Lecture 2: Formal Systems and Languages

CS150: Computer Science
University of Virginia
Computer Science

David Evans
http://www.cs.virginia.edu/evans

Menu

- Questions from Lecture 1 Notes
- Formal Systems
  - *MIU*-system
- Languages
  - English
  - Scheme

If it takes 60 seconds to compute a photomosaic for Problem Set 1 today on a typical PC, estimate how long it will take CS200 students in 2008 to compute the same photomosaic? How long will it take in 2011?

```
> (/ (* (- 2008 2005) 12) 18)
2
> (/ 60 (* 2 2))
15
> (/ (* (- 2011 2004) 12) 18)
4
> (/ 60 (* 2 2 2 2))
15/4
> (exact->inexact (/ 60 (* 2 2 2 2)))
3.75
```

Difference in years * 12 = number of months
Number of months / 18 = number of doublings according to Moore's Law

60 seconds today, 2 doublings by 2008
15 seconds in 2008

60 seconds today, 4 doublings by 2011
3.75 seconds in 2011

Reality check: Moore's "law" is just an "observation". We'll see one reason later today why it won't continue forever.

Are there any non-recursive natural languages? What would happen to a society that spoke one?

Not covered in Class 1...after today you should be able to answer this.

Formal Systems

Formal Systems

- Set of symbols
  - *Primitives*
- Set of rules for manipulating symbols
  - Hofstadter: Rules of Production, Rules of Inference
  - Also: Rules of Combination

The MIU System

- Symbols: M, I, U
- Rules of Production:
  - **Rule I:** If you have a string ending in I, you can add a U at the end.
  - **Rule II:** Suppose you have M*x*. Then you may add M*xx* to your collection.
  - **Rule III:** If III occurs in one of the strings in your collection you may make a new string with U in place of III.
  - **Rule IV:** If UU occurs inside one of your strings, you can drop it.

MIU System Example

Start with MUI, produce MIU

Rules of Production:
Rule I: If you have a string ending in I, you can add a U at the end.
Rule II: Suppose you have M*x*. Then you may add M*xx* to your collection.
Rule III: If III occurs in one of the strings in your collection you may make a new string with U in place of III.
Rule IV: If UU occurs inside one of your strings, you can drop it.

Survey Summary

- 26 Responses
- Years: 6 First, 1 Second, 9 Third, 8 Fourth
- Majors: 10 Cognitive Science, 4 Biology, 3 Undecided, 2 Economics, Art History, Chemistry, Electrical Engineering, Foreign Affairs, Mathematics, Neuroscience, Psychology, Sociology
- Previous computing:
  15 None, 5 A Little, 6 Lots

Survey Summary: Bodos vs. Krispy

- Food: 18 Bodos, 9 Krispy Kreme
- Full survey responses will be posted over the weekend

Languages

What is a language?

Webster:

A ~~systematic~~ means of communicating ~~ideas or feelings~~ by the use of ~~conventionalized~~ signs, sounds, gestures, or marks having ~~understood~~ meanings.

Linguist's Definition
(Charles Yang)

A description of pairs (*S*, *M*), where *S* stands for sound, or any kind of surface forms, and *M* stands for meaning.

A theory of language must specify the properties of *S* and *M*, and how they are related.

Languages and Formal Systems

What is the difference between a formal system and a language?

With a language, the surface forms have **meaning**.

Caveat: computer scientists often use *language* to mean just a set of surface forms.

What are languages made of?

- **Primitives** (almost all languages have these)
  - The simplest surface forms with **meaning**
- **Means of Combination** (**all** languages have these)
  - Like Rules of Production for Formal Systems
  - Ways to make new surface forms from ones you already have
- **Means of Abstraction** (all **powerful** languages have these)
  - Ways to use simple surface forms to represent complicated ones

What is the longest word in the English language?

According to Guinness

floccipoccinihilipilification

the act of rendering useless

Making Longer Words

antifloccipoccinihilipilification

the act of rendering not useless

antiantifloccipoccinihilipilification

the act of rendering not not useless

Language is *Recursive*

No matter what word you think is the longest word, I can always make up a longer one!

word ::= **anti-** *word*

If you have a word, you can always make up a new word by adding **anti** in front. Since the result is a word, you can make a longer new word by adding **anti-** in front again.

Recursive Definitions

- We can define things in terms of themselves
- Recursive definitions are different from circular definitions: they eventually end with something real

word ::= **anti-** *word*

word ::= **floccipoccinihilipilification**

Recursive Definitions

Allow us to express infinitely many things starting with a few.

This is powerful!

We will see **lots** of examples in this course.

Does English have these?

- Primitives
  - ~~Words (?)~~
    - ~~e.g., "antifloccipoccinihilipilification" – **not** a primitive~~
  - Morphemes – smallest units of meaning
    - e.g., **anti-** ("opposite")
- Means of combination
  - e.g., *Sentence* ::= *Subject Verb Object*
  - Precise rules, but not the ones you learned in grammar school

Ending a sentence with a preposition is something up with which we will not put.
Winston Churchill

Does English have these?

- Means of abstraction
  - Pronouns: she, he, it, they, which, etc.
  - Confusing since they don't always mean the same thing, it depends on where they are used.

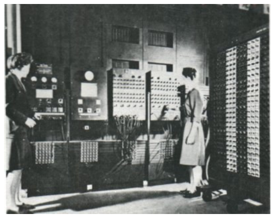
The "**these**" in the slide title is an abstraction for the three elements of language introduced 2 slides ago.

The "**they**" in the confusing sentence is an abstraction for pronouns.

How should we describe languages?

Detour:
History of Computer Programming

ENIAC: Electronic Numerical Integrator and Computer



- Early WWII computer
  - But **not** the world's first (PS4)
- Built to calculate bombing tables

Memory size:
twenty 10 decimal digit accumulators = 664 bits
ENIAC (1946): ½ mm
Apollo Guidance Computer (1969): 1 inch
You: 2.3 miles

Directions for Getting **6**

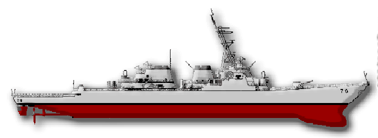
1. Choose any regular accumulator (ie. Accumulator #9).
2. Direct the Initiating Pulse to terminal *5i*.
3. The initiating pulse is produced by the initiating unit's *Io* terminal each time the Eniac is started. This terminal is usually, by default, plugged into Program Line 1-1 (described later). Simply connect a program cable from Program Line 1-1 to terminal *5i* on this Accumulator.
4. Set the Repeat Switch for Program Control 5 to 6.
5. Set the Operation Switch for Program Control 5 to .
6. Set the Clear-Correct switch to C.
7. Turn on and clear the Eniac.
8. Normally, when the Eniac is first started, a clearing process is begun. If the Eniac had been previously started, or if there are random neons illuminated in the accumulators, the ``Initial Clear'' button of the Initiating device can be pressed.
9. Press the ``Initiating Pulse Switch'' that is located on the Initiating device.
10. **Stand back.**

Admiral Grace Hopper
(1906-1992)

- Mathematics PhD Yale, 1934
- Entered Navy, 1943
- First to program Mark I (first "large" computer, 51 feet long)
- Wrote first compiler (1952) – program for programming computers
- Co-designer of COBOL (most widely used programming language until a few years ago)

"Nobody believed that I had a running compiler and nobody would touch it. They told me computers could only do arithmetic."

USS Hopper



"Dare and Do"

Guest on David Letterman

Nanostick

- How far does light travel in 1 nanosecond?

```
> (define nanosecond (/ 1 (* 1000 1000 1000))) ;; 1 billionth of a s
> (define lightspeed 299792458) ; m / s
> (* lightspeed nanosecond)
149896229/500000000
> (exact->inexact (* lightspeed nanosecond))
0.299792458
```
= just under 1 foot

Dell machines in Small Hall have "1.8-GHz Pentium 4 CPU"

GHz = GigaHertz = 1 Billion times per second
They must finish a step before light travels 6.6 inches!

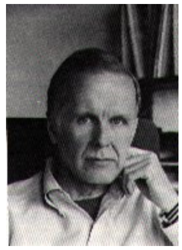
Code written by humans
⇩
Compiler
⇩
Code machine can run

Compiler translates from code in a high-level language to machine code

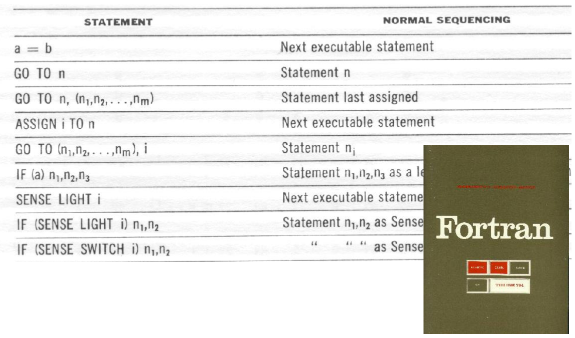
DrScheme uses an *interpreter*. An interpreter is like a compiler, except it runs quickly and quietly on small bits of code at a time.

John Backus

- Chemistry major at UVA (entered 1943)
- Flunked out after second semester
- Joined IBM as programmer in 1950
- Developed Fortran, first commercially successful programming language and compiler



IBM 704 Fortran manual, 1956



STATEMENT	NORMAL SEQUENCING
a = b	Next executable statement
GO TO n	Statement n
GO TO n, ($n_1, n_2, \ldots, n_m$)	Statement last assigned
ASSIGN i TO n	Next executable statement
GO TO ($n_1, n_2, \ldots, n_m$), i	Statement n_i
IF (a) n_1, n_2, n_3	Statement n_1, n_2, n_3 as a l…
SENSE LIGHT i	Next executable stateme…
IF (SENSE LIGHT i) n_1, n_2	Statement n_1, n_2 as Sense…
IF (SENSE SWITCH i) n_1, n_2	" " " as Sense…

Describing Languages

- Fortran language was described using English
  - Imprecise
  - Verbose, lots to read
  - Ad hoc

    DO 10 I=1.10

    Assigns `1.10` to the variable `DO10I`

    DO 10 I=1,10

    Loops for `I = 1` to `10`

  (Often incorrectly blamed for loss of Mariner-I)
- Wanted a more precise way of describing a language

Backus Naur Form

non-terminal ::= *replacement*

We can replace *non-terminal* with *replacement*

> *A* ::= *B* means anywhere you have an *A*, you can replace it with a *B*.

Some *replacements* are *terminals*: a terminal is something that never appears on the left side of a rule.

BNF Example

Sentence ::= *NP Verb*

NP ::= *Noun*

Noun ::= **Dave**

Noun ::= **Scheme**

Verb ::= **rocks**

Verb ::= **sucks**

What are the *terminals*?

Dave, Scheme, rocks, sucks

How many different things can we express with this language?

4, but only 2 are true.

BNF Example

Sentence ::= *NP Verb*

NP ::= *Noun*

NP ::= *Noun* **and** *NP*

Noun ::= **Dave**

Noun ::= **Scheme**

Verb ::= **rocks**

Verb ::= **sucks**

How many different things can we express with this language?

Infinitely many!
Recursion is powerful.

Essential Scheme

Expression ::= **(***Expression*$_1$ *Expression****)**
Expression ::= **(if** *Expression*$_1$
Expression$_2$
Expression$_3$ **)**
Expression ::= **(define** *name Expression***)**
Expression ::= *Primitive*
Primitive ::= *number*
Primitive ::= **+** | **-** | ***** | **/** | **<** | **>**| **=**
Primitive ::= ... (many other primitives)

Charge

- Lab Hours: posted on website
  - Take advantage of them!
  - If you can, follow the Wizards to lab now
- Problem Set 1: due Wednesday
- Don't floccipoccinihilipilificate