0) \wedge (b > 0) \wedge (n > 2) \wedge (a^n + b^n = c^n)$.

[Outline section III.B.2.b]

Mathematics as a Language

Which of the following are valid mathematical statements:

- $2 + 2 = 4$.
- $2 + 2 = 5$.
- $/7 *^4 3 - *5$.
- $\forall n \in \mathbb{Z}.\ n^2 \geq n$.
- $\exists a, b, c, n \in \mathbb{Z}.\ (a > 0) \wedge (b > 0) \wedge (n > 2) \wedge (a^n + b^n = c^n)$.
- $\forall n \in \mathbb{Z}^{>1}.\ \exists p, q \in \mathbb{Z}.\text{prime}(p) \wedge \text{prime}(q) \wedge (p + q + 2 * n)$.

[Outline section III.B.2.b]

Mathematics as a Language

Which of the following are valid mathematical statements:

- $2 + 2 = 4$.
- $2 + 2 = 5$.
- $/7 *^4 3 - *5$.
- $\forall n \in \mathbb{Z}.\ n^2 \geq n$.
- $\exists a, b, c, n \in \mathbb{Z}.\ (a > 0) \wedge (b > 0) \wedge (n > 2) \wedge (a^n + b^n = c^n)$.
- $\forall n \in \mathbb{Z}^{>1}.\ \exists p, q \in \mathbb{Z}.\text{prime}(p) \wedge \text{prime}(q) \wedge (p + q + 2 * n)$.

Can we make a machine that answers “YES” iff

- The statement is grammatically correct,
- The statement is true?

[Outline section III.B.2.b]

Even Simpler Languages

- Let $\Sigma = \{\mathtt{0}, \mathtt{1}\}$.

[Outline section III.B.2.c]

Even Simpler Languages

- Let $\Sigma = \{0, 1\}$.
- Can we make a machine that recognizes the language:
  - The string has at least one 1.

[Outline section III.B.2.c]

Even Simpler Languages

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$.
- Can we make a machine that recognizes the language:
  - The string has at least one `1`.
  - Every `1` is followed immediately by a `0`.

[Outline section III.B.2.c]

Even Simpler Languages

- Let $\Sigma = \{\mathtt{0}, \mathtt{1}\}$.
- Can we make a machine that recognizes the language:
  - The string has at least one `1`.
  - Every `1` is followed immediately by a `0`.
  - Every `1` is followed eventually by a `0`.

[Outline section III.B.2.c]

Even Simpler Languages

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$.
- Can we make a machine that recognizes the language:
  - The string has at least one `1`.
  - Every `1` is followed immediately by a `0`.
  - Every `1` is followed eventually by a `0`.
  - The number of `0`’s in the string is equal to the number of `1`’s.

[Outline section III.B.2.c]

Even Simpler Languages

- Let $\Sigma = \{\mathtt{0}, \mathtt{1}\}$.
- Can we make a machine that recognizes the language:
  - The string has at least one `1`.
  - Every `1` is followed immediately by a `0`.
  - Every `1` is followed eventually by a `0`.
  - The number of `0`'s in the string is equal to the number of `1`'s.
  - The length of the string is a prime number.

[Outline section III.B.2.c]

Even Simpler Languages

- Let $\Sigma = \{\mathtt{0}, \mathtt{1}\}$.
- Can we make a machine that recognizes the language:
  - The string has at least one `1`.
  - Every `1` is followed immediately by a `0`.
  - Every `1` is followed eventually by a `0`.
  - The number of `0`'s in the string is equal to the number of `1`'s.
  - The length of the string is a prime number.
  - The string is the ASCII encoding for a syntactically correct Java program.

[Outline section III.B.2.c]

Even Simpler Languages

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$.
- Can we make a machine that recognizes the language:
  - The string has at least one `1`.
  - Every `1` is followed immediately by a `0`.
  - Every `1` is followed eventually by a `0`.
  - The number of `0`'s in the string is equal to the number of `1`'s.
  - The length of the string is a prime number.
  - The string is the ASCII encoding for a syntactically correct Java program.
  - The string is the ASCII encoding for a correct mathematical proof.

[Outline section III.B.2.c]

Even Simpler Languages

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$.
- Can we make a machine that recognizes the language:
  - The string has at least one `1`.
  - Every `1` is followed immediately by a `0`.
  - Every `1` is followed eventually by a `0`.
  - The number of `0`'s in the string is equal to the number of `1`'s.
  - The length of the string is a prime number.
  - The string is the ASCII encoding for a syntactically correct Java program.
  - The string is the ASCII encoding for a correct mathematical proof.
  - The string is the ASCII encoding for a provably true mathematical statement.

[Outline section III.B.2.c]

Even Simpler Languages

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$.
- Can we make a machine that recognizes the language:
  - The string has at least one `1`.
  - Every `1` is followed immediately by a `0`.
  - Every `1` is followed eventually by a `0`.
  - The number of `0`'s in the string is equal to the number of `1`'s.
  - The length of the string is a prime number.
  - The string is the ASCII encoding for a syntactically correct Java program.
  - The string is the ASCII encoding for a correct mathematical proof.
  - The string is the ASCII encoding for a provably true mathematical statement.
- What if
  - The machine has a fixed amount of memory?

[Outline section III.B.2.c]

Even Simpler Languages

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$.
- Can we make a machine that recognizes the language:
  - The string has at least one `1`.
  - Every `1` is followed immediately by a `0`.
  - Every `1` is followed eventually by a `0`.
  - The number of `0`'s in the string is equal to the number of `1`'s.
  - The length of the string is a prime number.
  - The string is the ASCII encoding for a syntactically correct Java program.
  - The string is the ASCII encoding for a correct mathematical proof.
  - The string is the ASCII encoding for a provably true mathematical statement.
- What if
  - The machine has a fixed amount of memory?
  - The memory operates as a stack?

[Outline section III.B.2.c]

Even Simpler Languages

- Let $\Sigma = \{\mathtt{0}, \mathtt{1}\}$.
- Can we make a machine that recognizes the language:
  - The string has at least one `1`.
  - Every `1` is followed immediately by a `0`.
  - Every `1` is followed eventually by a `0`.
  - The number of `0`'s in the string is equal to the number of `1`'s.
  - The length of the string is a prime number.
  - The string is the ASCII encoding for a syntactically correct Java program.
  - The string is the ASCII encoding for a correct mathematical proof.
  - The string is the ASCII encoding for a provably true mathematical statement.
- What if
  - The machine has a fixed amount of memory?
  - The memory operates as a stack?
  - Some other restriction or generalization.

[Outline section III.B.2.c]