

```

(*****
(* HaskWeak: a non-dependently-typed but Coq-friendly version of HaskCore *)
(*****)

```

```

Generalizable All Variables.
Require Import Preamble.
Require Import General.
Require Import Coq.Lists.List.
Require Import HaskKinds.
Require Import HaskLiteralsAndTyCons.
Require Import HaskWeakVars.
Require Import HaskWeakTypes.

```

```

Inductive WeakAltCon :=
| WeakDataAlt : CoreDataCon -> WeakAltCon
| WeakLitAlt : HaskLiteral -> WeakAltCon
| WeakDEFAULT : WeakAltCon.

```

```

Inductive WeakExpr :=
| WEVar      : WeakExprVar                    -> WeakExpr
| WELit      : HaskLiteral                    -> WeakExpr
| WELet      : WeakExprVar -> WeakExpr        -> WeakExpr
| WELetRec   : Tree ??(WeakExprVar * WeakExpr) -> WeakExpr
| WECast     : WeakExpr                       -> WeakCoercion -> WeakExpr
| WENote     : Note                          -> WeakExpr      -> WeakExpr
| WEApp      : WeakExpr                       -> WeakExpr      -> WeakExpr
| WETyApp    : WeakExpr                       -> WeakType      -> WeakExpr
| WECoApp    : WeakExpr                       -> WeakCoercion -> WeakExpr
| WELam      : WeakExprVar                    -> WeakExpr      -> WeakExpr
| WETyLam    : WeakTypeVar                    -> WeakExpr      -> WeakExpr
| WECoLam    : WeakCoerVar                    -> WeakExpr      -> WeakExpr

```

```

(* The WeakType argument in WEBrak/WEEsc is used only when going back *)
(* from Weak to Core; it lets us dodge a possibly-failing type      *)
(* calculation. The CoreVar argument is the GlobalVar for the hetmet_brak *)
(* or hetmet_esc identifier                                           *)

```

```

| WEBrak     : WeakExprVar -> WeakTypeVar      -> WeakExpr -> WeakType -> WeakExpr
| WEEsc      : WeakExprVar -> WeakTypeVar      -> WeakExpr -> WeakType -> WeakExpr
| WECSP      : WeakExprVar -> WeakTypeVar      -> WeakExpr -> WeakType -> WeakExpr

| WECase     : forall (vsclut:WeakExprVar)

```

```
(scrutinee:WeakExpr)
(tbranches:WeakType)
(tc:TyCon)
(type_params:list WeakType)
(alts : Tree ??(WeakAltCon*list WeakTypeVar*list WeakCoerVar*list WeakExprVar*WeakExpr)),
WeakExpr.
```

```
Definition weakTypeOfLiteral (lit:HaskLiteral) : WeakType :=
  (WTyCon (haskLiteralToTyCon lit)).
```