

Android むけ Haskell コンパイラをリリースしました！

… Pre-release ですが。

(In English*¹ / In Japanese)



書いた日：2020年12月15日、書いた人：@unnohideyuki*²

はじめに

これは Haskell Advent Calendar 2020*³ 15 日目の記事です。

4 年まえに「Haskell コンパイラを書こう！」という記事で紹介した自作コンパイラを、いよいよリリースしますというお話です。その記事で述べた通り、Haskell 2010 仕様を満たし、かつ、FFI をサポートしたら Version 1 にしようと思っているのですが、まだそこまで至っていないため、今回は 0.9.0 版としてリリースします。

Bunny - A Haskell compiler for Android

Bunny は、Android 上で動くアプリケーションの作成をターゲットとした Haskell コンパイラです。このコンパイラを作成するにあたって、基礎的な部分を「最新コンパイラ構成技法」という本で勉強したので、その本の通称である Tiger Book にちなんで Bunny という名前にしました。

Bunny は、Haskell で書かれた Haskell コンパイラです。Haskell プログラムをソースコードとし、オブジェクトコードとしては Java のコードを吐きます。それを、ランタイムライブラリ（これも Bunny プロジェクトに含まれます）とともにビルドすることで、Android アプリを作ることができます。Haskell プログラムをコンパイルして、オブジェクトコードである Java プログラムに変換する過程については、4 年前の記事を参照してください。大まかな構造は当時から変わっていません。

Bunny は、GHC (Haskell Platform) と Android Studio がインストールされた Windows または Linux マシン上で動作します。Bunny の入手、および、ビルド方法などについては、Bunny プロジェクトのページをご覧ください。

*¹ <https://uhideyuki.sakura.ne.jp/studs/index.cgi/en/BunnyHaskellCompiler009>

*² <https://twitter.com/unnohideyuki>

*³ <https://qiita.com/advent-calendar/2020/haskell>

使用例

Bunny の 0.9.0 版では、まだ機能制限も多いのですが (制限の概要についてはリリースノート をご覧ください)*⁴、簡単なプログラムをコンパイルして、Android アプリをつくることができます。

一昨日の AtCoder Beginner Contest 185 から C 問題*⁵ を例にとって、この問題を解くプログラムを書いてみます。たとえば、以下のように書けます：

```
getInteger :: IO Integer
getInteger = do
  s <- getLine
  return $ read s

main = do
  l <- getInteger
  let ans = product [(l-11)..(l-1)] 'div' product [1..11]
  print ans
```

適当な作業ディレクトリ上でこのプログラムを abc185c.hs という名で保存したら、Bunny を用いてコンパイル、Android プロジェクトを作成します。

```
bunny android abc185.hs
```

コンパイルに成功すると、次のようなメッセージが表示されるはずです：

```
-----
An android project has been created!
  path: $HOME/BunnyProjects/abc185c

then you can try:
$ cd $HOME/BunnyProjects/abc185c
$ ./gradlew assembleDebug
$ adb install app/build/outputs/apk/debug/app-debug.apk
-----
```

*⁴ ここに書かれていないバグも相当あると思います。

*⁵ https://atcoder.jp/contests/abc185/tasks/abc185_c

このメッセージの勧めにしたがって、作成された Android プロジェクトのあるディレクトリに移動し、`./gradlew` コマンド (Windows の場合は `./gradlew.bat`) をつかって Android アプリをビルドします。まずは、`assemblyDebug` タスクで debug 用 APK を作成します。

```
cd $HOME/BunnyProjects/abc185c
./gradlew assembleDebug
```

次に、できた apk を Android 仮想マシン上で動かしてみます。これには、Android SDK に含まれる `adb` コマンドを使いますが、その前に仮想マシンを起動しておく必要があります。Android Studio を起動し、メニューから ” Open AVD Manager” で AVD Manager を起動、なにか適当な仮想マシンを立ち上げておきます。

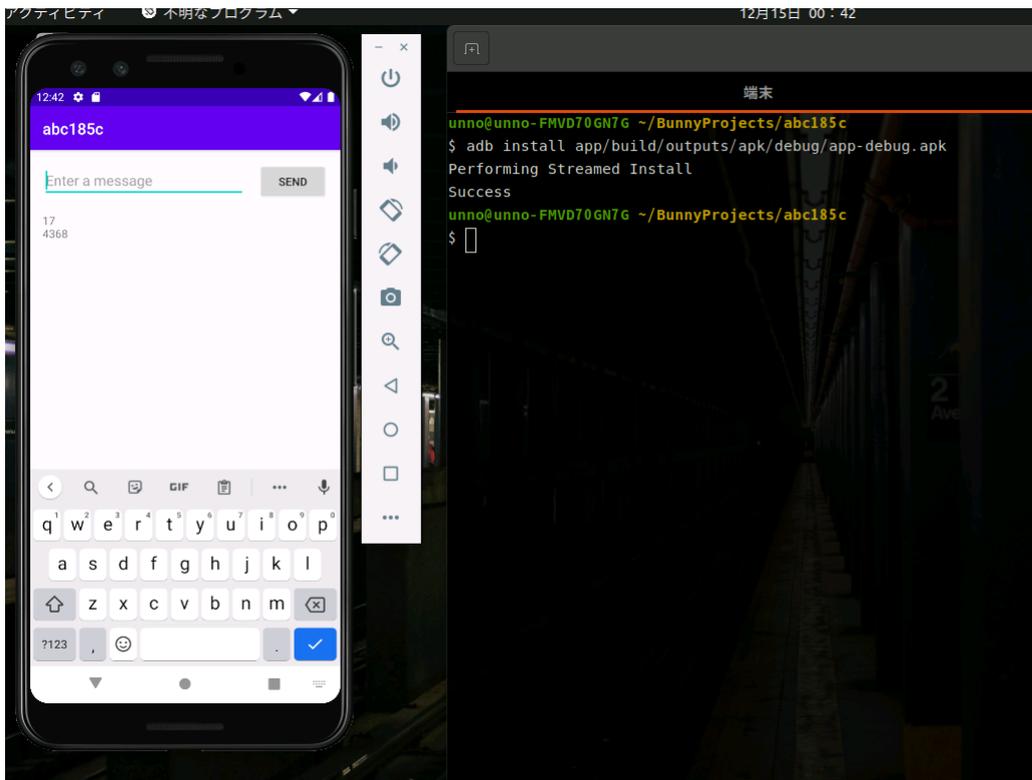
仮想マシンを立ち上げてから、`adb` コマンドをもちいて apk ファイルを `install` します。

```
adb install app/build/outputs/apk/debug/app-debug.apk
```

その後、仮想マシンを操作して `abc185c` という名前のアプリを起動します。このプログラムは、標準入力から一行読み込んで、それに応じた計算結果を表示するプログラムなので、はじめはなにも表示されていません。

ためしに、`ABC185C*6` の入力例 3 と同じ `17` と入力して、`SEND` ボタンを押します。表示領域に `17` と `4368` の二行が表示されたら成功です。

*6 https://atcoder.jp/contests/abc185/tasks/abc185_c



Version 1 にむけて

できることなら、今日「Version 1 できました！」と言いたかったところなのですが、かなり残項目がおおく間に合いませんでした。ですが、いままでずっと「作りかけ」の状態だったのを、pre-release であつてもリリースしようとしたことで、ツールとしての完成に大きく近づけることができたように思います。

Version 1 にむけては、とにかく、Haskell 2010 の仕様をみたくこと、および、FFI を実装することを目標に進めていきたいと思います。性能については、コンパイルにかかる時間も、コンパイルした結果できあがったアプリの実行性能も、いずれもかなり悪いのですが、性能問題に取り掛かるのは Version 1 に到達してからと思っています。

今回 0.9.0 版をリリースしたことで、かなり足場がかたまってきとことと、「また 1 年後」では、わたし自身が飽きてしまうので、できれば半年後の 5 月ころには Version 1 をリリースできればなあと思ってます（そのあたりは、ロードマップにも記載してあります）。

ひきつづき頑張ります！

…と意気込みを述べたところで、今回の記事はおしまいです。