07 GRAMMARS

Daily Grammar Exercise (Version 1.0, Updated 2025)
Grammatical Framework Resource Grammar Library (RGL) — Example 007
Based on: Aarne Ranta, *Grammatical Framework*, Chapter 3, and the GF Resource Grammar Library API, 2nd ed.

Daily Grammar Exercise provides examples of how languages can be parsed, analyzed, and generated using formal grammars, with a focus on practical applications in computational linguistics.

Grammar Formalism refers to the mathematical system used to describe the rules and structure of a language. Different formalisms have different strengths and are suited to different tasks.

Comparing 7 Grammatical Formalisms

- [ ] Context-free grammars
- [ ] Dependency grammars
- [ ] Categorial grammars
- [ ] Tree-adjoining grammars
- [ ] Head-driven phrase structure grammars
- [ ] Lexical functional grammars
- [ ] Combinatory categorial grammars

7.1 Grammars: Clarity and Ambiguity

7.1.1 Find ambiguities in natural programs

The first step in formalizing a language is finding its ambiguities.

Here are two examples taken from ordinary English:

```
I saw the man with the telescope.

The light bulb burned when the battery ran out but otherwise complained not.
```

Can you find the ambiguities in these sentences?

Ambiguity is not limited to natural languages. Computer programs, too, can be ambiguous. The programming task is to find spaces to resolve such ambiguous expressions.

7.1.2 Ambiguity in programming languages

The grammatical form of a program is usually not very interesting. Most programming languages are designed so that they are unambiguous, and even if they are not, this is rarely a problem.

Ambiguity is, however, a problem in data formats that are designed informally. Consider, for example, the following data in the CSV (comma-separated value) format:

```
1,Paris,France
2,Rome,Italy
3,Berlin,Germany
```

Suppose we want to add the name of the country's capital. Then we have to decide whether to write

```
1,Paris,France,Paris
2,Rome,Italy,Rome
3,Berlin,Germany,Berlin
```

or

```
1,Paris,France,Paris
2,Rome,Italy,Rome
3,Berlin,Germany,Berlin
```

Both are reasonable, but they cannot be mixed in the same file. Thus we have to choose one of them and stick to it.

7.1.3 Grammatical formalisms

Finding a grammar for a human, written or spoken language is difficult. Grammars have been developed for centuries, and they are still being developed. Computer grammars are a relatively new invention, and they are usually much simpler than grammars of natural languages.

Def. Grammatical formalism A grammatical formalism is a system for describing grammars. It consists of a notation for writing grammars and a definition of the meaning of expressions written in this notation.

Def. Computational grammar A computational grammar is a grammar that is written in a grammatical formalism that is suitable for computer processing.

7.1.4 Epistemological distinctions

Def. Descriptive grammar A descriptive grammar is a grammar that describes a language as it is used by its speakers.

Def. Prescriptive grammar A prescriptive grammar is a grammar that prescribes how a language should be used.

Def. Reference grammar A reference grammar is a grammar that is intended to be used as a reference work for a language.

Def. Pedagogical grammar A pedagogical grammar is a grammar that is intended to be used for teaching a language.

7.2 Theoretical positions and goals of formalization

7.2.1 Competence vs performance

Def. Competence Competence is the ideal, error-free knowledge of a speaker or writer of a language.

Def. Performance Performance is the actual use of a language by a speaker or writer, including errors and other imperfections.

7.2.2 Descriptive, exact and pedagogical, distinct concepts

7.2.3 Descriptive hypotheses – not demonstrated

7.2.4 Grammar as a model of reality

7.2.5 Paradigm variability and error

7.2.6 Descriptive methods and prescriptive

7.2.7 Descriptive methods and prescriptive

7.2.8 Transformation and process descriptive methods

7.2.9 Hypothesis and semantic distinct concepts

7.3 Applications of formal grammars

7.3.1 The purpose of the pipeline

7.3.2 Grammar writing

7.3.3 Word-level error

7.3.4 Sentence

7.3.5 What is left out – and how to simplify

7.3.6 Semantic distinction of concepts

7.3.7 Parsing and its challenges

7.4 Parsing as an intelligence

7.4.1 Structure of the pipeline

7.4.2 Lexical analysis

7.4.3 Syntactic analysis

7.4.4 Semantic analysis

7.4.5 What is left out – and how to simplify

7.4.6 Semantic distinction of concepts

7.4.7 Parsing as an intelligence

7.5 Computing science and its challenges – Chomsky's work

7.5.1 Traditional and scientific semantics

7.5.2 Chomsky's work and its challenges

7.5.3 Reconstruction of the historical debate

7.5.4 Chomsky's work and its challenges

7.5.5 Why initial errors were overlooked

7.5.6 Chomsky's work and its challenges

7.5.7 A linguistic hot topic: clash of the titans

7.5.8 Learning how to be human

7.5.9 Standards in the future

7.6 Case studies and hypothesis disambiguation

7.6.1 Case-study writing

7.6.2 Case-study writing and hypothesis disambiguation

7.6.3 Case – how study is not simple things to

7.6.4 Case-study writing and hypothesis disambiguation

7.6.5 Case-study writing and hypothesis disambiguation

7.6.6 Case-study writing and hypothesis disambiguation

7.7 Exercises, tasks and conclusions

7.7.1 Main points in a nutshell

7.7.2 Vocabulary exercises

7.7.3 Grammar exercises

7.7.4 Grammar exercises

7.7.5 Conclusions of the study and PhD

7.7.6 Conclusions concerning semantics

7.7.7 Conclusions concerning semantics

7.8 Computational structure and execution – a programme to explain

7.8.1 Impact on science oriented semantics

7.8.2 Impact on science oriented semantics

7.8.3 Impact on science oriented semantics

7.8.4 Impact on science oriented semantics

7.8.5 Impact on science oriented semantics

7.8.6 Impact on science oriented semantics

7.9 Semantic environments and collections

7.9.1 Processing formal grammars

7.9.2 Impact of development

7.9.3 Impact of development

7.9.4 Impact of development

7.9.5 Impact of development

7.9.6 Impact of development

7.9.7 Impact of development

7.9.8 Impact of development

7.9.9 Impact of development

7.10 Collaboration, misunderstanding and programme

7.10.1 Collaborative work and its challenges

7.10.2 Collaborative work and its challenges

7.10.3 Collaborative work and its challenges

7.10.4 Collaborative work and its challenges

7.10.5 Collaborative work and its challenges

7.10.6 Collaborative work and its challenges

7.11 Closing of the book – extension site

Chapter Conclusion Chapter conclusion and summary of the main points.

Further Reading Suggestions for further reading on the topic.

Appendix A – Selected Optional Digital Sections

Appendix A.1 – Digital Sections Additional digital sections and resources.

Appendix A.2 – Digital Sections Additional digital sections and resources.

Appendix A.3 – Digital Sections Additional digital sections and resources.

Appendix B – Program Evaluations – Critical Reflective Questions

Appendix B.1 – Program Evaluations Critical reflective questions for program evaluation.

Appendix B.2 – Program Evaluations Critical reflective questions for program evaluation.

Appendix B.3 – Program Evaluations Critical reflective questions for program evaluation.