094: Subst 高速化

↑ up

- issued: 2020-05-24
- 分類： 分類：C 改善項目
- status: Open

概要

いま最も消費時間の多い Subst の高速化。@@ をみるに、map のあとに少ない要素を追加することが多い。Data.Map.Strict は逆効果だったが、Vector がはまるか？

調査ログ

2020-05-25 (Mon)

まず、取り掛かるまえの実行時間（time ./tcompile testcases/sample152mod.hs): 36.5s

Data.Vector にしたバージョンは、主記憶不足で page swap のスラッシングが起きているような感じで、マウスがうごかなくなるような状態に。ぜんぜんだめ。

それならと Data.Sequence をつかってみたが、48.7s で悪化。

List にもどし、リスト内容式でかかれている @@ の実装を map に書き換えてみるなどしたが、これも 42s と悪化。もとの実装が一番はやかった。

Subst を大幅に速くするのは難しいのかも。いま、DictPass でやたら Subst が肥大化して遅くなっているのを改める（extSubst する回数を抑制する）必要があるのかもしれない。

たしかに、不要な Subst をいっぱいこしらえている気がする。主要な項（名前付き項）の型付けは済んでいるのだから、ローカルな項の型チェックにもちいた置換を、状態にため込みつづける必要はないのかもしれない。

継続

2020-05-26 (Tue)

複数の bind 間で Subst を共有する（すると、Subst は大きく肥大化する）必要はなかったので、これをやめた：

```
compiler/src/DictPass.hs
@@ -50,9 +50,10 @@ tcBind (Rec bs) ce maybest = tbloop st0 bs []
            pss' = tcPss st
            st' = st{tcPss=(pss++pss')}
            (e', st'') = runState (tcExpr e qt) st'
```

項目	lib/Prelude.hs	sample152.hs
修正前	5.430s (100%)	29.424s (100%)
修正後	1.263s (430%)	6.282s (468%)

```
          num = tcNum st''
      in
        if null qs then ((v, e'), st''{tcPss=pss'})
        else ((v, Lam (mkVs n qs) e'), st''{tcPss=pss'})
        if null qs then ((v, e'), st{tcNum=num})
        else ((v, Lam (mkVs n qs) e'), st{tcNum=num})
  tcbind _ _ = error "tcbind: must not occur."
```

この修正で、４倍以上速くなった：