# 1 Proof Rules and Proof Expressions

Each rule has a name that is the outer operator of the proof expression with slots to be filled in as the proof is developed. The partial proofs are organized as a tree generated in two passes. The first pass is *top down*, driven by the user creating terms with slots to be filled in on an algorithmic bottom up pass.

## 1.1 First-order refinement style proof rules over domain of discourse $D$

### Minimal Logic

#### Construction rules

- **And Construction**
  $H \vdash A\&B$ *by* $pair(slot_a; slot_b)$
  $H \vdash A$ *by* $slot_a$
  $H \vdash B$ *by* $slot_b$

- **Exists Construction**[a]
  $H \vdash \exists x.B(x)$ *by* $pair(d; slot_b(d))$
  $H \vdash d \in D$ *by* $obj(d)$
  $H \vdash B(d)$ *by* $slot_b(d)$

  [a] Also see Alternative Rules below.

- **Implication Construction**
  $H \vdash A \Rightarrow B$ *by* $\lambda(x.slot_b(x))$ *new* $x$
  $H,\ x : A \vdash B$ *by* $slot_b(x)$

- **All Construction**
  $H \vdash \forall x.B(x)$ *by* $\lambda(x.slot_b(x))$ *new* $x$
  $H,\ x : D \vdash B(x)$ *by* $slot_b(x)$

- **Or Construction - left**
  $H \vdash A \vee B$ *by* $inl(slot_l)$
  $H \vdash A$ *by* $slot_l$

- **Or Construction - right**
  $H \vdash A \vee B$ *by* $inr(slot_r)$
  $H \vdash B$ *by* $slot_r$

#### Decomposition rules

- **And Decomposition**

  $H, x : A\&B, H' \vdash G$ *by* $spread(x; l, r.slot_g(l, r))$ *new* $l, r$
  $H, l : A, r : B, H' \vdash G$ *by* $slot_g(l, r)$

- **Exists Decomposition**

  $H, x : \exists y.B(y), H' \vdash G$ *by* $spread(x; d, r.slot_g(d, r))$ *new* $d, r$
  $H, d : D, r : B(d), H' \vdash G$ *by* $slot_g(d, r)$

- **Implication Decomposition**

  $H, f : A \Rightarrow B, H' \vdash G$ *by* $apseq(f; slot_a; v.slot_g[ap(f; slot_a)/v])$ *new* $v$[1]
  $H, f : A \Rightarrow B, H' \vdash A$ *by* $slot_a$
  $H, f : A \Rightarrow B, H', v : B \vdash G$ *by* $slot_g(v)$

- **All Decomposition**

  $H, f : \forall x.B(x), H' \vdash G$ *by* $apseq(f; d; v.slot_g[ap(f; d)/v])$
  $H, f : \forall x.B(x), H' \vdash d \in D$ *by* $obj(d)$
  $H, f : \forall x.B(x), H', v : B(d) \vdash G$ *by* $slot_g(v)$[2]

[1] This notation shows that $ap(f; slot_a)$ is substituted for $v$ in $g(v)$. In the CTT logic we stipulate in the rule that $v = ap(f; slot_a)$ *in* $B$.

[2] In the CTT logic, we use equality to stipulate that $v = ap(f; d)$ *in* $B(v)$ just before the hypothesis $v : B(d)$.

- **Or Decomposition**

  $H, y : A \vee B, H' \vdash G$ *by* $decide(y; l.leftslot(l); r.rightslot(r))$

  1. $H, l : A, H' \vdash G$ *by* $leftslot(l)$
  2. $H, r : B, H' \vdash G$ *by* $rightslot(r)$

- **Hypothesis - domain (D)**
  $H, d : D, H' \vdash\ d \in D$ *by* $obj(d)$

- **Hypothesis - formula (A)**
  $H, x : A, H' \vdash\ A$ *by* $hyp(x)$
  We usually abbreviate the justifications to *by* $d$ and *by* $x$ respectively.

**Intuitionistic Rules**

- **False Decomposition**

  $H, f : False, H' \vdash G$ *by* $any(f)$

  This is the rule that distinguishes intuitionistic from minimal logic, called "ex falso quodlibet". We use the constant $False$ for intuitionistic formulas and $\perp$ for minimal ones to distinguish the logics. In practice, we would use only one constant, say $\perp$, and simply add the above rule with $\perp$ for $False$ to axiomatize iFOL. However, for our results it is especially important to be clear about the difference, so we use both notations.

Note that we use the term $d$ to denote objects in the domain of discourse $D$. In the classical evidence semantics, we assume that $D$ is non-empty by postulating the existence of some $d_0$ in it. Also note that in the rule for $False$ Decomposition, it is important to use the $any(f)$ term which allows us to thread the explanation for how $False$ was derived into the justification for $G$.

**Structural Rules**

- **Cut rule**

  $H \vdash G$ *by* $CutC(x.slot_g(slot_c))$ *new* $x$
  1. $H, x : C \vdash G$ *by* $slot_g(x)$
  2. $H \vdash C$ *by* $slot_c$.

**Classical Rules**

- **Non-empty Domain of Discourse**

  $H \vdash d_0 \in D$ *by* $obj(d_0)$

- **Law of Excluded Middle (LEM)** Define $\sim A$ as $(A \Rightarrow False)$
  $H \vdash (A \vee \sim A)$ *by* **magic**$(A)$
  Note that this is the only rule that mentions a formula in the rule name.

**Alterative Rule**

- **Exists Construction**

  $H\ \vdash\ \exists x.B(x)$ *by* $pair(slot_d/X; slot_b[slot_d/X])$
  $H\ \vdash D$ *by* $slot_d$
  $H\ \vdash B(X)$ *by* $slot_b(X)$
  Note, the substitution of $slot_d$ propagates to $B(X)$ as soon as the first subgoal determines the value of the slot for the goal rule. The term $X$ acts as a *logic variable*.