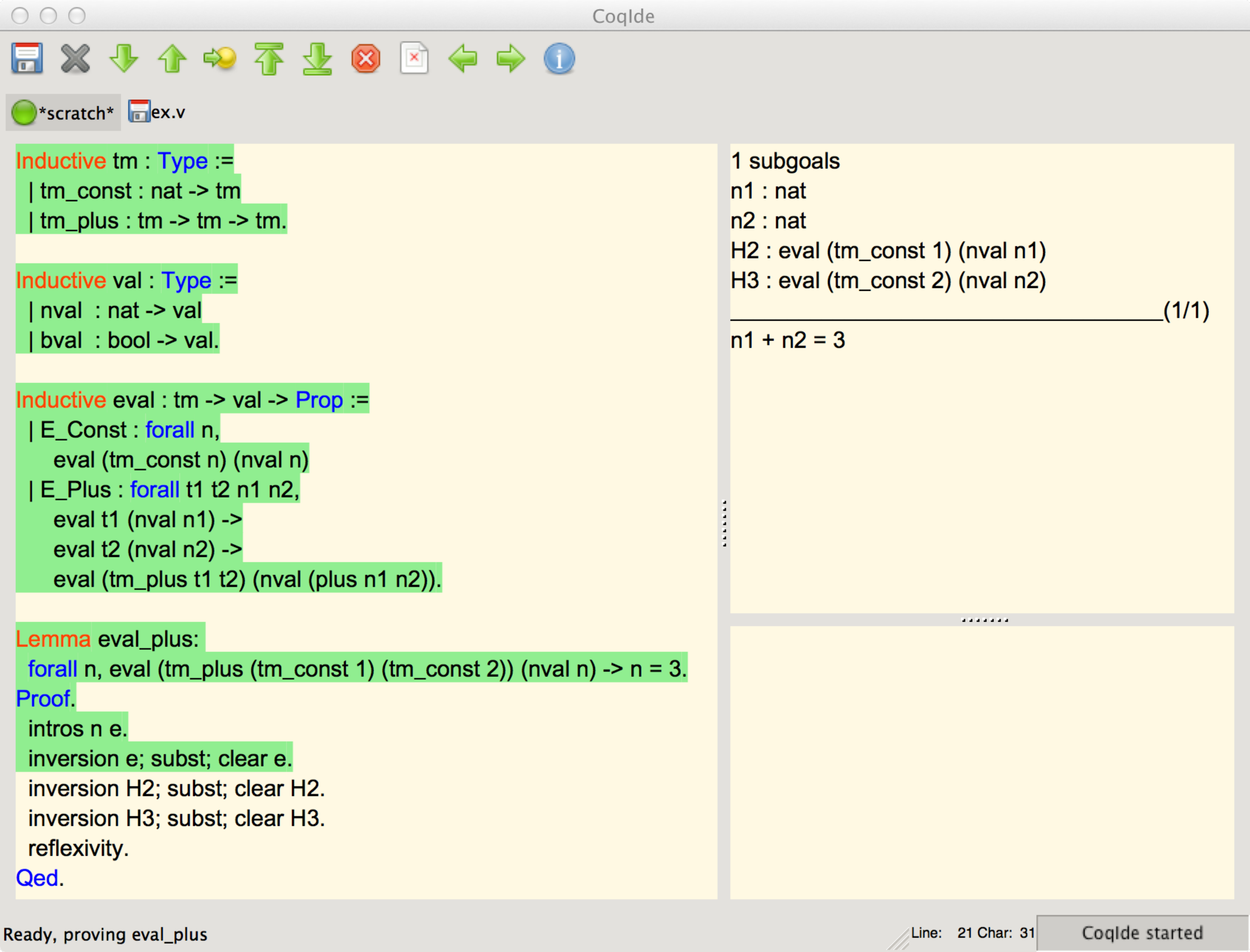

```
Inductive tm : Type :=
  | tm_const : nat -> tm
  | tm_plus : tm -> tm -> tm.

Inductive val : Type :=
  | nval  : nat -> val
  | bval  : bool -> val.

Inductive eval : tm -> val -> Prop :=
  | E_Const : forall n,
      eval (tm_const n) (nval n)
  | E_Plus : forall t1 t2 n1 n2,
      eval t1 (nval n1) ->
      eval t2 (nval n2) ->
      eval (tm_plus t1 t2) (nval (plus n1 n2)).

Lemma eval_plus:
  forall n, eval (tm_plus (tm_const 1) (tm_const 2)) (nval n) -> n = 3.
Proof.
  intros n e.
  inversion e; subst; clear e.
  inversion H2; subst; clear H2.
  inversion H3; subst; clear H3.
  reflexivity.
Qed.
```

```
1 subgoals
n1 : nat
n2 : nat
H2 : eval (tm_const 1) (nval n1)
H3 : eval (tm_const 2) (nval n2)
______________________________________(1/1)
n1 + n2 = 3
```

Ready, proving eval_plus

Line: 21 Char: 31

CoqIde started