

## 013: Typing で各項に型をつけるようにする

↑ up

- issued: 2020-04-11
- 分類 : C 改善項目
- status: Rejected (2020-04-15)

### 現象

Typing モジュールの結果のうち、変数に束縛されたものしか後段に送られていない。現状では、Assump を出力して、それを用いて辞書渡し変換などを実施している。

Typing の各項に型をつけてしまえばいいように思われる。

関連 Issue: - 009: nodup0.hs で dictionary not found

Steps: 1. ブランチをきる 2. Typing 後のモジュールの現実装における、型情報の用いられ方を整理 3. Typing で各項に型をつけて出力する方法の検討 4. 実装

### 調査ログ

2020-04-11 (Sat)

branch: i013

tiProgram の Subst をすべて引き継いだらダメなんだろうかと思い、それを tcBind に渡してみることにした（そのために複数の箇所を一時的に改変）。

dictionary not found になる箇所で、この、新たに渡した subst なら適切にその型変数に型がつけられるかと思ったが、うまくいかず。だめみたい。

ならば、このように多相になる項に、中間言語の時点で型推論前に変数に束縛してやればどうかなと思ったが、現状の tiProgram では、トップレベルの変数に対してしか、default による曖昧解決はしないようだった。

fy.hs :

```
f [] = 0
```

```
f (x:xs) = 1 + f xs
```

```
x = f "abcde"
```

```
main =
```

```
  let x0 = f "abcd"
```

```
  in
```

```
    putStrLn $ show x0
```

これを `-ddump-assump` 付きでコンパイルした結果の一部：

- `Main.x :: Prelude.Integer`
- `Main.13.10.x0 :: forall [Star] ([Prelude.Num v132] :=> v132)`

おぼろげに思い出してきたのだが、総称型変数 (`Gen i`) に対する置換は、ないんじゃないかな。

あきらめて (?), 腰を据えて、コード読み直そう。DictPass.hs からかな。

このブランチ (i013) は、当面このまま。この issue (013) を閉じるときに消す。

2020-04-15 (Wed)

当初、008 などの修正には、本件でのべたような変更が必要、または、有効なのではないかと思ったのだが、そうではなかった。

却下