XXIX Semana do IME/UFG e VI Seminário de Pesquisa e Pós-Graduação do IME/UFG

Formalização de Teoremas em Assistentes de Prova

Section 2: Caso de estudo - Teoria de Grupos

Thaynara Arielly de Lima (IME) UFG
Mauricio Ayala-Rincón (CIC-MAT) UnB

Funded by FAPDF DE grant 00193.0000.2144/2018-81, CNPq Research Grant 307672/2017-4

Oct 6 -8 , 2021

Talk's Plan

1. Section 2
   - Specification of algebraic notions
   - Induction in PVS
   - Exercises - A case study on Group Theory

Closure in a group

```
G: VAR set[T]

closed?(G): bool = FORALL (x,y:(G)): member(x*y,G)

group?(G): bool = closed?(G) AND
                  associative?[(G)](*) AND
                  member(e,G) AND identity?[(G)](*)(e) AND
                  inv_exists?(G)
```

Conjecture power_closed in pred_algebra.pvs

For all group G, $y \in G$ and $n \in \mathbb{N}$ one can prove that $y^n = \underbrace{y * \ldots * y}_{n-times} \in G$.

A recursive function in PVS

$$\wedge(y,n) = \prod_{i=1}^{n} y, \text{ defined as } e \text{ for } n = 0$$

In PVS:

```
^(y : T, n : nat) : RECURSIVE T =
                    IF n = 0 THEN e
                    ELSE y * ^(y, n-1) ENDIF
                    MEASURE n
```

Type Correctness Conditions (TCCs)

The specification provides two conditions to be verified:

- **A TCC about the type of the argument in the recursive call**

```
% Subtype TCC generated (at line 52, column 22) for  n - 1
    % expected type  nat
caret_TCC1: OBLIGATION FORALL (n: nat): NOT n = 0 IMPLIES n - 1 >= 0;
```

- **A TCC that guarantees the termination of the recursive call**

```
% Termination TCC generated (at line 52, column 17) for  ^(y, n - 1)
caret_TCC2: OBLIGATION FORALL (n: nat): NOT n = 0 IMPLIES n - 1 < n;
```

Induction scheme: weak induction on naturals

power_closed:

```
|---
[1] FORALL(G : (group?), y : (G), n : nat) : member(∧(y, n), G)
Rule? (induct"n")
```

- Base case: power_closed.1

```
|---
[1] FORALL(G : (group?), y : (G)) : member(∧(y, 0), G)
```

- Inductive Step: power_closed.2

```
|---
[1] FORALLj :
    (FORALL(G : (group?), y : (G)) : member((y ∧ j), G)) IMPLIES
     (FORALL(G : (group?), y : (G)) : member((y ∧ (j + 1)), G))
```

Strong induction on naturals

Fibonacci Sequence

```
fibonacci(n:nat): RECURSIVE nat =
                    IF n <= 1 THEN n ELSE
                    fibonacci(n-1) + fibonacci(n-2)
                    ENDIF
                    MEASURE n
```

Conjecture fibonacci_exp_lim in fibonacci.pvs

fibonacci(n) $\leq 1.7^n$, for all $n \in \mathbb{N}$.

Induction scheme: strong induction on naturals

fibonacci_exp_lim:

```
|---
[1] FORALL(n : nat) : fibonacci(n) <= expt(1.7, n)
Rule? (measure-induct+ "n" "n")
```

↓

```
[-1] FORALL(y : nat) : y < x!1  IMPLIES  fibonacci(y) <= expt(1.7, y)
|---
[1] fibonacci(x!1) <= expt(1.7, x!1)
```

Induction scheme: strong induction on naturals

Base cases:

```
fibonacci_exp_lim:

    [-1] FORALL(y : nat) : y < x!1  IMPLIES  fibonacci(y) <= expt(1.7, y)
    |---
    [1] fibonacci(x!1) <= expt(1.7, x!1)
    Rule? (case - replace "x!1 = 0")

fibonacci_exp_lim.1:

    [-1] x!1 = 0
    [-2] FORALL(y : nat) : y < x!1  IMPLIES  fibonacci(y) <= expt(1.7, y)
    |---
    [1] fibonacci(0) <= expt(1.7, 0)
    Rule? (grind)

This completes the proof of fibonacci_exp_lim.1.
```

Induction scheme: strong induction on naturals

Base cases:

fibonacci_exp_lim.2:

[−1] FORALL(y : nat) : y < x!1 IMPLIES fibonacci(y) <= expt(1.7, y)

|---

[1] $x!1 = 0$

[2] fibonacci(x!1) <= expt(1.7, x!1)

Rule? (case − replace "x!1 = 1")

fibonacci_exp_lim.2.1:

[−1] x!1 = 1

[−2] FORALL(y : nat) : y < x!1 IMPLIES fibonacci(y) <= expt(1.7, y)

|---

[1] $1 = 0$

[2] fibonacci(1) <= expt(1.7, 1)

Rule? (grind)

This completes the proof of fibonacci_exp_lim.2.1.

Induction scheme: strong induction on naturals

Inductive Step:

fibonacci_exp_lim.2.2:

```
[-1] FORALL(y : nat) : y < x!1  IMPLIES  fibonacci(y) <= expt(1.7, y)
|---
[1] x!1 = 1
[2] x!1 = 0
[3] fibonacci(x!1) <= expt(1.7, x!1)
Rule? (expand "fibonacci" 3) (assert)

[-1] FORALL(y : nat) : y < x!1  IMPLIES  fibonacci(y) <= expt(1.7, y)
|---
[1] x!1 = 1
[2] x!1 = 0
[3] fibonacci(x!1 - 1) + fibonacci(x!1 - 2) <= expt(17/10, x!1)
Rule? (inst - cp  - 1  "x!1 - 1")
```

Induction scheme: strong induction on naturals

Inductive Step:

```
[-1] FORALL(y : nat) : y < x!1  IMPLIES  fibonacci(y) <= expt(1.7, y)
[-2] x!1 - 1 < x!1  IMPLIES  fibonacci(x!1 - 1) <= expt(17/10, x!1 - 1)
|---
[1] x!1 = 1
[2] x!1 = 0
[3] fibonacci(x!1 - 1) + fibonacci(x!1 - 2) <= expt(17/10, x!1)
Rule? (inst  -1  "x!1 - 2")
```

```
[-1] x!1 - 2 < x!1  IMPLIES  fibonacci(x!1 - 2) <= expt(17/10, x!1 - 2)
[-2] x!1 - 1 < x!1  IMPLIES  fibonacci(x!1 - 1) <= expt(17/10, x!1 - 1)
|---
[1] x!1 = 1
[2] x!1 = 0
[3] fibonacci(x!1 - 1) + fibonacci(x!1 - 2) <= expt(17/10, x!1)
Rule? (assert)
```

Induction scheme: strong induction on naturals

Inductive Step:

$[-1]$ $\texttt{fibonacci}(\texttt{x!1}-2) <= \texttt{expt}(17/10, \texttt{x!1}-2)$

$[-2]$ $\texttt{fibonacci}(\texttt{x!1}-1) <= \texttt{expt}(17/10, \texttt{x!1}-1)$

|---

$[1]$ $x!1 = 1$

$[2]$ $x!1 = 0$

$[3]$ $\texttt{fibonacci}(\texttt{x!1}-1) + \texttt{fibonacci}(\texttt{x!1}-2) <= \texttt{expt}(17/10, \texttt{x!1})$

Exercises - A case study on Group Theory

See the file `pred_algebra.pvs` in Exercises directory