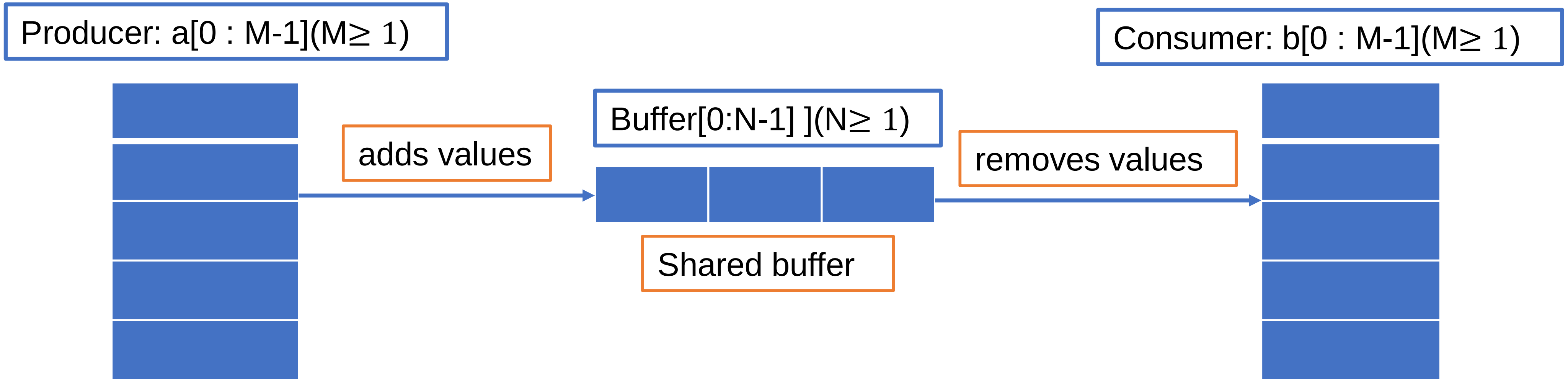
9.4 Case Study : Producer/Consumer Problem

Background



- Production of a value**:** **reading an integer value** from a finite section a[0 : M-1].
- Consumption of a value: **writing an integer value** into a corresponding section b[0 : M-1].

Synchronize

- The producer **never** attempts to **add a value into the full buffer**.
- The consumer **never** attempts to **remove a value from the empty buffer**.

Preparations

- **PC** models the **producer/onsumer problem**

  (a parallel program with **shared variable** and **await** statements)

- **PROD** models the **producer, CONS** models the **consumer**

- **Shared variable:** for a correct access of the buffer

  in : counting the number of values added to the buffer

  out : counting the number of values removed from the buffer

  Full: **in – out = N**, **Empty**: **in – out = 0**

  in mod N : the subscript of the buffer element **where the next value is to be added**

  out mod N: the subscript of the buffer element **where the next value is to be removed**

Parallel Program

$$PC \equiv in := 0;\ out := 0;\ i := 0;\ j := 0;\ [PROD \| CONS]$$

$$\begin{aligned} PROD \equiv\ & \textbf{while}\ i < M\ \textbf{do} \\ & \qquad x := a[i]; \\ & \qquad ADD(x); \\ & \qquad i := i + 1 \\ & \textbf{od} \end{aligned}$$

$$\begin{aligned} ADD(x) \equiv\ & \textbf{wait}\ in - out < N; \\ & buffer[in\ mod\ N] := x; \\ & in := in + 1 \end{aligned}$$

$$\begin{aligned} CONS \equiv\ & \textbf{while}\ j < M\ \textbf{do} \\ & \qquad REM(y); \\ & \qquad b[j] := y; \\ & \qquad j := j + 1 \\ & \textbf{od}. \end{aligned}$$

$$\begin{aligned} REM(y) \equiv\ & \textbf{wait}\ in - out > 0; \\ & y := buffer[out\ mod\ N]; \\ & out := out + 1. \end{aligned}$$

Property

- We claim the following total correctness property:

$$\models_{tot} \{\mathbf{true}\}\ PC\ \{\forall k : (0 \le k < M \rightarrow a[k] = b[k])\}, \qquad (9.13)$$

(1) PC is deadlock free

(2) PC terminates with all values from a[0 : M-1] copied in that order into b[0 : M-1]

Step1: PROD--Find Invariants and Bound Functions

- Loop invariant :

$$
\begin{aligned}
p_1 \equiv \quad & \forall k : (out \le k < in \rightarrow a[k] = buffer[k \bmod N]) && (9.14)\\
\wedge \quad & 0 \le in - out \le N && (9.15)\\
\wedge \quad & 0 \le i \le M && (9.16)\\
\wedge \quad & i = in && (9.17)
\end{aligned}
$$

- Bound function:

$$t_1 \equiv M - i.$$

- Abbreviation:

$$I \equiv (9.14) \wedge (9.15)$$

$$I_1 \equiv (9.14) \wedge (9.15) \wedge (9.16)$$

$$
\begin{aligned}
PROD \equiv \ & \textbf{while } i < M \textbf{ do}\\
& \quad x := a[i];\\
& \quad ADD(x);\\
& \quad i := i + 1\\
& \textbf{od}
\end{aligned}
$$

$$
\begin{aligned}
ADD(x) \equiv \ & \textbf{wait } in - out < N;\\
& buffer[in \bmod N] := x;\\
& in := in + 1
\end{aligned}
$$

Step1: PROD--Proof Outline

- A proof outline for weak total correctness of **PROD**

$$
\begin{array}{l}
\{\mathbf{inv}: p_1\}\{\mathbf{bd}: t_1\} \\
\mathbf{while}\ i < M\ \mathbf{do} \\
\qquad \{p_1 \wedge i < M\} \\
\qquad x := a[i]; \\
\qquad \{p_1 \wedge i < M \wedge x = a[i]\} \\
\qquad \mathbf{wait}\ in - out < N; \\
\qquad \{p_1 \wedge i < M \wedge x = a[i] \wedge in - out < N\} \\
\qquad buffer[in\ mod\ N] := x; \\
\qquad \{p_1 \wedge i < M \wedge a[i] = buffer[in\ mod\ N] \wedge in - out < N\} \\
\qquad in := in + 1; \\
\qquad \{I_1 \wedge i + 1 = in \wedge i < M\} \\
\qquad i := i + 1 \\
\mathbf{od} \\
\{p_1 \wedge i = M\}.
\end{array}
$$

$$\forall k : (out \leq k < in \rightarrow a[k] = buffer[k \ mod \ N]) \tag{9.14}$$
$$\wedge \ 0 \leq in - out \leq N \tag{9.15}$$

Step1: CONS--Find Invariants and Bound Functions

- Loop invariant :

$$I \equiv (9.14) \wedge (9.15)$$

$$\begin{aligned} p_2 \equiv \quad & I && (9.20) \\ \wedge \quad & \forall k : (0 \leq k < j \rightarrow a[k] = b[k]) && (9.21) \\ \wedge \quad & 0 \leq j \leq M && (9.22) \\ \wedge \quad & j = out, && (9.23) \end{aligned}$$

- Bound function:

$$t_2 \equiv M - j.$$

- Abbreviation:

$$I_2 \equiv (9.20) \wedge (9.21) \wedge (9.22).$$

$$\begin{aligned} CONS \equiv \ & \textbf{while } j < M \textbf{ do} \\ & \quad REM(y); \\ & \quad b[j] := y; \\ & \quad j := j + 1 \\ & \textbf{od}. \end{aligned}$$

$$\begin{aligned} REM(y) \equiv \ & \textbf{wait } in - out > 0; \\ & y := buffer[out \ mod \ N]; \\ & out := out + 1. \end{aligned}$$

Step1: CONS--Proof Outline

- A proof outline for weak total correctness of **CONS**

$$
\begin{array}{l}
\{\textbf{inv}: p_2\}\{\textbf{bd}: t_2\} \\
\textbf{while } j < M \textbf{ do} \\
\quad \{p_2 \wedge j < M\} \\
\quad \textbf{wait } in - out > 0; \\
\quad \{p_2 \wedge j < M \wedge in - out > 0\} \\
\quad y := buffer[out \ mod \ N]; \\
\quad \{p_2 \wedge j < M \wedge in - out > 0 \wedge y = a[j]\} \\
\quad out := out + 1; \\
\quad \{I_2 \wedge j + 1 = out \wedge j < M \wedge y = a[j]\} \\
\quad b[j] := y; \\
\quad \{I_2 \wedge j + 1 = out \wedge j < M \wedge a[j] = b[j]\} \\
\quad j := j + 1 \\
\textbf{od} \\
\{p_2 \wedge j = M\}.
\end{array}
$$

Step2: Interference Freedom

$\{\textbf{inv} : p_1\}\{\textbf{bd} : t_1\}$
while $i < M$ **do**
$\{p_1 \wedge i < M\}$
1 $x := a[i];$
$\{p_1 \wedge i < M \wedge x = a[i]\}$
2 **wait** $in - out < N;$
$\{p_1 \wedge i < M \wedge x = a[i] \wedge in - out < N\}$
3 $buffer[in \bmod N] := x;$
$\{p_1 \wedge i < M \wedge a[i] = buffer[in \bmod N] \wedge in - out < N$
4 $in := in + 1;$
$\{I_1 \wedge i + 1 = in \wedge i < M\}$
5 $i := i + 1$
od
$\{p_1 \wedge i = M\}.$

1
2 $\{\textbf{inv} : p_2\}\{\textbf{bd} : t_2\}$
while $j < M$ **do**
3 $\{p_2 \wedge j < M\}$
wait $in - out > 0;$
4 $\{p_2 \wedge j < M \wedge in - out > 0\}$
$y := buffer[out \bmod N];$
5 $\{p_2 \wedge j < M \wedge in - out > 0 \wedge y = a[j]\}$
$out := out + 1;$
6 $\{I_2 \wedge j + 1 = out \wedge j < M \wedge y = a[j]\}$
$b[j] := y;$
7 $\{I_2 \wedge j + 1 = out \wedge j < M \wedge a[j] = b[j]\}$
$j := j + 1$
8
od
$\{p_2 \wedge j = M\}.$

CHECK:5*8*2=80!!!

Step2: Interference Freedom

$$I \equiv \forall k : (out \le k < in \rightarrow a[k] = buffer[k \bmod N]) \wedge 0 \le in - out \le N$$

$\{\textbf{inv} : p_1\}\{\textbf{bd} : t_1\}$
while $i < M$ **do**
$\{p_1 \wedge i < M\}$
$x := a[i];$
$\{p_1 \wedge i < M \wedge x = a[i]\}$
wait $in - out < N;$
$\{p_1 \wedge i < M \wedge x = a[i] \wedge \boxed{in - out < N}\}$
$buffer[in \bmod N] := x;$
$\{p_1 \wedge i < M \wedge a[i] = buffer[in \bmod N] \wedge in - out < N\}$
$\boxed{in := in + 1;}$
$\{I_1 \wedge i + 1 = in \wedge i < M\}$
$i := i + 1$
od
$\{p_1 \wedge i = M\}.$

Only this conjunct contains a variable ***out*** changed in **CONS**

Only this assignment changes the shared variable ***in***

$\{\textbf{inv} : p_2\}\{\textbf{bd} : t_2\}$
while $j < M$ **do**
$\{p_2 \wedge j < M\}$
wait $in - out > 0;$
$\{p_2 \wedge j < M \wedge \boxed{in - out > 0}\}$
$y := buffer[out \bmod N];$
$\{p_2 \wedge j < M \wedge in - out > 0 \wedge y = a[j]\}$
$\boxed{out := out + 1;}$
$\{I_2 \wedge j + 1 = out \wedge j < M \wedge y = a[j]\}$
$b[j] := y;$
$\{I_2 \wedge j + 1 = out \wedge j < M \wedge a[j] = b[j]\}$
$j := j + 1$
od
$\{p_2 \wedge j = M\}.$

Only this conjunct contains a variable ***in*** changed in **PROD**

Only this assignment changes the shared variable ***out***

Step2: Interference Freedom

- I –part of p_1 and p_2 is kept invariant in both proof outlines.

  All assignments T in the proof outlines for PROD and CONS satisfy

  $$\{I \wedge pre(T)\}\ T\ \{I\}.$$

- Test the conjuction $\boldsymbol{in - out < N}$ in PROD

  $$\{in - out < N\}\ out := out + 1\ \{in - out < N\}.$$

- Test the conjuction $\boldsymbol{in - out > 0}$ in CONS

  $$\{in - out > 0\}\ in := in + 1\ \{in - out > 0\}.$$

Step3: Deadlock Freedom

Potential deadlocks

$$(\textbf{wait } in - out < N, \textbf{wait } in - out > 0),$$
$$(\textbf{wait } in - out < N, E),$$
$$(E, \textbf{wait } in - out > 0),$$

Logical consequences

$$(in - out \geq N, in - out \leq 0),$$
$$(in < M \wedge in - out \geq N, out = M),$$
$$(in = M, out < M \wedge in - out \leq 0).$$

Conjunction

False
False
False

Deadlock Freedom

$r_i \equiv pre(R_i) \wedge \neg B$ if $R_i \equiv$ **await** B **then** S **end**,
$r_i \equiv q_i$ if $R_i \equiv E$.

(R_1, . . .,R_n)
(r_1, . . ., r_n)

The Whole Proof: Applying Rules

- Apply rule 29 for the parallel composition of PROD and CONS

RULE 29: PARALLELISM WITH DEADLOCK FREEDOM

$$\frac{\begin{array}{l}(1)\ \text{The standard proof outlines } \{p_i\}\ S_i^*\ \{q_i\}, i \in \{1,\ldots,n\} \\ \quad \text{for weak total correctness are interference free,} \\ (2)\ \text{For every potential deadlock } (R_1,\ldots,R_n) \text{ of} \\ \quad [S_1\|\ldots\|S_n] \text{ the corresponding tuple of} \\ \quad \text{assertions } (r_1,\ldots,r_n) \text{ satisfies } \neg \bigwedge_{i=1}^n r_i.\end{array}}{\{\bigwedge_{i=1}^n p_i\}\ [S_1\|\ldots\|S_n]\ \{\bigwedge_{i=1}^n q_i\}}$$

$$\{p_1 \wedge p_2\}\ [PROD\|CONS]\ \{p_1 \wedge p_2 \wedge in = M \wedge j = M\}.$$

- Apply rule 6

RULE 6: CONSEQUENCE

$$\frac{p \rightarrow p_1, \{p_1\}\ S\ \{q_1\}, q_1 \rightarrow q}{\{p\}\ S\ \{q\}}$$

$$\{\mathbf{true}\}\ in := 0;\ out := 0;\ i := 0;\ j := 0\ \{p_1 \wedge p_2\}$$

$$p_1 \wedge p_2 \wedge i = M \wedge j = M \rightarrow \forall k : (0 \leq k < M \rightarrow a[k] = b[k])$$

$$\models_{tot} \{\mathbf{true}\}\ PC\ \{\forall k : (0 \leq k < M \rightarrow a[k] = b[k])\}.$$

Thank You