# 程序验证方法
# 研究生课程
## Chapter 10 (10.1-10.4)
## Nondeterministic Programs

朱惠彪
华东师范大学 软件工程学院

# 10.1 Syntax

Expanding the grammar for while programs by adding for each n≥ 1 the following production rules:

- if command or alternative command

  $S ::= \text{if } B_1 \rightarrow S_1 \square \ldots \square B_n \rightarrow S_n \text{ fi},$

- do command or repetitive command

  $S ::= \text{do } B_1 \rightarrow S_1 \square \ldots \square B_n \rightarrow S_n \text{ od}.$

Also written as

$\text{if } \square_{i=1}^{n} B_i \rightarrow S_i \text{ fi and do } \square_{i=1}^{n} B_i \rightarrow S_i \text{ od}.$

*Guarded Command*

*Guard*

# Execution

- if $B_1 \rightarrow S_1 \square . . . \square B_n \rightarrow S_n$ fi
  - If more than one guard $B_i$ evaluates to true, any of the corresponding statements $S_i$ may be executed next.
  - If all guards evaluate to false, the alternative command will signal a failure.

# 10.2 Semantics

(xx) $< \text{if } \square_{i=1}^{n} B_i \rightarrow S_i \text{ fi }, \sigma > \; \rightarrow \; < S_i, \sigma >$

where $\sigma \models B_i$ and $i \in \{1, . . . , n\}$,

(xxi) $< \text{if } \square_{i=1}^{n} B_i \rightarrow S_i \text{ fi }, \sigma > \; \rightarrow \; < E, \textbf{fail} >$

Here fail is an exceptional state.

where $\sigma \models \bigwedge_{i=1}^{n} \neg B_i$,

(xxii) $< \text{do } \square_{i=1}^{n} \rightarrow S_i \text{ od}, \sigma > \; \rightarrow \; < S_i; \text{do } \square_{i=1}^{n} B_i \rightarrow S_i \text{ od}, \sigma >$

where $\sigma \models B_i$ and $i \in \{1, . . . , n\}$,

(xxiii) $< \text{do } \square_{i=1}^{n} B_i \rightarrow S_i \text{ od}, \sigma > \; \rightarrow \; < E, \sigma >$

where $\sigma \models \bigwedge_{i=1}^{n} \neg B_i$.

# $\mathcal{M}$ [[S]]

- **Partial correctness semantics:**

$$\mathcal{M}[[S]](\sigma) = \{\tau \mid < S, \sigma > \to^* < E, \tau >\},$$

- **Total correctness semantics:**

$$\mathcal{M}_{tot}[[S]](\sigma) = \mathcal{M}[[S]](\sigma)$$

$$\cup \{\perp \mid S \text{ can diverge from } \sigma\}$$

$$\cup \{\text{fail} \mid S \text{ can fail from } \sigma\}.$$

*Chapter 3.7*

# Properties of Semantics

- **Lemma 10.1. (Bounded Nondeterminism)**

Bounded

Let $S$ be a nondeterministic program and $\sigma$ a proper state. Then $M_{tot}$ [[S]](σ) is either finite or it contains $\perp$.

- **Lemma 10.2. (Correspondence)**

(i) $M_{tot}$[[if $B$ then $S_1$ else $S_2$ fi ]] = $M_{tot}$ [[if $B \rightarrow S_1 \square \neg B \rightarrow S_2$ fi]],

(ii) $M_{tot}$[[while $B$ do $S$ od ]] = $M_{tot}$ [[do $B \rightarrow S$ od]].

if $B$ then $S_1$ else $S_2$ fi ≡ if $B \rightarrow S_1 \square \neg B \rightarrow S_2$ fi

while $B$ do $S$ od ≡ do $B \rightarrow S$ od.

# Syntactic Approximation of a loop

Let $\Omega$ be a nondeterministic program such that $M[[\Omega]](\sigma) = \emptyset$ holds for all proper states $\sigma$.

*kth* syntactic approximation of a loop do $B_i \rightarrow S_i$ od:

$(\text{do } \square_{i=1}^{n} B_i \rightarrow S_i \text{ od })^0 = \Omega$

$(\text{do } \square_{i=1}^{n} B_i \rightarrow S_i \text{ od})^{k+1} = \text{if } \square_{i=1}^{n}\ B_i \rightarrow S_i;\ (\text{do } \square_{i=1}^{n} B_i \rightarrow S_i \text{ od})^k$

$\square\ \wedge_{i=1}^{n} \neg B_i \rightarrow skip$

fi.

# $\mathcal{N}[[S]]$

Let $\mathcal{N}$ stand for $\mathcal{M}$ or $\mathcal{M}_{tot}$. We extend $\mathcal{N}$ to deal with the error states $\perp$ and fail by

$$\mathcal{M}[[S]](\perp) = \mathcal{M}[[S]](\text{fail}) = \emptyset$$

and

$$\mathcal{M}_{tot}[[S]](\perp) = \{\perp\} \text{ and } \mathcal{M}_{tot}[[S]](\text{fail}) = \{\text{fail}\}$$

and to deal with sets $X \subseteq \Sigma \cup \{\perp\} \cup \{\text{fail}\}$ by

$$\mathcal{N}[[S]](X) = \bigcup_{\sigma \in X} \mathcal{N}[[S]](\sigma).$$

# Lemma 10.3

**Lemma 10.3. (Input/Output)**

(i) $\mathcal{N}$[[S]] is monotonic; that is, $X \subseteq Y \subseteq \Sigma \cup \{\perp\}$ implies $\mathcal{N}$[[S]](X) ⊆ $\mathcal{N}$[[S]](Y).

(ii) $\mathcal{N}[[S_1; S_2]](X) = \mathcal{N}[[S_2]](\mathcal{N}[[S_1]](X))$.

(iii) $\mathcal{M}[[(S_1; S_2); S_3]](X) = \mathcal{N}[[S_1; (S_2; S_3)]](X)$.

(iv) $\mathcal{M}[[\text{if } \square_{i=1}^{n} B_i \rightarrow S_i \text{ fi}]](X) = \cup_{i=1}^{n} \mathcal{M}[[S_i]](X \cap [[B_i]])$.

(v) if $X \subseteq \cup_{i=1}^{n} [[B_i]]$ then

$$\mathcal{M}_{tot}[[\text{if } \square_{i=1}^{n} B_i \rightarrow S_i \text{ fi}]](X) = \cup_{i=1}^{n} \mathcal{M}_{tot}[[S_i]](X \cap [[B_i]]).$$

(vi) $\mathcal{M}[[\text{do } \square_{i=1}^{n} B_i \rightarrow S_i \text{ od}]] = \cup_{k=0}^{\infty} \mathcal{M}[[(\text{do } \square_{i=1}^{n} B_i \rightarrow S_i \text{ od})^k]]$.

# 10.4 Verification

- RULE 4: CONDITIONAL

$$\frac{\{p \land B\}\ S_1\ \{q\},\ \{p \land \neg B\}\ S_2\ \{q\}}{\{p\}\ \text{if } B \text{ then } S_1 \text{ else } S_2 \text{ fi}\ \{q\}}$$

- RULE 5: LOOP

$$\frac{\{p \land B\}\ S\ \{p\}}{\{p\}\ \text{while } B \text{ do } S \text{ od}\ \{p \land \neg B\}}$$

## Partial Correctness

- RULE 30: ALTERNATIVE COMMAND

$$\frac{\{p \land B_i\}\ S_i\ \{q\},\ i \in \{1, \ldots, n\}}{\{p\}\ \text{if } \square_{i=1}^{n}\ B_i \rightarrow S_i \text{ fi}\ \{q\}}$$

- RULE 31: REPETITIVE COMMAND

Invariant

$$\frac{\{p \land B_i\}\ S_i\ \{p\},\ i \in \{1, \ldots, n\}}{\{p\}\ \text{do } \square_{i=1}^{n}\ B_i \rightarrow S_i \text{ od}\ \{p \land \bigwedge_{i=1}^{n} \neg B_i\}}$$

# Proof System

- **PROOF SYSTEM PN :**

**This system consists of the group of axioms and rules 1, 2, 3, 6, 30, 31 and A2–A6.**

AXIOM 1: SKIP

$$\{p\}\ skip\ \{p\}$$

AXIOM 2: ASSIGNMENT

$$\{p[u := t]\}\ u := t\ \{p\}$$

RULE 3: COMPOSITION

$$\frac{\{p\}\ S_1\ \{r\}, \{r\}\ S_2\ \{q\}}{\{p\}\ S_1;\ S_2\ \{q\}}$$

RULE 6: CONSEQUENCE

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

# Total Correctness

We have to show absence of failures and absence of divergence.

- RULE 32: ALTERNATIVE COMMAND II

Failures arise only if none of the guards in an alternative command evaluates to true

$$\frac{p \rightarrow \bigvee_{i=1}^{n} B_i, \quad \{p \wedge B_i\}\ S_i\ \{q\},\ i \in \{1, \ldots, n\}}{\{p\}\ \text{if } \square_{i=1}^{n}\ B_i \rightarrow S_i \text{ fi } \{q\}}$$

- RULE 33: REPETITIVE COMMAND II

$$\frac{\{p \wedge B_i\}\ S_i\ \{p\},\ i \in \{1, \ldots, n\}, \quad \{p \wedge B_i \wedge t = z\}\ S_i\ \{t < z\},\ i \in \{1, \ldots, n\}, \quad p \rightarrow t \geq 0}{\{p\}\ \text{do } \square_{i=1}^{n}\ B_i \rightarrow S_i \text{ od } \{p \wedge \bigwedge_{i=1}^{n} \neg B_i\}}$$

$$\frac{\{p \wedge B\}\ S\ \{p\}, \quad \{p \wedge B \wedge t = z\}\ S\ \{t < z\}, \quad p \rightarrow t \geq 0}{\{p\}\ \text{while } B \text{ do } S \text{ od } \{p \wedge \neg B\}}$$

# PROOF SYSTEM

- **PROOF SYSTEM TN :**

This system consists of the group of axioms and rules1, 2, 3, 6, 32, 33 and A3–A6.

# Proof Outline for Total Correctness

- (xiii)

$$\frac{p \rightarrow \vee_{i=1}^{n} B_i, \quad \{p \wedge B_i\}\ S_i^*\{q\},\ i \in \{1, \dots, n\}}{\{p\}\ \text{if}\ \square_{i=1}^{n} B_i \rightarrow \{p \wedge B_i\}\ S_i^*\ \{q\}\ \text{fi}\ \{q\}}$$

- (xiv)

$$\frac{\{p \wedge B_i\}\ S_i^*\ \{p\},\ i \in \{1, \dots, n\}, \quad \{p \wedge B_i \wedge t = z\}\ S_i^{**}\{t < z\},\ i \in \{1, \dots, n\}, \quad p \rightarrow t \geq 0}{\{\text{inv}: p\}\{\text{bd}: t\}\ \text{do}\ \square_{i=1}^{n} B_i \rightarrow \{p \wedge B_i\}\ S_i^*\{p\}\ \text{od}\ \{p \wedge \wedge_{i=1}^{n} \neg B_i\}}$$

# Example 10.1

**Proof outline for total correctness** of the program GCD (mentioned in the beginning of Section 10.3):

$\{x = x_0 \land y = y_0 \land x_0 > 0 \land y_0 > 0\}$

$\{\text{inv} : p\}\ \{\text{bd} : t\}$

do $x > y \rightarrow \{p \land x > y\}$

$\quad\quad x := x - y$

$\quad \square\ x < y \rightarrow \{p \land x < y\}$

$\quad\quad y := y - x$

od

$\{p \land \neg(x > y) \land \neg(x < y)\}$

$\{x = y \land y = \text{gcd}(x_0, y_0)\}$.

The greatest common divisor of

**Invariant:**

$p \equiv \text{gcd}(x, y) = \text{gcd}(x_0, y_0) \land x > 0 \land y > 0$

**Bound function:**

$t \equiv x + y$.

**Example:**

If $x=16$ and $y=5$,then

$x>y \rightarrow$ $x=11$ $y=5$;

$x=6$ $y=5$;

$x=1$ $y=5$;

$x<y \rightarrow$ $x=1$ $y=4$;

$x=1$ $y=3$;

$x=1$ $y=2$;

$x=1$ $y=1$;

When exiting from loop, 'x=y' is satisfied!

# Soundness

**Theorem 10.1. (Soundness of PN and TN)**

(i) The proof system PN is sound for partial correctness of nondeterministic programs.

(ii) The proof system TN is sound for total correctness of nondeterministic programs.

Thank you