Modifications of GHC for HasFuse

David Sabel

September 1, 2003

This document contains a list of the modifications of the GHC source code (version 5.04.3) for HasFuse.

Notice: Modified or added lines in the source code are marked with `[DS]`, so you can easy search for those lines.

ghc/compiler/main/DriverFlags.hs

- I added a new flag `--hasfuse` which shows some information about HasFuse.
- I added a new flag `-fstrictness`, which turns on strictness analysis, because strictness analysis is off by default in optimisation level 0 and 1 (see comments about `DriverState.hs`).

ghc/compiler/main/DriverState.hs

- Changes in `hsc_minusO_flags`:
  - ignore interface pragmas
  - omit interface pragmas
  - don't do the foldr-build transformation
  - turn off cpr
- Changes in `hsc_minusO2_flags`:
  - There's a real definition now, because we turn foldr-build on and cpr not off. The rest of the flags are same as `-O1` flags.

- Big changes in `buildCoreToDo` for `-O1` and `-O2`:
  - Strictness analysis is now off by default at level 0 and 1, but it can be switched on by the flag `-fstrictness` (Worker Wrapper is then not on!).

    You can't switch off strictness analysis at level 2 (the flag `-fno-strictness` has no effect)
  - Optimisation level 1:

    Do only Floating-In and run the Simplifier
  - Optimisation level 2:

    Do every optimisation as before, that aren't tested for safeness.
  - The unsafe transformations Floating-Out and Common Subexpression Elimination aren't performed in any optimisation level! The liberate case transformation is also never performed, because we have nearly no documentation about it.

ghc/driver/ghc-usage.txt

- I added information about HasFuse and the `--hasfuse` flag.
- I added information for use of optimisation level 2, which isn't proven as safe.

ghc/mk/version.mk

- I changed the Projectname to HasFuse

ghc/compiler/coreSyn/CoreUtils.lhs

- I've made some changes in the definition of `exprIsCheap`:
  - An expression is now cheap, if it's a variable, an unapplied primitive operator with positive arity, a literal, a constructor or a constructor application where the arguments are cheap.
  - I also allow things about types to be cheap, because FUNDIO has no types.
  - Primitive operators are treated like abstractions, so primitive operators with arity 0 aren't cheap
- I've made some changes in the definition of `exprIsTrivial`:
  - I don't allow to duplicate primitive operators or foreign-call Ids with arity 0, so I changed `exprIsTrivial` for the `Var` case.

- I replaced the definition of `exprEtaExpandArity` by `exprArity`, because the eta-expansion performed by the GHC is not safe in the sense of FUNDIO.
- I've made some changes in the definition of `exprIsValue`:
  - Partial applications are no longer treated as values in general. I only allow applications to constructors with to few arguments as values.
  - `exprOkForSpeculation` is no longer used in `exprIsValue`, it's replaced by `False`. It would be nice to turn off `exprOkForSpeculation` fully. I tried that with the definition `exprOkForSpeculation = False`, but then the compiler fails while building the libraries (`Data/Array/Base.hs`).

ghc/compiler/simplCore/OccurAnal.lhs

- I've made some changes in `occurAnal` for the `Lam` case, so that `InsideLam` is now also `True` for one-shot-lambdas.

ghc/compiler/simplCore/SimplUtils.lhs

- I changed `mkCase1`: We don't check for `exprOkForSpeculation`, because we don't know if case elimination is correct in this case.

ghc/compiler/main/Main.hs

- We print a new message for the modified version (HasFuse), each time GHC is running.
- I changed the verbosity information for HasFuse.

ghc/compiler/simplCore/SimplMonad.lhs

- I turned off RULES completely, by changing the definition of `activeRule`, so that the result is always `Nothing`.

ghc/compiler/utils/Panic.lhs

- I changed the panic-message in `showGhcException (Panic s)`, bugs should be reported to the HasFuse webpage.

ghc/compiler/main/SysTool.lhs

- I added a new function `showHasFuseMess`, that's used for printing the HasFuse message, which comes with flag `--hasfuse`. Therefore I added the global var `v_Path_hasfusemess` with initialisation.

ghc/driver/hfmess.txt

- This file is new, it contains the message that comes with `--hasfuse`.