A Coalgebraic Decision Procedure for WS1S

Dmitriy Traytel

Technische Universität München

A Coalgebraic Decision Procedure for WS1S

Dmitriy Traytel

Logic-Automaton Connection

WS1S

$$\mathsf{T} \mid \mathsf{F} \mid x \in X \mid x < y \mid \varphi \vee \psi \mid \neg\varphi \mid \exists x.\, \varphi \mid \exists \underset{\substack{\nwarrow \\ \text{finite}}}{X}.\, \varphi$$

Logic-Automaton Connection

WS1S

$$\mathrm{T} \mid \mathrm{F} \mid x \in X \mid x < y \mid \varphi \vee \psi \mid \neg\varphi \mid \mathrm{FO}\ x \mid \exists \underset{\substack{\uparrow \\ \text{finite}}}{X}.\ \varphi$$

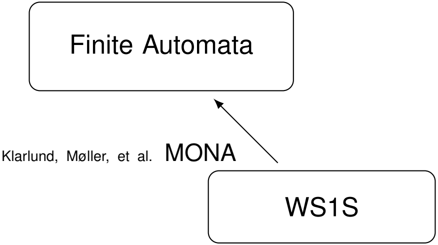
Logic-Automaton Connection

WS1S

$\forall \mathrm{I}.\ \ \mathrm{I} \models \varphi \iff \mathrm{I} \models \psi?$

$$\mathsf{T} \mid \mathsf{F} \mid x \in X \mid x < y \mid \varphi \vee \psi \mid \neg\varphi \mid \mathsf{FO}\ x \mid \exists \underset{\substack{\uparrow \\ \text{finite}}}{X}.\ \varphi$$

Logic-Automaton Connection



Finite Automata

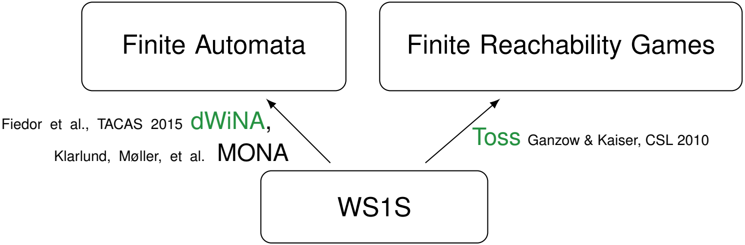
Klarlund, Møller, et al. MONA

WS1S

$\forall \mathrm{I}.\ \ \mathrm{I} \models \varphi \iff \mathrm{I} \models \psi ?$

$$\mathrm{T} \mid \mathrm{F} \mid x \in X \mid x < y \mid \varphi \vee \psi \mid \neg\varphi \mid \mathrm{FO}\ x \mid \exists \underset{\overset{\uparrow}{\text{finite}}}{X} .\, \varphi$$

Logic-Automaton Connection



Finite Automata

Finite Reachability Games

Fiedor et al., TACAS 2015 dWiNA,

Klarlund, Møller, et al. MONA

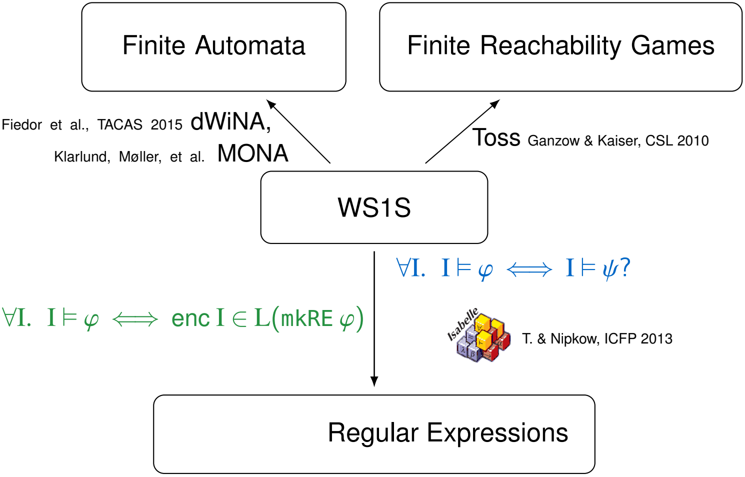
Toss Ganzow & Kaiser, CSL 2010

WS1S

$$\forall \mathrm{I}.\ \ \mathrm{I} \vDash \varphi \iff \mathrm{I} \vDash \psi?$$

$$\mathrm{T} \mid \mathrm{F} \mid x \in X \mid x < y \mid \varphi \vee \psi \mid \neg\varphi \mid \mathrm{FO}\ x \mid \exists \underset{\substack{\uparrow \\ \text{finite}}}{X}.\ \varphi$$

Logic-Automaton Connection



Finite Automata

Finite Reachability Games

Fiedor et al., TACAS 2015 dWiNA,

Klarlund, Møller, et al. MONA

Toss Ganzow & Kaiser, CSL 2010

WS1S

$\forall \mathrm{I}.\ \mathrm{I} \models \varphi \iff \mathrm{I} \models \psi?$

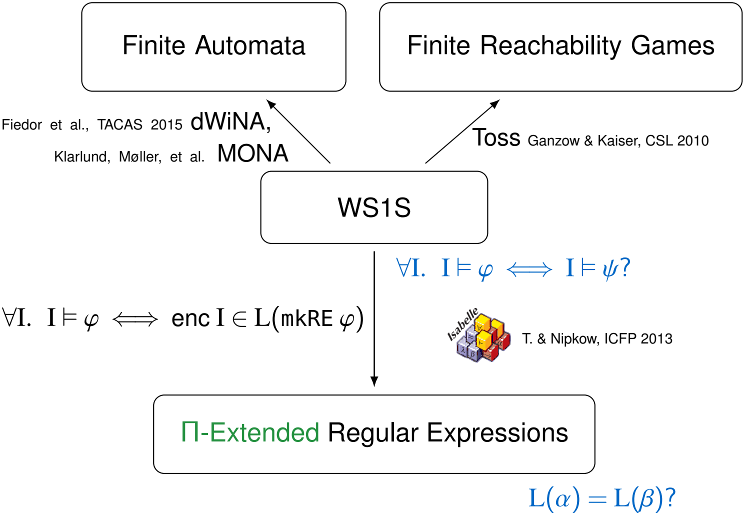
$\forall \mathrm{I}.\ \mathrm{I} \models \varphi \iff \mathsf{enc}\ \mathrm{I} \in \mathrm{L}(\mathtt{mkRE}\ \varphi)$

T. & Nipkow, ICFP 2013

Regular Expressions

$\mathrm{L}(\alpha) = \mathrm{L}(\beta)?$

Logic-Automaton Connection



Finite Automata

Finite Reachability Games

Fiedor et al., TACAS 2015 dWiNA,

Klarlund, Møller, et al. MONA

Toss Ganzow & Kaiser, CSL 2010

WS1S

$\forall \mathrm{I}.\ \ \mathrm{I} \models \varphi \iff \mathrm{I} \models \psi?$

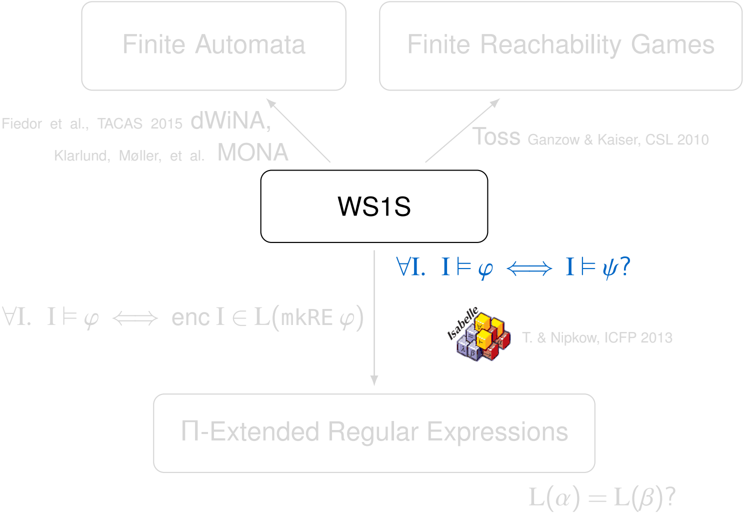
$\forall \mathrm{I}.\ \ \mathrm{I} \models \varphi \iff \mathsf{enc}\ \mathrm{I} \in \mathrm{L}(\mathtt{mkRE}\ \varphi)$

T. & Nipkow, ICFP 2013

Π-Extended Regular Expressions

$\mathrm{L}(\alpha) = \mathrm{L}(\beta)?$

Logic-Automaton Connection



Finite Automata

Finite Reachability Games

Fiedor et al., TACAS 2015 dWiNA,

Klarlund, Møller, et al. MONA

Toss Ganzow & Kaiser, CSL 2010

WS1S

$\forall \mathrm{I}.\ \mathrm{I} \models \varphi \iff \mathrm{I} \models \psi?$

$\forall \mathrm{I}.\ \mathrm{I} \models \varphi \iff \text{enc}\, \mathrm{I} \in \mathrm{L}(\texttt{mkRE}\ \varphi)$

T. & Nipkow, ICFP 2013

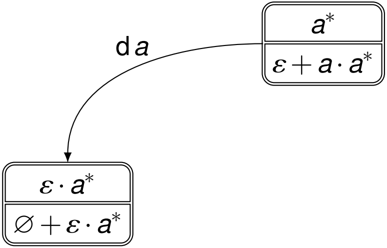
Π-Extended Regular Expressions

$\mathrm{L}(\alpha) = \mathrm{L}(\beta)?$

$a^* \overset{?}{=} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$

a^*
$\varepsilon + a \cdot a^*$

$a^* \overset{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$

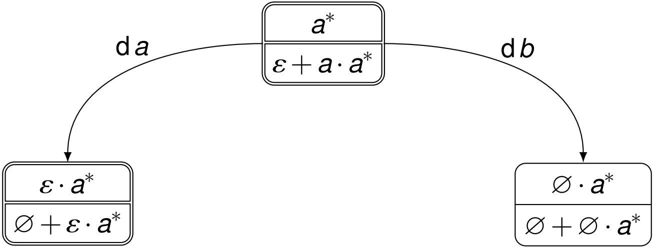


Brzozowski derivative

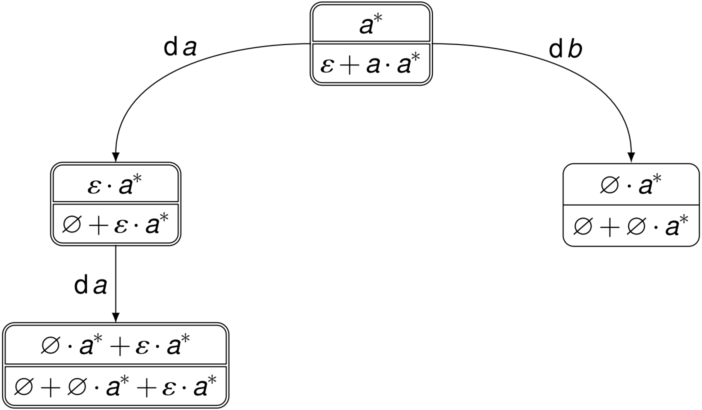
d : letter → regex → regex

$$\mathrm{L}(\mathrm{d}\, a\, r) = \{w \mid aw \in \mathrm{L}(r)\}$$

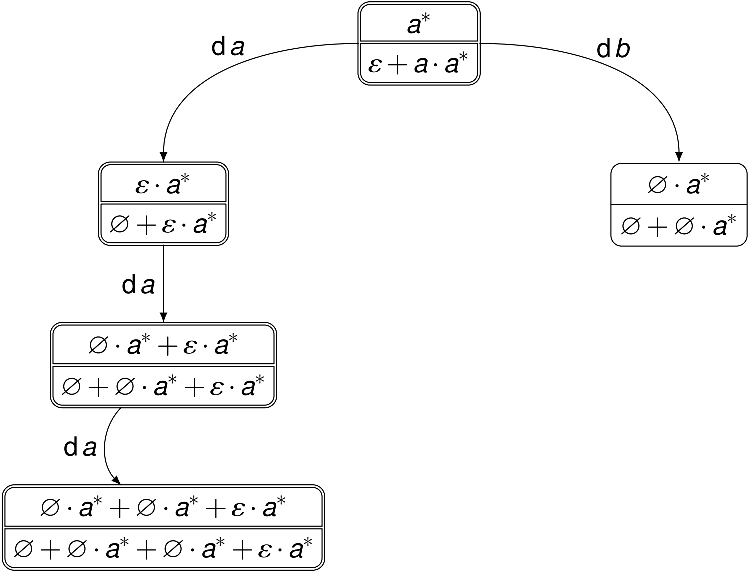
$a^* \overset{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



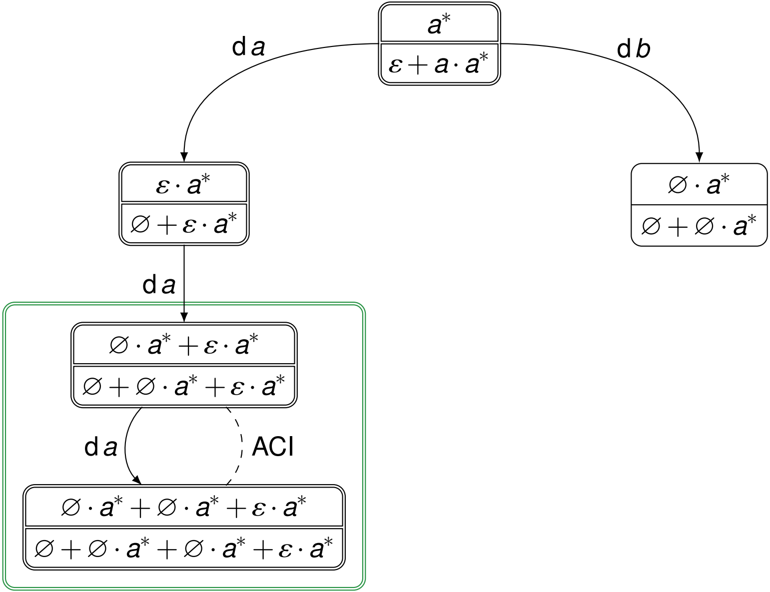
$a^* \overset{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



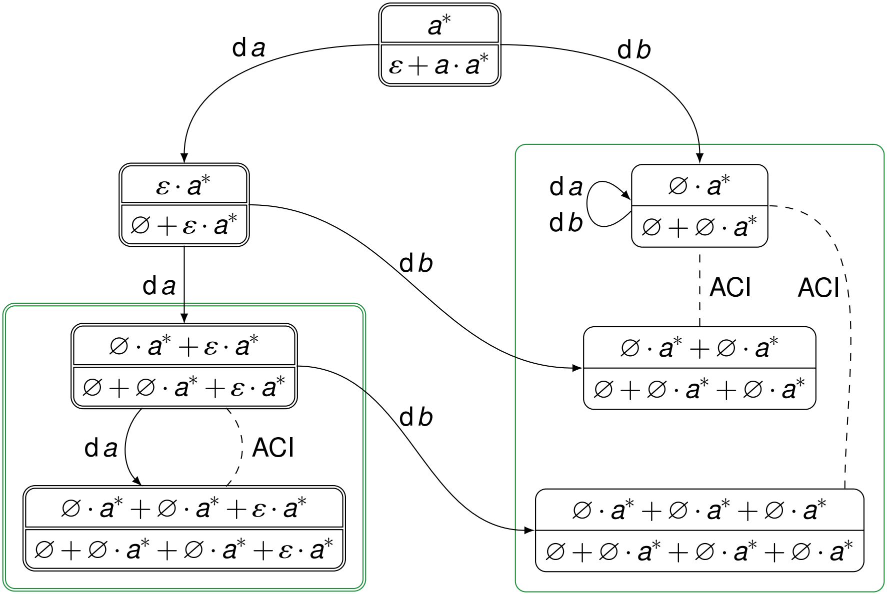
$a^* \overset{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



$$a^* \overset{?}{\equiv} \varepsilon + a \cdot a^* \text{ for } \Sigma = \{a, b\}$$



$a^* \overset{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



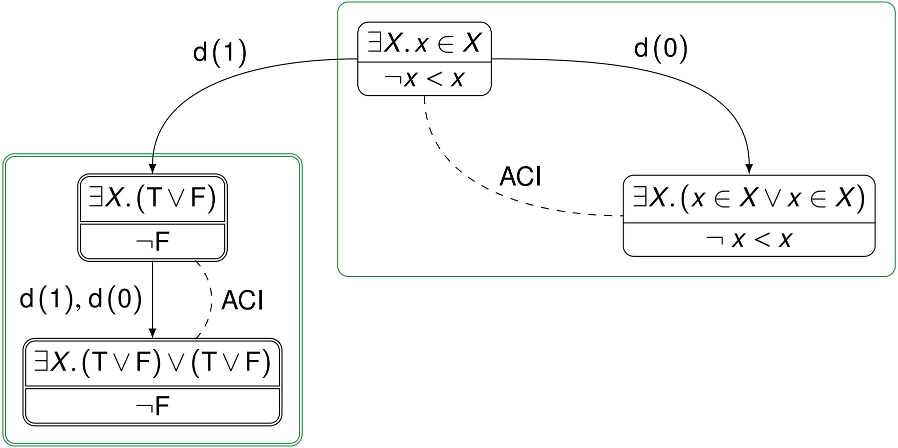
Key ingredients: derivative + ε-acceptance test

Key ingredients: $\underbrace{\text{derivative + }\varepsilon\text{-acceptance test}}_{\text{coalgebra}}$

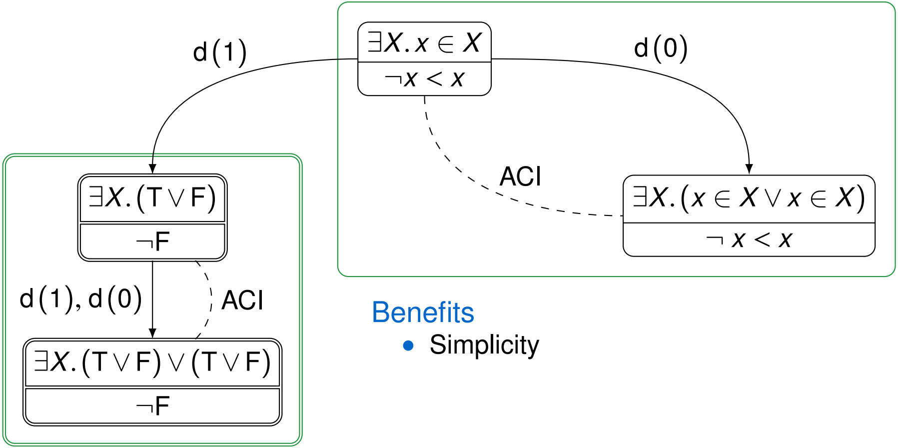
Key ingredients: $\underbrace{\text{derivative} + \varepsilon\text{-acceptance test}}_{\text{coalgebra}}$

Let’s define them on WS1S formulas directly!

$(\exists X.\, x \in X) \overset{?}{\equiv} (\neg x < x)$ for $\Sigma = \{(0),(1)\}$



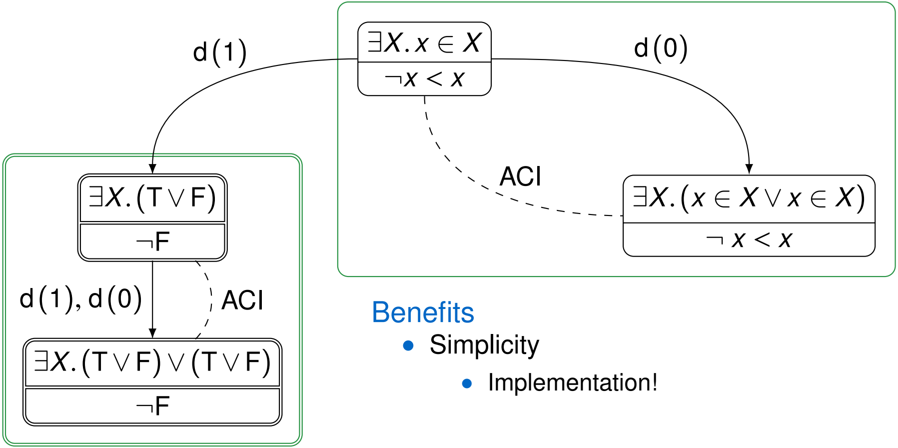
$(\exists X. x \in X) \overset{?}{\equiv} (\neg x < x)$ for $\Sigma = \{(0),(1)\}$



Benefits

- Simplicity

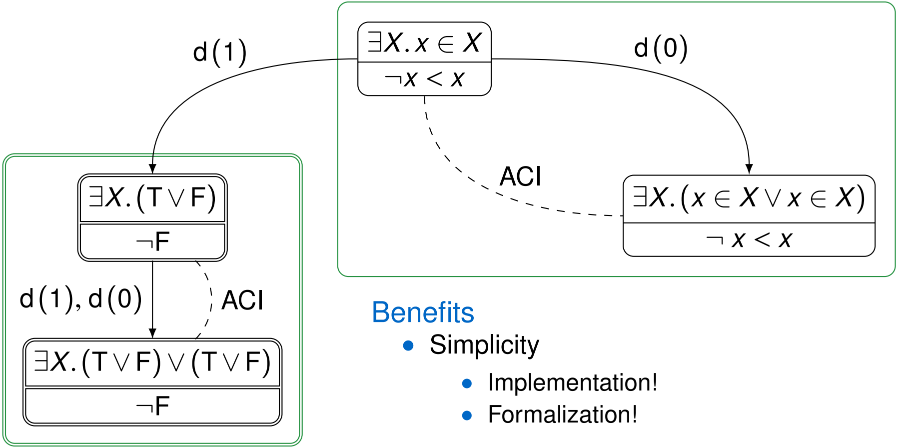
$(\exists X. x \in X) \overset{?}{\equiv} (\neg x < x)$ for $\Sigma = \{(0),(1)\}$



Benefits

- Simplicity
  - Implementation!

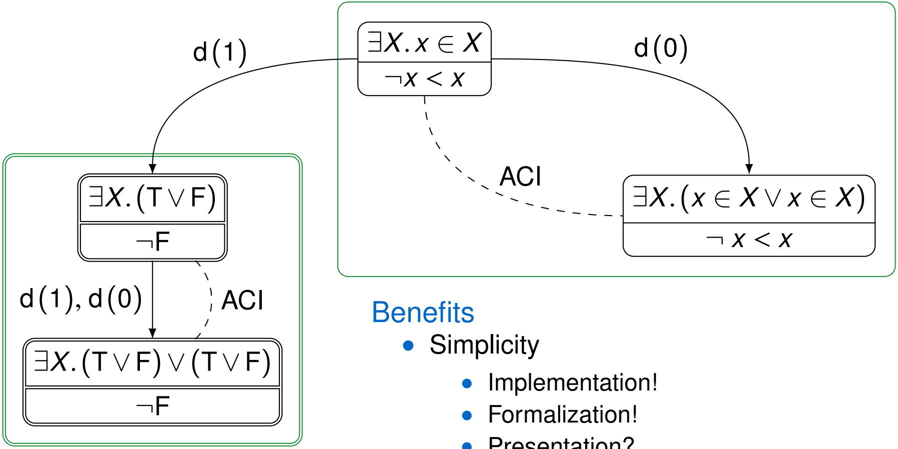
$(\exists X.\, x \in X) \overset{?}{\equiv} (\neg x < x)$ for $\Sigma = \{(0),(1)\}$



Benefits

- Simplicity
    - Implementation!
    - Formalization!

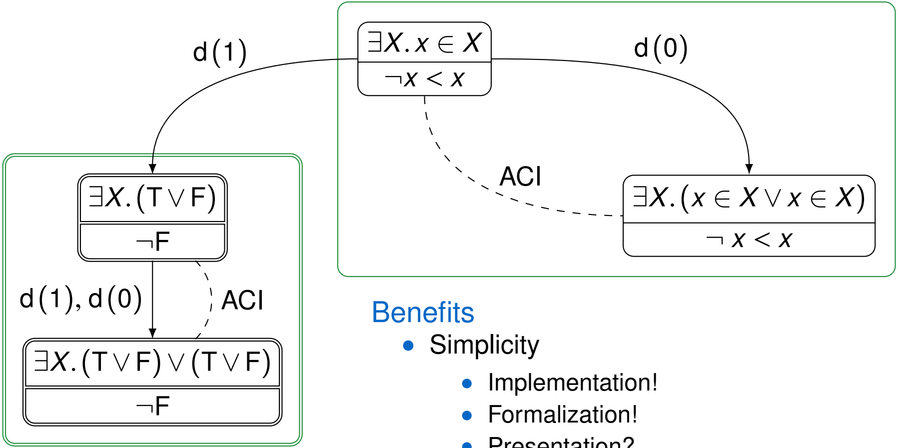
$(\exists X.\, x \in X) \overset{?}{\equiv} (\neg x < x)$ for $\Sigma = \{(0),(1)\}$



Benefits

- Simplicity
  - Implementation!
  - Formalization!
  - Presentation?

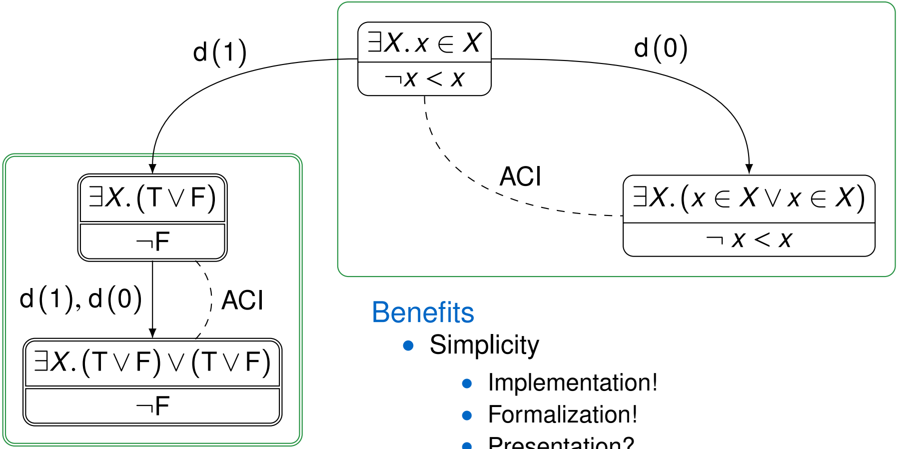
$(\exists X. x \in X) \overset{?}{\equiv} (\neg x < x)$ for $\Sigma = \{(0),(1)\}$



Benefits

- Simplicity
    - Implementation!
    - Formalization!
    - Presentation?
- Efficiency?

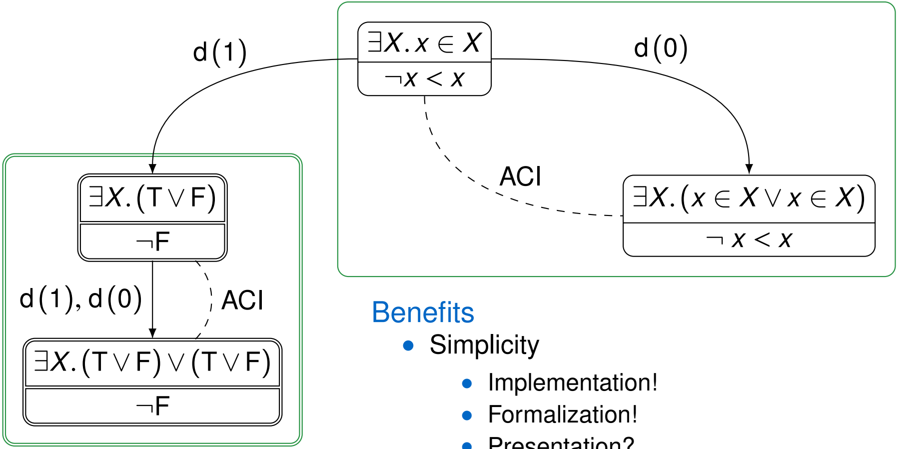
$(\exists X. x \in X) \overset{?}{\equiv} (\neg x < x)$ for $\Sigma = \{(0), (1)\}$



Benefits

- Simplicity
    - Implementation!
    - Formalization!
    - Presentation?
- Efficiency?
    - vs. MONA

$(\exists X. x \in X) \overset{?}{\equiv} (\neg x < x)$ for $\Sigma = \{(0),(1)\}$



Benefits

- Simplicity
  - Implementation!
  - Formalization!
  - Presentation?
- Efficiency?
  - vs. MONA

  → MonaCo (Pous & T., ongoing work)

Interlude I: Encoding of Interpretations

$$I = \begin{cases} X \mapsto \{1, 2, 3\} \\ Y \mapsto \{0, 2\} \\ Z \mapsto \{3\} \end{cases}$$

Interlude I: Encoding of Interpretations

$$\begin{tikzcd}
I = \begin{cases} X \mapsto \{1, 2, 3\} \\ Y \mapsto \{0, 2\} \\ Z \mapsto \{3\} \end{cases} \arrow[d, "\text{enc}"] \\
\begin{array}{c cccc} X & 0 & 1 & 1 & 1 \\ Y & 1 & 0 & 1 & 0 \\ Z & 0 & 0 & 0 & 1 \end{array}
\end{tikzcd}$$

Interlude I: Encoding of Interpretations

$$\begin{tikzcd}
I = \begin{cases} X \mapsto \{1, 2, 3\} \\ Y \mapsto \{0, 2\} \\ Z \mapsto \{3\} \end{cases} \arrow[d, "\text{enc}"] & \\
\begin{array}{ccccc} X & 0 & 1 & 1 & 1 \\ Y & 1 & 0 & 1 & 0 \\ Z & 0 & 0 & 0 & 1 \end{array} \arrow[r, "\text{tail}"'] & \begin{array}{cccc} X & 1 & 1 & 1 \\ Y & 0 & 1 & 0 \\ Z & 0 & 0 & 1 \end{array}
\end{tikzcd}$$

Interlude I: Encoding of Interpretations

$$\begin{tikzcd}[ampersand replacement=\&]
I = \begin{cases} X \mapsto \{1, 2, 3\} \\ Y \mapsto \{0, 2\} \\ Z \mapsto \{3\} \end{cases} \arrow[d, "\mathrm{enc}"'] \& \mathrm{TAIL}\ I = \begin{cases} X \mapsto \{0, 1, 2\} \\ Y \mapsto \{1\} \\ Z \mapsto \{2\} \end{cases} \arrow[d, "\mathrm{enc}"'] \\
\begin{array}{ccccc} X & 0 & 1 & 1 & 1 \\ Y & 1 & 0 & 1 & 0 \\ Z & 0 & 0 & 0 & 1 \end{array} \arrow[r, "\mathrm{tail}"'] \& \begin{array}{cccc} X & 1 & 1 & 1 \\ Y & 0 & 1 & 0 \\ Z & 0 & 0 & 1 \end{array}
\end{tikzcd}$$

Interlude I: Encoding of Interpretations

$$I \vDash \varphi \iff \mathrm{TAIL}\ I \vDash \mathrm{d}\ (\mathrm{HEAD}\ I)\ \varphi$$

$$\begin{tikzcd}[ampersand replacement=\&]
I = \begin{cases} X \mapsto \{1, 2, 3\} \\ Y \mapsto \{0, 2\} \\ Z \mapsto \{3\} \end{cases} \arrow[d, "\text{enc}"] \& \mathrm{TAIL}\ I = \begin{cases} X \mapsto \{0, 1, 2\} \\ Y \mapsto \{1\} \\ Z \mapsto \{2\} \end{cases} \arrow[d, "\text{enc}"] \\
\begin{array}{ccccc} X & 0 & 1 & 1 & 1 \\ Y & 1 & 0 & 1 & 0 \\ Z & 0 & 0 & 0 & 1 \end{array} \arrow[r, "\text{tail}"'] \& \begin{array}{cccc} X & 1 & 1 & 1 \\ Y & 0 & 1 & 0 \\ Z & 0 & 0 & 1 \end{array}
\end{tikzcd}$$

Interlude I: Encoding of Interpretations

$$I \models \varphi \iff \mathrm{TAIL}\ I \models \mathrm{d} \begin{bmatrix}0\\1\\0\end{bmatrix} \varphi$$

$$\begin{tikzcd}
I = \begin{cases} X \mapsto \{1,2,3\} \\ Y \mapsto \{0,2\} \\ Z \mapsto \{3\} \end{cases} \arrow[d, "\mathrm{enc}"] & & \mathrm{TAIL}\ I = \begin{cases} X \mapsto \{0,1,2\} \\ Y \mapsto \{1\} \\ Z \mapsto \{2\} \end{cases} \arrow[d, "\mathrm{enc}"] \\
\begin{array}{c|cccc} X & 0 & 1 & 1 & 1 \\ Y & 1 & 0 & 1 & 0 \\ Z & 0 & 0 & 0 & 1 \end{array} \arrow[rr, "\mathrm{tail}"'] & & \begin{array}{c|ccc} X & 1 & 1 & 1 \\ Y & 0 & 1 & 0 \\ Z & 0 & 0 & 1 \end{array}
\end{tikzcd}$$

Interlude II: First-Order Variables

Does $x \mapsto \{1, 2, 3\}$ satisfy FO x?

Interlude II: First-Order Variables

Does $x \mapsto \{1, 2, 3\}$ satisfy FO x?

No, only singleton sets do

Interlude II: First-Order Variables

Does $x \mapsto \{1,2,3\}$ satisfy FO x?

No, only singleton sets do	Yes, all non-empty sets do Minimum is the assigned value

Interlude II: First-Order Variables

Does $x \mapsto \{1, 2, 3\}$ satisfy FO x?

No, only singleton sets do	Yes, all non-empty sets do Minimum is the assigned value
→ my Ph.D. thesis draft	→ here (also used in MONA)

Derivative

d : letter $\rightarrow$ formula $\rightarrow$ formula

Derivative

$\mathsf{d} : \mathsf{letter} \rightarrow \mathsf{formula} \rightarrow \mathsf{formula}$

$$\begin{aligned} \mathsf{d}\ v\ \mathsf{T} &= \mathsf{T} \\ \mathsf{d}\ v\ \mathsf{F} &= \mathsf{F} \end{aligned}$$

Derivative

$$
\begin{aligned}
&\mathsf{d} : \text{letter} \to \text{formula} \to \text{formula} \\
&\mathsf{d}\ v\ \mathsf{T} &&= \mathsf{T} \\
&\mathsf{d}\ v\ \mathsf{F} &&= \mathsf{F} \\
&\mathsf{d}\ v\ (\mathsf{FO}\ x) &&= \begin{cases} \mathsf{FO}\ x & \text{if } \neg v[x] \\ \mathsf{T} & \text{otherwise} \end{cases}
\end{aligned}
$$

Derivative

$$\begin{array}{lcl}
\mathrm{d} : \text{letter} \rightarrow \text{formula} \rightarrow \text{formula} & & \\
\mathrm{d}\ v\ \mathrm{T} & = & \mathrm{T} \\
\mathrm{d}\ v\ \mathrm{F} & = & \mathrm{F} \\
\mathrm{d}\ v\ (\mathrm{FO}\ x) & = & \begin{cases} \mathrm{FO}\ x & \text{if } \neg v[x] \\ \mathrm{T} & \text{otherwise} \end{cases} \\
\mathrm{d}\ v\ (x \in X) & = & \begin{cases} x \in X & \text{if } \neg v[x] \\ \mathrm{T} & \text{if } v[x] \wedge v[X] \\ \mathrm{F} & \text{otherwise} \end{cases} \\
\mathrm{d}\ v\ (x < y) & = & \begin{cases} x < y & \text{if } \neg v[x] \wedge \neg v[y] \\ \mathrm{FO}\ y & \text{if } v[x] \wedge \neg v[y] \\ \mathrm{F} & \text{otherwise} \end{cases}
\end{array}$$

Derivative

$$
\begin{array}{lcl}
\mathrm{d} : \text{letter} \to \text{formula} \to \text{formula} & & \\
\mathrm{d}\ v\ \mathrm{T} & = & \mathrm{T} \\
\mathrm{d}\ v\ \mathrm{F} & = & \mathrm{F} \\
\mathrm{d}\ v\ (\mathrm{FO}\ x) & = & \begin{cases} \mathrm{FO}\ x & \text{if } \neg v[x] \\ \mathrm{T} & \text{otherwise} \end{cases} \\
\mathrm{d}\ v\ (x \in X) & = & \begin{cases} x \in X & \text{if } \neg v[x] \\ \mathrm{T} & \text{if } v[x] \wedge v[X] \\ \mathrm{F} & \text{otherwise} \end{cases} \\
\mathrm{d}\ v\ (x < y) & = & \begin{cases} x < y & \text{if } \neg v[x] \wedge \neg v[y] \\ \mathrm{FO}\ y & \text{if } v[x] \wedge \neg v[y] \\ \mathrm{F} & \text{otherwise} \end{cases} \\
\mathrm{d}\ v\ (\varphi \vee \psi) & = & \mathrm{d}\ v\ \varphi \vee \mathrm{d}\ v\ \psi \\
\mathrm{d}\ v\ (\neg \varphi) & = & \neg\, \mathrm{d}\ v\ \varphi
\end{array}
$$

Derivative

$$
\begin{array}{lcl}
\mathsf{d} : \text{letter} \rightarrow \text{formula} \rightarrow \text{formula} & & \\
\mathsf{d}\ v\ \mathsf{T} & = & \mathsf{T} \\
\mathsf{d}\ v\ \mathsf{F} & = & \mathsf{F} \\
\mathsf{d}\ v\ (\mathsf{FO}\ x) & = & \begin{cases} \mathsf{FO}\ x & \text{if } \neg v[x] \\ \mathsf{T} & \text{otherwise} \end{cases} \\
\mathsf{d}\ v\ (x \in X) & = & \begin{cases} x \in X & \text{if } \neg v[x] \\ \mathsf{T} & \text{if } v[x] \wedge v[X] \\ \mathsf{F} & \text{otherwise} \end{cases} \\
\mathsf{d}\ v\ (x < y) & = & \begin{cases} x < y & \text{if } \neg v[x] \wedge \neg v[y] \\ \mathsf{FO}\ y & \text{if } v[x] \wedge \neg v[y] \\ \mathsf{F} & \text{otherwise} \end{cases} \\
\mathsf{d}\ v\ (\varphi \vee \psi) & = & \mathsf{d}\ v\ \varphi \vee \mathsf{d}\ v\ \psi \\
\mathsf{d}\ v\ (\neg \varphi) & = & \neg\, \mathsf{d}\ v\ \varphi \\
\mathsf{d}\ v\ (\exists X.\, \varphi) & = & \exists X.\, (\mathsf{d}\ (v_{X \mapsto 1})\ \varphi \vee \mathsf{d}\ (v_{X \mapsto 0})\ \varphi)
\end{array}
$$

Acceptance Test

ε : formula $\to$ bool

Acceptance Test

$$\begin{array}{lcl}
\varepsilon : \text{formula} \to \text{bool} & & \\
\varepsilon\,\mathsf{T} & = & 1 \\
\varepsilon\,\mathsf{F} & = & 0 \\
\varepsilon\,(\mathsf{FO}\,x) & = & 0 \\
\varepsilon\,(x \in X) & = & 0 \\
\varepsilon\,(x < y) & = & 0 \\
\varepsilon\,(\varphi \vee \psi) & = & \varepsilon\,\varphi \vee \varepsilon\,\psi \\
\varepsilon\,(\neg\,\varphi) & = & \neg\,\varepsilon\,\varphi \\
\varepsilon\,(\exists X.\,\varphi) & = & \varepsilon\,\varphi
\end{array}$$

Acceptance Test

$$
\begin{array}{lcl}
\varepsilon : \text{formula} \to \text{bool} & & \\
\varepsilon\, \mathsf{T} & = & 1 \\
\varepsilon\, \mathsf{F} & = & 0 \\
\varepsilon\, (\mathsf{FO}\; x) & = & 0 \\
\varepsilon\, (x \in X) & = & 0 \\
\varepsilon\, (x < y) & = & 0 \\
\varepsilon\, (\varphi \vee \psi) & = & \varepsilon\, \varphi \vee \varepsilon\, \psi \\
\varepsilon\, (\neg\, \varphi) & = & \neg\, \varepsilon\, \varphi \\
\varepsilon\, (\exists X .\, \varphi) & = & \varepsilon\, \varphi
\end{array}
$$

Any objections?

Acceptance Test

$\varepsilon : \text{formula} \to \text{bool}$

$$
\begin{aligned}
\varepsilon\, \mathsf{T} &= 1 \\
\varepsilon\, \mathsf{F} &= 0 \\
\varepsilon\, (\mathsf{FO}\ x) &= 0 \\
\varepsilon\, (x \in X) &= 0 \\
\varepsilon\, (x < y) &= 0 \\
\varepsilon\, (\varphi \vee \psi) &= \varepsilon\, \varphi \vee \varepsilon\, \psi \\
\varepsilon\, (\neg\, \varphi) &= \neg\, \varepsilon\, \varphi \\
\varepsilon\, (\exists X.\, \varphi) &= \varepsilon\, \varphi
\end{aligned}
$$

Any objections?

Yes, this decides M2L(Str), not WS1S.

Acceptance Test

$$
\begin{aligned}
&\varepsilon : \text{formula} \to \text{bool} \\
&\varepsilon\, \mathsf{T} &&= && 1 \\
&\varepsilon\, \mathsf{F} &&= && 0 \\
&\varepsilon\, (\mathsf{FO}\ x) &&= && 0 \\
&\varepsilon\, (x \in X) &&= && 0 \\
&\varepsilon\, (x < y) &&= && 0 \\
&\varepsilon\, (\varphi \vee \psi) &&= && \varepsilon\, \varphi \vee \varepsilon\, \psi \\
&\varepsilon\, (\neg\, \varphi) &&= && \neg\, \varepsilon\, \varphi \\
&\varepsilon\, (\exists X.\, \varphi) &&= && \varepsilon\, \varphi
\end{aligned}
$$

Any objections?

Yes, this decides M2L(Str), not WS1S.

Careful with trailing zeros!

Trailing Zeros

$$\begin{array}{c} y \\ x \end{array} \quad \begin{bmatrix}0\\0\end{bmatrix} \begin{bmatrix}0\\1\end{bmatrix} \begin{bmatrix}0\\0\end{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} \quad \models \qquad \overset{\varphi}{x < y}$$

Trailing Zeros

$$\varphi$$

$$\begin{array}{c} y \\ x \end{array} \quad \begin{bmatrix}0\\0\end{bmatrix} \begin{bmatrix}0\\1\end{bmatrix} \begin{bmatrix}0\\0\end{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} \quad \models \quad x < y$$

$$x \quad [0] \; [1] \; [0] \; [0] \quad \models \quad \exists y.\, x < y$$

Trailing Zeros

$$\begin{array}{ccccccr}
 & & & & & & \varphi \\
\begin{matrix} y \\ x \end{matrix} & \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}1\\0\end{bmatrix} & \models & x < y \\
x & [0] & [1] & [0] & [0] & \models & \exists y.\, x < y \\
 & [\ ] & [\ ] & [\ ] & [\ ] & \models & \forall x.\, \exists y.\, x < y
\end{array}$$

Trailing Zeros

$$\begin{array}{cccccccr}
 & & & & & & \varphi & \varepsilon\,\varphi \\
\begin{matrix} y \\ x \end{matrix} & \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}1\\0\end{bmatrix} & \models & x < y & 0 \\
x & [0] & [1] & [0] & [0] & \models & \exists y.\, x < y & 0 \\
 & [\ ] & [\ ] & [\ ] & [\ ] & \models & \forall x.\, \exists y.\, x < y & 0
\end{array}$$

Trailing Zeros

			φ	$\varepsilon\,\varphi$
$\begin{array}{c} y \\ x \end{array}$	$\begin{bmatrix}0\\0\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}$	$\models$	$x < y$	0
x	$[0]\ [1]\ [0]\ [0]$	$\models$	$\exists y.\, x < y$	0
	$[\ ]\ [\ ]\ [\ ]\ [\ ]$	$\models$	$\forall x.\, \exists y.\, x < y$	0

For WS1S: *futurize* formula before applying ε

Trailing Zeros

$$\begin{array}{lcccccrr}
 & & & & & & \varphi & \varepsilon\,\varphi \\
\begin{matrix} y \\ x \end{matrix} & \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}1\\0\end{bmatrix} & \models & x < y & 0 \\
x & [0] & [1] & [0] & [0] & \models & \exists y.\, x < y & 0 \\
 & [\;] & [\;] & [\;] & [\;] & \models & \forall x.\, \exists y.\, x < y & 0
\end{array}$$

For WS1S: *futurize* formula before applying ε

futurize = derive *from the right* by $\begin{bmatrix}0\\\cdots\\0\end{bmatrix}^*$ under quantifiers

Trailing Zeros

$$\begin{array}{cccccccr}
 & & & & & & \varphi & \varepsilon\,\varphi \\
\begin{array}{c} y \\ x \end{array} & \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}1\\0\end{bmatrix} & \models & x < y & 0 \\
x & [0] & [1] & [0] & [0] & \models & \exists y.\, x < y & 0 \\
 & [\ ] & [\ ] & [\ ] & [\ ] & \models & \forall x.\, \exists y.\, x < y & 0
\end{array}$$

For WS1S: *futurize* formula before applying ε

futurize = derive *from the right* by $\begin{bmatrix}0\\\cdots\\0\end{bmatrix}^*$ under quantifiers → paper

Altogether

A decision procedure for WS1S that

Altogether

A decision procedure for WS1S that operates on formulas directly and

Altogether

A decision procedure for WS1S that

operates on formulas directly and

is verified in Isabelle and

Altogether

A decision procedure for WS1S that operates on formulas directly and is verified in Isabelle and outperforms MONA on carefully selected examples.

Altogether

A decision procedure for WS1S that operates on formulas directly and is verified in Isabelle and outperforms MONA on carefully selected examples.

Thanks. Questions?

A Coalgebraic Decision Procedure for WS1S

Dmitriy Traytel