第 1 讲: 算法的正确性

姓名： 林凡琪　　**学号：** 211240042

评分： ______　　**评阅：** ______

2022 年 2 月 23 日

请独立完成作业，不得抄袭。
若得到他人帮助, 请致谢。
若参考了其它资料，请给出引用。
鼓励讨论，但需独立书写解题过程。

1 作业 (必做部分)

题目 1 (DH Problem 5.8: *rev*(*X*))

Algorithm 1 reverse

解答：

```
procedure Rev(S)
    X ← S
    Y ← ∅
    while X ≠ ∅ do
        Y ← last(X).Y
        X ← allbutlast(X)
    end while
    return Y
end procedure
```

proof:
1.
We can know that, if and only if (2)→(2)is not an infinete loop, then the algorithm can stop properly.
The convergent that works in our case is simply the length of the string X. Each time the loop is traversed, X is made shorter by precisely one symbol, since it becomes the all-but-last of its previous value. However, its length cannot be less than 0 because when X is of length 0 (that is, it becomes the empty string) the loop is not traversed further and the algorithm terminates.
2.
assertion (1): S is a string.
assertion (2): S = X . rev(Y)
assertion (3): S = rev(Y)

3 hops:(1)→(2),(2)→(2),(2)→(3)
For (1)→(2):
Since $X \leftarrow S$ and $Y \leftarrow \emptyset$, so X . rev(Y) = S . $\emptyset$ = S.
For (2)→(2):
Define Z($S = X.Z$),if $|Z| = 0$, we can know that $Y = rev(Z)$
Assume that when $|Z| = n$, $Y = rev(Z)$ is true.
Then after that $Y = last(X).rev(Z)$, $Z = Z.last(X)$ so it remains that $Y = rev(Z)$.
For (2)→(3)
Untill $X = \emptyset$, which means $Z = S$ For that $Y = rev(Z)$, so $Y = rev(S)$, it also means $S = rev(Y)$
So the algorithm is partially correct.

This conclude the proof that the reverse algorithm is totally correct.

题目 2 (DH Problem 5.9: *equal*(X, Y))

Algorithm 2 equal

解答:

procedure EQUAL(X, Y)
  $Xori = X$
  $Yori = Y$
  $S \leftarrow \emptyset$
  $T \leftarrow \emptyset$
  while $X \neq \emptyset$ and $Y \neq \emptyset$ and eq($last(X), last(Y)$) **do**
    $S = last(X).S$
    $T = last(Y).T$
    X $= all - but - last(X)$
    $Y = all - butlast(Y)$
  end while
  if $X = \emptyset$ and $Y = \emptyset$ **then**
    return true
  else
    return false
  end if
end procedure

proof:
1.
We can know that, if and only if (2)→(2)is not an infinete loop, then the algorithm can stop properly.
The convergent that works in our case is simply the length of the string X and Y. Each time the loop is traversed, X and Y are made shorter by precisely one symbol, since thet become the all-but-last of their previous value. However, their length cannot be less than 0 because when X or Y is of length 0 (that is, thet become the empty strings) the loop is not traversed further and the algorithm terminates.
2.
assertion(1):X and Y are strings.
assertion(2):X. S = Xori and Y. T= Yori and S = T
assertion(3):If and only if $X = \emptyset$ and $Y = \emptyset$, eq(X, Y) = true.
3 hops:(1)→(2),(2)→(2),(2)→(3)

For (1)→(2):Since that $X = Xori$, $Y = Yori$,$S \leftarrow \emptyset$ and $T \leftarrow \emptyset$, so $X.S = Xori.\emptyset = Xori$ and $Y.T = Yori.\emptyset = Yori$ and $S = T = \emptyset$
For (2)→(2):We assumpt that before a traversation, $X.S = Xori$ and $Y.T = Yori$ hold. Then in the new traversation, (we can note the X, Y before the traversation as $X0$ and $Y0$, $S = last(X0).S$,$X = all - but - last(X0)$, so $X.S = all - but - last(X0).last(X0).S = X0.S = Xori$.The same, we can get that $Y = Yori$.
For that $S = last(X0).S$, $T = last(Y0).T$ and $eq(last(X0, Y0))$, we can know that $S = T$
So after the loop, (2) still holds.
For (2)→(3) If $X = \emptyset$ and $Y = \emptyset$, then $S = Xori$ and $T = Yori$.Since in (2) $S = T$, So $Xori = Yori$, then should return true.
So the algorithm is totally correct.

题目 3 (DH Problem 5.10: *Pal1*(S))

解答:
(a)
proof:
1.
There are 2 steps in the algorithm. Both of them have been prooved that they can stop properly, and there are no loop in the algorithm, so we can know the algorithm can stop properly.
2.
assertion(1):S is a string.
assertion(2):Y is reversed S.
assertion(3):If S = Y, S is palindrome.
2 hogs:(1) to (2) and (2) to (3).
For(1) to (2): Since S is a string, this satisfy the reqiure of $rev(S)$, which has been prooved to be totally correct, so Y is reversed S.
For(2) to (3): According the definition of palindrome (a string that is the same when read forwards and backwards), if Y = S, then S meets the standard of palindrome, so S is palindrome.
(b) “大型的新型软件，不再是一行行之星基本运算的代码构成，往往是过程调用的组合。这样的算法（代码）的终止性取决于各个被调用的过程的终止性。而传统代码的计算终止性，取决于算法中循环的终止性，传统代码的终止性模型缺省认为任意一行代码都是单位时间内任意一行代码都是单位时间内一定完成并给出正确结果的。”
————张三

题目 4 (DH Problem 5.12: *Pal2*(S))

解答:
(a)Proof:
Before state assertions, we do some definitions. Set 2 string Y and Z, $Y = Z = \emptyset$ in the beginning, and insert 2 sentences before line 4: $Y = head(X); Z = last(X)$
assertion(1):S is a string, E is true.
assertion(2):if E is true, then note the previous X as X1, X1 = Y.X.Z
assertion(3):if S is a palindrome, pal2(S) is true.

3 hops:(1)→(2),(2)→(2) and (2)→(3)
For (1) to (2): since X = S and E = true, then $Y.X.Z = \emptyset.X.\emptyset = X1$, then (2) holds.
For (2) to (2): we have X1 = Y · X · Z, then after the constructions we have that X1 = head(X1) · head(X) · X2 · last(X) · last(X1) = head(X1) · X · last(X1), so (2)→(2) holds.
For (2) to (3): if S is a palindrome, Y · X · Z, X = ∅, then pal2(S) = true.
(b)
Disproof:
if S isn' t palindrome, then E = false, but the loop cannot end because the length cannot decrease and X cannot be equal to ∅.

题目 5 (DH Problem 5.14 (a, b): *Pal4*(S))
Note: You don't have to consider *Pal3* in Problem 5.13.

解答:
(a)
(b)

Algorithm 3 equal

procedure PAL4(S)
  $X \leftarrow S$
  if $head(X) = last(X)$ **then**
    $E \leftarrow true$
  else$E \leftarrow false$
  end if
  while $X \neq \emptyset$ and $E = true$ **do**
    $Y \leftarrow tail(X)$
    if eq(head(X), last(Y)) **then**
      $X \leftarrow all - but - last(Y)$
    else$E \leftarrow false$
    end if
  end while
  return E
end procedure

Proof:
1.The algorithm in Q4 can not stop properly, as it doesn't consider well enough the beginning set of E. So in Pal4, I add a judging sentence so that solve the termination problem of the algorithm in Q4. So Pal4 can stop properly.
2.assertion(1): Pal3 is partially correct.
assertion(2): If $head(X) \neq last(X)$, then S is not a palindrome.
assertion(1) has been prooved in Q4.
assertion(2): According to the definition of palindrome. If S is a palindrome, then $head(X) = last(X)$, so If $head(X) \neq last(X)$, then S is not a palindrome.
All in all, pal4 is totally correct. (c)
Because when we use pal1 to judge whether S is a palindrome, we need to go over the whole string for 2 times, which means we need to traverse for 2n times, but by pal4, we only need to go over one string, so we can save much time.

---

2 作业 (选做部分)

题目 1 (Euclid's GCD Algorithm)
请证明 Euclid 算法 (迭代或递归版本) 的完全正确性。

解答:

---

3 Open Topics

Open Topics 1 (Insertion Sort and Dafny)
请证明 Insertion Sort 的正确性。要求证明严谨, 推荐在 Dafny 中实现并验证 Insertion Sort 的正确性。

参考资料:

- Insertion sort @ wiki
- Dafny
- Play with Dafny@rise4fun(最近该网站在维护，不可达，可以尝试直接在 VS Code 安装 Dafny)

Open Topics 2 (Cyclic Hanoi Problem)
介绍 Cyclic Hanoi 问题与算法，并证明算法的正确性。

- Tower of Hanoi @ wiki
- Chapter 5 of DH

4 反馈