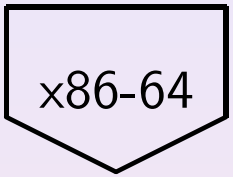
B5: Programming Language Processing

CS1101S: Programming Methodology

Martin Henz

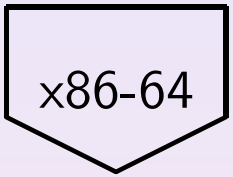
September 10, 2021

T-Diagrams



x86-64 Processor

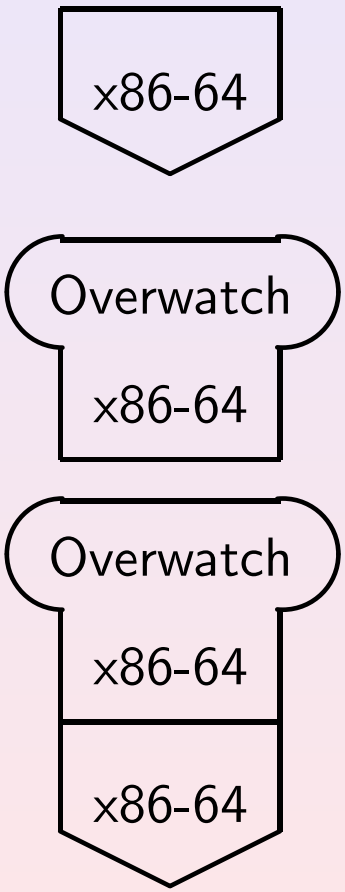
T-Diagrams



x86-64 Processor

Program "Overwatch" (x86-64 code)

T-Diagrams



x86-64 Processor

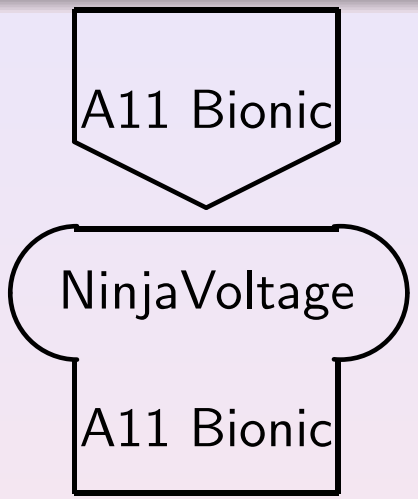
Program "Overwatch" (x86-64 code)

"Overwatch"
running on x86-64

Running an app on iPhone X

A11 Bionic, the processor of iPhone X

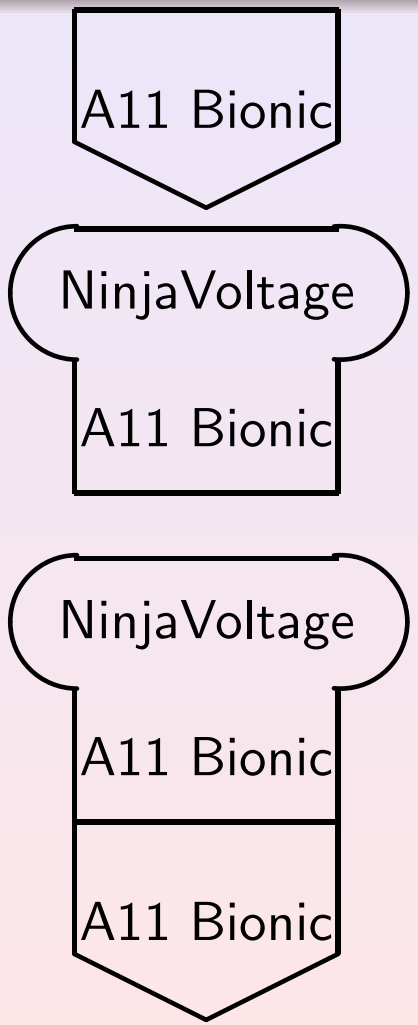
Running an app on iPhone X



A11 Bionic, the processor of iPhone X

Program “NinjaVoltage” (A11 build)

Running an app on iPhone X



A11 Bionic, the processor of iPhone X

Program "NinjaVoltage" (A11 build)

"NinjaVoltage"
running on
A11 Bionic

Interpreter

- Interpreter is program that executes another program

Interpreter

- Interpreter is program that executes another program
- The interpreter's *source language* is the language in which the interpreter is written

Interpreter

- Interpreter is program that executes another program
- The interpreter's *source language* is the language in which the interpreter is written
- The interpreter's *target language* is the language in which the programs are written which the interpreter can execute

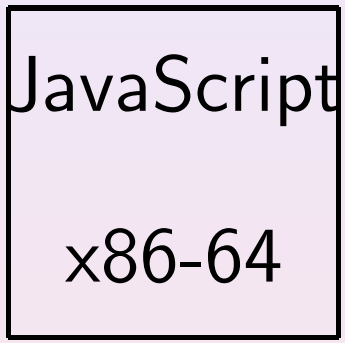
Interpreter

- Interpreter is program that executes another program
- The interpreter's *source language* is the language in which the interpreter is written
- The interpreter's *target language* is the language in which the programs are written which the interpreter can execute

Teaser for Lecture 11

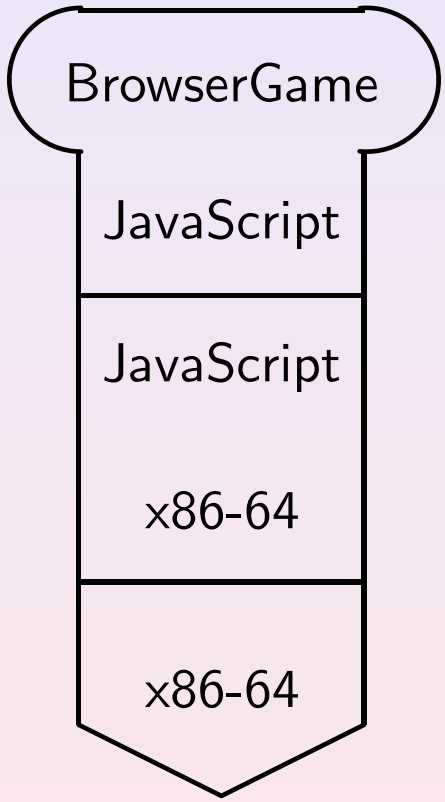
The evaluator (interpreter),
which determines the meaning
of statements in a language,
is just another program.
("Most fundamental idea
in programming")

Interpreters



Chrome browser for PC,
seen as interpreter for JavaScript,
written in x86-64 machine code

"Normal" Way of Running JavaScript on Chrome



The browser acts as
an interpreter for JavaScript.

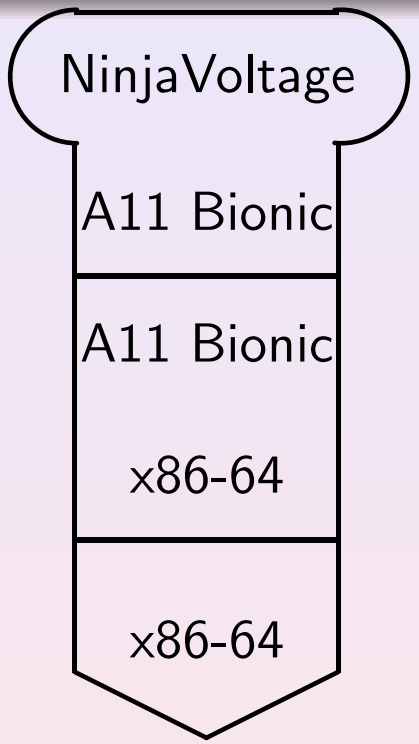
Another example: Elixir

Interpreter for Elixir,
written in x86-64 machine code

Running Elixir on Server

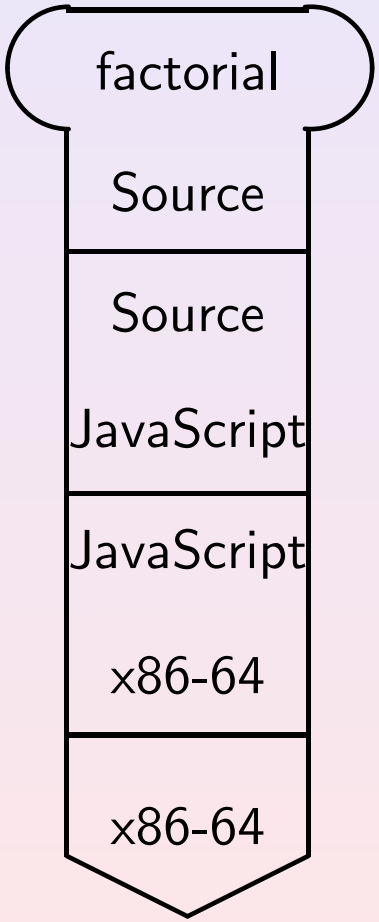
Elixir program "assessment"
running on x86-64
using interpretation

Hardware Emulation



"NinjaVoltage" app
running on a PC
using hardware emulator

Running Source §2 in Source Academy using Stepper



Source Academy stepper:

layer between your programs
and Chrome's native JavaScript

Compilers

Definition

A compiler is a program that
translates from one language
(the *from-language*)
to another language
(the *to-language*).

Compilers

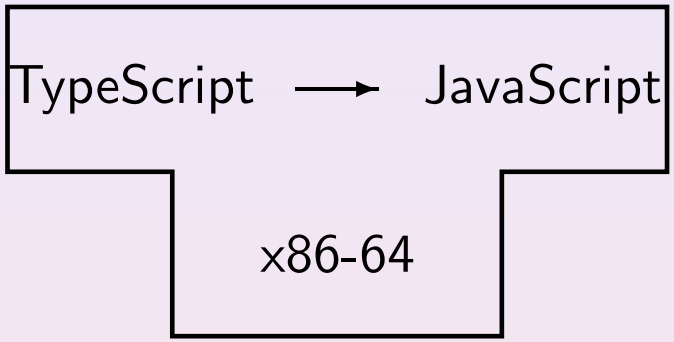
Definition

A compiler is a program that
translates from one language
(the *from-language*)
to another language
(the *to-language*).

Teaser for Lecture L12C

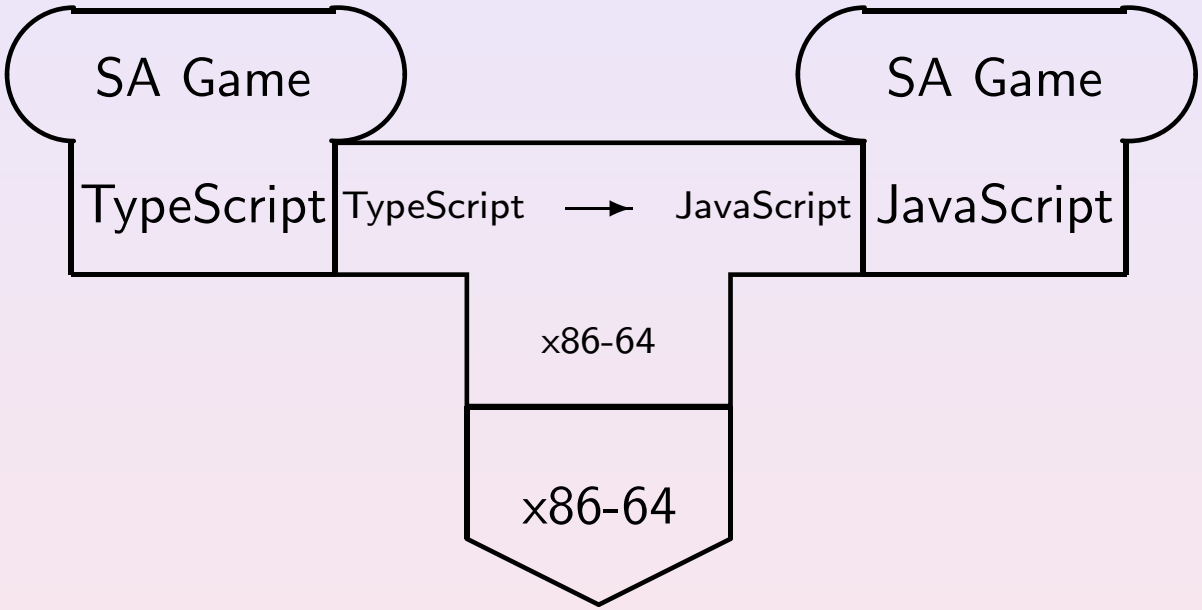
A compiler, which translates programs
from one language to another,
is just another program.
("Second most fundamental
idea in programming")

T-Diagram of Compiler



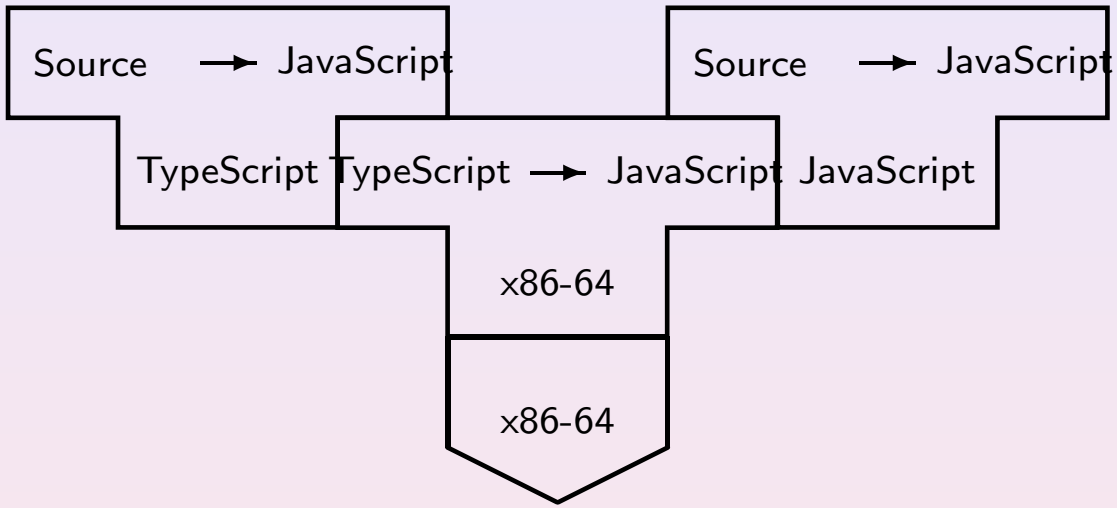
TypeScript-to-JavaScript compiler written in x86-64 machine code

Compiling a program (SA Game of Source Academy)



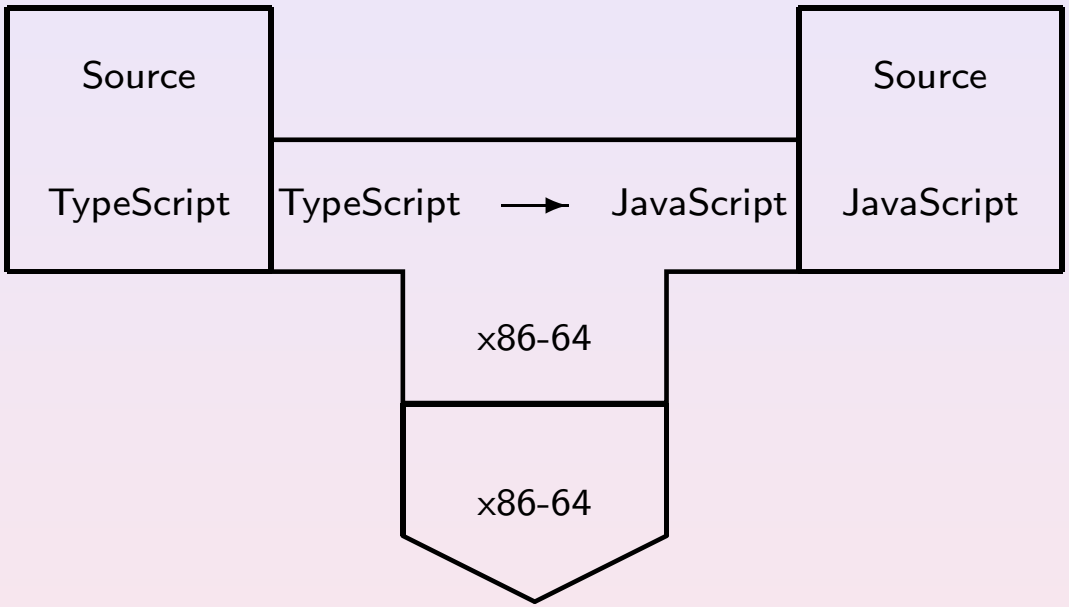
Compiling "SA Game"
from TypeScript to JavaScript

Compiling our Source Compiler



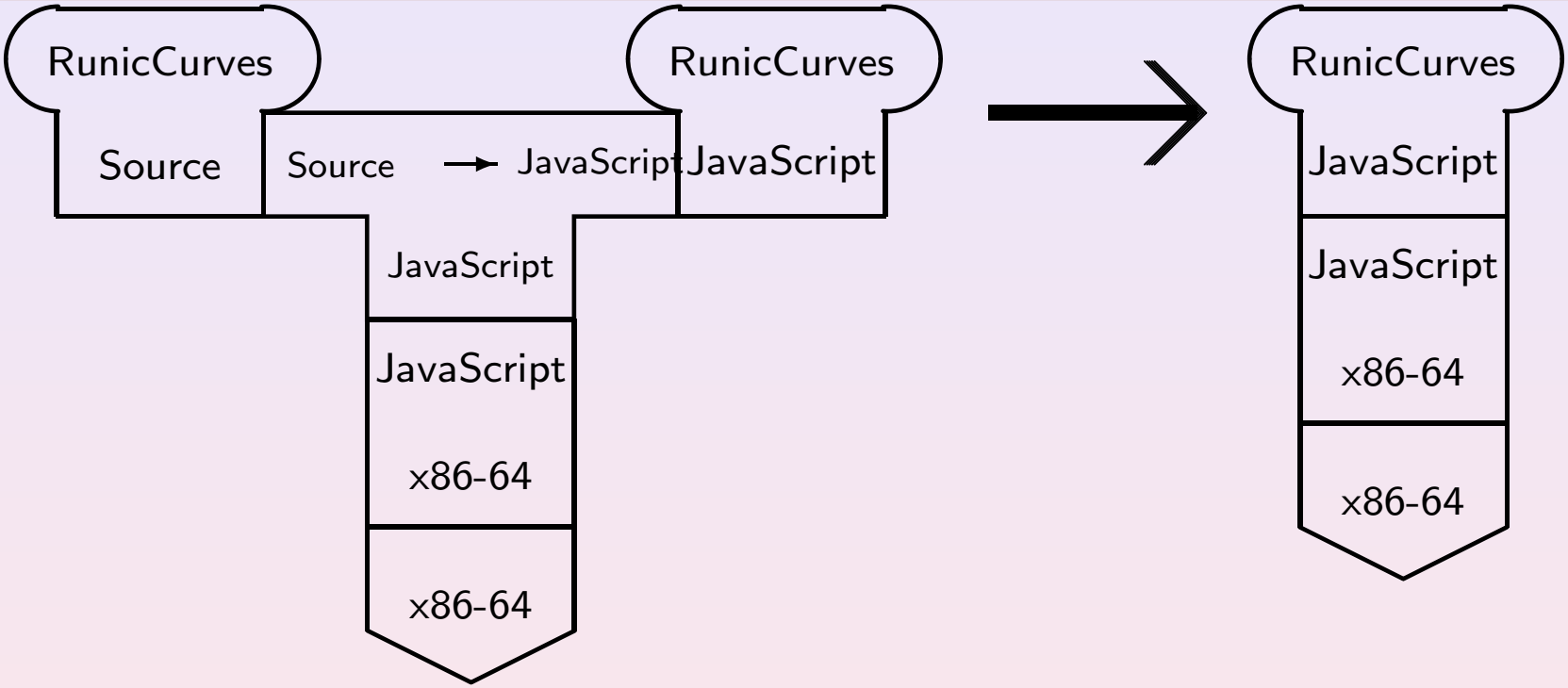
Compiling
Source-to-JavaScript compiler
from TypeScript to JavaScript

Compiling the Stepper



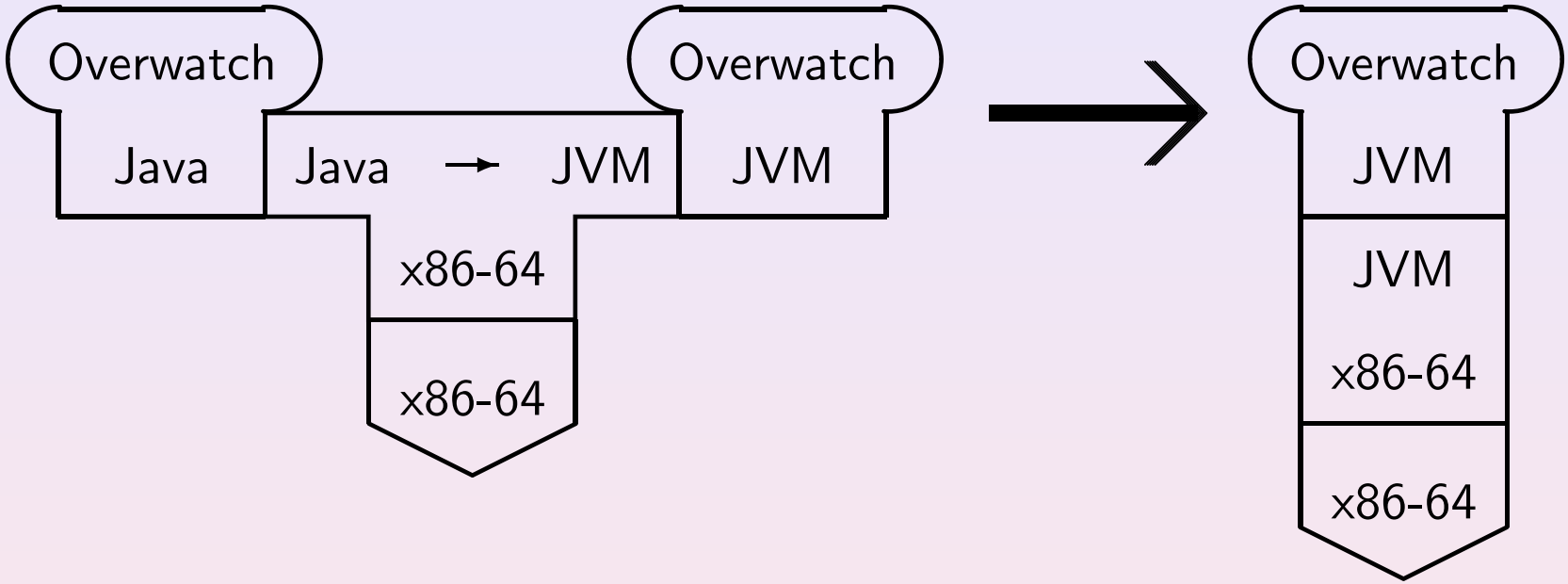
Compiling stepper tool
from TypeScript to JavaScript

A typical Source Academy session



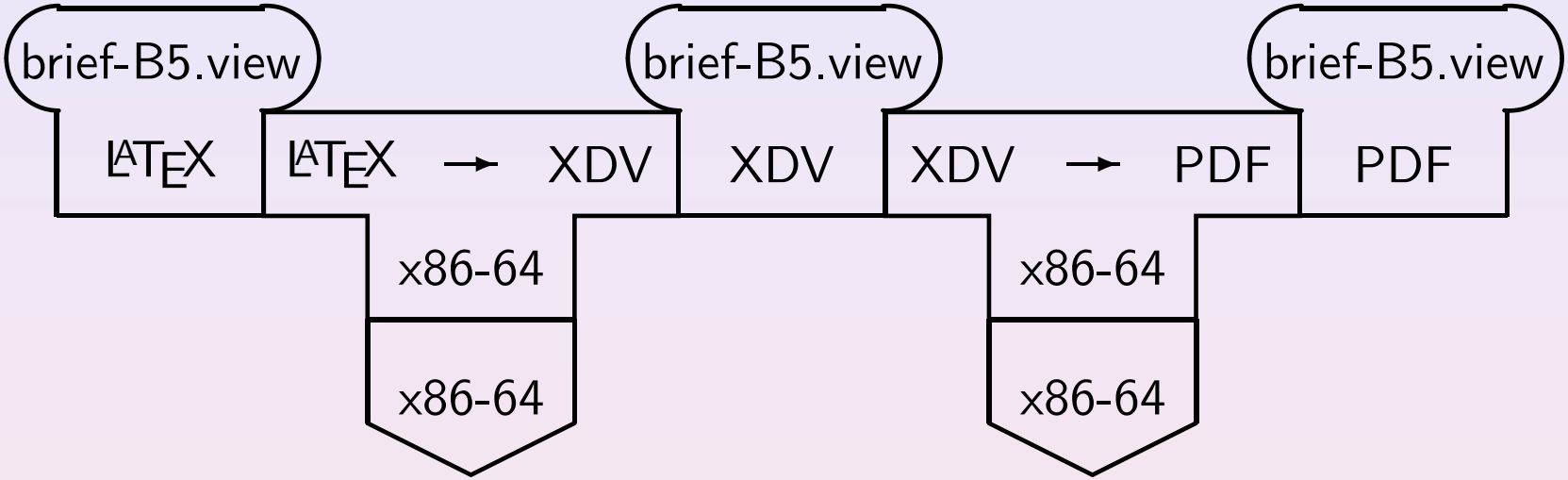
Compiling "RunicCurves" from Source to JavaScript in browser, and then running JavaScript program in browser.

Typical Execution of Java Programs



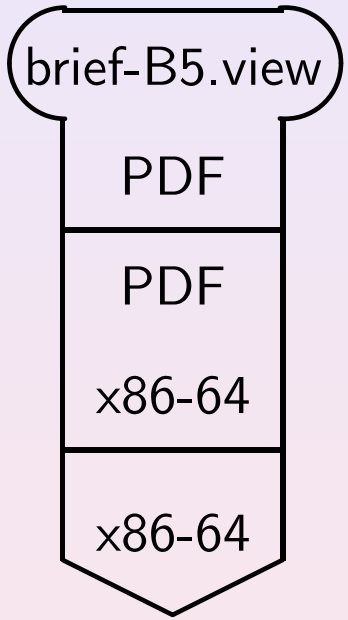
Compiling "Overwatch" from Java to JVM code, and running the JVM code on a JVM running on an x86-64

Excursion: Making these Slides



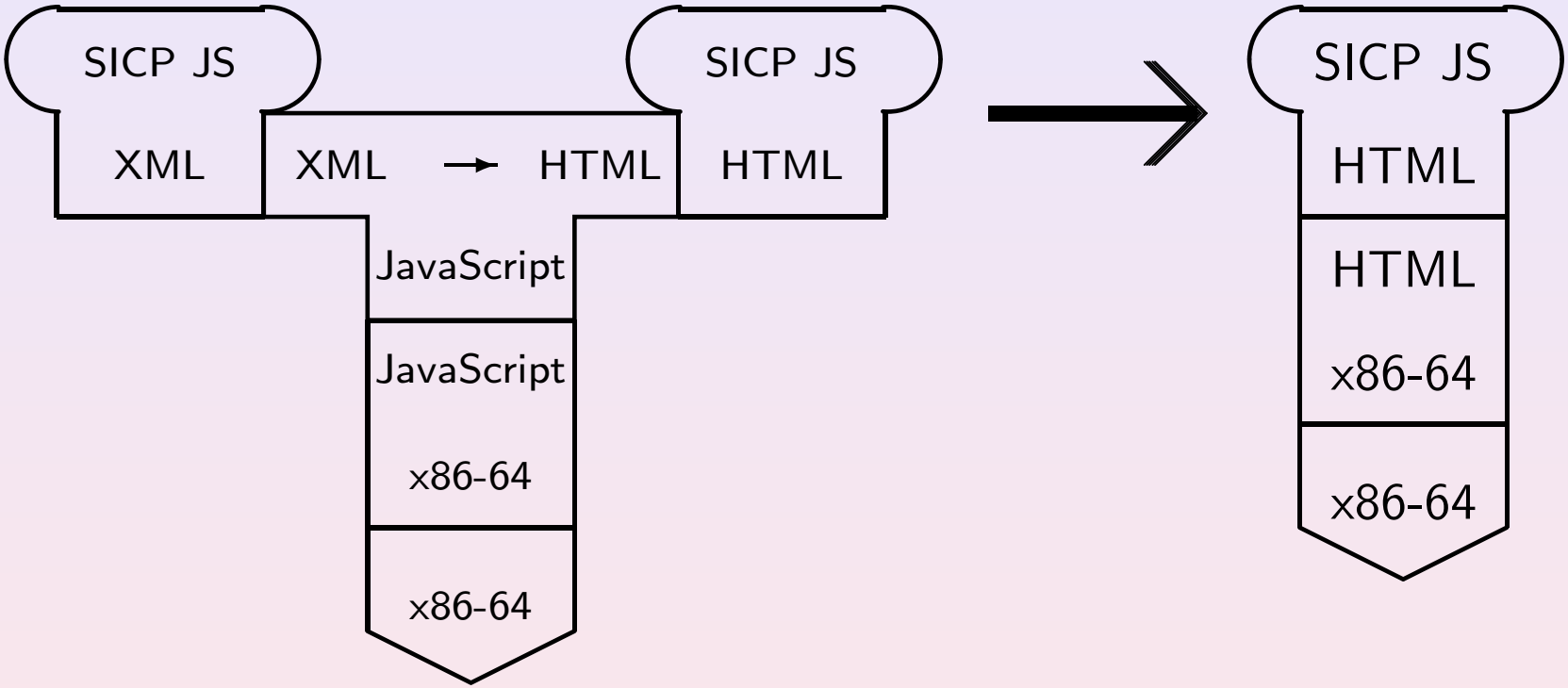
Compiling these slides
using the XeTeX *tool chain*
from LaTeX to XDV
to PDF on x86-64

Excursion: Viewing these Slides with Acrobat Reader



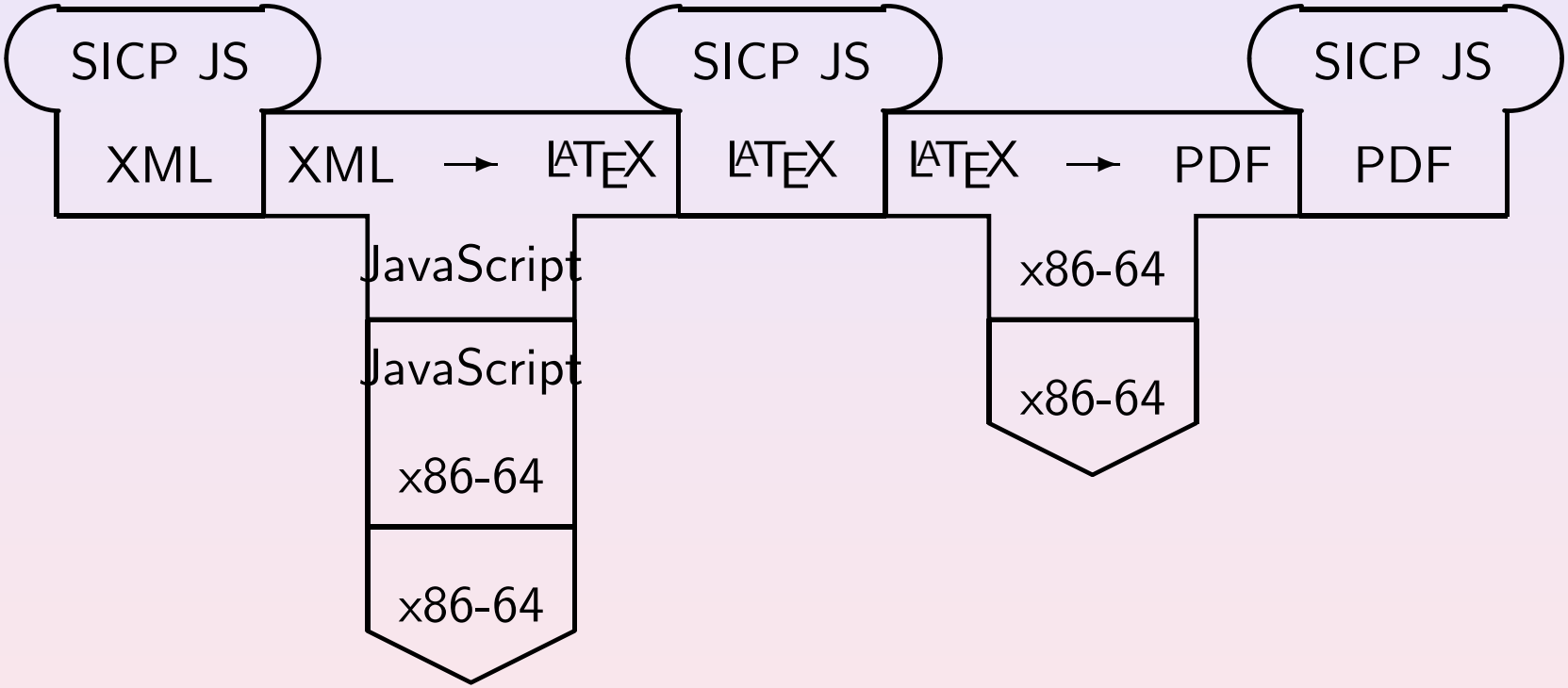
Viewing the slides on a PC

Excursion: Web edition of SICP JS



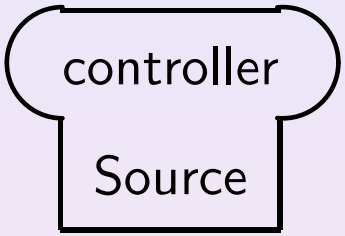
Compiling SICP JS textbook
from XML to HTML, then
Viewing textbook with browser

Excursion: PDF edition of SICP JS



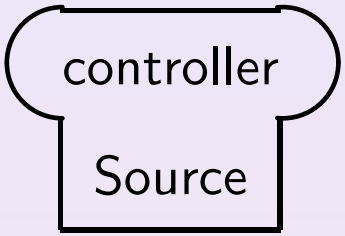
Compiling SICP JS
from XML to PDF via LaTeX

Programming Lego Mindstorms with Source



Program "controller"
(written in Source)

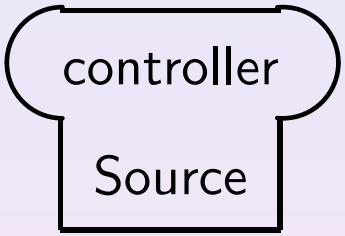
Programming Lego Mindstorms with Source



Program "controller"
(written in Source)

Lego Mindstorms EV3
(ARM5 processor)
running EV3dev,
an operating system
based on Debian Linux

Programming Lego Mindstorms with Source



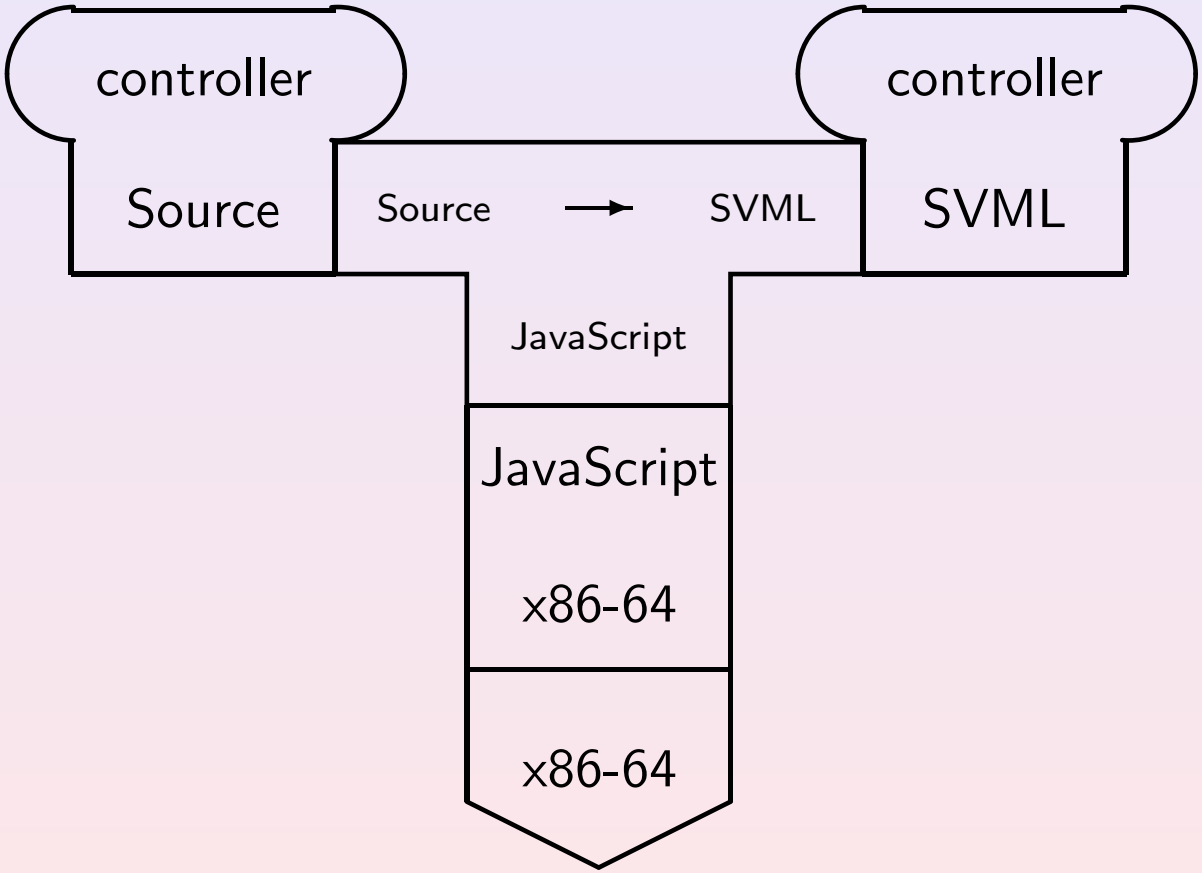
Program "controller"
(written in Source)

Lego Mindstorms EV3
(ARM5 processor)
running EV3dev,
an operating system
based on Debian Linux

Now what?

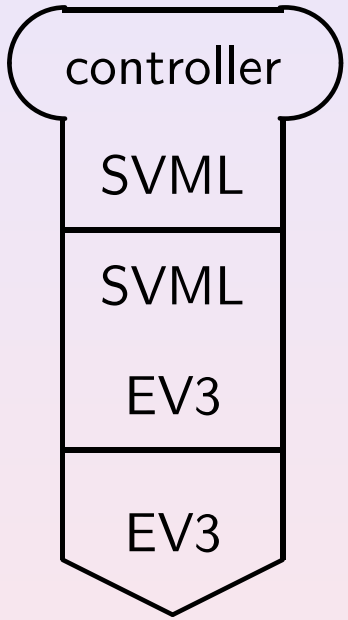
How to run Source on EV3?

SVML to the rescue!



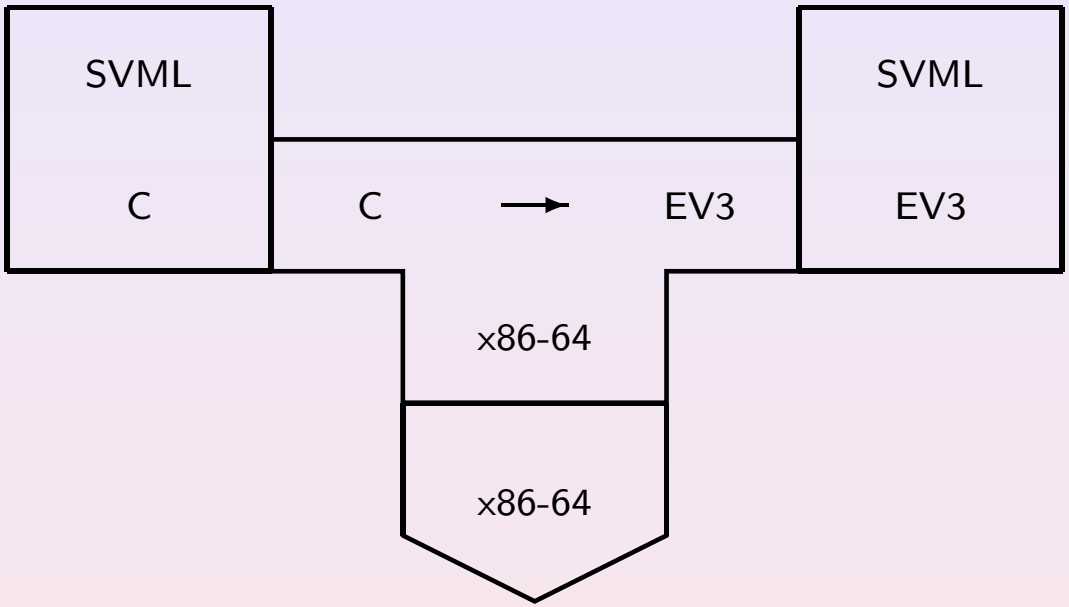
Compiling "controller"
from Source to SVML

SVML on EV3



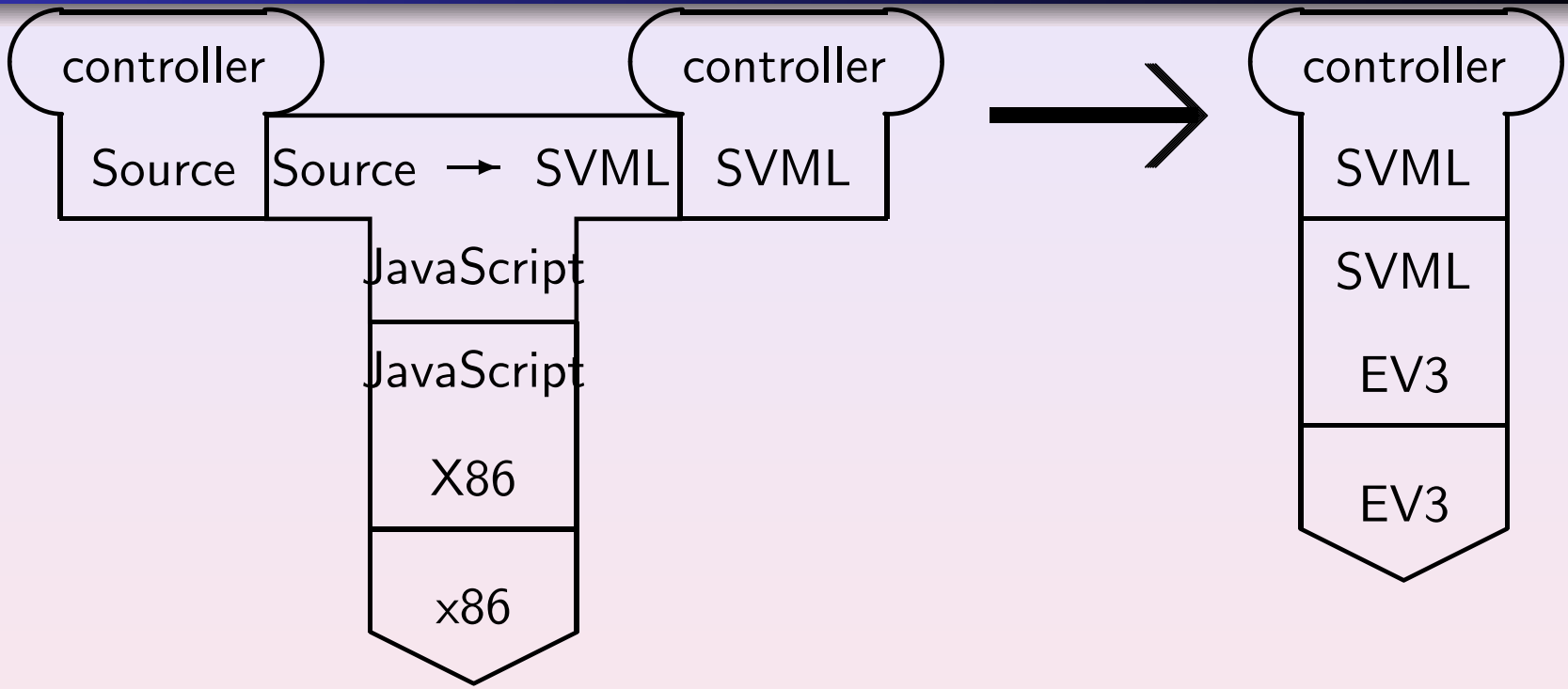
Running SVML program
on EV3 brick

Compiling SVML Emulator



Compiling SVML emulator from C to EV3

Overview: Source for Robotics



Compiling "controller"
from Source to SVML on a PC,
and then running the SVML
program on the EV3 brick.

Summary

Summary

- Components:
  programs, compilers, interpreters, machines

Summary

- Components:
  programs, compilers, interpreters, machines
- T-diagrams

Summary

- Components:
  programs, compilers, interpreters, machines
- T-diagrams
- Combination of interpretation and compilation (tool chains) are common

Summary

- Components:
  programs, compilers, interpreters, machines
- T-diagrams
- Combination of interpretation and compilation (tool chains) are common
- Source Academy is making use of Source interpreter and Source-to-JavaScript compiler, both written in TypeScript, compiled to JavaScript, and running on Chrome browser

Summary

- Components:
  programs, compilers, interpreters, machines
- T-diagrams
- Combination of interpretation and compilation (tool chains) are common
- Source Academy is making use of
  Source interpreter and
  Source-to-JavaScript compiler,
  both written in TypeScript,
  compiled to JavaScript,
  and running on Chrome browser
- Robotics in CS1101S is making use of
  SVML interpreter and a
  Source-to-SVML compiler