Proof Rules for Invariance and Liveness Properties

Thai Son Hoang

Chair of Information Security, Department of Computer Science
Swiss Federal Institute of Technology Zürich (ETH Zürich)

3rd February 2011, Ascona Meeting

Discrete Transition Systems (Recall)

Given the following transition system S

system S

variables $\overline{v} \in \overline{T}$

initially $init(\overline{v})$

events

evt$_i$ $\widehat{=}$ $G_i(\overline{v}) \longrightarrow \overline{v} := \overline{f}_i(\overline{v})$

- $\overline{v}$ denotes the vector of variables $v_1, \ldots, v_n$.
- $init(\overline{v})$ is the initialisation.
- $G_i(\overline{v})$ is the guard of event evt$_i$.
- evt$_i$ is said to be enabled in some state s if $G_i(\overline{v})$ holds in s.
- $\overline{v} := \overline{f}_i(\overline{v})$ is the action of event evt$_i$.

Executions and Traces (of States)

Executions $\alpha = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \ldots$

Traces $\sigma = s_0, s_1, s_2, s_3, \ldots$

$\mathcal{T}(S)$ denotes the set of all traces of system S.

Example

system *Counter*	**events**
variables $c \in \mathbb{Z}$	inc $\widehat{=}$ $c \neq 5 \longrightarrow c := c + 1$
initially $c = 0$	dec $\widehat{=}$ $c > 3 \longrightarrow c := c - 1$

$\sigma_{Counter}$: $\langle 0 \rangle, \langle 1 \rangle, \langle 2 \rangle, \langle 3 \rangle, \langle 4 \rangle, \langle 5 \rangle, \langle 4 \rangle, \langle 3 \rangle, \langle 4 \rangle, \langle 5 \rangle, \ldots$

Outline

Temporal Formulas

Temporal formulas to be interpreted over traces.

- A (basic) state formula $Q(\overline{v})$ is any first-order logic formula, e.g. $0 \leq c$, $\neg(c = 0) \wedge c < 2$, $\forall m \cdot m \neq 0 \Rightarrow m \leq c$.
- The basic formulas can be extended by combining the Boolean operators $(\neg, \wedge, \vee, \Rightarrow)$ with temporal operators:
  - $\Box$: always
  - $\Diamond$: eventually
  - $\mathcal{U}$: until
- Example of extended formulas:
  - $\Box\, c \in 0\,..\,5$
  - $\Box\, c \leq 6$
  - $\Diamond\, c = 2$
  - $\Box \Diamond\, c = 2$
  - $c \leq 2 \;\mathcal{U}\; c = 2$
  - $\Box(c \leq 2 \;\mathcal{U}\; c = 2)$
  - $(\Box\, c \in 0\,..\,5) \;\wedge\; (\Diamond\, c = 2)$

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

Length and Suffixes of Traces

Let $\sigma : s_o, s_1, \ldots$ be any non-empty trace.

- The length of σ denoted by $l(\sigma)$.
    - Finite trace $\sigma : s_0, \ldots, s_k$: $l(\sigma) = k + 1$.
    - Infinite trace: $l(\sigma) = \infty$.
- For $0 \leq k < l(\sigma)$, k-suffix of σ is defined as

$$\sigma^{(k)} = s_k, s_{k+1}, \ldots \ .$$

Interpretation

$\sigma \vDash \phi$ means that a trace σ satisfies formula ϕ

- For state formula ϕ, $\sigma \vDash \phi$ if all only if s_0 satisfies ϕ.
- Boolean operators are interpreted in the natural way, e.g.

$$\sigma \vDash \phi_1 \wedge \phi_2 \quad \text{if and only if} \quad \sigma \vDash \phi_1 \text{ and } \sigma \vDash \phi_2.$$

- Temporal operators are interpreted as follows.

$$\sigma \vDash \Box \phi \quad \text{if and only if} \quad \forall k \cdot 0 \leq k < l(\sigma), \sigma^{(k)} \vDash \phi$$

$$\sigma \vDash \Diamond \phi \quad \text{if and only if} \quad \exists k \cdot 0 \leq k < l(\sigma), \sigma^{(k)} \vDash \phi$$

$$\sigma \vDash \phi_1 \,\mathcal{U}\, \phi_2 \quad \text{if and only if} \quad \exists k \cdot 0 \leq k < l(\sigma) \text{ such that } \sigma^{(k)} \vDash \phi_2 \text{ and } \forall i \cdot 0 \leq i < k, \sigma^{(i)} \vDash \phi_1$$

Interpretation

Intuition

For the simple cases, when ϕ, ϕ_1, ϕ_2 are state predicates.

σ satisfies $\Box\,\phi$	if and only if	all states in σ satisfy ϕ
σ satisfies $\Diamond\,\phi$	if and only if	some states in σ satisfy ϕ
σ satisfies $\phi_1\,\mathcal{U}\,\phi_2$	if and only if	some state s_k satisfies ϕ_2 and all the states until s_k (excluding s_k) satisfy ϕ_2

$$\sigma_{Counter} : \quad \langle 0\rangle\,,\langle 1\rangle\,,\langle 2\rangle\,,\langle 3\rangle\,,\langle 4\rangle\,,\langle 5\rangle\,,\langle 4\rangle\,,\langle 3\rangle\,,\langle 4\rangle\,,\langle 5\rangle\,,\ldots$$

- $\sigma_{Counter} \models \Box\, c \in 0\,..\,5$
- $\sigma_{Counter} \models \Box\, c \leq 6$
- $\sigma_{Counter} \models \Diamond\, c = 2$
- $\sigma_{Counter} \not\models \Box\Diamond\, c = 2$
- $\sigma_{Counter} \models c \leq 2\;\mathcal{U}\; c = 2$
- $\sigma_{Counter} \not\models \Box(c \leq 2\;\mathcal{U}\; c = 2)$
- $\sigma_{Counter} \models (\Box\, c \in 0..5) \land (\Diamond\, c = 2)$

ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

Safety v.s. Liveness

- Safety properties: *something (bad) will never happen*.
    - Example: invariance properties.
    - Typically expressed by a temporal formula: $\Box \phi$ or $\phi_1 \Rightarrow \Box \phi_2$.
- Liveness properties: *something (good) will happen*.
    - Example: termination, responsiveness.
    - Typically expressed by a temporal formula:
      $\Diamond \phi$ or $\Box(\phi_1 \Rightarrow \Diamond \phi_2)$.
    - Extended: $\phi_1 \, \mathcal{U} \, \phi_2$ or $\Box(\phi_1 \Rightarrow \phi_2 \, \mathcal{U} \, \phi_3)$.

System Properties

- A system S satisfying property ϕ if all its traces satisfy ϕ.

$$S \models \phi \quad \text{if and only if} \quad \sigma \models \phi, \text{ for all } \sigma \in \mathcal{T}(S).$$

- $S \vdash \phi$ states that $S \models \phi$ is provable.

Proof Tools (1 of 3)

System Leads from ϕ_1 to ϕ_2

Event leads from ϕ_1 to ϕ_2

- Let evt be an event of the form $G(\overline{v}) \longrightarrow \overline{v} := \overline{f}(\overline{v})$
- Let $\phi_1(\overline{v})$ and $\phi_2(\overline{v})$ be two state formulas.
- Event evt leads from $\phi_1(\overline{v})$ to $\phi_2(\overline{v})$ if

$$\phi_1(\overline{v}) \land G(\overline{v}) \;\Rightarrow\; \phi_2(\overline{f}(\overline{v}))$$

System leads from ϕ_1 to ϕ_2

A system S leads from ϕ_1 to ϕ_2 if
every event evt in S leads from ϕ_1 to ϕ_2

- When S leads from ϕ_1 to ϕ_2 is provable, we write

$$\vdash S \text{ leads from } \phi_1 \text{ to } \phi_2$$

Invariance Rules (1/2)

$$\frac{\begin{array}{l}\vdash init(\overline{v}) \;\Rightarrow\; \phi \\ \vdash S \text{ leads from } \phi \text{ to } \phi\end{array}}{S \vdash \Box\phi}\quad \mathbf{INV_{induct}}$$

$Counter \vdash \Box c \in 0\,..\,5$

system *Counter*	**events**
variables $c \in \mathbb{Z}$	inc $\mathrel{\widehat{=}}$ $c \neq 5 \longrightarrow c := c + 1$
initially $c = 0$	dec $\mathrel{\widehat{=}}$ $c > 3 \longrightarrow c := c - 1$

- Initialisation: $\vdash c = 0 \;\Rightarrow\; c \in 0\,..\,5$
- inc: $c \in 0\,..\,5 \land c \neq 5 \;\Rightarrow\; c + 1 \in 0\,..\,5$
- dec: $c \in 0\,..\,5 \land c > 3 \;\Rightarrow\; c - 1 \in 0\,..\,5$

ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

Invariance Rules (1/2)

$$\frac{\vdash \mathit{init}(\overline{v}) \;\Rightarrow\; \phi \qquad \vdash S \text{ leads from } \phi \text{ to } \phi}{S \vdash \Box\phi} \quad \textbf{INV}_{\textbf{induct}}$$

$\mathit{Counter} \vdash \Box c \in 0\,..\,5$

system *Counter*	**events**
variables $c \in \mathbb{Z}$	inc $\hat{=}$ $c \neq 5 \longrightarrow c := c + 1$
initially $c = 0$	dec $\hat{=}$ $c > 3 \longrightarrow c := c - 1$

- Initialisation: $\vdash c = 0 \;\Rightarrow\; c \in 0\,..\,5$
- inc: $c \in 0\,..\,5 \land c \neq 5 \;\Rightarrow\; c + 1 \in 0\,..\,5$
- dec: $c \in 0\,..\,5 \land c > 3 \;\Rightarrow\; c - 1 \in 0\,..\,5$

ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

Invariance Rules (1/2)

$$\frac{\vdash init(\overline{v}) \Rightarrow \phi \qquad \vdash S \text{ leads from } \phi \text{ to } \phi}{S \vdash \Box \phi}\ \textbf{INV}_{\textbf{induct}}$$

$Counter \vdash \Box c \in 0\,..\,5$

system *Counter*	**events**
variables $c \in \mathbb{Z}$	inc $\widehat{=}$ $c \neq 5 \longrightarrow c := c + 1$
initially $c = 0$	dec $\widehat{=}$ $c > 3 \longrightarrow c := c - 1$

- Initialisation: $\vdash c = 0 \Rightarrow c \in 0\,..\,5$
- inc: $c \in 0\,..\,5 \wedge c \neq 5 \Rightarrow c + 1 \in 0\,..\,5$
- dec: $c \in 0\,..\,5 \wedge c > 3 \Rightarrow c - 1 \in 0\,..\,5$

Invariance Rules (2/2)

$$\frac{\begin{array}{rl} & \vdash \ \phi_2 \Rightarrow \phi_1 \\ S & \vdash \ \Box\,\phi_2 \end{array}}{S \ \vdash \ \Box\,\phi_1} \quad \mathbf{INV_{theorem}}$$

$Counter \ \vdash \ \Box\, c \leq 6$

Choose ϕ_2 to be $c \in 0 \,..\, 5$.

- $\vdash c \in 0 \,..\, 5 \ \Rightarrow \ c \leq 6$
- $Counter \ \vdash \ \Box\, c \in 0 \,..\, 5$

Invariance Rules (2/2)

$$\dfrac{\begin{array}{l} \vdash \ \phi_2 \Rightarrow \phi_1 \\ S \ \vdash \ \Box \phi_2 \end{array}}{S \ \vdash \ \Box \phi_1} \quad \textbf{INV}_{\textbf{theorem}}$$

Counter $\vdash \Box c \leq 6$

Choose ϕ_2 to be $c \in 0 \mathrel{..} 5$.

- $\vdash c \in 0 \mathrel{..} 5 \ \Rightarrow \ c \leq 6$
- *Counter* $\vdash \Box c \in 0 \mathrel{..} 5$

Invariance Rules (2/2)

$$\frac{\begin{array}{rl} & \vdash \phi_2 \Rightarrow \phi_1 \\ S & \vdash \Box\phi_2 \end{array}}{S \vdash \Box\phi_1} \quad \textbf{INV}_{\textbf{theorem}}$$

$Counter \vdash \Box c \leq 6$

Choose ϕ_2 to be $c \in 0..5$.

- $\vdash c \in 0..5 \Rightarrow c \leq 6$
- $Counter \vdash \Box c \in 0..5$

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

Proof Tools (2 of 3)

Convergence

- Let ϕ be a state formula.
- A trace is said to be convergent when ϕ holds if it does not end with an infinite sequences of states satisfying ϕ.
- System S is said to be convergent when ϕ holds if all its traces are convergent when ϕ holds.
- When the above fact is provable, we denote it as

$$\vdash S \downarrow \phi$$

Technique

- For system S with events $\mathsf{evt_i} \mathrel{\widehat{=}} G_i(\overline{v}) \longrightarrow \overline{v} := \overline{f}_i(\overline{v})$
- Give a integer variant $V(\overline{v})$
- S converges when ϕ holds if for all events $\mathsf{evt_i}$ of S
  - $\phi(\overline{v}) \wedge G_i(\overline{v}) \Rightarrow V(\overline{v}) \in \mathbb{N}$
  - $\phi(\overline{v}) \wedge G_i(\overline{v}) \;\Rightarrow\; V(\overline{f}_i(\overline{v})) < V(\overline{v})$

Proof Tools (3 of 3)

Deadlock-freeness

- Let ϕ be a state formula.
- System S is deadlock-free when ϕ holds if
  there exists an enabled event of S when ϕ holds.
- When the above fact is provable, we denote it as

$$\vdash S \text{ is deadlock-free when } \phi \text{ holds}$$

- This is guaranteed by proving the following.

$$\phi(\overline{v}) \;\Rightarrow\; G_1(\overline{v}) \lor \ldots \lor G_n(\overline{v})$$

Liveness Rules (1/3)

Always Eventually

$$\frac{\vdash S \downarrow \neg\phi \qquad \vdash S \text{ is deadlock-free when } \neg\phi \text{ holds}}{S \vdash \Box\Diamond\phi} \; \textbf{LIVE}_{\Box\Diamond}$$

Counter $\vdash \Box\Diamond c \geq 2$

- Convergence: Using variant $V \mathrel{\widehat{=}} 5 - c$.
  - $5 - c \in \mathbb{N}$ (using invariant $c \in 0 \,..\, 5$)
  - inc: $\neg c \geq 2 \land c \neq 5 \Rightarrow 5 - (c + 1) < 5 - c$
  - dec: $\neg c \geq 2 \land c > 3 \Rightarrow 5 - (c - 1) < 5 - c$
- Deadlock-free: $\neg c \geq 2 \Rightarrow c \neq 5 \lor c > 3$

ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

Liveness Rules (1/3)

Always Eventually

$$\frac{\vdash S \downarrow \neg\phi \qquad \vdash S \text{ is deadlock-free when } \neg\phi \text{ holds}}{S \vdash \Box\Diamond\phi} \quad \textbf{LIVE}_{\Box\Diamond}$$

Counter $\vdash \Box\Diamond c \geq 2$

- Convergence: Using variant $V \mathrel{\widehat{=}} 5 - c$.
  - $5 - c \in \mathbb{N}$ (using invariant $c \in 0 \,..\, 5$)
  - inc: $\neg c \geq 2 \wedge c \neq 5 \Rightarrow 5 - (c + 1) < 5 - c$
  - dec: $\neg c \geq 2 \wedge c > 3 \Rightarrow 5 - (c - 1) < 5 - c$
- Deadlock-free: $\neg c \geq 2 \Rightarrow c \neq 5 \vee c > 3$

ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

Liveness Rules (1/3)

Always Eventually

$\vdash S \downarrow \neg\phi$

$\vdash S$ is deadlock-free when $\neg\phi$ holds

$$\frac{\vdash S \downarrow \neg\phi \qquad \vdash S \text{ is deadlock-free when } \neg\phi \text{ holds}}{S \vdash \Box\Diamond\phi} \quad \textbf{LIVE}_{\Box\Diamond}$$

Counter $\vdash \Box\Diamond c \geq 2$

- Convergence: Using variant $V \mathrel{\widehat{=}} 5 - c$.
  - $5 - c \in \mathbb{N}$ (using invariant $c \in 0\,..\,5$)
  - inc: $\neg c \geq 2 \land c \neq 5 \Rightarrow 5 - (c + 1) < 5 - c$
  - dec: $\neg c \geq 2 \land c > 3 \Rightarrow 5 - (c - 1) < 5 - c$
- Deadlock-free: $\neg c \geq 2 \Rightarrow c \neq 5 \lor c > 3$

Liveness Rules (2/3)

Until

$$\frac{\begin{array}{rl} & \vdash\ S \text{ leads from } \phi_1 \wedge \neg\phi_2 \text{ to } \phi_1 \vee \phi_2 \\ S & \vdash\ \Box\Diamond(\neg\phi_1 \vee \phi_2)\end{array}}{S \vdash \Box(\phi_1 \Rightarrow \phi_1\,\mathcal{U}\,\phi_2)}\quad \textbf{LIVE}_{\mathcal{U}}$$

$\mathit{Counter} \vdash \Box(c < 2 \;\Rightarrow\; (c < 2\ \mathcal{U}\ c = 2))$

- *Counter* leads from $c < 2 \wedge \neg c = 2$ to $c < 2 \vee c = 2$, equivalently *Counter* leads from $c < 2$ to $c \leq 2$
  - inc: $c < 2 \wedge c \neq 5 \;\Rightarrow\; c + 1 \leq 2$
  - dec: $c < 2 \wedge c > 3 \;\Rightarrow\; c - 1 \leq 2$
- Eventually: $\Box\Diamond(\neg c < 2 \vee c = 2)$, equivalent to $\Box\Diamond\, c \geq 2$

Liveness Rules (2/3)

Until

$$\frac{\begin{array}{l} \vdash\ S \text{ leads from } \phi_1 \wedge \neg\phi_2 \text{ to } \phi_1 \vee \phi_2 \\ S\ \vdash\ \Box\Diamond(\neg\phi_1 \vee \phi_2) \end{array}}{S\ \vdash\ \Box(\phi_1 \Rightarrow \phi_1\,\mathcal{U}\,\phi_2)}\quad \textbf{LIVE}_{\mathcal{U}}$$

$Counter\ \vdash\ \Box(c < 2\ \Rightarrow\ (c < 2\ \mathcal{U}\ c = 2))$

- *Counter* leads from $c < 2 \wedge \neg c = 2$ to $c < 2 \vee c = 2$, equivalently *Counter* leads from $c < 2$ to $c \leq 2$
  - inc: $c < 2 \wedge c \neq 5\ \Rightarrow\ c + 1 \leq 2$
  - dec: $c < 2 \wedge c > 3\ \Rightarrow\ c - 1 \leq 2$
- Eventually: $\Box\Diamond(\neg c < 2 \vee c = 2)$, equivalent to $\Box\Diamond\, c \geq 2$

Liveness Rules (2/3)

Until

$$\frac{\begin{array}{l} \vdash\ S \text{ leads from } \phi_1 \wedge \neg\phi_2 \text{ to } \phi_1 \vee \phi_2 \\ S\ \vdash\ \Box\Diamond(\neg\phi_1 \vee \phi_2) \end{array}}{S\ \vdash\ \Box(\phi_1 \Rightarrow \phi_1\,\mathcal{U}\,\phi_2)}\quad \textbf{LIVE}_{\mathcal{U}}$$

$Counter\ \vdash\ \Box(c < 2\ \Rightarrow\ (c < 2\ \mathcal{U}\ c = 2))$

- *Counter* leads from $c < 2 \wedge \neg c = 2$ to $c < 2 \vee c = 2$, equivalently *Counter* leads from $c < 2$ to $c \leq 2$
  - inc: $c < 2 \wedge c \neq 5\ \Rightarrow\ c + 1 \leq 2$
  - dec: $c < 2 \wedge c > 3\ \Rightarrow\ c - 1 \leq 2$
- Eventually: $\Box\Diamond(\neg c < 2 \vee c = 2)$, equivalent to $\Box\Diamond\, c \geq 2$

Liveness Rules (3/3)

Response

$$\frac{\begin{array}{l} S \vdash \Box(\phi_1 \Rightarrow \phi_3) \\ S \vdash \Box(\phi_3 \Rightarrow (\phi_3\, \mathcal{U}\, \phi_2)) \end{array}}{S \vdash \Box(\phi_1 \Rightarrow \Diamond \phi_2)} \quad \textbf{LIVE}_{\textbf{response}}$$

$Counter \vdash \Box(c = 0 \Rightarrow \Diamond c = 2)$

Choose $\phi_3 \mathrel{\widehat{=}} c < 2$

- $\Box(c = 0 \Rightarrow c < 2)$
- $\Box(c < 2 \Rightarrow (c < 2\ \mathcal{U}\ c = 2))$

Liveness Rules (3/3)

Response

$$\frac{S \vdash \Box(\phi_1 \Rightarrow \phi_3) \qquad S \vdash \Box(\phi_3 \Rightarrow (\phi_3 \,\mathcal{U}\, \phi_2))}{S \vdash \Box(\phi_1 \Rightarrow \Diamond \phi_2)} \quad \textbf{LIVE}_{\textbf{response}}$$

$Counter \vdash \Box(c = 0 \Rightarrow \Diamond c = 2)$

Choose $\phi_3 \mathrel{\widehat{=}} c < 2$

- $\Box(c = 0 \Rightarrow c < 2)$
- $\Box(c < 2 \Rightarrow (c < 2 \,\mathcal{U}\, c = 2))$

ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

Liveness Rules (3/3)

Response

$$\frac{\begin{array}{l} S \vdash \Box(\phi_1 \Rightarrow \phi_3) \\ S \vdash \Box(\phi_3 \Rightarrow (\phi_3 \,\mathcal{U}\, \phi_2)) \end{array}}{S \vdash \Box(\phi_1 \Rightarrow \Diamond \phi_2)} \quad \textbf{LIVE}_{\textbf{response}}$$

$Counter \vdash \Box(c = 0 \Rightarrow \Diamond c = 2)$

Choose $\phi_3 \mathrel{\widehat{=}} c < 2$

- $\Box(c = 0 \Rightarrow c < 2)$
- $\Box(c < 2 \Rightarrow (c < 2 \,\mathcal{U}\, c = 2))$

Example. Reader and Writer

system *RdWr*	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\widehat{=}$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \wedge w = 0$	write $\widehat{=}$ $w < r + 3 \longrightarrow w := w + 1$

An execution

$\langle 0, 0\rangle \xrightarrow{write} \langle 0, 1\rangle \xrightarrow{write} \langle 0, 2\rangle \xrightarrow{write} \langle 0, 3\rangle \xrightarrow{read} \langle 1, 3\rangle \xrightarrow{read} \langle 2, 3\rangle \xrightarrow{read} \langle 3, 3\rangle$
$\xrightarrow{write} \langle 3, 4\rangle \xrightarrow{write} \langle 3, 5\rangle \xrightarrow{read} \langle 4, 5\rangle \xrightarrow{write} \langle 4, 6\rangle \xrightarrow{read} \langle 5, 6\rangle \xrightarrow{write} \langle 5, 7\rangle \ldots$

Example. Reader and Writer

system *RdWr*	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\hat{=}$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \land w = 0$	write $\hat{=}$ $w < r + 3 \longrightarrow w := w + 1$

An execution

$$\langle 0, 0\rangle \xrightarrow{write} \langle 0, 1\rangle \xrightarrow{write} \langle 0, 2\rangle \xrightarrow{write} \langle 0, 3\rangle \xrightarrow{read} \langle 1, 3\rangle \xrightarrow{read} \langle 2, 3\rangle \xrightarrow{read} \langle 3, 3\rangle$$

$$\xrightarrow{write} \langle 3, 4\rangle \xrightarrow{write} \langle 3, 5\rangle \xrightarrow{read} \langle 4, 5\rangle \xrightarrow{write} \langle 4, 6\rangle \xrightarrow{read} \langle 5, 6\rangle \xrightarrow{write} \langle 5, 7\rangle \ldots$$

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

Example. Reader and Writer

system *RdWr*	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\;\widehat{=}\;$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \wedge w = 0$	write $\;\widehat{=}\;$ $w < r + 3 \longrightarrow w := w + 1$

An execution

$$\langle 0,0\rangle \xrightarrow{write} \langle 0,1\rangle \xrightarrow{write} \langle 0,2\rangle \xrightarrow{write} \langle 0,3\rangle \xrightarrow{read} \langle 1,3\rangle \xrightarrow{read} \langle 2,3\rangle \xrightarrow{read} \langle 3,3\rangle$$
$$\xrightarrow{write} \langle 3,4\rangle \xrightarrow{write} \langle 3,5\rangle \xrightarrow{read} \langle 4,5\rangle \xrightarrow{write} \langle 4,6\rangle \xrightarrow{read} \langle 5,6\rangle \xrightarrow{write} \langle 5,7\rangle \ldots$$

A Progress Properties

Reader's progress

The Reader will eventually read the data that the Writer wrote.

Formalisation. First attempt

- Can we prove $RdWr \models \Box\Diamond\, r = w$?
  - No, the Reader might always be behind the Writer.

Formalisation. Second attempt

$$RdWr \models \Box(w = K \Rightarrow \Diamond\, r = K)\ ?$$

A Progress Properties

Reader's progress

The Reader will eventually read the data that the Writer wrote.

Formalisation. First attempt

- Can we prove $RdWr \models \Box \Diamond r = w$?
  - No, the Reader might always be behind the Writer (despite progressing).

Formalisation. Second attempt

$RdWr \models \Box(w = K \Rightarrow \Diamond r = K)$?

A Progress Properties

Reader's progress

The Reader will eventually read the data that the Writer wrote.

Formalisation. First attempt

- Can we prove $RdWr \models \Box \Diamond r = w$?
  - No, the Reader might always be behind the Writer (despite progressing).

Formalisation. Second attempt

$RdWr \models \Box(w = K \Rightarrow \Diamond r = K)$?

A Progress Properties

Reader's progress

The Reader will eventually read the data that the Writer wrote.

Formalisation. First attempt

- Can we prove $RdWr \models \Box\Diamond\, r = w$?
  - No, the Reader might always be behind the Writer (despite progressing).

Formalisation. Second attempt

$$RdWr \models \Box(w = K \Rightarrow \Diamond\, r = K) \;?$$

A Proof (1/6)

system *RdWr*	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\widehat{=}$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \wedge w = 0$	write $\widehat{=}$ $w < r + 3 \longrightarrow w := w + 1$

$$RdWr \vdash \Box(w = K \Rightarrow \Diamond r = K)$$

$\xrightarrow{\textbf{LIVE}_{\textbf{response}}}$

$$\begin{cases} RdWr \vdash \Box(w = K \Rightarrow \phi_3) & (1) \\ RdWr \vdash \Box(\phi_3 \Rightarrow (\phi_3 \,\mathcal{U}\, r = K)) & (2) \end{cases}$$

$$\frac{S \vdash \Box(\phi_1 \Rightarrow \phi_3) \qquad S \vdash \Box(\phi_3 \Rightarrow (\phi_3 \,\mathcal{U}\, \phi_2))}{S \vdash \Box(\phi_1 \Rightarrow \Diamond \phi_2)} \ \textbf{LIVE}_{\textbf{response}}$$

A Proof (1/6)

system *RdWr*	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\widehat{=}$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \wedge w = 0$	write $\widehat{=}$ $w < r + 3 \longrightarrow w := w + 1$

$$RdWr \vdash \square (w = K \Rightarrow \diamond r = K)$$

$\xrightarrow{\textbf{LIVE}_{\textbf{response}}}$

$$RdWr \vdash \square (w = K \Rightarrow r \leq K) \quad (1)$$

$$RdWr \vdash \square (r \leq K \Rightarrow (r \leq K \,\mathcal{U}\, r = K)) \quad (2)$$

$$\frac{S \vdash \square(\phi_1 \Rightarrow \phi_3) \qquad S \vdash \square(\phi_3 \Rightarrow (\phi_3 \,\mathcal{U}\, \phi_2))}{S \vdash \square(\phi_1 \Rightarrow \diamond \phi_2)} \ \textbf{LIVE}_{\textbf{response}}$$

A Proof (2/6)

system *RdWr*	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\widehat{=}$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \wedge w = 0$	write $\widehat{=}$ $w < r + 3 \longrightarrow w := w + 1$

$$\frac{\vdash \overline{v} = \overline{s}_0 \;\Rightarrow\; \phi \qquad \vdash S \text{ leads from } \phi \text{ to } \phi}{S \;\vdash\; \Box\phi} \; \textbf{INV}_{\textbf{induct}}$$

$$\frac{\vdash \; \phi_2 \Rightarrow \phi_1 \qquad S \;\vdash\; \Box\phi_2}{S \;\vdash\; \Box\phi_1} \; \textbf{INV}_{\textbf{theorem}}$$

(1) $RdWr \vdash \Box(w = K \;\Rightarrow\; r \leq K)$

$\xrightarrow{\textbf{INV}_{\textbf{theorem}}}$ $\begin{cases} r \leq w \Rightarrow (w = K \Rightarrow r \leq K) \\ RdWr \;\vdash\; \Box r \leq w \end{cases}$

- $\textbf{INV}_{\textbf{induct}}$ fails, hence apply $\textbf{INV}_{\textbf{theorem}}$ with ϕ_2 to be $r \leq w$.

ETH Eidgenössische Swiss Federal Instit

A Proof (2/6)

system *RdWr*	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\widehat{=}$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \wedge w = 0$	write $\widehat{=}$ $w < r + 3 \longrightarrow w := w + 1$

$$\frac{\vdash \overline{v} = \overline{s}_0 \;\Rightarrow\; \phi \qquad \vdash S \text{ leads from } \phi \text{ to } \phi}{S \;\vdash\; \Box\,\phi} \;\; \textbf{INV}_{\textbf{induct}}$$

$$\frac{\vdash \; \phi_2 \Rightarrow \phi_1 \qquad S \;\vdash\; \Box\,\phi_2}{S \;\vdash\; \Box\,\phi_1} \;\; \textbf{INV}_{\textbf{theorem}}$$

(1) $RdWr \vdash \Box(w = K \;\Rightarrow\; r \leq K)$

$\xrightarrow{\textbf{INV}_{\textbf{theorem}}}$ $\begin{cases} r \leq w \Rightarrow (w = K \Rightarrow r \leq K) \\ RdWr \;\vdash\; \Box\, r \leq w \end{cases}$

- $\textbf{INV}_{\textbf{induct}}$ fails, hence apply $\textbf{INV}_{\textbf{theorem}}$ with ϕ_2 to be $r \leq w$.

A Proof (3/6)

$$\frac{\begin{array}{rl} & \vdash S \text{ leads from } \phi_1 \wedge \neg\phi_2 \text{ to } \phi_1 \vee \phi_2 \\ S & \vdash \Box\Diamond(\neg\phi_1 \vee \phi_2)\end{array}}{S \vdash \Box(\phi_1 \Rightarrow \phi_1 \,\mathcal{U}\, \phi_2)} \quad \textbf{LIVE}_{\mathcal{U}}$$

(2) $RdWr \vdash \Box(r \leq K \Rightarrow (r \leq K \;\mathcal{U}\; r = K))$

$\xrightarrow{\textbf{LIVE}_{\mathcal{U}}}$
$\left\{ \begin{array}{ll} \vdash RdWr \text{ leads from } r \leq K \wedge \neg r = K \text{ to } r \leq K \vee r = K & (2.1) \\ RdWr \vdash \Box\Diamond(\neg r \leq K \vee r = K) & (2.2) \end{array} \right.$

A Proof (4/6)

system *RdWr*	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\widehat{=}$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \wedge w = 0$	write $\widehat{=}$ $w < r + 3 \longrightarrow w := w + 1$

(2.1) $\vdash$ *RdWr* leads from $r \leq K \wedge \neg r = K$ to $r \leq K \vee r = K$

$\xrightarrow{\textbf{logic}}$ $\vdash$ *RdWr* leads from $r < K$ to $r \leq K$

$\xrightarrow{\textbf{definition}}$ $\begin{cases} r < K \wedge r \neq w \Rightarrow r + 1 \leq K \\ r < K \wedge w < r + 1 \Rightarrow r \leq K \end{cases}$

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

A Proof (5/6)

$$\frac{\vdash S \downarrow \neg\phi \qquad \vdash S \text{ is deadlock-free when } \neg\phi \text{ holds}}{S \vdash \Box\Diamond\phi}\ \textbf{LIVE}_{\Box\Diamond}$$

(2.2) $\boxed{RdWr \ \vdash \ \Box\Diamond(\neg r \leq K \vee r = K)}$

$\xrightarrow{\textbf{logic}}$ $\boxed{RdWr \ \vdash \ \Box\Diamond\, r \geq K}$

$\xrightarrow{\textbf{LIVE}_{\Box\Diamond}}$ $\begin{cases} \boxed{\vdash RdWr \ \downarrow \ \neg r \geq K} & (2.2.1) \\ \boxed{\vdash RdWr \text{ is deadlock-free when } \neg r \geq K \text{ holds}} & (2.2.2) \end{cases}$

A Proof (6/6)

system $RdWr$	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\hat{=}$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \land w = 0$	write $\hat{=}$ $w < r + 3 \longrightarrow w := w + 1$

$$(2.2.1)\quad \boxed{\vdash RdWr \;\downarrow\; \neg r \geq K} \;\xrightarrow{\textbf{logic}}\; \boxed{\vdash RdWr \;\downarrow\; r < K}$$

- Use variant:

$$(K - r) \times 2 + (r + 3 - w)$$

$$(2.2.2)\quad \boxed{\vdash RdWr \text{ is deadlock-free when } \neg r \geq K \text{ holds}} \;\xrightarrow{\textbf{definition}}\; \boxed{\neg r \geq K \quad \Rightarrow \quad r \neq w \;\lor\; w < 3 + r}$$

ETH
Eidgenössische Tech.
Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

A Proof (6/6)

system *RdWr*	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\widehat{=}$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \wedge w = 0$	write $\widehat{=}$ $w < r + 3 \longrightarrow w := w + 1$

$$\boxed{\vdash RdWr \;\downarrow\; \neg r \geq K} \quad \xrightarrow[\text{logic}]{(2.2.1)} \quad \boxed{\vdash RdWr \;\downarrow\; r < K}$$

- Use variant:

$$(K - r) \times 2 + (r + 3 - w)$$

$$\boxed{\vdash RdWr \text{ is deadlock-free when } \neg r \geq K \text{ holds}} \quad \xrightarrow[\text{definition}]{(2.2.2)} \quad \boxed{\neg r \geq K \;\Rightarrow\; r \neq w \;\vee\; w < 3 + r}$$

A Proof (6/6)

system *RdWr*	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\widehat{=}$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \wedge w = 0$	write $\widehat{=}$ $w < r + 3 \longrightarrow w := w + 1$

$$\boxed{\vdash RdWr \ \downarrow \ \neg r \geq K} \quad \xrightarrow[\text{logic}]{(2.2.1)} \quad \boxed{\vdash RdWr \ \downarrow \ r < K}$$

- Use variant:

$$(K - r) \times 2 + (r + 3 - w)$$

$$\boxed{\vdash RdWr \text{ is deadlock-free when } \neg r \geq K \text{ holds}} \quad \xrightarrow[\text{definition}]{(2.2.2)} \quad \boxed{\neg r \geq K \ \Rightarrow \ r \neq w \ \vee \ w < 3 + r}$$

A Proof (6/6)

system *RdWr*	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\widehat{=}$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \wedge w = 0$	write $\widehat{=}$ $w < r + 3 \longrightarrow w := w + 1$

$$(2.2.1) \quad \boxed{\vdash RdWr \ \downarrow \ \neg r \geq K} \quad \xrightarrow{\textbf{logic}} \quad \boxed{\vdash RdWr \ \downarrow \ r < K}$$

- Use variant:

$$(K - r) \times 2 + (r + 3 - w)$$

$$(2.2.2) \quad \boxed{\vdash RdWr \text{ is deadlock-free when } \neg r \geq K \text{ holds}} \quad \xrightarrow{\textbf{definition}} \quad \boxed{\neg r \geq K \quad \Rightarrow \quad r \neq w \ \vee \ w < 3 + r}$$

A Proof (6/6)

system *RdWr*	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\mathrel{\widehat{=}}$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \land w = 0$	write $\mathrel{\widehat{=}}$ $w < r + 3 \longrightarrow w := w + 1$

$$(2.2.1) \quad \boxed{\vdash RdWr \;\downarrow\; \neg r \geq K} \;\; \xrightarrow{\textbf{logic}} \;\; \boxed{\vdash RdWr \;\downarrow\; r < K}$$

- Use variant:

$$(K - r) \times 2 + (r + 3 - w)$$

$$(2.2.2) \quad \boxed{\vdash RdWr \text{ is deadlock-free when } \neg r \geq K \text{ holds}} \;\; \xrightarrow{\textbf{definition}} \;\; \boxed{\neg r \geq K \quad \Rightarrow \quad r \neq w \;\vee\; w < 3 + r}$$

ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

A Proof (6/6)

system *RdWr*	**events**
variables $r, w \in \mathbb{Z}, \mathbb{Z}$	read $\widehat{=}$ $r \neq w \longrightarrow r := r + 1$
initially $r = 0 \land w = 0$	write $\widehat{=}$ $w < r + 3 \longrightarrow w := w + 1$

$$\boxed{\vdash RdWr \ \downarrow \ \neg r \geq K} \ \overset{\text{(2.2.1)}}{\underset{\textbf{logic}}{\longrightarrow}} \ \boxed{\vdash RdWr \ \downarrow \ r < K}$$

- Use variant:

$$(K - r) \times 2 + (r + 3 - w)$$

$$\boxed{\vdash RdWr \text{ is deadlock-free when } \neg r \geq K \text{ holds}} \ \overset{\text{(2.2.2)}}{\underset{\textbf{definition}}{\longrightarrow}} \ \boxed{\neg r \geq K \quad \Rightarrow \quad r \neq w \ \lor \ w < 3 + r}$$

Summary

- Proof rules for certain classes of invariance and liveness properties.
- The proof rules based on the reasoning about:
  - the system leads from ϕ_1 to ϕ_2
  - the system is convergence when ϕ holds
  - the system is deadlock-free when ϕ holds.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

Further Directions

- Proofs become tedious when the system becomes large.
- Refinement helps to reduce the complexity.
    - Invariance properties are maintained.
    - How about liveness?
- Concurrent systems: fairness assumptions.
    - Expect some weaker rules.
    - Interaction with refinement?

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

INFORMATION SECURITY

For Further Reading I

- Zohar Manna and Amir Pnueli.
  Adequate Proof Principles
  for Invariance and Liveness Properties of Concurrent Programs.
  Science of Computer Programming 4:259-289, 1984.
- Zohar Manna and Amir Pnueli.
  Completing the Temporal Picture.
  Theoretical Computer Science 81(1):97-130, 1991.