A Reading Guide to GHC source

- Exploring entry gates and mental models -

Takenobu T.

Rev. 0.00.0 WIP

NOTE

- This is not an official document by the ghc development team.
- Please refer to the official documents in detail.
- Don't forget “semantics”. It's very important.
- This is written for ghc 8.12.

Contents

Introduction

1. Compiler
   - Compilation pipeline stages
   - Internal representation syntax
   - Call graph

2. Runtime system

3. Core libraries

References

Introduction

Official resources are here

Official repository
URL

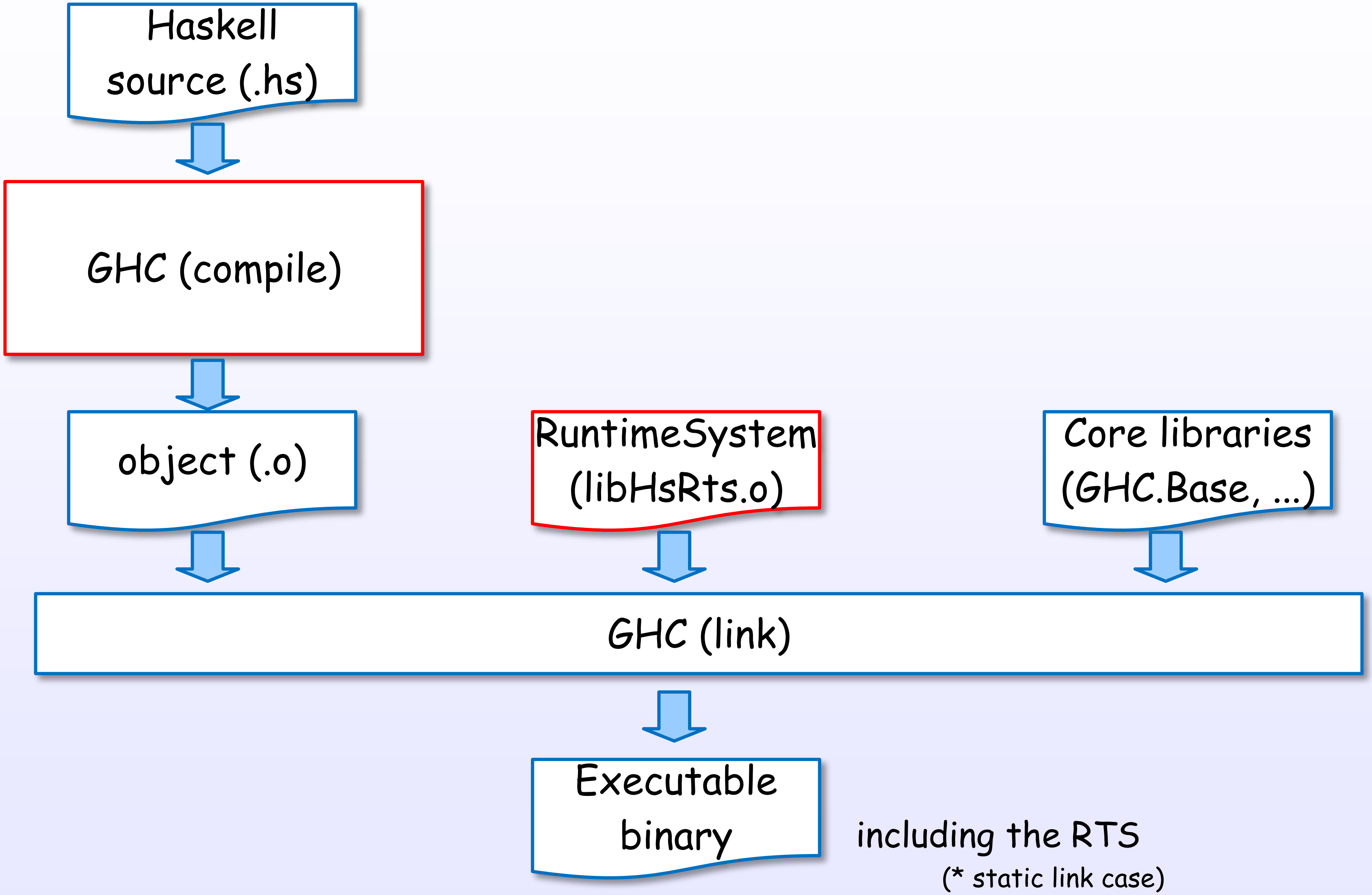
GHC Commentary wiki
URL

Web image

Web image

References : [C2], [22]

The GHC = Compiler + Runtime System (RTS) + Core libraries



including the RTS

(* static link case)

References : [1], [C1], [C3], [C12], [C21], [S7], [21], [22]

The GHC is developed by some languages

compiler
($(TOP)/compiler/*)

Haskell
+
Alex (lex)
Happy (yacc)
Cmm (C--)
Assembly

runtime system
($(TOP)/rts/*)

C
+
Cmm
Assembly

core library
($(TOP)/libraries/*)

Haskell
+
C

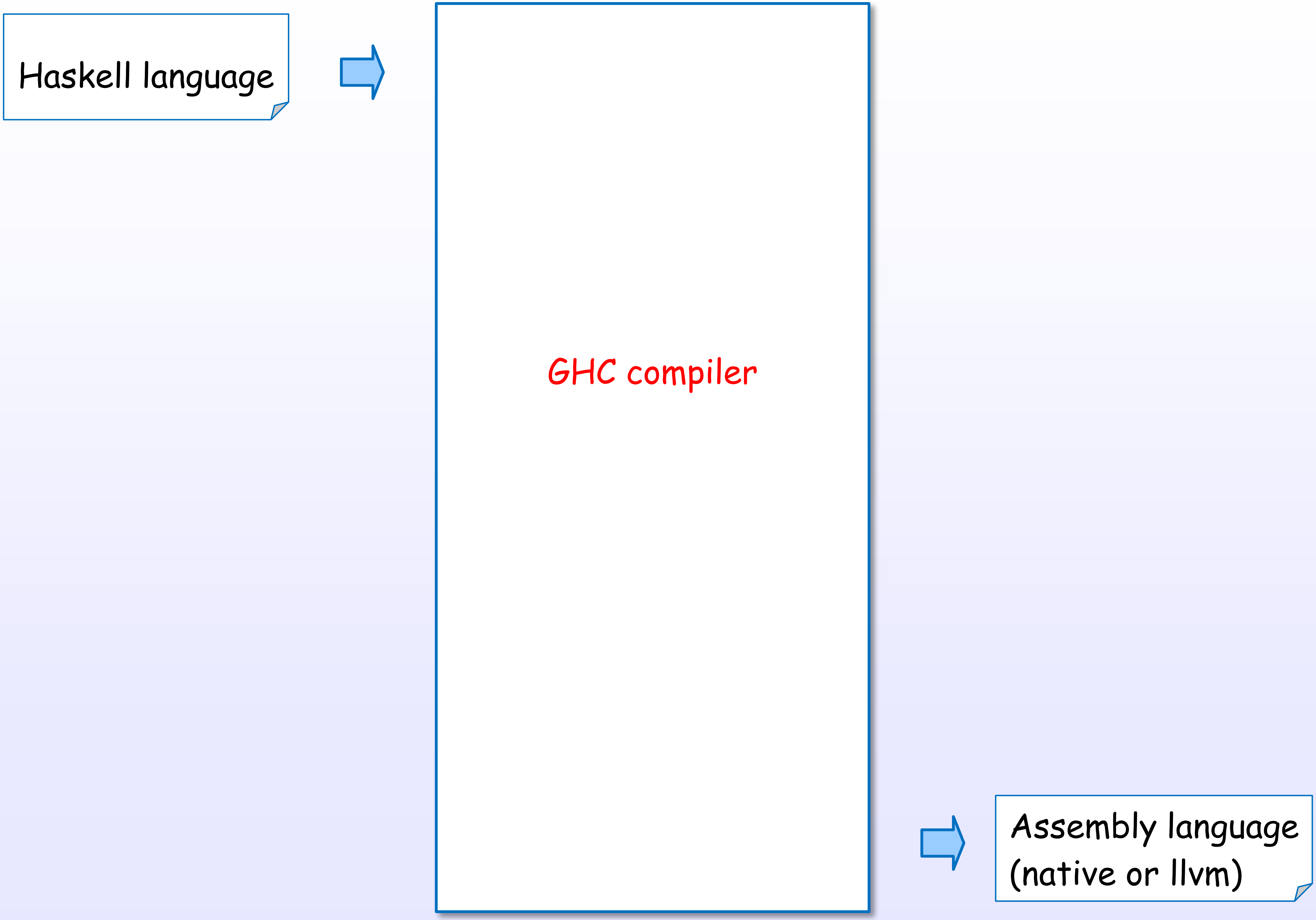
References : [C2], [22]

1. Compiler

1. Compiler

Compilation pipeline

The GHC compiler



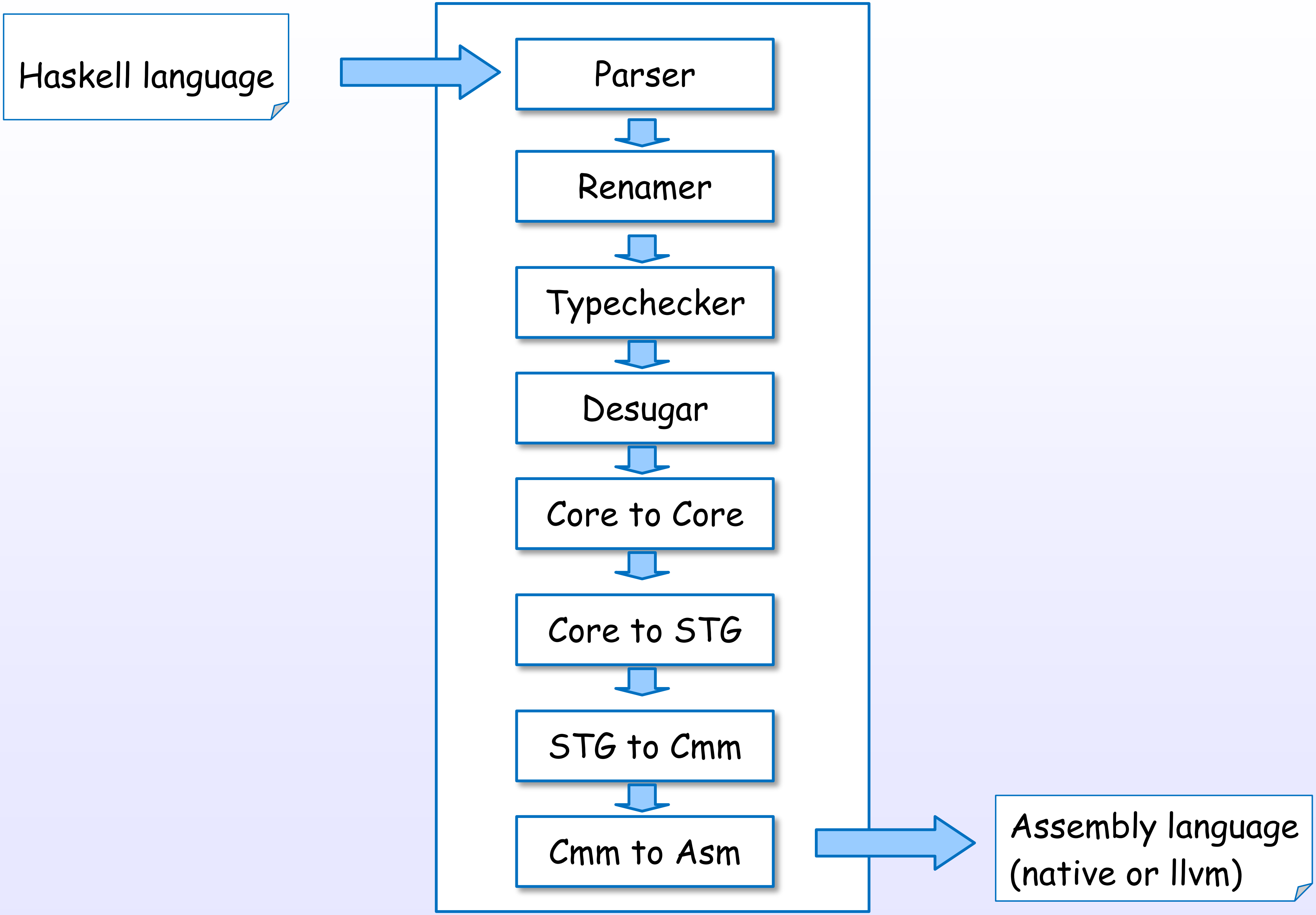
Haskell language

GHC compiler

Assembly language (native or llvm)

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

GHC compilation pipeline



Haskell language

Parser

Renamer

Typechecker

Desugar

Core to Core

Core to STG

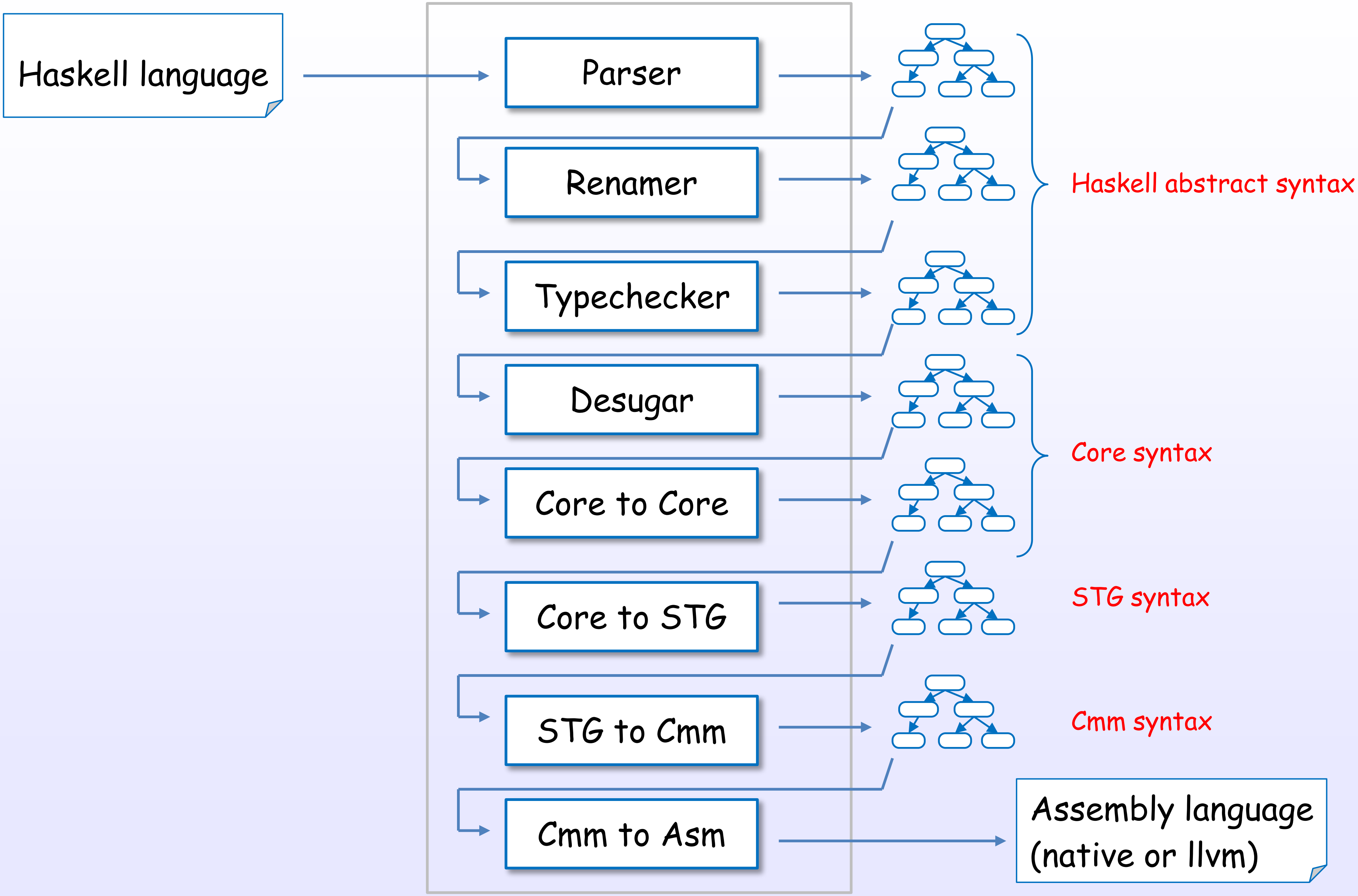
STG to Cmm

Cmm to Asm

Assembly language (native or llvm)

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

GHC compilation pipeline with intermediate languages



Haskell language

Parser

Renamer

Typechecker

Desugar

Core to Core

Core to STG

STG to Cmm

Cmm to Asm

Haskell abstract syntax

Core syntax

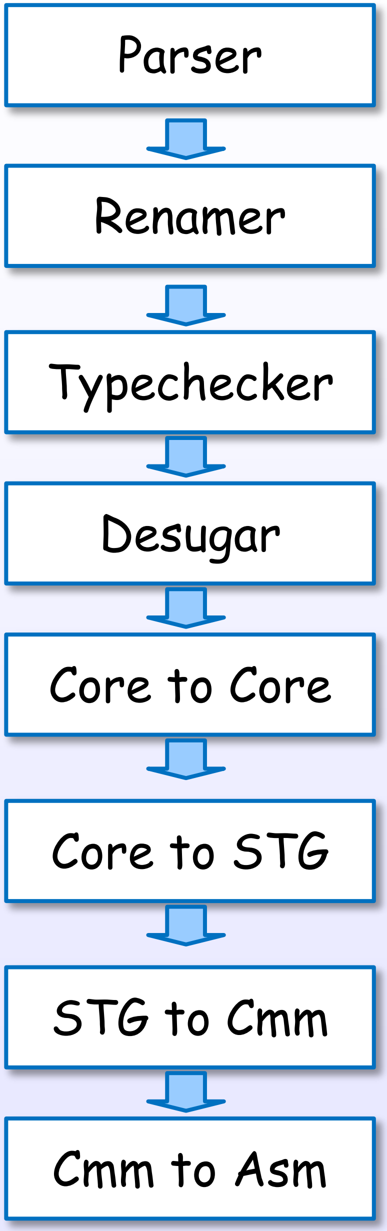
STG syntax

Cmm syntax

Assembly language (native or llvm)

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

Corresponding to source files



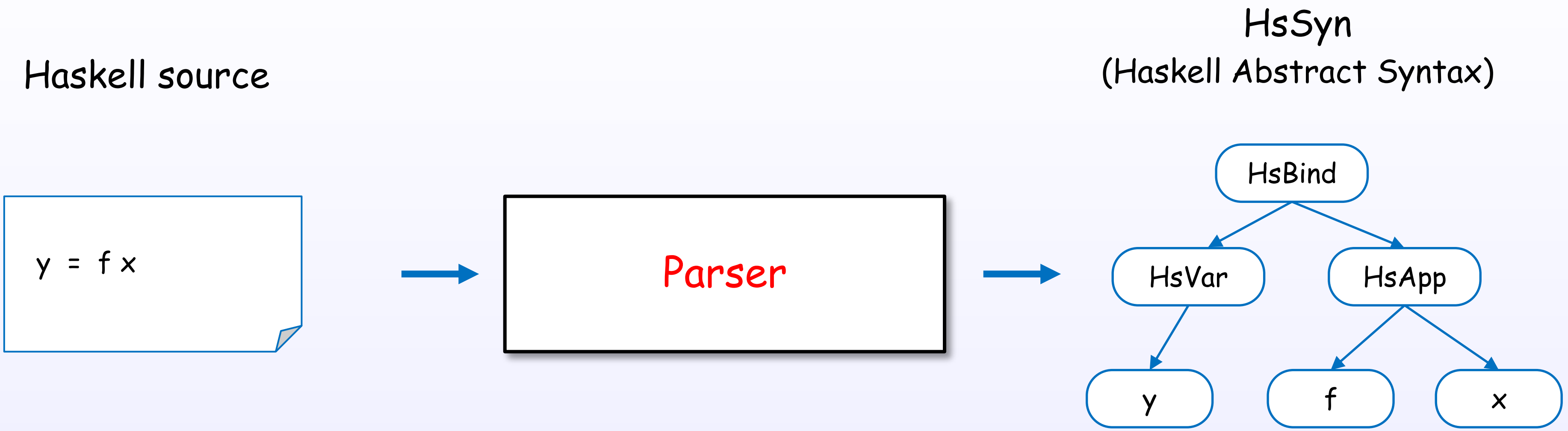
Parser	compiler/GHC/Parser
Renamer	compiler/GHC/Renamer
Typechecker	compiler/GHC/Tc
Desugar	compiler/GHC/HsToCore
Core to Core	compiler/GHC/Core/Opt
Core to STG	compiler/GHC/CoreToStg
STG to Cmm	compiler/GHC/StgToCmm
Cmm to Asm	compiler/GHC/CmmTo{Asm, Llvm}

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

1. Compiler

Each pipeline stages

Parser



Haskell source

y = f x

Parser

HsSyn
(Haskell Abstract Syntax)

HsBind

HsVar

HsApp

y

f

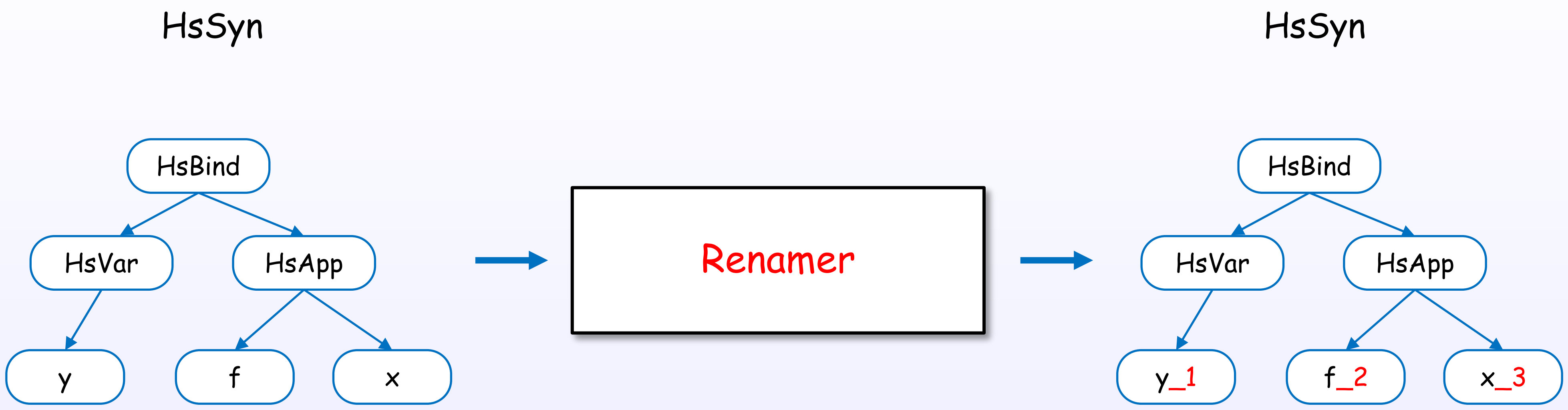
x

- Parse

Abstracted

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

Renamer



HsSyn

HsBind

HsVar

HsApp

y

f

x

Renamer

HsSyn

HsBind

HsVar

HsApp

y_1

f_2

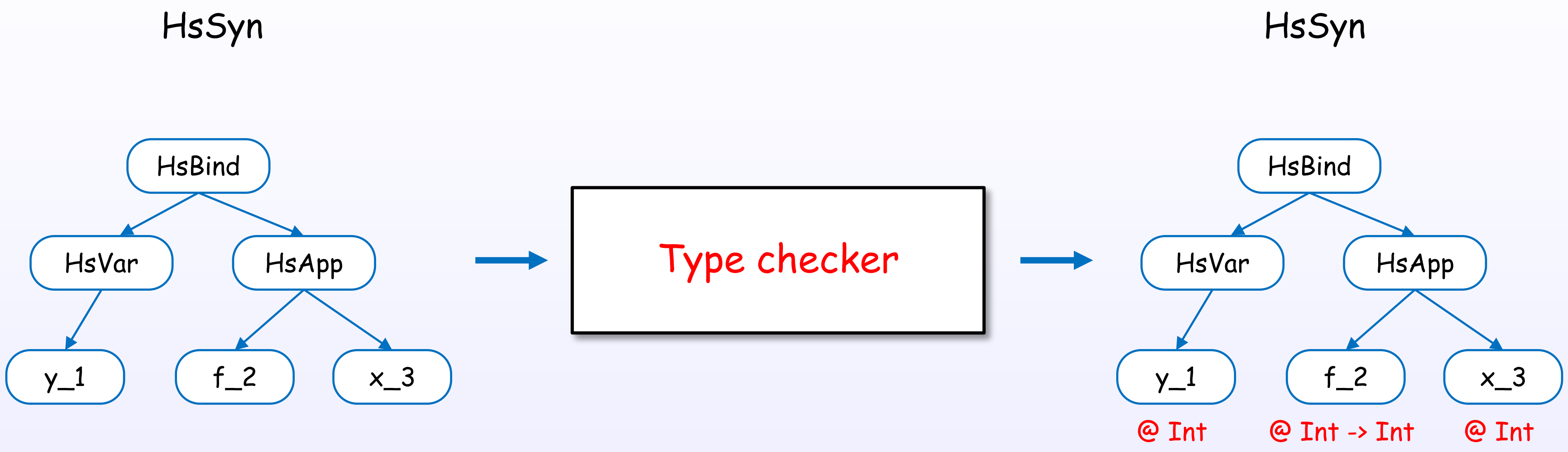
x_3

- Unify name
- Fixing
- Error check

Unique named

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

Type checker



HsSyn

HsBind

HsVar

HsApp

y_1

f_2

x_3

Type checker

HsSyn

HsBind

HsVar

HsApp

y_1

f_2

x_3

@ Int

@ Int -> Int

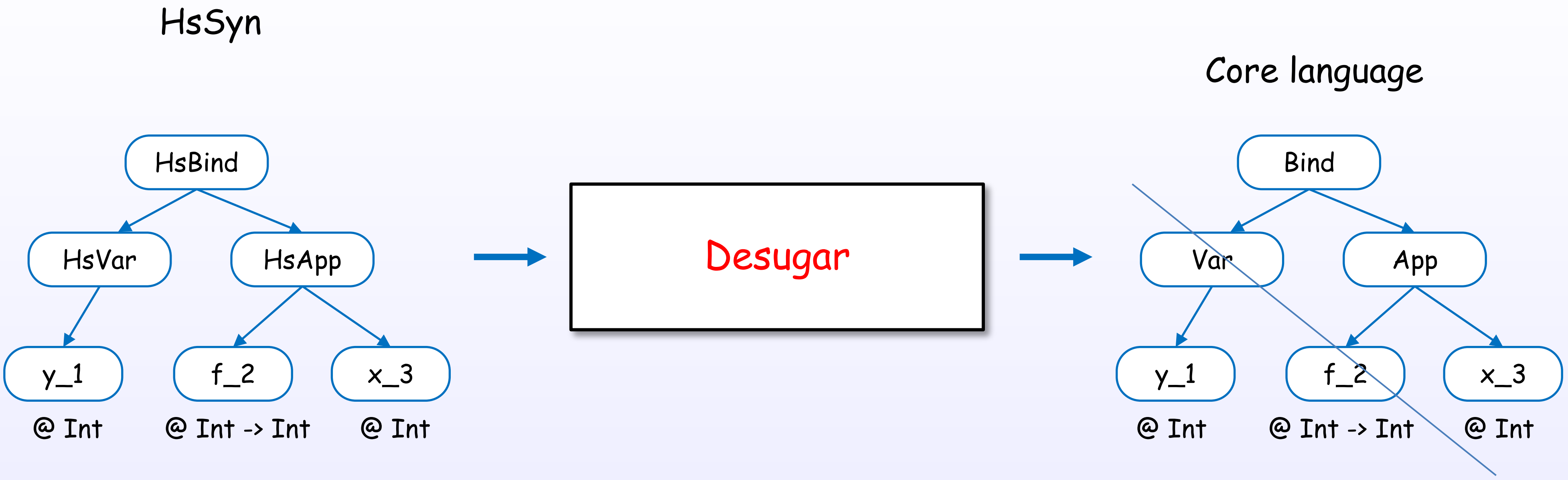
@ Int

- Infer type
- Check type error

Full typed

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

Desugar



HsSyn

HsBind

HsVar

HsApp

y_1

f_2

x_3

@ Int

@ Int -> Int

@ Int

Desugar

Core language

Bind

Var

App

y_1

f_2

x_3

@ Int

@ Int -> Int

@ Int

- Desugar to Core

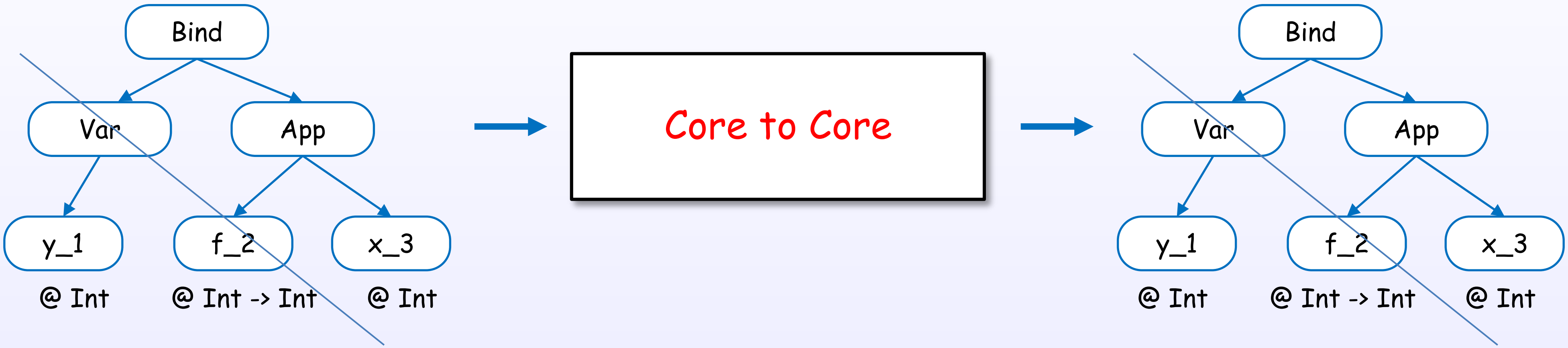
Squeeze to typed IR

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

Core to Core

Core language

Core language



Bind

Var

App

y_1

@ Int

f_2

@ Int -> Int

x_3

@ Int

Core to Core

Bind

Var

App

y_1

@ Int

f_2

@ Int -> Int

x_3

@ Int

- Simplify

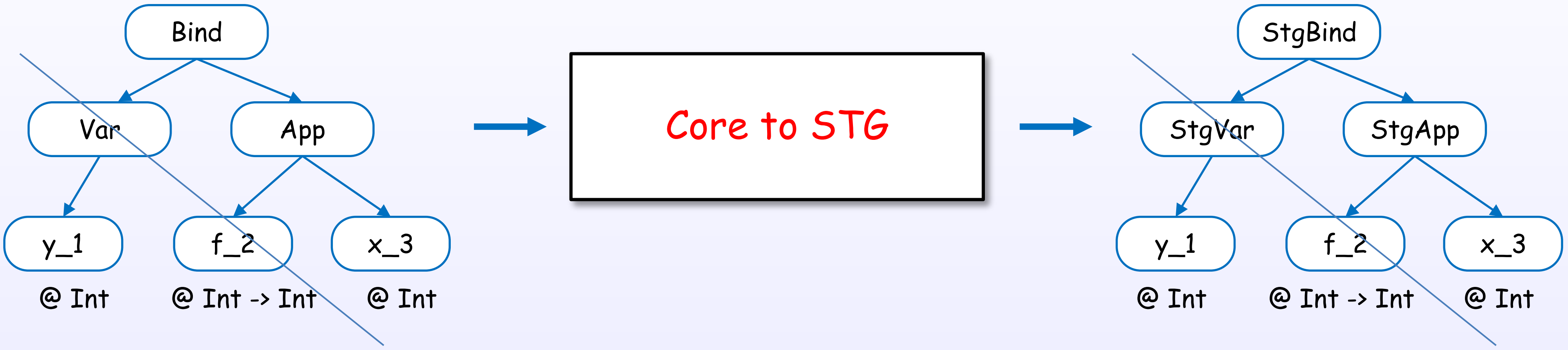
Optimized

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

Core to Stg

Core language

STG language



Bind

Var

App

y_1

@ Int

f_2

@ Int -> Int

x_3

@ Int

Core to STG

StgBind

StgVar

StgApp

y_1

@ Int

f_2

@ Int -> Int

x_3

@ Int

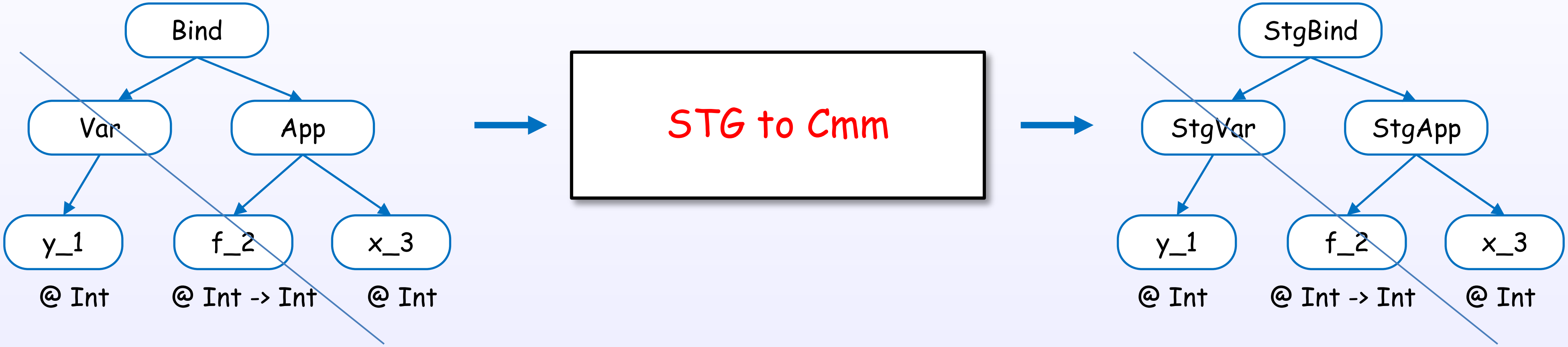
Abstract machine IR

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

STG to Cmm

STG language

Cmm language



Bind

Var

App

y_1

f_2

x_3

@ Int

@ Int -> Int

@ Int

STG to Cmm

StgBind

StgVar

StgApp

y_1

f_2

x_3

@ Int

@ Int -> Int

@ Int

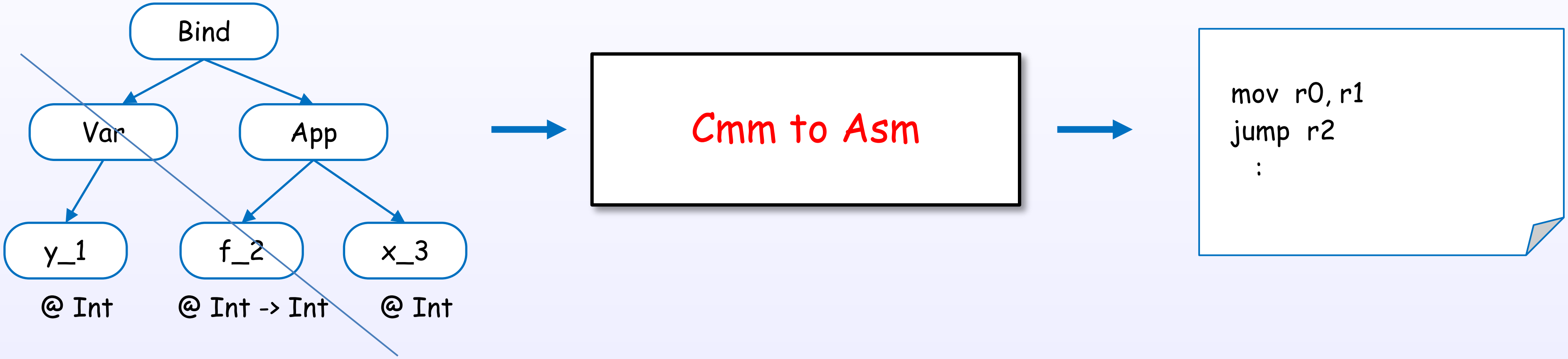
Portable Assembly

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

Cmm to Assembly

Cmm language

Assembly/LLVM language



Bind

Var

App

y_1

f_2

x_3

@ Int

@ Int -> Int

@ Int

Cmm to Asm

```
mov  r0, r1
jump  r2
  :
```

Native/LLVM code

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

1. Compiler

Internal representation syntax

HsSyn syntax

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

Core syntax

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

STG syntax

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

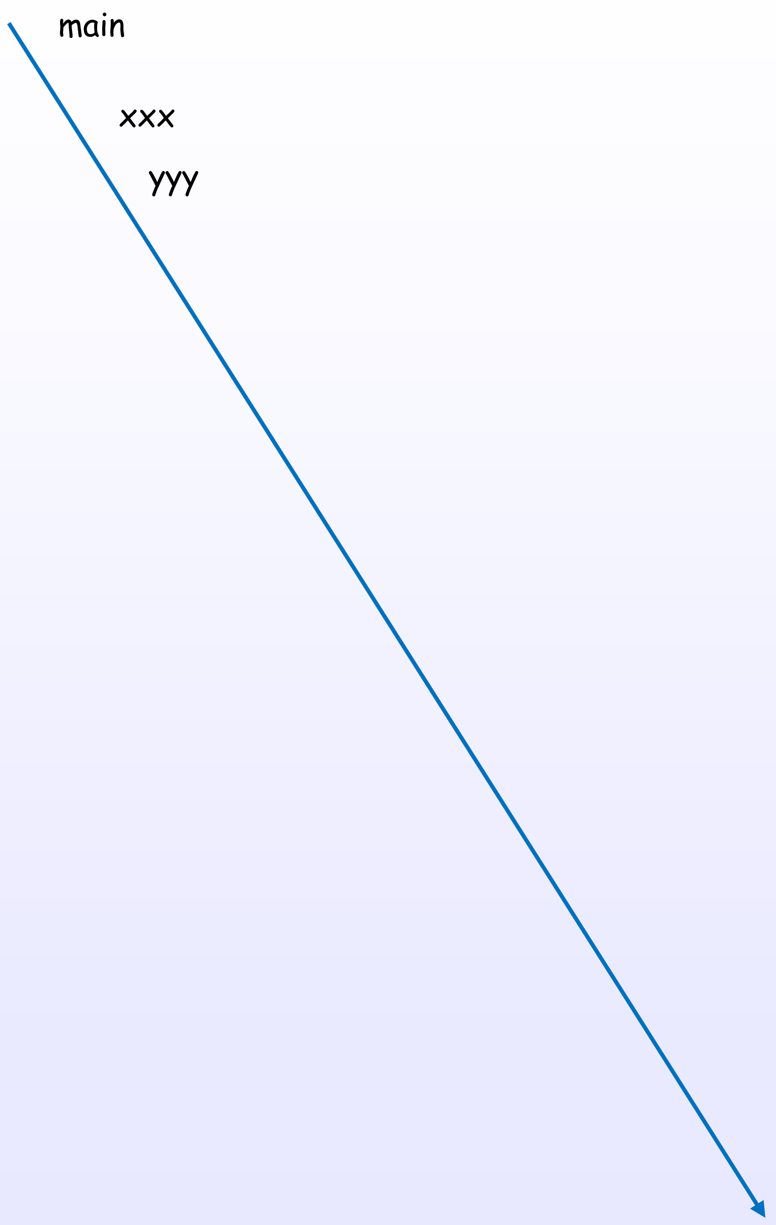
Cmm syntax

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

1. Compiler

Call graph

Example of call graph



main	ghc/main.hs
xxx	compiler/GHC/Driver/Main.hs
yyy	

References : [1], [C3], [C4], [9], [C5], [C6], [C7], [C8], [S7], [S8], [21], [22]

References

References

aosabook
dive-into-core
cs
users guide

Source code

[S1] compiler/GHC

The GHC Commentary

[C1] https://gitlab.haskell.org/ghc/ghc/-/wikis/commentary

Happy haskelling!

Here is the slide: https://github.com/takenobu-hs/haskell-ghc-reading