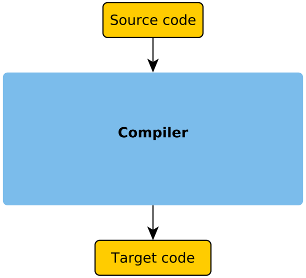
What are Formal Languages and Compilers?

Petr Zemek

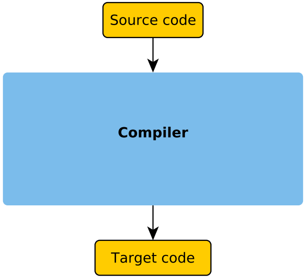
Brno University of Technology, Faculty of Information Technology
Božetěchova 2, 612 00 Brno, CZ
http://www.fit.vutbr.cz/∼izemek

Formal Languages and Compilers, 2013

What are Compilers?

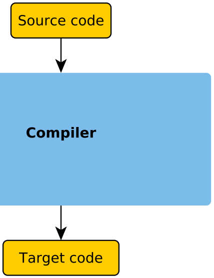


What are Compilers?



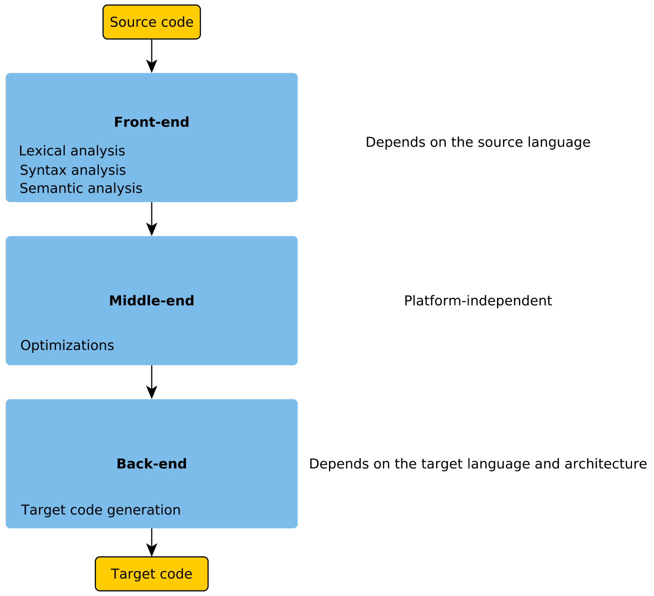
- Examples:
  - `gcc`: `.c` file → binary executable file
  - `javac`: `.java` file → `.class` file

What are Compilers?



- Examples:
  - `gcc`: `.c` file → binary executable file
  - `javac`: `.java` file → `.class` file
- typically high-level code → low-level code

What are Compilers?



What are Formal Languages?

- A language is a set of “legal” sentences.

What are Formal Languages?

- A language is a set of “legal” sentences.
- A sentence is a sequence of symbols.

What are Formal Languages?

- A language is a set of “legal” sentences.
- A sentence is a sequence of symbols.
- The symbols can be characters, words, punctuation, hieroglyphs, dots and dashes (Morse code), etc.

What are Formal Languages?

- A language is a set of “legal” sentences.
- A sentence is a sequence of symbols.
- The symbols can be characters, words, punctuation, hieroglyphs, dots and dashes (Morse code), etc.
- A formal language is a language defined by a finite set of unambiguous rules delimiting the legal sentences from the illegal ones.

What are Formal Languages?

- A language is a set of “legal” sentences.
- A sentence is a sequence of symbols.
- The symbols can be characters, words, punctuation, hieroglyphs, dots and dashes (Morse code), etc.
- A formal language is a language defined by a finite set of unambiguous rules delimiting the legal sentences from the illegal ones.

Example

Rules: $S \rightarrow aSb$, $S \rightarrow ab$
Starting symbol: S
Terminal symbols: a, b

What are Formal Languages?

- A language is a set of “legal” sentences.
- A sentence is a sequence of symbols.
- The symbols can be characters, words, punctuation, hieroglyphs, dots and dashes (Morse code), etc.
- A formal language is a language defined by a finite set of unambiguous rules delimiting the legal sentences from the illegal ones.

Example

Rules: $S \to aSb$, $S \to ab$
Starting symbol: S
Terminal symbols: a, b

The formal language: $\{a^n b^n : n \geq 1\}$

What are Formal Languages?

- A language is a set of “legal” sentences.
- A sentence is a sequence of symbols.
- The symbols can be characters, words, punctuation, hieroglyphs, dots and dashes (Morse code), etc.
- A formal language is a language defined by a finite set of unambiguous rules delimiting the legal sentences from the illegal ones.

Example

Rules: $S \rightarrow aSb$, $S \rightarrow ab$
Starting symbol: S
Terminal symbols: a, b

The formal language: $\{a^n b^n : n \geq 1\}$

- there are various models for describing formal languages

Why are Formal Languages Interesting?

Theoretical viewpoint:

- underly many areas of theoretical computer science (logic, complexity theory, automata theory, etc.)

Why are Formal Languages Interesting?

Theoretical viewpoint:

- underly many areas of theoretical computer science (logic, complexity theory, automata theory, etc.)
- provide formal models for describing formal languages

Why are Formal Languages Interesting?

Theoretical viewpoint:

- underly many areas of theoretical computer science (logic, complexity theory, automata theory, etc.)
- provide formal models for describing formal languages

Practical viewpoint: they have applications in many areas, like

- description of programming languages, compilers

Why are Formal Languages Interesting?

Theoretical viewpoint:

- underly many areas of theoretical computer science (logic, complexity theory, automata theory, etc.)
- provide formal models for describing formal languages

Practical viewpoint: they have applications in many areas, like

- description of programming languages, compilers
- computer-aided art (turtle graphics, fractals)

Why are Formal Languages Interesting?

Theoretical viewpoint:

- underly many areas of theoretical computer science (logic, complexity theory, automata theory, etc.)
- provide formal models for describing formal languages

Practical viewpoint: they have applications in many areas, like

- description of programming languages, compilers
- computer-aided art (turtle graphics, fractals)
- modeling and simulation of biological organisms (plant development)

Why are Formal Languages Interesting?

Theoretical viewpoint:

- underly many areas of theoretical computer science (logic, complexity theory, automata theory, etc.)
- provide formal models for describing formal languages

Practical viewpoint: they have applications in many areas, like

- description of programming languages, compilers
- computer-aided art (turtle graphics, fractals)
- modeling and simulation of biological organisms (plant development)
- molecular genetics

Why are Formal Languages Interesting?

Theoretical viewpoint:

- underly many areas of theoretical computer science (logic, complexity theory, automata theory, etc.)
- provide formal models for describing formal languages

Practical viewpoint: they have applications in many areas, like

- description of programming languages, compilers
- computer-aided art (turtle graphics, fractals)
- modeling and simulation of biological organisms (plant development)
- molecular genetics
- coding theory and cryptography

Why are Formal Languages Interesting?

Theoretical viewpoint:

- underly many areas of theoretical computer science (logic, complexity theory, automata theory, etc.)
- provide formal models for describing formal languages

Practical viewpoint: they have applications in many areas, like

- description of programming languages, compilers
- computer-aided art (turtle graphics, fractals)
- modeling and simulation of biological organisms (plant development)
- molecular genetics
- coding theory and cryptography
- natural language processing

Why are Formal Languages Interesting?

Theoretical viewpoint:

- underly many areas of theoretical computer science (logic, complexity theory, automata theory, etc.)
- provide formal models for describing formal languages

Practical viewpoint: they have applications in many areas, like

- description of programming languages, compilers
- computer-aided art (turtle graphics, fractals)
- modeling and simulation of biological organisms (plant development)
- molecular genetics
- coding theory and cryptography
- natural language processing
- design of embedded systems

Why are Formal Languages Interesting?

Theoretical viewpoint:

- underly many areas of theoretical computer science (logic, complexity theory, automata theory, etc.)
- provide formal models for describing formal languages

Practical viewpoint: they have applications in many areas, like

- description of programming languages, compilers
- computer-aided art (turtle graphics, fractals)
- modeling and simulation of biological organisms (plant development)
- molecular genetics
- coding theory and cryptography
- natural language processing
- design of embedded systems
- ... and many other

Formal Languages and Compilers

How exactly do formal languages and compilers relate to each other?

How exactly do formal languages and compilers relate to each other?

To build a compiler, we

1. specify our programming language by using formal models

Formal Languages and Compilers

How exactly do formal languages and compilers
relate to each other?

To build a compiler, we

1. specify our programming language by using formal models
2. turn these models into an implementation of the compiler

Purposes of This Course

- getting five credits :) oh yeah!

Purposes of This Course

- getting five credits :) oh yeah!
- introduce you to formal languages

Purposes of This Course

- getting five credits :) oh yeah!
- introduce you to formal languages
- introduce you to compiler construction

Purposes of This Course

- getting five credits :) oh yeah!
- introduce you to formal languages
- introduce you to compiler construction
- see applications of mathematics in computer science

Purposes of This Course

- getting five credits :) oh yeah!
- introduce you to formal languages
- introduce you to compiler construction
- see applications of mathematics in computer science
- improve your English skills