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**
  \[
  H \vdash A \& B by \text{pair}(\text{slot}_a; \text{slot}_b)
  \]
  \[
  H \vdash A by \text{slot}_a
  \]
  \[
  H \vdash B by \text{slot}_b
  \]

- **Exists Construction\(^a\)**
  \[
  H \vdash \exists x.B(x) by \text{pair}(d; \text{slot}_b(d))
  \]
  \[
  H \vdash d \in D by \text{obj}(d)
  \]
  \[
  H \vdash B(d) by \text{slot}_b(d)
  \]

\(^a\)Also see Alternative Rules below.

- **Implication Construction**
  \[
  H \vdash A \Rightarrow B by \lambda(x.\text{slot}_b(x)) new x
  \]
  \[
  H, x : A \vdash B by \text{slot}_b(x)
  \]

- **All Construction**
  \[
  H \vdash \forall x.B(x) by \lambda(x.\text{slot}_b(x)) new x
  \]
  \[
  H, x : D \vdash B(x) by \text{slot}_b(x)
  \]

- **Or Construction - left**
  \[
  H \vdash A \lor B by \text{inl}(\text{slot}_l)
  \]
  \[
  H \vdash A by \text{slot}_l
  \]

- **Or Construction - right**
  \[
  H \vdash A \lor B by \text{inr}(\text{slot}_r)
  \]
  \[
  H \vdash B by \text{slot}_r
  \]

**Decomposition rules**

- **And Decomposition**
  \[
  H, x : A \& B, H' \vdash G by \text{spread}(x; l, r.\text{slot}_g(l, r)) new l, r
  \]
  \[
  H, l : A, r : B, H' \vdash G by \text{slot}_g(l, r)
  \]

- **Exists Decomposition**
  \[
  H, x : \exists y.B(y), H' \vdash G by \text{spread}(x; d, r.\text{slot}_g(d, r)) new d, r
  \]
  \[
  H, d : D, r : B(d), H' \vdash G by \text{slot}_g(d, r)
  \]

- **Implication Decomposition**
  \[
  H, f : A \Rightarrow B, H' \vdash G by \text{apseq}(f; \text{slot}_a; v.\text{slot}_g[ap(f; \text{slot}_a)/v]) new v\(^1\)
  \]
  \[
  H, f : A \Rightarrow B, H' \vdash A by \text{slot}_a
  \]
  \[
  H, f : A \Rightarrow B, H', v : B \vdash G by \text{slot}_g(v)
  \]

- **All Decomposition**
  \[
  H, f : \forall x.B(x), H' \vdash G by \text{apseq}(f; d; v.\text{slot}_g[ap(f; d)/v])
  \]
  \[
  H, f : \forall x.B(x), H' \vdash d \in D by \text{obj}(d)
  \]
  \[
  H, f : \forall x.B(x), H', v : B(d) \vdash G by \text{slot}_g(v)\(^2\)
  \]

\(^1\)This notation shows that \(ap(f; \text{slot}_a)\) is substituted for \(v\) in \(g(v)\). In the CTT logic we stipulate in the rule that \(v = ap(f; \text{slot}_a)\) 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
\[ H, y : A \vee B, H' \vdash G \text{ by decide}(y; l.\leftslot(l); r.\rightslot(r)) \]
1. \( H, l : A, H' \vdash G \text{ by leftslot}(l) \)
2. \( H, r : B, H' \vdash G \text{ by rightslot}(r) \)

• Hypothesis - domain (D)
\[ H, d : D, H' \vdash d \in D \text{ by obj}(d) \]
• Hypothesis - formula (A)
\[ H, x : A, H' \vdash A \text{ by hyp}(x) \]
We usually abbreviate the justifications to by \( d \) and by \( x \) respectively.

Intuitionistic Rules
• False Decomposition
\[ H, f : \text{False}, H' \vdash G \text{ by any}(f) \]
This is the rule that distinguishes intuitionistic from minimal logic, called “ex falso quodlibet”. We use the constant \( \text{False} \) for intuitionistic formulas and \( \bot \) for minimal ones to distinguish the logics. In practice, we would use only one constant, say \( \bot \), and simply add the above rule with \( \bot \) for \( \text{False} \) 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 \( \text{any}(f) \) term which allows us to thread the explanation for how False was derived into the justification for \( G \).

Structural Rules
• Cut rule
\[ H \vdash G \text{ by Cut}(C) \]
1. \( H, x : C \vdash G \text{ by slot}(x) \)
2. \( H \vdash C \text{ by slot}(c) \).

Classical Rules
• Non-empty Domain of Discourse
\[ H \vdash d_0 \in D \text{ by obj}(d_0) \]
• Law of Excluded Middle (LEM) Define \( \sim A \) as \( (A \Rightarrow \text{False}) \)
\[ H \vdash (A \lor \sim A) \text{ by magic}(A) \]
Note that this is the only rule that mentions a formula in the rule name.

Alternative Rule
• Exists Construction
\[ H \vdash \exists x.B(x) \text{ by pair}(\text{slot}_d/X; \text{slot}_b[\text{slot}_d/X]) \]
\[ H \vdash D \text{ by slot}_d \]
\[ H \vdash B(X) \text{ by slot}_b(X) \]
Note, the substitution of \( \text{slot}_d \) 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.