Iris

Viktor Vafeiadis

MPI-SWS

EPIT 2018, May 2018

Slides by Ralf Jung (MPI-SWS)

Tons of prior program logics

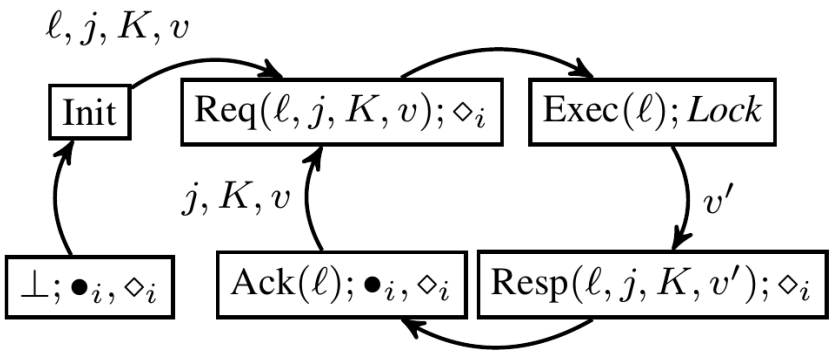
- CSL [O'H07]
- RGSep [VP07]
- SAGL [FFS07]
- LRG [Fen09]
- CAP [DY+10]
- HLRG [Fu+10]
- CaReSL [TDB13]
- SCSL [LWN13]
- HoCAP [SBP13]
- iCAP [SB14]
- FCSL [Nan+14]
- TaDA [dDYG14]

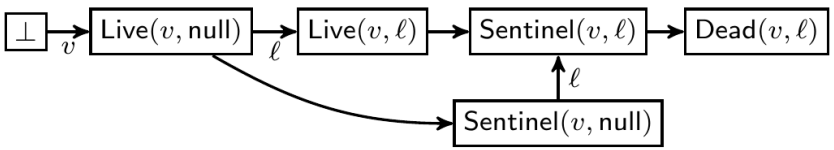
Yet another concurrency logic

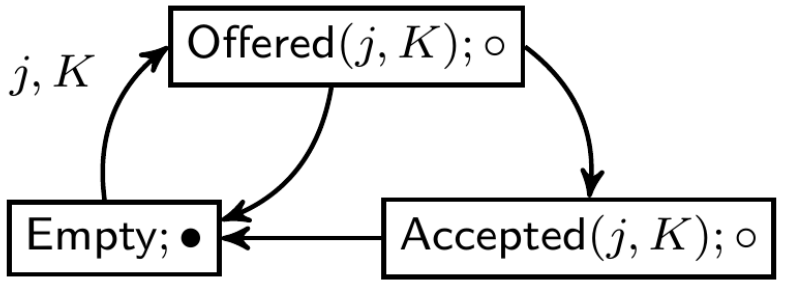
Iris addresses two problems

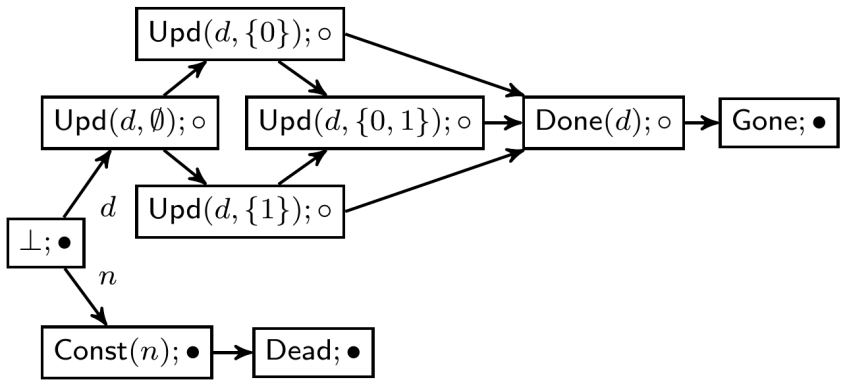
- Simplifying the foundations of concurrent reasoning
- Supporting a *logical* notion of atomicity

STSs in CaReSL









Complex rules built-in as primitives

CaReSL:

$$\frac{\mathcal{C} \vdash \forall b \overset{\text{rely}}{\sqsupseteq}_\pi b_0.\ (\!|\pi[\![b]\!] * P|\!)\ i \Mapsto_{\mathrm{I}} a\ (\!|x.\ \exists b' \overset{\text{guar}}{\sqsupseteq}_\pi b.\ \pi[\![b']\!] * Q|\!)}{\mathcal{C} \vdash \left\{\boxed{b_0}^n_\pi * \triangleright P\right\}\ i \Mapsto a\ \left\{x.\ \exists b'.\ \boxed{b'}^n_\pi * Q\right\}}\ \textsc{UpdIsl}$$

iCAP:

$$\frac{\begin{array}{c}\Gamma, \Delta \mid \Phi \vdash \mathsf{stable}(\mathsf{P}) \qquad \Gamma, \Delta \mid \Phi \vdash \forall y.\ \mathsf{stable}(\mathsf{Q}(y)) \\ \Gamma, \Delta \mid \Phi \vdash n \in C \qquad \Gamma, \Delta \mid \Phi \vdash \forall x \in X.\ (x, f(x)) \in \overline{T(A)} \vee f(x) = x \\ \Gamma \mid \Phi \vdash \forall x \in X.\ (\Delta).\langle \mathsf{P} * \circledast_{\alpha \in A}[\alpha]^{\mathsf{n}}_{g(\alpha)} * \triangleright I(x)\rangle\ c\ \langle \mathsf{Q}(x) * \triangleright I(f(x))\rangle^{C \setminus \{n\}}\end{array}}{\begin{array}{c}\Gamma \mid \Phi \vdash (\Delta).\ \langle \mathsf{P} * \circledast_{\alpha \in A}[\alpha]^{\mathsf{n}}_{g(\alpha)} * \mathsf{region}(X, T, I, n)\rangle \\ c \\ \langle \exists x.\ \mathsf{Q}(x) * \mathsf{region}(\{f(x)\}, T, I, n)\rangle^C\end{array}}\ \textsc{Atomic}$$

TaDA:

$$\frac{\begin{array}{c}\text{Use atomic rule} \\ a \notin \mathcal{A} \quad \forall x \in X.\ (x, f(x)) \in \mathcal{T}_{\mathbf{t}}(\mathrm{G})^* \\ \lambda; \mathcal{A} \vdash \mathbb{W} x \in X.\ \langle p_p \mid I(\mathbf{t}^\lambda_a(x)) * p(x) * [\mathrm{G}]_a\rangle\ \mathbb{C}\ \ \mathbb{E} y \in Y.\ \langle q_p(x,y) \mid I(\mathbf{t}^\lambda_a(f(x))) * q(x,y)\rangle\end{array}}{\lambda + 1; \mathcal{A} \vdash \mathbb{W} x \in X.\ \langle p_p \mid \mathbf{t}^\lambda_a(x) * p(x) * [\mathrm{G}]_a\rangle\ \mathbb{C}\ \ \mathbb{E} y \in Y.\ \langle q_p(x,y) \mid \mathbf{t}^\lambda_a(f(x)) * q(x,y)\rangle}$$

Complex rules built-in as primitives

CaReSL:

$$\frac{\mathcal{C} \vdash \forall b \overset{\text{rely}}{\sqsupseteq_\pi} b_0.\ (\!|\pi[\![b]\!] * P|\!)\ i \Rightarrow\!\!\!\Rightarrow_{\mathrm{I}} a\ (\!|x.\ \exists b' \overset{\text{guar}}{\sqsupseteq_\pi} b.\ \pi[\![b']\!] * Q|\!)}{\mathcal{C} \vdash \left\{ \boxed{b_0}^n_\pi * \triangleright P \right\}\ i \Rightarrow\!\!\!\Rightarrow\ a\ \left\{ x.\ \exists b'.\ \boxed{b'}^n_\pi * Q \right\}}\ \textsc{UpdIsl}$$

All you need are two simple primitives:

- *Monoids* to *express* protocols.
- *Invariants* to *enforce* protocols.

TaDA:

$$\frac{\begin{array}{c}\text{Use atomic rule} \\ a \notin \mathcal{A} \quad \forall x \in X.\ (x, f(x)) \in \mathcal{T}_{\mathbf{t}}(\mathrm{G})^* \\ \lambda; \mathcal{A} \vdash \mathbb{W} x \in X. \left\langle p_p \,\middle|\, I(\mathbf{t}^\lambda_a(x)) * p(x) * [\mathrm{G}]_a \right\rangle\ \mathbb{C}\ \ \exists y \in Y. \left\langle q_p(x,y) \,\middle|\, I(\mathbf{t}^\lambda_a(f(x))) * q(x,y) \right\rangle\end{array}}{\lambda + 1; \mathcal{A} \vdash \mathbb{W} x \in X. \left\langle p_p \,\middle|\, \mathbf{t}^\lambda_a(x) * p(x) * [\mathrm{G}]_a \right\rangle\ \mathbb{C}\ \ \exists y \in Y. \left\langle q_p(x,y) \,\middle|\, \mathbf{t}^\lambda_a(f(x)) * q(x,y) \right\rangle}$$

Protocols = Monoids + Invariants

A simple example

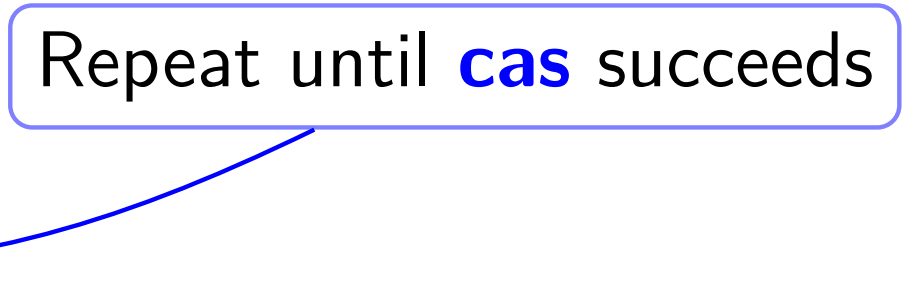
```
fn inc2(x) {
    do {
        v = !x;
        b = cas(x, v, v + 2);
    } while (not b);
}
```

A simple example

```
fn inc2(x) {
    do {
        v = !x;
        b = cas(x, v, v + 2);
    } while (not b);
}
```

A simple example

```
fn inc2(x) {
    do {
        v = !x;
        b = cas(x, v, v + 2);
    } while (not b);
}
```



Invariants

"x points to an even number"

```
fn inc2(x) {
  do {
    v = !x;
    b = cas(x, v, v + 2);
  } while (not b);
}
```

Invariants

"x points to an even number": $R \triangleq \exists i.\ x \mapsto i * \mathsf{ev}(i)$

```
fn inc2(x) {
  do {
    v = !x;
    b = cas(x, v, v + 2);
  } while (not b);
}
```

Invariants

"x points to an even number": $R \triangleq \exists i.\ x \mapsto i * \mathsf{ev}(i)$

```
fn inc2(x) {
  do {
    v = !x;
    b = cas(x, v, v + 2);
  } while (not b);
}
```

Before **cas**:
Obtain invariant

Invariants

"x points to an even number": $R \triangleq \exists i.\ x \mapsto i * \mathsf{ev}(i)$

```
fn inc2(x) {
    do {
        v = !x;
        b = cas(x, v, v + 2);
    } while (not b);
}
```

Before **cas**:
Obtain invariant

After **cas**:
Re-establish invariant

Invariants

"x points to an even number": $R \triangleq \exists i.\ x \mapsto i * \mathsf{ev}(i)$

fn *inc2*(x) {
  do {
    $v = !x;$
    $b = \mathbf{cas}(x, v, v + 2);$
  } **while** (**not** b);
}

$$\frac{\{R * P\}\ e\ \{R * Q\} \qquad e \text{ atomic}}{\boxed{R} \vdash \{P\}\ e\ \{Q\}}$$

cf. CSL [O'H07]

Invariants

"x points to an even number": $R \triangleq \exists i.\ x \mapsto i * \mathsf{ev}(i)$

fn *inc2*(x) {
  do {
    $v = !x;$
    $b = \mathbf{cas}(x, v, v + 2);$
  } **while** (**not** b);
}

$$\frac{\{R * P\}\ e\ \{R * Q\} \qquad e \text{ atomic}}{\boxed{R} \vdash \{P\}\ e\ \{Q\}}$$

Invariant

cf. CSL [O'H07]

Invariants

"x points to an even number": $R \triangleq \exists i.\ x \mapsto i * \mathsf{ev}(i)$

fn *inc2*(x) {
  do {
    $v = !x;$
    $b = \mathbf{cas}(x, v, v + 2);$
  } **while** (**not** b);
}

$$\frac{\{R * P\}\ e\ \{R * Q\} \qquad e \text{ atomic}}{\boxed{R} \vdash \{P\}\ e\ \{Q\}}$$

cf. CSL [O'H07]

Resources carried in

Invariants

"x points to an even number": $R \triangleq \exists i.\ x \mapsto i * \mathsf{ev}(i)$

```
fn inc2(x) {
    do {
        v = !x;
        b = cas(x, v, v + 2);
    } while (not b);
}
```

$$\frac{\{R * P\}\ e\ \{R * Q\} \qquad e \text{ atomic}}{\boxed{R} \vdash \{P\}\ e\ \{Q\}}$$

cf. CSL [O'H07]

Resources carried out

Invariants

"x points to an even number": $R \triangleq \exists i.\ x \mapsto i * \mathsf{ev}(i)$

fn *inc2*(x) {
  do {
    $v = !x;$
    $b = \mathbf{cas}(x, v, v + 2);$
  } **while** (**not** b);
}

$$\frac{\{R * P\}\ e\ \{R * Q\} \quad e \text{ atomic}}{\boxed{R} \vdash \{P\}\ e\ \{Q\}}$$

cf. CSL [O'H07]

Avoid interference

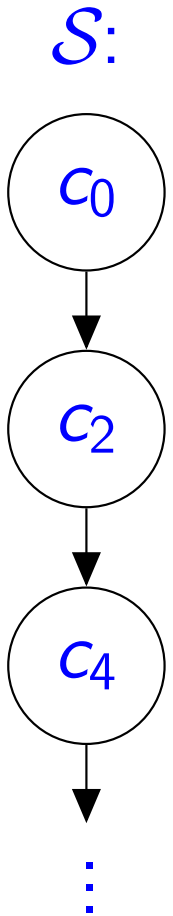
Invariants are not enough

"x points to a *monotonically increasing* even number"

```
fn inc2(x) {
  do {
    v = !x;
    b = cas(x, v, v + 2);
  } while (not b);
}
```

STS Example

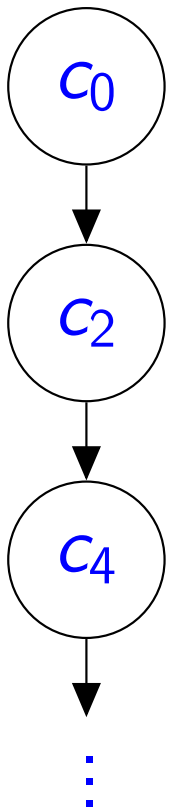
```
fn inc2(x) {
    do {
        v = !x;
        b = cas(x, v, v + 2);
    } while (not b);
}
```



STS Example

$\varphi(c_i) \triangleq x \mapsto i$

$\mathcal{S}$:




```
fn inc2(x) {
    do {
        v = !x;
        b = cas(x, v, v + 2);
    } while (not b);
}
```

STS Example

$\{\geq c_i\}$

fn *inc2*(x) {

  do {

    $v = !x;$

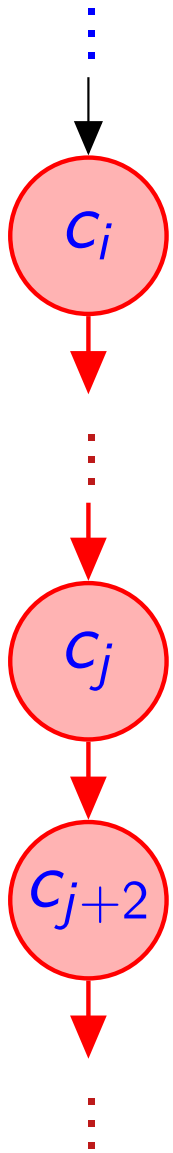
    $b = \mathbf{cas}(x, v, v + 2);$

  } **while** (**not** b);

}

$\{\geq c_{i+2}\}$

$\varphi(c_i) \triangleq x \mapsto i$



STS Example

$\{\geq c_i\}$

fn *inc2*(x) {

  do {

    $v = !x;$

    $b = \mathbf{cas}(x, v, v + 2);$

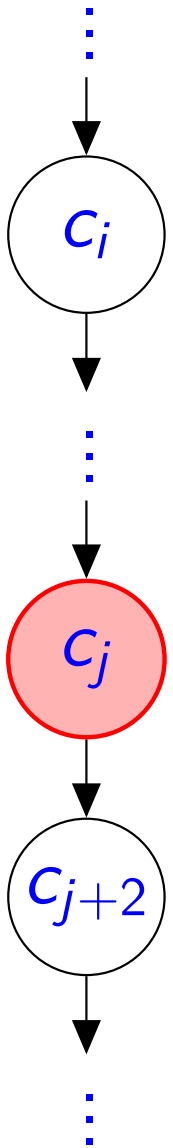
  } **while** (**not** b);

}

$\{\geq c_{i+2}\}$

$\varphi(c_i) \triangleq x \mapsto i$

Current state: c_j



STS Example

$\{\geq c_i\}$

fn *inc2*(x) {

  do {

    $v = !x;$

    $b = \mathbf{cas}(x, v, v + 2);$

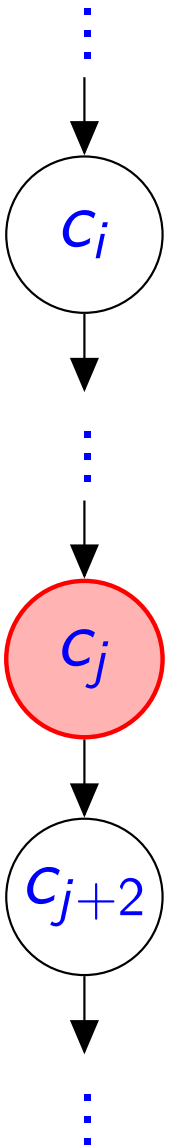
  } **while** (**not** b);

}

$\{\geq c_{i+2}\}$

$\varphi(c_i) \triangleq x \mapsto i$

Current state: c_j, so $x \mapsto j$



STS Example

$\{\geq c_i\}$

fn *inc2*(x) {

  do {

    $v = !x;$

    $b = \mathbf{cas}(x, v, v + 2);$

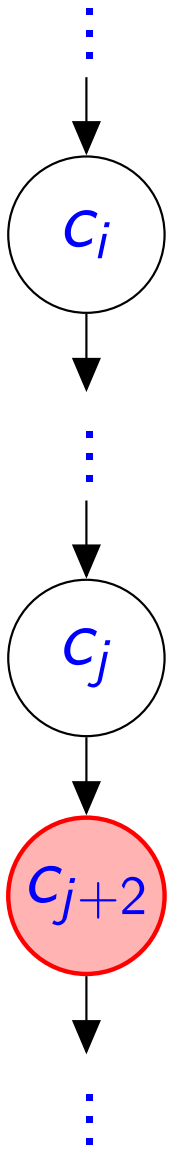
  } **while** (**not** b);

}

$\{\geq c_{i+2}\}$

$\varphi(c_i) \triangleq x \mapsto i$

Update state to c_{j+2}



STS Example

$\{\geq c_i\}$

fn *inc2*(x) {

  do {

    $v = !x;$

    $b = \mathbf{cas}(x, v, v + 2);$

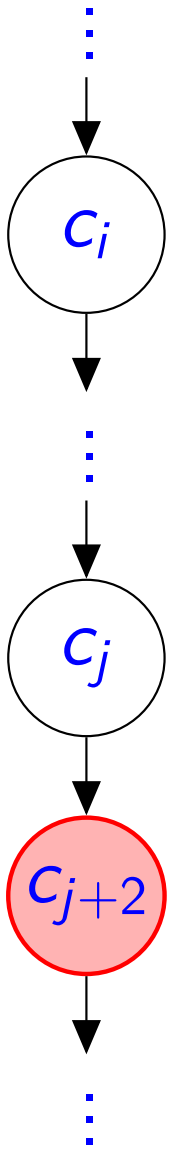
  } **while** (**not** b);

}

$\{\geq c_{i+2}\}$

$\varphi(c_i) \triangleq x \mapsto i$

Update state to c_{j+2}, show: $x \mapsto j + 2$



STS Example

$\{\geq c_i\}$

fn *inc2*(x) {

  do {

    $v = !x;$

    $b = \mathbf{cas}(x, v, v + 2);$

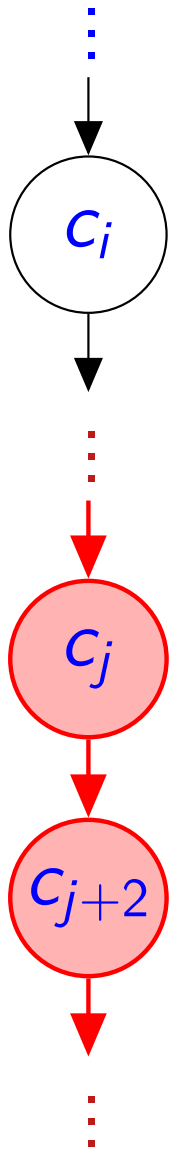
  } **while** (**not** b);

}

$\{\geq c_{i+2}\}$

$\varphi(c_i) \triangleq x \mapsto i$

Obtain $\geq c_{j+2}$



STS Example

$\{\geq c_i\}$

fn *inc2*(x) {

  do {

    $v = !x;$

    $b = \mathbf{cas}(x, v, v + 2);$

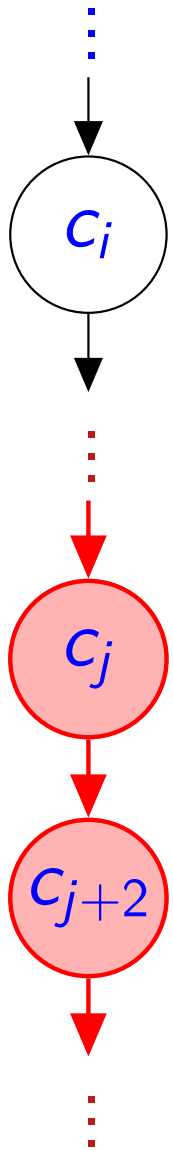
  } **while** (**not** b);

}

$\{\geq c_{i+2}\}$

$\varphi(c_i) \triangleq x \mapsto i$

Obtain $\geq c_{j+2}$



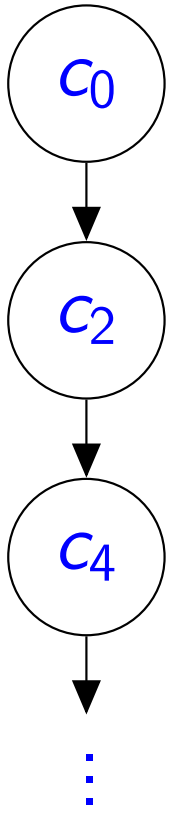
STS Example

$$\frac{\forall c.\ \{\hat{c} \to^* c * \varphi(c) * P\}\ e\ \{v.\ \exists c'.\ c \to^* c' * \varphi(c') * Q\}}{\mathsf{STS}(\mathcal{S}, \varphi) \vdash \{\boxed{\geq \hat{c}} * P\}\ e\ \{v.\ \exists c'.\ \boxed{\geq c'} * Q\}}$$

$\{\boxed{\geq c_i}\}$

$\varphi(c_i) \triangleq x \mapsto i$

$\mathcal{S}$:



fn *inc2*(x) {
  do {
    $v = !x;$
    $b = \mathbf{cas}(x, v, v + 2);$
  } **while** (**not** b);
}

$\{\boxed{\geq c_{i+2}}\}$

Complex rules built-in as primitives

CaReSL:

$$\frac{\mathcal{C} \vdash \forall b \overset{\text{rely}}{\sqsupseteq_\pi} b_0.\ (\!|\pi[\![b]\!] * P|\!)\ i \Rightarrow\!\!\!\Rightarrow_{\mathrm{I}} a\ (\!|x.\ \exists b' \overset{\text{guar}}{\sqsupseteq_\pi} b.\ \pi[\![b']\!] * Q|\!)}{\mathcal{C} \vdash \left\{ \boxed{b_0}^n_\pi * \triangleright P \right\}\ i \Rightarrow\!\!\!\Rightarrow\ a\ \left\{ x.\ \exists b'.\ \boxed{b'}^n_\pi * Q \right\}}\ \textsc{UpdIsl}$$

All you need are two simple primitives:

- Invariants
- Partial commutative monoids

TaDA:

$$\frac{\begin{array}{c}\text{Use atomic rule} \\ a \notin \mathcal{A} \quad \forall x \in X.\ (x, f(x)) \in \mathcal{T}_{\mathbf{t}}(\mathrm{G})^* \\ \lambda; \mathcal{A} \vdash \mathbb{W} x \in X. \left\langle p_p \,\middle|\, I(\mathbf{t}^\lambda_a(x)) * p(x) * [\mathrm{G}]_a \right\rangle\ \mathbb{C}\ \ \exists y \in Y. \left\langle q_p(x,y) \,\middle|\, I(\mathbf{t}^\lambda_a(f(x))) * q(x,y) \right\rangle\end{array}}{\lambda + 1; \mathcal{A} \vdash \mathbb{W} x \in X. \left\langle p_p \,\middle|\, \mathbf{t}^\lambda_a(x) * p(x) * [\mathrm{G}]_a \right\rangle\ \mathbb{C}\ \ \exists y \in Y. \left\langle q_p(x,y) \,\middle|\, \mathbf{t}^\lambda_a(f(x)) * q(x,y) \right\rangle}$$

Logical ("ghost") resources

Partial commutative monoid (PCM)

- Set M (carrier)
- An operation $\cdot$ on M (associative, commutative)
- A unit ε ("empty")
- A zero $\bot$ ("bottom", "undefined")

Logical ("ghost") resources

Partial commutative monoid (PCM)

- Set M (carrier)
- An operation $\cdot$ on M (associative, commutative)
- A unit ε ("empty")
- A zero $\bot$ ("bottom", "undefined")

Resource $a \in M$: Logical assertion $\boxed{a}$ ("own a")

Logical ("ghost") resources

Partial commutative monoid (PCM)

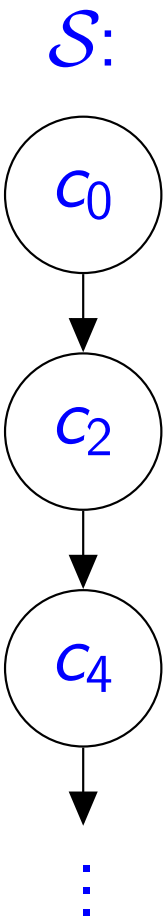
- Set M (carrier)
- An operation $\cdot$ on M (associative, commutative)
- A unit ε ("empty")
- A zero $\bot$ ("bottom", "undefined")

Resource $a \in M$: Logical assertion $\boxed{a}$ ("own a")

$$\frac{a \cdot b = c}{\boxed{a} * \boxed{b} \Leftrightarrow \boxed{c}}$$

$$\boxed{\bot} \Rightarrow \text{False}$$

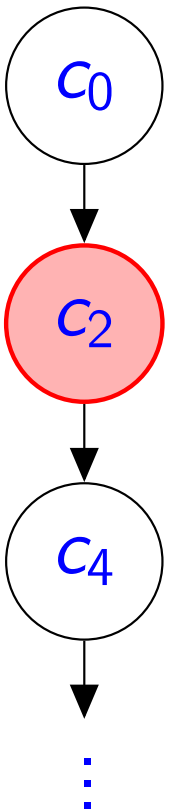
STS monoid: Intuition



STS monoid: Intuition

Curr c_2

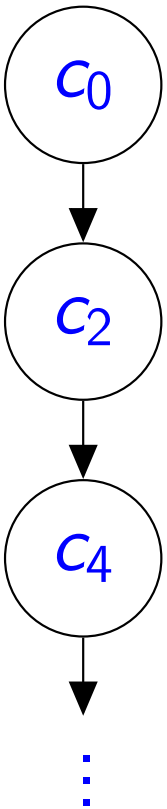
$\mathcal{S}$:



STS monoid: Intuition

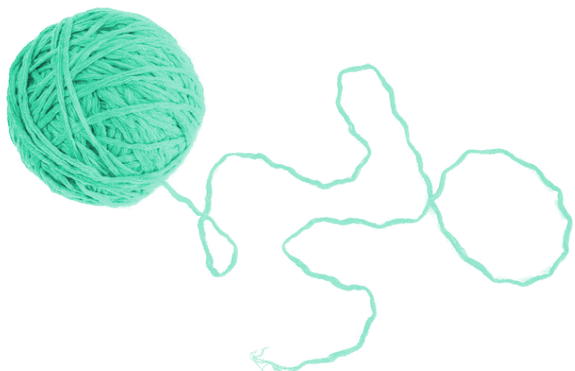
Curr c_2

$\mathcal{S}$:



Poss $\{c_j \in \mathcal{S} \mid j \geq 2\}$

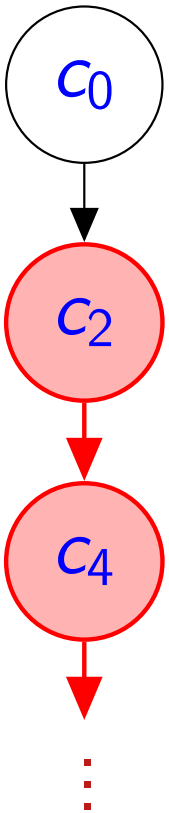
Poss $\mathcal{S}$



STS monoid: Intuition

Curr c_2

$\mathcal{S}$:



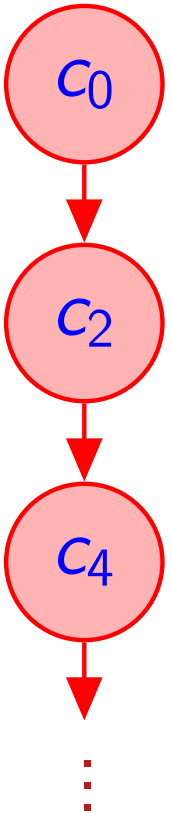
Poss $\{c_j \in \mathcal{S} \mid j \geq 2\}$

Poss $\mathcal{S}$

STS monoid: Intuition

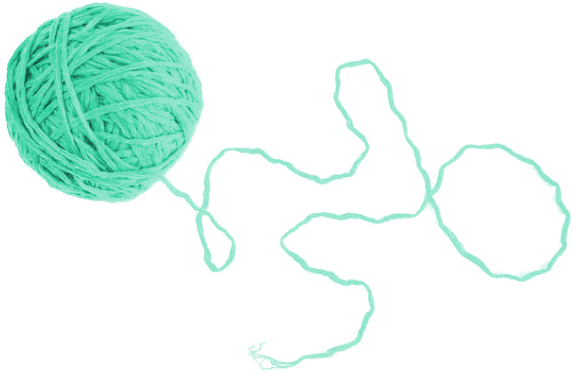
Curr c_2

$\mathcal{S}$:



Poss $\{c_j \in \mathcal{S} \mid j \geq 2\}$

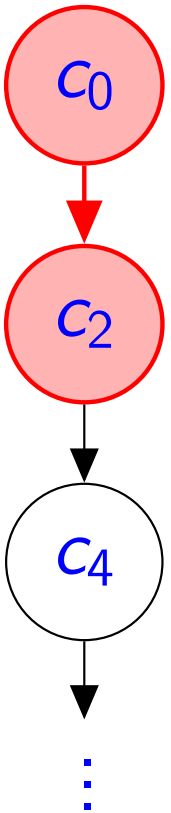
Poss $\mathcal{S}$



STS monoid: Intuition

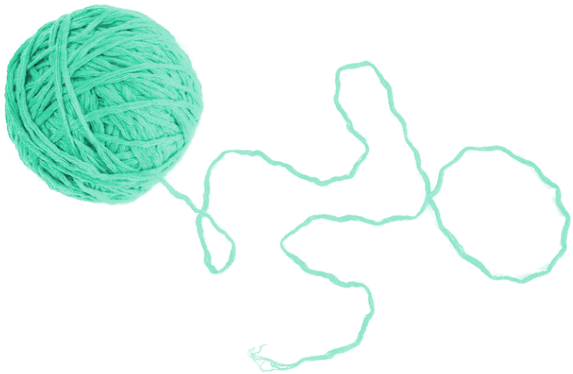
Curr c_2

$\mathcal{S}$:



Poss $\{c_j \in \mathcal{S} \mid j \geq 2\}$

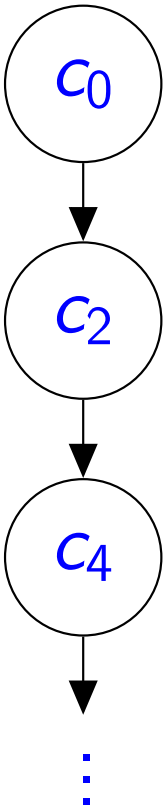
Poss $\{c_0, c_2\}$



STS monoid: Intuition

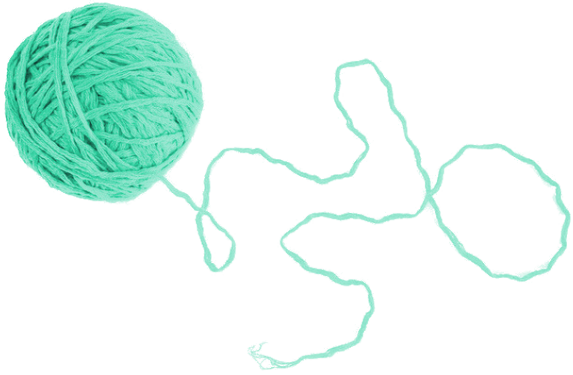
Curr c_2

$\mathcal{S}$:



Poss $\{c_j \in \mathcal{S} \mid j \geq 2\}$

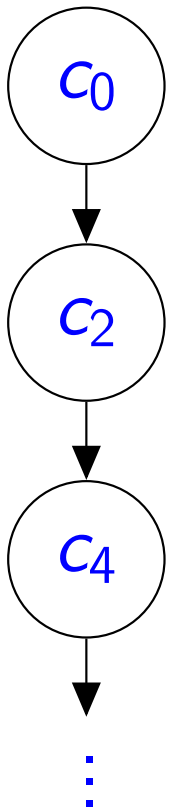
~~Poss $\{c_0, c_2\}$~~



STS monoid: Formal definition

$$M \triangleq \{\quad\} \cup \left\{\quad\right\}$$

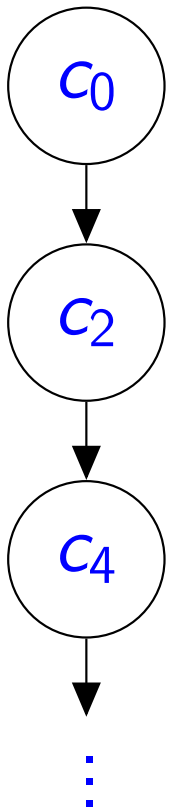
$\mathcal{S}$:



STS monoid: Formal definition

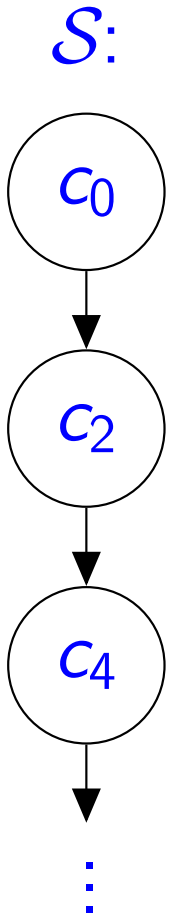
$$M \triangleq \{\mathsf{Curr}\ c \mid c \in \mathcal{S}\} \cup \left\{ \qquad \right\}$$

$\mathcal{S}$:



STS monoid: Formal definition

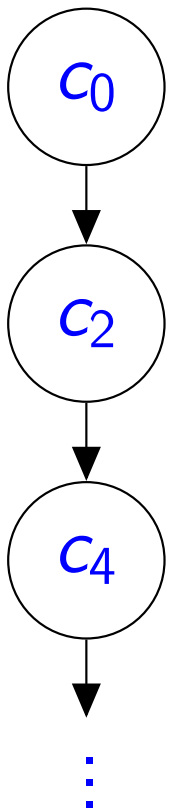
$$M \triangleq \{\mathsf{Curr}\ c \mid c \in \mathcal{S}\} \cup \left\{ \mathsf{Poss}\ B \;\middle|\; B \subseteq \mathcal{S} \wedge B \neq \emptyset \wedge \right\}$$



STS monoid: Formal definition

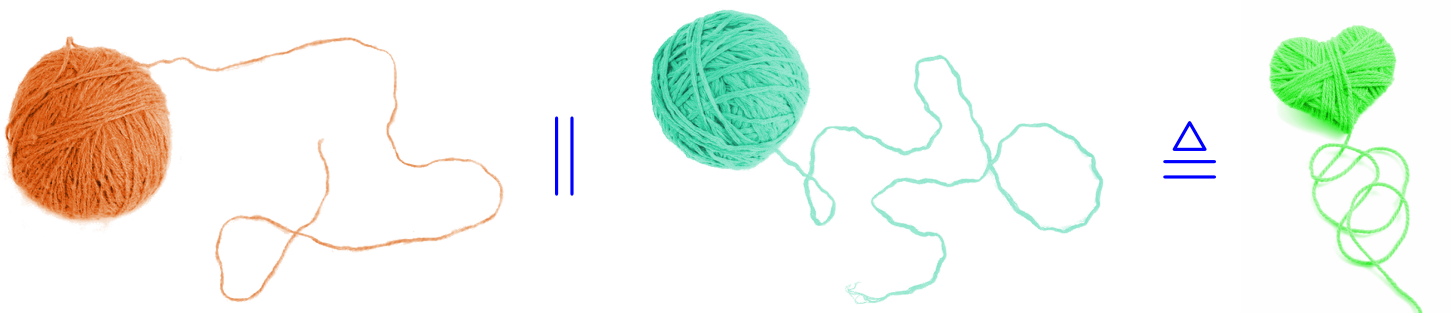
$$M \triangleq \{\text{Curr } c \mid c \in \mathcal{S}\} \cup \left\{ \text{Poss } B \;\middle|\; \begin{array}{l} B \subseteq \mathcal{S} \wedge B \neq \emptyset \wedge \\ B \text{ closed under } \rightarrow \end{array} \right\}$$

$\mathcal{S}$:



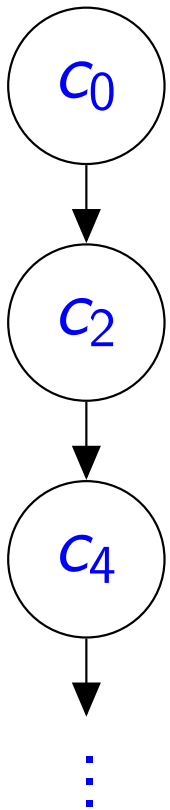
STS monoid: Formal definition

$$M \triangleq \{\text{Curr } c \mid c \in \mathcal{S}\} \cup \left\{ \text{Poss } B \;\middle|\; \begin{array}{l} B \subseteq \mathcal{S} \wedge B \neq \emptyset \wedge \\ B \text{ closed under } \rightarrow \end{array} \right\}$$



$$\text{Poss } B_1 \cdot \text{Poss } B_2 \triangleq \text{Poss } (B_1 \cap B_2)$$

$\mathcal{S}$:



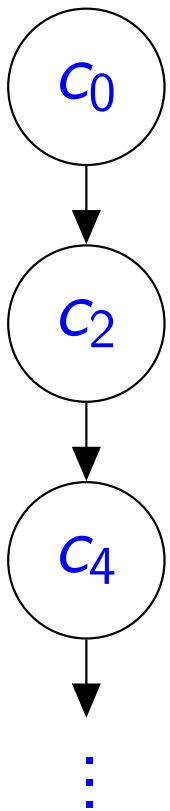
STS monoid: Formal definition

$$M \triangleq \{\text{Curr } c \mid c \in \mathcal{S}\} \cup \left\{ \text{Poss } B \;\middle|\; \begin{array}{l} B \subseteq \mathcal{S} \wedge B \neq \emptyset \wedge \\ B \text{ closed under } \rightarrow \end{array} \right\}$$

$\|$

$\triangleq$

$\mathcal{S}$:

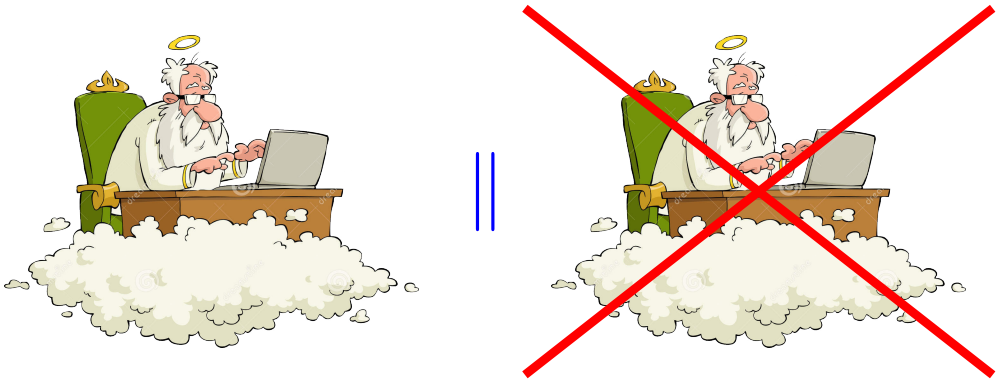


$$\text{Poss } B_1 \cdot \text{Poss } B_2 \triangleq \text{Poss } (B_1 \cap B_2)$$

$$\text{Poss } B \cdot \text{Curr } c \triangleq \text{Curr } c \quad \text{if } c \in B$$

STS monoid: Formal definition

$$M \triangleq \{\text{Curr } c \mid c \in \mathcal{S}\} \cup \left\{ \text{Poss } B \;\middle|\; \begin{array}{l} B \subseteq \mathcal{S} \wedge B \neq \emptyset \wedge \\ B \text{ closed under } \rightarrow \end{array} \right\}$$

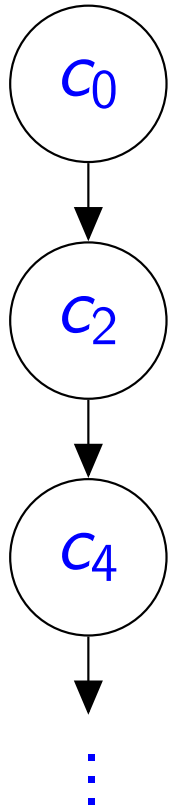


$$\text{Poss } B_1 \cdot \text{Poss } B_2 \triangleq \text{Poss } (B_1 \cap B_2)$$

$$\text{Poss } B \cdot \text{Curr } c \triangleq \text{Curr } c \quad \text{if } c \in B$$

$$\text{Curr } c_1 \cdot \text{Curr } c_2 \triangleq \bot$$

$\mathcal{S}$:



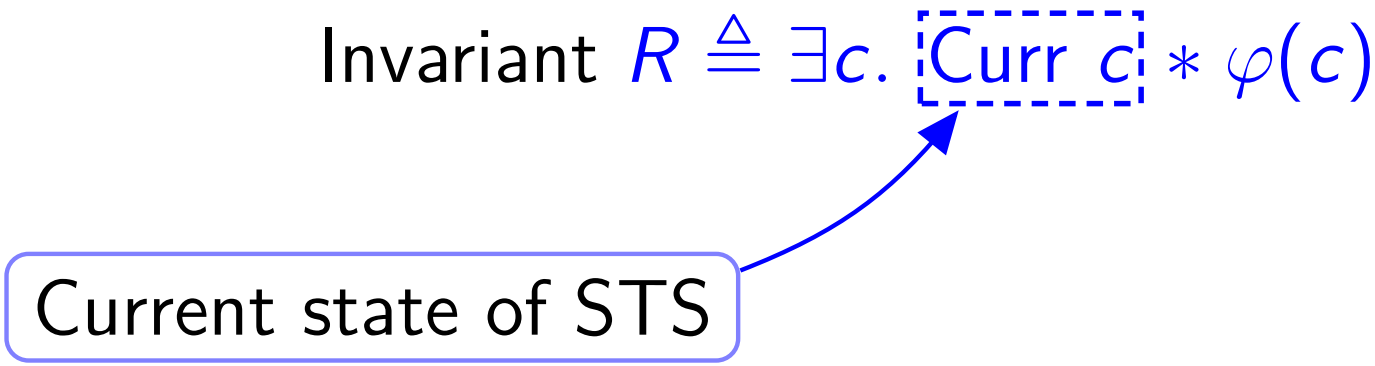
STS invariant

STS invariant

Interaction of monoids and invariants

- Monoids serve to *express* protocols.
- Invariants serve to *enforce* protocols on shared state.

STS invariant



Interaction of monoids and invariants

- Monoids serve to *express* protocols.
- Invariants serve to *enforce* protocols on shared state.

STS invariant

Invariant $R \triangleq \exists c.\ \boxed{\mathsf{Curr}\ c} * \varphi(c)$

STS interpretation

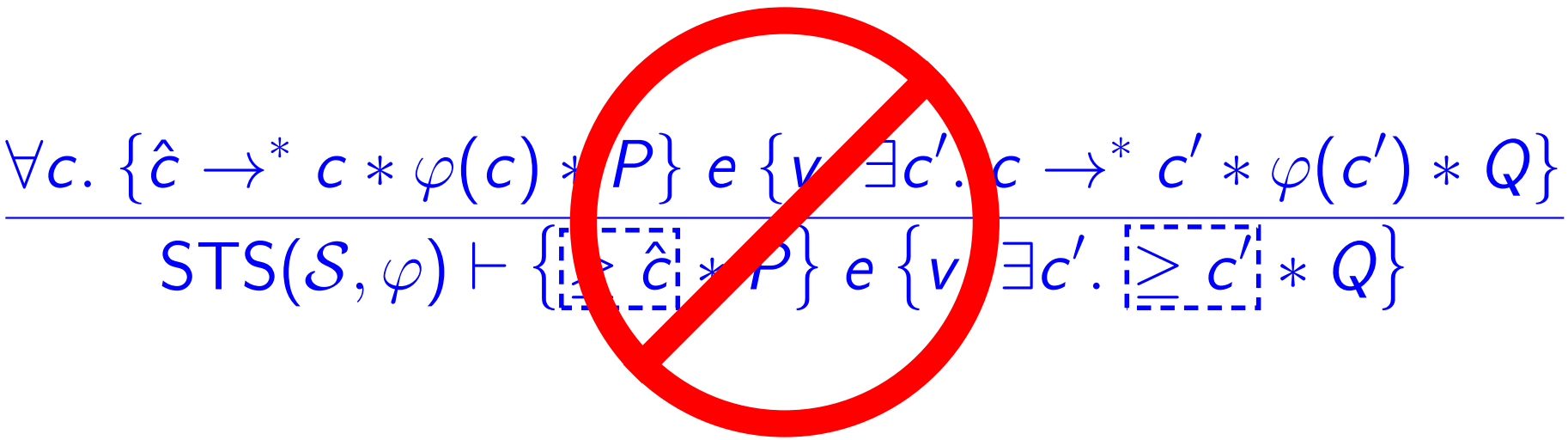
Interaction of monoids and invariants

- Monoids serve to *express* protocols.
- Invariants serve to *enforce* protocols on shared state.

Show $\{\boxed{\geq c_i}\}\ \mathit{inc2}(x)\ \{\boxed{\geq c_{i+2}}\}$

$$\frac{\forall c.\ \{\hat{c} \to^* c * \varphi(c) * P\}\ e\ \{v.\ \exists c'.\ c \to^* c' * \varphi(c') * Q\}}{\mathsf{STS}(\mathcal{S}, \varphi) \vdash \{\boxed{\geq \hat{c}} * P\}\ e\ \{v.\ \exists c'.\ \boxed{\geq c'} * Q\}}$$

Show $\{\boxed{\geq c_i}\}\ inc2(x)\ \{\boxed{\geq c_{i+2}}\}$



STS reasoning

$R \triangleq \exists c_i.\ \boxed{\text{Curr } c_i} * x \mapsto i$

$\{\boxed{\geq c_i}\}$

fn *inc2*(x) {

  do {

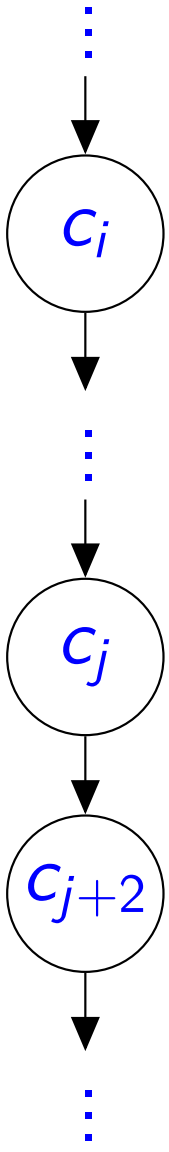
    $v = !x;$

    $b = \textbf{cas}(x, v, v + 2);$

  } **while** (**not** b);

}

$\{\boxed{\geq c_{i+2}}\}$



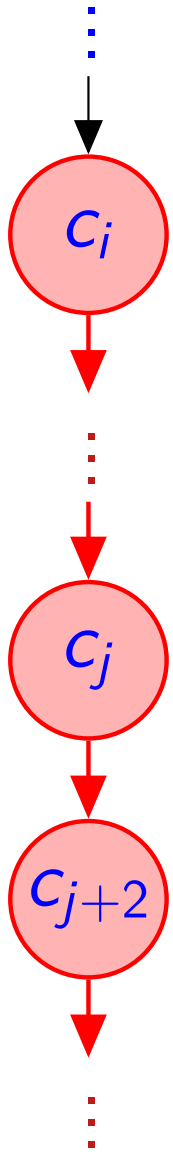
STS reasoning

$B_i \triangleq \{c_j \in \mathcal{S} \mid j \geq i\}$ $\quad R \triangleq \exists c_i.\ \boxed{\text{Curr } c_i} * x \mapsto i$

$\{\boxed{\text{Poss } B_i}\}$

fn *inc2*(x) {
  do {
    $v = !x;$
    $b = \textbf{cas}(x, v, v + 2);$
  } **while** (**not** b);
}

$\{\boxed{\text{Poss } B_{i+2}}\}$



STS reasoning

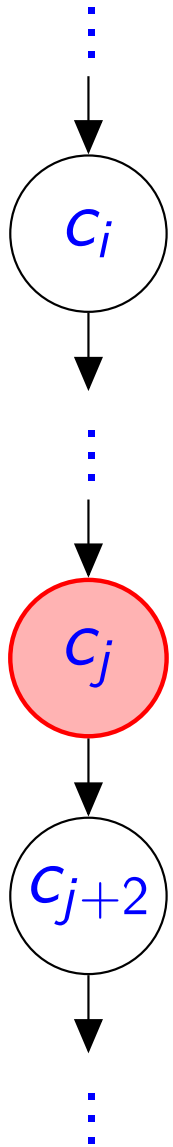
$B_i \triangleq \{c_j \in \mathcal{S} \mid j \geq i\}$ $R \triangleq \exists c_i.\ \boxed{\text{Curr } c_i} * x \mapsto i$

$\{\boxed{\text{Poss } B_i}\}$

fn *inc2*(x) {
  do {
    $v = !x;$
    $b = \textbf{cas}(x, v, v + 2);$
  } **while** (**not** b);
}

$\{\boxed{\text{Poss } B_{i+2}}\}$

Obtain R:
$\boxed{\text{Poss } B_i} * \boxed{\text{Curr } c_j} * x \mapsto j$



STS reasoning

$B_i \triangleq \{c_j \in \mathcal{S} \mid j \geq i\}$ $\quad R \triangleq \exists c_i.\ \boxed{\text{Curr } c_i} * x \mapsto i$

$\{\boxed{\text{Poss } B_i}\}$

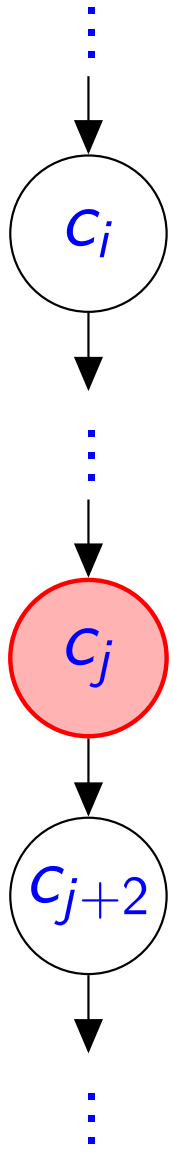
fn *inc2*(x) {
  do {
    $v = !x;$
    $b = \textbf{cas}(x, v, v + 2);$
  } **while** (**not** b);
}

$\{\boxed{\text{Poss } B_{i+2}}\}$

Obtain R:
$\boxed{\text{Poss } B_i} * \boxed{\text{Curr } c_j} * x \mapsto j$

Remember

$\text{Poss } B_i \cdot \text{Curr } c_j \triangleq \text{Curr } c_j$ if $c_j \in B_i$



STS reasoning

$B_i \triangleq \{c_j \in \mathcal{S} \mid j \geq i\}$ $\quad R \triangleq \exists c_i.\ \boxed{\text{Curr } c_i} * x \mapsto i$

$\{\boxed{\text{Poss } B_i}\}$

fn *inc2*(x) {
  do {
    $v = !x;$
    $b = \textbf{cas}(x, v, v + 2);$
  } **while** (**not** b);
}

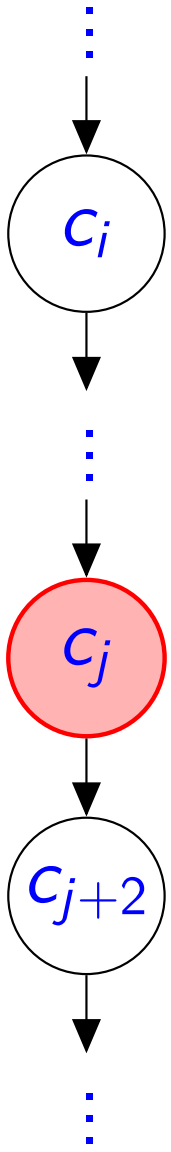
$\{\boxed{\text{Poss } B_{i+2}}\}$

So we have:

$c_j \in B_i * \boxed{\text{Curr } c_j} * x \mapsto j$

Remember

$\text{Poss } B_i \cdot \text{Curr } c_j \triangleq \text{Curr } c_j$ if $c_j \in B_i$



STS reasoning

$B_i \triangleq \{c_j \in \mathcal{S} \mid j \geq i\}$ $\quad R \triangleq \exists c_i.\ \boxed{\text{Curr } c_i} * x \mapsto i$

$\{\boxed{\text{Poss } B_i}\}$

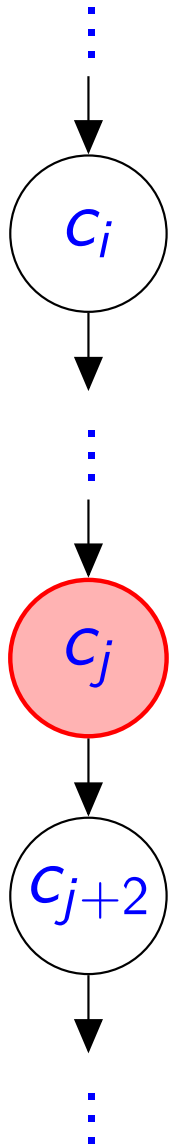
fn *inc2*(x) {
  do {
    $v = !x;$
    $b = \textbf{cas}(x, v, v + 2);$
  } **while** (**not** b);
}

$\{\boxed{\text{Poss } B_{i+2}}\}$

So we have:
$j \geq i * \boxed{\text{Curr } c_j} * x \mapsto j$

Remember

$\text{Poss } B_i \cdot \text{Curr } c_j \triangleq \text{Curr } c_j$ if $c_j \in B_i$



STS reasoning

$B_i \triangleq \{c_j \in \mathcal{S} \mid j \geq i\}$ $\quad R \triangleq \exists c_i.\ \boxed{\text{Curr } c_i} * x \mapsto i$

$\{\boxed{\text{Poss } B_i}\}$

fn *inc2*(x) {

  do {

    $v = !x;$

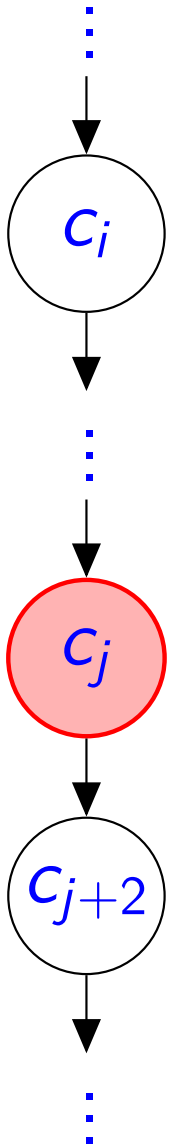
    $b = \textbf{cas}(x, v, v + 2);$

  } **while** (**not** b);

}

$\{\boxed{\text{Poss } B_{i+2}}\}$

We have:
$\boxed{\text{Curr } c_j} * x \mapsto j + 2,$
We want: R



STS reasoning

$B_i \triangleq \{c_j \in \mathcal{S} \mid j \geq i\}$ $R \triangleq \exists c_i.\ \boxed{\text{Curr } c_i} * x \mapsto i$

$\{\boxed{\text{Poss } B_i}\}$

fn *inc2*(x) {

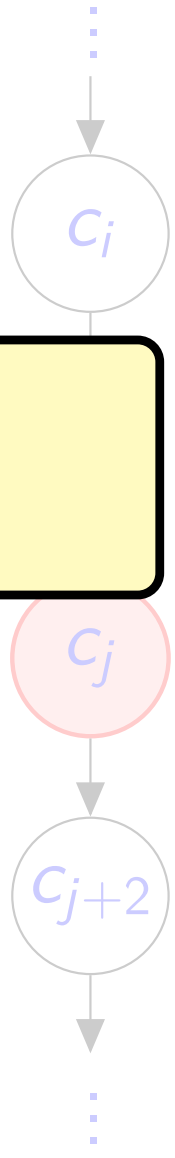
$b = \mathbf{cas}(x, v, v + 2)$;

} **while** (**not** b);

}

$\{\boxed{\text{Poss } B_{i+2}}\}$

We need a way to update $\boxed{\text{Curr } c_j}$ to $\boxed{\text{Curr } c_{j+2}}$



STS reasoning

$B_i \triangleq \{c_j \in \mathcal{S} \mid j \geq i\}$ $\quad R \triangleq \exists c_i.\ \boxed{\text{Curr } c_i} * x \mapsto i$

$\{\boxed{\text{Poss } B_i}\}$

fn *inc2*(x) {

  do {

    $v = !x;$

    $b = \textbf{cas}(x, v, v + 2);$

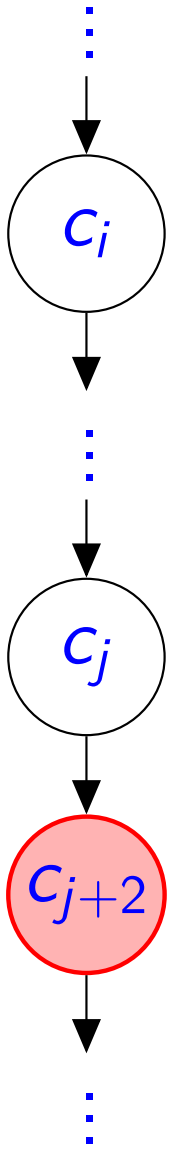
  } **while** (**not** b);

}

$\{\boxed{\text{Poss } B_{i+2}}\}$

We have R:

$\boxed{\text{Curr } c_{j+2}} * x \mapsto j + 2$



STS reasoning

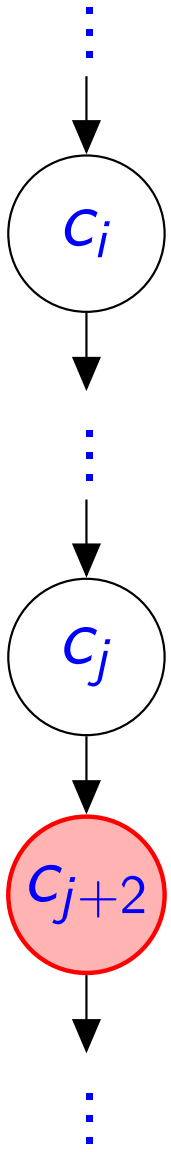
$B_i \triangleq \{c_j \in \mathcal{S} \mid j \geq i\}$ $\quad R \triangleq \exists c_i.\ \boxed{\text{Curr } c_i} * x \mapsto i$

$\{\boxed{\text{Poss } B_i}\}$

fn *inc2*(x) {
  do {
    $v = !x;$
    $b = \textbf{cas}(x, v, v + 2);$
  } **while** (**not** b);
}

$\{\boxed{\text{Poss } B_{i+2}}\}$

We have R:
$\boxed{\text{Curr } c_{j+2}} * x \mapsto j + 2,$
We want: $\boxed{\text{Poss } B_{i+2}}$



STS reasoning

$B_i \triangleq \{c_j \in \mathcal{S} \mid j \geq i\}$ $\quad R \triangleq \exists c_i.\ \boxed{\text{Curr } c_i} * x \mapsto i$

$\{\boxed{\text{Poss } B_i}\}$

fn *inc2*(x) {

  do {

    $v = !x;$

    $b = \textbf{cas}(x, v, v + 2);$

  } **while** (**not** b);

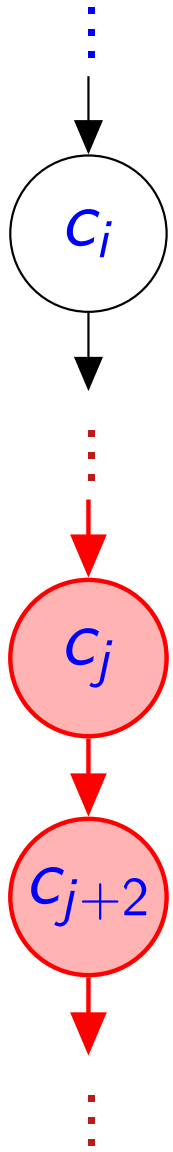
}

$\{\boxed{\text{Poss } B_{i+2}}\}$

We have R:
$\boxed{\text{Curr } c_{j+2}} * x \mapsto j + 2,$
We want: $\boxed{\text{Poss } B_{i+2}}$

Remember

$\text{Poss } B_{i+2} \cdot \text{Curr } c_{j+2}$
$\triangleq \text{Curr } c_{j+2}$
if $c_{j+2} \in B_{i+2}$



Frame-preserving ghost update

$$\boxed{\mathsf{Curr}\ c_j} \Rrightarrow \boxed{\mathsf{Curr}\ c_{j+2}}$$

Frame-preserving ghost update

$$\boxed{a} \Rrightarrow \boxed{b}$$

Frame-preserving ghost update

$$\boxed{a} \Rrightarrow \boxed{b}$$

	Us	vs.	the world
We always have	a	$\#$	a_f (for some *frame*)

where $a \# a_f \triangleq a \cdot a_f \neq \bot$

Frame-preserving ghost update

$$\boxed{a} \Rrightarrow \boxed{b}$$

	Us	vs.	the world
We always have	a	$\#$	a_f (for some *frame*)
So if we can show	b	$\#$	a_f

where $a \# a_f \triangleq a \cdot a_f \neq \bot$

Frame-preserving ghost update

$$\boxed{a} \Rrightarrow \boxed{b}$$

	Us	vs.	the world
We always have	a	$\#$	a_f (for some *frame*)
So if we can show	b	$\#$	a_f
We obtain	$\boxed{a} \Rrightarrow \boxed{b}$		

where $a \# a_f \triangleq a \cdot a_f \neq \bot$

Frame-preserving ghost update

$$\frac{\forall a_f.\ a \# a_f \Rightarrow b \# a_f}{\boxed{a} \Rrightarrow \boxed{b}}$$

cf. Views [DY+13]

	Us	vs.	the world
We always have	a	$\#$	a_f (for some *frame*)
So if we can show	b	$\#$	a_f
We obtain	$\boxed{a} \Rrightarrow \boxed{b}$		

where $a \# a_f \triangleq a \cdot a_f \neq \bot$

Frame-preserving ghost update

$$\boxed{\mathsf{Curr}\ c_j} \Rrightarrow \boxed{\mathsf{Curr}\ c_{j+2}}$$

Frame-preserving ghost update

$$\frac{c \rightarrow^* c'}{\boxed{\mathsf{Curr}\ c} \Rrightarrow \boxed{\mathsf{Curr}\ c'}}$$

Frame-preserving ghost update

$$\frac{c \to^* c'}{\boxed{\mathsf{Curr}\ c} \Rrightarrow \boxed{\mathsf{Curr}\ c'}}$$

	Us	vs.	the world
We have	$\mathsf{Curr}\ c$	$\#$	a_f

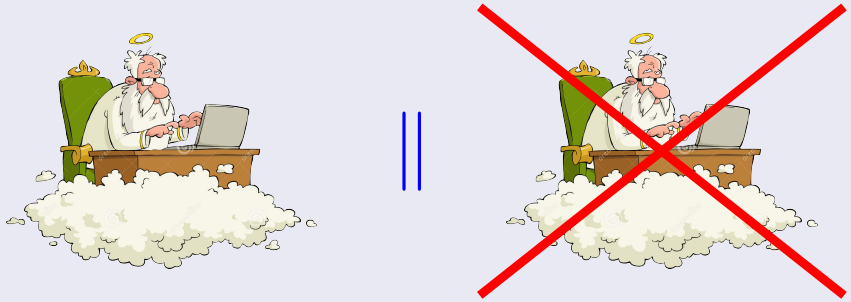
Frame-preserving ghost update

$$\frac{c \to^* c'}{\boxed{\mathsf{Curr}\ c} \Rrightarrow \boxed{\mathsf{Curr}\ c'}}$$

	Us	vs.	the world
We have	$\mathsf{Curr}\ c$	$\#$	?

Remember

$$\mathsf{Curr}\ c_1 \cdot \mathsf{Curr}\ c_2 \triangleq \bot$$



Frame-preserving ghost update

$$\frac{c \rightarrow^* c'}{\boxed{\mathsf{Curr}\ c} \Rrightarrow \boxed{\mathsf{Curr}\ c'}}$$

Us	vs.	the world
We have	$\mathsf{Curr}\ c$	$\#$ $\mathsf{Poss}\ B$

Remember

$$\mathsf{Poss}\ B \cdot \mathsf{Curr}\ c \triangleq \mathsf{Curr}\ c \quad \text{if } c \in B$$

$\|$

$\triangleq$

Frame-preserving ghost update

$$\frac{c \rightarrow^* c'}{\boxed{\mathsf{Curr}\ c} \Rrightarrow \boxed{\mathsf{Curr}\ c'}}$$

Us	vs.	the world
We have	$\mathsf{Curr}\ c$	$\#$ $\mathsf{Poss}\ B$, $c \in B$

Remember

$$\mathsf{Poss}\ B \cdot \mathsf{Curr}\ c \triangleq \mathsf{Curr}\ c \quad \text{if } c \in B$$

$\|$

$\triangleq$

Frame-preserving ghost update

$$\frac{c \rightarrow^* c'}{\boxed{\mathsf{Curr}\ c} \Rrightarrow \boxed{\mathsf{Curr}\ c'}}$$

	Us	vs.	the world
We have	$\mathsf{Curr}\ c$	#	$\mathsf{Poss}\ B,\ c \in B$
Show	$\mathsf{Curr}\ c'$	#	$\mathsf{Poss}\ B\colon\ c' \in B$

Remember

$$\mathsf{Poss}\ B \cdot \mathsf{Curr}\ c \triangleq \mathsf{Curr}\ c \quad \text{if } c \in B$$

$\parallel$

$\triangleq$

Frame-preserving ghost update

$$\frac{c \to^* c'}{\boxed{\text{Curr } c} \Rrightarrow \boxed{\text{Curr } c'}}$$

	Us	vs.	the world
We have	Curr c	#	Poss B, $c \in B$
Show	Curr c'	#	Poss B: $c' \in B$

Remember

$$M \triangleq \{\text{Curr } c \mid c \in \mathcal{S}\} \cup \left\{ \text{Poss } B \;\middle|\; \begin{array}{l} B \subseteq \mathcal{S} \land B \neq \emptyset \land \\ B \text{ closed under } \to \end{array} \right\}$$

Frame-preserving ghost update

$$\frac{c \to^* c'}{\boxed{\mathsf{Curr}\ c} \Rrightarrow \boxed{\mathsf{Curr}\ c'}}$$

	Us	vs.	the world
We have	$\mathsf{Curr}\ c$	$\#$	$\mathsf{Poss}\ B,\ c \in B$
Show	$\mathsf{Curr}\ c'$	$\#$	$\mathsf{Poss}\ B\colon\ c' \in B$

$$\boxed{\mathsf{Curr}\ c} \Rrightarrow \boxed{\mathsf{Curr}\ c'}$$

Summary: Rules for PCMs and invariants

$$\frac{\{R * P\}\ e\ \{R * Q\} \qquad e \text{ atomic}}{\boxed{R} \vdash \{P\}\ e\ \{Q\}}$$

$$\frac{\forall a_f.\ a \# a_f \Rightarrow b \# a_f}{\boxed{a} \Rrightarrow \boxed{b}}$$

$$\frac{a \cdot b = c}{\boxed{a} * \boxed{b} \Leftrightarrow \boxed{c}}$$

$$\boxed{\bot} \Rightarrow \text{False}$$