

Grammar Matrix; Test suites, [incr tsdb()]

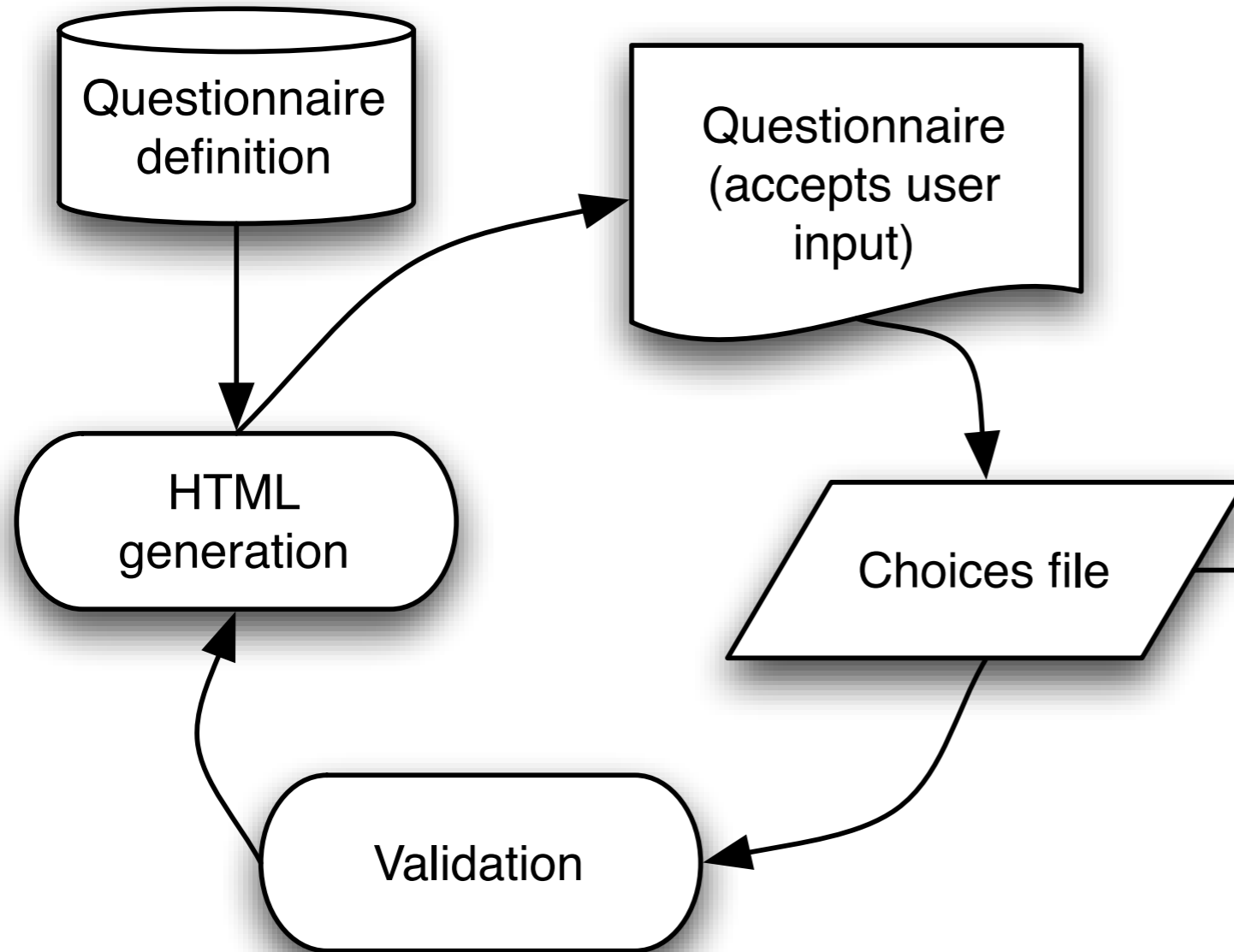
Ling 567

Jan 13, 2026

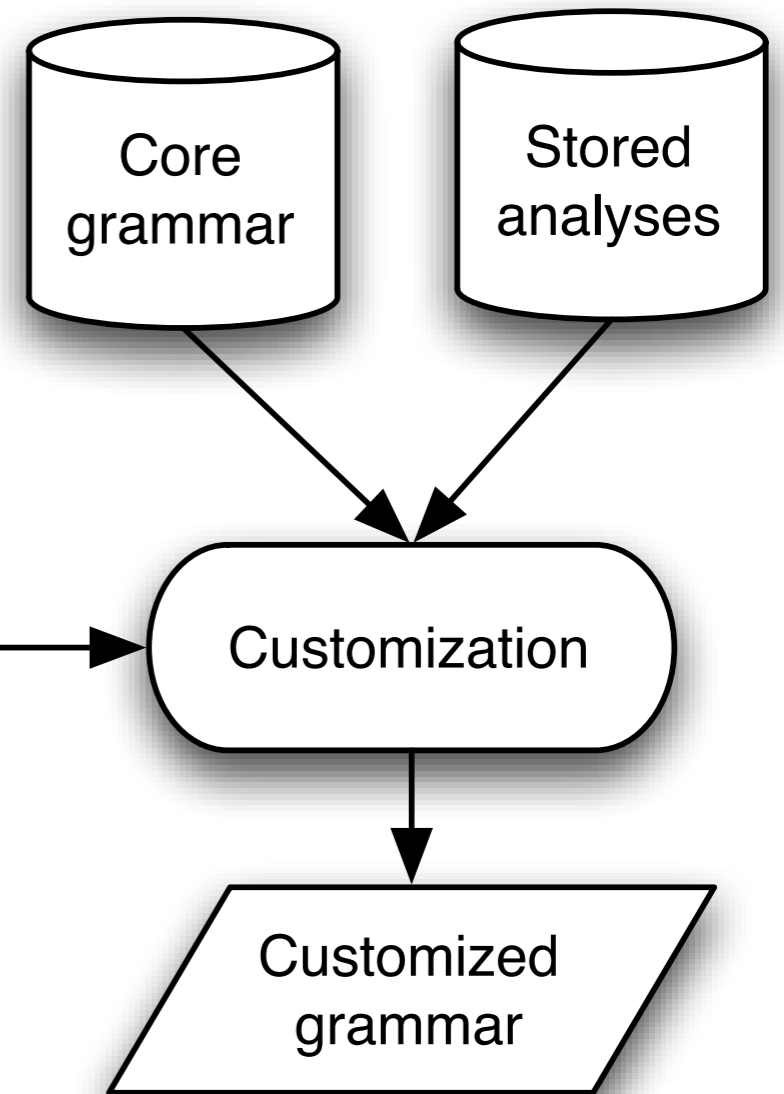
Outline

- Grammar Matrix
- Questions from Lab 1
- Testsuites
- If time: `[incr tsdb()]` demo

Elicitation of typological information



Grammar creation



Creating a library for the customization system

- Choose phenomenon
- Review typological on phenomenon
- Refine definition of phenomenon
- Conceptualize range of variation within phenomenon
- Review HPSG (& broader syntactic) literature on phenomenon
- Pin down target MRSs
- Develop HPSG analyses for each variant
- Implement analyses in tdl
- Develop questionnaire & customization logic
- Run regression tests
- Test with pseudo-languages
- Test with illustrative languages
- Test with held-out languages
- Add tests to regression tests
- Add to MatrixDoc pages

Names of things

- LinGO: a research group (part of the DELPH-IN Consortium)
- LinGO Grammar Matrix: a specific project
 - For short: The Matrix, the Grammar Matrix
- LinGO Grammar Matrix customization system: what you are interacting with through the questionnaire
 - For short: The customization system, the questionnaire

Questions from Lab 1: Chain of identities

- The chain of identities problem totally had me lost. I wasn't sure where to start. Where is the INDEX? And what does the question mean when it asks what it is identified with?
- In 566 lexical entries linked ARG values with INDEX, which then got passed around the tree by various rules. How does that basic step happen here? Lexical entries just seem to define the name of the relation.
- Is there a strategy to which of the supertypes to investigate first when looking at interactive unification? Or is it more about gaining familiarity with the matrix and rules?
- How does the INDEX value get shared from the lower N node to the upper N node, above "cat", in "The cat chased me."? Is the Semantic Inheritance Principle encoded somewhere?

Questions from Lab 1: Software

- Where is the shared folder in the UTM VM?
- Does Ikb/lui have a way to view what definition, lexical rule, etc, is causing a dependency? Is there a way to do that besides digging through definitions in the .tdl files?
- How can I quickly search through all the .tdl files to find all these super types?
- I am wondering if the numbers for the tags stay consistent within a tree or only within a node.

Questions from Lab 1: Software

- Are there ways to see the descendants of a certain type?
- Do you have tips for not getting lost in the size of the feature structures? Collapsing the items that didn't have the point of failure for interactive unification was helpful while working on questions 3-4, but I want to make sure I'm not missing anything.
- When I do unification, why am I seeing lexical rules as features on things like ARGS? Is the "rule" now the lexeme/word?

Questions from Lab 1: Features

- What are CAT & CONT? => <https://delph-in.github.io/docs/matrix/GeFaqFeatureGeometry/>
- How come some values have angle brackets around them and some don't?
- How semantics is represented seems like it received a major overhaul? What is CONT? And what is an mrs?
- For same-cont-lex-rule, why is only the HOOK value copied up to the rule's C-CONT, when there is more information present in CONT?
- What is the purpose of C-CONT? It seems to be an intermediate step of some kind.
- You addressed the fact that agreement now is primarily within the semantic indices. Why is that?

Outline

- Grammar Matrix
- Questions from Lab 1
- Testsuites
- If time: [incr tsdb()] demo
- Morphotactics in the Grammar Matrix (preview)

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

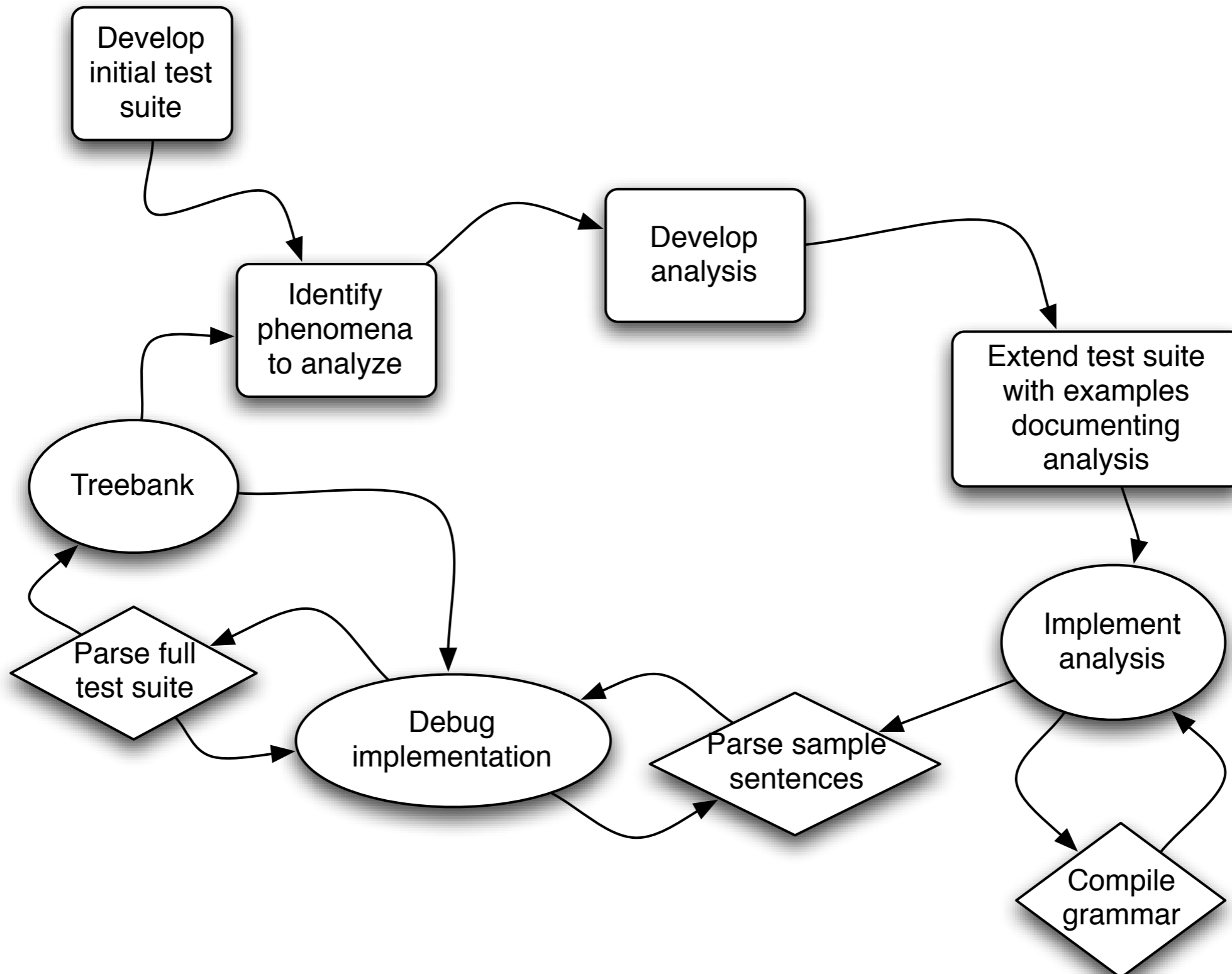
Test data

- Test corpora
 - Naturally occurring
 - More open vocabulary
 - Haphazard ungrammatical examples
 - Application-focused

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

Typical 567 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

On the importance of simple examples

- Why keep examples simple?
- How simple is too simple?
- What kinds of things make an example not simple enough?

On the importance of simple examples

- Awtuw [awt] (Feldman 1986:67)

(70) Yowmen Yawur du-k-puy-ey
Yomen Yawur DUR-IMPF-hit-IMPF
'Yowmen and Yawur are hitting (someone).' [awt]

- Basque [eus] (adapted from Joppen and Wunderlich 1995:129)

(112) Zuek lagun-ei opari polit-ak ema-ten dizkiezue.
you.PL.ERG friend-PL.DAT present nice-PL.ABS give-IMPF 3A.have.PLA.3PLD.2PLE
'You(pl) always give nice presents to your friends.' [eus]

On the importance of simple examples

- Russian [rus] (Bender 2013:92)

a. Человек укусил собаку.
Chelovek ukusi-l sobak-u.
man.NOM.SG.M bite-PAST.PFV.SG.M dog-ACC.SG.F
'The man bit the dog.' [rus]

[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
- -> If time: Demo

Outline

- Grammar Matrix
- Questions from Lab 1
- Testsuites
- If time: [incr tsdb()] demo