

# Assignment 3: Derivation

Maxwell Bo 43926871

May 17, 2017

1. (a)  $n$  is a **value** parameter.  $m$  is a **result** parameter.
- (b) Our post-condition,  $mr\text{run}(A, n_0, m)$ , expands to  $lrun(A, n_0, m) \wedge (m < A.\text{len} \Rightarrow A_{n_0} \neq A_m)$ . This satisfies the form  $Q_1 \wedge Q_2$ .  $Q_1$  was chosen as the invariant. Thus,

$$inv \triangleq lrun(A, n_0, m)$$

(c) Let

$$\begin{aligned} pre &\triangleq lrun(A, n, n+1) \\ post &\triangleq mr\text{run}(A, n_0, m) \end{aligned}$$

s.t.

$$\begin{aligned} &n, m : [pre, post] \\ \sqsubseteq &\{ \text{Composition: middle predicate is } inv \} \\ &n, m : [pre, inv]; \quad n, m : [inv, post] \\ \sqsubseteq &\{ \text{Assignment: } pre \Rightarrow inv[m \setminus n+1] \} \\ &m := n+1; \quad n, m : [inv, post] \end{aligned}$$

$\therefore$

$$\begin{aligned} inv[m \setminus n+1] &\equiv lrun(A, n_0, m)[m \setminus n+1] \\ &\equiv lrun(A, n_0, n+1) \end{aligned}$$

$\therefore$

$$lrun(A, n, n+1) \Rightarrow lrun(A, n_0, n+1)$$

Let

$$guard \triangleq (m < A.\text{len} \wedge A_{n_0} = A_m)$$

s.t.

$$\begin{aligned} \sqsubseteq &\{ \text{Strengthen post: } inv \wedge \neg guard \Rightarrow post \} \\ &m := n+1; \quad n, m : [inv, inv \wedge \neg guard] \end{aligned}$$

$\therefore$

$$\begin{aligned} &inv \wedge \neg guard \Rightarrow post \\ \equiv &\{ \text{Expansion of definitions} \} \\ &lrun(A, n_0, m) \wedge \neg(m < A.\text{len} \wedge A_{n_0} = A_m) \Rightarrow mr\text{run}(A, n_0, m) \\ \equiv &\{ \text{Expansion of functions} \} \end{aligned}$$

$$\begin{aligned}
& lrun(A, n_0, m) \wedge \neg(m < A.len \wedge A_{n_0} = A_m) \Rightarrow lrun(A, n_0, m) \wedge (m < A.len \Rightarrow A_{n_0} \neq A_m) \\
\equiv & \{ \text{De Morgan's law - negation of conjunction} \} \\
& lrun(A, n_0, m) \wedge (\neg(m < A.len) \vee \neg(A_{n_0} = A_m)) \Rightarrow lrun(A, n_0, m) \wedge (m < A.len \Rightarrow A_{n_0} \neq A_m) \\
\equiv & \{ P \Rightarrow Q \equiv \neg P \vee Q \} \\
& lrun(A, n_0, m) \wedge (\neg(m < A.len) \vee \neg(A_{n_0} = A_m)) \Rightarrow lrun(A, n_0, m) \wedge (\neg(m < A.len) \vee (A_{n_0} \neq A_m)) \\
\equiv & \{ \} \\
& \text{true}
\end{aligned}$$

$$\begin{aligned}
\sqsubseteq & \{ \text{Repetition} \} \\
& m := n + 1; \\
& \text{do } (m < A.len \wedge A_{n_0} = A_m) \rightarrow \\
& \quad n, m : [inv \wedge guard, inv \wedge (0 \leq V < V_0)] \\
& \text{od}
\end{aligned}$$

where

$$V \triangleq A.len - m$$

$$\begin{aligned}
\sqsubseteq & \{ \text{Assignment: } inv \wedge guard \Rightarrow (inv \wedge (0 \leq V < V_0))[m \setminus m + 1] \} \\
& m := n + 1; \\
& \text{do } (m < A.len \wedge A_{n_0} = A_m) \rightarrow \\
& \quad m := m + 1 \\
& \text{od}
\end{aligned}$$

$\therefore$

$$\begin{aligned}
(inv \wedge (0 \leq V < V_0))[m \setminus m + 1] & \equiv (lrun(A, n_0, m) \wedge (0 \leq (A.len - m) < (A.len - m_0)))[m, m_0 \setminus m + 1, m] \\
& \equiv lrun(A, n_0, m + 1) \wedge (0 \leq (A.len - (m + 1)) < (A.len - m))
\end{aligned}$$

$\therefore$

$$\begin{aligned}
inv \wedge guard & \Rightarrow lrun(A, n_0, m + 1) \wedge (0 \leq (A.len - (m + 1)) < (A.len - m)) \\
lrun(A, n_0, m) \wedge (m < A.len \wedge A_{n_0} = A_m) & \Rightarrow lrun(A, n_0, m + 1) \wedge (0 \leq (A.len - (m + 1)) < (A.len - m))
\end{aligned}$$

Furthermore,

$$m < A.len \Rightarrow 0 \leq (A.len - (m + 1)) < (A.len - m)$$

is trivially true.

$$lrun(A, n_0, m) \Rightarrow lrun(A, n_0, m + 1)$$

Reiterating

$$lrun(A, i, j) \triangleq run(A, i, j) \wedge (i > 0 \Rightarrow A_{i-1} \neq A_i)$$

Because  $A_{n_0} = A_m$ , the  $run(A, n_0, m + 1)$  holds, noting that  $run(A, n_0, m + 1)$  describes a run up to, but not including index  $m + 1$ . Thus, we are free to perform the assignment, expanding the *run* range.

All conjuncts hold, and are entailed by the LHS.

□

2.

$$\begin{aligned}
pre & \triangleq A.len > 0 \\
post & \triangleq mrun(A, \ell, h) \wedge (\forall p, q \cdot mrun(A, p, q) \Rightarrow (h - \ell) \leq (q - p))
\end{aligned}$$

$$\begin{aligned}
& \ell, h : [pre, post] \\
\sqsubseteq & \quad \{\text{Composition: middle predicate is } inv\} \\
& \ell, h : [pre, inv]; \quad \ell, h : [inv, post]
\end{aligned}$$

where

$$inv \triangleq mrun(A_{[0,i]}, \ell, h) \wedge (\forall p, q \cdot mrun(A_{[0,i]}, p, q) \Rightarrow (h - \ell) \leq (q - p))$$

This invariant was chosen as the postcondition refers to a constant  $A$ , which can be written  $A_{[0,A.len]}$ , which is of the form  $A^B$ , where  $B$  is  $A.len$ . We replace  $A.len$  with a program variable  $i$ , create the invariant. We can further derive the negation of our guard to be  $i = A.len$ , such that the guard is  $i \neq A.len$ .

```

ℓ := 0;
do (m < A.len ∧ An0 = Am) →
    m := m + 1
od

```