Metatheory of the Logic of Hereditary Harrop Formulas in Coq

Chelsea Battell

University of Ottawa

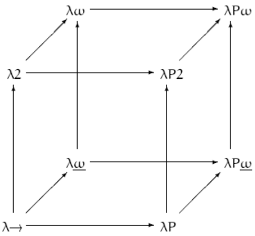
June 17, 2016

Coq

Interactive theorem prover developed at

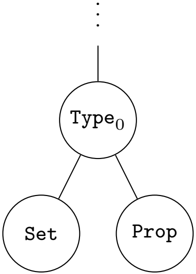
Inria

Implementation of extension of Calculus of Constructions (CoC) created by Thierry Coquand



← CoC

Calculus of Constructions



Notation: $A : B$ means A has type B

If $P : \texttt{Prop}$ and $t : P$, then P is a theorem and t is a proof of P

Example:

$$(\forall (n : \texttt{nat}), 0 + n = n) : \texttt{Prop}$$
$$(\texttt{fun}\ (n : \texttt{nat}) \Rightarrow \texttt{eq\_refl}) : \forall (n : \texttt{nat}), 0 + n = n$$

Inference Rule

$$\frac{P_1 \quad \cdots \quad P_n}{C} \; name$$

Vertical "implication" notation

Inference Rule

$$\frac{P_1 \quad \cdots \quad P_n}{C}\ name$$

Vertical “implication” notation

- If P_1, ..., P_n, then C

Inference Rule

$$\frac{P_1 \quad \cdots \quad P_n}{C}\ name$$

Vertical "implication" notation

- If P_1, ..., P_n, then C
- To prove C, show P_1, ..., P_n all true

Inference Rule

$$\frac{P_1 \quad \cdots \quad P_n}{C}\ name$$

Vertical "implication" notation

- If P_1, ..., P_n, then C
- To prove C, show P_1, ..., P_n all true
- If you can build P_1, ..., P_n, then you can build C

Natural Deduction

Set of inference rules to encode "natural" reasoning

e.g. $\dfrac{A \quad B}{A \wedge B} \wedge_I \qquad \dfrac{A \wedge B}{A} \wedge_{E_1} \qquad \dfrac{A \wedge B}{B} \wedge_{E_2}$

Claim: If $p \wedge q$, then $q \wedge p$

Proof:

$$\dfrac{\dfrac{p \wedge q}{q} \wedge_{E_2} \qquad \dfrac{p \wedge q}{p} \wedge_{E_1}}{q \wedge p} \wedge_I$$

Sequent Calculus

Sequent

$$\Gamma \vdash P$$

P is provable in context Γ (a set of assumptions)

Example Sequent Rule

$$\frac{\Gamma, P \vdash Q}{\Gamma \vdash P \to Q} \to_I$$

Prove "If P then Q" by assuming P and deriving Q

Sequents as Dependent Types in Coq

Goal-Reduction Sequent

$\texttt{grseq} : \texttt{context} \rightarrow \texttt{oo} \rightarrow \texttt{Prop}$

$\Gamma \rhd \beta$ is notation for $\texttt{grseq}\ \Gamma\ \beta$

Backchaining Sequent

$\texttt{bcseq} : \texttt{context} \rightarrow \texttt{oo} \rightarrow \texttt{atm} \rightarrow \texttt{Prop}$

$\Gamma, [\beta] \rhd \alpha$ is notation for $\texttt{bcseq}\ \Gamma\ \beta\ \alpha$

[Wang, Chaudhuri, Gacek, Nadathur; 2012]

Sequents as Dependent Types in Coq

Goal-Reduction Sequent

$\texttt{grseq} : \boxed{\texttt{context}} \to \texttt{oo} \to \texttt{Prop}$

$\Gamma \rhd \beta$ is notation for $\texttt{grseq}\ \Gamma\ \beta$

Backchaining Sequent

$\texttt{bcseq} : \boxed{\texttt{context}} \to \texttt{oo} \to \texttt{atm} \to \texttt{Prop}$

$\Gamma, [\beta] \rhd \alpha$ is notation for $\texttt{bcseq}\ \Gamma\ \beta\ \alpha$

[Wang, Chaudhuri, Gacek, Nadathur; 2012]

Sequents as Dependent Types in Coq

Goal-Reduction Sequent

$\texttt{grseq} : \texttt{context} \to \boxed{\texttt{oo}} \to \texttt{Prop}$

$\Gamma \rhd \beta$ is notation for $\texttt{grseq}\ \Gamma\ \beta$

Backchaining Sequent

$\texttt{bcseq} : \texttt{context} \to \boxed{\texttt{oo}} \to \texttt{atm} \to \texttt{Prop}$

$\Gamma, [\beta] \rhd \alpha$ is notation for $\texttt{bcseq}\ \Gamma\ \beta\ \alpha$

[Wang, Chaudhuri, Gacek, Nadathur; 2012]

Sequents as Dependent Types in Coq

Goal-Reduction Sequent

$\texttt{grseq} : \texttt{context} \rightarrow \texttt{oo} \rightarrow \texttt{Prop}$

$\Gamma \rhd \beta$ is notation for $\texttt{grseq}\ \Gamma\ \beta$

Backchaining Sequent

$\texttt{bcseq} : \texttt{context} \rightarrow \texttt{oo} \rightarrow \boxed{\texttt{atm}} \rightarrow \texttt{Prop}$

$\Gamma, [\beta] \rhd \alpha$ is notation for $\texttt{bcseq}\ \Gamma\ \beta\ \alpha$

[Wang, Chaudhuri, Gacek, Nadathur; 2012]

The Specification Logic (Hereditary Harrop)

Goal-Reduction Rules

$$\frac{A :- G \quad \Gamma \rhd G}{\Gamma \rhd \langle\, A \,\rangle}\ \text{g\_prog} \qquad \frac{D \in \Gamma \quad \Gamma, [D] \rhd A}{\Gamma \rhd \langle\, A \,\rangle}\ \text{g\_dyn} \qquad \frac{\Gamma \rhd G_1 \quad \Gamma \rhd G_2}{\Gamma \rhd G_1 \,\&\, G_2}\ \text{g\_and}$$

$$\frac{\Gamma\,,\, D \rhd G}{\Gamma \rhd D \longrightarrow G}\ \text{g\_imp} \qquad \frac{}{\Gamma \rhd \mathtt{T}}\ \text{g\_tt} \qquad \frac{\mathtt{proper}\ E \quad \Gamma \rhd G\ E}{\Gamma \rhd \mathtt{Some}\ G}\ \text{g\_some}$$

$$\frac{\forall (E : \mathtt{expr\ \ con}), (\mathtt{proper}\ E \to \Gamma \rhd G\ E)}{\Gamma \rhd \mathtt{All}\ G}\ \text{g\_all} \qquad \frac{\forall (E : \mathtt{X}), (\Gamma \rhd G\ E)}{\Gamma \rhd \mathtt{Allx}\ G}\ \text{g\_allx}$$

Backchaining Rules

$$\frac{}{\Gamma, [\langle\, A \,\rangle] \rhd A}\ \text{b\_match} \qquad \frac{\Gamma, [D_1] \rhd A}{\Gamma, [D_1 \,\&\, D_2] \rhd A}\ \text{b\_and1} \qquad \frac{\Gamma, [D_2] \rhd A}{\Gamma, [D_1 \,\&\, D_2] \rhd A}\ \text{b\_and2}$$

$$\frac{\Gamma \rhd G \quad \Gamma, [D] \rhd A}{\Gamma, [G \longrightarrow D] \rhd A}\ \text{b\_imp} \qquad \frac{\mathtt{proper}\ E \quad \Gamma, [D\ E] \rhd A}{\Gamma, [\mathtt{All}\ D] \rhd A}\ \text{b\_all} \qquad \frac{\Gamma, [D\ E] \rhd A}{\Gamma, [\mathtt{Allx}\ D] \rhd A}\ \text{b\_allx}$$

$$\frac{\forall (E : \mathtt{expr\ \ con}), (\mathtt{proper}\ E \to \Gamma, [D\ E] \rhd A)}{\Gamma, [\mathtt{Some}\ D] \rhd A}\ \text{b\_some}$$

The Specification Logic (Hereditary Harrop)

Goal-Reduction Rules

$$\frac{A \mathrel{:-} G \quad \Gamma \rhd G}{\Gamma \rhd \langle\, A \,\rangle}\ \text{g\_prog} \qquad \frac{D \in \Gamma \quad \Gamma, [D] \rhd A}{\Gamma \rhd \langle\, A \,\rangle}\ \text{g\_dyn} \qquad \boxed{\frac{\Gamma \rhd G_1 \quad \Gamma \rhd G_2}{\Gamma \rhd G_1 \,\&\, G_2}\ \text{g\_and}}$$

$$\frac{\Gamma\,,\, D \rhd G}{\Gamma \rhd D \longrightarrow G}\ \text{g\_imp} \qquad \frac{}{\Gamma \rhd \mathtt{T}}\ \text{g\_tt} \qquad \frac{\mathtt{proper}\ E \quad \Gamma \rhd G\ E}{\Gamma \rhd \mathtt{Some}\ G}\ \text{g\_some}$$

$$\frac{\forall (E : \mathtt{expr\ \ con}), (\mathtt{proper}\ E \to \Gamma \rhd G\ E)}{\Gamma \rhd \mathtt{All}\ G}\ \text{g\_all} \qquad \frac{\forall (E : \mathtt{X}), (\Gamma \rhd G\ E)}{\Gamma \rhd \mathtt{Allx}\ G}\ \text{g\_allx}$$

Backchaining Rules

$$\frac{}{\Gamma, [\langle\, A \,\rangle] \rhd A}\ \text{b\_match} \qquad \frac{\Gamma, [D_1] \rhd A}{\Gamma, [D_1 \,\&\, D_2] \rhd A}\ \text{b\_and1} \qquad \frac{\Gamma, [D_2] \rhd A}{\Gamma, [D_1 \,\&\, D_2] \rhd A}\ \text{b\_and2}$$

$$\frac{\Gamma \rhd G \quad \Gamma, [D] \rhd A}{\Gamma, [G \longrightarrow D] \rhd A}\ \text{b\_imp} \qquad \frac{\mathtt{proper}\ E \quad \Gamma, [D\ E] \rhd A}{\Gamma, [\mathtt{All}\ D] \rhd A}\ \text{b\_all} \qquad \frac{\Gamma, [D\ E] \rhd A}{\Gamma, [\mathtt{Allx}\ D] \rhd A}\ \text{b\_allx}$$

$$\frac{\forall (E : \mathtt{expr\ \ con}), (\mathtt{proper}\ E \to \Gamma, [D\ E] \rhd A)}{\Gamma, [\mathtt{Some}\ D] \rhd A}\ \text{b\_some}$$

Encoding Sequents as Inductive Dependent Types

$$\vdots$$

$$\frac{\Gamma \rhd G_1 \quad \Gamma \rhd G_2}{\Gamma \rhd G_1 \,\&\, G_2}\ \text{g\_and}$$

$$\vdots$$

$$\frac{\Gamma \rhd G \quad \Gamma, [D] \rhd A}{\Gamma, [G \longrightarrow D] \rhd A}\ \text{b\_imp}$$

$$\vdots$$

```
Inductive grseq : context -> oo -> Prop :=
...
| g_and :
 forall (L : context) (G1 G2 : oo),
 grseq L G1 -> grseq L G2 ->
 grseq L (G1 & G2)
...
with bcseq : context -> oo ->  atm -> Prop :=
| b_imp :
 forall (L : context) (F G : oo) (A : atm),
 grseq L G -> bcseq L D A
  -> bcseq L (G ---> D) A.
...
```

Inductive Types

Encapsulate infinite collection in finite set of rules

Example:

```
Inductive nat : Set :=
| O : nat
| S : nat -> nat.
```

Induction principle for property $P : \texttt{nat} \to \texttt{Prop}$:

$$\frac{\mathrm{P}\ \ 0 \quad \forall(\mathrm{n} : \mathrm{nat}), \mathrm{P}\ \ \mathrm{n} \to \mathrm{P}\ \ (\mathrm{S}\ \ \mathrm{n})}{\forall(\mathrm{n} : \mathrm{nat}), \mathrm{P}\ \ \mathrm{n}}$$

In linear form:

$$\begin{aligned}
&\forall(P : \texttt{nat} \to \texttt{Prop}), \\
&\quad (P\ 0) \to \\
&\quad (\forall(n : \texttt{nat}), P\ n \to P\ (S\ n)) \to \\
&\ \forall(n : \texttt{nat}), P\ n.
\end{aligned}$$

Sequent Mutual Induction Principle

$$\vdots$$

$$\frac{\Gamma \rhd G_1 \quad \Gamma \rhd G_2}{\Gamma \rhd G_1 \,\&\, G_2}\ \text{g\_and}$$

$$\vdots$$

$$\frac{\Gamma \rhd G \quad \Gamma, [D] \rhd A}{\Gamma, [G \longrightarrow D] \rhd A}\ \text{b\_imp}$$

$$\vdots$$

$$
\begin{aligned}
\texttt{seq\_mutind} : \forall (P_1 &: \texttt{context} \to \texttt{oo} \to \texttt{Prop}) \\
&(P_2 : \texttt{context} \to \texttt{oo} \to \texttt{atm} \to \texttt{Prop}), \\
&\vdots \\
&(\forall (c : \texttt{context})(o_1\ o_2 : \texttt{oo}), \\
&\quad c \rhd o_1 \to P_1\ c\ o_1 \to c \rhd o_2 \to P_1\ c\ o_2 \to \\
&\quad P_1\ c\ (o_1 \& o_2)) \to \\
&\vdots \\
&(\forall (c : \texttt{context})(o_1\ o_2 : \texttt{oo})(a : \texttt{atm}), \\
&\quad c \rhd o_1 \to P_1\ c\ o_1 \to c, [o_2] \rhd a \to P_2\ c\ o_2\ a \to \\
&\quad P_2\ c\ (o_1 \longrightarrow o_2)\ a) \to \\
&\vdots \\
(\forall (c :\ &\texttt{context})(o : \texttt{oo}), \\
&\qquad c \rhd o \to P_1\ c\ o) \wedge \\
(\forall (c :\ &\texttt{context})(o : \texttt{oo})(a : \texttt{atm}), \\
&\qquad c, [o] \rhd a \to P_2\ c\ o\ a)
\end{aligned}
$$

Structural Rules

$$\frac{\Gamma \rhd \beta_2}{\Gamma\,,\beta_1 \rhd \beta_2}\ \texttt{gr\_weakening} \qquad \frac{\Gamma, [\beta_2] \rhd \alpha}{\Gamma\,,\beta_1, [\beta_2] \rhd \alpha}\ \texttt{bc\_weakening}$$

$$\frac{\Gamma\,,\ \beta_1\,,\beta_1 \rhd \beta_2}{\Gamma\,,\beta_1 \rhd \beta_2}\ \texttt{gr\_contraction} \qquad \frac{\Gamma\,,\ \beta_1\,,\ \beta_1, [\beta_2] \rhd \alpha}{\Gamma\,,\ \beta_1, [\beta_2] \rhd \alpha}\ \texttt{bc\_contraction}$$

$$\frac{\Gamma\,,\ \beta_2\,,\ \beta_1 \rhd \beta_3}{\Gamma\,,\beta_1\,,\ \beta_2 \rhd \beta_3}\ \texttt{gr\_exchange} \qquad \frac{\Gamma\,,\ \beta_2\,,\ \beta_1, [\beta_3] \rhd \alpha}{\Gamma\,,\ \beta_1\,,\ \beta_2, [\beta_3] \rhd \alpha}\ \texttt{bc\_exchange}$$

These are all corollaries of a general theorem:

Theorem (`monotone`)

$$\frac{\Gamma \subseteq \Gamma' \quad \Gamma \rhd \beta}{\Gamma' \rhd \beta} \wedge \frac{\Gamma \subseteq \Gamma' \quad \Gamma, [\beta] \rhd \alpha}{\Gamma', [\beta] \rhd \alpha}$$

Theorem (monotone)

$$
\begin{aligned}
&(\ \forall(\Gamma : \texttt{context})(\beta : \texttt{oo}),\\
&\qquad \Gamma \rhd \beta \to \forall(\Gamma' : \texttt{context}), \Gamma \subseteq \Gamma' \to \Gamma' \rhd \beta\ \ ) \wedge\\
&(\ \forall(\Gamma : \texttt{context})(\beta : \texttt{oo})(\alpha : \texttt{atm}),\\
&\qquad \Gamma, [\beta] \rhd \alpha \to \forall(\Gamma' : \texttt{context}), \Gamma \subseteq \Gamma' \to \Gamma', [\beta] \rhd \alpha\ \ )
\end{aligned}
$$

Define

$$
\begin{aligned}
P_1 &:= \lambda(\Gamma : \texttt{context})(\beta : \texttt{oo})\ .\\
&\qquad \forall(\Gamma' : \texttt{context}), \Gamma \subseteq \Gamma' \to \Gamma' \rhd \beta\\
P_2 &:= \lambda(\Gamma : \texttt{context})(\beta : \texttt{oo})(\alpha : \texttt{atm})\ .\\
&\qquad \forall(\Gamma' : \texttt{context}), \Gamma \subseteq \Gamma' \to \Gamma', [\beta] \rhd \alpha
\end{aligned}
$$

Proof

By induction over $\Gamma \rhd \beta$ and $\Gamma, [\beta] \rhd \alpha$ using `seq_mutind`.

Theorem (monotone)

$$
\begin{aligned}
&(\ \forall(\Gamma : \texttt{context})(\beta : \texttt{oo}),\\
&\qquad \Gamma \rhd \beta \rightarrow \boxed{\forall(\Gamma' : \texttt{context}), \Gamma \subseteq \Gamma' \rightarrow \Gamma' \rhd \beta}\ ) \land\\
&(\ \forall(\Gamma : \texttt{context})(\beta : \texttt{oo})(\alpha : \texttt{atm}),\\
&\qquad \Gamma, [\beta] \rhd \alpha \rightarrow \forall(\Gamma' : \texttt{context}), \Gamma \subseteq \Gamma' \rightarrow \Gamma', [\beta] \rhd \alpha\ \ )
\end{aligned}
$$

Define

$$
\begin{aligned}
P_1 &\coloneqq \lambda(\Gamma : \texttt{context})(\beta : \texttt{oo})\ .\\
&\qquad \boxed{\forall(\Gamma' : \texttt{context}), \Gamma \subseteq \Gamma' \rightarrow \Gamma' \rhd \beta}\\
P_2 &\coloneqq \lambda(\Gamma : \texttt{context})(\beta : \texttt{oo})(\alpha : \texttt{atm})\ .\\
&\qquad \forall(\Gamma' : \texttt{context}), \Gamma \subseteq \Gamma' \rightarrow \Gamma', [\beta] \rhd \alpha
\end{aligned}
$$

Proof

By induction over $\Gamma \rhd \beta$ and $\Gamma, [\beta] \rhd \alpha$ using `seq_mutind`.

Theorem (`monotone`)

$$
\begin{aligned}
&(\ \forall(\Gamma : \mathtt{context})(\beta : \mathtt{oo}),\\
&\qquad \Gamma \rhd \beta \to \forall(\Gamma' : \mathtt{context}), \Gamma \subseteq \Gamma' \to \Gamma' \rhd \beta\ )\ \wedge\\
&(\ \forall(\Gamma : \mathtt{context})(\beta : \mathtt{oo})(\alpha : \mathtt{atm}),\\
&\qquad \Gamma, [\beta] \rhd \alpha \to \boxed{\forall(\Gamma' : \mathtt{context}), \Gamma \subseteq \Gamma' \to \Gamma', [\beta] \rhd \alpha}\ )
\end{aligned}
$$

Define

$$
\begin{aligned}
P_1 &:= \lambda(\Gamma : \mathtt{context})(\beta : \mathtt{oo})\ .\\
&\qquad \forall(\Gamma' : \mathtt{context}), \Gamma \subseteq \Gamma' \to \Gamma' \rhd \beta\\
P_2 &:= \lambda(\Gamma : \mathtt{context})(\beta : \mathtt{oo})(\alpha : \mathtt{atm})\ .\\
&\qquad \boxed{\forall(\Gamma' : \mathtt{context}), \Gamma \subseteq \Gamma' \to \Gamma', [\beta] \rhd \alpha}
\end{aligned}
$$

Proof

By induction over $\Gamma \rhd \beta$ and $\Gamma, [\beta] \rhd \alpha$ using `seq_mutind`.

Proof of `monotone`

Case $\dfrac{\Gamma\ ,\ D \rhd G}{\Gamma \rhd D \longrightarrow G}$ **g_imp:**

From `seq_mutind`, proving

$$\dfrac{\begin{array}{l} H\ : c, o_1 \rhd o_2 \\ IH\ : P_1\ (c, o_1)\ o_2 \end{array}}{P_1\ c\ (o_1 \longrightarrow o_2)}$$

Proof of `monotone`

Case $\dfrac{\Gamma\,,\; D \rhd G}{\Gamma \rhd D \longrightarrow G}$ **g_imp:**

From `seq_mutind`, proving

$$\dfrac{\begin{array}{l} H \;:\, c, o_1 \rhd o_2 \\ IH \;:\, P_1 \; (c, o_1) \; o_2 \end{array}}{P_1 \; c \; (o_1 \longrightarrow o_2)}$$

Next: unfold P_1

Proof of `monotone`

Case $\dfrac{\Gamma\,,\ D \rhd G}{\Gamma \rhd D \longrightarrow G}$ **g_imp:**

From `seq_mutind`, proving

$$\dfrac{\begin{array}{l} H\ : c, o_1 \rhd o_2 \\ IH\ : \forall(\Gamma_0 : \texttt{context}), (c, o_1) \subseteq \Gamma_0 \to \Gamma_0 \rhd o_2 \end{array}}{\forall(\Gamma' : \texttt{context}), c \subseteq \Gamma' \to \Gamma' \rhd o_1 \longrightarrow o_2}$$

Proof of `monotone`

Case $\dfrac{\Gamma\,,\;D \rhd G}{\Gamma \rhd D \longrightarrow G}$ **g_imp:**

From `seq_mutind`, proving

$$\dfrac{\begin{array}{l} H \;: c, o_1 \rhd o_2 \\ IH \;: \forall(\Gamma_0 : \texttt{context}), (c, o_1) \subseteq \Gamma_0 \to \Gamma_0 \rhd o_2 \end{array}}{\forall(\Gamma' : \texttt{context}), c \subseteq \Gamma' \to \Gamma' \rhd o_1 \longrightarrow o_2}$$

Next: introduce from goal

Proof of `monotone`

Case $\dfrac{\Gamma\,,\; D \rhd G}{\Gamma \rhd D \longrightarrow G}$ **g_imp:**

From `seq_mutind`, proving

$$\begin{array}{r l} H & : c, o_1 \rhd o_2 \\ IH & : \forall (\Gamma_0 : \texttt{context}), (c, o_1) \subseteq \Gamma_0 \to \Gamma_0 \rhd o_2 \\ P & : c \subseteq \Gamma' \\ \hline & \Gamma' \rhd o_1 \longrightarrow o_2 \end{array}$$

Proof of `monotone`

Case $\dfrac{\Gamma\,,\;D \rhd G}{\Gamma \rhd D \longrightarrow G}$ **g_imp:**

From `seq_mutind`, proving

$$\frac{\begin{array}{rl} H & : c, o_1 \rhd o_2 \\ IH & : \forall(\Gamma_0 : \texttt{context}), (c, o_1) \subseteq \Gamma_0 \to \Gamma_0 \rhd o_2 \\ P & : c \subseteq \Gamma' \end{array}}{\Gamma' \rhd o_1 \longrightarrow o_2}$$

Next: backchain with g_imp on goal

Proof of `monotone`

Case $\dfrac{\Gamma\,,\ D \rhd G}{\Gamma \rhd D \longrightarrow G}$ **g_imp:**

From `seq_mutind`, proving

$$\begin{array}{rl} H & : c, o_1 \rhd o_2 \\ IH & : \forall(\Gamma_0 : \texttt{context}), (c, o_1) \subseteq \Gamma_0 \to \Gamma_0 \rhd o_2 \\ P & : c \subseteq \Gamma' \\ \hline & \Gamma', o_1 \rhd o_2 \end{array}$$

Proof of `monotone`

Case $\dfrac{\Gamma\,,\; D \rhd G}{\Gamma \rhd D \longrightarrow G}$ **g_imp:**

From `seq_mutind`, proving

$$\frac{\begin{array}{rl} H & : c, o_1 \rhd o_2 \\ IH & : \forall(\Gamma_0 : \texttt{context}), (c, o_1) \subseteq \Gamma_0 \rightarrow \Gamma_0 \rhd o_2 \\ P & : c \subseteq \Gamma' \end{array}}{\Gamma', o_1 \rhd o_2}$$

Next: backchain with IH on goal with $(\Gamma_0 := \Gamma', o_1)$

Proof of `monotone`

Case $\dfrac{\Gamma\,,\; D \rhd G}{\Gamma \rhd D \longrightarrow G}$ **g_imp:**

From `seq_mutind`, proving

$$\dfrac{\begin{array}{r l} H & : c, o_1 \rhd o_2 \\ IH & : \forall(\Gamma_0 : \texttt{context}), (c, o_1) \subseteq \Gamma_0 \to \Gamma_0 \rhd o_2 \\ P & : c \subseteq \Gamma' \end{array}}{(c, o_1) \subseteq (\Gamma', o_1)}$$

Proof of `monotone`

Case $\dfrac{\Gamma\,,\ D \rhd G}{\Gamma \rhd D \longrightarrow G}$ **g_imp:**

From `seq_mutind`, proving

$$\dfrac{\begin{array}{r l} H & : c, o_1 \rhd o_2 \\ IH & : \forall(\Gamma_0 : \texttt{context}), (c, o_1) \subseteq \Gamma_0 \to \Gamma_0 \rhd o_2 \\ P & : c \subseteq \Gamma' \end{array}}{(c, o_1) \subseteq (\Gamma', o_1)}$$

Next: backchain with context lemma that says if $c \subseteq \Gamma'$ then $(c, o_1) \subseteq (\Gamma', o_1)$

Proof of `monotone`

Case $\dfrac{\Gamma\,,\;D \rhd G}{\Gamma \rhd D \longrightarrow G}$ **g_imp:**

From `seq_mutind`, proving

$$\dfrac{\begin{array}{rl} H & : c, o_1 \rhd o_2 \\ IH & : \forall(\Gamma_0 : \texttt{context}), (c, o_1) \subseteq \Gamma_0 \to \Gamma_0 \rhd o_2 \\ P & : c \subseteq \Gamma' \end{array}}{c \subseteq \Gamma'}$$

Proof of `monotone`

Case $\dfrac{\Gamma\,,\ D \rhd G}{\Gamma \rhd D \longrightarrow G}$ **g_imp:**

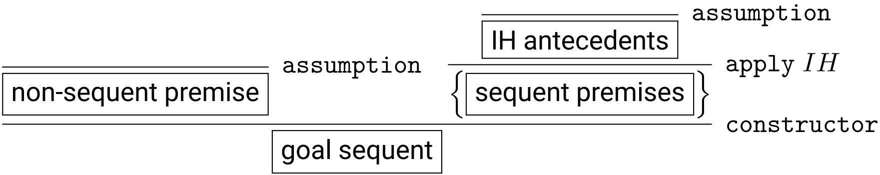
From `seq_mutind`, proving

$$\dfrac{\begin{array}{rl} H & : c, o_1 \rhd o_2 \\ IH & : \forall(\Gamma_0 : \texttt{context}), (c, o_1) \subseteq \Gamma_0 \rightarrow \Gamma_0 \rhd o_2 \\ P & : c \subseteq \Gamma' \end{array}}{c \subseteq \Gamma'}$$

Goal is provable by assumption P

Structural Rules Summary

Proof with 15 subcases proven automatically in Coq




```
Proof.
Hint Resolve context_sub_sup.
eapply seq_mutind; intros;
try (econstructor; eauto; eassumption).
Qed.
```

Prove cut admissibility with this one weird trick...

Theorem (`cut_admissible`)

$$\frac{\Gamma, \delta \rhd \beta \quad \Gamma \rhd \delta}{\Gamma \rhd \beta} \wedge \frac{\Gamma, \delta, [\beta] \rhd \alpha \quad \Gamma \rhd \delta}{\Gamma, [\beta] \rhd \alpha}$$

Proof by nested induction over δ then mutual structural induction over $\Gamma, \delta \rhd \beta$ and $\Gamma, \delta, [\beta] \rhd \alpha$

98 of 105 cases proven automatically in Coq

```
Proof.
Hint Resolve gr_weakening context_swap.
induction delta; eapply seq_mutind; intros;
subst; try (econstructor; eauto; eassumption).
...
```

[Battell, Felty; LFMTP-16]

Takeaway

- Coq type system is useful for formal theorem proving

Takeaway

- Coq type system is useful for formal theorem proving
- Rich type systems make induction even more awesome

Takeaway

- Coq type system is useful for formal theorem proving
- Rich type systems make induction even more awesome
- Inference systems can be encoded as inductive types in Coq

Takeaway

- Coq type system is useful for formal theorem proving
- Rich type systems make induction even more awesome
- Inference systems can be encoded as inductive types in Coq
- Structural properties of logics can be proven in Coq

Takeaway

- Coq type system is useful for formal theorem proving
- Rich type systems make induction even more awesome
- Inference systems can be encoded as inductive types in Coq
- Structural properties of logics can be proven in Coq

Thank you!