

$\bullet \Gamma \text{ proof} - \Gamma$

- I.H.
- $\bullet \Gamma \text{ proof (induction } x) \vdash \text{goals}[\text{proof-level}] = \bigwedge_{\forall x's} x's \Rightarrow P(x's)$
 - $\rightarrow \text{for } I x \Rightarrow P_x \text{ Unify}(x, x \text{ in rule}) = \{x \Rightarrow x's\}$ s.t.
 - $\bullet \Gamma \text{ proof (induction ev. induct) } \vdash \text{goals}[\text{proof-level}] =$
 $\bigwedge_{\forall x's} \text{I. assumptions } x's \Rightarrow P x's \Rightarrow P(x's)$

note: P could be an implication itself.

$$\frac{\Gamma \vdash T^1 \quad \Gamma' C_1 T''}{\Gamma' C_1 C_2 T''} \text{ Sequences}$$

For note: the assumption is that we have an AST of half proof, to know where the splits of code would be. i.e. its already process.

- \sqcap assure $\ell_1 \dots \ell_n \sqcap [\ell_i : \text{proof-level}] = \ell_i : \text{proof}$.
 \sqcap assnes. Inc.
- \sqcap from $\ell_1 \dots \ell_n \sqcap [\ell_i : \text{proof-level}] = \ell_i : \text{proof}$.
- \sqcap unq $\ell_1 \dots \ell_n \sqcap [\ell_i : \text{proof-level}] = \ell_i : \text{proof}$
- \sqcap fix $x_1 \dots x_n \sqcap [x_i : \text{proof-level}] = x_i$

- (Note that Prop \sqcap -assumes case since it only defines a local fact,
so unification with premises are not need).
- \sqcap have Prop \vdash
 - \sqcap . Current-prop = Prop

- If $\text{Unify}(\text{Prop}, \text{Some goal}^{\text{in}})$
 \sqcap .Current-prop = Prop
and \sqcap .assumes = $\boxed{}$ (empty)
- \sqcap Shows Prop \sqcap .Current-prop = Prop if $\text{Unify}(\text{prop}, \text{Some goal in})$
 \sqcap .goals
and \sqcap .assumes == True and
 $\text{Unify}(\sqcap.\text{local-facts}(),$
 $\sqcap.\text{goal}[\text{Prop}], \text{premises})$

• \lceil by method $\lceil[\text{label}.\text{proof-level}] = \lceil[\text{current-pop}]$ if $\lceil[\text{shans}] =$
 $\lceil[\text{goals.remove(current-pop)}]$
 $\lceil[\text{shans}] = \text{False}$

• \lceil by method $\lceil[\text{label}.\text{proof-level}] = \lceil[\text{current-pop}]$ if $\lceil[\text{true}] =$
 $\lceil[\text{shans}] = \text{False}$

7 qed \sqcap [label · proof-level] = \sqcap [current - prep]
 7. remove all-facts-at-level (current proof-level)
 7. assumes = false

- \sqcap next \sqcap . remove-all-facts-at-level (previous proof-level)
 \sqcap . assns = False

• [obtain x_1, \dots, x_n where 0 : " $P(x_1, \dots, x_n)$ " \cap "[post_level. x_i]" = x_i
such that $P(x_1, \dots, x_n)$ is
satisfied]