Sapir-Whorf and programming languages

Dr. G. Maggiore

Hogeschool Rotterdam
Rotterdam, Netherlands

Introduction

Introduction

Lecture topics

- Linguistic relativity in natural languages
- From natural to artificial (programming) languages
- Syntax, semantics, and translatability (Turing-equivalence)
- Linguistic relativity in artificial languages
- (Designing artificial languages)

Introduction

Problem discussion

- Reasoning about languages, expressive power, and potential for translation
- Using artificial, mathematical languages to quantify these concepts

Introduction

Linguistic relativity in natural languages (*Sapir–Whorf hypothesis*)

- Structure of a language affects world view and/or cognitive processes
- Strong version **language determines thought**
- Weak version **language influences thought**

Introduction

Relevance for philosophy, psychology, and linguistics

- Are human psychological faculties innate, or are they a result of learning?
- Universalism: same biological construct makes culture irrelevant
- Constructivism: human faculties and concepts come from socially constructed and learned categories

Introduction

Relevance for philosophy, psychology, and linguistics

- Relation between language and thought
- Universalism: thought as a form of innate internal speech
- Constructivism: thought is learned while acquiring language

Introduction

Relevance for philosophy, psychology, and linguistics

- Ultimately, it boils down to a deep question
- Is there an objectively known/knowable reality, shared by all humans?
- If so, is this reality objectively depicted with language constructs, and thus communicable?
- If not, how do we bridge the gap between languages?

Introduction

Examples

- In English, WATER is described with many possible words
  - water as a LIQUID
  - water in the form of a large expanse (LAKE)
  - water as running in a large body or in a small body (RIVER and BROOK)
  - water in the form of RAIN, DEW, WAVE, and FOAM
  - ...

Introduction

Examples

- In Eskimo, SNOW is described with many possible words
  - *aput*, expressing SNOW ON THE GROUND
  - *qana*, FALLING SNOW
  - *piqsirpoq*, DRIFTING SNOW
  - *qimuqsuq*, A SNOWDRIFT
  - ...

Introduction

Examples

- Guugu Yimithirr (Australian aboriginal language) only uses absolute directions when describing spatial relations
- A person is *north of the house*, not *in front of the house*
- Guugu Yimithirr speakers are better at navigating open terrain
- English speakers are better at positioning objects relative to the speaker
  - For example consider describing a round table with forks to the right of the plate and knives to the left in Guugu Yimithirr

Introduction

Central question

- **Does the choice of language make us more or less effective at experiencing, navigating, and communicating the world?**

From natural to artificial (programming) languages

From natural to artificial (programming) languages

Introduction

- Languages are not limited to natural ones
- There are many artificial languages of daily use
- Think of the one closest to you...

From natural to artificial (programming) languages

Introduction

- Languages are not limited to natural ones
- There are many artificial languages of daily use
- Think of the one closest to you... **in your pocket: your mobile phone GUI!**

From natural to artificial (programming) languages

The computer-whisperer

- Communication with machines happens through a variety of (programming) languages
- Some very limited languages are simple visual tools and GUI's
- Some very complex programming languages (PL's) have full blown syntaxes and semantics

From natural to artificial (programming) languages

The computer-whisperer

- Modern computers seem to be capable of amazing approximations of complex human thoughts
- Issue is, computers are **very fast at being very stupid**
- PL's are complex, articulated, and hard to use

From natural to artificial (programming) languages

The computer-whisperer

- Note the plural: language**S**
- Why would there be multiple languages to talk to computers, which are the same?

From natural to artificial (programming) languages

The computer-whisperer

- Note the plural: language**S**
- Why would there be multiple languages to talk to computers, which are the same? Because of linguistic relativity!
- It is empirically evident that
  - Different PL's have...
  - ...different expressive power...
  - ...in some specific domain(s)

From natural to artificial (programming) languages

The computer-whisperer

- Different PL's have different expressive power
- Assembly language
  - Expresses any possible program
  - Is very fast
  - Is very verbose
  - Accepts many nonsensical programs
  - Works in terms of machine capabilities
- Haskell
  - Expresses only non-strict programs
  - Is very slow
  - Is very compact
  - Accepts very few nonsensical programs
  - Works in terms of categorical constructions

Syntax, semantics, and Turing-completeness

Syntax, semantics, and Turing-completeness

Introduction

- Let us now take a (shallow) dive in the core of PL's
- We begin with the lambda calculus (λ-calculus)

Syntax, semantics, and Turing-completeness

Syntax of the λ-calculus

Valid programs are made up of

variables x

functions $\lambda x.t$

applications $t\ s$

Syntax, semantics, and Turing-completeness

β semantics of the λ-calculus

- Semantics of the λ-calculus are based on rewriting
- There is nothing but the language, transforming into itself
- Complex concepts are thus *unfolded* from abstracted descriptions
- **Language is thought and thought is language**

Syntax, semantics, and Turing-completeness

β semantics of the λ-calculus

- $(\lambda x.M)N \rightarrow_\beta M[x \mapsto N]$
- **Meaning as transformation**

Syntax, semantics, and Turing-completeness

Representation of Church/Peano numerals

`0`	$\lambda s.\lambda z.z$
`1`	$\lambda s.\lambda z.s\ z$
`2`	$\lambda s.\lambda z.s(s\ z)$
`N`	$\lambda s.\lambda z.s^N\ z$

Syntax, semantics, and Turing-completeness

Representation of Church/Peano numerals

PLUS $\lambda m.\lambda n.\lambda s.\lambda z.m\ s\ (n\ s\ z)$

MULT $\lambda m.\lambda n.m\ (PLUS\ n)\ 0$

Syntax, semantics, and Turing-completeness

Other languages

- The λ-calculus is not the only possible core language
- Combinatory logic is also another possibility[a]

[a] Just look at the shape of the languages, complete understanding is not needed!

Syntax, semantics, and Turing-completeness

Syntax and semantics of combinatory logic

Valid programs are made up of

- I $Ix \to x$
- K $Kxy \to x$
- S $Sxyz \to xz(yz)$
- B $Bxyz \to x(yz)$
- C $Cxyz \to xzy$

Syntax, semantics, and Turing-completeness

Combinatory logic vs λ-calculus

- Clearly two very different things
- Can one do things the other cannot?

Syntax, semantics, and Turing-completeness

Combinatory logic vs λ-calculus

- Clearly two very different things
- Can one do things the other cannot?
- **No!** They are absolutely equivalent
- We can translate back and forth without losing expressive power

Syntax, semantics, and Turing-completeness

Rule	Expression	Translation	Condition
1	x	x	(unconditional)
2	MN	M^*N^*	(unconditional)
3	$\lambda x[M]$	**K**M	x does not occur freely in M
4	$\lambda x[x]$	**I**	(unconditional)
5	$\lambda x[Mx]$	M	x does not occur freely in M
6	$\lambda x[MN]$	**B**M$(\lambda x[N])^*$	x does not occur freely in M
7	$\lambda x[MN]$	**C**$(\lambda x[M])^*$N	x does not occur freely in N
8	$\lambda x[MN]$	**S**M^*N^*	x occurs freely in both M and N

Linguistic relativity in artificial languages

Linguistic relativity in artificial languages

A partial conclusion

- Within the logical domain, there is no difference in expressive power
- When a language becomes powerful enough, then any concept can be expressed in it

Linguistic relativity in artificial languages

A partial conclusion

- A PL is **powerful enough** according to the Church-Turing hypothesis
- Translate back and forth into the λ-calculus
- No language was ever found[a] that can express programs that the λ-calculus cannot

[a] Yet.

Linguistic relativity in artificial languages

Translatability is not equivalence?

- Some programs in one language are very short in the other (reading overhead)
- Some proofs/tests of correctness in one language are very compact in the other (verification overhead)
- Some programs in one language are very fast in the other (runtime overhead)
- ...

Linguistic relativity in artificial languages

Translatability is not equivalence?

- Performance in writing, executing, and reasoning about programs make a lot of difference
- A lot of progress[a] hinges on our ability to write better software
- Bugs do sometimes cost millions[b], or even lives[c]

[a] From self-driving cars to intelligent robots
[b] Mariner I
[c] Toyota accelerator

Linguistic relativity in artificial languages

Translatability is not equivalence?

- As we put machine intelligence in charge of cranes, cars, money, and private information, the centrality of programming and PL's becomes even more prominent
- For this reason new PL's are designed almost constantly
- Some of these are domain specific, others attempt to be a *Jack of all trades*

Linguistic relativity in artificial languages

Translatability is not equivalence?

- The Sapir-Whorf hypothesis remains, even in the PL's world
- The choice of PL's changes the ability of a programmer to build things with it
- We look for the abstract patterns of the previous generation of languages and turn them into syntactic and semantic constructions

Linguistic relativity in artificial languages

Candidates for language design elements

- Time and concurrency
- Image processing/machine learning
- Totality of functions (termination)
- ...

Conclusion

Conclusion

Looking back

- Available language constructs influence thoughts
- PL's target machines instead of other humans, but have the same issues
- Modern research in PL's is looking for *Holy-Grail languages*
- With the ideal PL, it might be possible to unlock new frontiers of computation and bring on new information revolutions

This is it!

The best of luck, and thanks for the attention!