Coq report 2

The part of Sets

First of all

Make a set

```
Section Sets.
```

Something else

```
Open Scope type_scope.
Set Implicit Arguments.
Implicit Arguments inl [A B].
Implicit Arguments inr [A B].
```

- Using `Open Scope type_scope.` to set the scope of argument scopes.

Finite Sets

Inductive

Using `Inductive` to define the set what you want:
E.g. To define the empty set

```
Inductive empty_set : Set := .
```

To define the set of color only include two colors:

```
Inductive Color : Set :=
  | white: Color
  | black : Color.
```

Also can define the Boolean set:

```
Inductive bool :  Set := true | false.
```

Products

Tips

- `Definition ident :type := define.` Defines an object that can selectively declare its type, but must have a definition.

E.g. before explaining that, we need define a Rank as a set:

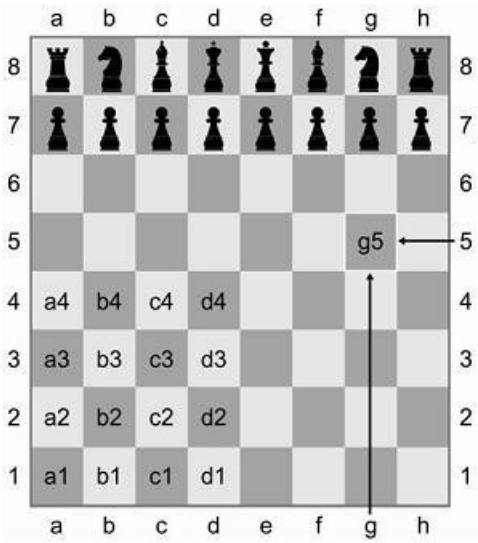
```
Inductive Rank : Set :=
  | pawn : Rank
  | rook : Rank
  | knight : Rank
  | bishop : Rank
  | queen : Rank
  | king : Rank.
```

To define a new set `Color * Rank` which called `Piece`. It's the set of pairs `(Color * Rank)`.
In this case, the `blackKnight` is the set pairs `(Color * Rank)`. And also the black and knight that belong to `Color` and `Rank` set.

```
Definition Piece : Set := Color * Rank.
Definition blackKnight : Piece := (black , knight).
```

In this case, we also need to define the coordinates

```
Inductive XCoord : Set :=
  | xa : XCoord
  | xb : XCoord
  | xc : XCoord
  | xd : XCoord
  | xe : XCoord
  | xf : XCoord
  | xg : XCoord
  | xh : XCoord.

Inductive YCoord : Set :=
  | y1 : YCoord
  | y2 : YCoord
  | y3 : YCoord
  | y4 : YCoord
  | y5 : YCoord
  | y6 : YCoord
  | y7 : YCoord
  | y8 : YCoord.
```



Define some fuctions

Using this template to define the some functions:

```
Definition function(X: Type)(x: X) : nat->X :=
  fun (k:nat) => x.
```

It needs a return value or a function.
In this case, we need to define some generic operations:

```
Definition fst(A B : Set)(p : A * B) : A :=
   match p with
   | (a , b) => a
   end.

Definition snd(A B : Set)(p : A * B) : B :=
   match p with
   | (a , b) => b
   end.
```

Extract the components

```
Eval compute in fst blackKnight.
Eval compute in snd blackKnight.
Eval compute in (fst blackKnight,snd blackKnight).
```

Destruct

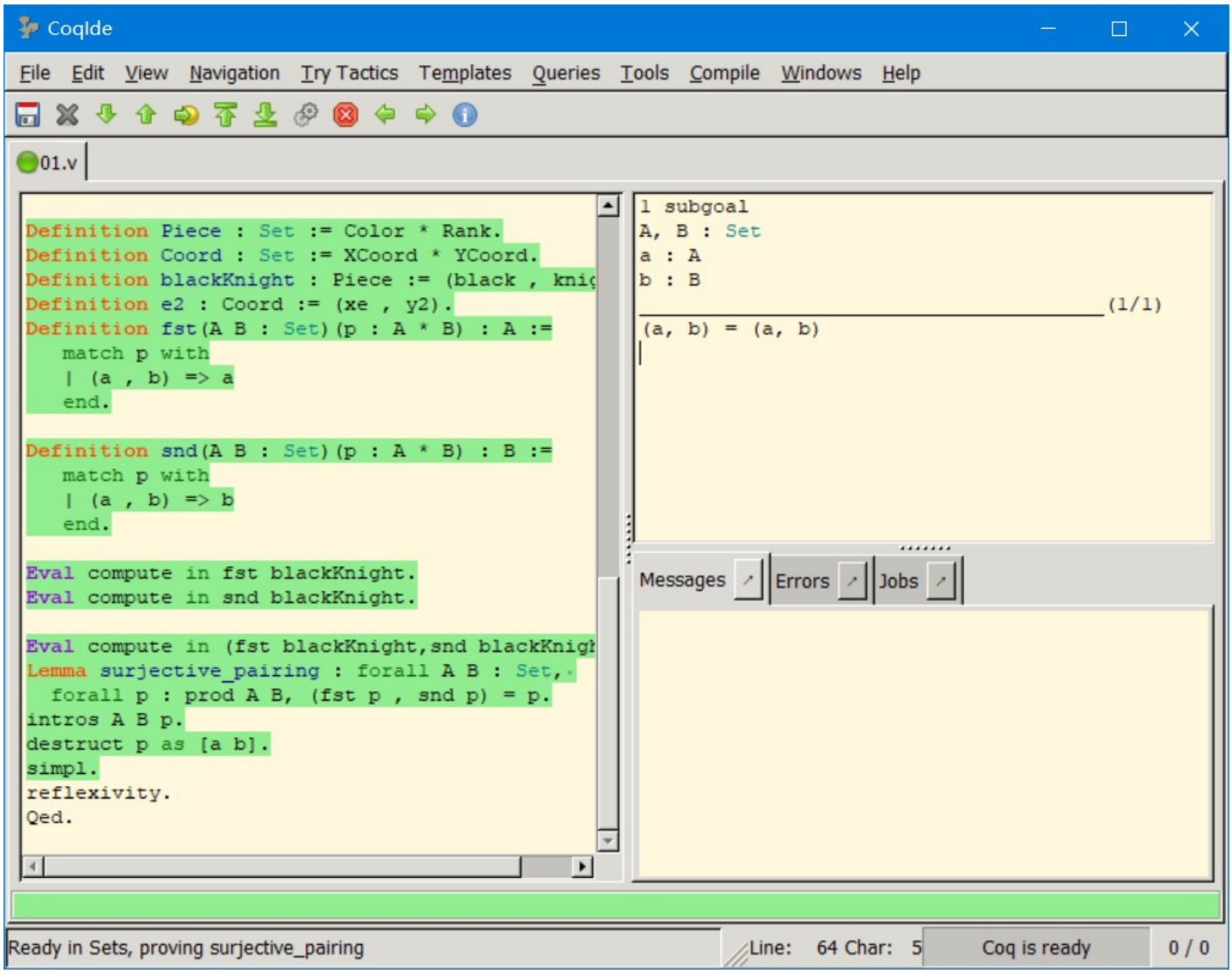
```
destruct p as [a b].
```

All source code in that case

```
Section Sets.

Open Scope type_scope.
Set Implicit Arguments.
Implicit Arguments inl [A B].
Implicit Arguments inr [A B].

Inductive Color : Set :=
  | white: Color
  | black : Color.

Inductive Rank : Set :=
  | pawn : Rank
  | rook : Rank
  | knight : Rank
  | bishop : Rank
  | queen : Rank
  | king : Rank.

Inductive XCoord : Set :=
  | xa : XCoord
```

```
  | xb : XCoord
  | xc : XCoord
  | xd : XCoord
  | xe : XCoord
  | xf : XCoord
  | xg : XCoord
  | xh : XCoord.

Inductive YCoord : Set :=
  | y1 : YCoord
  | y2 : YCoord
  | y3 : YCoord
  | y4 : YCoord
  | y5 : YCoord
  | y6 : YCoord
  | y7 : YCoord
  | y8 : YCoord.

Definition Piece : Set := Color * Rank.
Definition Coord : Set := XCoord * YCoord.
Definition blackKnight : Piece := (black , knight).
Definition e2 : Coord := (xe , y2).
Definition fst(A B : Set)(p : A * B) : A :=
   match p with
   | (a , b) => a
   end.

Definition snd(A B : Set)(p : A * B) : B :=
   match p with
   | (a , b) => b
   end.

Eval compute in fst blackKnight.
Eval compute in snd blackKnight.

Eval compute in (fst blackKnight,snd blackKnight).
Lemma surjective_pairing : forall A B : Set,
  forall p : prod A B, (fst p , snd p) = p.
intros A B p.
destruct p as [a b].
simpl.
reflexivity.
Qed.
```

Run result screenshot


CoqIde
File Edit View Navigation Try Tactics Templates Queries Tools Compile Windows Help
01.v
Definition Piece : Set := Color * Rank.
Definition Coord : Set := XCoord * YCoord.
Definition blackKnight : Piece := (black , knig
Definition e2 : Coord := (xe , y2).
Definition fst(A B : Set)(p : A * B) : A :=
match p with
| (a , b) => a
end.
Definition snd(A B : Set)(p : A * B) : B :=
match p with
| (a , b) => b
end.
Eval compute in fst blackKnight.
Eval compute in snd blackKnight.
Eval compute in (fst blackKnight,snd blackKnigh
Lemma surjective_pairing : forall A B : Set,
forall p : prod A B, (fst p , snd p) = p.
intros A B p.
destruct p as [a b].
simpl.
reflexivity.
Qed.
1 subgoal
A, B : Set
a : A
b : B
(1/1)
(a, b) = (a, b)
Messages
Errors
Jobs
Ready in Sets, proving surjective_pairing
Line: 64 Char: 5
Coq is ready
0 / 0