2025/01/21

ssreflectを用いたCoqの証明パターンとその応用

情報科学研究科修士一年 佐藤龍之介

Table of contents

01 What is Coq?

Coqについての概要

02 SSReflectと mathcomp

それぞれの違いについて

03 タクティク

スクリプトで主に使用される
命令について

04 実際に問題を解く！

具体的に簡単な問題を通して
ssreflect/mathcompを使ってみる

01

What is Coq?

Coqについての概要

用語：形式化

定理証明支援系に定義・定理・証明などを入力するためにそれらの言明を支援系の言語に翻訳する作業を意味する

支援の例：

（記号の導入）この際に、Σや∀などの記法を用いることで形式化が直感的に行える、加えて証明がコンパクトになる

（検索機能）補題に対して必要な言明を探すこと、エディタ上で行われる

Coqとは

概要:
Coqは形式的証明（formal proof）のためのインタラクティブな定理証明支援ツール（proof assistant）論理的な定理を証明し、プログラムの正確性を形式的に検証するために使用される。
特徴:

- 強力な型システム（依存型）を持つ
- 証明を構築するためのタクティック（命令）のセットを提供
- 高度な抽象性とモジュール性をサポート

用途:

- プログラムの正当性の証明
- 数学的な定理の形式的証明
- 高信頼性が求められるソフトウェアやハードウェアの検証

Coqの標準ライブラリとその欠点：

標準ライブラリ

- ブール代数ライブラリ Bool
- 自然数のライブラリ Arith, NArith
- 二進法のライブラリ ZArith
- 有理数のライブラリ QArith
- リストデータ構造 Lists
- 集合論のライブラリ Sets

欠点

- 集合論のライブラリを使用した有限集合や無限集合の扱いが複雑
- 数学の形式化が揃っていない
- 関数型プログラミングに必要なリストデータ構造の形式化が不十分

02

SSReflectとmathcomp

それぞれの概要

証明言語 SSReflect

きっかけ

“コンティエの四色定理の形式化“
この膨大な証明の為のCoqのライブラリとして開発された
⇨
証明の基本的なステップ“**タクティク**”をまとめた

用途

Coqで使用される言語
Gallinaによる**ラムダ計算の記述の簡略化**

SSReflectの標準ライブラリ

標準ライブラリ

- 関数に関するライブラリ ssrfun
- ブール代数に関するライブラリssrbool
- 同値関係などの決定的な型に関するライブラリ ssrnat
- リストに関するライブラリ seq
- 有限集合とその型を扱うライブラリ fintype

MathComp

標準ライブラリ

- 和, 積のライブラリ bigop
- 整数論のライブラリ div, prime
- 有限領域の関数のライブラリ finfun
- 有限集合のライブラリ finset
- 群のライブラリ fingroup
- 環や体のライブラリ ssralg
- 多項式のライブラリ poly, polydiv
- 行列のライブラリ matrix　etc...

用途

奇数位数定理の形式化

・集合論の形式化

・代数学の形式化

03

タクティク

スクリプトで主に使用される命令について

SSReflectのタクティク

move	ゴールや仮定の変数を導入したり、式を整理して証明を進める
apply	適用可能な補題や仮定を使って現在のゴールを証明する
by [ ]	ゴールが簡約可能（自明）である場合に自動的に証明する
rewrite	ゴールや仮定において、等式を使って式を書き換える
have	新しい補題や中間結果を導入し、それを証明した上で後続の証明に利用する

SSReflectのタクティクの利点

- Coq標準のタクティクよりも短く証明を記述できる
- 証明の流れを整理しやすい
- move, byは簡単な証明から複雑な証明まで頻繁に使用される

04

実際に問題を解く！

例題1. 偶数性の証明

例題1. 偶数性の証明

問題：**任意の自然数nに対して、２＊nは偶数であることを証明せよ**

偶数とは、ある自然数ｋに対してｋ＊２の形で表される数のことを指す

証明のアプローチ

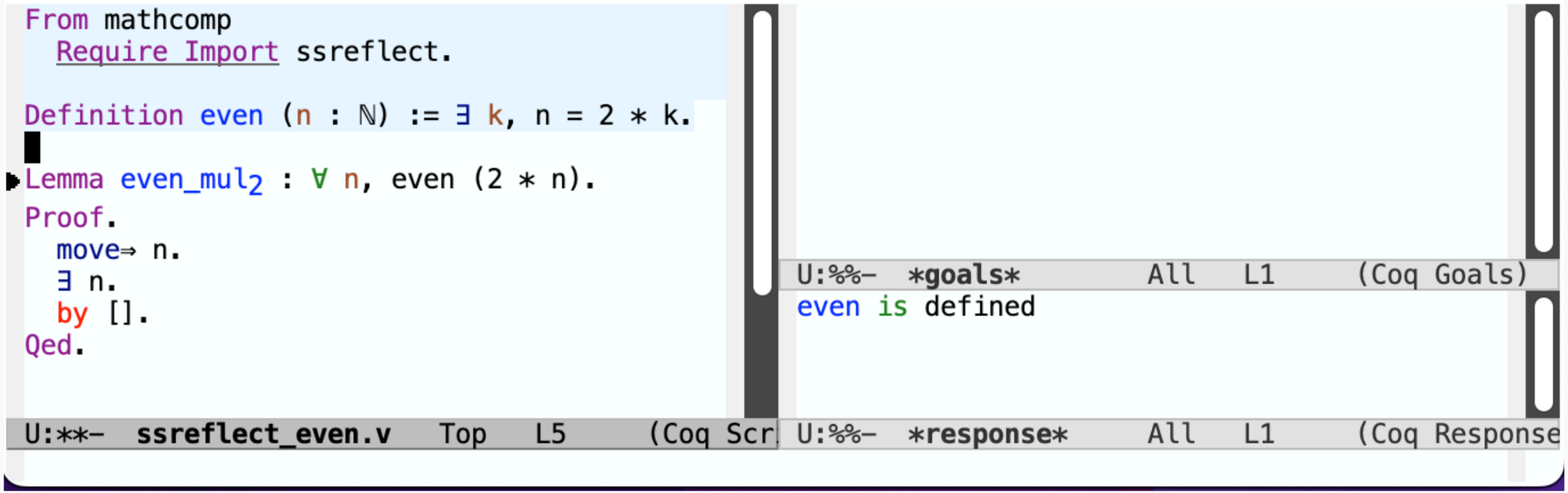
帰納法を用いて、以下の場合について証明する

- n=0の場合
- n=kの場合が成り立つと仮定した時のn=k+1の場合

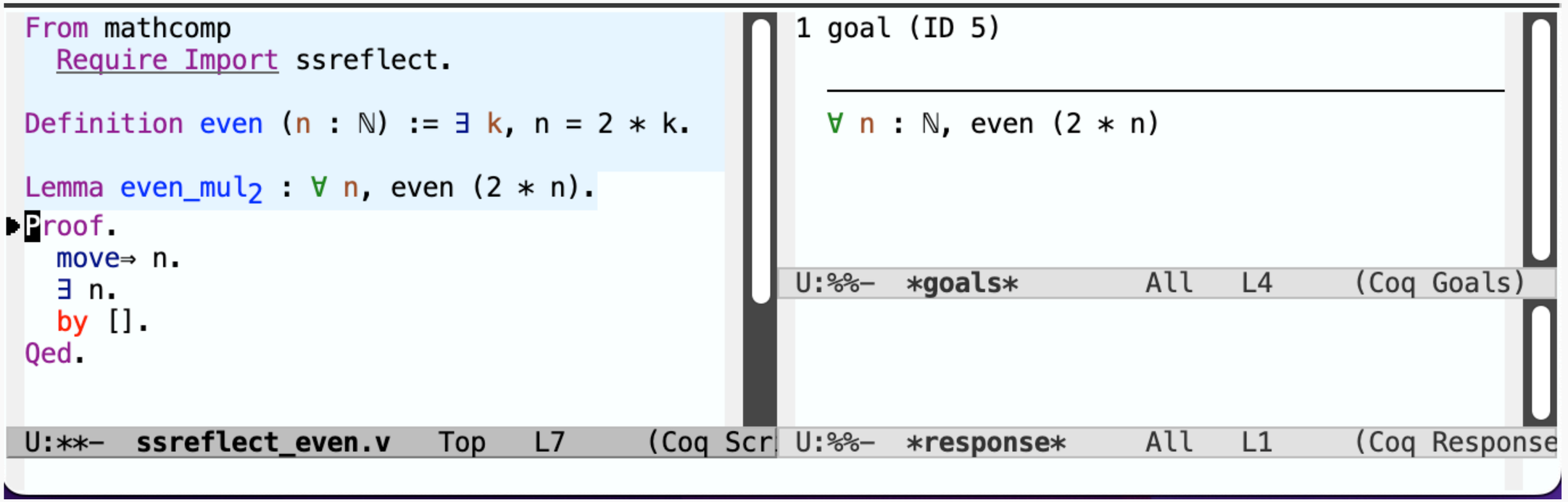
使用するタクティク

- moveを用いて、任意のnを仮定として導入する
- ∃ を使った存在証明
- by []を用いて、簡約化する

Coq上のコード（偶数の定義）

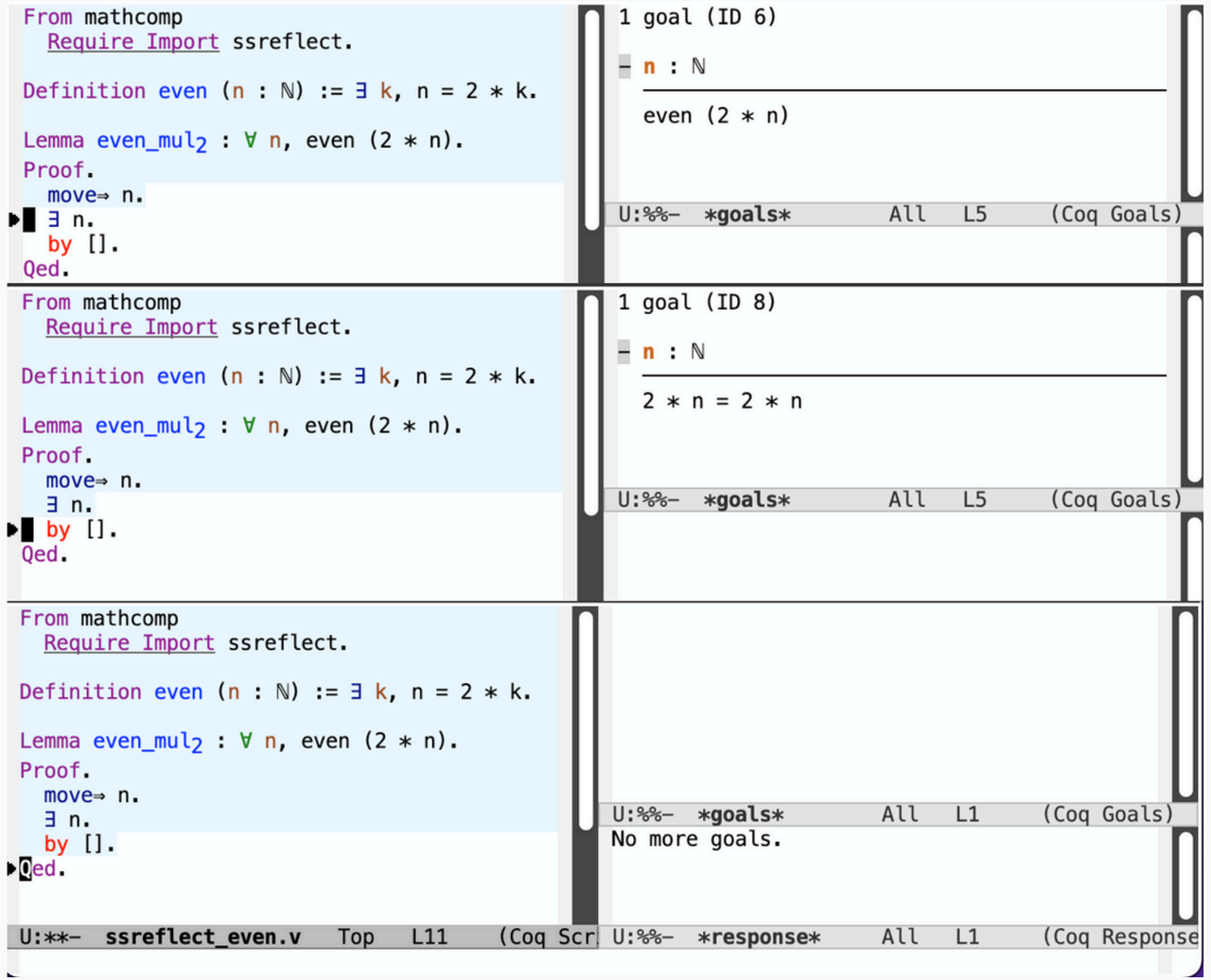


Coq上のコード（命題）


From mathcomp
Require Import ssreflect.
Definition even (n : ℕ) := ∃ k, n = 2 * k.
Lemma even_mul2 : ∀ n, even (2 * n).
Proof.
move⇒ n.
∃ n.
by [].
Qed.
U:**- ssreflect_even.v Top L7 (Coq Scr
1 goal (ID 5)
∀ n : ℕ, even (2 * n)
U:%%- *goals* All L4 (Coq Goals)
U:%%- *response* All L1 (Coq Response

Coq上のコード（証明）

- move=> n: 任意の自然数 n を導入
- ∃ n: 存在する k として n を指定
- by []: 自動で簡約して証明完了



結果

2 * nが偶数であることを形式的に証明した

この証明は、より複雑な整数の性質の証明の基礎になる

応用例：
奇数性の証明、整数論における性質の形式化 など

今後の研究について

双方向変換のレンズ則を形式化して証明する

- PutGet則
  - get(put s v) = v
- GetPut則
  - put(s, get s) = s


slidesgo