SaDS HW 4

Inti Mendoza

Problem 5.1

After every usage of the method functions *getSize*, *push* and *pop*, the private element *size* is only updated correctly to reflect the current size of the stack in the functions *push* and *pop* where size is actually changed. Therefore $length(elements)$ will always equal *size*, as *size* is always correctly updated as a pointer to the last element of the stack (which happens to be the length of it as well).

$\therefore length(elements) == size$ is a good class invariant.

Problem 5.2

See next page.

Problem 5.3

Proof p:
Step case:

$$zero + n == n + zero$$

Due to $zero\_right$ and $zero\_left$, $zero + n \rightsquigarrow n$ and $n + zero \rightsquigarrow n$
$\rightarrow$ m == m $\rightsquigarrow true$
$\therefore true$
Induction step:
Goal to prove:

$$succ(m) + n == n + succ(m)$$

succ(m) + n $\rightsquigarrow succ(m + n)$	succ_left
succ(m + n) $\rightsquigarrow succ(n + m)$	Induction hypothesis
succ(n + m) $\rightsquigarrow n + succ(m)$	succ_right

$\therefore succ(m) + n == n + succ(m)$ (same vice versa).

Problem 5.2

```
class Date {
private:
    int year;
    int month;
    int day;

public:
Date(int year, int month, int day) {
    this->year  = year;
    this->month = month;
    this->day   = day;
}
~Date(){}

Date yesterday() {
    int year;
    int month;
    int day;
    int r_month = 0;
    int r_year  = 0;

    if(this->day == 1) {
        r_month = 1;
        day = getDay(this->month);
        if(this->month == 3) {
            day = 28;
        } else if(this->month == 5 || this->month == 7
            || this->month == 10 || this->month == 12) {
            day = 30;
        } else {
            day = 31;
        }
    } else {
        day = this->day - 1;
    }

    if(r_month == 1) {
        if(month == 1) {
            r_year = 1;
        } else {
            month = this->month - 1;
        }
    } else {
        month = this->month;
    }

    if(r_year == 1) {
        year = this->year - 1;
    } else {
        year = this->year;
    }

    Date yesterday = new Date(year, month, day);
    return yesterday;
}
```

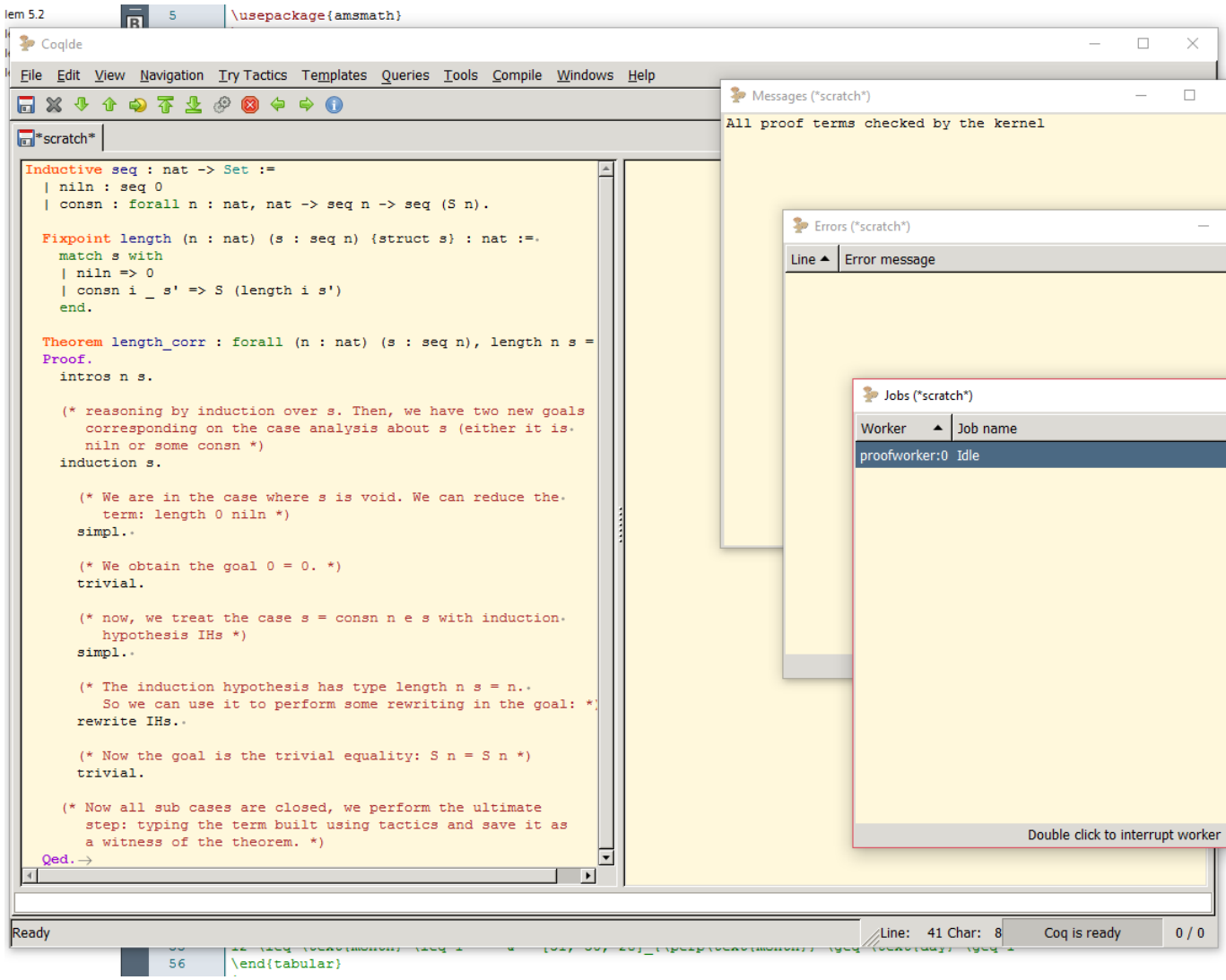
```
Date tomorrow() {
    int year;
    int month;
    int day;

    if(this->day == 31) {
        day = 1;
        if(this->month = 12) {
            month   = 1;
            year    = this->year + 1;
        } else {
            month   = this->month + 1;
            year    = this->year;
        }
    } else {
        day     = this->day + 1;
        month   = this->month;
        year    = this->year;
    }
    Date tomorrow = new Date(year, month, day);
    return tomorrow;
}
};
```

Class invariant:

$12 \leq \text{month} \leq 1 \mid [31, 30, 28]_{\perp\text{month}} \geq \text{day} \geq 1$

Problem 5.4


lem 5.2
5
\usepackage{amsmath}
CoqIde
File Edit View Navigation Try Tactics Templates Queries Tools Compile Windows Help
scratch
Inductive seq : nat -> Set :=
| niln : seq 0
| consn : forall n : nat, nat -> seq n -> seq (S n).
Fixpoint length (n : nat) (s : seq n) {struct s} : nat :=
match s with
| niln => 0
| consn i _ s' => S (length i s')
end.
Theorem length_corr : forall (n : nat) (s : seq n), length n s =
Proof.
intros n s.
(* reasoning by induction over s. Then, we have two new goals
corresponding on the case analysis about s (either it is
niln or some consn *)
induction s.
(* We are in the case where s is void. We can reduce the
term: length 0 niln *)
simpl.
(* We obtain the goal 0 = 0. *)
trivial.
(* now, we treat the case s = consn n e s with induction
hypothesis IHs *)
simpl.
(* The induction hypothesis has type length n s = n.
So we can use it to perform some rewriting in the goal: *)
rewrite IHs.
(* Now the goal is the trivial equality: S n = S n *)
trivial.
(* Now all sub cases are closed, we perform the ultimate
step: typing the term built using tactics and save it as
a witness of the theorem. *)
Qed.
Messages (*scratch*)
All proof terms checked by the kernel
Errors (*scratch*)
Line
Error message
Jobs (*scratch*)
Worker
Job name
proofworker:0 Idle
Double click to interrupt worker
Ready
Line: 41 Char: 8
Coq is ready
0 / 0
56
\end{tabular}