New language stuff in GHC 7.8

- Overloaded list literals (Achim Krause)
- Minimal-class-ops pragma (Twan van Laarhoven)
- Type holes and deferred type errors
- Closed type families (Richard Eisenberg)
- Type-level naturals (Iavor Diatchki)
- Role checking (Richard Eisenberg) - finally fixes the Dreaded Newtype Deriving bug
New language stuff in GHC 7.8

- Polykinded Typeable (Pedro Magalhães)
- Coercible (Joachim Breitner, Richard Eisenberg), finally fixes #1496 GND bug
- New Template Haskell (Geoff Mainland)
- Pattern synonyms (Gergo Erdi)
Finally, the ability to abstract over patterns

```
data T = App String [T]
pattern Arrow a b = App "->" [a,b]
f :: T -> Bool
f (Arrow a b) = True
f _ = False
```

- Can be exported/imported
- Works for existentials, GADTs
- Bidirectional pattern synonyms
New, parallel I/O manager (Andreas Voellemey)

New STG -> C-- code generator (Simon Marlow)

New primops for
  - SIMD (Geoff Mainland)
  - Atomic memory ops (Ryan Newton)

Unboxed Booleans (Jan Stolarek)

Dynamic linking....

Better "hooks" for the GHC API (Luite Stegeman)
Infrastructure

- GHC --make -j8 (parcs)
- Better cross compilation
- Phabricator for code reviews
- Travis (and Phabricator) for continuous integration
- See GHC's status page
  https://ghc.haskell.org/trac/ghc/wiki/Status
GHC 7.10

- Release candidate by Christmas 2014
- Release in Feb 2015

So, what will be in it?
Partial type signatures (Thomas Winant et al)

\[
\begin{align*}
f & : \_ \to \text{Int} \to [\_]\_ \\
g & : (\_\_) \Rightarrow \text{a} \Rightarrow \text{a}
\end{align*}
\]

GADT-aware pattern-match overlap/exhaustiveness warnings (Tom Schrijvers et al)

Support for Cloud Haskell 'static' (Mathieu Boespflug and Facundo Domínguez)

Applicative do-notation (Simon Marlow)
7.10: language

- Injective type families (Jan Stolarek)
- Strict Haskell (Johan Tibbell)
- Applicative as a superclass of Monad (finally)
- Overloaded record fields (Adam Gundry)
- Explicit type application (Stephanie Weirich, Richard Eisenberg and Hamidhasan Ahmed)
Using an SMT solver in the type inference engine (Iavor Diatchki)

GADTs at the kind level (Stephanie Weirich, Richard Eisenberg)
7.10: internals

- Better profiling (Peter Wortman, Karolis Velicka)
  - DWARF-based stack tracking
  - Real-time processing of GHC events stream
- New GMP Integer library (Herbert Valerio Riedel) without relying on giving GMP a GHC-specific custom memory allocator
- Out-of-process Template Haskell (Luite Stegeman)
- Signature sections
- $(\cdot\cdot \text{Int}) = \lambda x \mapsto x :: \text{Int}$
- Faster arrays. Faster. Faster (Johan)
- Generalising overloaded list syntax, \texttt{Hcons, hNil} (Carter)
- Left-fold fusion (Joachim)
- Pattern families (Baldur)
- relocatable \texttt{GHC installs} (Christian)
Goal: turn Hackage into a library of software components:

- Each package depends on some module signatures (its holes)
- ...and exposes some modules to its clients
- Compose packages together to make bigger packages

Example: a package depends on a String signature; later plug in regular Strings, or lazy ByteStrings, or strict ByteStrings
Backpack code describes packages

Currently foo.cabal describes package foo

Glorious plan: gradually migrate .cabal files, so that they become more expressive

"Backpack is the calculus, cabal is the concrete source language"
Stages on the way

- Possible to install (eg) text-2.9 more than once, each instance compiled against different versions of its dependencies.

  Major improvement to Cabal Hell

- Selective import, renaming, and re-export from dependencies

exposed-modules: MyModule
reexported-modules: String
build-depends:
  bytestring(Data.ByteString.Lazy as String)
Stages on the way

- Module signatures, packages with holes
  (the big one, in flight)
- All this requires a delicate dance between GHC and cabal
THE GHC COMMUNITY
The big picture

- Simon Marlow left MSR for Facebook
- Ian Lynagh moved to Semle
- New GHC support team at Well Typed
  - Austin Seipp
  - Edsko de Vries
- Many new active contributors to GHC
- FP Complete launches
- Core libraries committee starts
- SPJ hits 55, the children leave home, and the dog dies

GHC is growing up and leaving home
The big picture

- Simon Marlow left MSR for Facebook
- Ian Lynagh moved to Semle
- New GHC support team at Well Typed
  - Austin Seipp
  - Edsko de Vries
- Many new active contributors to GHC
- FP Complete launches
- Core libraries committee starts

- SPJ hits 55, the children leave home, and the dog dies
- SPJ hits 56, adopts three children. Haskell the cat is alive and well.

GHC is growing up and leaving home
What does that mean?

- GHC is flourishing
  - lots of users
  - lots of innovation
  - lots of contributors
  - Trac tickets approaching 10,000

- GHC is more and more a community project, both in leadership and execution

- This is not a bad thing. But it relies on people actually stepping up.

Introducing the new Well Typed GHC support team:
- Austin Seipp
- Mikolaj Konarski

- Major role
- Not so much doing, but enabling others to do
What does that mean?

Bottom line

GHC and
(especially) its ecosystem
(cabal, Hackage, Haskell Platform...)
badly need your help
GHC has a huge surface area

- type system
- optimisation
- code generation for many platforms
- SIMD instructions
- dynamic linking
- GHCi
- the GHC API
- plugins
- FFI

- concurrency, STM
- Cloud Haskell
- the build system
- packages
- garbage collection, finalisers
- run-time system, scheduling
- profiling
- ....and more...