A Verified Decision Procedure

for

Monadic Second-Order Logic on Strings

Dmitriy Traytel   Tobias Nipkow

Technische Universität München

Overview

MSO

Overview

MSO

$\mathscr{L}_{\text{MSO}}(\varphi) = \mathscr{L}_{\text{MSO}}(\psi)$?

Overview

MSO

$\mathscr{L}_{\text{MSO}}(\varphi) = \mathscr{L}_{\text{MSO}}(\psi)$?

Regular Expression Equivalence

Overview

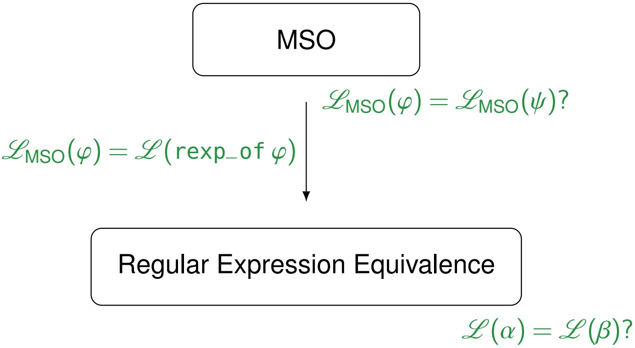
MSO

$\mathscr{L}_{\text{MSO}}(\varphi) = \mathscr{L}_{\text{MSO}}(\psi)$?

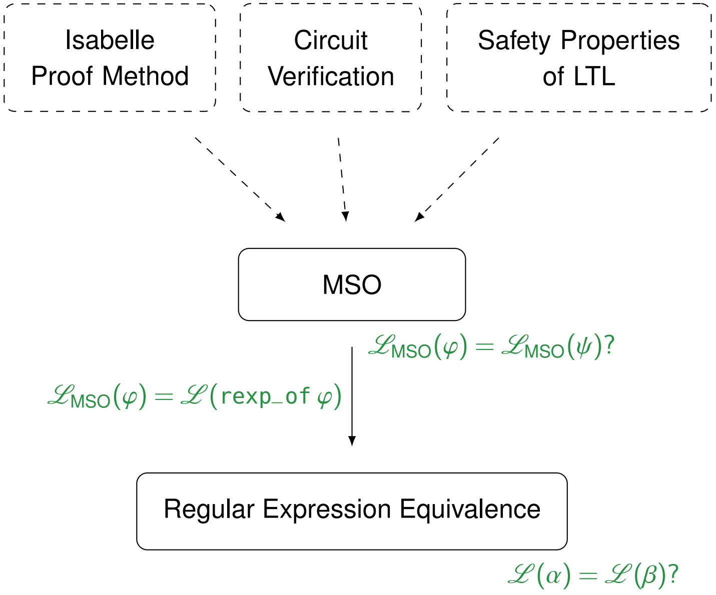
Regular Expression Equivalence

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

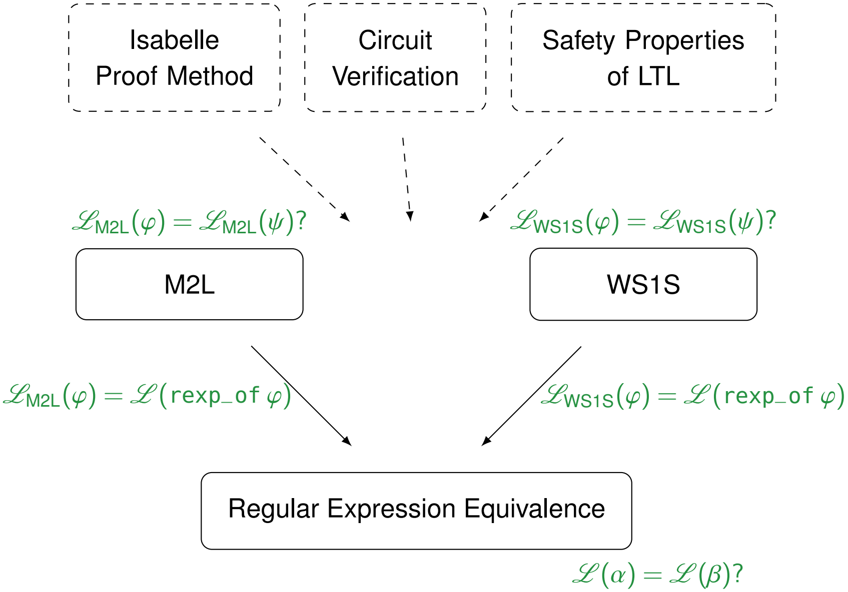
Overview


MSO
$\mathcal{L}_{\text{MSO}}(\varphi) = \mathcal{L}_{\text{MSO}}(\psi)$?
$\mathcal{L}_{\text{MSO}}(\varphi) = \mathcal{L}(\texttt{rexp\_of}\ \varphi)$
Regular Expression Equivalence
$\mathcal{L}(\alpha) = \mathcal{L}(\beta)$?

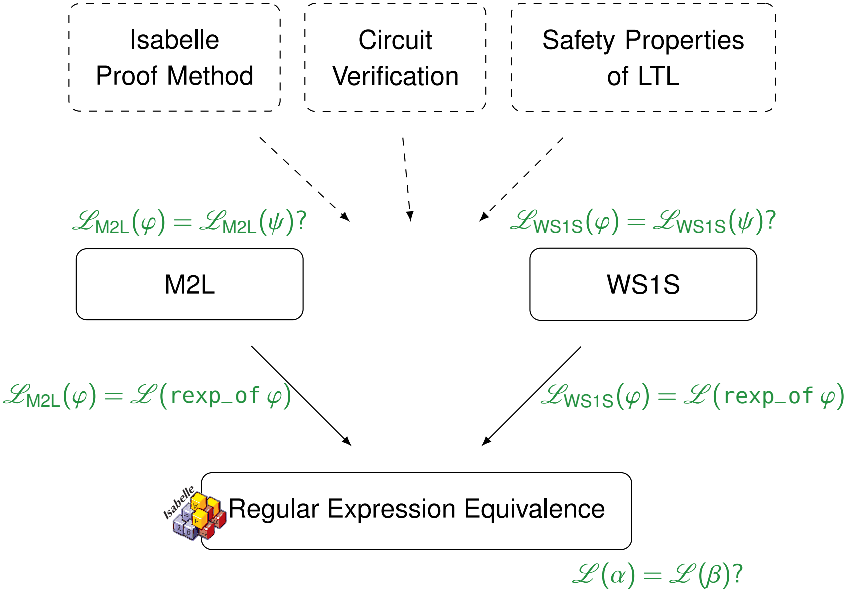
Overview


Isabelle
Proof Method
Circuit
Verification
Safety Properties
of LTL
MSO
$\mathscr{L}_{\text{MSO}}(\varphi) = \mathscr{L}_{\text{MSO}}(\psi)$?
$\mathscr{L}_{\text{MSO}}(\varphi) = \mathscr{L}(\texttt{rexp\_of}\ \varphi)$
Regular Expression Equivalence
$\mathscr{L}(\alpha) = \mathscr{L}(\beta)$?

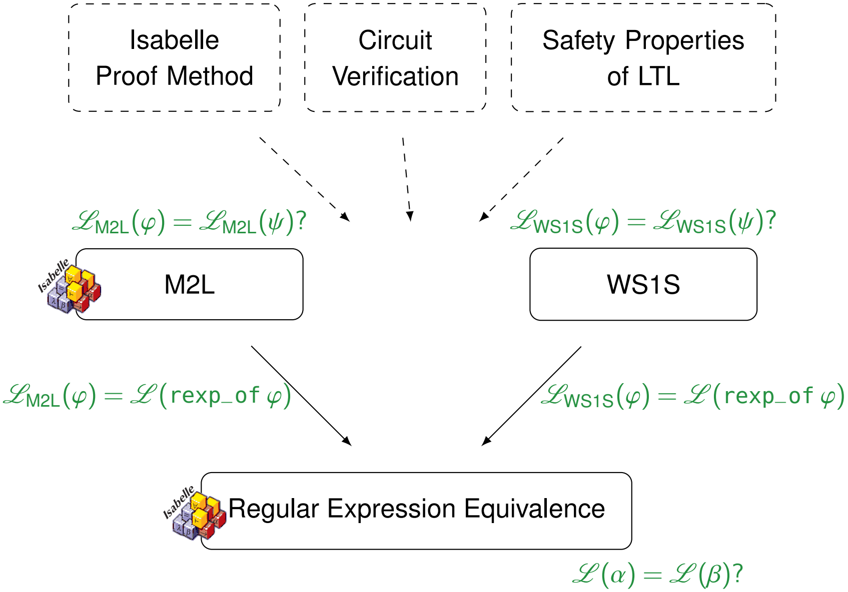
Overview


Isabelle
Proof Method
Circuit
Verification
Safety Properties
of LTL
$\mathscr{L}_{\mathsf{M2L}}(\varphi) = \mathscr{L}_{\mathsf{M2L}}(\psi)$?
$\mathscr{L}_{\mathsf{WS1S}}(\varphi) = \mathscr{L}_{\mathsf{WS1S}}(\psi)$?
M2L
WS1S
$\mathscr{L}_{\mathsf{M2L}}(\varphi) = \mathscr{L}(\texttt{rexp\_of}\ \varphi)$
$\mathscr{L}_{\mathsf{WS1S}}(\varphi) = \mathscr{L}(\texttt{rexp\_of}\ \varphi)$
Regular Expression Equivalence
$\mathscr{L}(\alpha) = \mathscr{L}(\beta)$?

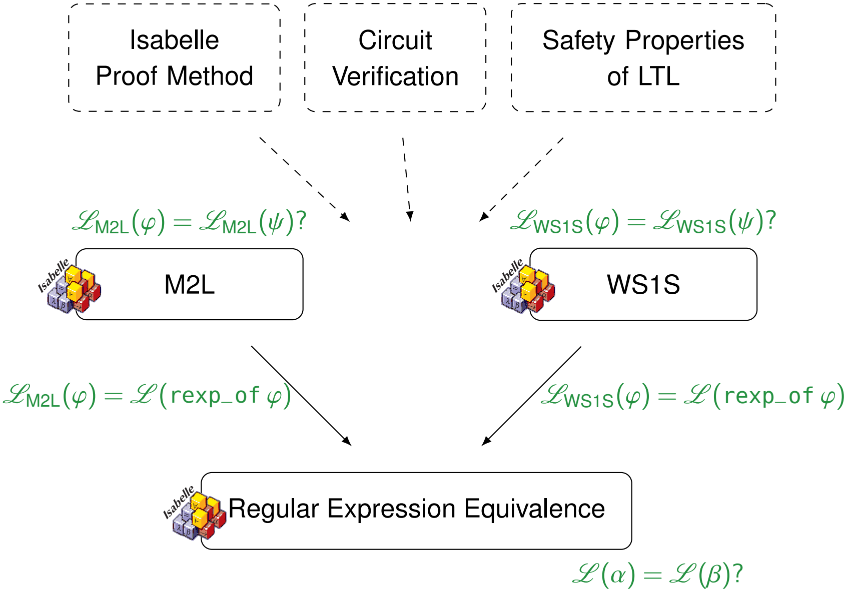
Overview


Isabelle Proof Method
Circuit Verification
Safety Properties of LTL
$\mathscr{L}_{\mathsf{M2L}}(\varphi) = \mathscr{L}_{\mathsf{M2L}}(\psi)$?
$\mathscr{L}_{\mathsf{WS1S}}(\varphi) = \mathscr{L}_{\mathsf{WS1S}}(\psi)$?
M2L
WS1S
$\mathscr{L}_{\mathsf{M2L}}(\varphi) = \mathscr{L}(\mathtt{rexp\_of}\ \varphi)$
$\mathscr{L}_{\mathsf{WS1S}}(\varphi) = \mathscr{L}(\mathtt{rexp\_of}\ \varphi)$
Regular Expression Equivalence
$\mathscr{L}(\alpha) = \mathscr{L}(\beta)$?

Overview


Isabelle
Proof Method
Circuit
Verification
Safety Properties
of LTL
$\mathscr{L}_{\mathsf{M2L}}(\varphi) = \mathscr{L}_{\mathsf{M2L}}(\psi)$?
$\mathscr{L}_{\mathsf{WS1S}}(\varphi) = \mathscr{L}_{\mathsf{WS1S}}(\psi)$?
M2L
WS1S
$\mathscr{L}_{\mathsf{M2L}}(\varphi) = \mathscr{L}(\texttt{rexp\_of}\ \varphi)$
$\mathscr{L}_{\mathsf{WS1S}}(\varphi) = \mathscr{L}(\texttt{rexp\_of}\ \varphi)$
Regular Expression Equivalence
$\mathscr{L}(\alpha) = \mathscr{L}(\beta)$?

Overview


Isabelle Proof Method
Circuit Verification
Safety Properties of LTL
$\mathscr{L}_{\mathsf{M2L}}(\varphi) = \mathscr{L}_{\mathsf{M2L}}(\psi)$?
$\mathscr{L}_{\mathsf{WS1S}}(\varphi) = \mathscr{L}_{\mathsf{WS1S}}(\psi)$?
M2L
WS1S
$\mathscr{L}_{\mathsf{M2L}}(\varphi) = \mathscr{L}(\texttt{rexp\_of}\ \varphi)$
$\mathscr{L}_{\mathsf{WS1S}}(\varphi) = \mathscr{L}(\texttt{rexp\_of}\ \varphi)$
Regular Expression Equivalence
$\mathscr{L}(\alpha) = \mathscr{L}(\beta)$?

Outline

Regular Expressions Equivalence

MSO

Syntax of Π-Extended Regular Expressions

$$
\begin{array}{rcl}
\texttt{rexp} & = & \varnothing \\
& | & \varepsilon \\
& | & a \\
& | & \texttt{rexp} + \texttt{rexp} \\
& | & \texttt{rexp} \cdot \texttt{rexp} \\
& | & \texttt{rexp}^* \\
& | & \texttt{rexp} \cap \texttt{rexp} \\
& | & \neg\, \texttt{rexp} \\
& | & \Pi\, \texttt{rexp}
\end{array}
$$

Semantics of Π-Extended Regular Expressions

$$
\begin{aligned}
\mathscr{L}(\varnothing) &= \{\} \\
\mathscr{L}(\varepsilon) &= \{\varepsilon\} \\
\mathscr{L}(a) &= \{a\} && a \in \Sigma \\
\mathscr{L}(\alpha+\beta) &= \mathscr{L}(\alpha) \cup \mathscr{L}(\beta) \\
\mathscr{L}(\alpha \cdot \beta) &= \mathscr{L}(\alpha) \cdot \mathscr{L}(\beta) \\
\mathscr{L}(\alpha^*) &= \mathscr{L}(\alpha)^*
\end{aligned}
$$

Semantics of Π-Extended Regular Expressions

$$
\begin{aligned}
\mathscr{L}(\emptyset) &= \{\} \\
\mathscr{L}(\varepsilon) &= \{\varepsilon\} \\
\mathscr{L}(a) &= \{a\} && a \in \Sigma \\
\mathscr{L}(\alpha+\beta) &= \mathscr{L}(\alpha) \cup \mathscr{L}(\beta) \\
\mathscr{L}(\alpha\cdot\beta) &= \mathscr{L}(\alpha) \cdot \mathscr{L}(\beta) \\
\mathscr{L}(\alpha^*) &= \mathscr{L}(\alpha)^* \\
\mathscr{L}(\alpha\cap\beta) &= \mathscr{L}(\alpha) \cap \mathscr{L}(\beta) \\
\mathscr{L}(\neg\,\alpha) &= \Sigma^* \setminus \mathscr{L}(\alpha)
\end{aligned}
$$

Semantics of Π-Extended Regular Expressions

$$
\begin{aligned}
\mathscr{L}(\varnothing) &= \{\} \\
\mathscr{L}(\varepsilon) &= \{\varepsilon\} \\
\mathscr{L}(a) &= \{a\} && a \in \Sigma \\
\mathscr{L}(\alpha + \beta) &= \mathscr{L}(\alpha) \cup \mathscr{L}(\beta) \\
\mathscr{L}(\alpha \cdot \beta) &= \mathscr{L}(\alpha) \cdot \mathscr{L}(\beta) \\
\mathscr{L}(\alpha^*) &= \mathscr{L}(\alpha)^* \\
\mathscr{L}(\alpha \cap \beta) &= \mathscr{L}(\alpha) \cap \mathscr{L}(\beta) \\
\mathscr{L}(\neg\, \alpha) &= \Sigma^* \setminus \mathscr{L}(\alpha) \\
\mathscr{L}(\Pi\, \alpha) &= \{ \quad w \mid w \in \mathscr{L} \quad (\alpha)\}
\end{aligned}
$$

Semantics of Π-Extended Regular Expressions

$$
\begin{aligned}
\mathscr{L}_n(\emptyset) &= \{\} \\
\mathscr{L}_n(\varepsilon) &= \{\varepsilon\} \\
\mathscr{L}_n(a) &= \{a\} & a \in \Sigma_n \\
\mathscr{L}_n(\alpha+\beta) &= \mathscr{L}_n(\alpha) \cup \mathscr{L}_n(\beta) \\
\mathscr{L}_n(\alpha\cdot\beta) &= \mathscr{L}_n(\alpha) \cdot \mathscr{L}_n(\beta) \\
\mathscr{L}_n(\alpha^*) &= \mathscr{L}_n(\alpha)^* \\
\mathscr{L}_n(\alpha\cap\beta) &= \mathscr{L}_n(\alpha) \cap \mathscr{L}_n(\beta) \\
\mathscr{L}_n(\neg\,\alpha) &= \Sigma_n^* \setminus \mathscr{L}_n(\alpha) \\
\mathscr{L}_n(\Pi\,\alpha) &= \{\qquad w \mid w \in \mathscr{L}_{n+1}(\alpha)\}
\end{aligned}
$$

Semantics of Π-Extended Regular Expressions

$$
\begin{array}{rcll}
\mathscr{L}_n(\varnothing) & = & \{\} & \\
\mathscr{L}_n(\varepsilon) & = & \{\varepsilon\} & \\
\mathscr{L}_n(a) & = & \{a\} & a \in \Sigma_n \\
\mathscr{L}_n(\alpha + \beta) & = & \mathscr{L}_n(\alpha) \cup \mathscr{L}_n(\beta) & \\
\mathscr{L}_n(\alpha \cdot \beta) & = & \mathscr{L}_n(\alpha) \cdot \mathscr{L}_n(\beta) & \\
\mathscr{L}_n(\alpha^*) & = & \mathscr{L}_n(\alpha)^* & \\
\mathscr{L}_n(\alpha \cap \beta) & = & \mathscr{L}_n(\alpha) \cap \mathscr{L}_n(\beta) & \\
\mathscr{L}_n(\neg\, \alpha) & = & \Sigma_n^* \setminus \mathscr{L}_n(\alpha) & \\
\mathscr{L}_n(\Pi\, \alpha) & = & \{\mathtt{map}\ \pi\ w \mid w \in \mathscr{L}_{n+1}(\alpha)\} &
\end{array}
$$

$$\pi : \Sigma_{n+1} \to \Sigma_n$$

Example

$$\Sigma_n = \{xs \mid \texttt{length}\ xs = n\}$$

$$\pi = \texttt{tail}$$

$$\begin{aligned}\pi^{-1} a &= \{xs \mid \texttt{tail}\ xs = a\} \\ &= \{a_0 a \mid a_0 \in \Sigma_1\}\end{aligned}$$

Derivatives of Regular Expressions

$$
\begin{aligned}
\mathscr{D}_a(\varnothing) &= \varnothing \\
\mathscr{D}_a(\varepsilon) &= \varnothing \\
\mathscr{D}_a(b) &= \texttt{if}\ a = b\ \texttt{then}\ \varepsilon\ \texttt{else}\ \varnothing \\
\mathscr{D}_a(\alpha + \beta) &= \mathscr{D}_a(\alpha) + \mathscr{D}_a(\beta) \\
\mathscr{D}_a(\alpha \cdot \beta) &= \texttt{if}\ \varepsilon \in \mathscr{L}(\alpha)\ \texttt{then}\ \mathscr{D}_a(\alpha) \cdot \beta + \mathscr{D}_a(\beta)\ \texttt{else}\ \mathscr{D}_a(\alpha) \cdot \beta \\
\mathscr{D}_a(\alpha^*) &= \mathscr{D}_a(\alpha) \cdot \alpha^*
\end{aligned}
$$

Derivatives of Regular Expressions

$$
\begin{aligned}
\mathscr{D}_a(\varnothing) &= \varnothing \\
\mathscr{D}_a(\varepsilon) &= \varnothing \\
\mathscr{D}_a(b) &= \texttt{if } a = b \texttt{ then } \varepsilon \texttt{ else } \varnothing \\
\mathscr{D}_a(\alpha + \beta) &= \mathscr{D}_a(\alpha) + \mathscr{D}_a(\beta) \\
\mathscr{D}_a(\alpha \cdot \beta) &= \texttt{if } \varepsilon \in \mathscr{L}(\alpha) \texttt{ then } \mathscr{D}_a(\alpha) \cdot \beta + \mathscr{D}_a(\beta) \texttt{ else } \mathscr{D}_a(\alpha) \cdot \beta \\
\mathscr{D}_a(\alpha^*) &= \mathscr{D}_a(\alpha) \cdot \alpha^* \\
\mathscr{D}_a(\alpha \cap \beta) &= \mathscr{D}_a(\alpha) \cap \mathscr{D}_a(\beta) \\
\mathscr{D}_a(\neg\, \alpha) &= \neg\, \mathscr{D}_a(\alpha)
\end{aligned}
$$

Derivatives of Regular Expressions

$$
\begin{aligned}
\mathscr{D}_a(\varnothing) &= \varnothing \\
\mathscr{D}_a(\varepsilon) &= \varnothing \\
\mathscr{D}_a(b) &= \texttt{if } a = b \texttt{ then } \varepsilon \texttt{ else } \varnothing \\
\mathscr{D}_a(\alpha + \beta) &= \mathscr{D}_a(\alpha) + \mathscr{D}_a(\beta) \\
\mathscr{D}_a(\alpha \cdot \beta) &= \texttt{if } \varepsilon \in \mathscr{L}(\alpha) \texttt{ then } \mathscr{D}_a(\alpha) \cdot \beta + \mathscr{D}_a(\beta) \texttt{ else } \mathscr{D}_a(\alpha) \cdot \beta \\
\mathscr{D}_a(\alpha^*) &= \mathscr{D}_a(\alpha) \cdot \alpha^* \\
\mathscr{D}_a(\alpha \cap \beta) &= \mathscr{D}_a(\alpha) \cap \mathscr{D}_a(\beta) \\
\mathscr{D}_a(\neg\, \alpha) &= \neg\, \mathscr{D}_a(\alpha) \\
\mathscr{D}_a(\sqcap\, \alpha) &= \sqcap \left(\bigoplus_{b \in \pi^{-1} a} \mathscr{D}_b(\alpha) \right)
\end{aligned}
$$

Derivatives of Regular Expressions

$$
\begin{aligned}
\mathscr{D}_a(\varnothing) &= \varnothing \\
\mathscr{D}_a(\varepsilon) &= \varnothing \\
\mathscr{D}_a(b) &= \texttt{if } a = b \texttt{ then } \varepsilon \texttt{ else } \varnothing \\
\mathscr{D}_a(\alpha + \beta) &= \mathscr{D}_a(\alpha) + \mathscr{D}_a(\beta) \\
\mathscr{D}_a(\alpha \cdot \beta) &= \texttt{if } \varepsilon \in \mathscr{L}(\alpha) \texttt{ then } \mathscr{D}_a(\alpha) \cdot \beta + \mathscr{D}_a(\beta) \texttt{ else } \mathscr{D}_a(\alpha) \cdot \beta \\
\mathscr{D}_a(\alpha^*) &= \mathscr{D}_a(\alpha) \cdot \alpha^* \\
\mathscr{D}_a(\alpha \cap \beta) &= \mathscr{D}_a(\alpha) \cap \mathscr{D}_a(\beta) \\
\mathscr{D}_a(\neg\, \alpha) &= \neg\, \mathscr{D}_a(\alpha) \\
\mathscr{D}_a(\Pi\, \alpha) &= \Pi \left(\bigoplus_{b \in \pi^{-1} a} \mathscr{D}_b(\alpha) \right)
\end{aligned}
$$

Theorem

$$\mathscr{L}_n(\mathscr{D}_a(\alpha)) = \{ w \mid aw \in \mathscr{L}_n(\alpha) \}$$

Decision Procedure

Main Theorem Let $\mathscr{B} = \{(\lfloor \mathscr{D}_w(\alpha) \rfloor, \lfloor \mathscr{D}_w(\beta) \rfloor) \mid w \in \Sigma_n^*\}$

Decision Procedure

Main Theorem Let $\mathscr{B} = \{(\lfloor \mathscr{D}_w(\alpha) \rfloor, \lfloor \mathscr{D}_w(\beta) \rfloor) \mid w \in \Sigma_n^*\}$

$$\mathscr{L}_n(\alpha) = \mathscr{L}_n(\beta)$$

$$\Leftrightarrow$$

$$\forall (\alpha', \beta') \in \mathscr{B}. \quad \varepsilon \in \mathscr{L}_n(\alpha') \Leftrightarrow \varepsilon \in \mathscr{L}_n(\beta')$$

Decision Procedure

Main Theorem Let $\mathscr{B} = \{(\lfloor \mathscr{D}_w(\alpha) \rfloor, \lfloor \mathscr{D}_w(\beta) \rfloor) \mid w \in \Sigma_n^*\}$

$$\mathscr{L}_n(\alpha) = \mathscr{L}_n(\beta)$$

$$\Leftrightarrow$$

$$\forall (\alpha', \beta') \in \mathscr{B}. \quad \varepsilon \in \mathscr{L}_n(\alpha') \Leftrightarrow \varepsilon \in \mathscr{L}_n(\beta')$$

1964 Brzozowski Informal proof

2011 Krauss & Nipkow Formal soundness proof ($\Leftarrow$) for regular expressions

Decision Procedure

Main Theorem Let $\mathscr{B} = \{(\lfloor \mathscr{D}_w(\alpha) \rfloor, \lfloor \mathscr{D}_w(\beta) \rfloor) \mid w \in \Sigma_n^*\}$

$$\mathscr{L}_n(\alpha) = \mathscr{L}_n(\beta)$$

$$\Leftrightarrow$$

$$\forall (\alpha', \beta') \in \mathscr{B}. \quad \varepsilon \in \mathscr{L}_n(\alpha') \Leftrightarrow \varepsilon \in \mathscr{L}_n(\beta')$$

1964 Brzozowski Informal proof

2011 Krauss & Nipkow Formal soundness proof ($\Leftarrow$) for regular expressions

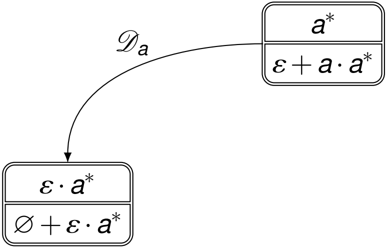
2013 Traytel & Nipkow

- Formal soundness proof ($\Leftarrow$) for Π-extended regular expressions
- Formal completeness proof ($\Rightarrow$)
- Formal termination proof ($\mathscr{B}$ is finite)

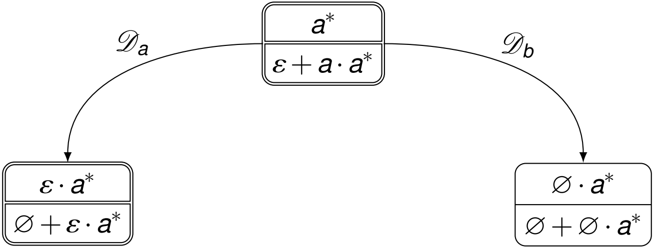
Example: $a^* \stackrel{?}{=} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$

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

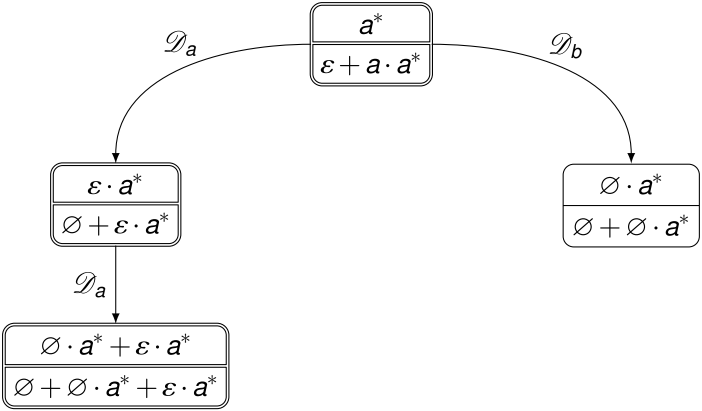
Example: $a^* \stackrel{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



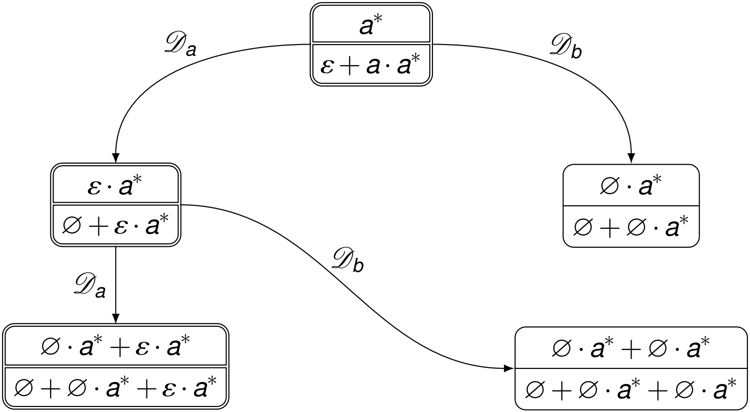
Example: $a^* \stackrel{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



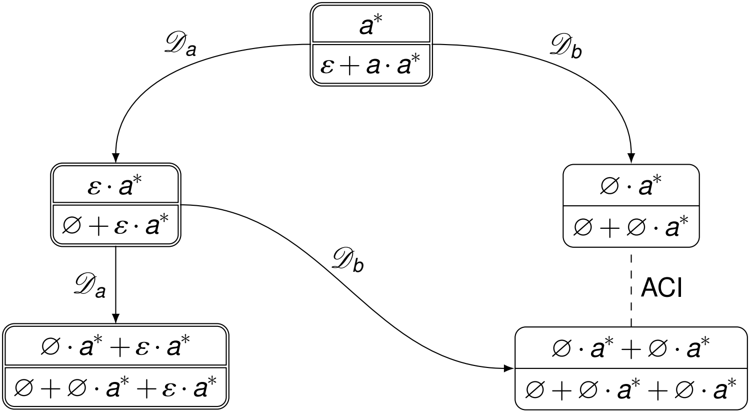
Example: $a^* \stackrel{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



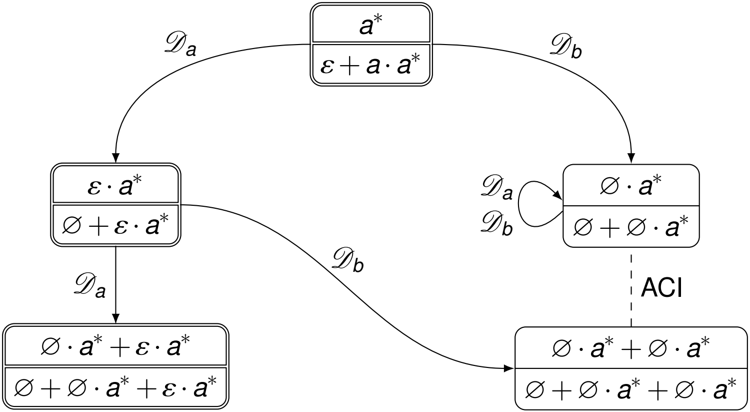
Example: $a^* \stackrel{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



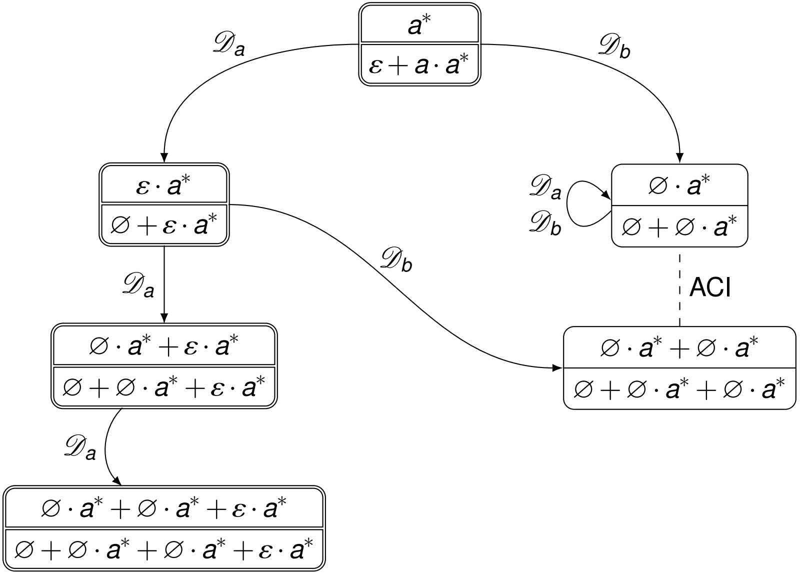
Example: $a^* \stackrel{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



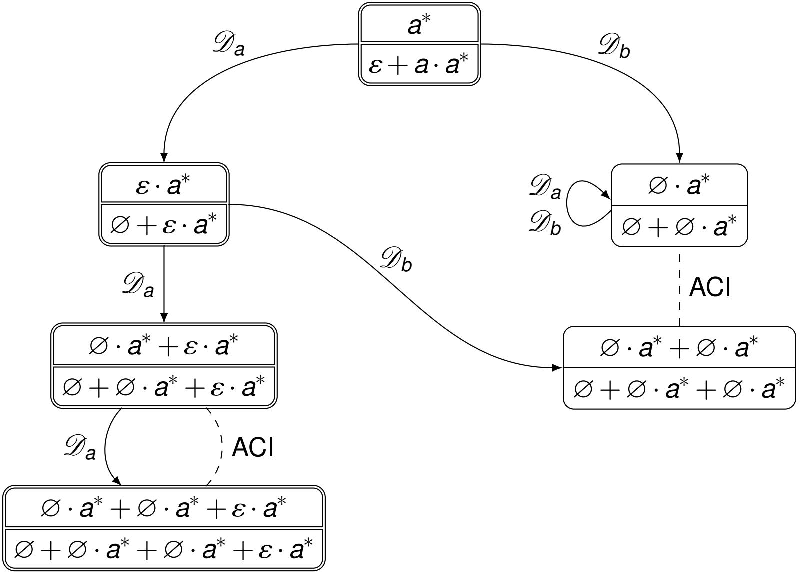
Example: $a^* \stackrel{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



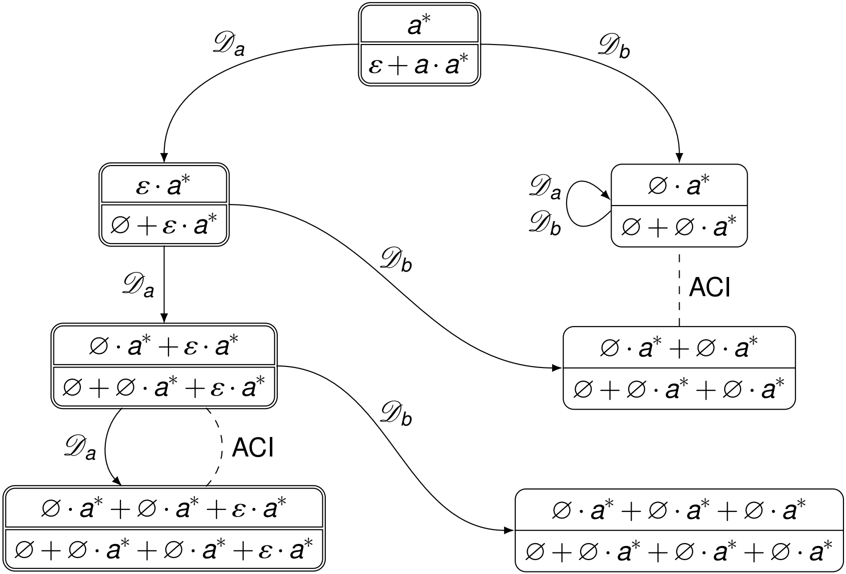
Example: $a^* \stackrel{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



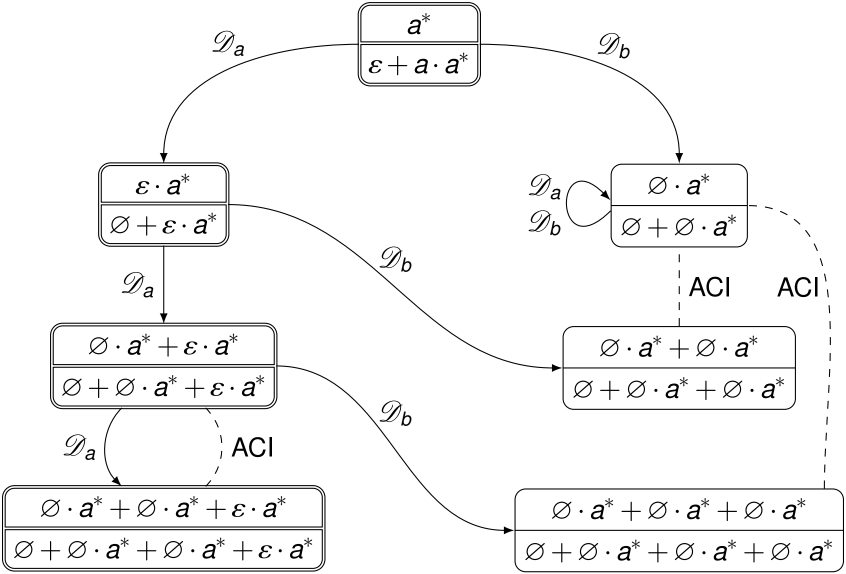
Example: $a^* \stackrel{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



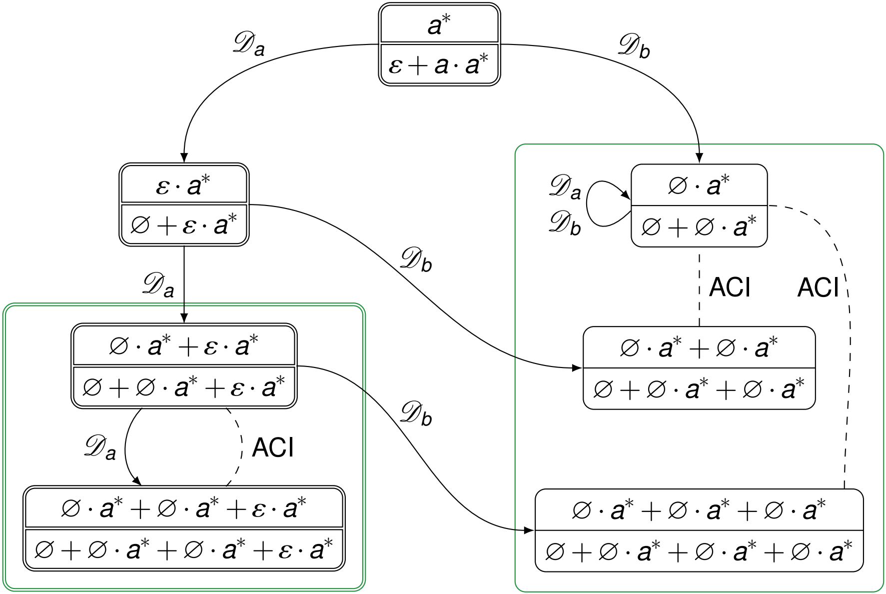
Example: $a^* \stackrel{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



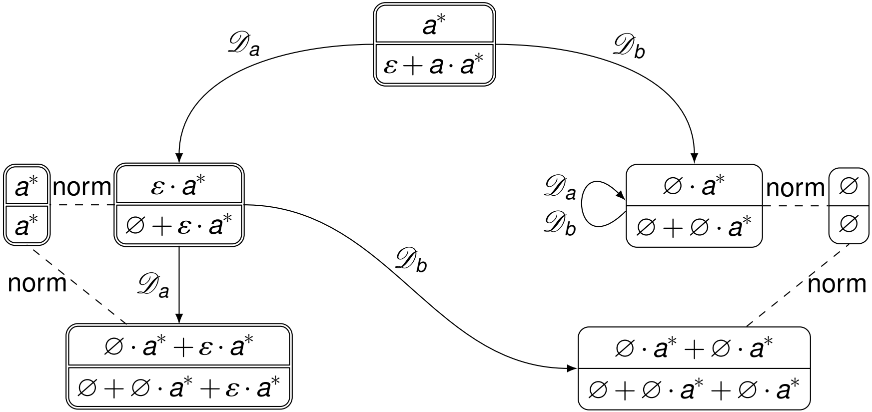
Example: $a^* \stackrel{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



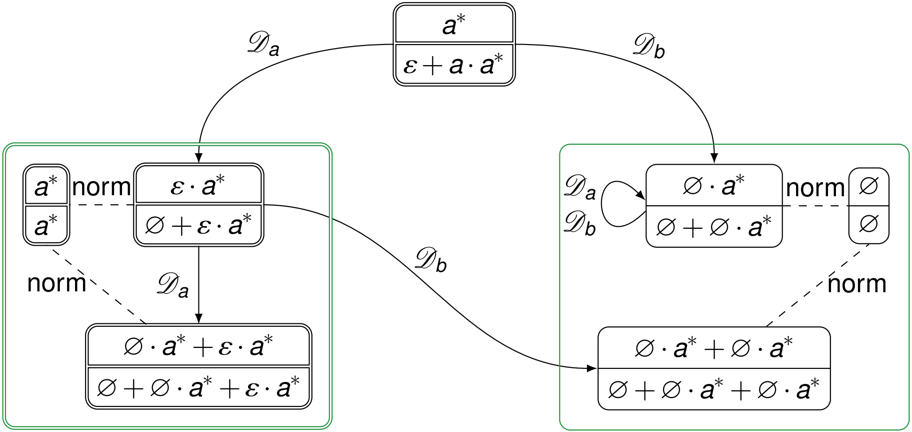
Example: $a^* \stackrel{?}{\equiv} \varepsilon + a \cdot a^*$ for $\Sigma = \{a, b\}$



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



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



Outline

Regular Expressions Equivalence

MSO

Syntax of MSO

$$
\begin{array}{rcl}
\texttt{formula} & = & Q_a(x) \\
& | & x < y \\
& | & x \in X \\
& | & \neg\ \texttt{formula} \\
& | & \texttt{formula} \vee \texttt{formula} \\
& | & \exists x.\texttt{formula} \\
& | & \exists X.\texttt{formula}
\end{array}
$$

M2L Semantics

$$(w,\mathfrak{I}) \models \mathsf{Q}_a(x) \;\Leftrightarrow\; w[\mathfrak{I}(x)] = a$$

M2L Semantics

$$(w, \mathfrak{I}) \models \mathrm{Q}_a(x) \iff w[\mathfrak{I}(x)] = a$$

$$(w, \mathfrak{I}) \models x < y \iff \mathfrak{I}(x) < \mathfrak{I}(y)$$

M2L Semantics

$$(w,\mathfrak{I}) \models \mathrm{Q}_a(x) \;\Leftrightarrow\; w[\mathfrak{I}(x)] = a$$

$$(w,\mathfrak{I}) \models x < y \;\Leftrightarrow\; \mathfrak{I}(x) < \mathfrak{I}(y)$$

$$(w,\mathfrak{I}) \models x \in X \;\Leftrightarrow\; \mathfrak{I}(x) \in \mathfrak{I}(X)$$

M2L Semantics

$$(w,\mathfrak{I}) \models \mathrm{Q}_a(x) \iff w[\mathfrak{I}(x)] = a$$

$$(w,\mathfrak{I}) \models x < y \iff \mathfrak{I}(x) < \mathfrak{I}(y)$$

$$(w,\mathfrak{I}) \models x \in X \iff \mathfrak{I}(x) \in \mathfrak{I}(X)$$

$$(w,\mathfrak{I}) \models \neg\varphi \iff (w,\mathfrak{I}) \not\models \varphi$$

M2L Semantics

$$
\begin{aligned}
(w,\mathfrak{I}) \models \mathrm{Q}_a(x) &\Leftrightarrow w[\mathfrak{I}(x)] = a \\
(w,\mathfrak{I}) \models x < y &\Leftrightarrow \mathfrak{I}(x) < \mathfrak{I}(y) \\
(w,\mathfrak{I}) \models x \in X &\Leftrightarrow \mathfrak{I}(x) \in \mathfrak{I}(X) \\
(w,\mathfrak{I}) \models \neg\varphi &\Leftrightarrow (w,\mathfrak{I}) \not\models \varphi \\
(w,\mathfrak{I}) \models \varphi \vee \psi &\Leftrightarrow (w,\mathfrak{I}) \models \varphi \vee (w,\mathfrak{I}) \models \psi
\end{aligned}
$$

M2L Semantics

$$
\begin{aligned}
(w,\mathfrak{I}) \models \mathrm{Q}_a(x) &\Leftrightarrow w[\mathfrak{I}(x)] = a \\
(w,\mathfrak{I}) \models x < y &\Leftrightarrow \mathfrak{I}(x) < \mathfrak{I}(y) \\
(w,\mathfrak{I}) \models x \in X &\Leftrightarrow \mathfrak{I}(x) \in \mathfrak{I}(X) \\
(w,\mathfrak{I}) \models \neg\varphi &\Leftrightarrow (w,\mathfrak{I}) \not\models \varphi \\
(w,\mathfrak{I}) \models \varphi \vee \psi &\Leftrightarrow (w,\mathfrak{I}) \models \varphi \vee (w,\mathfrak{I}) \models \psi \\
(w,\mathfrak{I}) \models \exists x.\varphi &\Leftrightarrow \text{there exists a } p \in \{1,\ldots,|w|\} \text{ s.t.} \\
&\ (w,\mathfrak{I}(x := p)) \models \varphi
\end{aligned}
$$

M2L Semantics

$$
\begin{aligned}
(w,\mathfrak{I}) \models \mathrm{Q}_a(x) &\Leftrightarrow w[\mathfrak{I}(x)] = a \\
(w,\mathfrak{I}) \models x < y &\Leftrightarrow \mathfrak{I}(x) < \mathfrak{I}(y) \\
(w,\mathfrak{I}) \models x \in X &\Leftrightarrow \mathfrak{I}(x) \in \mathfrak{I}(X) \\
(w,\mathfrak{I}) \models \neg\varphi &\Leftrightarrow (w,\mathfrak{I}) \not\models \varphi \\
(w,\mathfrak{I}) \models \varphi \vee \psi &\Leftrightarrow (w,\mathfrak{I}) \models \varphi \vee (w,\mathfrak{I}) \models \psi \\
(w,\mathfrak{I}) \models \exists x.\varphi &\Leftrightarrow \text{there exists a } p \in \{1,\ldots,|w|\} \text{ s.t.} \\
& (w,\mathfrak{I}(x := p)) \models \varphi \\
(w,\mathfrak{I}) \models \exists X.\varphi &\Leftrightarrow \text{there exists a } P \subseteq \{1,\ldots,|w|\} \text{ s.t.} \\
& (w,\mathfrak{I}(X := P)) \models \varphi
\end{aligned}
$$

M2L Semantics

$$
\begin{aligned}
(w,\mathfrak{I}) \models \mathrm{Q}_a(x) &\Leftrightarrow w[\mathfrak{I}(x)] = a \\
(w,\mathfrak{I}) \models x < y &\Leftrightarrow \mathfrak{I}(x) < \mathfrak{I}(y) \\
(w,\mathfrak{I}) \models x \in X &\Leftrightarrow \mathfrak{I}(x) \in \mathfrak{I}(X) \\
(w,\mathfrak{I}) \models \neg\varphi &\Leftrightarrow (w,\mathfrak{I}) \not\models \varphi \\
(w,\mathfrak{I}) \models \varphi \vee \psi &\Leftrightarrow (w,\mathfrak{I}) \models \varphi \vee (w,\mathfrak{I}) \models \psi \\
(w,\mathfrak{I}) \models \exists x.\varphi &\Leftrightarrow \text{there exists a } p \in \{1,\ldots,|w|\} \text{ s.t.} \\
&\ (w,\mathfrak{I}(x := p)) \models \varphi \\
(w,\mathfrak{I}) \models \exists X.\varphi &\Leftrightarrow \text{there exists a } P \subseteq \{1,\ldots,|w|\} \text{ s.t.} \\
&\ (w,\mathfrak{I}(X := P)) \models \varphi
\end{aligned}
$$

$$\mathscr{L}_{\mathrm{M2L}}(\varphi) = \{\mathsf{enc}(w,\mathfrak{I}) \mid (w,\mathfrak{I}) \models \varphi\}$$

Representation of Interpretations as Words

$$\left(w = aba, \qquad \mathfrak{I} = \{x \mapsto 1,\ y \mapsto 3,\ X \mapsto \{1,2\}\}\right)$$

Representation of Interpretations as Words

$$\big(w = aba, \qquad \mathfrak{I} = \{x \mapsto 1,\ y \mapsto 3,\ X \mapsto \{1,2\}\}\big)$$

$$\Big\downarrow \ \text{enc}$$

$$\Sigma_n = \Sigma \times \{0,1\}^n$$

	a	b	a
x	1	0	0
y	0	0	1
X	1	1	0

Representation of Interpretations as Words

$$(w = aba, \qquad \mathfrak{I} = \{x \mapsto 1,\ y \mapsto 3,\ X \mapsto \{1,2\}\})$$

$\downarrow$ enc

$$\Sigma_n = \Sigma \times \{0,1\}^n$$

	a	b	a
x	1	0	0
y	0	0	1
X	1	1	0

$$\begin{aligned}
\pi\,(a, bs) &= (a, \texttt{tail}\ bs) \\
\pi^{-1}(a, bs) &= \{(a, bs') \mid \texttt{tail}\ bs' = bs\} \\
&= \{(a, 0bs), (a, 1bs)\}
\end{aligned}$$

From MSO Formulas to Regular Expressions

$$\texttt{rexp\_of}\; n\,(\mathsf{Q}_a(m)) \;=\; \Sigma_n^* \cdot \begin{pmatrix} a \\ {}^0\!/_1 \\ 1 \\ {}^0\!/_1 \end{pmatrix} \cdot \Sigma_n^* \cap \texttt{WF}\; n\,(\mathsf{Q}_a(m))$$

From MSO Formulas to Regular Expressions

$$\texttt{rexp\_of}\ n\,(\mathrm{Q}_a(m)) \;=\; \Sigma_n^* \cdot \begin{pmatrix} a \\ {}^0\!/_1 \\ 1 \\ {}^0\!/_1 \end{pmatrix} \cdot \Sigma_n^* \cap \texttt{WF}\ n\,(\mathrm{Q}_a(m))$$

$$\vdots$$

$$\texttt{rexp\_of}\ n\,(\varphi_1 \vee \varphi_2) \;=\; (\texttt{rexp\_of}\ n\,\varphi_1 + \texttt{rexp\_of}\ n\,\varphi_2) \cap \texttt{WF}\ n\,(\varphi_1 \vee \varphi_2)$$

From MSO Formulas to Regular Expressions

$$\begin{aligned}
\mathtt{rexp_of}\; n\,(\mathrm{Q}_a(m)) &= \Sigma_n^* \cdot \begin{pmatrix} a \\ {}^0/_1 \\ 1 \\ {}^0/_1 \end{pmatrix} \cdot \Sigma_n^* \cap \mathtt{WF}\; n\,(\mathrm{Q}_a(m)) \\
&\vdots \\
\mathtt{rexp_of}\; n\,(\varphi_1 \vee \varphi_2) &= (\mathtt{rexp_of}\; n\,\varphi_1 + \mathtt{rexp_of}\; n\,\varphi_2) \cap \mathtt{WF}\; n\,(\varphi_1 \vee \varphi_2) \\
&\vdots \\
\mathtt{rexp_of}\; n\,(\exists x.\varphi) &= \Pi\,(\mathtt{rexp_of}\,(n+1)\,\varphi) \\
\mathtt{rexp_of}\; n\,(\exists X.\varphi) &= \Pi\,(\mathtt{rexp_of}\,(n+1)\,\varphi)
\end{aligned}$$

From MSO Formulas to Regular Expressions

$$
\begin{aligned}
\texttt{rexp\_of}\; n\, (\mathrm{Q}_a(m)) &= \Sigma_n^* \cdot \begin{pmatrix} a \\ {}^0/_1 \\ 1 \\ {}^0/_1 \end{pmatrix} \cdot \Sigma_n^* \\
&\vdots \\
\texttt{rexp\_of}\; n\, (\varphi_1 \vee \varphi_2) &= \texttt{rexp\_of}\; n\, \varphi_1 + \texttt{rexp\_of}\; n\, \varphi_2 \\
&\vdots \\
\texttt{rexp\_of}\; n\, (\exists x.\varphi) &= \Pi\, (\texttt{rexp\_of}\, (n+1)\, \varphi \cap \texttt{WF}\, (n+1)\, \varphi) \\
\texttt{rexp\_of}\; n\, (\exists X.\varphi) &= \Pi\, (\texttt{rexp\_of}\, (n+1)\, \varphi \cap \texttt{WF}\, (n+1)\, \varphi)
\end{aligned}
$$

From MSO Formulas to Regular Expressions

$$
\begin{aligned}
\texttt{rexp\_of}\ n\,(\mathrm{Q}_a(m)) &= \Sigma_n^* \cdot \begin{pmatrix} a \\ {}^0\!/_1 \\ 1 \\ {}^0\!/_1 \end{pmatrix} \cdot \Sigma_n^* \\
&\ \vdots \\
\texttt{rexp\_of}\ n\,(\varphi_1 \vee \varphi_2) &= \texttt{rexp\_of}\ n\,\varphi_1 + \texttt{rexp\_of}\ n\,\varphi_2 \\
&\ \vdots \\
\texttt{rexp\_of}\ n\,(\exists x.\varphi) &= \Pi\,(\texttt{rexp\_of}\,(n+1)\,\varphi \cap \texttt{WF}\,(n+1)\,\varphi) \\
\texttt{rexp\_of}\ n\,(\exists X.\varphi) &= \Pi\,(\texttt{rexp\_of}\,(n+1)\,\varphi \cap \texttt{WF}\,(n+1)\,\varphi)
\end{aligned}
$$

Theorem

$$\mathscr{L}_{\mathrm{M2L}}(\varphi) = \mathscr{L}_n(\texttt{rexp\_of}\ n\,\varphi \cap \texttt{WF}\ n\,\varphi) - \{\varepsilon\}$$

Future Plans

- Optimizations (use BDDs)
- Verified DP for S1S based on ω-regular expressions
- Verified DP for (W)S2S

Future Plans

- Optimizations (use BDDs)
- Verified DP for S1S based on ω-regular expressions
- Verified DP for (W)S2S

Thanks for listening!

A Verified Decision Procedure

for

Monadic Second-Order Logic on Strings

Dmitriy Traytel   Tobias Nipkow

Technische Universität München