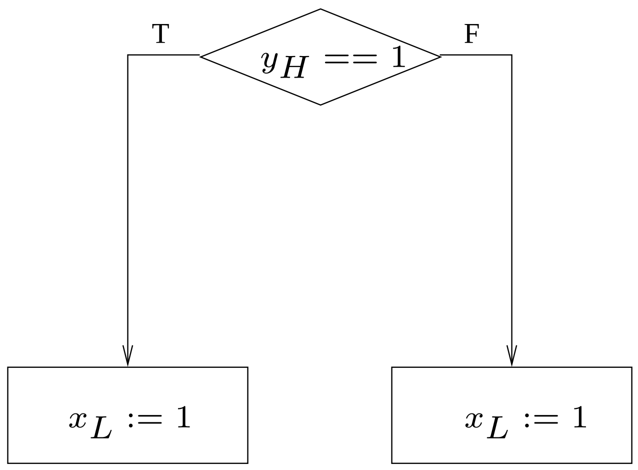
Noninterference for sequential and multi-threaded languages

Thuy Duong Vu,
Programming Research Group,
University of Amsterdam
tdvu@science.uva.nl

Noninterference

Program variables are classified into ($High$) and (Low) security levels.

Noninterference means that the high output of the program does not depends on the low input.



Motivations

- Noninterference is an important property of security information flow.
- The previous approaches appear confusing when parallelism is introduced in the language.

Our approach

- We define *behavioral noninterference* in thread algebra, a semantics for sequential and multi-threaded languages given by [Bergstra, Loots in 2002] and [Bergstra, Middelburg in 2004].

- We define *cyclic interleaving with persistence* that turns a thread vector to a single thread.

- We show that our definition of noninterference composes under cyclic interleaving with persistence.

Basic Polarized Process Algebra (BPPA)

Let Σ be the set of *basic actions*. Each basic action is supposed to return a boolean value after its execution.

$BPPA$ is defined as follows:

- **Successful termination** S
- **deadlock** D, represents inaction behavior.
- **Postconditional composition**: $P \unlhd a \unrhd Q$, where $a \in \Sigma$,
- **Action prefix**: $a \circ P = P \unlhd a \unrhd P$.

Infinite polarized processes

Let $BPPA_\Sigma$ be the set of **finite** polarized processes. For every $n \in \mathbb{N}$, the **approximation operator** $\pi_n : BPPA_\Sigma \to BPPA_\Sigma$ is defined inductively by

$$\begin{array}{lcl}
\pi_0(P) & = & D, \\
\pi_{n+1}(S) & = & S, \\
\pi_{n+1}(D) & = & D, \\
\pi_{n+1}(P \unlhd a \unrhd Q) & = & \pi_n(P) \unlhd a \unrhd \pi_n(Q).
\end{array}$$

An infinite process P can be represented by its **projective sequence** $(P_n)_{n \in \mathbb{N}}$: $\pi_n(P_{n+1}) = P_n$.

Consider two programs:

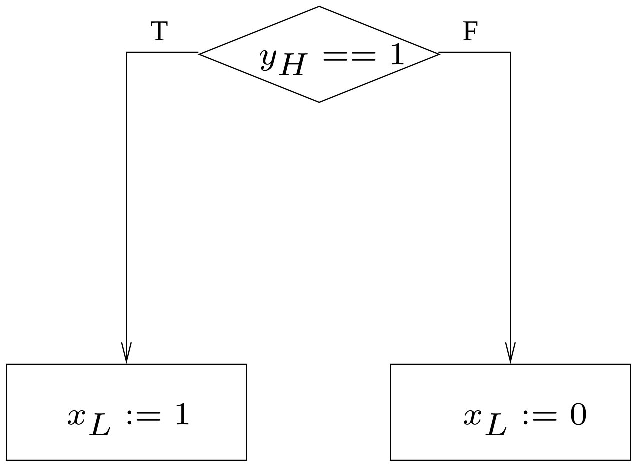
$$
\begin{array}{lcl}
X & ::= & \mathit{if}\ y_H{=}{=}1\ \mathit{then}\ x_L{:=}1\ \mathit{else}\ x_L{:=}0. \\
Y & ::= & y_H{:=}0; \\
 & & \mathit{while}\ y_H \geq 0\ \mathit{do} \\
 & & y_H{:=}y_H+1;\ \mathit{end\ while}; \\
 & & x_L{:=}1.
\end{array}
$$

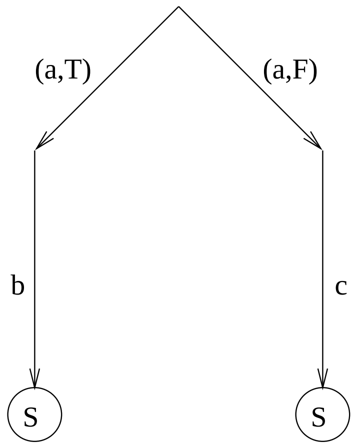
Their behaviors:

$$|X| = ((x_L{:=}1)\circ S) \trianglelefteq y_h{=}{=}1 \trianglerighteq ((x_L{:=}0)\circ S)$$

$$|Y| = (y_H := 0)\circ P$$

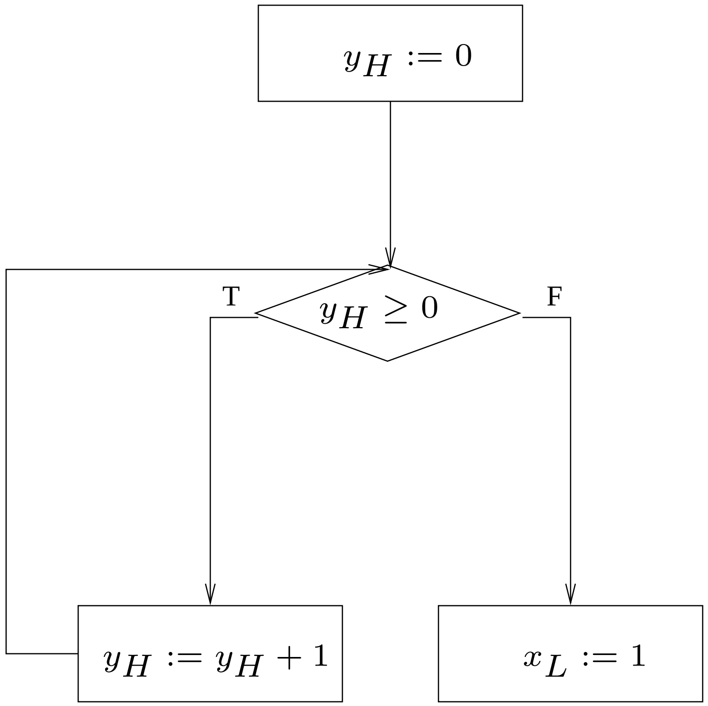
$$P = ((y_H{:=}y_H+1)\circ P) \trianglelefteq y_H \geq 0 \trianglerighteq ((x_L{:=}\ 1)\circ S).$$

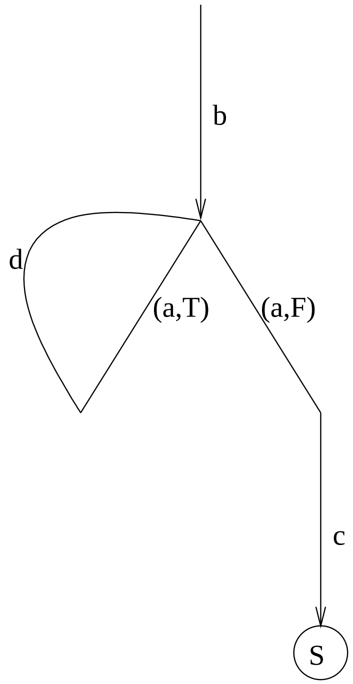




$$a = (y_H == 1),\ b = (x_L := 1),\ c = (x_L := 0)$$

$$P = (b \circ S) \unlhd a \unrhd (c \circ S)$$





$a = (y_H \geq 0)$, $b = (y_H := 0)$, $c = (x_L := 0)$, $d = (y_H := y_H + 1)$

$$Q = b \circ P$$

$$R = (d \circ R) \trianglelefteq a \trianglerighteq (c \circ S)$$

Secure and implicit actions in BPPA

- A basic action a is **secure** if it does not reveal any secret information.
- A basic action a is **implicit** if it has effect or yield on the high space.

The action $x_L := y_H$ is insecure because it reveals information of y_H. The assignment $y_H := 1$ and the test $y_H == 1$ are implicit actions.

Let Σ_s be the set of secure actions, and H the set of implicit actions.

Labeled transition systems (LTS)

A LTS with S and D is a pair $(\mathbb{P}, \longrightarrow)$:

- $\mathbb{P}$: a set of polarized processes,
- $\longrightarrow \subseteq \mathbb{P} \times (\Sigma \times \{T, F\}) \times \mathbb{P}$: a set of **transition**.

We write $P \xrightarrow{(a,\rho)} Q$ if $(P, (a, \rho), Q) \in \longrightarrow$.
If $P = P_1 \trianglelefteq a \trianglerighteq P_2$ then $P \xrightarrow{a,T} P_1$ and $P \xrightarrow{a,F} P_2$.

Branching bisimulation

Assume the existence of an **internal action** t. A **branching bisimulation** relation $\mathcal{B}$ is a symmetric binary relation satisfying:

1. If $(P, Q) \in \mathcal{B}$ and $P = S$, then $\exists Q \overset{t}{\Rightarrow} Q'$ with $(P, Q') \in \mathcal{B}$ and $Q' = S$.

2. If $(P, Q) \in \mathcal{B}$ and $P \xrightarrow{a,\rho} P'$ then either:

   (a) $a = t$, and $(P', Q) \in \mathcal{B}$, or:
   (b) $\exists Q \overset{t}{\Rightarrow} Q_1 \xrightarrow{a,\rho} Q'$ such that $(P, Q_1) \in \mathcal{B}$ and $(P', Q') \in \mathcal{B}$.

The hiding operator

- The hiding operator t_H renames all implicit actions with t.

$$
\begin{array}{lcll}
t_H(S) & = & S & \\
t_H(D) & = & D & \\
t_H(P \trianglelefteq a \trianglerighteq Q) & = & t_H(P) \trianglelefteq t \trianglerighteq t_H(Q) & \textit{if} \quad a \in H \\
t_H(P \trianglelefteq a \trianglerighteq Q) & = & t_H(P) \trianglelefteq a \trianglerighteq t_H(Q) & \textit{otherwise.}
\end{array}
$$

Divergence

- A polarized process P **diverges immediately** if $P = P_0 \xrightarrow{t,\rho_0} P_1 \xrightarrow{t,\rho_1} \cdots$

- A polarized process P is **divergent** if it has a descendant that diverges immediately.

Behavioral noninterference

Let $\sigma(P)$ be the set of actions occurring in P.
A polarized process P is **behaviorally noninterferent** ($P \in$ **BNI**) if

1. $\sigma(P) \subseteq \Sigma_s$,

2. $t_H(P)$ is non-divergent, and

3. for all descendants Q of P, if $Q \xrightarrow{a,\rho} Q'$ with $a \in H$ then $t_H(Q) \Leftrightarrow_b t_H(Q')$.

Thread algebra

- **Thread algebra** is an extension of BPPA.
- A **thread** is considered as a polarized process.
- A **thread vector** is a sequence of threads.
- **Strategic interleaving operators** turn a thread vector of arbitrary length into a single thread.
- This single thread obtained via a strategic interleaving operator is also called a **multi-thread**.

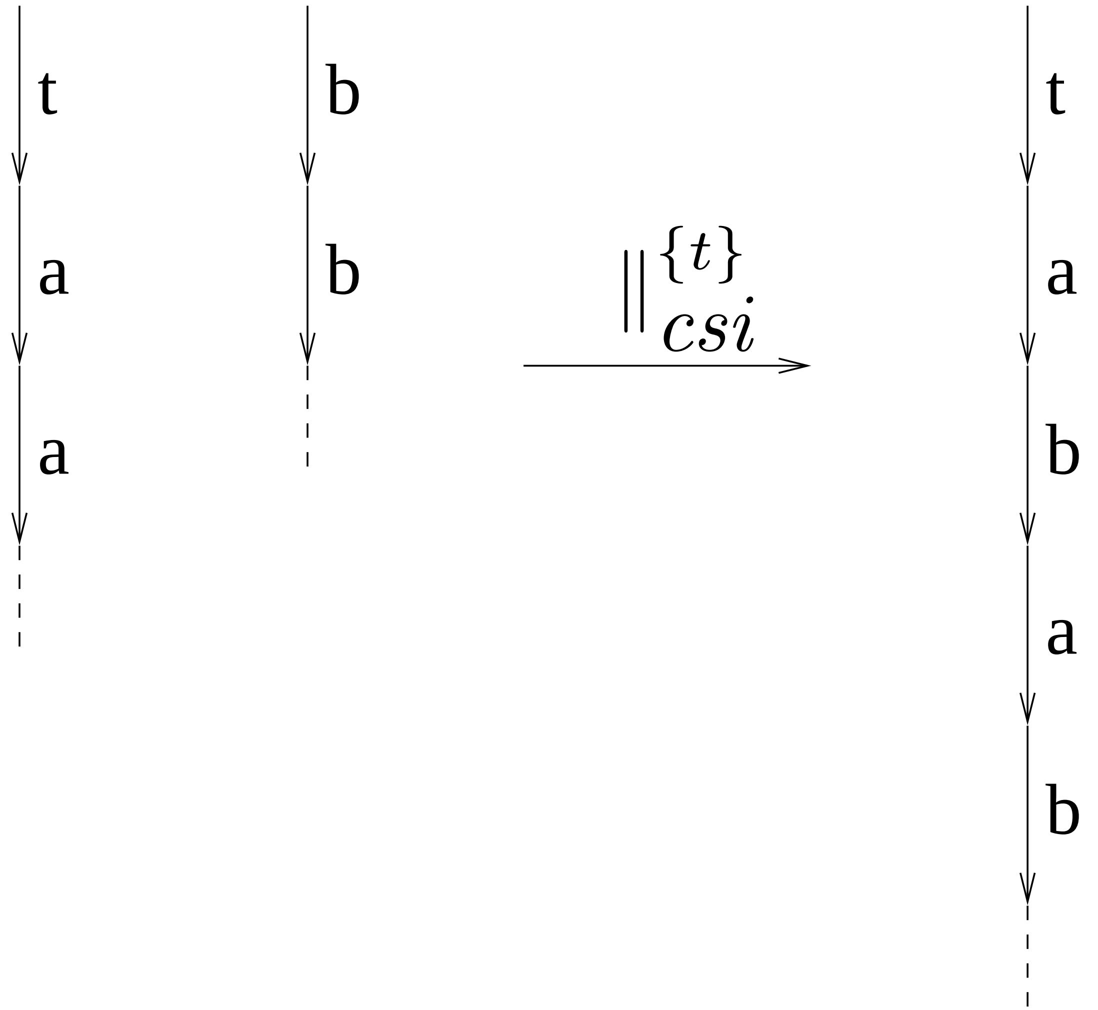
The cyclic interleaving with persistence operator

The axioms for the **cyclic interleaving with persistence operator** are given as follows:

$$
\begin{array}{lcl}
\|_{csi}^{\Phi}\,(\langle\rangle) & = & S \\
\|_{csi}^{\Phi}\,(\langle S\rangle \curvearrowright \alpha) & = & \|_{csi}^{\Phi}\,(\alpha) \\
\|_{csi}^{\Phi}\,(\langle D\rangle \curvearrowright \alpha) & = & S_D(\|_{csi}^{\Phi}\,(\alpha)) \\
\|_{csi}^{\Phi}\,(\langle x \trianglelefteq a \trianglerighteq y\rangle \curvearrowright \alpha) & = & \|_{csi}^{\Phi}\,(\langle x\rangle \curvearrowright \alpha) \trianglelefteq a \trianglerighteq \|_{csi}^{\Phi}\,(\langle y\rangle \curvearrowright \alpha) \\
for\ a \in \Phi & & \\
\|_{csi}^{\Phi}\,(\langle x \trianglelefteq a \trianglerighteq y\rangle \curvearrowright \alpha) & = & \|_{csi}^{\Phi}\,(\alpha \curvearrowright \langle x\rangle) \trianglelefteq a \trianglerighteq \|_{csi}^{\Phi}\,(\alpha \curvearrowright \langle y\rangle) \\
otherwise & &
\end{array}
$$

where S_D is defined as follows:

$$\begin{aligned}
S_D(S) &= D \\
S_D(D) &= D \\
S_D(x \trianglelefteq a \trianglerighteq y) &= S_D(x) \trianglelefteq a \trianglerighteq S_D(y)
\end{aligned}$$


t
a
a
b
b
$\|^{\{t\}}_{csi}$
t
a
b
a
b

Congruence

Let P_i and Q_i $(1 \leq i \leq n)$ be non-divergent polarized processes such that $P_i \Leftrightarrow_b Q_i$. Then for any Φ such that $t \in \Phi$,

$$\|_{csi}^{\Phi} (\langle P_1 \rangle \curvearrowright \langle P_2 \rangle \curvearrowright \cdots \curvearrowright \langle P_n \rangle) \Leftrightarrow_b \|_{csi}^{\Phi} (\langle Q_1 \rangle \curvearrowright \langle Q_2 \rangle \curvearrowright \cdots \curvearrowright \langle Q_n \rangle)$$

Compositionality

Let P_i $(1 \leq i \leq n)$ be polarized processes. Then for any $\Phi \supseteq H$,

$$\forall i : 1 \leq i \leq n \wedge P_i \in BNI \Rightarrow \|_{csi}^{\Phi} (\langle P_1 \rangle \curvearrowright \langle P_2 \rangle \curvearrowright \cdots \curvearrowright \langle P_n \rangle) \in BNI.$$

Conclusion and future works

- We have defined noninterference in the setting of thread algebra.
- We have shown that our definition satisfies compositionality.
- Our work can be extended for object-oriented programming languages.
- A variant of behavioral noninterference can be defined with respect to the condition that the program terminates successfully.