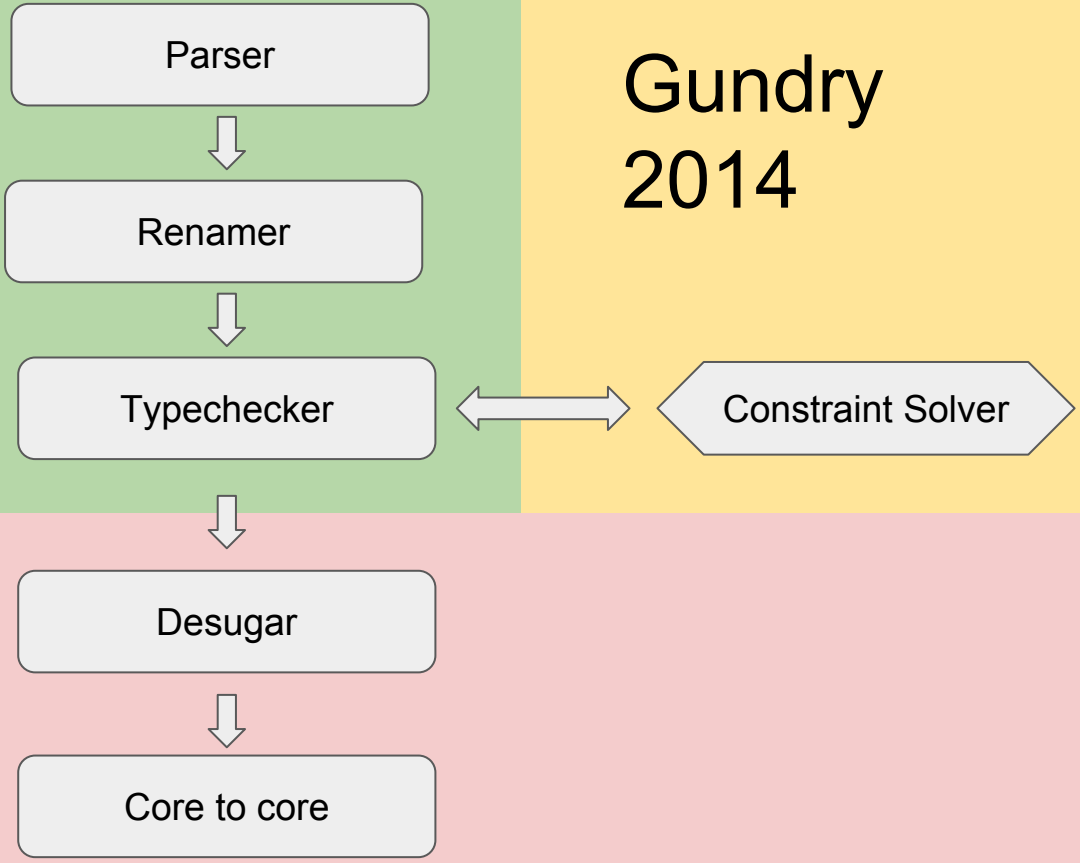
Source Plugins

Matthew Pickering

Németh 2018

Gundry 2014



Bolingbroke & Seipp 2008

Parser

```
parsed :: [CommandLineOption] -> ModSummary
               -> HsParsedModule -> Hsc HsParsedModule
```

Renamer

```
renamed :: [CommandLineOption] -> TcGblEnv
               -> HsGroup GhcRn
               -> TcM (TcGblEnv, HsGroup GhcRn)
```

Typechecker

```
typechecked :: [CommandLineOption] -> ModSummary
                    -> TcGblEnv -> TcM TcGblEnv
```

Németh
2018

```
module SourcePlugin (plugin) where

import GhcPlugins

plugin :: Plugin
plugin = defaultPlugin { parsedResultAction = parsedPlugin }

parsedPlugin :: [CommandLineOption] -> ModSummary
                -> HsParsedModule -> Hsc HsParsedModule
parsedPlugin opts _ pm
  = do liftIO . putStrLn $ "parsePlugin(" ++ intercalate ","
opts ++ ")"
       return pm
```

```
>>> ghc A.hs -fplugin=SourcePlugin
parsePlugin()
```

Plugins

- Extend
- Experiment
- Analyse

Source Plugins

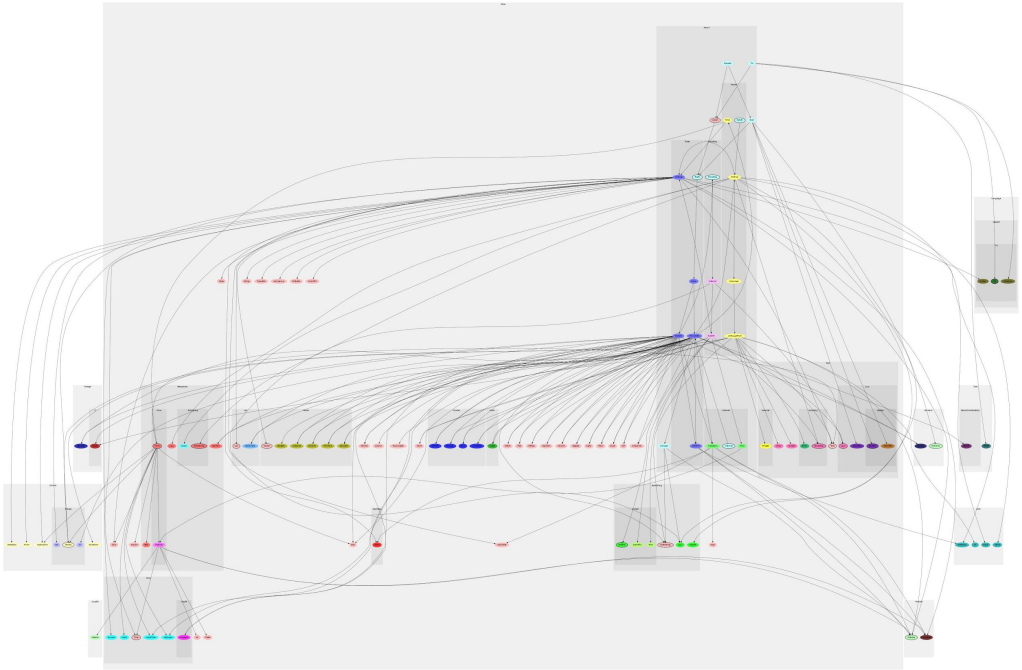
- Extend, the parser, renamer and type checker
- Experiment, language extensions, **user facing**
- Analyse, the source code

ROBUST!

Works with cabal, nix and stack and even GHC itself!

Examples

Plugin 1: `graphmod-plugin`



Module graph for `aeson`

- Typechecker plugin
- Extracts information already present in `TcGblEnv`
- Reimplementation of `graphmod`

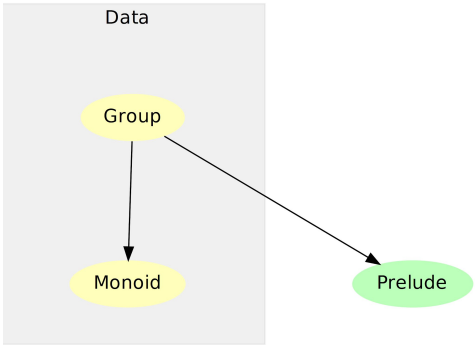
`http://mpickering.github.io/posts/2018-08-09-source-plugin-graphmod.html`

Interlude: Using plugins with nix

- Automatically runs the finaliser for plugins like `graphmod-plugin`

```
addPlugin plugins.graphmod-plugin group
```

```
> ls /nix/store/9hxjiwclgkb6v7lvvzli3fpb967i4gmb-either-5.0.1-GraphMod
modules.png  out.dot  output
```



http://mpickering.github.io/posts/2018-06-24-haskell-nix-plugins.html

Plugin 2: `idioms-plugins`

```
-- Just “foobar”
print ([ mappend (Just “foo”) (Just “bar”) ])

-- Nothing
print ([ const (Just “foo”) Nothing ])

-- Just 3
print ([ Just 1 + Just 2 ])


([ 5 ])
=>
pure 5
=>
[5]
```

- Parser plugin
- Adds new syntax for idioms brackets to GHC

http://oleg.fi/gists/posts/2018-07-06-idiom-brackets-via-source-pluging.html

Plugin 3: `assert-explainer`

```
assert (length xs == 4)


✘ Assertion failed!
    length xs == 4 /= True
(at Test.hs:18:12-25)

  I found the following
sub-expressions:
    - length xs = 3
    - xs = [1,2,3]
```

- Typechecker plugin
- Interaction with constraint solver
- Instrumenting code, producing expressions

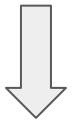
https://github.com/ocharles/assert-explainer

Plugin 4: `smuggler`

```
module Main where

import Data.Bool (bool)

main = putStrLn “Hello World”
```



```
module Main where

main = putStrLn “Hello World”
```

- Typechecker plugin
- Uses GHC API functions
- Rewriting source code

https://github.com/kowainik/smuggler

Plugin n?

- `haskell-indexer`?
- `SourceGraph`?
- A common compiler output for writing plugins?

This page provides an index of different resources about GHC plugins.

Blog posts

- mpickering - 2018-08-10 - Specifying how a plugin affects recompilation
- mpickering - 2018-08-09 - Reimplementing graphmod as a source plugin
- phadej - 2018-07-06 - Idiom brackets via source plugin
- mpickering - 2018-06-24 - Nix scaffolding for running Haskell plugins
- mpickering - 2018-06-11 - Source Plugins: Four ways to build a typechecked Haskell expression
- nboldi - 2018-04-14 - Proposal 0017: Source Plugins
- nomeata - 2018-02-02 - The magic "Just do it" type class
- Tweag I/O - 2017-09-22 - GHC compiler plugins in the wild: typing Java
- christiaanb - 2016-08-17 - Solving custom operatns in KnownNat constraints
- christiaanb - 2016-08-10 - Solving GHC's KnownNat constraints
- Adam Gundry - 2015-07-17 - A Typechecker Plugin for Units of Measure
- Eric Seidel - 2014-12-04 - (Ab)using Compiler Plugins to Improve Embedded DSLs
- Max Bolingbroke - Compiler Plugins For GHC 123456
- Max Bolingbroke - 2008-04-24 - The Summer Of Code, or Compiler Development for the Masses

Tutorial plugins

`plugin-constraint` - mpickering
How to interact with the contraint solve and generate `HsExpr GhcTc`.

`hashtag-coerce` - mpickering
How to perform a simple static analysis using `syb` to traverse the syntax tree.

`ghc-typelits-gcd` - christiaanb
An example of implementing a constraint solver plugin

Source plugins

`hlint-source-plugin` - ocharles
Runs `hlint` during compilation and reports errors are though they are GHC errors.

`assert-explainer` - ocharles
Instruments an assertion to print the value of its failed parts when it fails.

`what-it-do` - ocharles
Rewrites `do` expressions to trace all binds.

`smuggler` - kowainik
Plugin which removes unused imports automatically.

`lift-plugin` - mpickering
An experimental plugin which introduces an operator which acts like an overloadable bracket.

`graphmod-plugin` - mpickering
A reimplementation of `graphmod` as a source plugin.

`idioms-plugins` - phadej
Implementation of idiom brackets using a parser plugin.

Core plugins

`dump-core` - yav
A pretty printer which renders HTML for GHC's internal representation

`HerbiePlugin` - mikeizbicki
GHC plugin that improves Haskell code's numerical stability

`sbvPlugin` - LeventErkok
Prove properties about Haskell programs using SBV/SMT.

`ghc-justdoit` - nomeata
Plugin which synthesises a definition from a type, like `djinn`.

`inspection-testing` - nomeata
Embed assertions about the intermediate code and have them checked by GHC.

`ghc-srcspan-plugin` - gridaphobe
Generic GHC Plugin for annotating Haskell code with source location data.

http://mpickering.github.io/plugins.html

popular plugins (& plugin-based tools)

★ 3617	Zinc
★ 3433	Scala.js
★ 3360	Scala Native
★ 806	WartRemover
★ 650	Scalameta
★ 521	Kind Projector
★ 412	scoverage
★ 347	Scalafix
★ 269	Scapegoat
★ 256	Linter
★ 231	better-monadic-for
★ 225	Splain
★ 194	Acyclic
★ 147	Macro Paradise
★ 115	Silencer
★ 72	Twotails
★ 64	Sculpt
★ 57	Continuations
★ 40	Genjavadoc
★ 30	Applicative Syntax
★ 20	scala-xgettext

…...

Conversations in Montrose Avenue, PA

Dang! Anything is possible with a source plugin.

Ryan Scott