COMS22201: 2015/16

Language Engineering (Semantics)

Dr Oliver Ray
(csxor@Bristol.ac.uk)

Department of Computer Science
University of Bristol

Tuesday 26th January, 2016 (9am!)

Linguistics

- The science of language is known as linguistics
- It recognises three key aspects of any language:

Syntax	which expressions are allowed?
Semantics	what do those expressions mean?
Pragmatics	how are they useful?

Semiosis

- The study of signs more generally is known as semiosis
- It recognises three key aspects of any language:

Syntax	relation of signs to other signs
Semantics	relation of signs to their designata
Pragmatics	relation of signs to their interpreters

Natural Languages

- The study of natural languages (English) is called descriptive linguistics
- It recognises three key aspects of any language:

Syntax	e.g. someone says "Wow, that's nice!"
Semantics	e.g. to convey the idea of liking something
Pragmatics	e.g. thereby lying in order to win a favour

Artificial Languages

- The study of artificial languages (C) is called pure linguistics
- It recognises three key aspects of any language:

Syntax	e.g. someone writes "`z:=x;`"
Semantics	e.g. in order to initialise a temporary variable
Pragmatics	e.g. thereby finding a way to swap two values

Computer Programming

- This unit is mainly concerned with programming languages
- It will focus on the first two of the following key aspects of language:

Syntax	concerned with the form of expressions and whether or not the program compiles
Semantics	concerned with the meaning of expressions and what the program does when it runs
Pragmatics	concerned with issues like design patterns, program style, industry standards, code complexity, compiler options, development environment, commenting, …

Language Issues

Syntactic Complexity

Jack built the house the malt the rat the cat killed ate lay in.

Syntactic Ambiguity

Let him have it Chris!

Semantic Complexity

It depends on what the meaning of the word "is" is

Semantic Ambiguity

I haven't slept for ten days.

Semantic Undefinedness

Colorless green ideas sleep furiously.

Interaction of syntax and semantics

Time flies like an arrow; fruit flies like a banana.

Language Issues

Syntactic Complexity	
a = b < c ? * p + b * c : 1 << d ()	% spaghetti code
Syntactic Ambiguity	
`if (..) if (..) .. ; else ..`	% dangling if
Semantic Complexity	
`y = x++ + x++`	% sequence points
Semantic Ambiguity	
`(x%2==1) ? "odd" : "even"`	% undefined in C89 if x<0
Semantic Undefinedness	
`while (x/x)`	% error (x=0) or infinite loop
Interaction of syntax and semantics	
`A * B`	% lexer hack

Review: Formal Language Theory

Grammar	Language	Automaton	Productions	Recognition	Example
Phrase structure grammar	Recursively enumerable languages	Turing machine	Unrestricted *Baa → A*	Undecidable ?	…
Context sensitive grammar	Context-sensitive languages	Linear bounded	Context sensitive *At → aA*	Exponential? NP-complete	$a^n b^n a^n$
Context Free grammar	Context-free languages	Pushdown (stack)	Context free *S → gSc*	Polynomial O(n³)	$a^n b^n$
Regular Expression	Regular languages	Finite-state automaton	Regular *A → cA*	Linear O(n)	$a^n b$

Fragment of English Grammar

The following is a very simple fragment of English sentence grammar:

S	→	NP | NP VP	Sentence
NP	→	N | NP RC | NA N	Noun Phrase
VP	→	V NP	Verb Phrase
RC	→	NP V	Relative Clause
NA	→	PN	Noun Adjunct

Fragment of English Grammar

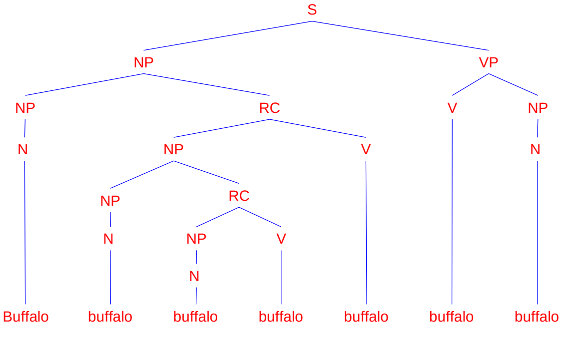
The following is a very simple fragment of English sentence grammar:

S	→	NP | NP VP	Sentence	
NP	→	N | NP RC | NA N	Noun Phrase	
VP	→	V NP	Verb Phrase	
RC	→	NP V	Relative Clause	
NA	→	PN	Noun Adjunct	
V	→	buffalo	Verb	(to bully)
N	→	buffalo	Noun	(a bison)
PN	→	Buffalo	Proper Noun	(NY state)

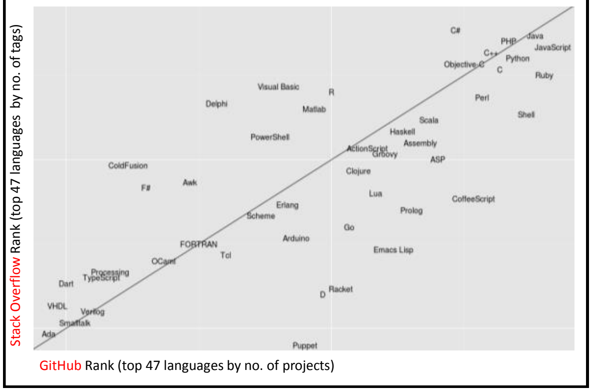
But it can be used to parse some surprisingly correct sentences

Buffalo buffalo buffalo buffalo buffalo buffalo.

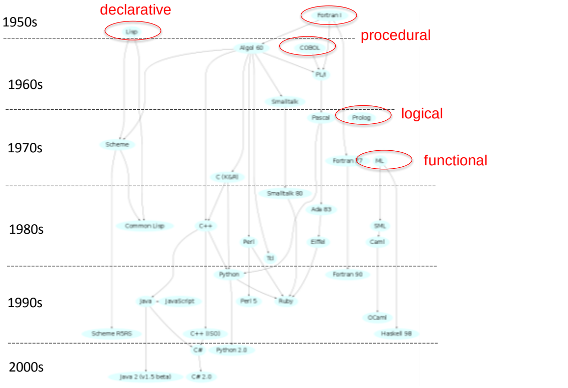
parse



Programming Languages: by Popularity



Programming Languages: by Type



Semantics

- In 1900, Breal defined semantics as the study of the way words change their meaning: e.g. the English word “nice” has evolved greatly:

kind, thoughtful	(mid 19c.)
agreeable, pleasing	(mid 18c.)
precise, careful	(late16c.)
fussy, fastidious	(mid 16c.)
coy, shy	(early 16c.)
strange, rare	(mid 15c.)
wanton, extravagant	(mid 14c.)
ignorant, foolish	(late 13c.)

- In 1939, Carnap: defined semantics as study of the relationships between expressions and their designata

What’s in a Name?

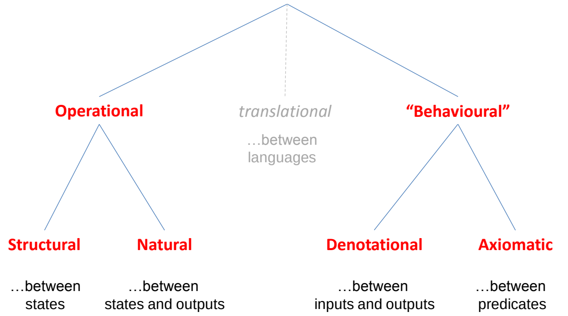
The meaning of a string can be arrived at in any number of ways – for example the string “1101” could mean any of the following:

- The number one thousand one hundred and one (if a decimal)
- The number thirteen (if binary)
- The number minus three (if in two’s complement)
- The character represented by any of the above (if ASCII)
- The number two followed by the number one (if a unary encoding)
- I have completed three of the four tasks on my todo checklist

There can always be **imoret thani**

Overview of Course

Meaning of program statements represented by mathematical relations (of increasing abstraction) …



Exercises

1. Consider the following C program, which computes y = x ! (the factorial of x):

```
int y=x; while (x --> 1) y *= x;
```

   Note this program apparently uses a special “down-to” operator `-->` that will progressively decrement the variable on its left (while returning `true`) until it falls below the value on its right (at which point it finally returns `false`).

   a) First explain how this program actually works, given that the language does not officially support such a “down-to” operator

   b) Now use a loop invariant to prove this program computes x! for all x>0

2. Consider the language L of signed decimal numerals (…, -1, 0, 1, 2, 3, …)

   a) Write an EBNF grammar for L which ensures there is exactly one numeral representing each and every integer.

   b) Convert your grammar to BNF

   c) Explain how you can represent such numerals using Haskell data types