程序验证方法
研究生课程
Chapter 8
Parallel Programs with Shared Variables
(8.1-8.5)

朱惠彪

华东师范大学 软件工程学院

8.1 Access to Shared Variables

Example 8.1

Consider the two component programs

$$S_1 \equiv x := x + 2$$

$$S_1' \equiv x := x + 1;\ x := x + 1$$

- **In isolation: both programs exhibit the same input/output behavior**
- **When executed in parallel with the component** $S_2 \equiv x := 0$ **,**

  S_1 and S_1' behave differently.

  ✓ **Upon termination of $[S_1 \| S_2]$, the value of x can be either 0 or 2.**

  ✓ **Upon termination of $[S_1' \| S_2]$, the value of x can be 0, 1 or 2.**

8.2 Syntax

Syntax

- **while programs** **in Chapter 3 together with the following clause for** **atomic regions**:

$$S \ ::= < S_0 >$$

while programs:

$$S ::= skip \mid u := t \mid S_1; S_2 \mid if\ B\ then\ S_1\ else\ S_2\ fi \mid while\ B\ do\ S_1 od.$$

- **Parallel programs** **with shared variables (or simply parallel programs):**

$$S \ ::= \ [S_1 || \ldots || Sn]$$

Where $S_1, \ldots, S_n$ are component programs ($n > 1$).

8.3 Semantics

Semantics

- **(xviii)** $\dfrac{< S,\sigma > \to * < E,\tau >}{< <S>,\ \sigma > \to < E,\tau >}$
- **Chapter3 while Programs**

(i) $< skip, \sigma > \ \to \ < E, \sigma >,$

(ii) $< u := t, \sigma > \ \to \ < E, \sigma[u := \sigma(t)] >,$

(iii) $\dfrac{< S_1, \sigma > \ \to \ < S_2, \tau >}{< S_1;\ S, \sigma > \ \to \ < S_2;\ S, \tau >},$

(iv) $< \textbf{if } B \textbf{ then } S_1 \textbf{ else } S_2 \textbf{ fi}, \sigma > \ \to \ < S_1, \sigma >$ where $\sigma \models B,$

(v) $< \textbf{if } B \textbf{ then } S_1 \textbf{ else } S_2 \textbf{ fi}, \sigma > \ \to \ < S_2, \sigma >$ where $\sigma \models \neg B,$

(vi) $< \textbf{while } B \textbf{ do } S \textbf{ od}, \sigma > \ \to \ < S;\ \textbf{while } B \textbf{ do } S \textbf{ od}, \sigma >$
where $\sigma \models B,$

(vii) $< \textbf{while } B \textbf{ do } S \textbf{ od}, \sigma > \ \to \ < E, \sigma >,$ where $\sigma \models \neg B.$

- **Chapter7.2 (xvii)**

$$\frac{< Si,\sigma > \to < T_i,\tau >}{< [S_1||...||S_i||...||Sn],\sigma > \to < [S_1||...||T_i||...||Sn],\tau >},\ \textbf{where i} \in \{1,...,n\}.$$

8.4 Verification: Partial Correctness

Component Programs

- **RULE 26: ATOMIC REGION**

$$\frac{\{p\}\ S\ \{q\}}{\{p\} < S > \{q\}}$$

- **Proof outlines** **for partial correctness of component**

Programs are generated by the formation rules (i)–(vii) (page 80) given for while programs plus the following one.

(x)

$$\frac{\{p\}\ S^*\ \{q\}}{\{p\} < S^* > \{q\}}$$

where as usual S* stands for an annotated version of S.

- **pre(T):** **the precondition of subprogram T**

(i) $\{p\}\ skip\ \{p\}$

(ii) $\{p[u := t]\}\ u := t\ \{p\}$

(iii) $\dfrac{\{p\}\ S_1^*\ \{r\}, \{r\}\ S_2^*\ \{q\}}{\{p\}\ S_1^*;\ \{r\}\ S_2^*\ \{q\}}$

(iv) $\dfrac{\{p \wedge B\}\ S_1^*\ \{q\}, \{p \wedge \neg B\}\ S_2^*\ \{q}{\{p\}\ \text{if } B \text{ then } \{p \wedge B\}\ S_1^*\ \{q\} \text{ else } \{p \wedge \neg B\}}$

(v) $\dfrac{\{p \wedge B\}\ S^*\ \{p\}}{\{inv : p\}\ \text{while } B \text{ do } \{p \wedge B\}\ S^*\ \{p\} \text{ od } \{p}$

(vi) $\dfrac{p \to p_1,\ \ \{p_1\}\ S^*\ \{q_1\},\ q_1 \to q}{\{p\}\{p_1\}\ S^*\ \{q_1\}\{q\}}$

vii) $\dfrac{\{p\}\ S^*\ \{q\}}{\{p\}\ S^{**}\ \{q\}}$

No Compositionality of Input/Output Behavior

- **Example**

$$\models \{p\}\ x := x + 2\ \{q\} \quad \text{iff} \models \{p\}\ x := x + 1;\ x := x + 1\ \{q\}$$

parallel composition $\{p_1\}$ $x=0$ $\{q_1\}$

We have $\models \{true\}\ [x := x + 2 || x=0]\ \{x=0 \vee x=2\}$,

but $\not\models \{true\}\ [x := x + 1;\ x := x + 1 || x=0]\ \{x=0 \vee x=2\}$

- **Question**

If $\{p_i\}\ S_i\ \{q_i\}$,

can we have $\{\wedge_{i=1}^{n}\ p_i\}[S_1 || S_2 || \ldots || S_n]\{\wedge_{i=1}^{n}\ q_i\}$?

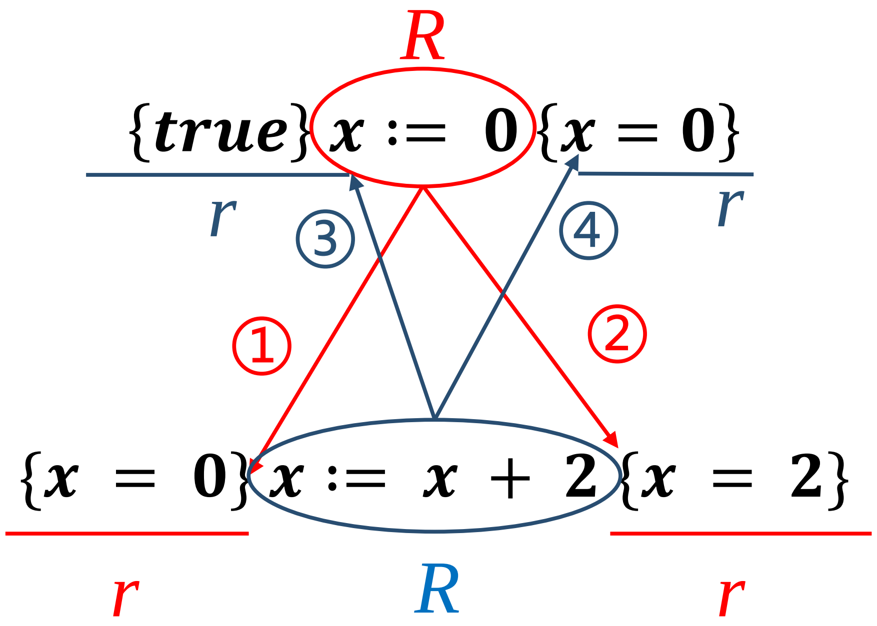
Parallel Composition: Interference Freedom

Example: consider the parallel program [x := x + 2||x := 0]

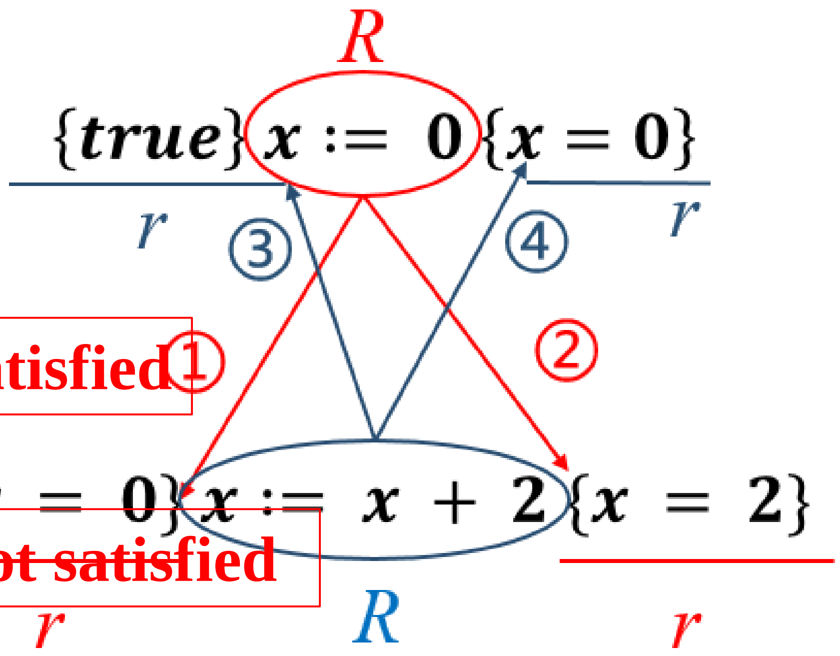
(1) Proof outlines

{pre(R)/\r} R {r}

(2) Interference Freedom



{r ∧ pre(R)} R {r}



- R ≡ x:=0 , pre(R) ≡ true
  - Case 1: r ≡ x=0

    {x=0 ∧ true} x:=0 {x=0} **Satisfied**

  - Case 2: r ≡ x=2

    {x=2 ∧ true}x:=0 {x=2} **Not satisfied**

- R≡ x:=x+2, pre(R) ≡ x=0
  - Case 3: r ≡ true

    {true ∧ x=0} x:=x+2 {true} **Satisfied**

  - Case 4: r ≡ x=0

    {x=0 ∧ x=0} x:=x+2 {x=0} **Not satisfied**

Definition 8.1. (Interference Freedom: Partial Correctness)

(i) Let S be a component program. Consider a standard proof outline $\{p\}\ S^*\ \{q\}$ for partial correctness and a statement R with the precondition pre(R).

We say that R does not interfere with $\{p\}\ S^*\ \{q\}$ if

- **for all assertions r in $\{p\}\ S^*\ \{q\}$ the correctness formula**

$$\{r \wedge pre(R)\}\ R\ \{r\}$$

holds in the sense of partial correctness.

(ii) Let $[S_1\|\ldots\|S_n]$ be a parallel program. Standard proof outlines $\{p_i\}\ S_i^*\ \{q_i\}$, $i \in \{1,\ldots,n\}$, for partial correctness are called interference free if no *normal assignment or atomic region* of a program S_i interferes with the proof outline $\{p_j\}\ S_j^*\ \{q_j\}$ of another program S_j where $i \neq j$.

RULE 27: PARALLELISM WITH SHARED VARIABLES

$$\frac{\textit{The standard proof outlines } \{pi\}\ Si^{*}\ \{qi\},\quad i \in \{1,\ldots,n\}, \textit{ are interference free}}{\{\wedge_{i=1}^{n}\ p_i\}[S_1||S_2||\ldots||S_n]\{\wedge_{i=1}^{n}\ q_i\}}$$

Example 8.2

Example 8.2. As a first application of the parallelism with shared variables rule let us prove partial correctness of the parallel programs considered in Section 8.1.

(i) First we consider the program [x := x + 2 || x := 0]. The standard proof outlines

$$\{x=0\}x:=x + 2\{x=2\}$$

and

$$\{true\}x:= 0\{x=0\}$$

are obviously correct, but they are not interference free.

Please check!

Example 8.2

Weakening the postconditions

$$\boxed{\{r \wedge pre(R)\}\ R\ \{r\}}$$

$$\underbrace{\{x=0\}}_{r}\ \overset{R}{x := x+2}\ \underbrace{\{x=0 \vee x=2\}}_{r}$$

1 2 3 4

$$\underbrace{\{\mathbf{true}\}}_{r}\ \underset{R}{x := 0}\ \underbrace{\{x=0 \vee x=2\}}_{r}$$

interference-free checking:
4 cases

Example 8.2

Case 2:

$$\boxed{\{r \wedge pre(R)\}\ R\ \{r\}}$$

$$\{x = 0\}\ \overbrace{x := x + 2}^{R}\ \{x = 0 \vee x = 2\}$$

②

$$\{\mathbf{true}\}\ x := 0\ \underbrace{\{x = 0 \vee x = 2\}}_{r}$$

$$\boxed{\{x = 0 \wedge (x = 0 \vee x = 2)\}}\ x := x + 2\ \{x = 0 \vee x = 2\}$$

satisfied

$$x = 0$$

Thus,

$$\{x = 0\}\ [\ x := x + 2 \parallel x := 0\ ]\ \{x = 0 \vee x = 2\}$$

Example 8.2

(ii) Next we study the program $[x := x + 1;\ x := x + 1 \| x := 0]$. Consider the following proof outlines:

$$\{x = 0\}$$
R $\quad x := x + 1;$
$$\{x = 0 \lor x = 1\}$$
R $\quad x := x + 1$
$$\{\mathbf{true}\}$$

and

$$\{\mathbf{true}\}\ x := 0\ \{x = 0 \lor x = 1 \lor x = 2\}.$$

R

Example 8.2

To establish their interference freedom seven interference freedom checks need to be made.

All of them hold.

Thus, we have:

$$\{x = 0\}\ [\ x := x + 1;\ x := x + 1 \ \|\ \mathrm{x} := 0\ ]\ \{x = 0 \vee x = 1 \vee x = 2\}$$

Example 8.2

(iii) Consider

$$[\ \langle x := x + 1;\ x := x + 1 \rangle\ \|\ x := 0\ ]$$

The proof outlines

$$\underbrace{\{x = 0\}}_{r}\ \overbrace{\langle x := x + 1;\ x := x + 1 \rangle}^{R}\ \underbrace{\{\mathit{true}\}}_{r}$$

and

$$\underbrace{\{\mathit{true}\}}_{r}\ \underbrace{x := 0}_{R}\ \underbrace{\{x = 0 \lor x = 2\}}_{r}$$

are clearly interference free.

Thus, we have

$$\{x = 0\}\ [\ \langle x := x + 1;\ x := x + 1 \rangle\ \|\ x := 0\ ]\ \{x = 0 \lor x = 2\}$$

Auxiliary Variables Needed

The parallelism with shared variables rule 27 becomes too weak to reason about partial correctness.

Lemma 8.6. (Incompleteness) ***The correctness formula***

$$\{\mathbf{true}\}\ [x := x + 2 \,\|\, x := 0]\ \{x = 0 \lor x = 2\} \qquad (8.1)$$

is not a theorem in the proof system PW + rule 27.

RULE 27: PARALLELISM WITH SHARED VARIABLES

$$\frac{\text{The standard proof outlines } \{p_i\}\ S_i^*\ \{q_i\},\ i \in \{1,\ldots,n\}, \text{ are interference free}}{\{\bigwedge_{i=1}^{n} p_i\}\ [S_1\|\ldots\|S_n]\ \{\bigwedge_{i=1}^{n} q_i\}}$$

Auxiliary Variables Needed

Proof.

Suppose by contradiction that this correctness formula can be proved in the system *PW* + rule 27. Then, for some interference free proof outlines

$$\{p_1\}\ x := x + 2\ \{q_1\},$$

and

$$\{p_2\}\ x := 0\ \{q_2\},$$

the implications

$$\mathbf{true} \rightarrow p_1 \land p_2 \tag{8.2}$$

and

$$q_1 \land q_2 \rightarrow x = 0 \lor x = 2 \tag{8.3}$$

hold. Then by (8.2) both p_1 and p_2 are true.

Auxiliary Variables Needed

Thus {true} $x := x + 2$ $\{q_1\}$ holds, so by the Soundness Theorem 3.1 the assertion $q_1[x := x + 2]$ is true. Since x ranges over all integers,

$$q_1 \tag{8.4}$$

itself is true.

Also, {true} $x := 0$ $\{q_2\}$ implies by the Soundness Theorem 3.1

$$q_2[x := 0]. \tag{8.5}$$

Moreover, by interference freedom $\{true \land q_2\}$ $x := x + 2$ $\{q_2\}$ which gives

$$q_2 \rightarrow q_2[x := x + 2]. \tag{8.6}$$

Theorem 3.1. (Soundness of PW and TW)

(i) The proof system PW is sound for partial correctness of **while** *programs.*
(ii) The proof system TW is sound for total correctness of **while** *programs.*

Auxiliary Variables Needed

By induction (8.5) and (8.6) imply

$$\forall x : (x \geq 0 \wedge \text{even}(x) \rightarrow q_2). \qquad (8.7)$$

Now by (8.3) and (8.4) we obtain from (8.7)

$$\forall x : (x \geq 0 \wedge \text{even}(x) \rightarrow x = 0 \vee x = 2)$$

which gives a contradiction.

$q_1 \wedge q_2 \rightarrow x = 0 \vee x = 2$	(8.3)
q_1	(8.4)
$q_2[x := 0].$	(8.5)
$q_2 \rightarrow q_2[x := x + 2].$	(8.6)

Auxiliary Variables Needed

Summarizing, in any interference free proof outline of the above form, the postcondition q_2 of $x := 0$ would hold for every even $x \geq 0$, whereas it should hold only for $x = 0$ or $x = 2$.

The reason for this mismatch is that we cannot express in terms of the variable x the fact that the first component $x := x + 2$ should still be executed.

What is needed: the rule of auxiliary variables (rule 25, Chapter 7).

RULE 25: AUXILIARY VARIABLES

$$\frac{\{p\}\ S\ \{q\}}{\{p\}\ S_0\ \{q\}}$$

where for some set of auxiliary variables A of S with $free(q) \cap A = \emptyset$, the program S_0 results from S by deleting all assignments to variables in A.

Example 8.3

Aim: For proving $\{\text{true}\}\ [\ x := x + 2 \parallel x := 0\ ]\ \{x = 0 \vee x = 2\}$, (8.1)

Method: Auxiliary Boolean variable "*done*" ------ indicating whether the assignment $x := x + 2$ has been executed.

Consider the correctness formula:

$$\begin{aligned}
&\{\text{true}\}\\
&\mathit{done} := \text{false}; \qquad\qquad (8.8)\\
&[\langle\, x := x + 2;\ \mathit{done} := \text{true}\,\rangle \parallel x := 0]\\
&\{x = 0 \vee x = 2\}.
\end{aligned}$$

Now we consider the proof outlines:

$$\{\neg \mathit{done}\}\ \langle\, x := x + 2;\ \mathit{done} := \text{true}\,\rangle\ \{\text{true}\} \qquad (8.9)$$

and

$$\{\text{true}\}\ x := 0\ \{(x = 0 \vee x = 2) \wedge (\neg \mathit{done} \rightarrow x = 0)\}. \qquad (8.10)$$

Example 8.3

Interference-free checking:

For example, consider the case: R

$$\{\neg done\} < x := x + 2;\ done := \text{true} > \{\text{true}\}$$

$$\{\text{true}\}\ x := 0\ \{\underline{(x = 0 \lor x = 2) \land (\neg done \to x = 0)}\}\ r$$

The checking is:

$$\{\underbrace{\boxed{(x = 0 \lor x = 2) \land (\neg done \to x = 0)}}_{r} \land \underbrace{\boxed{\neg done}}_{pre(R)}\}$$

$$\{x = 0\}$$

$$< x := x + 2;\ done := \text{true} >$$

$$\{x = 2 \land done\}$$

$$\{\underbrace{\boxed{(x = 0 \lor x = 2) \land (\neg done \to x = 0)}}_{r}\}.$$

{pre(T)∧r} R {r}

Example 8.3

By applying rule 27 to (8.9) and (8.10), and by using the consequence rule, we have:

$$\{\neg done\}$$
$$[\ < x := x + 2;\ done := \text{true} > \ \|\ x := 0] \qquad (8.11)$$
$$\{x = 0 \lor x = 2\}.$$

Also we have:

$$\{\text{true}\}\ done := \text{false}\ \{\neg done\}$$

Then we have (8.8).

$$\{\text{true}\}$$
$$done := \text{false}; \qquad (8.8)$$
$$[< x := x + 2;\ done := \text{true} > \ \|\ x := 0]$$
$$\{x = 0 \lor x = 2\}.$$

$$\{\neg done\} < x := x + 2;\ done := \text{true} > \{\text{true}\} \qquad (8.9)$$
$$\{\text{true}\}\ x := 0\ \{(x = 0 \lor x = 2) \land (\neg done \rightarrow x = 0)\} \qquad (8.10)$$

Proof System PSV

PROOF SYSTEM PSV:
This system consists of the group of axioms and rules 1-6, 25-27, andA2-A6.

RULE 25: AUXILIARY VARIABLES

$$\frac{\{p\}\ S\ \{q\}}{\{p\}\ S_0\ \{q\}}$$

where for some set of auxiliary variables A of S with $free(q) \cap A = \emptyset$, the program S_0 results from S by deleting all assignments to variables in A.

RULE 26: ATOMIC REGION

$$\frac{\{p\}\ S\ \{q\}}{\{p\}\ \langle S \rangle\ \{q\}}$$

RULE 27: PARALLELISM WITH SHARED VARIABLES

$$\frac{\begin{array}{l}\text{The standard proof outlines } \{p_i\}\ S_i^*\ \{q_i\}, \\ i \in \{1, \ldots, n\}, \text{ are interference free}\end{array}}{\{\bigwedge_{i=1}^{n} p_i\}\ [S_1 \| \ldots \| S_n]\ \{\bigwedge_{i=1}^{n} q_i\}}$$

Soundness

Lemma 8.7. (Auxiliary Variables) ***The rule of auxiliary variables (rule 25, page 257) is sound for partial (and total) correctness of parallel programs.***

Corollary 8.1. (Parallelism with Shared Variables) ***The parallelism with shared variables rule 27 is sound for partial correctness of parallel programs.***

Corollary 8.2. (Soundness of PSV) ***The proof system PSV is sound for partial correctness of parallel programs.***

8.5 Verification: Total Correctness

Component Programs

The proof outline for total correctness

$$\frac{\{p \wedge B\}\ S^*\ \{p\},\quad \{p \wedge B \wedge t = z\}\ S^{**}\ \{t < z\},\quad p \rightarrow t \geq 0}{\{\text{inv} : p\}\{\text{bd} : t\}\ \textbf{while}\ B\ \textbf{do}\ \{p \wedge B\}\ S^*\ \{p\}\ \textbf{od}\ \{p \wedge \neg B\}}$$

too weak for our purpose

Reason:

In the context of parallel programs it is possible that components interfere with the termination proofs of other components.

Component Programs

Solution:

We require that in proof outlines of loops *while B do S od* the bound function *t* is such that

(i) all normal assignments and atomic regions inside *S* decrease *t* or leave it unchanged,

(ii) on each syntactically possible path through *S* at least one normal assignment or atomic region decreases *t*.

Path(S)

Definition 8.2. For a sequential component S, we define the set $path(S)$ by induction on S:

- $path(skip) = \{\varepsilon\}$,
- $path(u := t) = \{u := t\}$,
- $path(<S>) = \{<S>\}$,
- $path(S_1; S_2) = path(S_1) \,;\, path(S_2)$,
- $path(\text{if } B \text{ then } S_1 \text{ else } S_2 \text{ fi}) = path(S_1) \cup path(S_2)$,
- $path(\text{while } B \text{ do } S \text{ od}) = \{\varepsilon\}$.

$$\Pi 1; \Pi 2 = \{\pi 1; \pi 2 \mid \pi 1 \in \Pi 1 \text{ and } \pi 2 \in \Pi 2\}.$$

Proof Outline: Total Correctness

Definition 8.3. (Proof Outline: Total Correctness) Proof outlines and standard proof outlines for the total correctness of component programs are generated by the same formation axioms and rules as those used for defining (standard) proof outlines for the partial correctness of component programs. The only exception is the formation rule (v) dealing with *while* loops which is replaced by the following formation rule.

(xi)

$$\frac{\begin{array}{ll}(1) & \{p \wedge B\}\ S^*\ \{p\} \text{ is standard,} \\ (2) & \{pre(R) \wedge t = z\}\ R\ \{t \leq z\} \text{ for every normal} \\ & \text{assignment and atomic region } R \text{ within } S, \\ (3) & \text{for each path } \pi \in path(S) \text{ there exists} \\ & \text{a normal assignment or atomic region } R \text{ in } \pi \\ & \text{such that} \\ & \{pre(R) \wedge t = z\}\ R\ \{t < z\}, \\ (4) & p \rightarrow t \geq 0\end{array}}{\{\text{inv} : p\}\{\text{bd} : t\}\ \text{while } B \text{ do } \{p \wedge B\}\ S^*\ \{p\}\ \text{od}\ \{p \wedge \neg B\}}$$

where t is an integer expression and z is an integer variable not occurring in p, t, B or S^*, and where $pre(R)$ stands for the assertion preceding R in the standard proof outline $\{p \wedge B\}\ S^*\ \{p\}$ for total correctness.

Parallel Composition: Interference Freedom

Definition 8.4. (Interference Freedom: Total Correctness)

(1) Let S be a component program. Consider a standard proof outline $\{p\}\ S^*\ \{q\}$ for total correctness and a statement A with the precondition $pre(A)$. We say that A does not interfere with $\{p\}\ S^*\ \{q\}$ if the following two conditions are satisfied:

(i) for all assertions r in $\{p\}\ S^*\ \{q\}$ the correctness formula

$$\{r \land pre(A)\}\ A\ \{r\}$$

holds in the sense of total correctness,

(ii) for all bound functions t in $\{p\}\ S^*\ \{q\}$ the correctness formula

$$\{pre(A) \land t = z\}\ A\ \{t \leq z\}$$

holds in the sense of total correctness, where z is an integer variable not occurring in A, t or $pre(A)$.

(2) Let $[S_1 \parallel \ldots \parallel S_n]$ be a parallel program. Standard proof outlines $\{p_i\}\ S_i^*\ \{q_i\}$, $i \in \{1, \ldots, n\}$, for total correctness are called *interference free* if no normal assignment or atomic region A of a component program S_i interferes with the proof outline $\{p_j\}\ S_j^*\ \{q_j\}$ of another component program S_j where $i \neq j$.

Proof System TSV

PROOF SYSTEM TSV:
This system consists of the group of axioms and rules 1-5, 7, 25-27, andA3-A6.

Example 8.4

Now we prove that

$$S \equiv [\textbf{while } x > 2 \textbf{ do } x := x - 2 \textbf{ od} \;\|\; x := x - 1]$$

satisfies the correctness formula

$$\{x > 0 \land even(x)\}\ S\ \{x = 1\}$$

in the sense of total correctness.

Example 8.4

Proof. The proof outlines for the components of S:

$$
\begin{array}{l}
\{\mathrm{inv} : x > 0\}\{\mathrm{bd} : x\} \\
\mathrm{while}\ x > 2\ \mathrm{do} \\
r\ \underline{\{x > 2\}} \\
x := x - 2 \\
\mathrm{od} \\
\{x = 1 \vee x = 2\}
\end{array}
$$

and

$$\{even(x)\}\ \overset{R}{x := x - 1}\ \{odd(x)\}.$$

The only path in the loop body: $x := x{-}2$

The bound function: $t \equiv x$

These proof outlines satisfy the requirements of Def. 8.4.

Example 8.4

Interference freedom checking (one case):

$\{\text{inv} : x > 0\}\{\text{bd} : x\}$
while $x > 2$ do
r $\underline{\{x > 2\}}$
$x := x - 2$
od
$\{x = 1 \lor x = 2\}$

R
$\{even(x)\}\ x := x - 1\ \{odd(x)\}$

$$\{x > 2 \land even(x)\}\ x := x - 1\ \{x > 2\}$$

satisfied

We get the desired correctness result.

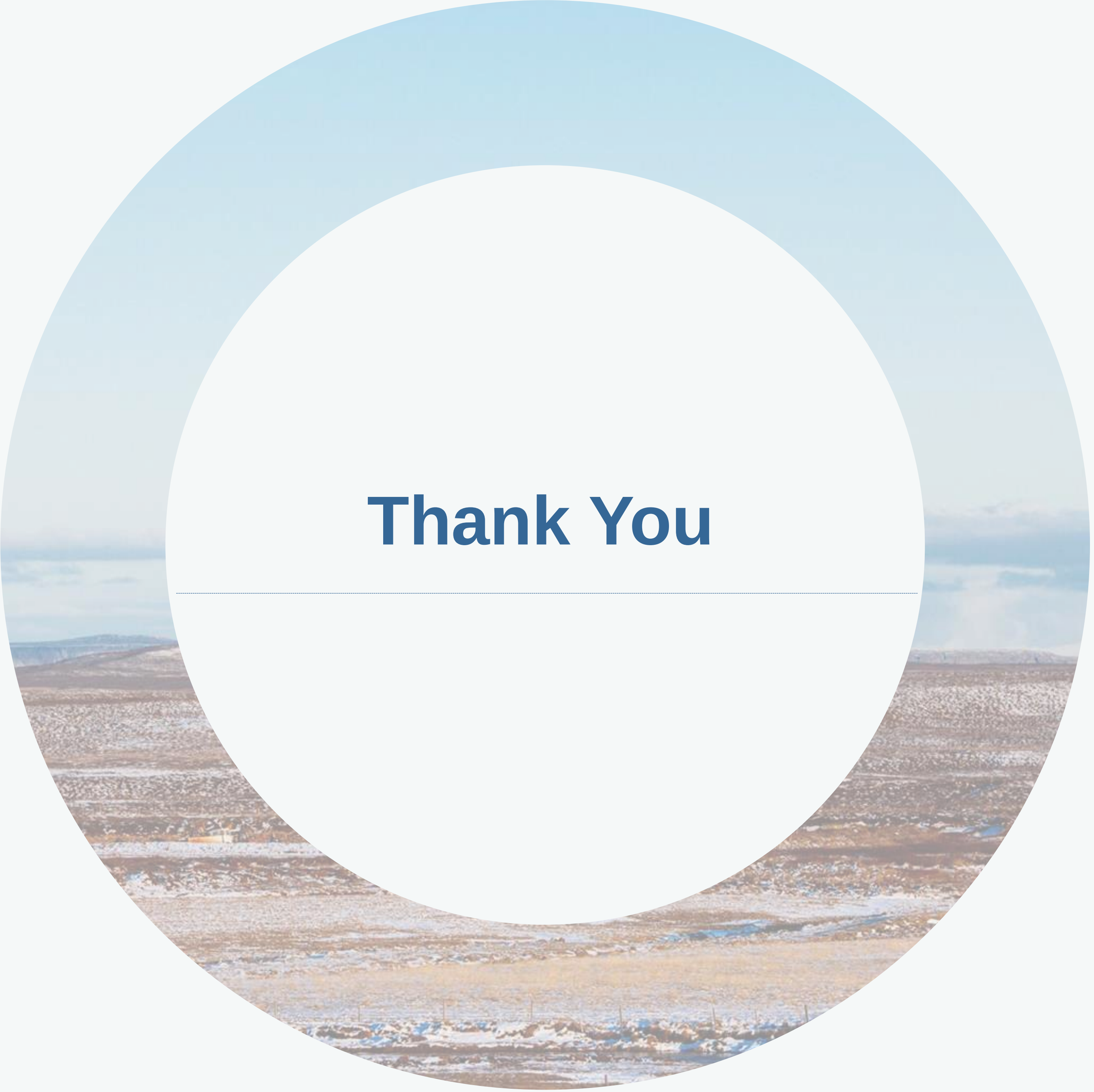
Soundness

Lemma 8.9. (Termination) ***Let $\{p_i\}\ S_i^*\ \{q_i\}$, $i \in \{1, \ldots, n\}$, be interference free standard proof outlines for total correctness for component programs S_i. Then***

$$\bot \notin \mathcal{M}_{tot}[\![[S_1 \| \ldots \| S_n]]\!]([\![\wedge_{i=1}^{n} p_i]\!]). \tag{8.13}$$

Corollary 8.3. (Parallelism with Shared Variables) ***The parallelism with shared variables rule 27 is sound for total correctness of parallel programs.***

Corollary 8.4. (Soundness of TSV) ***The proof system TSV is sound for total correctness of parallel programs.***



Thank You