2-1 The Correctness of Algorithms

魏恒峰

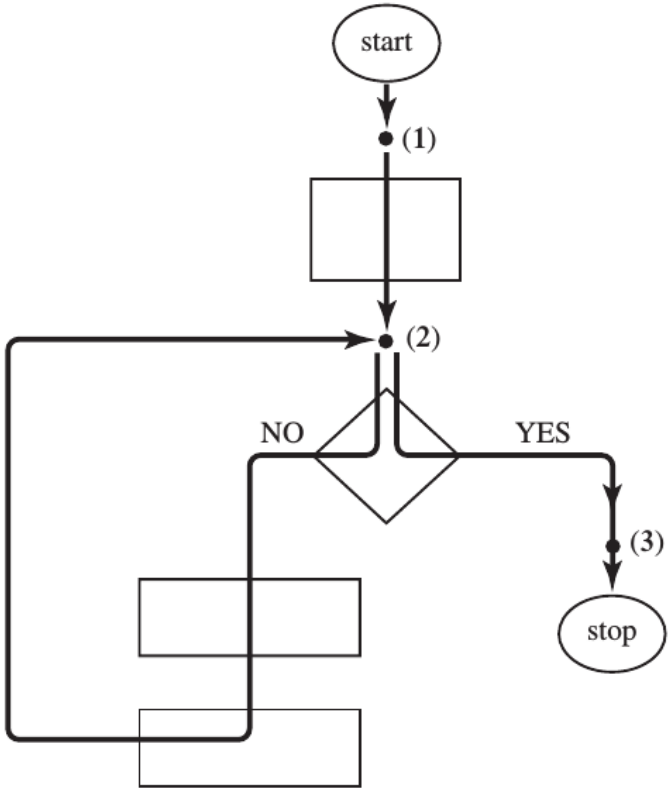
hfwei@nju.edu.cn

2018 年 03 月 12 日

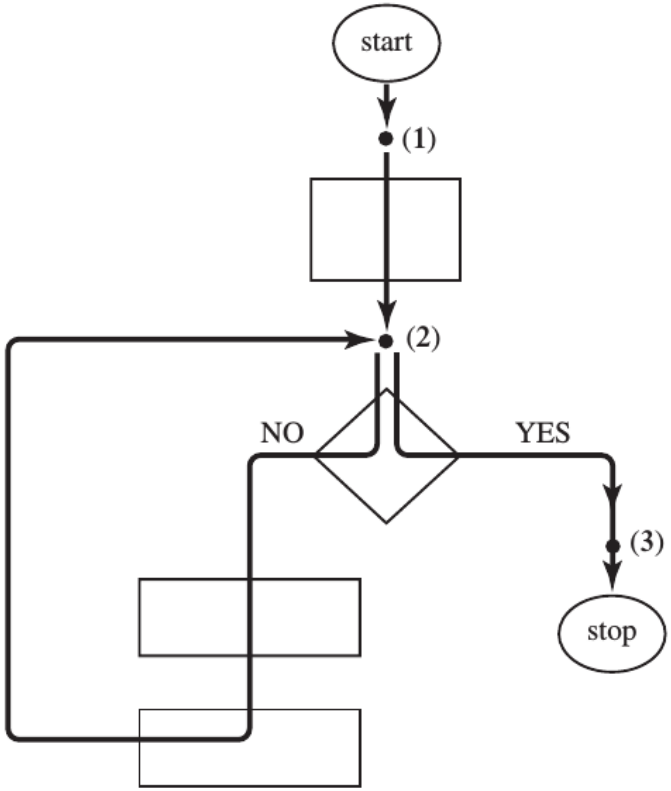
学生反馈

Q : Assertion, Invariant, Loop invariant 之间是什么关系?

Q : How to prove a loop partially correct?

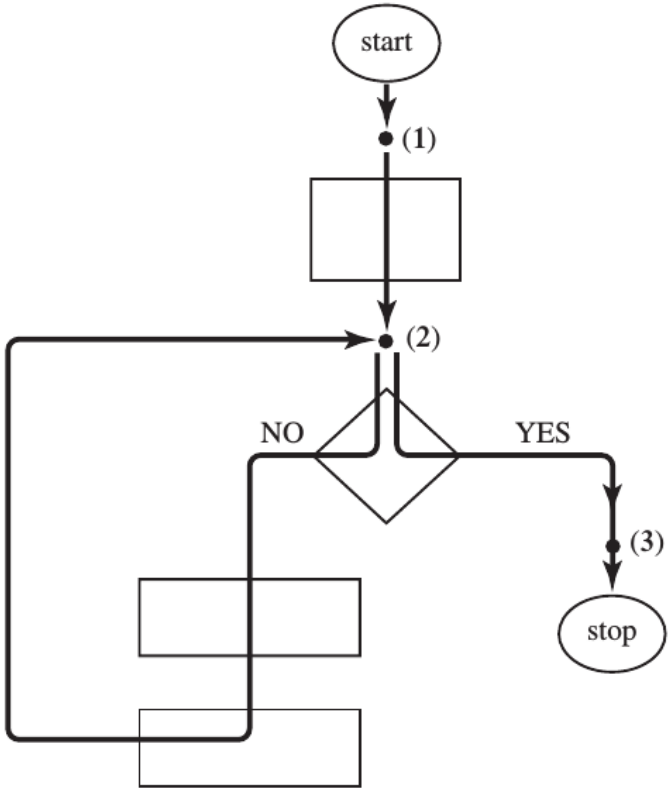


Q : How to prove a loop partially correct?



$\{P\}$ `loop` $\{Q\}$

Q : How to prove a loop partially correct?



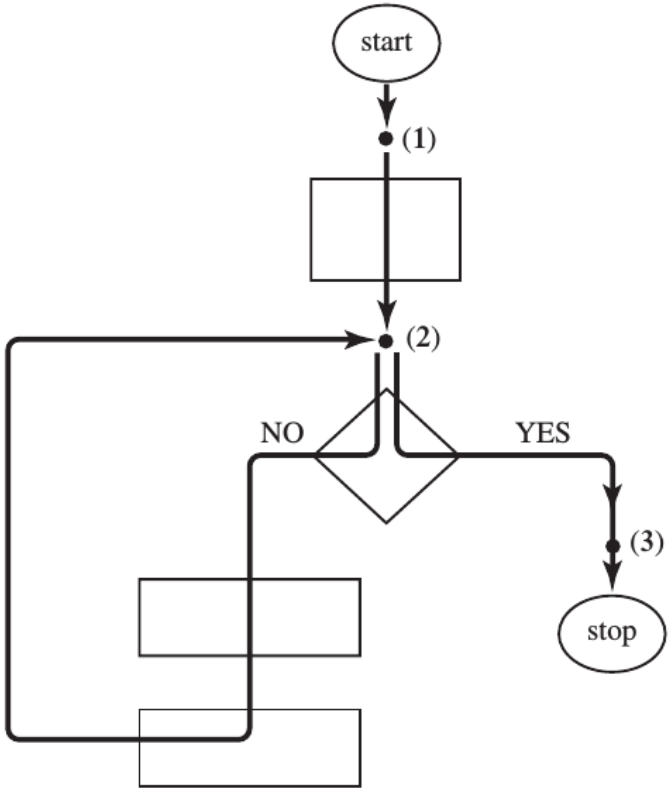
$\{P\}$ `loop` $\{Q\}$

(1) $\{P\}$ `init` $\{I\}$

(2) $\{I \wedge C\}$ `body` $\{I\}$

(3) $\{I \wedge \neg C\} \Longrightarrow \{Q\}$

Q : How to prove a loop partially correct?



$\{P\}$ `loop` $\{Q\}$

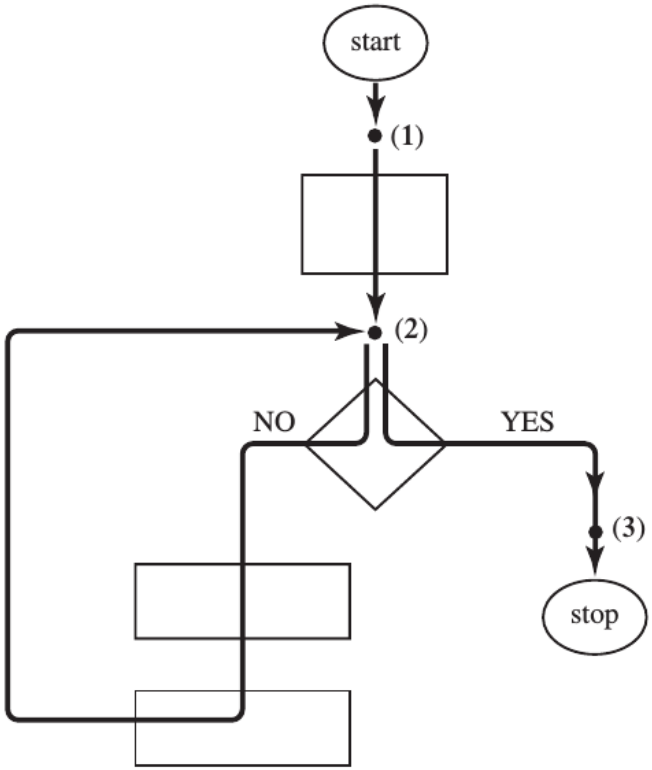
(1) $\{P\}$ `init` $\{I\}$

(2) $\{I \wedge C\}$ `body` $\{I\}$

(3) $\{I \wedge \neg C\} \Longrightarrow \{Q\}$

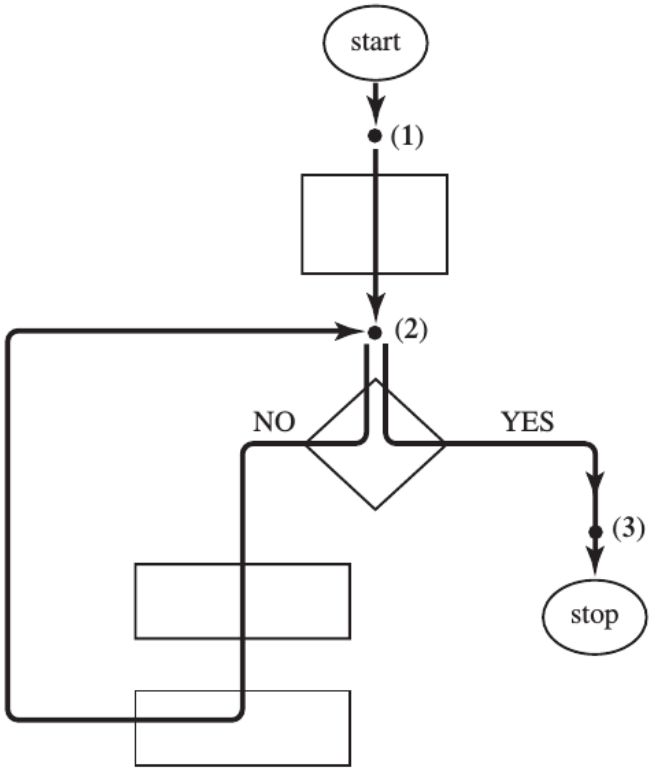
I is *before* the loop.

Q : How to prove a loop totally correct?



$D(X)$

Q : How to prove a loop totally correct?



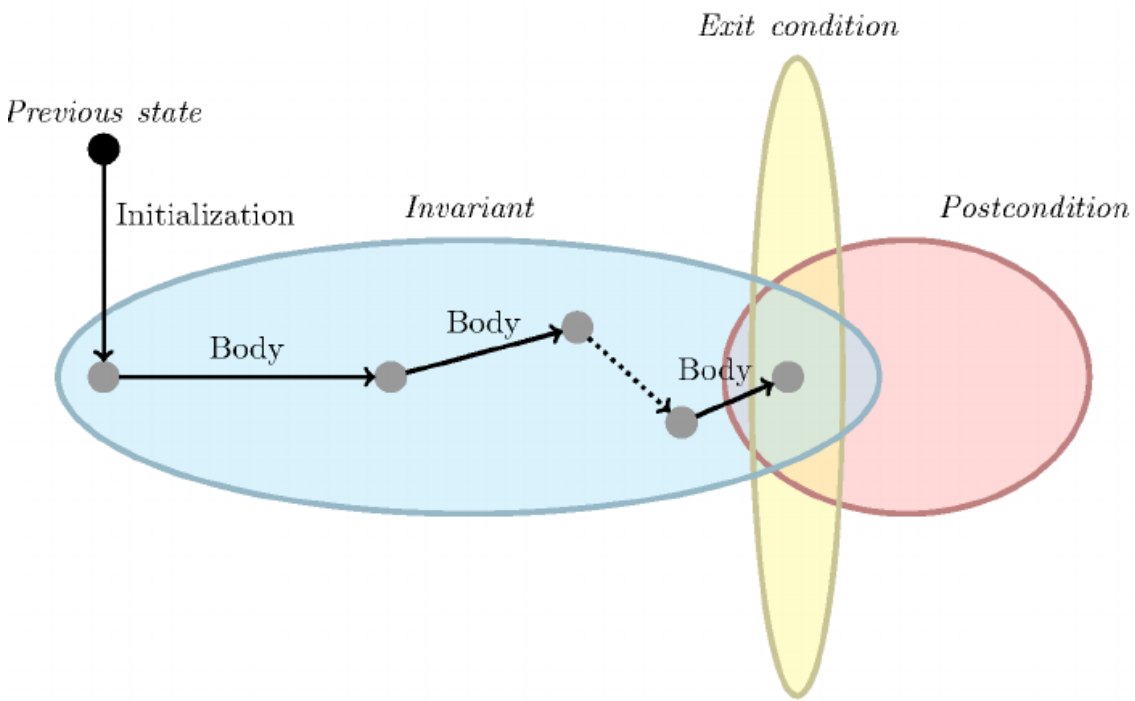
$$D(X)$$

(1) $\{I \wedge C\}$ `body` $\{D(X') < D(X)\}$

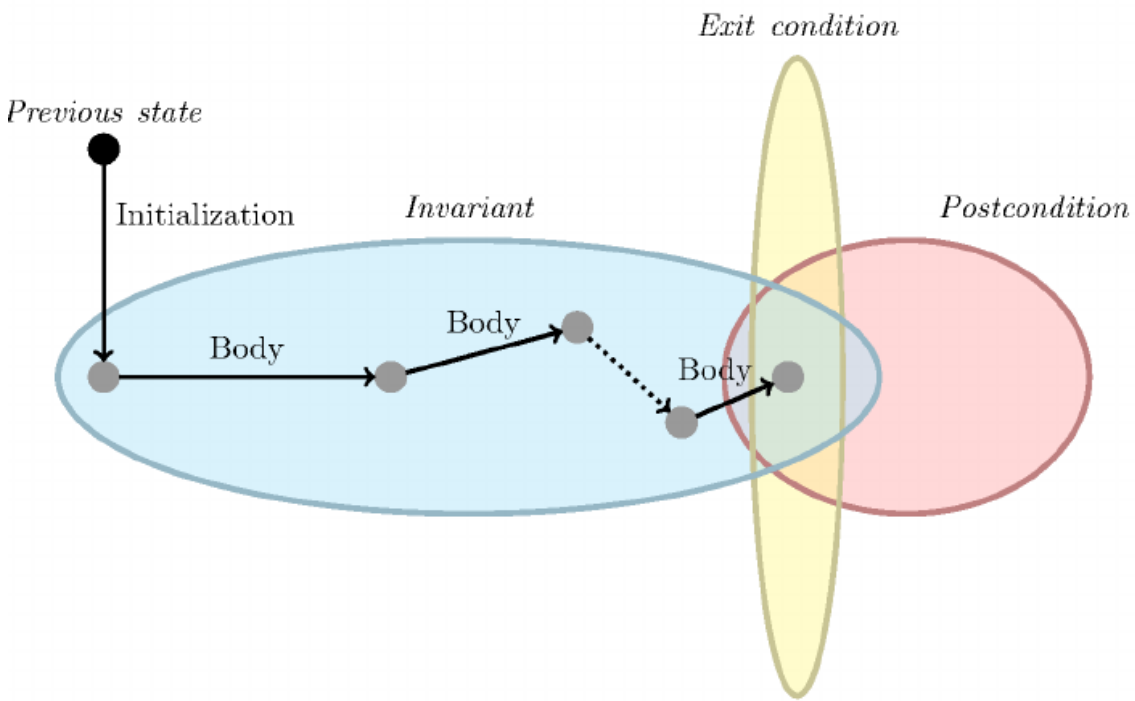
(2) $\{I \wedge D(X) = \min\} \implies \neg C$

Q : How to develop loop invariants?

Q : How to develop loop invariants?

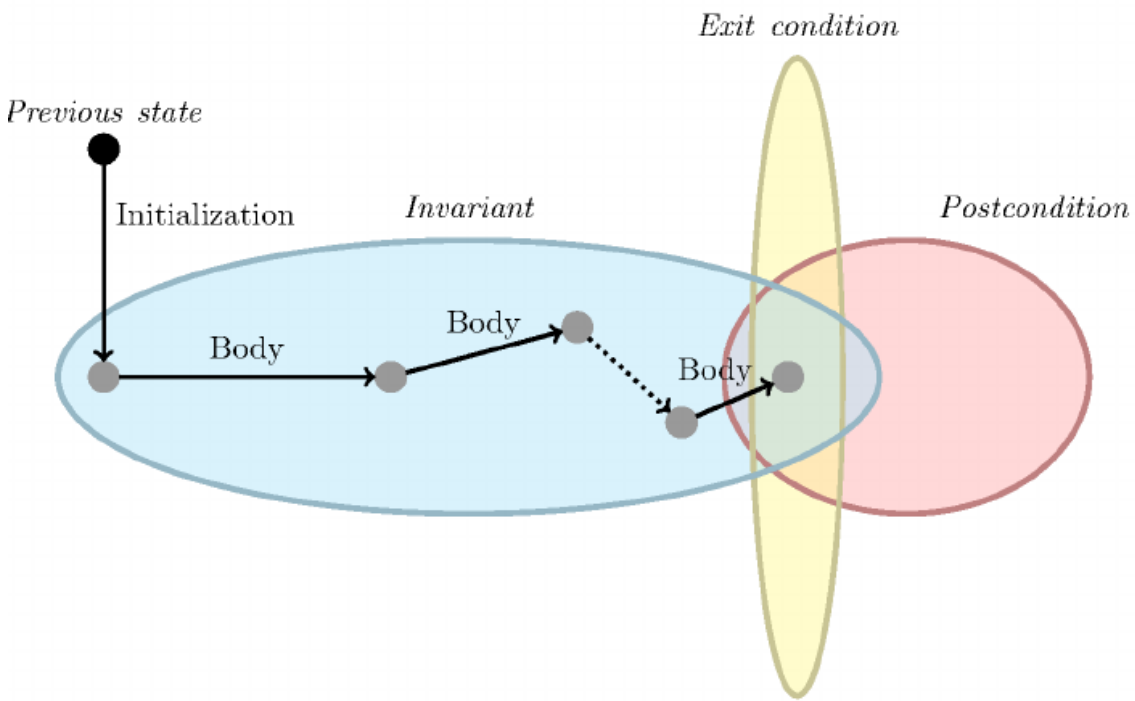


Q : How to develop loop invariants?



$$I \equiv \Big(\texttt{totalWork} = \texttt{workDone} + \texttt{workToDo}\Big)$$

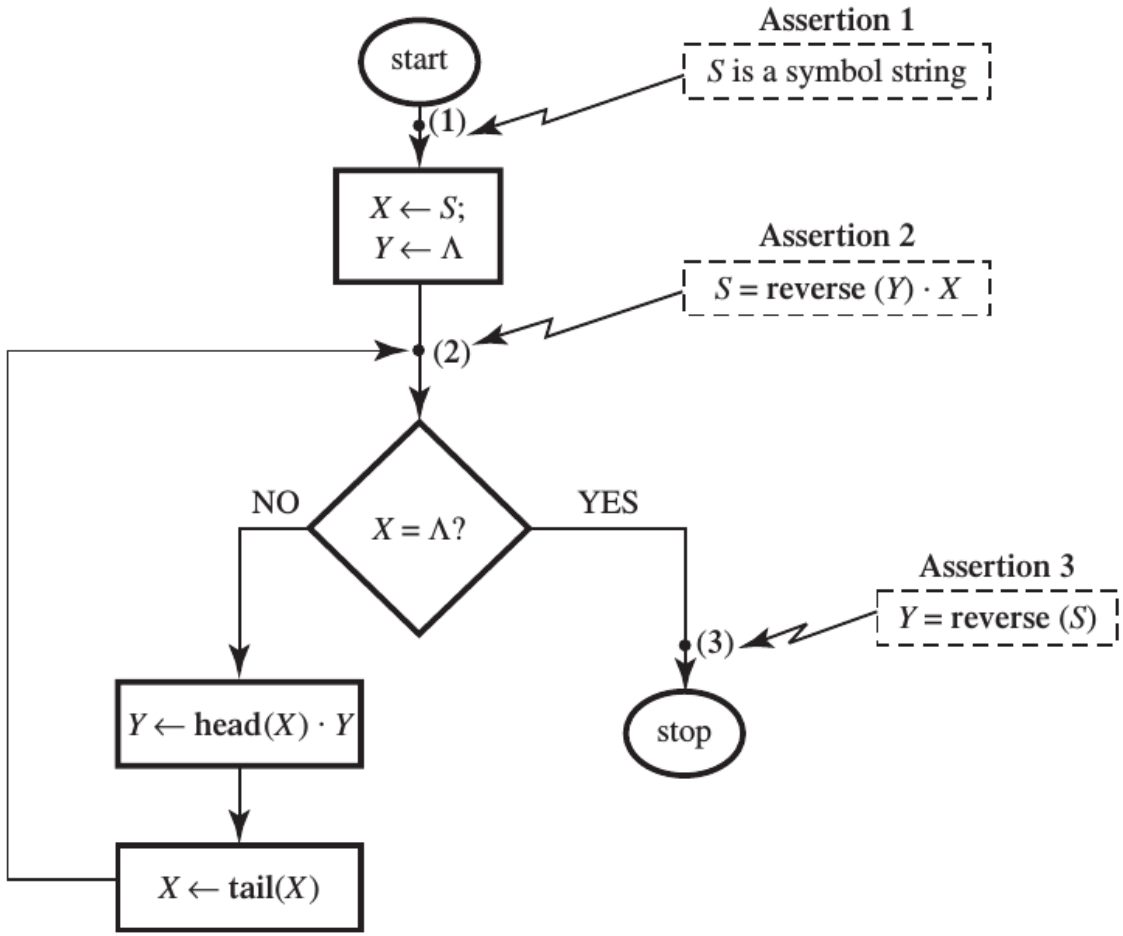
Q : How to develop loop invariants?



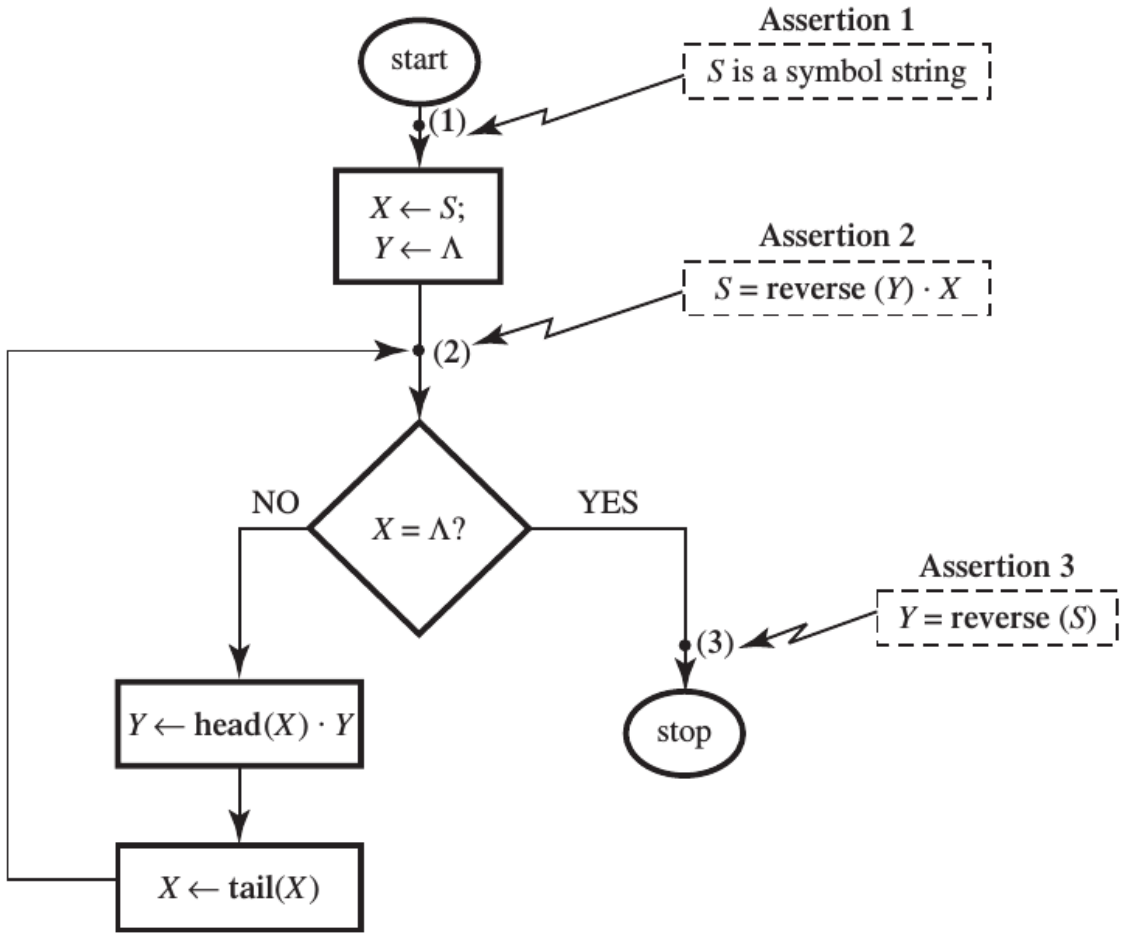
$$I \equiv \Big(\texttt{totalWork} = \texttt{workDone} + \texttt{workToDo}\Big)$$

$$\texttt{workDone} \xLeftarrow{\texttt{data}} \texttt{workToDo}$$

Reverse(S)


start
(1)
$X \leftarrow S$;
$Y \leftarrow \Lambda$
(2)
$X = \Lambda$?
NO
YES
$Y \leftarrow \text{head}(X) \cdot Y$
$X \leftarrow \text{tail}(X)$
(3)
stop
Assertion 1
S is a symbol string
Assertion 2
$S = \text{reverse}(Y) \cdot X$
Assertion 3
$Y = \text{reverse}(S)$

Reverse(S)



$$I \equiv \Big(\texttt{reverse}(S) = \texttt{reverse}(X) \cdot Y\Big)$$

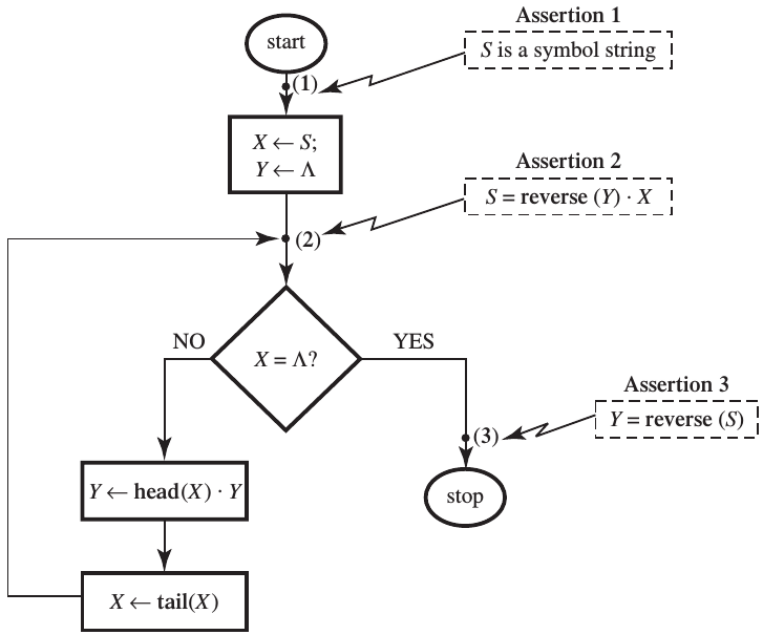
of invariants (DH Problem 5.6)

(a) Structure of $\texttt{Reverse}(X)$

(b) Only two invariants

(c) Sufficent #invariants for two loops

(d) Necessary #invariants for two loops



equal(S_1, S_2) (Problem 5.9)

- head(X)
- tail(X)
- last(X)
- all-but-last(X)
- eq(s, t)

```
while X ≠ Λ ∧ Y ≠ Λ
  if eq(head(X), head(Y))
    X ← tail(X)
    Y ← tail(Y)
  else
    return false

if X = Λ ∧ Y = Λ
  return true
return false
```

```
X ← S_1
Y ← S_2

E ← ⊤

while X ≠ Λ ∧ Y ≠ Λ ∧ E = ⊤
  if eq(head(X), head(Y))
    X ← tail(X)
    Y ← tail(Y)
  else
    E ← ⊥

if ¬(X = Λ ∧ Y = Λ)
  E ← ⊥

return E
```

```
while X ≠ Λ ∧ Y ≠ Λ
  if eq(head(X), head(Y))
    X ← tail(X)
    Y ← tail(Y)
  else
    return false

if X = Λ ∧ Y = Λ
  return true
return false
```

```
X ← S1
Y ← S2

E ← ⊤

while X ≠ Λ ∧ Y ≠ Λ ∧ E = ⊤
  if eq(head(X), head(Y))
    X ← tail(X)
    Y ← tail(Y)
  else
    E ← ⊥

if ¬(X = Λ ∧ Y = Λ)
  E ← ⊥
// S1 = S2 ⟺ E = ⊤

return E
```

```
while X ≠ Λ ∧ Y ≠ Λ
    if eq(head(X), head(Y))
        X ← tail(X)
        Y ← tail(Y)
    else
        return false

if X = Λ ∧ Y = Λ
    return true
return false
```

```
X ← S_1
Y ← S_2

E ← ⊤
// S_1 = S_2 ⟺ E = ⊤ ∧ X = Y
while X ≠ Λ ∧ Y ≠ Λ ∧ E = ⊤
    if eq(head(X), head(Y))
        X ← tail(X)
        Y ← tail(Y)
    else
        E ← ⊥

if ¬(X = Λ ∧ Y = Λ)
    E ← ⊥
// S_1 = S_2 ⟺ E = ⊤

return E
```

肖 江

$X \leftarrow S_1$
$Y \leftarrow S_2$

$E \leftarrow \top$
while $X \neq \Lambda \wedge E == \top$
  if $Y == \Lambda$
    $E \leftarrow \bot$
  else if $\mathtt{eq}(\mathtt{head}(X), \mathtt{head}(Y))$
      $X \leftarrow \mathtt{tail}(X)$
      $Y \leftarrow \mathtt{tail}(Y)$
  else
    $E \leftarrow \bot$

if $Y \neq \Lambda$
  $E \leftarrow \bot$

return E

Pal1(S) (Problem 5.10)

```
Y ← rev(S)
return equal(S, Y)
```

```
Y ← rev(S)

E ← equal(S, Y)

return E
```

(a) Total correctness of Pal1

(b) Termination of Pal1

Pal1(S) (Problem 5.10)

```
Y ← rev(S)
return equal(S, Y)
```

```
Y ← rev(S)

E ← equal(S, Y)
// isPal(S) ⟺ E = ⊤
return E
```

(a) Total correctness of Pal1

(b) Termination of Pal1

Pal1(S) (Problem 5.10)

```
Y ← rev(S)
return equal(S, Y)
```

```
Y ← rev(S)
// Y = reverse(S)
E ← equal(S, Y)
// isPal(S) ⟺ E = ⊤
return E
```

(a) Total correctness of Pal1

(b) Termination of Pal1

Pal1(S) (Problem 5.10)

```
Y ← rev(S)
return equal(S, Y)
```

```
Y ← rev(S)
// Y = reverse(S)
E ← equal(S, Y)
// isPal(S) ⟺ E = ⊤
return E
```

(a) Total correctness of Pal1

(b) Termination of Pal1

$$\Big(Y = \texttt{reverse}(S) \wedge E = \texttt{equal}(S, Y)\Big) \Longrightarrow \Big(\texttt{isPal}(S) \iff E = \top\Big)$$

Pal2(S) (Problem 5.12)

$X \leftarrow S$
$E \leftarrow \top$

while $X \neq \Lambda$
  if $\texttt{eq}(\texttt{head}(X), \texttt{last}(X))$
    $X \leftarrow \texttt{all-but-last}(\texttt{tail}(X))$
  else
    $E \leftarrow \bot$

return E

Pal2(S) (Problem 5.12)

$X \leftarrow S$
$E \leftarrow \top$

while $X \neq \Lambda \ \wedge E == \top$
  if $\texttt{eq}(\texttt{head}(X), \texttt{last}(X))$
    $X \leftarrow \texttt{all-but-last}(\texttt{tail}(X))$
  else
    $E \leftarrow \bot$

return E

Pal2(S) (Problem 5.12)

$X \leftarrow S$
$E \leftarrow \top$

// $I \equiv \texttt{isPal}(S) \iff E = \top \wedge \texttt{isPal}(X)$
while $X \neq \Lambda \ \wedge E == \top$
  if $\texttt{eq}(\texttt{head}(X), \texttt{last}(X))$
    $X \leftarrow \texttt{all-but-last}(\texttt{tail}(X))$
  else
    $E \leftarrow \bot$

return E

Pal2(S) (Problem 5.12)

$X \leftarrow S$
$E \leftarrow \top$

// $I \equiv \texttt{isPal}(S) \iff E = \top \wedge \texttt{isPal}(X)$
while $X \neq \Lambda \ \wedge E == \top$
  if $\texttt{eq}(\texttt{head}(X), \texttt{last}(X))$
    $X \leftarrow \texttt{all-but-last}(\texttt{tail}(X))$
  else
    $E \leftarrow \bot$
// $F \equiv \texttt{isPal}(S) \iff E = \top$

return E

Pal2(S) (Problem 5.12)

$X \leftarrow S$
$E \leftarrow \top$

// $I \equiv \texttt{isPal}(S) \iff E = \top \wedge \texttt{isPal}(X)$
while $X \neq \Lambda \ \wedge E == \top$
  if $\texttt{eq}(\texttt{head}(X), \texttt{last}(X))$
    $X \leftarrow \texttt{all-but-last}(\texttt{tail}(X))$
  else
    $E \leftarrow \bot$
// $F \equiv \texttt{isPal}(S) \iff E = \top$

return E

$$I \Longrightarrow F$$


Thank
You!