Manual de Hue

Nicolas Cabral

30 de enero de 2017

Resumen

The abstract text goes here.

1. Hue

Es un asistente de pruebas basados en el Calculo de Construccions λC

2. PTS: Pure Type System

Definición 2.1 (PTS)**.** Un PTS $\mathcal{P}$ esta definido por $\{\mathcal{S}, \mathcal{V}, \mathcal{A}, \mathcal{R}\}$ donde

- $\mathcal{S}$ es un conjunto sorts
- $\mathcal{V}$ es un conjunto de variables
- $\mathcal{A}$ es un conjunto no vacio de $\mathcal{S} \times \mathcal{S}$ llamados axiomas
- $\mathcal{R}$ es un conjunto de $\mathcal{S} \times \mathcal{S} \times \mathcal{S}$ llamadas $\Pi - Reglas$

Definición 2.2 (Pseudotérminos)**.** El conjunto T de pseudoterminos de un PTS $\mathcal{P} = \{\mathcal{S}, \mathcal{V}, \mathcal{A}, \mathcal{R}\}$ es el menor conjunto que satisface lo siguiente:

- $\mathcal{S} \cup \mathcal{V} \subset \mathcal{T}$
- Si $a \in \mathcal{T}$ y $b \in \mathcal{T}$ entonces $ab \in \mathcal{T}$
- Si $A \in \mathcal{T}$, $B \in \mathcal{T}$ y $x \in \mathcal{V}$ entonces $(\lambda x : A.B) \in \mathcal{T}$
- Si $A \in \mathcal{T}$, $B \in \mathcal{T}$ y $x \in \mathcal{V}$ entonces $(\Pi x : A.B) \in \mathcal{T}$

Definición 2.3 (PTS funcional)**.** Un PTS se dice *funcional* si

- $\langle s_1, s\rangle \in \mathcal{A}$ y $\langle s_1, s'\rangle \in \mathcal{A}$ implica $s = s'$;
- $\langle s_1, s_2, s\rangle \in \mathcal{R}$ y $\langle s_1, s_2, s'\rangle \in \mathcal{R}$ implica $s = s'$.

Definición 2.4 (PTS full)**.** Un PTS se dice *full* si para todo $s_1, s_2 \in \mathcal{S}$ existe un $s_3 \in \mathcal{S}$ con $\langle s_1, s_2, s_3\rangle \in \mathcal{R}$.

Definición 2.5 (PTS semifull)**.** Un PTS se dice *semi-full* si para todo $s_1 \in \mathcal{S}$ $\exists s_2, s_3[\langle s_1, s_2, s_3\rangle \in \mathcal{R}] \Rightarrow \forall s_2 \exists s_3[\langle s_1, s_2, s_3\rangle \in \mathcal{R}]$

Lema 2.1. PTS full $\Rightarrow$ PTS semi full ??
PTS funcional $\Rightarrow$ PTS full o semi full??

Definición 2.6 (Relación $\vdash$)**.** Definimos a la relacion $\vdash$, con $\vdash \subseteq \mathcal{C} \times \mathcal{T} \times \mathcal{T}$ como la menor relación que cumple:

$$(Srt) \qquad \frac{}{\emptyset \vdash s_1 : s_2} \qquad (s_1, s_2) \in \mathcal{A}$$

$$(Var) \qquad \frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}$$

$$(Wk) \qquad \frac{\Gamma \vdash b : B \quad \Gamma \vdash A : s}{\Gamma, x : A \vdash b : B} \qquad b \in \mathcal{S} \cup \mathcal{V}$$

$$(Pi) \qquad \frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash \Pi x : A.B : s_3} \qquad (s_1, s_2, s_3) \in \mathcal{R}$$

$$(Lda) \qquad \frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash b : B \quad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash \lambda a : A.b : \Pi x : A.B} \qquad (s_1, s_2, s_3) \in \mathcal{R}$$

$$(App) \qquad \frac{\Gamma \vdash a : \Pi x : B.A \quad \Gamma \vdash b : B}{\Gamma \vdash ab : A[x := b]}$$

$$(Cnv) \qquad \frac{\Gamma \vdash a : A \quad \Gamma \vdash B : s}{\Gamma \vdash a : B} \qquad A \simeq B$$

2.1. Ejemplos

Sea $\mathcal{S} = \{\mathsf{Prop}, \mathsf{Type}\}$ y $\mathcal{A} = \{(\mathsf{Prop} : \mathsf{Type})\}$ tenemos que

- $\lambda^{\rightarrow}$ es el PTS con $(\mathcal{S}, \mathcal{V}, \mathcal{A}, \mathcal{R}_1)$
- $\lambda\Pi$ es el PTS con $(\mathcal{S}, \mathcal{V}, \mathcal{A}, \mathcal{R}_2)$
- λC es el PTS con $(\mathcal{S}, \mathcal{V}, \mathcal{A}, \mathcal{R}_3)$

2.2. Optimización

Podemos introducir un modificación a $\vdash$ orientada a la implementación. En lugar de tener que usar Wk para obtener un contexto valido podemos asumir que el mismo es valido. De esta forma las reglas $\vdash_{vtyp}$ se definiría de la siguiente manera:

Definición 2.7 (Relación $\vdash_{vtyp}$)**.** Definimos a la relacion $\vdash_{vtyp}$, con $\vdash_{vtyp} \subseteq \mathcal{C} \times \mathcal{T} \times \mathcal{T}$ como la menor relación que cumple:

$$(Srt - vtyp) \qquad \frac{}{\Gamma \vdash_{vtyp} s_1 : s_2} \qquad (s_1, s_2) \in \mathcal{A}$$

$$(Var - vtyp) \qquad \frac{}{\Gamma \vdash_{vtyp} x : A} \qquad x : A \in \Gamma$$

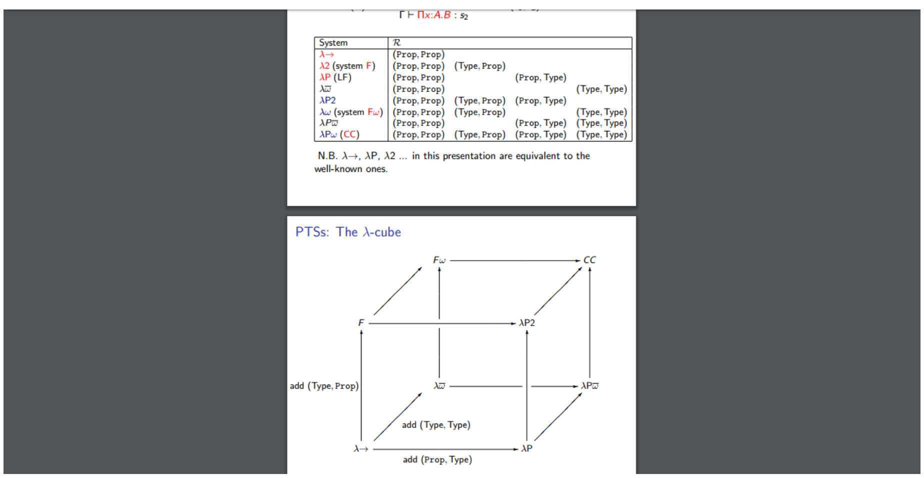
$$(Pi - vtyp) \qquad \frac{\Gamma \vdash_{vtyp} A : s_1 \quad \Gamma, x : A \vdash_{vtyp} B : s_2}{\Gamma \vdash_{vtyp} \Pi x : A.B : s_3} \qquad \begin{array}{r} (s_1, s_2, s_3) \in \mathcal{R} \\ x \notin dom(\Gamma) \end{array}$$

$$(Lda - vtyp) \qquad \frac{\Gamma \vdash_{vtyp} A : s_1 \quad \Gamma, x : A \vdash_{vtyp} b : B \quad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash_{vtyp} \lambda a : A.b : \Pi x : A.B} \qquad \begin{array}{r} (s_1, s_2, s_3) \in \mathcal{R} \\ x \notin dom(\Gamma) \end{array}$$

$$(App - vtyp) \qquad \frac{\Gamma \vdash_{vtyp} a : \Pi x : B.A \quad \Gamma \vdash_{vtyp} b : B}{\Gamma \vdash_{vtyp} ab : A[x := b]}$$

$$(Cnv - vtyp) \qquad \frac{\Gamma \vdash_{vtyp} a : A \quad \Gamma \vdash_{vtyp} B : s}{\Gamma \vdash_{vtyp} a : B} \qquad A \simeq B$$

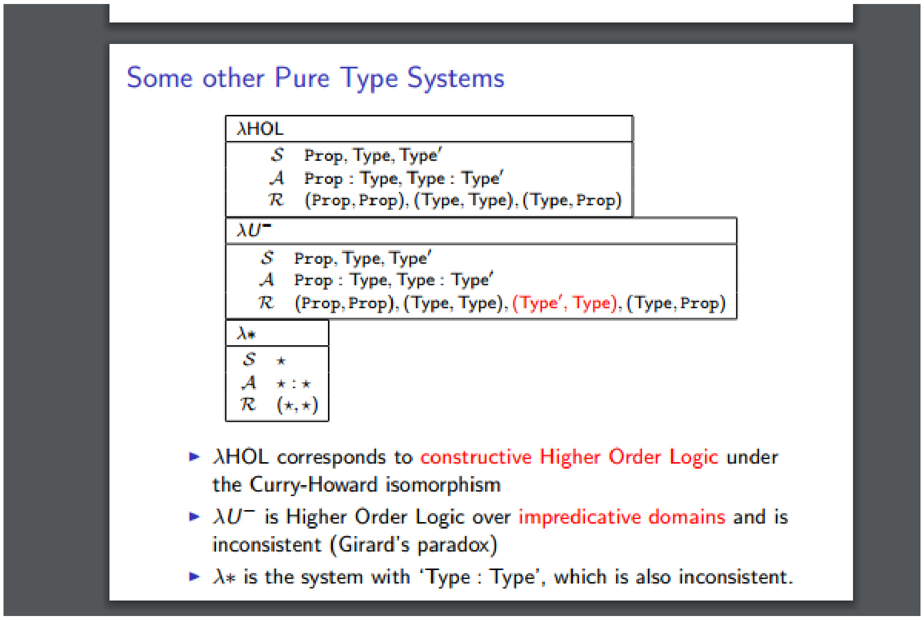
PASAR A LATEX



Definición 2.8 (Relación $\vdash_{vctx}$)**.** Definimos a la relacion $\vdash_{vctx}$, con el menor predicado sobre $\mathcal{C}$ que satisface:

$$(Nil - vctx) \qquad \frac{}{\emptyset \vdash_{vctx}}$$

$$(Cons - vctx) \qquad \frac{\Gamma \vdash_{vctx} \quad \Gamma \vdash_{vtyp} A : s}{\Gamma, x : A \vdash_{vctx}}$$



Lema 2.2. Equivalencia de $\vdash$ y $\vdash_{vtyp}$

$$\Gamma \vdash a : A \Leftrightarrow \Gamma \vdash_{vctx} \wedge \Gamma \vdash_{vtyp} a : A \tag{1}$$

3. PTS Semifull

Definición 3.1 ($\vdash_s f$)**.** $\vdash_s f \subseteq \mathcal{C} \times \mathcal{T} \times \mathcal{T}$ es la menor relación que satisface las relgas de $\vdash$ pero se reemplaza la regla lda por

$$(Lda1 - sf) \qquad \frac{\Gamma \vdash_{sf} A : s_1 \quad \Gamma x : A \vdash_{sf} b : B}{\Gamma \vdash_{sf} \lambda x : A.b : \Pi x : A.B} \qquad \begin{array}{r} \langle s_1, s_2, s_3 \rangle \in \mathcal{R} \\ B \in \mathcal{S} \end{array}$$

$$(Lda2 - sf) \qquad \frac{\Gamma \vdash_{sf} A : s_1 \quad \Gamma x : A \vdash_{sf} b : s_b}{\Gamma \vdash_{sf} \lambda x : A.b : \Pi x : A.s_b} \qquad \begin{array}{r} \langle s_1, s_2, s_3 \rangle \in \mathcal{R} \\ \langle s_b, s_4 \rangle \in \mathcal{A} \end{array}$$

Lema 3.1. Equivalencia entre $\vdash$ y $\vdash_{sf}$ Dado un PTS semifull tenemos que

$$\Gamma \vdash a : A \Leftrightarrow \Gamma \vdash_{sf} a : A \tag{2}$$

Definición 3.2 ($\vdash_s fnsd$)**.** Usando $\vdash_{sf}$ podemos de definir una nueva relación casi dirigida por sintaxis, $\vdash_{sfnsd} \subseteq \mathcal{C} \times \mathcal{T} \times \mathcal{T}$ es la menor relacion que satisface:

$$(Srt - sfnsd) \qquad \frac{}{\emptyset \vdash_{sfnsd} s_1 : s_2} \qquad (s_1, s_2) \in \mathcal{A}$$

$$(Var - sfnsd) \qquad \frac{\Gamma \vdash_{sfnsd} A :\twoheadrightarrow s}{\Gamma, x : A \vdash_{sfnsd} x : A}$$

$$(Wk - sfnsd) \qquad \frac{\Gamma \vdash_{sfnsd} b : B \quad \Gamma \vdash_{sfnsd} A :\twoheadrightarrow s}{\Gamma, x : A \vdash_{sfnsd} b : B} \qquad b \in \mathcal{S} \cup \mathcal{V}$$

$$(Pi - sfnsd) \qquad \frac{\Gamma \vdash_{sfnsd} A :\twoheadrightarrow s_1 \quad \Gamma, x : A \vdash_{sfnsd} B :\twoheadrightarrow s_2}{\Gamma \vdash_{sfnsd} \Pi x : A.B : s_3} \qquad (s_1, s_2, s_3) \in \mathcal{R}$$

$$(Lda1 - sfnsd) \qquad \frac{\Gamma \vdash_{sfnsd} A :\twoheadrightarrow s_1 \quad \Gamma x : A \vdash_{sfnsd} b : B}{\Gamma \vdash_{sfnsd} \lambda x : A.b : \Pi x : A.B} \qquad \begin{array}{r} \langle s_1, s_2, s_3 \rangle \in \mathcal{R} \\ B \in \mathcal{S} \end{array}$$

$$(Lda2 - sfnsd) \qquad \frac{\Gamma \vdash_{sfnsd} A :\twoheadrightarrow s_1 \quad \Gamma x : A \vdash_{sfnsd} b : s_b}{\Gamma \vdash_{sfnsd} \lambda x : A.b : \Pi x : A.s_b} \qquad \begin{array}{r} \langle s_1, s_2, s_3 \rangle \in \mathcal{R} \\ \langle s_b, s_4 \rangle \in \mathcal{A} \end{array}$$

$$(App - sfnsd) \qquad \frac{\Gamma \vdash_{sfnsd} a :\twoheadrightarrow^{wh} \Pi x : B_0.A \quad \Gamma \vdash_{sfnsd} b : B_1}{\Gamma \vdash_{sfnsd} ab : A[x := b]} \qquad B_0 \simeq B_1$$

Lema 3.2. Equivalencia entre $\vdash_{sf}$ y $\vdash_{sfnsd}$ Dado un PTS semifull tenemos que

- Si $\Gamma \vdash_{sfnsd} a : A$ entonces $\Gamma \vdash_{sf} a : A$
- Si $\Gamma \vdash_{sf} a : A$ entonces $\exists A_0 | \Gamma \vdash_{sfnsd} a : A_0 \wedge A_0 \simeq A$

Lema 3.3. Si tenemos un PTS semifull y funcional entonces $\vdash_{sfnsd}$ son reglas dirigidas por sintaxis

4. Teoría del calculo de λC

Definición 4.1. λC es un PTS definido por $\{\mathcal{S}, \mathcal{V}, \mathcal{A}, \mathcal{R}\}$ tales que:

- $\mathcal{S} = \{\mathsf{Prop}, \mathsf{Type}\}$
- $\mathcal{V} = \dots$
- $\mathcal{A} = \{(\mathsf{Prop}, \mathsf{Type})\}$
- $\mathcal{R} = \{(\mathsf{Prop}, \mathsf{Prop}, \mathsf{Prop}), (\mathsf{Prop}, \mathsf{Type}, \mathsf{Type}), (\mathsf{Type}, \mathsf{Prop}, \mathsf{Prop}), (\mathsf{Type}, \mathsf{Type}, \mathsf{Type})\}$

Lema 4.1. λC es un PTS *full.*

Demostración. □

5. Hue: Typechecker

Definición 5.1. Los terminos del λ-calculo esta definidos por:

```
data CoCT =  IdT Int
           | VarT String
           | LamT CoCT CoCT
           | AppT CoCT CoCT
           | PiT CoCT CoCT
           | PropT
           | TypeT deriving (Eq)

type Type = CoCT
type Term = CoCT
```

Definición 5.2. Los contextos en Hue estan definidos de la siguiente manera:

```
type Context = [ (String, (Maybe Term, Type)) ]
```

Definición 5.3. Definimos *whnf* como el operador que reduce un termino a su forma normal a la cabeza.

```
whnf :: Context -> CoCT -> NTerm
whnf c (IdT n)    = NAtom (AId n)
whnf c (VarT s)   = maybe (NAtom (AVar s)) (whnf c) (getDef c s)
whnf c (PiT s t)  = NAtom (APi s t)
whnf c PropT      = NAtom (AProp)
whnf c TypeT      = NAtom (AType)
whnf c (LamT s t) = NLam s t
whnf c (AppT m n) = case whnf c m of
            NLam _ t -> whnf c (app t n)
            NAtom t  -> NAtom (AApp t n)
```

Definición 5.4. Definimos la relacion $A \simeq_{\delta\beta} B$ de la siguiente manera

```
conv :: Context -> CoCT -> CoCT -> Bool
conv c s t = convT c (whnf c s) (whnf c t)
```

Definición 5.5. Definimos el operador $\twoheadrightarrow^* \Pi x : A.B$ de la siguiente manera

```
getPi :: Context -> CoCT -> Maybe (CoCT,CoCT)
getPi c t = case whnf c t of
              NAtom (APi m n) -> Just (m, n)
              _       -> Nothing
```

Definimos el operador $\twoheadrightarrow^* s$ donde $s \in \mathcal{S}$ de la siguiente manera

```
getSort :: Context -> CoCT -> Maybe CoCT
getSort c t = case whnf c t of
                NAtom AProp -> Just PropT
                NAtom AType -> Just TypeT
                _           -> Nothing
```

Definición 5.6 (Relación $\vdash_{hue}$)**.** Definimos a la relacion $\vdash_{hue}$, con $\vdash_{hue} \subseteq \mathcal{C} \times \mathcal{T} \times \mathcal{T}$ como la menor relación que cumple:

$$(Srt-hue) \qquad \frac{}{\Gamma \vdash_{hue} s_1 : s_2} \qquad (s_1, s_2) \in \mathcal{A}$$

$$(Var-hue) \qquad \frac{}{\Gamma \vdash_{hue} x : A} \qquad x : A \in \Gamma$$

$$(Pi-hue) \qquad \frac{\Gamma \vdash_{hue} A :\twoheadrightarrow^* s_1 \quad \Gamma, x : A \vdash_{hue} B :\twoheadrightarrow^* s_2}{\Gamma \vdash_{hue} \Pi x : A.B : s_3} \qquad (s_1, s_2, s_3) \in \mathcal{R}$$

$$(Lda-hue) \qquad \frac{\Gamma \vdash_{vtyp} A :\twoheadrightarrow^* s_1 \quad \Gamma, x : A \vdash_{vtyp} b : B}{\Gamma \vdash_{vtyp} \lambda a : A.b : \Pi x : A.B} \qquad \begin{array}{c} x \notin dom(\Gamma) \\ B \notin \mathcal{S} \vee \mathcal{A}(B) \end{array}$$

$$(App-hue) \qquad \frac{\Gamma \vdash_{hue} a :\twoheadrightarrow^* \Pi x : B.A \quad \Gamma \vdash_{hue} b : B'}{\Gamma \vdash_{hue} ab : A[x := b]} \qquad B \simeq B'$$

Lema 5.1. Equivalencia de $\vdash$ y $\vdash_{hue}$ Sea $\vdash$ la relación del PTS asociado a λC entonces:

$$\Gamma \vdash a : A \Leftrightarrow \Gamma \vdash_{vctx} \wedge \Gamma \vdash_{hue} a : A \tag{3}$$

Demostración. chan! □

5.1. Terminos

5.2. Reduccion

5.3. Contextos

5.4. Typechecker

[3] [1] [2]

$$\frac{}{\vdash s_1 : s_2}\ \text{Srtnsd} \tag{4}$$

$$\frac{\Gamma \vdash A :\twoheadrightarrow s}{\Gamma, x : A \vdash_{nsd} x : A}\ \text{Varnsd} \tag{5}$$

$$\frac{\Gamma \vdash_{nsd} b : B \quad \Gamma \vdash A :\twoheadrightarrow s}{\Gamma, x : A \vdash_{nsd} b : B}\ \text{Wknsd} \tag{6}$$

$$\frac{\Gamma \vdash A :\twoheadrightarrow s_1 \quad \Gamma, x : A \vdash_{nsd} B :\twoheadrightarrow s_2}{\Gamma \vdash_{nsd} \Pi x : A.B : s_3}\ \text{Pinsd} \tag{7}$$

$$\frac{\Gamma \vdash A :\twoheadrightarrow s_1 \quad \Gamma, x : A \vdash_{nsd} b :\twoheadrightarrow B \quad \Gamma, x : A \vdash_{nsd} B :\twoheadrightarrow s_2}{\Gamma \vdash_{nsd} \lambda x : A.b : \Pi x : A.B}\ \text{Ldansd} \tag{8}$$

$$\frac{\Gamma \vdash a :\twoheadrightarrow \Pi x : A.B \quad \Gamma \vdash_{nsd} b :\twoheadrightarrow B}{\Gamma \vdash_{nsd} a\ b : A[x := b]}\ \text{Appnsd} \tag{9}$$

6. Conclusion

Write your conclusion here.

Referencias

[1] L. S. van Benthem Jutting, J. McKinna, and R. Pollack. Checking algorithms for pure type systems. In H. Barendregt and T. Nipkow, editors, *Types for Proofs and Programs*, pages 19–61. Springer, Berlin, Heidelberg, 1993.

[2] L. S. van Benthem Jutting, James McKinna, and Robert Pollack. Checking algorithms for pure type systems. In Henk Barendregt and Tobias Nipkow, editors, *Types for Proofs and Programs, International Workshop TYPES'93, Nijmegen, The Netherlands, May 24-28, 1993, Selected Papers*, volume 806 of *Lecture Notes in Computer Science*, pages 19–61. Springer, 1993.

[3] L.S. van Benthem Jutting, J. McKinna, and R. Pollack. Checking algorithms for pure type systems, 1993.