Integrating Loop Acceleration into Bounded Model Checking

Florian Frohn and Jürgen Giesl

Programming Languages and Verification, RWTH Aachen University, Germany

September 11, 2024

Safety Problems by Example

Post Variables:
$$x', y'$$
 (dom

(domain: \mathbb{Z} – in this talk

Start States:
$$\psi_s := x \le 0 \land y \le 0$$

Error States:
$$\psi_e := y \ge 100$$

Transition Formula:
$$\tau := (x < 100 \land x' = x + 1 \land y' = y) \lor (x = 100 \land x' = 0 \land y' = y + 1)$$
 (9)

Example (Transition Relation)

$$(23,42) \rightarrow_{\tau} (24,42)$$
 as $[x/23, y/42, x'/24, y'/42] \models \tau$

Dependency Graphs



Safety Problems by Example

Pre Variables: x, y

Post Variables: x', y'

(domain: \mathbb{Z} – in this talk)

Start States: $\psi_s := x \le 0 \land y \le 0$

Error States: $\psi_e := y \geq 1$

Transition Formula: $z := (x \le 100 \text{ A } y' = x)$

 $(x = y) \lor (x = 100 \land x' = 0 \land y' = y + 1)$

Example (Transition Relation)

 $(23,42) \rightarrow_{\tau} (24,42)$ as $[x/23,y/42,x'/24,y'/42] \models \tau$



Safety Problems by Example

Pre Variables: x, y

Post Variables: x', y'

(domain: \mathbb{Z} – in this talk)

Start States: $\psi_s := x \le 0 \land y \le 0$

Error States: $\psi_e := y \ge 10$

Transition Formula: $au:=(x<100 \land x'=x+1 \land y'=y) \lor (x=100 \land x'=0 \land y'=y+1)$

Example (Transition Relation)

 $(23,42) \rightarrow_{\tau} (24,42)$ as $[x/23,y/42,x'/24,y'/42] \models \tau$





Safety Problems by Example

Pre Variables: x, y Post Variables: x', y'

Start States: $\psi_s := x \le 0 \land y \le 0$ Error States: $\psi_e := y \ge 100$

Transition Formula: $\tau := (x < 100 \land x' = x + 1 \land y' = y) \lor (x = 100 \land x' = 0 \land y' = y + 1)$ (QF)

Example (Transition Relation)

 $(23,42) \rightarrow_{\tau} (24,42)$ as $[x/23, y/42, x'/24, y'/42] \models \tau$





(domain: \mathbb{Z} – in this talk)

Safety Problems by Example

Pre Variables: x, v

Post Variables: x', y' (domain: \mathbb{Z} – in this talk)

Start States: $\psi_s := x \le 0 \land y \le 0$

Error States: $\psi_e := y \ge 100$

Transition Formula:
$$\tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r}$$
 (QF)



Safety Problems by Example

Pre Variables: x, v

Post Variables: x', y' (domain: \mathbb{Z} – in this talk)

Start States: $\psi_s := x \le 0 \land y \le 0$

Error States: $\psi_e := y \ge 100$

Transition Formula:
$$\tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r}$$
 (QF)

Example (Transition Relation)

$$(23,42) \rightarrow_{\tau} (24,42)$$
 as $[x/23, y/42, x'/24, y'/42] \models \tau$



Safety Problems by Example

Pre Variables: x, v

Post Variables: x', y' (domain: \mathbb{Z} – in this talk)

Start States: $\psi_s := x \le 0 \land y \le 0$ Error States: $\psi_e := y \ge 100$

Transition Formula: $\tau := (x < 100 \land x' = x + 1 \land y' = y) \lor (x = 100 \land x' = 0 \land y' = y + 1)$ (QF)

Example (Transition Relation)

$$(23,42) \rightarrow_{\tau} (24,42)$$
 as $[x/23, y/42, x'/24, y'/42] \models \tau$

Dependency Graphs



Leading Example

```
\begin{array}{ll} \text{Start States: } \psi_s := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_e := y \geq 100 \\ \text{Transition Formula: } \tau := (x < 100 \land x' = x + 1 \land y' = y) \lor (x = 100 \land x' = 0 \land y' = y + 1) \end{array}
```

ВМС

```
\begin{array}{lll} b \leftarrow 0; & \mathsf{add}(\mathsf{vr}_b(\psi_{\mathfrak{s}})) & \mathsf{where} & \mathsf{vr}_b(x) \coloneqq x_b, \mathsf{vr}_b(x') \coloneqq x_{b+1}, \dots \\ & \mathsf{while} \; \top \; \mathsf{do} \\ & \mathsf{push}(); & \mathsf{add}(\mathsf{vr}_b(\psi_e)) \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{sat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{trace}() \; \; \mathsf{else} \; \; \mathsf{pop}(); \; \; \mathsf{add}(\mathsf{vr}_b(\tau)) \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{unsat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{safe} \; \; \mathsf{else} \; \; b++ \end{array}
```

```
• \operatorname{vr}_0(\psi_s) \wedge \operatorname{vr}_0(\tau) \wedge \operatorname{vr}_1(\psi_e) \curvearrowright \operatorname{unsat}

• \operatorname{vr}_0(\psi_s) \wedge \operatorname{vr}_0(\tau) \wedge \operatorname{vr}_1(\tau) \wedge \operatorname{vr}_2(\psi_e) \curvearrowright \operatorname{unsat}
```

- . .
- $\mathsf{vr}_0(\psi_s) \wedge \mathsf{vr}_0(\tau) \wedge \ldots \wedge \mathsf{vr}_{10099}(\tau) \wedge \mathsf{vr}_{10100}(\psi_e) \curvearrowright \mathsf{sat}$
- trace() = $[\tau_i]_{i=0}$

Leading Example

```
\begin{array}{ll} \text{Start States: } \psi_s := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_e := y \geq 100 \\ \text{Transition Formula: } \tau := (x < 100 \land x' = x + 1 \land y' = y) \lor (x = 100 \land x' = 0 \land y' = y + 1) \end{array}
```

ВМС

```
\begin{array}{lll} b \leftarrow 0; & \mathsf{add}(\mathsf{vr}_b(\psi_s)) & \mathsf{where} & \mathsf{vr}_b(x) \coloneqq x_b, \mathsf{vr}_b(x') \coloneqq x_{b+1}, \dots \\ & \mathsf{while} \; \top \; \mathsf{do} & \\ & \mathsf{push}(); & \mathsf{add}(\mathsf{vr}_b(\psi_e)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{sat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{trace}() \; \; \mathsf{else} \; \; \mathsf{pop}(); & \mathsf{add}(\mathsf{vr}_b(\tau)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{unsat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{safe} \; \; \mathsf{else} \; \; b++ & \end{array}
```

- $\operatorname{vr}_0(\psi_s) \wedge \operatorname{vr}_0(\psi_e) \curvearrowright \operatorname{unsat}$
- $\operatorname{vr}_0(\psi_s) \wedge \operatorname{vr}_0(\tau) \wedge \operatorname{vr}_1(\psi_e) \curvearrowright \operatorname{unsat}$
- $\operatorname{vr}_0(\psi_s) \wedge \operatorname{vr}_0(\tau) \wedge \operatorname{vr}_1(\tau) \wedge \operatorname{vr}_2(\psi_e) \curvearrowright \operatorname{unsat}$
- .
- $\mathsf{vr}_0(\psi_s) \wedge \mathsf{vr}_0(\tau) \wedge \ldots \wedge \mathsf{vr}_{10099}(\tau) \wedge \mathsf{vr}_{10100}(\psi_e) \curvearrowright \mathsf{sat}$
- trace() = $[\tau_i]_{i=0}$

Leading Example

```
\begin{array}{ll} \text{Start States: } \psi_s := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_e := y \geq 100 \\ \text{Transition Formula: } \tau := (x < 100 \land x' = x + 1 \land y' = y) \lor (x = 100 \land x' = 0 \land y' = y + 1) \end{array}
```

ВМС

```
\begin{array}{lll} b \leftarrow 0; & \mathsf{add}(\mathsf{vr}_b(\psi_s)) & \mathsf{where} & \mathsf{vr}_b(x) \coloneqq x_b, \mathsf{vr}_b(x') \coloneqq x_{b+1}, \dots \\ & \mathsf{while} \; \top \; \mathsf{do} & \\ & \mathsf{push}(); & \mathsf{add}(\mathsf{vr}_b(\psi_e)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{sat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{trace}() \; \; \mathsf{else} \; \; \mathsf{pop}(); & \mathsf{add}(\mathsf{vr}_b(\tau)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{unsat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{safe} \; \; \mathsf{else} \; \; b++ & \end{array}
```

- $\operatorname{vr}_0(\psi_s) \wedge \operatorname{vr}_0(\psi_e) \curvearrowright \operatorname{unsat}$
- ullet vr $_0(\psi_s) \wedge ext{vr}_0(au) \wedge ext{vr}_1(\psi_e) \sim ext{unsat}$

ullet vr $_0(\psi_s) \wedge$ vr $_0(au) \wedge$ vr $_1(au) \wedge$ vr $_2(\psi_e) \curvearrowright$ unsat

- . . .
- $\operatorname{vr}_0(\psi_s) \wedge \operatorname{vr}_0(\tau) \wedge \ldots \wedge \operatorname{vr}_{10099}(\tau) \wedge \operatorname{vr}_{10100}(\psi_e) \curvearrowright \operatorname{sat}$
- trace() = $[\tau_i]_{i=0}$

Leading Example

```
\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := (x < 100 \wedge x' = x + 1 \wedge y' = y) \vee (x = 100 \wedge x' = 0 \wedge y' = y + 1) \end{array}
```

ВМС

```
\begin{array}{lll} b \leftarrow 0; & \mathsf{add}(\mathsf{vr}_b(\psi_s)) & \mathsf{where} & \mathsf{vr}_b(x) \coloneqq x_b, \mathsf{vr}_b(x') \coloneqq x_{b+1}, \dots \\ & \mathsf{while} \; \top \; \mathsf{do} & \\ & \mathsf{push}(); & \mathsf{add}(\mathsf{vr}_b(\psi_e)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{sat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{trace}() \; \; \mathsf{else} \; \; \mathsf{pop}(); & \mathsf{add}(\mathsf{vr}_b(\tau)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{unsat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{safe} \; \; \mathsf{else} \; \; b++ & \end{array}
```

- \bullet vr $_0(\psi_s) \wedge$ vr $_0(\psi_e) \curvearrowright$ unsat
- ullet vr $_0(\psi_s) \wedge$ vr $_0(au) \wedge$ vr $_1(\psi_e) \curvearrowright$ unsat
- $\operatorname{vr}_0(\psi_s) \wedge \operatorname{vr}_0(\tau) \wedge \operatorname{vr}_1(\tau) \wedge \operatorname{vr}_2(\psi_e) \curvearrowright \operatorname{unsat}$
- .
- $\mathsf{vr}_0(\psi_s) \wedge \mathsf{vr}_0(\tau) \wedge \ldots \wedge \mathsf{vr}_{10099}(\tau) \wedge \mathsf{vr}_{10100}(\psi_e) \curvearrowright \mathsf{sat}$
- trace() = $[\tau_i]_{i=0}^{b-1}$

Leading Example

```
\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := (x < 100 \wedge x' = x + 1 \wedge y' = y) \vee (x = 100 \wedge x' = 0 \wedge y' = y + 1) \end{array}
```

ВМС

```
\begin{array}{lll} b \leftarrow 0; & \mathsf{add}(\mathsf{vr}_b(\psi_s)) & \mathsf{where} & \mathsf{vr}_b(x) \coloneqq x_b, \mathsf{vr}_b(x') \coloneqq x_{b+1}, \dots \\ \mathsf{while} \; \top \; \mathsf{do} & \\ & \mathsf{push}(); & \mathsf{add}(\mathsf{vr}_b(\psi_e)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{sat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{trace}() \; \; \mathsf{else} \; \; \mathsf{pop}(); & \mathsf{add}(\mathsf{vr}_b(\tau)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{unsat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{safe} \; \; \mathsf{else} \; \; b++ & \end{array}
```

- \bullet vr $_0(\psi_s) \wedge$ vr $_0(\psi_e) \curvearrowright$ unsat
- $\mathsf{vr}_0(\psi_s) \wedge \mathsf{vr}_0(\tau) \wedge \mathsf{vr}_1(\psi_e) \curvearrowright \mathsf{unsat}$
- $\operatorname{vr}_0(\psi_s) \wedge \operatorname{vr}_0(\tau) \wedge \operatorname{vr}_1(\tau) \wedge \operatorname{vr}_2(\psi_e) \curvearrowright \operatorname{unsat}$
- . . .
- $\mathsf{vr}_0(\psi_s) \wedge \mathsf{vr}_0(\tau) \wedge \ldots \wedge \mathsf{vr}_{10099}(\tau) \wedge \mathsf{vr}_{10100}(\psi_e) \curvearrowright \mathsf{sat}$
- trace() = $[\tau_i]_{i=0}^{b-1}$

Leading Example

```
\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := (x < 100 \wedge x' = x + 1 \wedge y' = y) \vee (x = 100 \wedge x' = 0 \wedge y' = y + 1) \end{array}
```

ВМС

```
\begin{array}{lll} b \leftarrow 0; & \mathsf{add}(\mathsf{vr}_b(\psi_s)) & \mathsf{where} & \mathsf{vr}_b(x) \coloneqq x_b, \mathsf{vr}_b(x') \coloneqq x_{b+1}, \dots \\ \mathsf{while} \; \top \; \mathsf{do} & \\ & \mathsf{push}(); & \mathsf{add}(\mathsf{vr}_b(\psi_e)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{sat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{trace}() \; \; \mathsf{else} \; \; \mathsf{pop}(); & \mathsf{add}(\mathsf{vr}_b(\tau)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{unsat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{safe} \; \; \mathsf{else} \; \; b++ & \end{array}
```

- $\operatorname{vr}_0(\psi_s) \wedge \operatorname{vr}_0(\psi_e) \curvearrowright \operatorname{unsat}$
- \bullet $\mathsf{vr}_0(\psi_s) \wedge \mathsf{vr}_0(au) \wedge \mathsf{vr}_1(\psi_e) \curvearrowright \mathsf{unsat}$

- ullet vr $_0(\psi_s) \wedge$ vr $_0(au) \wedge$ vr $_1(au) \wedge$ vr $_2(\psi_e) \curvearrowright$ unsat
- . . .
- \bullet $\mathsf{vr}_0(\psi_s) \wedge \mathsf{vr}_0(au) \wedge \ldots \wedge \mathsf{vr}_{10099}(au) \wedge \mathsf{vr}_{10100}(\psi_e) \curvearrowright \mathsf{sat}$
- trace() = $[\tau_i]_{i=1}^{\infty}$

Leading Example

```
\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := (x < 100 \wedge x' = x + 1 \wedge y' = y) \vee (x = 100 \wedge x' = 0 \wedge y' = y + 1) \end{array}
```

ВМС

```
\begin{array}{lll} b \leftarrow 0; & \mathsf{add}(\mathsf{vr}_b(\psi_{\mathfrak{s}})) & \mathsf{where} & \mathsf{vr}_b(x) \coloneqq x_b, \mathsf{vr}_b(x') \coloneqq x_{b+1}, \dots \\ & \mathsf{while} \; \top \; \mathsf{do} \\ & | \; \mathsf{push}(); & \mathsf{add}(\mathsf{vr}_b(\psi_e)) \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{sat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{trace}() \; \; \mathsf{else} \; \; \mathsf{pop}(); & \mathsf{add}(\mathsf{vr}_b(\tau)) \\ & | \; \mathsf{if} \; \mathsf{check}() = \mathsf{unsat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{safe} \; \; \mathsf{else} \; \; b++ \end{array}
```

- \bullet vr $_0(\psi_s) \wedge$ vr $_0(\psi_e) \curvearrowright$ unsat
- $\mathsf{vr}_0(\psi_s) \wedge \mathsf{vr}_0(\tau) \wedge \mathsf{vr}_1(\psi_e) \curvearrowright \mathsf{unsat}$

- ullet vr $_0(\psi_s) \wedge$ vr $_0(au) \wedge$ vr $_1(au) \wedge$ vr $_2(\psi_e) \curvearrowright$ unsat
- ...
- ullet vr $_0(\psi_s) \wedge$ vr $_0(au) \wedge \ldots \wedge$ vr $_{10099}(au) \wedge$ vr $_{10100}(\psi_e) \curvearrowright$ sat
- trace() = $[\tau_i]_{i=0}^{b-1}$ where $\tau_i = \bigwedge \{\ell \in literals(\tau) \mid \mathsf{vr}_i(\ell) \text{ is true in current model} \}$

Leading Example

```
\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := (x < 100 \wedge x' = x + 1 \wedge y' = y) \vee (x = 100 \wedge x' = 0 \wedge y' = y + 1) \end{array}
```

ВМС

```
\begin{array}{lll} b \leftarrow 0; & \mathsf{add}(\mathsf{vr}_b(\psi_s)) & \mathsf{where} & \mathsf{vr}_b(x) \coloneqq x_b, \mathsf{vr}_b(x') \coloneqq x_{b+1}, \dots \\ & \mathsf{while} \; \top \; \mathsf{do} & \\ & \mathsf{push}(); & \mathsf{add}(\mathsf{vr}_b(\psi_e)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{sat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{trace}() \; \; \mathsf{else} \; \; \mathsf{pop}(); & \mathsf{add}(\mathsf{vr}_b(\tau)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{unsat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{safe} \; \; \mathsf{else} \; \; b++ & \end{array}
```

- \bullet vr $_0(\psi_s) \wedge$ vr $_0(\psi_e) \curvearrowright$ unsat
- $\mathsf{vr}_0(\psi_s) \wedge \mathsf{vr}_0(\tau) \wedge \mathsf{vr}_1(\psi_e) \curvearrowright \mathsf{unsat}$

- ullet vr $_0(\psi_s) \wedge$ vr $_0(au) \wedge$ vr $_1(au) \wedge$ vr $_2(\psi_e) \curvearrowright$ unsat
- ...
- $\mathsf{vr}_0(\psi_s) \wedge \mathsf{vr}_0(\tau) \wedge \ldots \wedge \mathsf{vr}_{10099}(\tau) \wedge \mathsf{vr}_{10100}(\psi_e) \curvearrowright \mathsf{sat}$
- trace() = $[\tau_i]_{i=0}^{b-1}$ where $\tau_i = \bigwedge \{\ell \in \mathit{literals}(\tau) \mid \mathsf{vr}_i(\ell) \text{ is true in current model} \}$

Leading Example

```
\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := (x < 100 \wedge x' = x + 1 \wedge y' = y) \vee (x = 100 \wedge x' = 0 \wedge y' = y + 1) \end{array}
```

ВМС

```
\begin{array}{lll} b \leftarrow 0; & \mathsf{add}(\mathsf{vr}_b(\psi_s)) & \mathsf{where} & \mathsf{vr}_b(x) \coloneqq x_b, \mathsf{vr}_b(x') \coloneqq x_{b+1}, \dots \\ & \mathsf{while} \; \top \; \mathsf{do} & \\ & \mathsf{push}(); & \mathsf{add}(\mathsf{vr}_b(\psi_e)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{sat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{trace}() \; \; \mathsf{else} \; \; \mathsf{pop}(); & \mathsf{add}(\mathsf{vr}_b(\tau)) & \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{unsat} \; \mathsf{then} \; \; \mathsf{return} \; \mathsf{safe} \; \; \mathsf{else} \; \; b++ & \end{array}
```

- \bullet vr $_0(\psi_s) \wedge$ vr $_0(\psi_e) \curvearrowright$ unsat
- $\mathsf{vr}_0(\psi_s) \wedge \mathsf{vr}_0(\tau) \wedge \mathsf{vr}_1(\psi_e) \curvearrowright \mathsf{unsat}$

- ullet vr $_0(\psi_s) \wedge$ vr $_0(au) \wedge$ vr $_1(au) \wedge$ vr $_2(\psi_e) \curvearrowright$ unsat
- ...
- ullet vr $_0(\psi_s) \wedge$ vr $_0(au) \wedge \ldots \wedge$ vr $_{10099}(au) \wedge$ vr $_{10100}(\psi_e) \curvearrowright$ sat
- trace() = $[\tau_i]_{i=0}^{b-1}$ where $\tau_i = \bigwedge \{\ell \in \mathit{literals}(\tau) \mid \mathsf{vr}_i(\ell) \text{ is true in current model}\}$

Leading Example

$$\tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r}$$

Definition (Acceleration)

A function with $\operatorname{accel}(\tau) := \tau^{\oplus}$ where $\to_{\tau^{\oplus}} \subseteq \to_{\tau}^+$.

Example

Leading Example

$$\tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r}$$

Definition (Acceleration)

A function with $\operatorname{accel}(\tau) := \tau^{\oplus}$ where $\to_{\tau^{\oplus}} \subseteq \to_{\tau}^+$.

Example

Leading Example

$$\tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r}$$

Definition (Acceleration)

A function with $\operatorname{accel}(\tau) := \tau^{\oplus}$ where $\rightarrow_{\tau^{\oplus}} \subseteq \rightarrow_{\tau}^{+}$.

Example

$$egin{array}{ll} au := au_i ee au_r \ au_i := au_i < 100 \wedge x' = x + 1 \wedge y' = y \ ^\oplus := x + n - 1 < 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

Leading Example

$$\tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r}$$

Definition (Acceleration)

A function with $\operatorname{accel}(\tau) := \tau^{\oplus}$ where $\to_{\tau^{\oplus}} \subseteq \to_{\tau}^+$.

Example

$$\tau := \tau_i \lor \tau_r
\tau_i := x < 100 \land x' = x + 1 \land y' = y
\stackrel{\text{(f)}}{:} := x + n - 1 < 100 \land x' = x + n \land y' = y \land n > 0$$

Leading Example

$$\tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r}$$

Definition (Acceleration)

A function with $\operatorname{accel}(\tau) := \tau^{\oplus}$ where $\to_{\tau^{\oplus}} \subseteq \to_{\tau}^+$.

Example

$$\tau := \tau_i \vee \tau_r$$

$$\tau_i := x < 100 \wedge x' = x + 1 \wedge y' = y$$

Leading Example

$$\tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r}$$

Definition (Acceleration)

A function with $\operatorname{accel}(\tau) := \tau^{\oplus}$ where $\to_{\tau^{\oplus}} \subseteq \to_{\tau}^+$.

Example

$$\begin{array}{ll} \tau := \tau_i \vee \tau_r \\ \tau_i := & x < 100 \wedge x' = x + 1 \wedge y' = y \\ \tau_i^{\oplus} := x + n - 1 < 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

How? Many techniques, e.g.,

monotonicity criteria: $au_i \implies x < x'$

Leading Example

$$\tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r}$$

Definition (Acceleration)

A function with $\operatorname{accel}(\tau) := \tau^{\oplus}$ where $\rightarrow_{\tau^{\oplus}} \subseteq \rightarrow_{\tau}^{+}$.

Example

$$\begin{array}{ll} \tau := \tau_i \vee \tau_r \\ \tau_i := & x < 100 \wedge x' = x + 1 \wedge y' = y \\ \tau_i^{\oplus} := x + n - 1 < 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

- monotonicity criteria: $\tau_i \implies x < x'$
- recurrence solving

Leading Example

$$\tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r}$$

Definition (Acceleration)

A function with $\operatorname{accel}(\tau) := \tau^{\oplus}$ where $\to_{\tau^{\oplus}} \subseteq \to_{\tau}^+$.

Example

$$\begin{split} \tau &:= \tau_i \vee \tau_r \\ \tau_i &:= \qquad x < 100 \wedge x' = x + 1 \wedge y' = y \\ \tau_i^{\oplus} &:= x + n - 1 < 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{split}$$

- monotonicity criteria: $\tau_i \implies x < x'$
- recurrence solving

Leading Example

$$\tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r}$$

Definition (Acceleration)

A function with $\operatorname{accel}(\tau) := \tau^{\oplus}$ where $\to_{\tau^{\oplus}} \subseteq \to_{\tau}^+$.

Example

$$\begin{split} \tau &:= \tau_i \vee \tau_r \\ \tau_i &:= \qquad x < 100 \wedge x' = x + 1 \wedge y' = y \\ \tau_i^{\oplus} &:= x + n - 1 < 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{split}$$

- monotonicity criteria: $\tau_i \implies x < x'$
- recurrence solving

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_s := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_e := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r} \\ \tau_i^\oplus := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

ABMC

```
b \leftarrow 0; \mathsf{add}(\mathsf{vr}_b(\psi_s))

if \mathsf{check}() = \mathsf{unsat} then return \mathsf{safe}

while \top do

\mathsf{push}(); \mathsf{add}(\mathsf{vr}_b(\psi_e))

if \mathsf{check}() = \mathsf{sat} then return \vec{\tau}

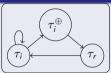
else

\mathsf{pop}(); \vec{\tau} \leftarrow \mathsf{trace}();

if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then

\mathsf{add}(\mathsf{vr}_b(\tau \vee \mathsf{accel}(\vec{\tau}^{\circlearrowleft})))

else \mathsf{add}(\mathsf{vr}_b(\tau))
```



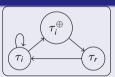
Ь	$ec{ au}$	SMT Problem	Model
0		$\operatorname{vr}_0(\psi_s) \wedge \operatorname{vr}_0(\tau)$	$[x_1/1,\ldots/0]$
1	$[\tau_i]$	$\ldots \wedge \operatorname{vr}_1(\tau \vee \tau_i^{\oplus})$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^{\oplus}]$	$\dots \wedge vr_2(\tau)$	\cup [$x_3/0, y_3/1$]
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(\tau \vee accel(\bar{\tau}^{\circlearrowleft}))$	$ \cup [x_4/0, y_4/100]$
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \wedge x' = x + 1 \wedge y' = y\right)}_{\tau_{i}} \vee \underbrace{\left(x = 100 \wedge x' = 0 \wedge y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

ABMC

$$\begin{array}{ll} b \leftarrow 0; & \mathsf{add}(\mathsf{vr}_b(\psi_s)) \\ \mathsf{if} & \mathsf{check}() = \mathsf{unsat} \; \mathsf{then} \; \mathsf{return} \; \mathsf{safe} \\ \mathsf{while} \; \top \; \mathsf{do} \\ & \mathsf{push}(); & \mathsf{add}(\mathsf{vr}_b(\psi_e)) \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{sat} \; \mathsf{then} \; \; \mathsf{return} \; \vec{\tau} \\ & \mathsf{else} \\ & \middle| \; \mathsf{pop}(); \; \vec{\tau} \leftarrow \mathsf{trace}(); \\ & \mathsf{if} \; \vec{\tau} \; \mathsf{ends} \; \mathsf{with} \; \mathsf{loop} \; \vec{\tau}^\circlearrowleft \; \mathsf{then} \\ & \middle| \; \mathsf{add}(\mathsf{vr}_b(\tau \vee \mathsf{accel}(\vec{\tau}^\circlearrowleft))) \\ & \mathsf{else} \; \mathsf{add}(\mathsf{vr}_b(\tau)) \end{array}$$



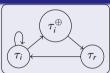
Ь	$ec{ au}$	SMT Problem	Model
0		$\operatorname{vr}_0(\psi_s) \wedge \operatorname{vr}_0(\tau)$	$[x_1/1, \dots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(\tau \vee \tau_i^{\oplus})$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^{\oplus}]$	$\dots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[\tau_i, \tau_i^{\oplus}, \tau_r]$	$\wedge \operatorname{vr}_3(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft}))$	$\ldots \cup [x_4/0, y_4/100]$
4		\wedge Vr ₄ (ψ_e)	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \wedge x' = x + 1 \wedge y' = y\right)}_{\tau_{i}} \vee \underbrace{\left(x = 100 \wedge x' = 0 \wedge y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

ABMC

$$b \leftarrow 0$$
; $\operatorname{add}(\operatorname{vr}_b(\psi_s))$
if $\operatorname{check}() = \operatorname{unsat}$ then return safe
while \top do
 $\operatorname{push}()$; $\operatorname{add}(\operatorname{vr}_b(\psi_e))$
if $\operatorname{check}() = \operatorname{sat}$ then return $\vec{\tau}$
else
 $\operatorname{pop}()$; $\vec{\tau} \leftarrow \operatorname{trace}()$;
if $\vec{\tau}$ ends with loop $\vec{\tau}^{\circlearrowleft}$ then
 $\operatorname{add}(\operatorname{vr}_b(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\operatorname{add}(\operatorname{vr}_b(\tau))$



Ь	$ec{ au}$	SMT Problem	Model
0		$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(\tau \vee \tau_i^{\oplus})$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^{\oplus}]$	$\dots \wedge vr_2(\tau)$	\cup [$x_3/0, y_3/1$]
3	$[\tau_i, \tau_i^{\oplus}, \tau_r]$	17-6333	$\ldots \cup [x_4/0, y_4/100]$
4		$\operatorname{accel}(au^\circ))$ $\ldots \wedge \operatorname{vr}_4(\psi_e)$	

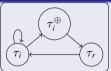
Leading Example

Start States:
$$\psi_s := x \le 0 \land y \le 0$$
 Error States: $\psi_e := y \ge 100$
Transition Formula: $\tau := \underbrace{(x < 100 \land x' = x + 1 \land y' = y)}_{\tau_i} \lor \underbrace{(x = 100 \land x' = 0 \land y' = y + 1)}_{\tau_r}$

$$\tau_i^{\oplus} := x + n \le 100 \land x' = x + n \land y' = y \land n > 0$$

ABMC

$$b \leftarrow 0$$
; $\operatorname{add}(\operatorname{vr}_b(\psi_s))$
if $\operatorname{check}() = \operatorname{unsat}$ then return safe
while \top do
 $\operatorname{push}()$; $\operatorname{add}(\operatorname{vr}_b(\psi_e))$
if $\operatorname{check}() = \operatorname{sat}$ then return $\vec{\tau}$
else
 $\operatorname{pop}()$; $\vec{\tau} \leftarrow \operatorname{trace}()$;
if $\vec{\tau}$ ends with loop $\vec{\tau}^{\circlearrowleft}$ then
 $\operatorname{add}(\operatorname{vr}_b(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\operatorname{add}(\operatorname{vr}_b(\tau))$



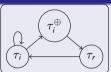
ь	$ec{ au}$	SMT Problem	Model
0		$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\dots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(\tau \vee \tau_i^{\oplus})$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge \operatorname{vr}_3(\tau \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ec{ au}^{\circlearrowleft}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_s := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_e := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r} \\ \tau_i^\oplus := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

ABMC

$$b \leftarrow 0$$
; $\operatorname{add}(\operatorname{vr}_b(\psi_s))$
if $\operatorname{check}() = \operatorname{unsat}$ then return safe
while \top do
 $\operatorname{push}()$; $\operatorname{add}(\operatorname{vr}_b(\psi_e))$
if $\operatorname{check}() = \operatorname{sat}$ then return $\vec{\tau}$
else
 $\operatorname{pop}()$; $\vec{\tau} \leftarrow \operatorname{trace}()$;
if $\vec{\tau}$ ends with loop $\vec{\tau}^{\circlearrowleft}$ then
 $\operatorname{add}(\operatorname{vr}_b(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\operatorname{add}(\operatorname{vr}_b(\tau))$



Ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1, \ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(\tau \vee \tau_i^{\oplus})$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0, y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(au \lor accel(ar{ au}^{\circlearrowleft}))$	$\ldots \cup [x_4/0, y_4/100]$
4		$\dots \wedge vr_4(\psi_e)$	

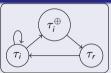
Leading Example

Start States:
$$\psi_s := x \le 0 \land y \le 0$$
 Error States: $\psi_e := y \ge 100$
Transition Formula: $\tau := \underbrace{(x < 100 \land x' = x + 1 \land y' = y)}_{\tau_i} \lor \underbrace{(x = 100 \land x' = 0 \land y' = y + 1)}_{\tau_r}$

$$\tau_i^{\oplus} := x + n \le 100 \land x' = x + n \land y' = y \land n > 0$$

ABMC

$$b \leftarrow 0$$
; add $(\mathsf{vr}_b(\psi_s))$
if $\mathsf{check}() = \mathsf{unsat}$ then return safe
while \top do
 $\mathsf{push}()$; add $(\mathsf{vr}_b(\psi_e))$
if $\mathsf{check}() = \mathsf{sat}$ then return $\vec{\tau}$
else
 $\mathsf{pop}()$; $\vec{\tau} \leftarrow \mathsf{trace}()$;
if $\vec{\tau}$ ends with loop $\vec{\tau}^{\circlearrowleft}$ then
 $\mathsf{add}(\mathsf{vr}_b(\tau \vee \mathsf{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\mathsf{add}(\mathsf{vr}_b(\tau))$



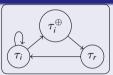
Ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1, \ldots/0]$
1	$[au_i]$	$\ldots \wedge \operatorname{vr}_1(\tau \vee \tau_i^{\oplus})$	$\dots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\dots \wedge \operatorname{vr}_2(\tau)$	\cup [$x_3/0, y_3/1$]
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(\tau \vee accel(\vec{\tau}^{\circlearrowleft}))$	$ \cup [x_4/0, y_4/100]$
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_s := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_e := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r} \\ \tau_i^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

ABMC

$$b \leftarrow 0$$
; $\operatorname{add}(\operatorname{vr}_b(\psi_s))$
if $\operatorname{check}() = \operatorname{unsat}$ then return safe
while \top do
 $\operatorname{push}()$; $\operatorname{add}(\operatorname{vr}_b(\psi_e))$
if $\operatorname{check}() = \operatorname{sat}$ then return $\vec{\tau}$
else
 $\operatorname{pop}()$; $\vec{\tau} \leftarrow \operatorname{trace}()$;
if $\vec{\tau}$ ends with loop $\vec{\tau}^{\circlearrowleft}$ then
 $\operatorname{add}(\operatorname{vr}_b(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\operatorname{add}(\operatorname{vr}_b(\tau))$



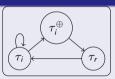
b	$ec{ au}$	SMT Problem	Model
0	[]	$vr_{0}(\psi_{s}) \wedge vr_{0}(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\dots \wedge \operatorname{vr}_2(\tau)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge \operatorname{vr}_3(\tau \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ar{ au}^{\circlearrowright}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\text{s}} := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_{i}} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

ABMC

$$b \leftarrow 0$$
; $\operatorname{add}(\operatorname{vr}_b(\psi_s))$
if $\operatorname{check}() = \operatorname{unsat}$ then return safe
while \top do
 $|\operatorname{push}()$; $\operatorname{add}(\operatorname{vr}_b(\psi_e))$
if $\operatorname{check}() = \operatorname{sat}$ then return $\vec{\tau}$
else
 $|\operatorname{pop}()$; $\vec{\tau} \leftarrow \operatorname{trace}()$;
if $\vec{\tau}$ ends with $loop$ $\vec{\tau}^{\circlearrowleft}$ then
 $|\operatorname{add}(\operatorname{vr}_b(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\operatorname{add}(\operatorname{vr}_b(\tau))$



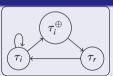
b	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1, \dots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(\tau \vee \tau_i^{\oplus})$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^{\oplus}]$	$\dots \wedge vr_2(\tau)$	\cup [$x_3/0, y_3/1$]
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(au \vee accel(ar{ au}^{\circlearrowleft}))$	$ \cup [x_4/0, y_4/100]$
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \wedge x' = x + 1 \wedge y' = y\right)}_{\tau_{i}} \vee \underbrace{\left(x = 100 \wedge x' = 0 \wedge y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

ABMC

$$\begin{array}{ll} b \leftarrow 0; & \mathsf{add}(\mathsf{vr}_b(\psi_s)) \\ \textbf{if} \; \mathsf{check}() = \mathsf{unsat} \; \textbf{then} \; \textbf{return} \; \mathsf{safe} \\ \textbf{while} \; \top \; \textbf{do} \\ & | \; \mathsf{push}(); \; \; \mathsf{add}(\mathsf{vr}_b(\psi_e)) \\ & \mathsf{if} \; \mathsf{check}() = \mathsf{sat} \; \textbf{then} \; \; \textbf{return} \; \vec{\tau} \\ & \mathsf{else} \\ & | \; \mathsf{pop}(); \; \; \vec{\tau} \leftarrow \mathsf{trace}(); \\ & \mathsf{if} \; \vec{\tau} \; \textit{ends} \; \textit{with} \; loop \; \vec{\tau}^\circlearrowleft \; \textbf{then} \\ & | \; \; \mathsf{add}(\mathsf{vr}_b(\tau \vee \mathsf{accel}(\vec{\tau}^\circlearrowleft))) \\ & \mathsf{else} \; \mathsf{add}(\mathsf{vr}_b(\tau)) \end{array}$$



Ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1, \dots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au \lor au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\dots \wedge vr_2(\tau)$	\cup [$x_3/0, y_3/1$]
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge \operatorname{vr}_3(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft}))$	$\dots \cup [x_4/0, y_4/100]$
4		$\ldots \wedge$ vr4 (ψ_e)	

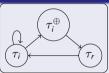
Leading Example

Start States:
$$\psi_s := x \le 0 \land y \le 0$$
 Error States: $\psi_e := y \ge 100$
Transition Formula: $\tau := \underbrace{(x < 100 \land x' = x + 1 \land y' = y)}_{\tau_i} \lor \underbrace{(x = 100 \land x' = 0 \land y' = y + 1)}_{\tau_r}$

$$\tau_i^{\oplus} := x + n \le 100 \land x' = x + n \land y' = y \land n > 0$$

ABMC

$$b \leftarrow 0$$
; add $(\mathsf{vr}_b(\psi_s))$
if $\mathsf{check}() = \mathsf{unsat}$ then return safe
while \top do
 $\mathsf{push}()$; add $(\mathsf{vr}_b(\psi_e))$
if $\mathsf{check}() = \mathsf{sat}$ then return $\vec{\tau}$
else
 $|\mathsf{pop}()$; $\vec{\tau} \leftarrow \mathsf{trace}()$;
if $\vec{\tau}$ ends with loop $\vec{\tau}^{\circlearrowleft}$ then
 $|\mathsf{add}(\mathsf{vr}_b(\tau \vee \mathsf{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\mathsf{add}(\mathsf{vr}_b(\tau))$
if $\mathsf{check}() = \mathsf{unsat}$ then return safe else $b++$

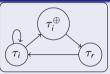


ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\dots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(\tau \vee \tau_i^{\oplus})$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0, y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge \operatorname{vr}_3(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft}))$	$\ldots \cup [x_4/0, y_4/100]$
4		$\dots \wedge \operatorname{Vr}_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \wedge x' = x + 1 \wedge y' = y\right)}_{\tau_{i}} \vee \underbrace{\left(x = 100 \wedge x' = 0 \wedge y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

$$b \leftarrow 0$$
; $\operatorname{add}(\operatorname{vr}_b(\psi_s))$
if $\operatorname{check}() = \operatorname{unsat}$ then return safe
while \top do
 $\operatorname{push}()$; $\operatorname{add}(\operatorname{vr}_b(\psi_e))$
if $\operatorname{check}() = \operatorname{sat}$ then return $\vec{\tau}$
else
 $|\operatorname{pop}(); \vec{\tau} \leftarrow \operatorname{trace}();$
if $\vec{\tau}$ ends with loop $\vec{\tau}^{\circlearrowleft}$ then
 $|\operatorname{add}(\operatorname{vr}_b(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\operatorname{add}(\operatorname{vr}_b(\tau))$
if $\operatorname{check}() = \operatorname{unsat}$ then return safe else $b++$

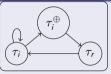


b	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\dots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au \lor au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\dots \wedge vr_2(\tau)$	\cup [$x_3/0, y_3/1$]
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge \operatorname{vr}_3(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft}))$	$ \cup [x_4/0, y_4/100]$
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \wedge x' = x + 1 \wedge y' = y\right)}_{\tau_{i}} \vee \underbrace{\left(x = 100 \wedge x' = 0 \wedge y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

$$b \leftarrow 0$$
; $\operatorname{add}(\operatorname{vr}_b(\psi_s))$
if $\operatorname{check}() = \operatorname{unsat}$ then return safe
while \top do
 $\operatorname{push}()$; $\operatorname{add}(\operatorname{vr}_b(\psi_e))$
if $\operatorname{check}() = \operatorname{sat}$ then return $\vec{\tau}$
else
 $\operatorname{pop}()$; $\vec{\tau} \leftarrow \operatorname{trace}()$;
if $\vec{\tau}$ ends with loop $\vec{\tau}^{\circlearrowleft}$ then
 $\operatorname{add}(\operatorname{vr}_b(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\operatorname{add}(\operatorname{vr}_b(\tau))$
if $\operatorname{check}() = \operatorname{unsat}$ then return safe else $b++$



b	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(\tau \vee \tau_i^{\oplus})$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(au \lor accel(\vec{ au}^{\circlearrowleft}))$	$ \cup [x_4/0, y_4/100]$
4		$\ldots \wedge vr_4(\psi_e)$	

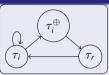
Leading Example

Start States:
$$\psi_s := x \le 0 \land y \le 0$$
 Error States: $\psi_e := y \ge 100$
Transition Formula: $\tau := \underbrace{(x < 100 \land x' = x + 1 \land y' = y)}_{\tau_i} \lor \underbrace{(x = 100 \land x' = 0 \land y' = y + 1)}_{\tau_r}$

$$\tau_i^{\oplus} := x + n \le 100 \land x' = x + n \land y' = y \land n > 0$$

ABMC

```
b \leftarrow 0; \operatorname{add}(\operatorname{vr}_b(\psi_s)) if \operatorname{check}() = \operatorname{unsat} then return safe while \top do |\operatorname{push}(); \operatorname{add}(\operatorname{vr}_b(\psi_e))| if \operatorname{check}() = \operatorname{sat} then return \vec{\tau} else |\operatorname{pop}(); \vec{\tau} \leftarrow \operatorname{trace}(); if \vec{\tau} ends with loop \vec{\tau}^{\circ} then |\operatorname{add}(\operatorname{vr}_b(\tau \vee \operatorname{accel}(\vec{\tau}^{\circ})))| else \operatorname{add}(\operatorname{vr}_b(\tau))
```



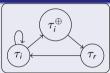
ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\dots/0]$
1	$[au_i]$	$\ldots \wedge \operatorname{vr}_1(\tau \vee \tau_i^{\oplus})$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ec{ au}^{\circlearrowleft}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \wedge x' = x + 1 \wedge y' = y\right)}_{\tau_{i}} \vee \underbrace{\left(x = 100 \wedge x' = 0 \wedge y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

ABMC

$$b \leftarrow 0$$
; add $(\mathsf{vr}_b(\psi_s))$
if $\mathsf{check}() = \mathsf{unsat}$ then return safe
while \top do
 $\mathsf{push}()$; add $(\mathsf{vr}_b(\psi_e))$
if $\mathsf{check}() = \mathsf{sat}$ then return $\vec{\tau}$
else
 $\mathsf{pop}()$; $\vec{\tau} \leftarrow \mathsf{trace}()$;
if $\vec{\tau}$ ends with loop $\vec{\tau}^{\circlearrowleft}$ then
 $\mathsf{add}(\mathsf{vr}_b(\tau \lor \mathsf{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\mathsf{add}(\mathsf{vr}_b(\tau))$



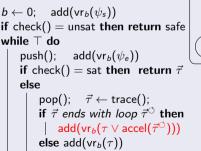
ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\dots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(\tau \vee \tau_i^{\oplus})$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ec{ au}^{\circlearrowleft}))$	
4		$\dots \wedge vr_4(\psi_e)$	

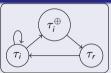
Leading Example

Start States:
$$\psi_s := x \le 0 \land y \le 0$$
 Error States: $\psi_e := y \ge 100$
Transition Formula: $\tau := \underbrace{(x < 100 \land x' = x + 1 \land y' = y)}_{\tau_i} \lor \underbrace{(x = 100 \land x' = 0 \land y' = y + 1)}_{\tau_r}$

$$\tau_i^{\oplus} := x + n \le 100 \land x' = x + n \land y' = y \land n > 0$$

ABMC





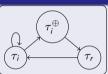
ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$ \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3		$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ec{ au}^{\circlearrowleft}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \wedge x' = x + 1 \wedge y' = y\right)}_{\tau_{i}} \vee \underbrace{\left(x = 100 \wedge x' = 0 \wedge y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

ABMC

$$b \leftarrow 0$$
; add $(\mathsf{vr}_b(\psi_s))$
if $\mathsf{check}() = \mathsf{unsat}$ then return safe
while \top do
 $|\mathsf{push}()$; add $(\mathsf{vr}_b(\psi_e))$
if $\mathsf{check}() = \mathsf{sat}$ then return $\vec{\tau}$
else
 $|\mathsf{pop}()$; $\vec{\tau} \leftarrow \mathsf{trace}()$;
if $\vec{\tau}$ ends with loop $\vec{\tau}^{\circlearrowleft}$ then
 $|\mathsf{add}(\mathsf{vr}_b(\tau \vee \mathsf{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\mathsf{add}(\mathsf{vr}_b(\tau))$



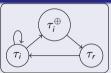
Ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ar{ au}^{\circlearrowright}))$	
4		$\ldots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \wedge x' = x + 1 \wedge y' = y\right)}_{\tau_{i}} \vee \underbrace{\left(x = 100 \wedge x' = 0 \wedge y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

ABMC

$$\begin{array}{l} b \leftarrow 0; \quad \operatorname{add}(\operatorname{vr}_b(\psi_s)) \\ \text{if } \operatorname{check}() = \operatorname{unsat then return safe} \\ \text{while } \top \text{ do} \\ & \quad \operatorname{push}(); \quad \operatorname{add}(\operatorname{vr}_b(\psi_e)) \\ & \quad \operatorname{if } \operatorname{check}() = \operatorname{sat then return } \vec{\tau} \\ & \quad \operatorname{else} \\ & \quad \mid \operatorname{pop}(); \quad \vec{\tau} \leftarrow \operatorname{trace}(); \\ & \quad \operatorname{if } \vec{\tau} \text{ ends with loop } \vec{\tau}^{\circlearrowleft} \text{ then} \\ & \quad \mid \operatorname{add}(\operatorname{vr}_b(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft}))) \\ & \quad \operatorname{else } \operatorname{add}(\operatorname{vr}_b(\tau)) \end{array}$$



ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\dots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ec{ au}^{\circlearrowleft}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_s := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_e := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r} \\ \tau_i^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

```
b \leftarrow 0; add(vr_b(\psi_s))

if check() = unsat then return safe

while \top do

push(); add(vr_b(\psi_e))

if check() = sat then return \vec{\tau}

else

pop(); \vec{\tau} \leftarrow trace();

if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then

add(vr_b(\tau \lor accel(\vec{\tau}^{\circlearrowleft})))

else add(vr_b(\tau))

if check() = unsat then return safe else b++
```

ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[\tau_i,\tau_i^\oplus]$	$\dots \wedge \operatorname{vr}_2(\tau)$	$\ldots \cup [x_3/0,y_3/1]$
3		$\dots \wedge \operatorname{vr}_3(\tau \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ar{ au}^{\circlearrowright}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\text{s}} := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_{i}} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

```
b \leftarrow 0; \quad \operatorname{add}(\operatorname{vr}_b(\psi_s)) if \operatorname{check}() = \operatorname{unsat} then return safe while \top do  | \operatorname{push}(); \quad \operatorname{add}(\operatorname{vr}_b(\psi_e))  if \operatorname{check}() = \operatorname{sat} then return \vec{\tau} else  | \operatorname{pop}(); \quad \vec{\tau} \leftarrow \operatorname{trace}();  if \vec{\tau} ends with loop \ \vec{\tau}^{\circlearrowleft} then  | \operatorname{add}(\operatorname{vr}_b(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft})))  else \operatorname{add}(\operatorname{vr}_b(\tau)) if \operatorname{check}() = \operatorname{unsat} then return safe else b++
```

Ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$\boxed{ [x_1/1,\ldots/0] }$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[\tau_i,\tau_i^\oplus]$	$\dots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3		$\dots \wedge \operatorname{vr}_3(\tau \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ec{ au}^{ riangle}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_s := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_e := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_i} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_r} \\ \tau_i^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

ABMC

```
b \leftarrow 0; \operatorname{add}(\operatorname{vr}_b(\psi_s))

if \operatorname{check}() = \operatorname{unsat} then return safe

while \top do

|\operatorname{push}(); \operatorname{add}(\operatorname{vr}_b(\psi_e))

if \operatorname{check}() = \operatorname{sat} then return \vec{\tau}

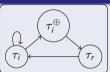
else

|\operatorname{pop}(); \vec{\tau} \leftarrow \operatorname{trace}();

if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then

|\operatorname{add}(\operatorname{vr}_b(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft})))

else \operatorname{add}(\operatorname{vr}_b(\tau))
```



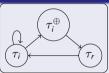
Ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\dots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[\tau_i,\tau_i^{\oplus}]$	$\dots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ec{ au}^{\circlearrowleft}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\text{s}} := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_{i}} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

ABMC

```
\begin{array}{l} b \leftarrow 0; \quad \mathsf{add}(\mathsf{vr}_b(\psi_s)) \\ \textbf{if} \; \mathsf{check}() = \mathsf{unsat} \; \textbf{then} \; \mathsf{return} \; \mathsf{safe} \\ \textbf{while} \; \top \; \textbf{do} \\ & \; \mathsf{push}(); \quad \mathsf{add}(\mathsf{vr}_b(\psi_e)) \\ & \; \mathsf{if} \; \mathsf{check}() = \mathsf{sat} \; \textbf{then} \; \; \mathsf{return} \; \vec{\tau} \\ & \; \mathsf{else} \\ & \; | \; \mathsf{pop}(); \quad \vec{\tau} \leftarrow \mathsf{trace}(); \\ & \; \mathsf{if} \; \vec{\tau} \; ends \; with \; loop \; \vec{\tau}^\circlearrowleft \; \textbf{then} \\ & \; | \; \; \mathsf{add}(\mathsf{vr}_b(\tau \vee \mathsf{accel}(\vec{\tau}^\circlearrowleft))) \\ & \; \; \mathsf{else} \; \mathsf{add}(\mathsf{vr}_b(\tau)) \end{array}
```



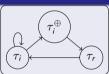
ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[\tau_i,\tau_i^{\oplus}]$	$\dots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3		$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ec{ au}^{\circlearrowleft}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

Start States:
$$\psi_s := x \leq 0 \land y \leq 0$$
 Error States: $\psi_e := y \geq 100$
Transition Formula: $\tau := \underbrace{(x < 100 \land x' = x + 1 \land y' = y)}_{\tau_i} \lor \underbrace{(x = 100 \land x' = 0 \land y' = y + 1)}_{\tau_r}$

$$\tau_i^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0$$

$$b \leftarrow 0$$
; add $(\mathsf{vr}_b(\psi_s))$
if $\mathsf{check}() = \mathsf{unsat}$ then return safe
while \top do
 $\mathsf{push}()$; add $(\mathsf{vr}_b(\psi_e))$
if $\mathsf{check}() = \mathsf{sat}$ then return $\vec{\tau}$
else
 $\mathsf{pop}()$; $\vec{\tau} \leftarrow \mathsf{trace}()$;
if $\vec{\tau}$ ends with loop $\vec{\tau}^{\circlearrowleft}$ then
 $\mathsf{add}(\mathsf{vr}_b(\tau \vee \mathsf{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\mathsf{add}(\mathsf{vr}_b(\tau))$
if $\mathsf{check}() = \mathsf{unsat}$ then return safe else $b++$

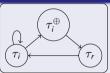


ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[\tau_i,\tau_i^{\oplus}]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0, y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ec{ au}^{\circlearrowleft}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \wedge x' = x + 1 \wedge y' = y\right)}_{\tau_{i}} \vee \underbrace{\left(x = 100 \wedge x' = 0 \wedge y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

```
b \leftarrow 0; add(vr<sub>b</sub>(\psi_s))
if check() = unsat then return safe
while \top do
    push(); add(vr_b(\psi_e))
    if check() = sat then return \vec{\tau}
     else
          pop(): \vec{\tau} \leftarrow \text{trace()}:
         if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then
               add(vr_b(\tau \vee accel(\vec{\tau}^{\circlearrowleft})))
          else add(vr_b(\tau))
     if check() = unsat then return safe else b++
```

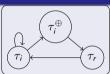


ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[\tau_i,\tau_i^{\oplus}]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0, y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ar{ au}^{\circlearrowright}))$	
4		$\ldots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\text{s}} := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_{i}} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

$$b \leftarrow 0$$
; add $(\mathsf{vr}_b(\psi_s))$
if $\mathsf{check}() = \mathsf{unsat}$ then return safe
while \top do
 $\mathsf{push}()$; add $(\mathsf{vr}_b(\psi_e))$
if $\mathsf{check}() = \mathsf{sat}$ then return $\vec{\tau}$
else
 $\mathsf{pop}()$; $\vec{\tau} \leftarrow \mathsf{trace}()$;
if $\vec{\tau}$ ends with loop $\vec{\tau}^{\circlearrowleft}$ then
 $\mathsf{add}(\mathsf{vr}_b(\tau \lor \mathsf{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\mathsf{add}(\mathsf{vr}_b(\tau))$
if $\mathsf{check}() = \mathsf{unsat}$ then return safe else $b++$



ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[\tau_i,\tau_i^{\oplus}]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3		$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ar{ au}^{\circlearrowright}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\text{s}} := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_{i}} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

ABMC

```
b \leftarrow 0; \operatorname{add}(\operatorname{vr}_b(\psi_s))

if \operatorname{check}() = \operatorname{unsat} then return safe

while \top do

\operatorname{push}(); \operatorname{add}(\operatorname{vr}_b(\psi_e))

if \operatorname{check}() = \operatorname{sat} then return \vec{\tau}

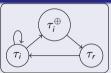
else

\operatorname{pop}(); \vec{\tau} \leftarrow \operatorname{trace}();

if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then

\operatorname{add}(\operatorname{vr}_b(\tau \vee \operatorname{accel}(\vec{\tau}^{\circlearrowleft})))

else \operatorname{add}(\operatorname{vr}_b(\tau))
```



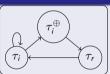
Ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[\tau_i,\tau_i^{\oplus}]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge \operatorname{vr}_3(\tau \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ec{ au}^{\circlearrowleft}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\text{s}} := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_{i}} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

ABMC

```
\begin{array}{l} b \leftarrow 0; \quad \mathsf{add}(\mathsf{vr}_b(\psi_s)) \\ \textbf{if } \mathsf{check}() = \mathsf{unsat} \ \textbf{then return safe} \\ \textbf{while} \ \top \ \textbf{do} \\ & | \quad \mathsf{push}(); \quad \mathsf{add}(\mathsf{vr}_b(\psi_e)) \\ & \quad \mathsf{if } \mathsf{check}() = \mathsf{sat} \ \textbf{then return} \ \vec{\tau} \\ & \quad \mathsf{else} \\ & | \quad \mathsf{pop}(); \quad \vec{\tau} \leftarrow \mathsf{trace}(); \\ & \quad \mathsf{if } \vec{\tau} \ \textit{ends with loop} \ \vec{\tau}^\circlearrowleft \ \textbf{then} \\ & | \quad \mathsf{add}(\mathsf{vr}_b(\tau \lor \mathsf{accel}(\vec{\tau}^\circlearrowleft))) \\ & \quad \mathsf{else} \ \mathsf{add}(\mathsf{vr}_b(\tau)) \end{array}
```



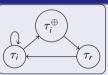
ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
	$[\tau_i,\tau_i^{\oplus}]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_{\it i}, au_{\it i}^{\oplus}, au_{\it r}]$	$\dots \wedge \operatorname{vr}_3(\tau \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ec{ au}^{\circlearrowleft}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

Start States:
$$\psi_s := x \leq 0 \land y \leq 0$$
 Error States: $\psi_e := y \geq 100$
Transition Formula: $\tau := \underbrace{(x < 100 \land x' = x + 1 \land y' = y)}_{\tau_i} \lor \underbrace{(x = 100 \land x' = 0 \land y' = y + 1)}_{\tau_r}$

$$\tau_i^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0$$

```
b \leftarrow 0; add(vr<sub>b</sub>(\psi_s))
if check() = unsat then return safe
while \top do
    push(); add(vr_b(\psi_e))
    if check() = sat then return \vec{\tau}
     else
          pop(): \vec{\tau} \leftarrow \text{trace()}:
          if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then
               add(vr_b(\tau \vee accel(\vec{\tau}^{\circlearrowleft})))
          else add(vr_b(\tau))
     if check() = unsat then return safe else b++
```



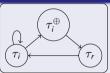
Ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[\tau_i,\tau_i^{\oplus}]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_{\it i}, au_{\it i}^{\oplus}, au_{\it r}]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
		$accel(ar{ au}^{\circlearrowright}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

Start States:
$$\psi_s := x \le 0 \land y \le 0$$
 Error States: $\psi_e := y \ge 100$
Transition Formula: $\tau := \underbrace{(x < 100 \land x' = x + 1 \land y' = y)}_{\tau_i} \lor \underbrace{(x = 100 \land x' = 0 \land y' = y + 1)}_{\tau_r}$

$$\tau_i^{\oplus} := x + n \le 100 \land x' = x + n \land y' = y \land n > 0$$

```
b \leftarrow 0; add(vr<sub>b</sub>(\psi_s))
if check() = unsat then return safe
while \top do
    push(); add(vr_b(\psi_e))
    if check() = sat then return \vec{\tau}
     else
          pop(): \vec{\tau} \leftarrow \text{trace()}:
          if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then
               add(vr_b(\tau \vee accel(\vec{\tau}^{\circlearrowleft})))
          else add(vr_b(\tau))
     if check() = unsat then return safe else b++
```

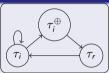


$ec{ au}$	SMT Problem	Model
[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
$[\tau_i,\tau_i^{\oplus}]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
$[au_{\it i}, au_{\it i}^\oplus, au_{\it r}]$	$\dots \wedge \operatorname{vr}_3(\tau \vee$	$\ldots \cup [x_4/0, y_4/100]$
$\vec{\tau}$	$accel(ec{ au}^{\circlearrowleft}))$	
	$\dots \wedge vr_4(\psi_e)$	
	$[au_i, au_i^\oplus]$	$ \begin{array}{ c c c }\hline [& vr_0(\psi_s) \wedge vr_0(\tau) \\ \hline [\tau_i] & \dots \wedge vr_1(\tau \vee \tau_i^\oplus) \\ \hline [\tau_i, \tau_i^\oplus] & \dots \wedge vr_2(\tau) \\ \hline [\tau_i, \tau_i^\oplus, \tau_r] & \dots \wedge vr_3(\tau \vee vc_1(\tau)) \\ \hline \end{array} $

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\text{s}} := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_{i}} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

```
b \leftarrow 0; add(vr<sub>b</sub>(\psi_s))
if check() = unsat then return safe
while \top do
    push(); add(vr_b(\psi_e))
    if check() = sat then return \vec{\tau}
     else
          pop(): \vec{\tau} \leftarrow \text{trace()}:
         if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then
               add(vr_b(\tau \lor accel(\vec{\tau}^{\circlearrowleft})))
          else add(vr_b(\tau))
     if check() = unsat then return safe else b++
```



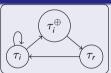
ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[\tau_i,\tau_i^{\oplus}]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0, y_3/1]$
3	$[au_{\it i}, au_{\it i}^{\oplus}, au_{\it r}]$	$\dots \wedge \operatorname{vr}_3(\tau \vee$	$\dots \cup [x_4/0, y_4/100]$
	$\vec{\tau}$	$accel(ec{ au}^{\circlearrowleft}))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

Start States:
$$\psi_s := x \le 0 \land y \le 0$$
 Error States: $\psi_e := y \ge 100$
Transition Formula: $\tau := \underbrace{(x < 100 \land x' = x + 1 \land y' = y)}_{\tau_i} \lor \underbrace{(x = 100 \land x' = 0 \land y' = y + 1)}_{\tau_r}$

$$\tau_i^{\oplus} := x + n \le 100 \land x' = x + n \land y' = y \land n > 0$$

```
b \leftarrow 0; add(vr<sub>b</sub>(\psi_s))
if check() = unsat then return safe
while \top do
    push(); add(vr_b(\psi_e))
    if check() = sat then return \vec{\tau}
     else
          pop(): \vec{\tau} \leftarrow \text{trace()}:
         if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then
               add(vr_b(\tau \lor accel(\vec{\tau}^{\circlearrowleft})))
          else add(vr_b(\tau))
     if check() = unsat then return safe else b++
```



Ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[\tau_i,\tau_i^{\oplus}]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_{\it i}, au_{\it i}^{\oplus}, au_{\it r}]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
	$\vec{\tau}$	$accel(ec{ au}^\circlearrowleft))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\text{s}} := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_{i}} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

ABMC

```
b \leftarrow 0; add(\mathsf{vr}_b(\psi_s))

if \mathsf{check}() = \mathsf{unsat} then return safe

while \top do

\mathsf{push}(); add(\mathsf{vr}_b(\psi_e))

if \mathsf{check}() = \mathsf{sat} then return \vec{\tau}

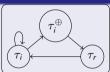
else

\mathsf{pop}(); \vec{\tau} \leftarrow \mathsf{trace}();

if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then

\mathsf{add}(\mathsf{vr}_b(\tau \lor \mathsf{accel}(\vec{\tau}^{\circlearrowleft})))

else \mathsf{add}(\mathsf{vr}_b(\tau))
```

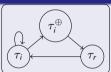


ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
	$\vec{\tau}$	$accel(ec{ au}^\circlearrowleft))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \wedge x' = x + 1 \wedge y' = y\right)}_{\tau_{i}} \vee \underbrace{\left(x = 100 \wedge x' = 0 \wedge y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

```
b \leftarrow 0; add(vr<sub>b</sub>(\psi_s))
if check() = unsat then return safe
while \top do
    push(); add(vr_b(\psi_e))
    if check() = sat then return \vec{\tau}
     else
          pop(): \vec{\tau} \leftarrow \text{trace()}:
         if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then
               add(vr_b(\tau \vee accel(\vec{\tau}^{\circlearrowleft})))
          else add(vr_b(\tau))
     if check() = unsat then return safe else b++
```



Ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$\boxed{ [x_1/1,\ldots/0] }$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[au_i, au_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_{\it i}, au_{\it i}^\oplus, au_{\it r}]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
	$\vec{\tau}$	$accel(ec{ au}^\circlearrowleft))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{\mathfrak{s}} := x \leq 0 \wedge y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \wedge x' = x + 1 \wedge y' = y\right)}_{\tau_{i}} \vee \underbrace{\left(x = 100 \wedge x' = 0 \wedge y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \wedge x' = x + n \wedge y' = y \wedge n > 0 \end{array}$$

ABMC

```
b \leftarrow 0; add(\mathsf{vr}_b(\psi_s))

if \mathsf{check}() = \mathsf{unsat} then return safe

while \top do

\mathsf{push}(); add(\mathsf{vr}_b(\psi_e))

if \mathsf{check}() = \mathsf{sat} then return \vec{\tau}

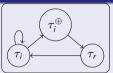
else

\mathsf{pop}(); \vec{\tau} \leftarrow \mathsf{trace}();

if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then

\mathsf{add}(\mathsf{vr}_b(\tau \vee \mathsf{accel}(\vec{\tau}^{\circlearrowleft})))

else \mathsf{add}(\mathsf{vr}_b(\tau))
```



ь	$ec{ au}$	SMT Problem	Model
0		$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[\tau_i,\tau_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_{\it i}, au_{\it i}^{\oplus}, au_{\it r}]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
	$\vec{\tau}$	$accel(ec{ au}^\circlearrowleft))$	
4		$\dots \wedge vr_4(\psi_e)$	

Leading Example

Start States:
$$\psi_s := x \leq 0 \land y \leq 0$$
 Error States: $\psi_e := y \geq 100$
Transition Formula: $\tau := \underbrace{(x < 100 \land x' = x + 1 \land y' = y)}_{\tau_i} \lor \underbrace{(x = 100 \land x' = 0 \land y' = y + 1)}_{\tau_r}$

$$\tau_i^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0$$

ABMC

```
b \leftarrow 0; add(\mathsf{vr}_b(\psi_s))

if \mathsf{check}() = \mathsf{unsat} then return safe

while \top do

\mathsf{push}(); add(\mathsf{vr}_b(\psi_e))

if \mathsf{check}() = \mathsf{sat} then return \vec{\tau}

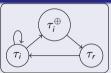
else

\mathsf{pop}(); \vec{\tau} \leftarrow \mathsf{trace}();

if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then

\mathsf{add}(\mathsf{vr}_b(\tau \vee \mathsf{accel}(\vec{\tau}^{\circlearrowleft})))

else \mathsf{add}(\mathsf{vr}_b(\tau))
```

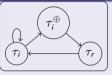


Ь	$ec{ au}$	SMT Problem	Model
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$
2	$[\tau_i,\tau_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$
3	$[au_{\it i}, au_{\it i}^{\oplus}, au_{\it r}]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$
	$\vec{\tau}$	$accel(ec{ au}^\circlearrowleft))$	
4		$\ldots \wedge vr_{4}(\psi_{e})$	

Leading Example

$$\begin{array}{ll} \text{Start States: } \psi_{s} := x \leq 0 \land y \leq 0 & \text{Error States: } \psi_{e} := y \geq 100 \\ \text{Transition Formula: } \tau := \underbrace{\left(x < 100 \land x' = x + 1 \land y' = y\right)}_{\tau_{i}} \lor \underbrace{\left(x = 100 \land x' = 0 \land y' = y + 1\right)}_{\tau_{r}} \\ \tau_{i}^{\oplus} := x + n \leq 100 \land x' = x + n \land y' = y \land n > 0 \end{array}$$

```
b \leftarrow 0; add(vr<sub>b</sub>(\psi_s))
if check() = unsat then return safe
while \top do
    push(); add(vr_b(\psi_e))
    if check() = sat then return \vec{\tau}
     else
          pop(): \vec{\tau} \leftarrow \text{trace()}:
         if \vec{\tau} ends with loop \vec{\tau}^{\circlearrowleft} then
               add(vr_b(\tau \vee accel(\vec{\tau}^{\circlearrowleft})))
          else add(vr_b(\tau))
     if check() = unsat then return safe else b++
```



Ь	$ec{ au}$	SMT Problem	Model	
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$	
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$	
2	$[\tau_i,\tau_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$	
3	$[au_i, au_i^\oplus, au_r]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$	
	$\vec{\tau}$	$accel(ec{ au}^\circlearrowleft))$		
4		$\ldots \wedge vr_{4}(\psi_{e})$		

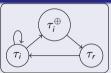
Leading Example

Start States:
$$\psi_s := x \le 0 \land y \le 0$$
 Error States: $\psi_e := y \ge 100$
Transition Formula: $\tau := \underbrace{(x < 100 \land x' = x + 1 \land y' = y)}_{\tau_i} \lor \underbrace{(x = 100 \land x' = 0 \land y' = y + 1)}_{\tau_r}$

$$\tau_i^{\oplus} := x + n \le 100 \land x' = x + n \land y' = y \land n > 0$$

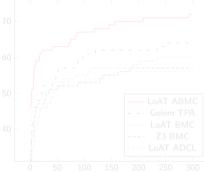
ABMC

$$b \leftarrow 0$$
; add $(\mathsf{vr}_b(\psi_s))$
if $\mathsf{check}() = \mathsf{unsat}$ then return safe
while \top do
 $\mathsf{push}()$; add $(\mathsf{vr}_b(\psi_e))$
if $\mathsf{check}() = \mathsf{sat}$ then return $\vec{\tau}$
else
 $\mathsf{pop}()$; $\vec{\tau} \leftarrow \mathsf{trace}()$;
if $\vec{\tau}$ ends with loop $\vec{\tau}^{\circlearrowleft}$ then
 $\mathsf{add}(\mathsf{vr}_b(\tau \lor \mathsf{accel}(\vec{\tau}^{\circlearrowleft})))$
else $\mathsf{add}(\mathsf{vr}_b(\tau))$



Ь	$ec{ au}$	SMT Problem	Model	
0	[]	$vr_0(\psi_s) \wedge vr_0(au)$	$[x_1/1,\ldots/0]$	
1	$[au_i]$	$\ldots \wedge vr_1(au ee au_i^\oplus)$	$\ldots \cup [x_2/100, y_2/0]$	
2	$[\tau_i,\tau_i^\oplus]$	$\ldots \wedge vr_2(au)$	$\ldots \cup [x_3/0,y_3/1]$	
3	$[au_{\it i}, au_{\it i}^{\oplus}, au_{\it r}]$	$\dots \wedge vr_3(au \vee$	$\ldots \cup [x_4/0, y_4/100]$	
	$\vec{\tau}$	$accel(ec{ au}^\circlearrowleft))$		
4		$\ldots \wedge vr_{4}(\psi_{e})$		

CHC-Comp '23	unsafe	safe
Benchmarks	√	√
LoAT ABMC	72	75
Golem TPA	64	83
LoAT BMC	60	36
Z3 BMC	57	21
LoAT ADCL	56	0
Golem BMC	55	20
Spacer (Z3)	51	151
Eldarica	46	107





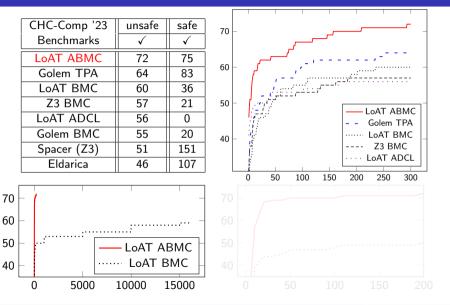




x: length of c.ex. y: unsat proofs

CHC-Comp '23 unsafe safe		70		
Benchmarks	√	✓] "	
LoAT ABMC	72	75		_,
Golem TPA	64	83	60	
LoAT BMC	60	36		J. 2007
Z3 BMC	57	21	50	LoAT ABMC
LoAT ADCL	56	0	1	LOAT ABMC
Golem BMC	55	20	20 151	LoAT BMC Z3 BMC
Spacer (Z3)	51	151		
Eldarica	46	107] [···· LoAT ADCL
			1	0 50 100 150 200 250 300

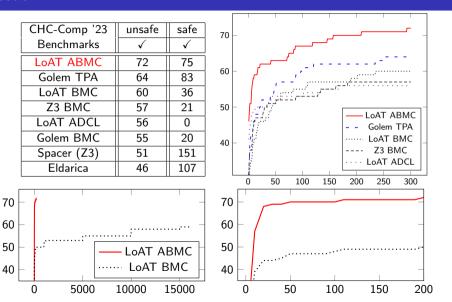
x: runtime in s y: unsat proofs



x: runtime in s v: unsat proofs

x: length of c.ex.

y: unsat proofs



- x: runtime in s
- y: unsat proofs

- x: length of c.ex.
- y: unsat proofs

See paper for...

- lazy exploration of dependency graph
- heuristics to fine-tune acceleration
- proving safety via Blocking Clauses

See paper for...

- lazy exploration of dependency graph
- heuristics to fine-tune acceleration
- proving safety via Blocking Clauses

See paper for...

- lazy exploration of dependency graph
- heuristics to fine-tune acceleration
- proving safety via Blocking Clauses

See paper for...

- lazy exploration of dependency graph
- heuristics to fine-tune acceleration
- proving safety via Blocking Clauses

See paper for...

- lazy exploration of dependency graph
- heuristics to fine-tune acceleration
- proving safety via Blocking Clauses

See paper for...

- lazy exploration of dependency graph
- heuristics to fine-tune acceleration
- proving safety via Blocking Clauses