Computational Semantics with Haskell

Yulia Zinova

Winter 2016/2017

Overview

- What is a formal study of Language?
- Lambda Calculus, Functional Programming
- Haskell: introduction
- Formal Syntax for Fragments
- Formal Semantics for Fragments
- Model Checking
- Meaning Composition
- Intension and Extension
- Parsing
- Further topics (if time allows)

We follow Van Eijck and Unger 2010, electronic access from the library

Requirements for BNs and APs

- Attendance:
  - you should attend at least 50% of theoretical classes;
  - when you attend, please participate in class actively.
- Active participation:
  - comments and questions during the class;
  - answering questions (even incorrectly);
  - brings you 10 points.

Requirements for BNs and APs

- There will be probably 5 graded assignments for 14 points each (total of 70 points);
- final project (required only for an AP) can bring you the last 20 points;
- for a BN you need 50 points (out of 80 points without the final project).

AP – Grades

- 1.0: 95 – 100
- 1.3: 91 – 94
- 1.7: 87 – 90
- 2.0: 83 – 86
- 2.3: 80 – 82
- 2.7: 75 – 79
- 3.0: 70 – 74
- 3.3: 65 – 69
- 3.7: 60 – 65
- 4.0: 50 – 59

What is *language*?

- Linguistics studies human language
- How can we define language?

What is *language*?

- Linguistics studies human language
- How can we define language?
- Chomsky (1957): human language is a set of all correct (grammatical) sentences of that language.
- Any problems with such definition?

What is *language*?

- Linguistics studies human language
- How can we define language?
- Chomsky (1957): human language is a set of all correct (grammatical) sentences of that language.
- Any problems with such definition?
- It is an abstraction that does not take into account cognitive limitations
- *Competence* – the ability of language users to recognize the members of this set at least in principle
- *Performance* – the actual abilities of language users (including memory limitations, distraction, errors)
- Examples that show the need for such distinction?

Grammars

- Users of language have limited access to it
- Rules and regularities of language are implicit for most speakers
- Want to try? You have 20 minutes to formulate rules for German plural formation (for nouns). Work in groups of 2-3 persons.

Grammars

- Users of language have limited access to it
- Rules and regularities of language are implicit for most speakers
- Want to try? You have 20 minutes to formulate rules for German plural formation (for nouns). Work in groups of 2-3 persons.
- Now check the following words: (1) die Katze, (2) der Tisch, (3) der Löffel, (4) der Student, (5) der Pullover, (6) die Tüte, (7) das Bett, (8) der Garten, (9) die Pflanze, (10) der Schrank, (11) der Nagel, (12) der Finger, (13) der Idiot, (14) das Bein, (15) der Brief, (16) das Dach, (17) der Laden, (18) die Karotte, (19) die Diskussion, (20) das Haar, (21) der Schlüssel, (22) die Straße, (23) die Stadt, (24) die Wolke, (25) der Vorhang, (26) der Turm, (27) die Uhr, (28) der Mensch
- Online form: `https://goo.gl/forms/guuwgCbGY4c1QKlh2`

Grammars

- Explicit descriptions of the rules and regularities are called *grammars*
- Given what we have discussed, answer the following question `http://directpoll.com/r?XDbzPBd3ixYqg84eAdYPzhxP7cR6jIkFo0nFd3UCc4`

Representation levels

- *Phonology*

Representation levels

- *Phonology* explores what the smallest meaning-distinguishing units (sounds) are and how they are combined into smallest meaning-carrying units (morphemes)
- *Morphology*

Representation levels

- *Phonology* explores what the smallest meaning-distinguishing units (sounds) are and how they are combined into smallest meaning-carrying units (morphemes)
- *Morphology* is concerned with how morphemes are combined int words
- *Syntax*

Representation levels

- *Phonology* explores what the smallest meaning-distinguishing units (sounds) are and how they are combined into smallest meaning-carrying units (morphemes)
- *Morphology* is concerned with how morphemes are combined int words
- *Syntax* studies how words are combined into phrases and sentences
- *Semantics*

Representation levels

- *Phonology* explores what the smallest meaning-distinguishing units (sounds) are and how they are combined into smallest meaning-carrying units (morphemes)
- *Morphology* is concerned with how morphemes are combined int words
- *Syntax* studies how words are combined into phrases and sentences
- *Semantics* investigates meaning of the basic expressions and how meaning is assigned to complex expressions based on the meaning of simpler expressions and syntactic structure

For this course

- We will not be concerned with phonology and morphology.
- Our grammars should
  - be capable of building exactly those expressions that are well-formed in the language of our choice,
  - determine the constituents of complex linguistic expressions, as well as their internal structure,
  - allow us to assign appropriate meanings to syntactically well-formed expressions on the basis of their structure.

Syntax-Semantics-Pragmatics

- **Syntax** is the study of strings and structure imposed on them by grammars generating them (form).
- **Semantics** is the study of the relation between strings and their meanings, i.e. their relation with the extralinguistic structure they are about (form + content).
- **Pragmatics** is the study of the use of meaningful strings to communicate about extralinguistic structure in an interaction process between users of natural language (form + content + use).

Formal vs. Natural languages

- Let us consider language in its written form. Then both natural and formal languages can be viewed as sets of strings. What is the difference?

Formal vs. Natural languages

- Let us consider language in its written form. Then both natural and formal languages can be viewed as sets of strings. What is the difference?
- Formal languages are given in definitions: a string belongs to a language if and only if it is produced by a grammar of that language, or recognized by a parsing algorithm for that language.
- For natural languages, a grammar can be wrong in a sense that it does not agree with speakers' intuitions.

Two kinds of meaning

- Two broad classes of explanations of meaning: *knowing how* and *knowing what*.
- Operational meaning (knowing how) can be formalized as algorithms for performing (cognitive) actions.
- Denotational meaning (knowing what) can be formalized as knowledge of the conditions for truths in the situations.
- Imagine that you ask the directions to the Opera House and hear "Turn right at the next traffic light and you will see it in front of you."
- Being able to follow the directions is operational meaning.
- Being able to distinguish situations in which those directions provide a true descriptions from other situations is denotational meaning.
- Answer the question `http://directpoll.com/r?XDbzPBd3ixYqg8xp22SjiFakG3wOETrWwVBbmQtS0`

Purposes of communication

- What can we use language for?

Purposes of communication

- What can we use language for?
- Find the truth
- Make someone believe something (not necessarily true)
- Irony
- Describe the state of affairs
- Reason about the world
- Formal languages are often designed as query tools or reasoning tools

Basic fragment

- Consider expressions such as ‘it is cold’, ‘the sun is shining’, ‘it is raining’
- To deny something, you need negation (‘it is not cold’)
- To express uncertainty, you need something like ‘either it is raining or it is cold’
- You also need conjunction and implication (‘it is cold and it is raining’ and ‘if it is raining, it is cold’)
- The study of what can be expressed by such fragment of natural language is called *propositional logic*, or *Boolean logic* (George Boole, 1815-1864)

Propositional logic

- Suppose we want to talk about two facts *a* and *b*
- There are four options: both *a* and *b* are true, both *a* and *b* are false, *a* is true and *b* is false, *a* is false and *b* is true.
- I tell you '*a* or *b*'. How many possibilities you are left with?
- If there are 10 basic facts, how many possibilities (in a situation of total ignorance) there are? `http://directpoll.com/r?XDbzPBd3ixYqg8KtMUfMkovtEnT8epNtg8mzO4O5`

Predicate logic

- Suppose you have romantic relationship. Propositional logic allows you to say 'There is love', but if you want to use pronouns, like in 'I love you', you need predicate logic.
- Predicate logic includes propositional logic and also uses relations between subjects and objects and quantification ('I love no-one but you').
- Predicate logic is more expressive than propositional logic. In fact, it is very expressive.

Typed higher-order logic and epistemic logic

- If predicate logic is not enough and you want to say more abstract things, such as ‘To love someone like you makes me very happy’, you need *typed higher-order logic*.
- Yet another level of abstraction is *epistemic logic*, or logic of knowledge. This is needed if you wish to say something like ‘I am not completely sure whether I still love you’. I am not completely sure we will have enough time to discuss it.

Relation between logical and natural languages

- It is possible to view logical languages like the language of propositional logic as fragments of natural language.
- This is done by focusing on a specific set of sentences of natural languages that can be translated into the logical language.
- This method was proposed by Richard Montague in the 1970s: "There is in my opinion no important theoretical difference between natural languages and the artificial languages of logicians; indeed, I consider it possible to comprehend the syntax and semantics of both kinds of languages within a single natural and mathematically precise theory." (Montague, 1970, p. 373)

Formal semantics

- Montague's pioneering work started the tradition of *formal semantics:* logical approaches to natural language semantics.
- Our goal: form an adequate model of parts of our language competence.
- Adequate means that the model has to be realistic in terms of complexity and learnability.

Compositionality

- Principle of compositionality as formulated by Frege (1848-1925): the meaning of a complex expression depends on the meanings of its parts and the way they are combined syntactically.
- Vague: what meanings are, what counts as a part of an expression, what does 'depend' mean?
- Only meaningful if meaning assignment is systematic. This means that if we know the meaning of *white unicorn* and *brown elk*, we also know the meaning of *brown unicorn* and *white elk*.
- Problems: pronoun resolution, presupposition. Give up or enrich?

What formal semantics is not

- Common prejudice: formal semantics is a typesetting exercise.
- *Toto barked and Dorothy smiled* = A **and** B, where A = *Toto barked* and B = *Dorothy smiled*
- *and* is **and**?

What formal semantics is not

- Common prejudice: formal semantics is a typesetting exercise.
- *Toto barked and Dorothy smiled* = A **and** B, where A = *Toto barked* and B = *Dorothy smiled*
- *and* is **and**?
- **and** refers to an operation of taking a Boolean meet of two objects in a Boolean structure
- Important: first to know about Boolean structures
- Want to play?

Computational Semantics and Functional Programming

- Functional programming naturally fits Montague's approach
- Montague grammar and functional programming are based on typed lambda calculus
- Implementation forces linguists to be precise
- Two basic usages: automatizing construction of meaning representations; operation on the results of meaning representations.
- In which applications do we need it?

Computational Semantics and Functional Programming

- Functional programming naturally fits Montague's approach
- Montague grammar and functional programming are based on typed lambda calculus
- Implementation forces linguists to be precise
- Two basic usages: automatizing construction of meaning representations; operation on the results of meaning representations.
- In which applications do we need it?
- Information retrieval: search, dialogue systems, question-answering systems, AI.

Haskell vs. Prolog

- Most current work in computational semantics uses Prolog.
- Prolog programs are meant to be implementations of logical predicates (logic programming), but some additional control operators are needed ('assert', 'retract', 'cut').
- As a result, Prolog programs are often hard to understand and debug.
- Functional programming allows for logical purity.
- No limitations in Haskell, Prolog can be embedded in Haskell (Spivey and Seres, 1999).

References:

Chomsky, N. (1957). *Syntactic Structures*. Mouton & Co, The Hague.

Montague, R. (1970). Universal grammar. *Theoria*, **36**, 373–398.

Spivey, J. M. and Seres, S. (1999). Embedding prolog in haskell. In *Proceedings of Haskell*, volume 99, pages 1999–28. Citeseer.

Van Eijck, J. and Unger, C. (2010). *Computational semantics with functional programming*. Cambridge University Press.