Foundations of Computer Software

1月13日
担当： 小林
http://www.kb.ecei.tohoku.ac.jp/~koba/class/soft-kiso/

Top-level commands (Vernacular commands)

Definition

Form:
Definition <name> := <term>.

Example:
Definition X := 1.
(** Bind name X to 1 **)

Inductive

Form:
Inductive <name>: <type> :=
<constructor> : <type>
| ...
| <constructor>: <type>.

Example:
Inductive mynat: Set :=
Z : mynat
| S : mynat -> mynat.
(* mynat is the set constructed from Z and S *)

Fixpoint

Form:
Fixpoint <func> (<arg>*: <type>)
{struct <arg>}: <type> :=
<body>

Example:
Fixpoint plus (m n: mynat) {struct m} : mynat :=
match m with
Z => n
| S m' => S(plus m' n)
end.

Remark: The argument specified by "struct" (m, in the example above) must decrease monotonically upon each recursive call.

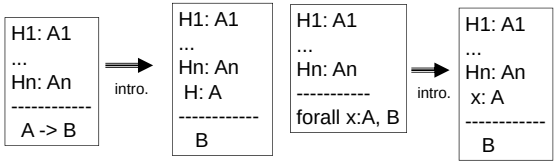
Check/Print/Eval

- Check <term>
  - Check the type of <term>.
    E.g. Definition X := 1.
    Check X.
    X: nat
- Print <term>
  - Print (the definition of) <term> and its type.
    E.g. Definition X:= 1.
    Print X.
    X = 1 : nat
- Eval compute in <term>.
  - Reduce <term> to a normal form.
    E.g. Eval compute in 1+2.
    = 3: nat

Proof Commands (Tactics)

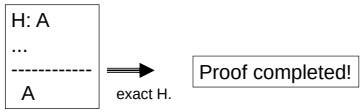
intro

Applicable when the goal is of the form
A-> B or forall x:A, B



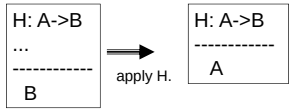
exact

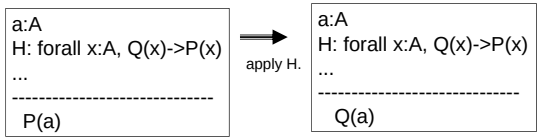
Directly specify a proof (as a λ-term)



apply

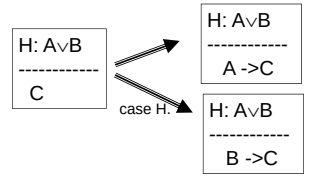
- Applicable when you have an assumption or a theorem of the form A->B, and the goal is B

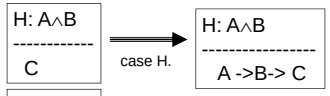


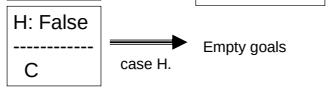


case <term>

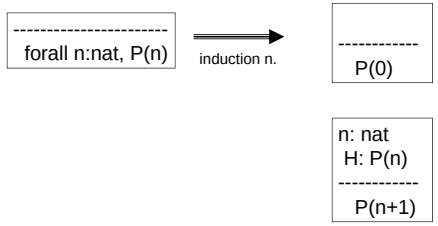
- Case analysis on <term>





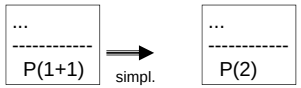


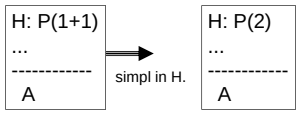
induction



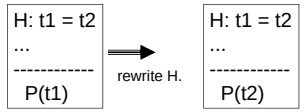
simpl

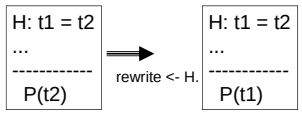
Simplifies terms in the goal or an assumption



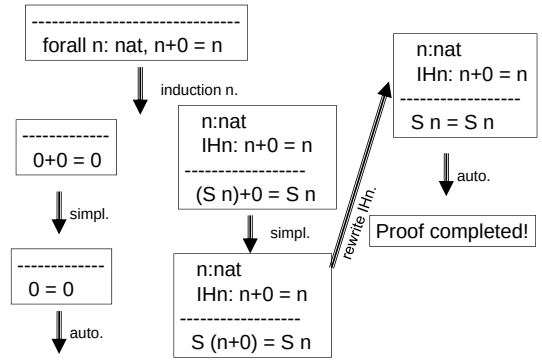


rewrite





Example: induction, simpl, rewrite

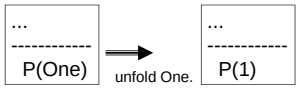


unfold

Unfold a name according to the definition

Example:

Definition One := 1.



Applications of theorems

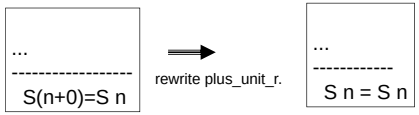
- Theorems and axioms can be used as arguments of apply, rewrite, exact

Example:

Assume

Theorem plus_unit_r: forall n:nat, n+0 = n.

has been proven.



A list of tactics

apply <term>.	**Apply a hypothesis or theorem <term>, having the current goal as a conclusion**
auto.	**Automated proof search**
case <term>	**Case analysis on <term>.**
exact <term>.	**Give <term> as a proof.**
induction <term>.	**Prove by induction on <term>**
intro.	**Applicable when the goal is A->B or forall x:A, B (just move A to a hypothesis)**
left.	**When the goal is A∨B, choose A and prove it.**
rewrite <term>.	**Rewrite the goal with an equality A=B**
right.	**When the goal is A∨B, choose B and prove it.**
simpl [in H].	**Simplify the goal or a hypothesis H.**
split.	**Split the goal A∧B into A and B.**
unfold X.	**Unfold a name X.**