Short proofs without interference

Adrián Rebola-Pardo

TU Wien, JKU Linz

Timisoara, Romania
23 September 2025

Supported by FWF 10.55776/COE12

Trying to merge proofs…

all unsat, over the same variables

F_1 F_2 F_3

Trying to merge proofs...

$u_1 \vee F_1$ $\qquad$ $u_2 \vee F_2$ $\qquad$ $u_3 \vee F_3$

Trying to merge proofs...

still unsat!

$$u_1 \lor F_1 \qquad u_2 \lor F_2 \qquad u_3 \lor F_3 \qquad \overline{u_1} \lor \overline{u_2} \lor \overline{u_3}$$

Trying to merge proofs...

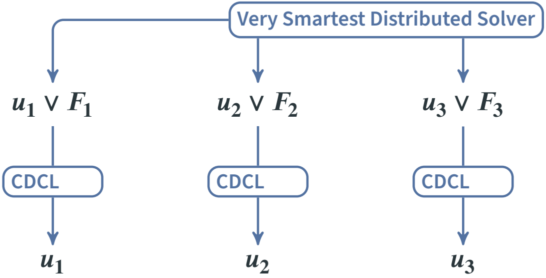
$u_1 \vee F_1$ $\quad u_2 \vee F_2$ $\quad u_3 \vee F_3$ $\quad \overline{u_1} \vee \overline{u_2} \vee \overline{u_3}$

Very Smartest Distributed Solver

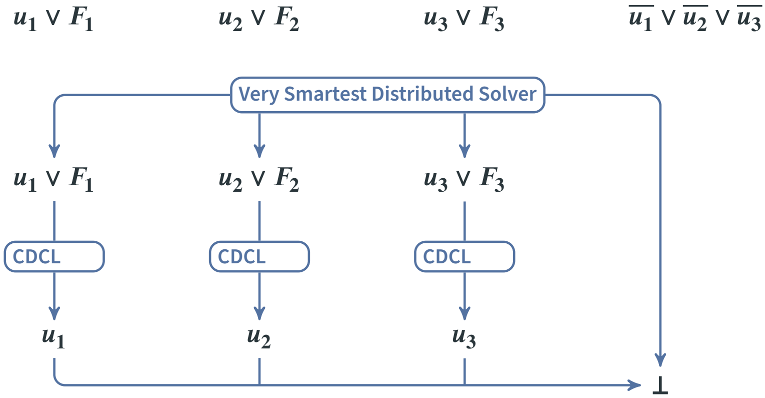
$u_1 \vee F_1$ $\quad u_2 \vee F_2$ $\quad u_3 \vee F_3$

Trying to merge proofs...

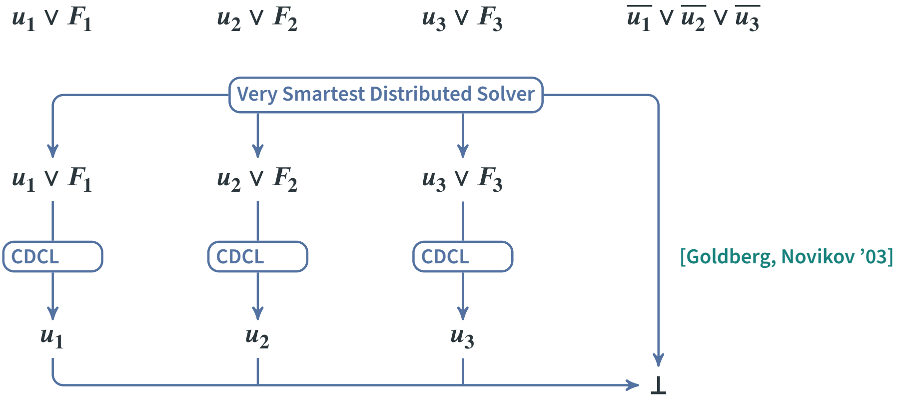
$u_1 \vee F_1$ $\quad$ $u_2 \vee F_2$ $\quad$ $u_3 \vee F_3$ $\quad$ $\overline{u_1} \vee \overline{u_2} \vee \overline{u_3}$



Trying to merge proofs...



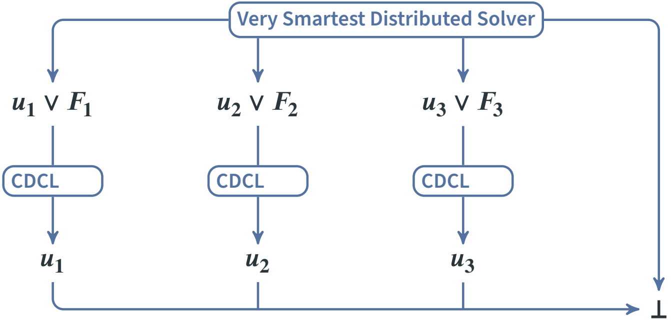
Trying to merge proofs...



$$\pi_1 : u_1 \lor F_1 \vdash u_1$$
$$\pi_2 : u_2 \lor F_2 \vdash u_2$$
$$\pi_3 : u_3 \lor F_3 \vdash u_3$$

Trying to merge proofs...

$u_1 \lor F_1 \qquad u_2 \lor F_2 \qquad u_3 \lor F_3 \qquad \overline{u_1} \lor \overline{u_2} \lor \overline{u_3}$



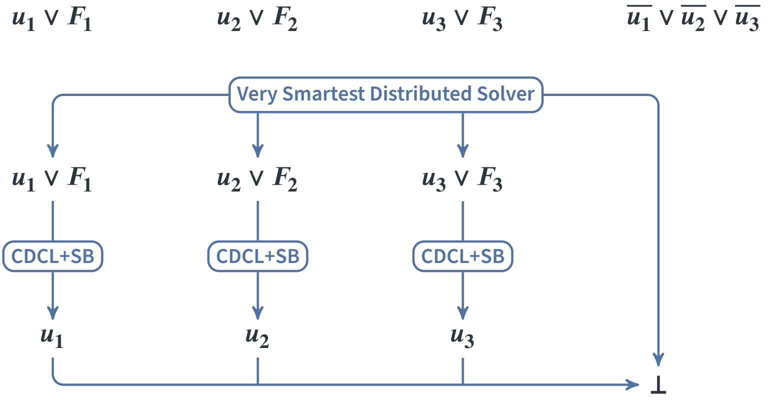
$\pi_1 : u_1 \lor F_1 \vdash u_1$

$\pi_2 : u_2 \lor F_2 \vdash u_2$

$\pi_3 : u_3 \lor F_3 \vdash u_3$

$\pi : u_1 \land u_2 \land u_3 \land (\overline{u_1} \lor \overline{u_2} \lor \overline{u_3}) \vdash \bot$

Trying to merge proofs...



$$\pi_1 : u_1 \lor F_1 \vdash u_1$$

$$\pi_2 : u_2 \lor F_2 \vdash u_2$$

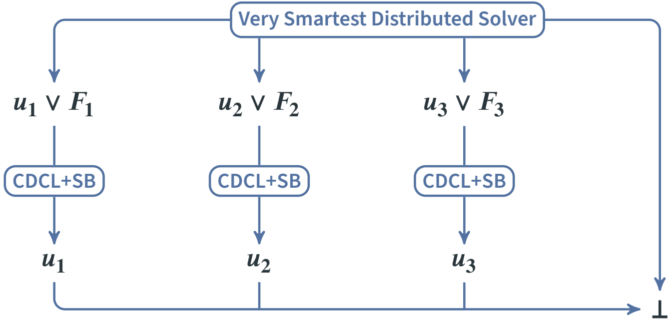
$$\pi_3 : u_3 \lor F_3 \vdash u_3$$

$$\pi : u_1 \land u_2 \land u_3 \land (\overline{u_1} \lor \overline{u_2} \lor \overline{u_3}) \vdash \perp$$

Trying to merge proofs...

$\sigma_1 = \{x_1 \leftrightarrow y_1\}$ $\sigma_2 = \{x_2 \leftrightarrow y_2\}$ $\sigma_3 = \{x_3 \leftrightarrow y_3\}$

$u_1 \vee F_1$ $u_2 \vee F_2$ $u_3 \vee F_3$ $\overline{u_1} \vee \overline{u_2} \vee \overline{u_3}$



$\pi_1 : u_1 \vee F_1 \vdash u_1$

$\pi_2 : u_2 \vee F_2 \vdash u_2$

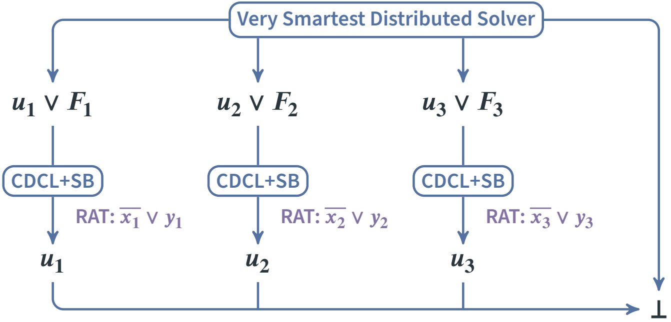
$\pi_3 : u_3 \vee F_3 \vdash u_3$

$\pi : u_1 \wedge u_2 \wedge u_3 \wedge (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3}) \vdash \perp$

Trying to merge proofs...

$\sigma_1 = \{x_1 \leftrightarrow y_1\}$ $\qquad$ $\sigma_2 = \{x_2 \leftrightarrow y_2\}$ $\qquad$ $\sigma_3 = \{x_3 \leftrightarrow y_3\}$

$u_1 \vee F_1$ $\qquad$ $u_2 \vee F_2$ $\qquad$ $u_3 \vee F_3$ $\qquad$ $\overline{u_1} \vee \overline{u_2} \vee \overline{u_3}$



[Heule, Hunt, Wezler ’15]

$\pi_1 : u_1 \vee F_1 \vdash u_1$

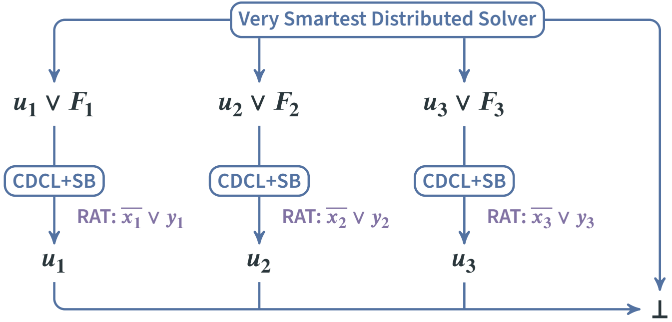
$\pi_2 : u_2 \vee F_2 \vdash u_2$ $\qquad$ $\pi : u_1 \wedge u_2 \wedge u_3 \wedge (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3}) \vdash \perp$

$\pi_3 : u_3 \vee F_3 \vdash u_3$

Trying to merge proofs...

$\sigma_1 = \{x_1 \leftrightarrow y_1\}$ $\quad$ $\sigma_2 = \{x_2 \leftrightarrow y_2\}$ $\quad$ $\sigma_3 = \{x_3 \leftrightarrow y_3\}$

$u_1 \vee F_1$ $\quad$ $u_2 \vee F_2$ $\quad$ $u_3 \vee F_3$ $\quad$ $\overline{u_1} \vee \overline{u_2} \vee \overline{u_3}$



[Järvisalo, Heule, Biere ’12]

$\pi_1 : u_1 \vee F_1 \vdash u_1$

$\pi_2 : u_2 \vee F_2 \vdash u_2$

$\pi_3 : u_3 \vee F_3 \vdash u_3$

$\pi : u_1 \wedge u$ … NOPE … $\vee \overline{u_3}) \vdash \perp$

… and failing because of interference

What are DRAT proofs really doing?

$\pi : F \vdash G$ proves that for each $I \models F$ we have $\mathrm{mut}(I) \models G$

… and failing because of interference

What are DRAT proofs really doing?

$\pi : F \vdash G$ proves that for each $I \models F$ we have $\mathrm{mut}(I) \models G$

mut is a sequence of operations like if $I \models T$, then $I := I \circ \sigma$ [RP, Suda '18]

… and failing because of interference

What are DRAT proofs really doing?

$\pi : F \vdash G$ proves that for each $I \models F$ we have $\mathrm{mut}(I) \models G$

mut is a sequence of operations like if $I \models T$, then $I := I \circ \sigma$ [RP, Suda '18]

each SR addition of C upon σ introduces a new operation with $T = \overline{C}$

… and failing because of interference

What are DRAT proofs really doing?

$\pi : F \vdash G$ proves that for each $I \vDash F$ we have $\mathrm{mut}(I) \vDash G$

mut is a sequence of operations like if $I \vDash T$, then $I := I \circ \sigma$ [RP, Suda '18]

each SR addition of C upon σ introduces a new operation with $T = \overline{C}$

$$\pi_1 : u_1 \vee F_1 \vdash (u_1 \vee F_1) \wedge (\overline{x_1} \vee y_1) \vdash u_1$$

… and failing because of interference

What are DRAT proofs really doing?

$\pi : F \vdash G$ proves that for each $I \vDash F$ we have $\mathrm{mut}(I) \vDash G$

mut is a sequence of operations like if $I \vDash T$, then $I := I \circ \sigma$ [RP, Suda '18]

each SR addition of C upon σ introduces a new operation with $T = \overline{C}$

$\pi_1 : u_1 \vee F_1 \vdash (u_1 \vee F_1) \wedge (\overline{x_1} \vee y_1) \vdash u_1$ $\qquad I \vDash u_1 \vee F_1 \Rightarrow \mathrm{mut}_1(I) \vDash u_1$

What are DRAT proofs really doing?

$\pi : F \vdash G$ proves that for each $I \models F$ we have $\mathrm{mut}(I) \models G$

mut is a sequence of operations like if $I \models T$, then $I := I \circ \sigma$ [RP, Suda '18]

each SR addition of C upon σ introduces a new operation with $T = \overline{C}$

$\pi_1 : u_1 \vee F_1 \vdash (u_1 \vee F_1) \wedge (\overline{x_1} \vee y_1) \vdash u_1$ $I \models u_1 \vee F_1 \Rightarrow \mathrm{mut}_1(I) \models u_1$

mut_1 is "if $I \models x_1 \wedge \overline{y_1}$ then $I := I \circ \{x_1 \leftrightarrow y_1\}$"

… and failing because of interference

What are DRAT proofs really doing?

$\pi : F \vdash G$ proves that for each $I \models F$ we have $\mathrm{mut}(I) \models G$

mut is a sequence of operations like if $I \models T$, then $I := I \circ \sigma$ [RP, Suda '18]

each SR addition of C upon σ introduces a new operation with $T = \overline{C}$

$$\pi_1 : u_1 \lor F_1 \vdash (u_1 \lor F_1) \land (\overline{x_1} \lor y_1) \vdash u_1 \qquad I \models u_1 \lor F_1 \Rightarrow \mathrm{mut}_1(I) \models u_1$$
$$\pi_2 : u_2 \lor F_2 \vdash (u_2 \lor F_2) \land (\overline{x_2} \lor y_2) \vdash u_2 \qquad I \models u_2 \lor F_2 \Rightarrow \mathrm{mut}_2(I) \models u_2$$
$$\pi_3 : u_3 \lor F_3 \vdash (u_3 \lor F_3) \land (\overline{x_3} \lor y_3) \vdash u_3 \qquad I \models u_3 \lor F_3 \Rightarrow \mathrm{mut}_3(I) \models u_3$$

… and failing because of interference

What are DRAT proofs really doing?

$\pi : F \vdash G$ proves that for each $I \models F$ we have $\mathrm{mut}(I) \models G$

mut is a sequence of operations like if $I \models T$, then $I := I \circ \sigma$ [RP, Suda '18]

each SR addition of C upon σ introduces a new operation with $T = \overline{C}$

$$\pi_1 : u_1 \lor F_1 \vdash (u_1 \lor F_1) \land (\overline{x_1} \lor y_1) \vdash u_1 \qquad I \models u_1 \lor F_1 \Rightarrow \quad I \models u_1$$

$$\pi_2 : u_2 \lor F_2 \vdash (u_2 \lor F_2) \land (\overline{x_2} \lor y_2) \vdash u_2 \qquad I \models u_2 \lor F_2 \Rightarrow \quad I \models u_2$$

$$\pi_3 : u_3 \lor F_3 \vdash (u_3 \lor F_3) \land (\overline{x_3} \lor y_3) \vdash u_3 \qquad I \models u_3 \lor F_3 \Rightarrow \quad I \models u_3$$

what we need is this!

… and failing because of interference

What are DRAT proofs really doing?

$\pi : F \vdash G$ proves that for each $I \vDash F$ we have $\mathrm{mut}(I) \vDash G$

mut is a sequence of operations like if $I \vDash T$, then $I := I \circ \sigma$ [RP, Suda '18]

each SR addition of C upon σ introduces a new operation with $T = \overline{C}$

$$\pi_1 : u_1 \lor F_1 \vdash (u_1 \lor F_1) \land (\overline{x_1} \lor y_1) \vdash u_1 \qquad I \vDash u_1 \lor F_1 \Rightarrow \mathrm{mut}_1(I) \vDash u_1$$
$$\pi_2 : u_2 \lor F_2 \vdash (u_2 \lor F_2) \land (\overline{x_2} \lor y_2) \vdash u_2 \qquad I \vDash u_2 \lor F_2 \Rightarrow \mathrm{mut}_2(I) \vDash u_2$$
$$\pi_3 : u_3 \lor F_3 \vdash (u_3 \lor F_3) \land (\overline{x_3} \lor y_3) \vdash u_3 \qquad I \vDash u_3 \lor F_3 \Rightarrow \mathrm{mut}_3(I) \vDash u_3$$

Interference-based proof systems force a single concurrent mut prefix…

... and failing because of interference

What are DRAT proofs really doing?

$\pi : F \vdash G$ proves that for each $I \models F$ we have $\mathrm{mut}(I) \models G$

mut is a sequence of operations like if $I \models T$, then $I := I \circ \sigma$ [RP, Suda '18]

each SR addition of C upon σ introduces a new operation with $T = \overline{C}$

$\pi_1 : u_1 \vee F_1 \vdash (u_1 \vee F_1) \wedge (\overline{x_1} \vee y_1) \vdash u_1$	$I \models u_1 \vee F_1 \Rightarrow \mathrm{mut}_1(I) \models u_1$
$\pi_2 : u_2 \vee F_2 \vdash (u_2 \vee F_2) \wedge (\overline{x_2} \vee y_2) \vdash u_2$	$I \models u_2 \vee F_2 \Rightarrow \mathrm{mut}_2(I) \models u_2$
$\pi_3 : u_3 \vee F_3 \vdash (u_3 \vee F_3) \wedge (\overline{x_3} \vee y_3) \vdash u_3$	$I \models u_3 \vee F_3 \Rightarrow \mathrm{mut}_3(I) \models u_3$

Interference-based proof systems force a single concurrent mut prefix...

... because of the accumulated formula

Kripke semantics for redundance rules

$\mathrm{mut}(I)$ is $I \circ \sigma$ if $I \vDash T$, or I otherwise.

Kripke semantics for redundance rules

$\mathrm{mut}(I)$ is $I \circ \sigma$ if $I \vDash T$, or I otherwise.

I maps variables to bits $\rightsquigarrow$ memory states

Kripke semantics for redundance rules

$\mathrm{mut}(I)$ is $I \circ \sigma$ if $I \vDash T$, or I otherwise.

I maps variables to bits $\rightsquigarrow$ memory states

$\mathrm{mut}(I)$ transforms a memory state into a memory state $\rightsquigarrow$ programs

Kripke semantics for redundance rules

$\mathrm{mut}(I)$ is $I \circ \sigma$ if $I \vDash T$, or I otherwise.

I maps variables to bits $\rightsquigarrow$ memory states

$\mathrm{mut}(I)$ transforms a memory state into a memory state $\rightsquigarrow$ programs

constraints semantics given by a set of (satisfying) assignments

$J \vDash C$

Kripke semantics for redundance rules

$\mathrm{mut}(I)$ is $I \circ \sigma$ if $I \vDash T$, or I otherwise.

I maps variables to bits $\rightsquigarrow$ memory states

$\mathrm{mut}(I)$ transforms a memory state into a memory state $\rightsquigarrow$ programs

constraints semantics given by a set of (satisfying) assignments

Programs semantics given by a binary relation of (transitioning) assignments

$J \vDash C$ $\qquad$ $I \otimes J \vDash \varepsilon$

Kripke semantics for redundance rules

$\mathrm{mut}(I)$ is $I \circ \sigma$ if $I \vDash T$, or I otherwise.

I maps variables to bits $\rightsquigarrow$ memory states

$\mathrm{mut}(I)$ transforms a memory state into a memory state $\rightsquigarrow$ programs

constraints semantics given by a set of (satisfying) assignments

Programs semantics given by a binary relation of (transitioning) assignments

Dynamic constraints

$$J \vDash C \qquad\qquad I \otimes J \vDash \varepsilon$$

Kripke semantics for redundance rules

$\mathrm{mut}(I)$ is $I \circ \sigma$ if $I \vDash T$, or I otherwise.

I maps variables to bits $\rightsquigarrow$ memory states

$\mathrm{mut}(I)$ transforms a memory state into a memory state $\rightsquigarrow$ programs

Static constraints semantics given by a set of (satisfying) assignments

Programs semantics given by a binary relation of (transitioning) assignments

Dynamic constraints

$$J \vDash C \qquad\qquad I \otimes J \vDash \varepsilon$$

Kripke semantics for redundance rules

$\mathrm{mut}(I)$ is $I \circ \sigma$ if $I \vDash T$, or I otherwise.

I maps variables to bits $\rightsquigarrow$ memory states

$\mathrm{mut}(I)$ transforms a memory state into a memory state $\rightsquigarrow$ programs

Static constraints semantics given by a set of (satisfying) assignments

Programs semantics given by a binary relation of (transitioning) assignments

Dynamic constraints a static constraint holds executing a program

$$I \vDash \varepsilon.C \quad \text{iff} \quad J \vDash C \text{ for all } J \text{ such that } I \otimes J \vDash \varepsilon$$

What’s in the box? Programs!

What's in the box? Programs!

Noop $I \otimes J \vDash 1$ iff $I = J$

What's in the box? Programs!

Noop $I \otimes J \vDash 1 \quad \text{iff} \quad I = J$

Composition $I \otimes J \vDash \varepsilon_1\, \varepsilon_2 \quad \text{iff} \quad \exists K,\ I \otimes K \vDash \varepsilon_1 \text{ and } K \otimes J \vDash \varepsilon_2$

What's in the box? Programs!

Noop $I \otimes J \vDash 1 \quad \text{iff} \quad I = J$

Composition $I \otimes J \vDash \varepsilon_1\, \varepsilon_2 \quad \text{iff} \quad \exists K,\ I \otimes K \vDash \varepsilon_1 \text{ and } K \otimes J \vDash \varepsilon_2$

Assignment $I \otimes J \vDash \langle \sigma \rangle \quad \text{iff} \quad J = I \circ \sigma$

What's in the box? Programs!

Noop $I \otimes J \vDash 1 \quad \text{iff} \quad I = J$

Composition $I \otimes J \vDash \varepsilon_1\, \varepsilon_2 \quad \text{iff} \quad \exists K,\ I \otimes K \vDash \varepsilon_1 \text{ and } K \otimes J \vDash \varepsilon_2$

Assignment $I \otimes J \vDash \langle \sigma \rangle \quad \text{iff} \quad J = I \circ \sigma$

Choice $I \otimes J \vDash \varepsilon_1 \sqcup \varepsilon_2 \quad \text{iff} \quad I \otimes J \vDash \varepsilon_1 \text{ or } I \otimes J \vDash \varepsilon_2$

What's in the box? Programs!

Noop $I \otimes J \vDash 1 \quad \text{iff} \quad I = J$

Composition $I \otimes J \vDash \varepsilon_1\, \varepsilon_2 \quad \text{iff} \quad \exists K,\ I \otimes K \vDash \varepsilon_1 \text{ and } K \otimes J \vDash \varepsilon_2$

Assignment $I \otimes J \vDash \langle \sigma \rangle \quad \text{iff} \quad J = I \circ \sigma$

Choice $I \otimes J \vDash \varepsilon_1 \sqcup \varepsilon_2 \quad \text{iff} \quad I \otimes J \vDash \varepsilon_1 \text{ or } I \otimes J \vDash \varepsilon_2$

Test $I \otimes J \vDash T? \quad \text{iff} \quad I = J \text{ and } I \vDash T$

What's in the box? Programs!

Noop $I \otimes J \models 1$ iff $I = J$

Composition $I \otimes J \models \varepsilon_1\, \varepsilon_2$ iff $\exists K,\ I \otimes K \models \varepsilon_1$ and $K \otimes J \models \varepsilon_2$

Assignment $I \otimes J \models \langle \sigma \rangle$ iff $J = I \circ \sigma$

Choice $I \otimes J \models \varepsilon_1 \sqcup \varepsilon_2$ iff $I \otimes J \models \varepsilon_1$ or $I \otimes J \models \varepsilon_2$

Test $I \otimes J \models T?$ iff $I = J$ and $I \models T$

Branch $I \otimes J \models \nabla(T : \varepsilon_1 \parallel \varepsilon_0)$ iff $I \otimes J \models (T?\, \varepsilon_1) \sqcup (\overline{T}?\, \varepsilon_0)$

What's in the box? Programs!

Noop $I \otimes J \vDash 1 \quad \text{iff} \quad I = J$

Composition $I \otimes J \vDash \varepsilon_1\, \varepsilon_2 \quad \text{iff} \quad \exists K,\ I \otimes K \vDash \varepsilon_1 \text{ and } K \otimes J \vDash \varepsilon_2$

Assignment $I \otimes J \vDash \langle \sigma \rangle \quad \text{iff} \quad J = I \circ \sigma$

Choice $I \otimes J \vDash \varepsilon_1 \sqcup \varepsilon_2 \quad \text{iff} \quad I \otimes J \vDash \varepsilon_1 \text{ or } I \otimes J \vDash \varepsilon_2$

Test $I \otimes J \vDash T? \quad \text{iff} \quad I = J \text{ and } I \vDash T$

Branch $I \otimes J \vDash \nabla(T : \varepsilon_1 \parallel \varepsilon_0) \quad \text{iff} \quad I \otimes J \vDash (T?\, \varepsilon_1) \sqcup (\overline{T}?\, \varepsilon_0)$

Not even a new thing! [Fischer, Ladner '79] [Balbiani, Herzig, Troquard '13]

Propositional dynamic logic (PDL) defines modalities for each program

What's in the box? Programs!

Noop $I \otimes J \vDash 1$ iff $I = J$

Composition $I \otimes J \vDash \varepsilon_1\, \varepsilon_2$ iff $\exists K,\ I \otimes K \vDash \varepsilon_1$ and $K \otimes J \vDash \varepsilon_2$

Assignment $I \otimes J \vDash \langle \sigma \rangle$ iff $J = I \circ \sigma$

Choice $I \otimes J \vDash \varepsilon_1 \sqcup \varepsilon_2$ iff $I \otimes J \vDash \varepsilon_1$ or $I \otimes J \vDash \varepsilon_2$

Test $I \otimes J \vDash T?$ iff $I = J$ and $I \vDash T$

Branch $I \otimes J \vDash \nabla(T : \varepsilon_1 \parallel \varepsilon_0)$ iff $I \otimes J \vDash (T?\, \varepsilon_1) \sqcup (\overline{T}?\, \varepsilon_0)$

Not even a new thing! [Fischer, Ladner '79] [Balbiani, Herzig, Troquard '13]

Propositional dynamic logic (PDL) defines modalities for each program

Necessitation law (a.k.a. I can apply programs to a proof)

If $F \vDash G$ holds, then $\varepsilon.F \vDash \varepsilon.G$ holds too

RUP-like inferences in dynamic logic

Unit propagation (BCP) a *blazingly* fast sound but incomplete algorithm for inconsistency in CNF formulas [Zhang, Madigan, Moskewicz, Malik '01]

RUP-like inferences in dynamic logic

Unit propagation (BCP) a *blazingly* fast sound but incomplete algorithm for inconsistency in CNF formulas [Zhang, Madigan, Moskewicz, Malik '01]

Reverse unit propagation (RUP) a *blazingly* fast sound but incomplete algorithm for implication in CNF formulas [Goldberg, Novikov '03]

RUP-like inferences in dynamic logic

Unit propagation (BCP) a *blazingly* fast sound but incomplete algorithm for inconsistency in CNF formulas [Zhang, Madigan, Moskewicz, Malik ’01]

Reverse unit propagation (RUP) a *blazingly* fast sound but incomplete algorithm for implication in CNF formulas [Goldberg, Novikov ’03]

$$F \wedge \overline{C} \models \bot \qquad \text{iff} \qquad F \models C$$

RUP-like inferences in dynamic logic

Unit propagation (BCP) a *blazingly* fast sound but incomplete algorithm for inconsistency in CNF formulas [Zhang, Madigan, Moskewicz, Malik ’01]

Reverse unit propagation (RUP) a *blazingly* fast sound but incomplete algorithm for implication in CNF formulas [Goldberg, Novikov ’03]

Can we have this for dynamic formulas? yes, but not built on consistency.

$$F \land \overline{C} \models \bot \quad \text{iff} \quad F \models C$$

RUP-like inferences in dynamic logic

Unit propagation (BCP) a *blazingly* fast sound but incomplete algorithm for inconsistency in CNF formulas [Zhang, Madigan, Moskewicz, Malik ’01]

Reverse unit propagation (RUP) a *blazingly* fast sound but incomplete algorithm for implication in CNF formulas [Goldberg, Novikov ’03]

Can we have this for dynamic formulas? yes, but not built on consistency.

$$F \wedge \overline{\varepsilon.C} \models \bot \quad \text{iff} \quad F \models \varepsilon.C$$

I don’t have diamonds!

RUP-like inferences in dynamic logic

Unit propagation (BCP) a *blazingly* fast sound but incomplete algorithm for inconsistency in CNF formulas [Zhang, Madigan, Moskewicz, Malik ’01]

Reverse unit propagation (RUP) a *blazingly* fast sound but incomplete algorithm for implication in CNF formulas [Goldberg, Novikov ’03]

Can we have this for dynamic formulas? yes, but not built on consistency.

Instead we reduce a dynamic implication check to (several) RUP checks

$F \wedge \overline{\varepsilon.C} \models \bot$ iff $F \models \varepsilon.C$ I don’t have diamonds!

Look Ma! No interference!

$$F = (u_1 \vee F_1) \wedge (u_2 \vee F_2) \wedge (u_3 \vee F_3) \wedge (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Look Ma! No interference!

$$F = (u_1 \lor F_1) \land (u_2 \lor F_2) \land (u_3 \lor F_3) \land (\overline{u_1} \lor \overline{u_2} \lor \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \lor y_1$ in $u_1 \lor F_1$

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

symmetry: $\sigma_1 = \{x_1 \leftrightarrow y_1\}$

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

symmetry: $\sigma_1 = \{x_1 \leftrightarrow y_1\}$ program: $\nabla(B_1 : 1 \parallel \langle\sigma_1\rangle)$

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

symmetry: $\sigma_1 = \{x_1 \leftrightarrow y_1\}$ program: $\nabla(B_1 \,:\, 1 \parallel \langle \sigma_1 \rangle)$

$$F \vdash \nabla(B_1 \,:\, 1 \parallel \langle \sigma_1 \rangle).\, B_1$$

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

symmetry: $\sigma_1 = \{x_1 \leftrightarrow y_1\}$ program: $\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle)$

$$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).\, B_1$$

$$F \wedge B_1 \vdash 1.\, B_1 \qquad\qquad F \wedge \overline{B_1} \vdash \langle \sigma_1 \rangle.\, B_1$$

Look Ma! No interference!

$$F = (u_1 \lor F_1) \land (u_2 \lor F_2) \land (u_3 \lor F_3) \land (\overline{u_1} \lor \overline{u_2} \lor \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \lor y_1$ in $u_1 \lor F_1$

symmetry: $\sigma_1 = \{x_1 \leftrightarrow y_1\}$ program: $\nabla(B_1 : 1 \parallel \langle\sigma_1\rangle)$

$$F \vdash \nabla(B_1 : 1 \parallel \langle\sigma_1\rangle).\, B_1$$

$$F \land B_1 \vdash 1.\, B_1 \qquad F \land \overline{B_1} \vdash \langle\sigma_1\rangle.\, B_1$$

$$F \land B_1 \vdash B_1$$

trivial by inclusion

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

symmetry: $\sigma_1 = \{x_1 \leftrightarrow y_1\}$ program: $\nabla(B_1 \,:\, 1 \parallel \langle \sigma_1 \rangle)$

$$F \vdash \nabla(B_1 \,:\, 1 \parallel \langle \sigma_1 \rangle).\, B_1$$

$$F \wedge B_1 \vdash 1.\, B_1 \qquad\qquad F \wedge \overline{B_1} \vdash \langle \sigma_1 \rangle.\, B_1$$

$$F \wedge B_1 \vdash B_1 \qquad\qquad F \wedge \overline{B_1} \vdash B_1\big|_{\sigma_1}$$

trivial by inclusion holds by RUP

Look Ma! No interference!

$$F = (u_1 \vee F_1) \wedge (u_2 \vee F_2) \wedge (u_3 \vee F_3) \wedge (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

symmetry: $\sigma_1 = \{x_1 \leftrightarrow y_1\}$ program: $\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle)$

$$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).\, B_1$$

$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).\, C$ for $C \in u_1 \vee F_1$

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

symmetry: $\sigma_1 = \{x_1 \leftrightarrow y_1\}$ program: $\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle)$

$$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).\, B_1$$

$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).\, C$ for $C \in u_1 \vee F_1$

$F \wedge B_1 \vdash 1.\, C$ $\qquad$ $F \wedge \overline{B_1} \vdash \langle \sigma_1 \rangle.\, C$

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

symmetry: $\sigma_1 = \{x_1 \leftrightarrow y_1\}$ program: $\nabla(B_1 : 1 \parallel \langle\sigma_1\rangle)$

$$F \vdash \nabla(B_1 : 1 \parallel \langle\sigma_1\rangle).\, B_1$$

$$F \vdash \nabla(B_1 : 1 \parallel \langle\sigma_1\rangle).\, C \quad \text{for } C \in u_1 \vee F_1$$

$$F \wedge B_1 \vdash 1.\, C \qquad\qquad F \wedge \overline{B_1} \vdash \langle\sigma_1\rangle.\, C$$

$$F \wedge B_1 \vdash C$$

trivial by inclusion

Look Ma! No interference!

$$F \quad = \quad (u_1 \lor F_1) \quad \land \quad (u_2 \lor F_2) \quad \land \quad (u_3 \lor F_3) \quad \land \quad (\overline{u_1} \lor \overline{u_2} \lor \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \lor y_1$ in $u_1 \lor F_1$

symmetry: $\sigma_1 = \{x_1 \leftrightarrow y_1\}$ program: $\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle)$

$$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).\, B_1$$

$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).\, C$ for $C \in u_1 \lor F_1$

$$F \land B_1 \vdash 1.\, C \qquad F \land \overline{B_1} \vdash \langle \sigma_1 \rangle.\, C$$

$$F \land B_1 \vdash C \qquad F \land \overline{B_1} \vdash C|_{\sigma_1}$$

trivial by inclusion holds by RUP (because of symmetry)

Look Ma! No interference!

$$F \quad = \quad (u_1 \lor F_1) \quad \land \quad (u_2 \lor F_2) \quad \land \quad (u_3 \lor F_3) \quad \land \quad (\overline{u_1} \lor \overline{u_2} \lor \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \lor y_1$ in $u_1 \lor F_1$

$F \vdash \nabla(B_1 : 1 \parallel \langle\sigma_1\rangle).(u_1 \lor F_1) \land B_1$

symmetry: $\sigma_1 = \{x_1 \leftrightarrow y_1\}$ program: $\nabla(B_1 : 1 \parallel \langle\sigma_1\rangle)$

$$F \vdash \nabla(B_1 : 1 \parallel \langle\sigma_1\rangle).B_1$$

$F \vdash \nabla(B_1 : 1 \parallel \langle\sigma_1\rangle).C$ for $C \in u_1 \lor F_1$

$F \land B_1 \vdash 1.C$ $F \land \overline{B_1} \vdash \langle\sigma_1\rangle.C$

$F \land B_1 \vdash C$ $F \land \overline{B_1} \vdash C|_{\sigma_1}$

trivial by inclusion holds by RUP (because of symmetry)

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).(u_1 \vee F_1) \wedge B_1$

Step 2 derive u_1 from $(u_1 \vee F_1) \wedge B_1$

Look Ma! No interference!

$$F \quad = \quad (u_1 \lor F_1) \quad \land \quad (u_2 \lor F_2) \quad \land \quad (u_3 \lor F_3) \quad \land \quad (\overline{u_1} \lor \overline{u_2} \lor \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \lor y_1$ in $u_1 \lor F_1$

$F \vdash \nabla(B_1 \,:\, 1 \,\|\, \langle \sigma_1 \rangle).\, (u_1 \lor F_1) \land B_1$

Step 2 derive u_1 from $(u_1 \lor F_1) \land B_1$

$(u_1 \lor F_1) \land B_1 \vdash u_1$ by a bunch of RUP steps

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle). (u_1 \vee F_1) \wedge B_1$

Step 2 derive u_1 from $(u_1 \vee F_1) \wedge B_1$

$(u_1 \vee F_1) \wedge B_1 \vdash u_1$ by a bunch of RUP steps

we haven't derived $(u_1 \vee F_1) \wedge B_1$ though...

Look Ma! No interference!

$$F \quad = \quad (u_1 \lor F_1) \quad \land \quad (u_2 \lor F_2) \quad \land \quad (u_3 \lor F_3) \quad \land \quad (\overline{u_1} \lor \overline{u_2} \lor \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \lor y_1$ in $u_1 \lor F_1$

$F \vdash \nabla(B_1 \,:\, 1 \parallel \langle \sigma_1 \rangle).\, (u_1 \lor F_1) \land B_1$

Step 2 derive u_1 from $(u_1 \lor F_1) \land B_1$

$(u_1 \lor F_1) \land B_1 \vdash u_1$ by a bunch of RUP steps

we haven't derived $(u_1 \lor F_1) \land B_1$ though...

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).\,(u_1 \vee F_1) \wedge B_1$

Step 2 derive u_1 from $(u_1 \vee F_1) \wedge B_1$

$(u_1 \vee F_1) \wedge B_1 \vdash u_1$ by a bunch of RUP steps

we haven't derived $(u_1 \vee F_1) \wedge B_1$ though...

but we have necessitation!

$$\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).\,(u_1 \vee F_1) \wedge B_1 \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).u_1$$

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

$F \vdash \nabla(B_1 \,:\, 1 \parallel \langle \sigma_1 \rangle).\,(u_1 \vee F_1) \wedge B_1$

Step 2 derive u_1 from $(u_1 \vee F_1) \wedge B_1$

$\nabla(B_1 \,:\, 1 \parallel \langle \sigma_1 \rangle).\,(u_1 \vee F_1) \wedge B_1 \vdash \nabla(B_1 \,:\, 1 \parallel \langle \sigma_1 \rangle).u_1$

Step 3 derive u_1 from $u_1 \vee F_1$

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).(u_1 \vee F_1) \wedge B_1$

Step 2 derive u_1 from $(u_1 \vee F_1) \wedge B_1$

$\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).(u_1 \vee F_1) \wedge B_1 \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).u_1$

Step 3 derive u_1 from $u_1 \vee F_1$

$$\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).u_1 \vdash u_1$$

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).\,(u_1 \vee F_1) \wedge B_1$

Step 2 derive u_1 from $(u_1 \vee F_1) \wedge B_1$

$\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).\,(u_1 \vee F_1) \wedge B_1 \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).u_1$

Step 3 derive u_1 from $u_1 \vee F_1$

$$\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).u_1 \vdash u_1$$

$$B_1 \wedge (1.\,u_1) \vdash u_1 \qquad\qquad \overline{B_1} \wedge (\langle \sigma_1 \rangle.\,u_1) \vdash u_1$$

Look Ma! No interference!

$$F \quad = \quad (u_1 \lor F_1) \quad \land \quad (u_2 \lor F_2) \quad \land \quad (u_3 \lor F_3) \quad \land \quad (\overline{u_1} \lor \overline{u_2} \lor \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \lor y_1$ in $u_1 \lor F_1$

$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).\, (u_1 \lor F_1) \land B_1$

Step 2 derive u_1 from $(u_1 \lor F_1) \land B_1$

$\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).\, (u_1 \lor F_1) \land B_1 \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).u_1$

Step 3 derive u_1 from $u_1 \lor F_1$

$$\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).u_1 \vdash u_1$$

$B_1 \land (1.\, u_1) \vdash u_1$

$\overline{B_1} \land (\langle \sigma_1 \rangle.\, u_1) \vdash u_1$

$B_1 \land u_1 \vdash u_1$

trivial by inclusion

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).(u_1 \vee F_1) \wedge B_1$

Step 2 derive u_1 from $(u_1 \vee F_1) \wedge B_1$

$\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).(u_1 \vee F_1) \wedge B_1 \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).u_1$

Step 3 derive u_1 from $u_1 \vee F_1$

$$\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).u_1 \vdash u_1$$

$B_1 \wedge (1.u_1) \vdash u_1$	$\overline{B_1} \wedge (\langle \sigma_1 \rangle.u_1) \vdash u_1$
$B_1 \wedge u_1 \vdash u_1$	$B_1 \wedge u_1 \vdash u_1\big\vert_{\sigma_1}$
trivial by inclusion	holds by RUP (because of cleanliness)

[Fazekas, Biere, Scholl ’19]

Look Ma! No interference!

$$F \quad = \quad (u_1 \vee F_1) \quad \wedge \quad (u_2 \vee F_2) \quad \wedge \quad (u_3 \vee F_3) \quad \wedge \quad (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3})$$

Step 1 introduce the symmetry breaker $B_1 = \overline{x_1} \vee y_1$ in $u_1 \vee F_1$

$F \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).(u_1 \vee F_1) \wedge B_1$

Step 2 derive u_1 from $(u_1 \vee F_1) \wedge B_1$

$\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).(u_1 \vee F_1) \wedge B_1 \vdash \nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).u_1$

Step 3 derive u_1 from $u_1 \vee F_1$

$\nabla(B_1 : 1 \parallel \langle \sigma_1 \rangle).u_1 \vdash u_1$

Step 4 repeat for u_2 and u_3, then resolve

Conclusion

You don’t need interference for RAT (or PR, or SR, or WSR)

You don’t even need an accumulated formula

You can get new reasoning tools if you stop thinking in terms of redundancy notions

Proof logging and checking is not necessarily more complex