程序验证方法

研究生课程

Chapter 7 (7.3, 7.4)
Disjoint Parallel Programs

朱惠彪
华东师范大学软件工程学院

Review ——PW&PT of While Program

PROOF SYSTEM PW :

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 4: CONDITIONAL

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

RULE 5: LOOP

$$\frac{\{p \wedge B\}\ S\ \{p\}}{\{p\}\ \textbf{while}\ B\ \textbf{do}\ S\ \textbf{od}\ \{p \wedge \neg B\}}$$

RULE 6: CONSEQUENCE

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

Review ——PW&PT of While Program

PROOF SYSTEM TW :
This system consists of the group of axioms and rules 1–4, 6, 7.

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 4: CONDITIONAL

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

RULE 7: LOOP II

$$\frac{\begin{array}{l}\{p \wedge B\}\ S\ \{p\},\\ \{p \wedge B \wedge t = z\}\ S\ \{t < z\},\\ p \ \rightarrow t \geq 0\end{array}}{\{p\}\ \textbf{while}\ B\ \textbf{do}\ S\ \textbf{od}\ \{p \wedge \neg\ B\}}$$

RULE 6: CONSEQUENCE

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

PW&PT of Disjoint Parallel Program

partial correctness

$$\models \{p\}\ S\ \{q\} \text{ iff } \mathcal{M}[\![S]\!]([\![p]\!]) \subseteq [\![q]\!]$$

total correctness

$$\models_{tot} \{p\}\ S\ \{q\} \text{ iff } \mathcal{M}_{tot}[\![S]\!]([\![p]\!]) \subseteq [\![q]\!]$$

Review

$$\mathcal{M}[\![S]\!] : \Sigma \rightarrow \mathcal{P}(\Sigma) \qquad \mathcal{M}[\![S]\!](\sigma) = \{\tau \mid < S, \sigma > \ \rightarrow^* \ < E, \tau >\}$$

$$\mathcal{M}_{tot}[\![S]\!] : \Sigma \rightarrow \mathcal{P}(\Sigma \cup \{\perp\}) \qquad \mathcal{M}_{tot}[\![S]\!](\sigma) = \mathcal{M}[\![S]\!](\sigma) \cup \{\perp \mid S \text{ can diverge from } \sigma\}$$

PW&PT of Disjoint Parallel Program

$\boxed{\text{Review}}$

Lemma 7.7. (Sequentialization) *Let $S_1, \ldots, S_n$ be pairwise disjoint* **while** *programs. Then*

$$\mathcal{M}[\![[S_1 \| \ldots \| S_n]]\!] = \mathcal{M}[\![S_1;\ \ldots;\ S_n]\!],$$

and

$$\mathcal{M}_{tot}[\![[S_1 \| \ldots \| S_n]]\!] = \mathcal{M}_{tot}[\![S_1;\ \ldots;\ S_n]\!].$$

RULE 23: SEQUENTIALIZATION

$$\frac{\{p\}\ S_1;\ \ldots;\ S_n\ \{q\}}{\{p\}\ [S_1 \| \ldots \| S_n]\ \{q\}}$$

Sound for both partial and total correctness by Lemma 7.7

We get a sound PW (PT) proof system of Disjoint Parallel Programs by adding RULE 23 to previous PW (PT) of While Programs.

PW&PT of Disjoint Parallel Program

Drawback of RULE 23

It's convenient to use RULE 23 to show:

$$\models_{tot} \{x = y\}\ [x := x + 1 \| y := y + 1]\ \{x = y\};$$

However, considering more complex situations:

S1 …Sn are independent programs, each has its owe pre- and post-assertions.

We want to prove: $\models_{tot} \{p\}\ [S_1 \| \ldots \| S_n]\ \{q\}$

First we should show: $\models_{tot} \{p\}\ S_1;\ \ldots;\ S_n\ \{q\}$

Then by the composition rule: $\{p\}\ S_1\ \{r_1\}, \ldots, \{r_{i-1}\}\ S_i\ \{r_i\}, \ldots, \{r_{n-1}\}\ S_n\ \{q\}$

Drawback, can we simplify this work?

the pre- and post-assertions of different components of $[S1 \| \ldots \| Sn]$ must fit exactly.

PW&PT of Disjoint Parallel Program

Solution :

RULE 24: DISJOINT PARALLELISM

$$\frac{\{p_i\}\ S_i\ \{q_i\}, i \in \{1, \ldots, n\}}{\{\bigwedge_{i=1}^{n}\ p_i\}\ [S_1\|\ldots\|S_n]\ \{\bigwedge_{i=1}^{n}\ q_i\}}$$

where $free(p_i, q_i) \cap change(S_j) = \emptyset$ for $i \neq j$.

Due to this restriction,
RULE 24 is weaker than RULE 23.

Each $\{p_i\}\ S_i\ \{q_i\}$ can be proved in PW or TW for while programs, which means we cut the whole prove process into several parts, and then we combine the results using RULE 24.

$$\{x = y\}\ [x := x + 1\|y := y + 1]\ \{x = y\}$$

For example, the correctness formula above cannot be proved using RULE 24.

We have an improved solution to solve this limitation.

PW&PT of Disjoint Parallel Program

Improved solution :

Clearly, we can use a $\boxed{\text{fresh variable } z}$ to prove

$$\{x = z\}\ x := x + 1\ \{x = z + 1\}$$

and

$$\{y = z\}\ y := y + 1\ \{y = z + 1\}.$$

$\Downarrow$ RULE 24

$$\{x = z \wedge y = z\}\ [x := x + 1 \| y := y + 1]\ \{x = z + 1 \wedge y = z + 1\}.$$

$x = z + 1 \wedge y = z + 1 \rightarrow x = y$ $\Downarrow$ Consequence RULE

$$\boxed{\{x = z \wedge y = z\}}\ [x := x + 1 \| y := y + 1]\ \{x = y\}.$$

$\Downarrow$

What to do next?

PW&PT of Disjoint Parallel Program

Improved solution :

$$\boxed{\{x = z \wedge y = z\}}\ [x := x + 1 \| y := y + 1]\ \{x = y\}.$$

What to do next?

Obviously we have: $\{x = y\}\ z := x\ \{x = z \wedge y = z\};$

Composition RULE

$$\{x = y\}\ z := x;\ [x := x + 1 \| y := y + 1]\ \{x = y\}$$

next?

Solution: RULE 25 to drop z:=x

Auxiliary Variables

Definition 7.5. Let A be a set of simple variables in a program S. We call A a *set of auxiliary variables* of S if each variable from A occurs in S only in assignments of the form $z := t$ with $z \in A$. $\square$

They do not appear in Boolean expressions. → They cannot influence the control flow in S.

They are not used in assignments to variables outside of A. → They cannot influence the data flow in S.

e.g. $S \equiv z := x;\ [x := x + 1 \| y := y + 1].$

$\emptyset, \{y\}, \{z\}, \{x, z\}, \{y, z\}, \{x, y, z\}$ are all sets of auxiliary variables of S.

Auxiliary Variables

RULE 25: AUXILIARY VARIABLES

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

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

Attention: taking $A = \{y\}$ and

$$S \equiv z := x;\ [x := x + 1 \| y := y + 1],$$

the literal deletion of the assignment $y := y + 1$ would yield

$$z := x;\ [x := x + 1 \| \boxed{\bullet}\ ]$$

hole

We fill in such "holes" by skip:

$$S' \equiv z := x;\ [x := x + 1 \| skip].$$

PW&PT of Disjoint Parallel Program

Summarizing, for proofs of *partial* correctness of disjoint *parallel* programs we use the following proof system *PP*.

PROOF SYSTEM *PP* :
This system consists of the group of axioms and rules 1–6, 24, 25 and A2–A6.

For proofs of *total* correctness of disjoint *parallel* programs we use the following proof system *TP*.

PROOF SYSTEM *TP* :
This system consists of the group of axioms and rules 1–5, 7, 24, 25 and A3–A6.

Case Study: Find Positive Element

An integer array a.

A constant N ≥ 1.

The task is to find

the smallest index k ∈ {1,...,N} with a[k] > 0 if such an element of a exists;
otherwise the dummy value k = N + 1 should be returned.

$\{\mathbf{true}\}$
$FIND$
$\{1 \leq k \leq N + 1 \wedge \forall(1 \leq l < k) : a[l] \leq 0 \wedge (k \leq N \rightarrow a[k] > 0)\}$

We'll prove this correctness formula in the sense of total correctness.

Clearly, we require $a \notin change(FIND)$

Case Study: Find Positive Element

$$\{\mathbf{true}\}$$
$$\mathit{FIND}$$
$$\{1 \le k \le N+1 \wedge \forall (1 \le l < k) : a[l] \le 0 \wedge (k \le N \rightarrow a[k] > 0)\}$$

End of the search

We split FIND into two parallel components :

$$\begin{aligned} S_1 \equiv\ & \textbf{while } i < \boxed{oddtop} \textbf{ do} \\ & \quad \textbf{if } a[i] > 0 \textbf{ then } oddtop := i \\ & \quad\quad \textbf{else } i := i + 2 \textbf{ fi} \\ & \textbf{od}. \end{aligned}$$

Odd index

$$\begin{aligned} S_2 \equiv\ & \textbf{while } j < \boxed{eventop} \textbf{ do} \\ & \quad \textbf{if } a[j] > 0 \textbf{ then } eventop := j \\ & \quad\quad \textbf{else } j := j + 2 \textbf{ fi} \\ & \textbf{od}. \end{aligned}$$

Even index

$$\begin{aligned} \mathit{FIND} \equiv\ & i := 1;\ j := 2;\ oddtop := N+1;\ eventop := N+1; \\ & [S_1 \| S_2]; \\ & k := min(oddtop, eventop). \end{aligned}$$

Case Study: Find Positive Element

$$\{\mathbf{true}\}$$
$$FIND$$
$$\{1 \le k \le N+1 \land \forall(1 \le l < k) : a[l] \le 0 \land (k \le N \to a[k] > 0)\}$$

First , we prove: $\{i = 1 \land oddtop = N+1\}\ S_1\ \{q_1\}$

An adaptation of the postcondition of FIND.

$$\begin{aligned} q_1 \equiv\ & 1 \le oddtop \le N+1 \\ \land\ & \forall l : (\boxed{odd(l)} \land 1 \le l < oddtop \to a[l] \le 0) \\ \land\ & (oddtop \le N \to a[oddtop] > 0). \end{aligned}$$

l is odd.

loop invariant p_1

$$\begin{aligned} p_1 \equiv\ & 1 \le oddtop \le N+1 \land odd(i) \land 1 \le i \le oddtop + 1 \\ \land\ & \forall l : (odd(l) \land 1 \le l < i \to a[l] \le 0) \\ \land\ & (oddtop \le N \to a[oddtop] > 0). \end{aligned}$$

bound function t_1

$$t_1 \equiv oddtop + 1 - i.$$

$$\begin{aligned} S_1 \equiv\ & \mathbf{while}\ i < oddtop\ \mathbf{do} \\ & \quad \mathbf{if}\ a[i] > 0\ \mathbf{then}\ oddtop := i \\ & \quad \mathbf{else}\ \ i := i + 2\ \mathbf{fi} \\ & \mathbf{od}. \end{aligned}$$

Odd index

$$
\begin{aligned}
p_1 \equiv \quad & 1 \le oddtop \le N+1 \wedge odd(i) \wedge 1 \le i \le oddtop + 1 \\
\wedge \;\; & \forall l : (odd(l) \wedge 1 \le l < i \rightarrow a[l] \le 0) \\
\wedge \;\; & (oddtop \le N \rightarrow a[oddtop] > 0).
\end{aligned}
$$

$\{\mathbf{inv} : p_1\}\{\mathbf{bd} : t_1\}$
while $i < oddtop$ **do**
  $\{p_1 \wedge i < oddtop\}$
  if $a[i] > 0$ **then** $\{p_1 \wedge i < oddtop \wedge a[i] > 0\}$

$$
\begin{aligned}
\{ \quad & 1 \le i \le N+1 \wedge odd(i) \wedge 1 \le i \le i+1 \\
\wedge \;\; & \forall l : (odd(l) \wedge 1 \le l < i \rightarrow a[l] \le 0) \\
\wedge \;\; & (i \le N \rightarrow a[i] > 0)\}
\end{aligned}
$$

    $oddtop := i$
    $\{p_1\}$
  else $\{p_1 \wedge i < oddtop \wedge a[i] \le 0\}$

$$
\begin{aligned}
\{ \quad & 1 \le oddtop \le N+1 \wedge odd(i+2) \\
\wedge \;\; & 1 \le i+2 \le oddtop + 1 \\
\wedge \;\; & \forall l : (odd(l) \wedge 1 \le l < i+2 \rightarrow a[l] \le 0) \\
\wedge \;\; & (oddtop \le N \rightarrow a[oddtop] > 0)\}
\end{aligned}
$$

    $i := i + 2$
    $\{p_1\}$
  fi
  $\{p_1\}$
od
$\{p_1 \wedge oddtop \le i\}$
$\{q_1\}$.

AXIOM 2: ASSIGNMENT

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

$$
\begin{aligned}
S_1 \equiv \ & \mathbf{while}\ i < oddtop\ \mathbf{do} \\
& \quad \mathbf{if}\ a[i] > 0\ \mathbf{then}\ oddtop := i \\
& \quad \quad \mathbf{else}\ \ i := i + 2\ \mathbf{fi} \\
& \mathbf{od}.
\end{aligned}
$$

Odd index

$$\{\textbf{inv}: p_1\}\{\textbf{bd}: t_1\}$$
$$\textbf{while } i < oddtop \textbf{ do}$$
$$\{p_1 \wedge i < oddtop\}$$
$$\textbf{if } a[i] > 0 \textbf{ then } \boxed{\{p_1 \wedge i < oddtop \wedge a[i] > 0\}}$$
$$\begin{array}{l} \{ \quad 1 \le i \le N+1 \wedge odd(i) \wedge 1 \le i \le i+1 \\ \quad \wedge \ \forall l : (odd(l) \wedge 1 \le l < i \rightarrow a[l] \le 0) \\ \quad \wedge \ (i \le N \rightarrow a[i] > 0)\} \\ oddtop := i \\ \{p_1\} \end{array}$$
$$\textbf{else } \{p_1 \wedge i < oddtop \wedge a[i] \le 0\}$$
$$\begin{array}{l} \{ \quad 1 \le oddtop \le N+1 \wedge odd(i+2) \\ \quad \wedge \ 1 \le i+2 \le oddtop+1 \\ \quad \wedge \ \forall l : (odd(l) \wedge 1 \le l < i+2 \rightarrow a[l] \le 0) \\ \quad \wedge \ (oddtop \le N \rightarrow a[oddtop] > 0)\} \\ i := i+2 \\ \{p_1\} \end{array}$$
$$\textbf{fi}$$
$$\{p_1\}$$
$$\textbf{od}$$
$$\{p_1 \wedge oddtop \le i\}$$
$$\{q_1\}.$$

$$\begin{array}{ll} p_1 \equiv & 1 \le oddtop \le N+1 \wedge odd(i) \wedge 1 \le i \le oddtop+1 \\ & \wedge \ \forall l : (odd(l) \wedge 1 \le l < i \rightarrow a[l] \le 0) \\ & \wedge \ (oddtop \le N \rightarrow a[oddtop] > 0). \end{array}$$

RULE 6: CONSEQUENCE

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

$$\begin{array}{l} S_1 \equiv \textbf{while } i < oddtop \textbf{ do} \\ \qquad \textbf{if } a[i] > 0 \textbf{ then } oddtop := i \\ \qquad\qquad \textbf{else } \ i := i+2 \textbf{ fi} \\ \quad \textbf{od}. \end{array}$$

Odd index

$$p_1 \equiv \quad 1 \leq oddtop \leq N+1 \wedge odd(i) \wedge 1 \leq i \leq oddtop+1$$
$$\wedge\ \forall l : (odd(l) \wedge 1 \leq l < i \rightarrow a[l] \leq 0)$$
$$\wedge\ (oddtop \leq N \rightarrow a[oddtop] > 0).$$

$\{\textbf{inv} : p_1\}\{\textbf{bd} : t_1\}$
while $i < oddtop$ **do**
$\{p_1 \wedge i < oddtop\}$
if $a[i] > 0$ **then** $\{p_1 \wedge i < oddtop \wedge a[i] > 0\}$

$$\{ \quad 1 \leq i \leq N+1 \wedge odd(i) \wedge 1 \leq i \leq i+1$$
$$\wedge\ \forall l : (odd(l) \wedge 1 \leq l < i \rightarrow a[l] \leq 0)$$
$$\wedge\ (i \leq N \rightarrow a[i] > 0)\}$$

$oddtop := i$
$\{p_1\}$
else $\{p_1 \wedge i < oddtop \wedge a[i] \leq 0\}$

$$\{ \quad 1 \leq oddtop \leq N+1 \wedge odd(i+2)$$
$$\wedge\ 1 \leq i+2 \leq oddtop+1$$
$$\wedge\ \forall l : (odd(l) \wedge 1 \leq l < i+2 \rightarrow a[l] \leq 0)$$
$$\wedge\ (oddtop \leq N \rightarrow a[oddtop] > 0)\}$$

$i := i+2$
$\{p_1\}$
fi
$\{p_1\}$
od
$\{p_1 \wedge oddtop \leq i\}$
$\{q_1\}$.

Proof of the ELSE part is the same as IF part's.

$S_1 \equiv$ **while** $i < oddtop$ **do**
if $a[i] > 0$ **then** $oddtop := i$
else $i := i+2$ **fi**
od.

Odd index

$$p_1 \equiv \quad 1 \le oddtop \le N+1 \wedge odd(i) \wedge 1 \le i \le oddtop+1$$
$$\wedge \ \forall l : (odd(l) \wedge 1 \le l < i \rightarrow a[l] \le 0)$$
$$\wedge \ (oddtop \le N \rightarrow a[oddtop] > 0).$$

$\{\mathbf{inv} : p_1\}\{\mathbf{bd} : t_1\}$
while $i < oddtop$ **do**
 $\{p_1 \wedge i < oddtop\}$
 if $a[i] > 0$ **then** $\{p_1 \wedge i < oddtop \wedge a[i] > 0\}$
  $\{\quad 1 \le i \le N+1 \wedge odd(i) \wedge 1 \le i \le i+1$
  $\wedge \ \forall l : (odd(l) \wedge 1 \le l < i \rightarrow a[l] \le 0)$
  $\wedge \ (i \le N \rightarrow a[i] > 0)\}$
  $oddtop := i$
  $\{p_1\}$
 else $\{p_1 \wedge i < oddtop \wedge a[i] \le 0\}$
  $\{\quad 1 \le oddtop \le N+1 \wedge odd(i+2)$
  $\wedge \ 1 \le i+2 \le oddtop+1$
  $\wedge \ \forall l : (odd(l) \wedge 1 \le l < i+2 \rightarrow a[l] \le 0)$
  $\wedge \ (oddtop \le N \rightarrow a[oddtop] > 0)\}$
  $i := i+2$
  $\{p_1\}$
 fi
 $\{p_1\}$
od
$\boxed{\{p_1 \wedge oddtop \le i\}}$
$\boxed{\{q_1\}}.$

RULE 6: CONSEQUENCE

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

$S_1 \equiv$ **while** $i < oddtop$ **do**
  if $a[i] > 0$ **then** $oddtop := i$
   else $i := i+2$ **fi**
 od.

Odd index

RULE 24: DISJOINT PARALLELISM

$$\frac{\{p_i\}\ S_i\ \{q_i\}, i \in \{1, \ldots, n\}}{\{\bigwedge_{i=1}^{n}\ p_i\}\ [S_1\|\ldots\|S_n]\ \{\bigwedge_{i=1}^{n}\ q_i\}}$$

$$\{p_1 \wedge p_2\}$$
$$[S_1\|S_2]$$
$$\{q_1 \wedge q_2\}.$$

where $free(p_i, q_i) \cap change(S_j) = \emptyset$ for $i \neq j$.

$$\boxed{\begin{aligned} p_1 \equiv \quad & 1 \leq oddtop \leq N+1 \wedge odd(i) \wedge 1 \leq i \leq oddtop+1 \\ & \wedge\ \forall l : (odd(l) \wedge 1 \leq l < i \rightarrow a[l] \leq 0) \\ & \wedge\ (oddtop \leq N \rightarrow a[oddtop] > 0). \end{aligned}}$$

$$\boxed{\begin{aligned} p_2 \equiv \quad & 2 \leq eventop \leq N+1 \wedge even(j) \wedge j \leq eventop+1 \\ & \wedge\ \forall l : (even(l) \wedge 1 \leq l < j \rightarrow a[l] \leq 0) \\ & \wedge\ (eventop \leq N \rightarrow a[eventop] > 0), \end{aligned}}$$

$$
\begin{aligned}
FIND \equiv\ & i := 1;\ j := 2;\ oddtop := N+1;\ eventop := N+1; \\
& [S_1 \| S_2]; \\
& k := min(oddtop, eventop).
\end{aligned}
$$

$\{\mathbf{true}\}$
$i := 1;\ j := 2;\ oddtop := N+1;\ eventop := N+1;$
$\{p_1\ \wedge\ p_2\}$

$[S_1 \| S_2];$

$\{q_1\ \wedge\ q_2\}$

$$
\begin{aligned}
\{\quad & 1 \le min(oddtop, eventop) \le N+1 \\
\wedge\ & \forall(1 \le l < min(oddtop, eventop)) : a[l] \le 0 \\
\wedge\ & (min(oddtop, eventop) \le N \rightarrow a[min(oddtop, eventop)] > 0)\}
\end{aligned}
$$

$k := min(oddtop, eventop)$
$\{1 \le k \le N+1\ \wedge\ \forall(1 \le l < k) : a[l] \le 0\ \wedge\ (k \le N \rightarrow a[k] > 0)\}.$

ASSIGNMENT RULE
Then
CONSEQUENCE RULE

$$FIND \equiv i := 1;\ j := 2;\ oddtop := N+1;\ eventop := N+1;$$
$$[S_1 \| S_2];$$
$$k := min(oddtop, eventop).$$

$\{\mathbf{true}\}$
$i := 1;\ j := 2;\ oddtop := N+1;\ eventop := N+1;$
$\{p_1 \ \wedge\ p_2\}$

$[S_1 \| S_2];$

$\{q_1 \ \wedge\ q_2\}$
$\{\quad 1 \leq min(oddtop, eventop) \leq N+1$
$\quad \wedge\ \forall(1 \leq l < min(oddtop, eventop)) : a[l] \leq 0$
$\quad \wedge\ (min(oddtop, eventop) \leq N \rightarrow a[min(oddtop, eventop)] > 0)\}$
$k := min(oddtop, eventop)$
$\{1 \leq k \leq N+1 \ \wedge\ \forall(1 \leq l < k) : a[l] \leq 0 \ \wedge\ (k \leq N \rightarrow a[k] > 0)\}.$

CONSEQUENCE RULE

ASSIGNMENT RULE