

# Test suites, [incr tsdb()]

---

Ling 567

January 12, 2016

# Overview

---

- Questions from Lab 1
- Evaluation and computational linguistics
- Evaluation and precision grammars
- Test suites and precision grammars
- Our test suites
- Features of [incr tsdb()]
- Look at Lab 2 instructions

# Ask more questions!

---

- This class is not designed so that you can complete the work on your own with the information provided.
- I'm *relying* on you to ask questions, and not spend lots of time stuck. The 10 minute rule is for real!
- For GoPost, I try to answer very quickly --- but that means I can sometimes miss things if you put two questions in the same thread...

# Questions from Lab 1: HPSG and analyses

---

- What exactly is the hook value encapsulating? I saw a note for some lex-rule the HOOK should be copied to the mother's C-CONT rather than the mother's HOOK, why this distinction?
- What are some real-world examples of LIGHT?
- How might we add the ability to parse "cats chase dogs"?

# Questions from Lab 1: Formalism

---

- Does the ordering of inherited supertypes matter when there is multiple inheritance?
- I am unsure what the hashtags used in some places mean in the context of the TDL.

```
HEAD-DTR [ SYNSEM [ LOCAL [ CONT.HOOK #hdhook ],  
  NON-HEAD-DTR.SYNSEM  
    [ LOCAL [ CAT [ VAL [ SPEC < [ LOCAL [ CONT.HOOK #hdhook ] > ] ],  
      CONT.HOOK #hook ] ],  
  C-CONT [ HOOK #hook ] ].
```

- How do we best interpret the constraints grammar rules such as HEAD-COMP without getting confused about the constraints of the lexeme structures. Is there a convenient way to refer to constraints coming from either source?

# Questions from Lab 1: Grammar Matrix

---

- Why were some of the terms different between this grammar and that developed in 566?
- Unlike the chain of identities problems from 566, this one felt a lot more convoluted for me, in that previous exercises felt much like a chain and this felt more like a web. Is that likely because I'm less familiar with the grammar, or is it really more complicated?
- How many of the rules in matrix.tdl are invoked eventually by 567\_english.tdl? All of them? Or are some only particular to other grammars?
- How big of a rules file should we expect for our grammars? Four rules seems quite small.
- How do the ARP and Valence Principles from 566 manifest in this grammar?

# Questions from Lab 1: LKB software

---

- Is it possible to input a semantic frame by hand to use the generation feature? (instead of a string as input to be parsed and then re-generated).
- Where would I find additional troubleshooting information on how to run LKB in CygWin?/Ubuntu not VirtualBox?
- I would like to know if the TDL can be generated from the LKB material. So far, we were able to do the vice versa situation.

# Questions from Lab 1: LKB Software

---

- Is there a limit to number of parses that will be tried in larger grammars? It seems that analyzing unification failures from a parse chart could become much more complex in larger grammars.
- Is there an easier way to navigate the type hierarchy than simply searching through tdl files? I tried view->type hierarchy from lkb, however the result had too many glbtypes
- Is there a way to unify the output of a rule into other rules? For example, if I need to combine two elements in my parse chart with the head-specifier rule and then make the output of that head specifier rule the 2nd thing on the ARG-ST of the head-complement rule, is it possible to unify those?



# Overview

---

- Questions from Lab 1
- Evaluation and computational linguistics
- Evaluation and precision grammars
- Test suites and precision grammars
- Our test suites
- Features of [incr tsdb()]
- Look at Lab 2 instructions

# Evaluation and Computational Linguistics

---

- Why is evaluation so prominent in computational linguistics?
- Why is it not so prominent in other subfields of linguistics?
- What about CS?

# Intrinsic v. extrinsic evaluation

---

- Intrinsic: How well does this system perform its own task, including generalizing to new data?
- Extrinsic: To what extent does this system contribute to the solution of some problem?
- Examples of intrinsic and extrinsic evaluation of parsers?

# Test data

---

- Test suites
  - Hand constructed examples
  - Positive and negative examples
  - Controlled vocabulary
  - Controlled ambiguity
  - Careful grammatical coverage

# Test data

---

- Test corpora
  - Naturally occurring
  - More open vocabulary
  - Haphazard ungrammatical examples
  - Application-focused
- Which test data for which purposes?

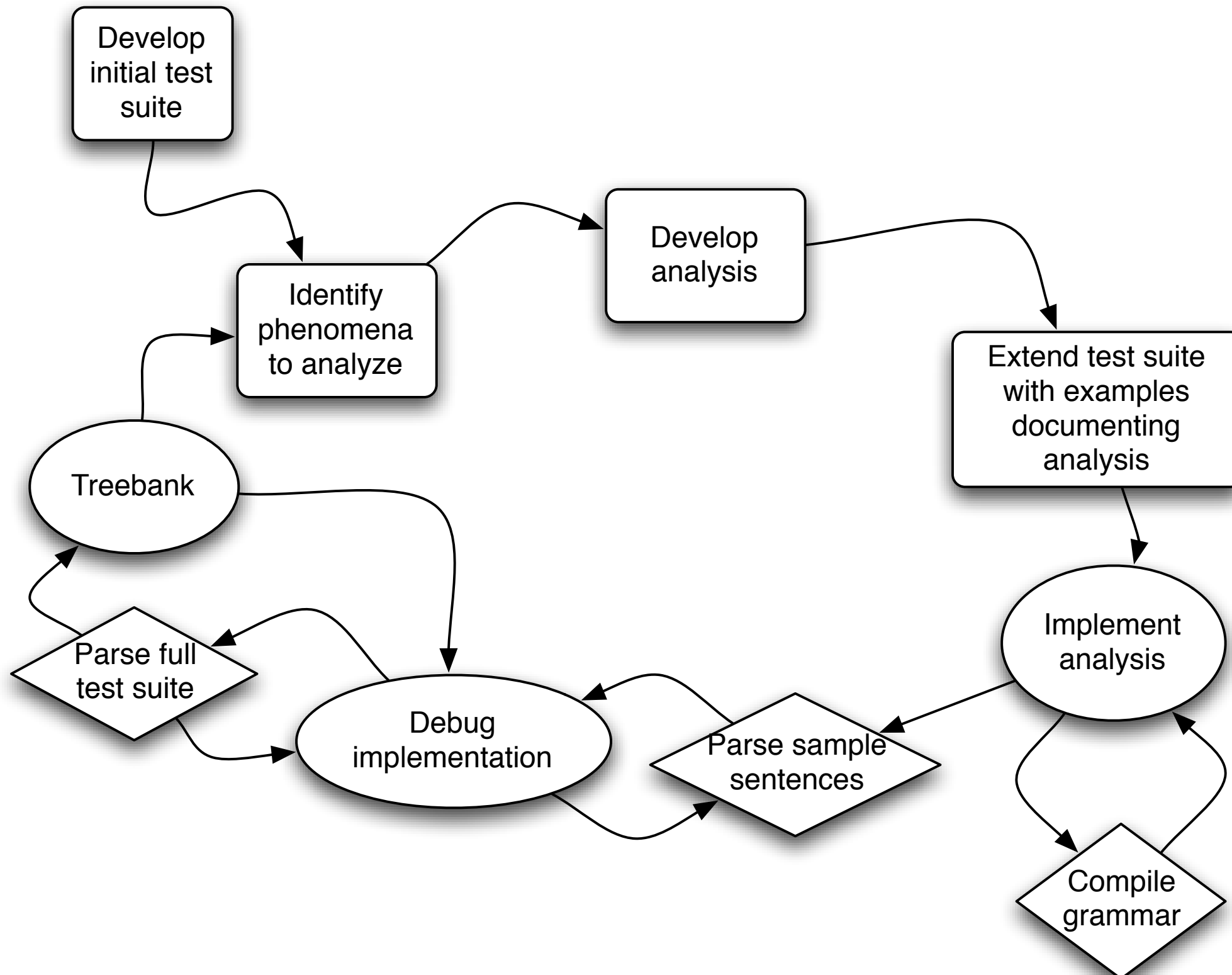
# Uses of test data

---

- How far do I have left to go?
  - Internal metric
  - Objective comparison of different systems
- Where have I been?
  - Regression testing
  - Documentation

# Grammar engineering workflow

---



# Evaluating precision grammars

- Coverage over some corpus
  - Which corpus?
  - Challenges of lexical acquisition
- Coverage of phenomena
  - How does one choose phenomena?
- Comparison across languages



# Levels of adequacy

- grammaticality
- “right” structure
- “right” dependencies
- “right” full semantics
- only legit parses (how can you tell?)
- some set of parses including the preferred one
- preferred parse only/within first N

# Our test suites

- Map out territory we hope to cover
- Include both positive and negative examples
- Serve as an exercise in understanding the description of the language
  - IGT format
  - Creating examples where necessary

# [incr tsdb()] basics

- [incr tsdb()] stores test suite profiles as (plain text) relational databases: Each is a directory with a fixed set of files in it.
- Most files are empty.
- A profile that has not been processed has only two non-empty files: item (the items to be processed) and relations (always the same)
- Once the profile has been processed, the result of the processing is stored in some of the other files (in particular, parse and result)

# [incr tsdb()] basics

- A test suite *skeleton* consists of just the item and relations files and can be used to create new test suite profiles
- [incr tsdb()] allows the user to compare two profiles to see how they differ
- It can also produce graphs plotting summary data from many profiles to visualize grammar evolution over time
- -> Demo

# Overview

---

- Questions from Lab 1
- Evaluation and computational linguistics
- Evaluation and precision grammars
- Test suites and precision grammars
- Our test suites
- Features of [incr tsdb()]
- Look at Lab 2 instructions