# Characterizing Conclusive Approximations by Logical Formulae

RP 2011

Y. Boichut, B. Dao and V. Murat

Genova, DISI, 09/29/2011

# Contents

# Contents

# Term Reachability Analysis as a Verification Technique

Rewriting techniques used for

- Verification of Java programs : [5]
- Verification of security protocols : [3,13,15]
- Communication protocols : [2]

State of the studied system = Term

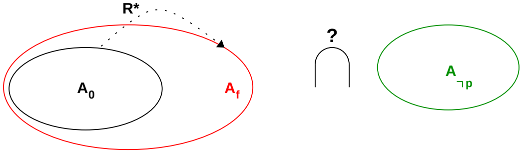
Sets of terms specified by tree automata languages

Behavior of the system = Rewriting relation

2 general approaches : Exact computation or approximated computation of rewriting successors

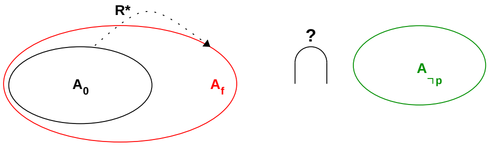
# Verifying a System $\Sigma$ using Exact Computations

- $\mathcal{R}$ : TRS or tree transducer
- Computation of theoretical fix-point automaton $\mathcal{A}_f$ from an initial one $\mathcal{A}_0$ representing **initial configurations** of $\Sigma$
- Verifying a property $p$
    - $\mathcal{A}_{\neg p}$ : set of bad configuration **forbidden terms**
    - $\mathcal{L}(\mathcal{A}_f) \bigcap \mathcal{L}(\mathcal{A}_{\neg p}) = \emptyset$

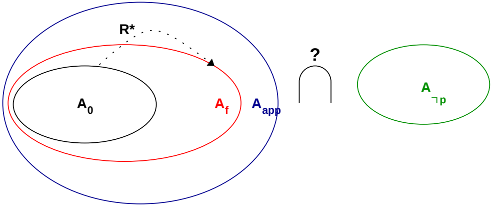



$$\mathcal{L}(\mathcal{A}_0) \subseteq \mathcal{L}(\mathcal{A}_1) \subseteq \ldots \subseteq \mathcal{L}(\mathcal{A}_f)$$

# Verifying a System Σ using Approximations

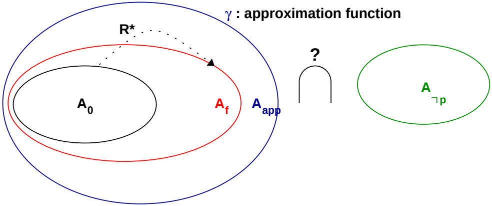


# Verifying a System Σ using Approximations



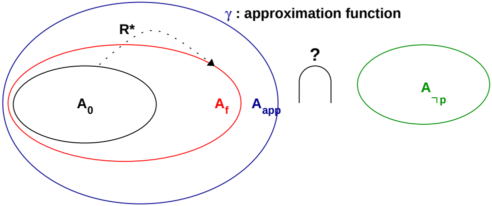

- Computation of an automaton $\mathcal{A}_{app}$ : $\mathcal{L}(\mathcal{A}_f) \subseteq \mathcal{L}(\mathcal{A}_{app})$

# Verifying a System Σ using Approximations




- Computation of an automaton $\mathcal{A}_{app}$ : $\mathcal{L}(\mathcal{A}_f) \subseteq \mathcal{L}(\mathcal{A}_{app})$
- Fine grained approximation functions for precise approximations

# Verifying a System Σ using Approximations




- Computation of an automaton $\mathcal{A}_{app} : \mathcal{L}(\mathcal{A}_f) \subseteq \mathcal{L}(\mathcal{A}_{app})$
- Fine grained approximation functions for precise approximations
- High level Expertise is required

# To summarize

- Verification of $\Sigma$ can be done via computation of **rewriting approximations**

# To summarize

- Verification of $\Sigma$ can be done via computation of **rewriting approximations**
- High level expertise is required for **conclusive approximations**

# To summarize

- Verification of $\Sigma$ can be done via computation of **rewriting approximations**
- High level expertise is required for **conclusive approximations**
- Question : Can we kick out these expertise ?

# To summarize

- Verification of $\Sigma$ can be done via computation of **rewriting approximations**
- High level expertise is required for **conclusive approximations**
- Question : Can we kick out these expertise ?
- First and naive answer : YES WE CAN...

# To summarize

- Verification of $\Sigma$ can be done via computation of **rewriting approximations**
- High level expertise is required for **conclusive approximations**
- Question : Can we kick out these expertise ?
- First and naive answer : YES WE CAN. . . **Exhaustive exploration** of the computable automata is sufficient. . .

# To summarize

- Verification of $\Sigma$ can be done via computation of **rewriting approximations**
- High level expertise is required for **conclusive approximations**
- Question : Can we kick out these expertise ?
- First and naive answer : YES WE CAN. . . **Exhaustive exploration** of the computable automata is sufficient. . . but intractable

# To summarize

- Verification of $\Sigma$ can be done via computation of **rewriting approximations**
- High level expertise is required for **conclusive approximations**
- Question : Can we kick out these expertise ?
- First and naive answer : YES WE CAN... **Exhaustive exploration** of the computable automata is sufficient... but intractable
- Second and more intricate answer :

# To summarize

- Verification of $\Sigma$ can be done via computation of **rewriting approximations**
- High level expertise is required for **conclusive approximations**
- Question : Can we kick out these expertise ?
- First and naive answer : YES WE CAN... **Exhaustive exploration** of the computable automata is sufficient... but intractable
- Second and more intricate answer : **characterize conclusive approximations** by logical formulae and solve them

# Contents

# Tree Automaton and Term Recognition

Example $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$\Sigma = \{a : 0, b : 0, g : 1, f : 2\}$

$\mathcal{Q} = \{q_1, q_2, q_3, q_f\}$, $\mathcal{F} = \{q_f\}$

$$\delta = \{ \quad a \to q_1$$
$$g(q_1) \to q_2$$
$$b \to q_3$$
$$f(q_2, q_3) \to q_f \quad \}$$

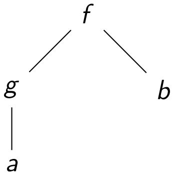
# Tree Automaton and Term Recognition

Example $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$\Sigma = \{a : 0, b : 0, g : 1, f : 2\}$

$\mathcal{Q} = \{q_1, q_2, q_3, q_f\}$, $\mathcal{F} = \{q_f\}$

$$\delta = \{ \quad a \rightarrow q_1,\quad g(q_1) \rightarrow q_2,\quad b \rightarrow q_3,\quad f(q_2, q_3) \rightarrow q_f \quad \}$$



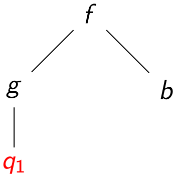
# Tree Automaton and Term Recognition

Example $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$\Sigma = \{a : 0, b : 0, g : 1, f : 2\}$

$\mathcal{Q} = \{q_1, q_2, q_3, q_f\}$, $\mathcal{F} = \{q_f\}$

$$\delta = \{ \quad a \to q_1 \\ g(q_1) \to q_2 \\ b \to q_3 \\ f(q_2, q_3) \to q_f \quad \}$$



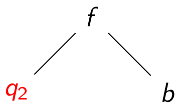
# Tree Automaton and Term Recognition

Example $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$\Sigma = \{a : 0, b : 0, g : 1, f : 2\}$

$\mathcal{Q} = \{q_1, q_2, q_3, q_f\}$, $\mathcal{F} = \{q_f\}$

$$\delta = \{ \quad a \rightarrow q_1 \\ g(q_1) \rightarrow q_2 \\ b \rightarrow q_3 \\ f(q_2, q_3) \rightarrow q_f \quad \}$$



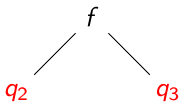
# Tree Automaton and Term Recognition

Example $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$\Sigma = \{a : 0, b : 0, g : 1, f : 2\}$

$\mathcal{Q} = \{q_1, q_2, q_3, q_f\}$, $\mathcal{F} = \{q_f\}$

$$\delta = \{ \quad a \to q_1,\quad g(q_1) \to q_2,\quad b \to q_3,\quad f(q_2, q_3) \to q_f \quad \}$$



# Tree Automaton and Term Recognition

Example $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$\Sigma=\{a:0, b:0, g:1, f:2\}$

$\mathcal{Q}=\{q_1, q_2, q_3, q_f\}$, $\mathcal{F}=\{q_f\}$

$\delta=\{ \quad a \rightarrow q_1$
$g(q_1) \rightarrow q_2$
$b \rightarrow q_3$
$f(q_2, q_3) \rightarrow q_f \quad \}$

$q_f$

# Tree Automaton and Term Recognition

Example $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$\Sigma = \{a : 0, b : 0, g : 1, f : 2\}$

$\mathcal{Q} = \{q_1, q_2, q_3, q_f\}$, $\mathcal{F} = \{q_f\}$

$$\delta = \{ \quad a \rightarrow q_1, \quad g(q_1) \rightarrow q_2, \quad b \rightarrow q_3, \quad f(q_2, q_3) \rightarrow q_f \quad \}$$

$q_f$

$f(g(a), b) \rightarrow^*_{\mathcal{A}} q_f$

# Tree Automaton and Term Recognition

Example $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$\Sigma = \{a:0, b:0, g:1, f:2\}$

$\mathcal{Q} = \{q_1, q_2, q_3, q_f\}$, $\mathcal{F} = \{q_f\}$

$$\delta = \{ \quad a \rightarrow q_1, \quad g(q_1) \rightarrow q_2, \quad b \rightarrow q_3, \quad f(q_2, q_3) \rightarrow q_f \quad \}$$

$q_f$

$f(g(a), b) \rightarrow^*_{\mathcal{A}} q_f$

Language of $\mathcal{A}$

$\mathcal{L}(\mathcal{A}) = \{t | t \rightarrow^*_{\mathcal{A}} q_f \wedge q_f \in F\}$

$\mathcal{L}(\mathcal{A}, q) = \{t | t \rightarrow^*_{\mathcal{A}} q\}$

# Does a Pattern have a Solution in $\mathcal{A}$?

- Given a term $t$ containing variables and $\mathcal{A}$, a solution of $t$ on the state $q \in \mathcal{Q}$ is a substitution $\sigma : \mathcal{X} \mapsto \mathcal{Q}$ such that

$$t\sigma \rightarrow^*_{\mathcal{A}} q$$

# Does a Pattern have a Solution in $\mathcal{A}$?

- Given a term $t$ containing variables and $\mathcal{A}$, a solution of $t$ on the state $q \in \mathcal{Q}$ is a substitution $\sigma : \mathcal{X} \mapsto \mathcal{Q}$ such that

$$t\sigma \rightarrow^*_{\mathcal{A}} q$$

Example : $t = f(g(x), b)$ and
$\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$$\delta = \{ \quad a \rightarrow q_1, \quad g(q_1) \rightarrow q_2, \quad b \rightarrow q_3, \quad f(q_2, q_3) \rightarrow q_f \quad \}$$

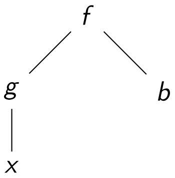
# Does a Pattern have a Solution in $\mathcal{A}$ ?

- Given a term $t$ containing variables and $\mathcal{A}$, a solution of $t$ on the state $q \in \mathcal{Q}$ is a substitution $\sigma : \mathcal{X} \mapsto \mathcal{Q}$ such that

$$t\sigma \rightarrow^*_{\mathcal{A}} q$$

Example : $t = f(g(x), b)$ and
$\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$$\delta = \{ \quad a \rightarrow q_1, \quad g(q_1) \rightarrow q_2, \quad b \rightarrow q_3, \quad f(q_2, q_3) \rightarrow q_f \quad \}$$




**An example of solution**

$\sigma = \{x \mapsto q_1\}$ is a solution of $t$ on $q_f$

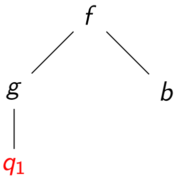
# Does a Pattern have a Solution in $\mathcal{A}$?

- Given a term $t$ containing variables and $\mathcal{A}$, a solution of $t$ on the state $q \in \mathcal{Q}$ is a substitution $\sigma : \mathcal{X} \mapsto \mathcal{Q}$ such that

$$t\sigma \to^*_{\mathcal{A}} q$$

Example : $t = f(g(x), b)$ and $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$$\delta = \{ \quad a \to q_1,\ g(q_1) \to q_2,\ b \to q_3,\ f(q_2, q_3) \to q_f \quad \}$$




**An example of solution**

$\sigma = \{x \mapsto q_1\}$ is a solution of $t$ on $q_f$

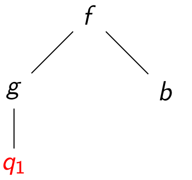
# Does a Pattern have a Solution in $\mathcal{A}$?

- Given a term $t$ containing variables and $\mathcal{A}$, a solution of $t$ on the state $q \in \mathcal{Q}$ is a substitution $\sigma : \mathcal{X} \mapsto \mathcal{Q}$ such that

$$t\sigma \rightarrow^*_{\mathcal{A}} q$$

Example : $t = f(g(x), b)$ and $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$$\delta = \{ \quad a \rightarrow q_1 \quad g(q_1) \rightarrow q_2 \quad b \rightarrow q_3 \quad f(q_2, q_3) \rightarrow q_f \quad \}$$




**An example of solution**

$\sigma = \{x \mapsto q_1\}$ is a solution of $t$ on $q_f$

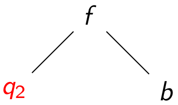
# Does a Pattern have a Solution in $\mathcal{A}$?

- Given a term $t$ containing variables and $\mathcal{A}$, a solution of $t$ on the state $q \in \mathcal{Q}$ is a substitution $\sigma : \mathcal{X} \mapsto \mathcal{Q}$ such that

$$t\sigma \rightarrow^*_{\mathcal{A}} q$$

Example : $t = f(g(x), b)$ and
$\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$$\delta = \{ \begin{array}{l} a \rightarrow q_1 \\ g(q_1) \rightarrow q_2 \\ b \rightarrow q_3 \\ f(q_2, q_3) \rightarrow q_f \end{array} \}$$




**An example of solution**

$\sigma = \{x \mapsto q_1\}$ is a solution of $t$ on $q_f$

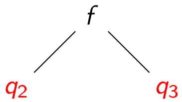
# Does a Pattern have a Solution in $\mathcal{A}$?

- Given a term $t$ containing variables and $\mathcal{A}$, a solution of $t$ on the state $q \in \mathcal{Q}$ is a substitution $\sigma : \mathcal{X} \mapsto \mathcal{Q}$ such that

$$t\sigma \rightarrow^*_{\mathcal{A}} q$$

Example : $t = f(g(x), b)$ and
$\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$$\delta = \{ \quad a \rightarrow q_1, \quad g(q_1) \rightarrow q_2, \quad b \rightarrow q_3, \quad f(q_2, q_3) \rightarrow q_f \quad \}$$




**An example of solution**

$\sigma = \{x \mapsto q_1\}$ is a solution of $t$ on $q_f$

# Does a Pattern have a Solution in $\mathcal{A}$?

- Given a term $t$ containing variables and $\mathcal{A}$, a solution of $t$ on the state $q \in \mathcal{Q}$ is a substitution $\sigma : \mathcal{X} \mapsto \mathcal{Q}$ such that

$$t\sigma \rightarrow^*_{\mathcal{A}} q$$

Example : $t = f(g(x), b)$ and $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$$\delta = \{ \quad a \rightarrow q_1 \quad g(q_1) \rightarrow q_2 \quad b \rightarrow q_3 \quad f(q_2, q_3) \rightarrow q_f \quad \}$$

$q_f$

**An example of solution**

$\sigma = \{x \mapsto q_1\}$ is a solution of $t$ on $q_f$

# Does a Pattern have a Solution in $\mathcal{A}$?

- Given a term $t$ containing variables and $\mathcal{A}$, a solution of $t$ on the state $q \in \mathcal{Q}$ is a substitution $\sigma : \mathcal{X} \mapsto \mathcal{Q}$ such that

$$t\sigma \rightarrow^*_{\mathcal{A}} q$$

Example : $t = f(g(x), b)$ and $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \mathcal{F}, \delta \rangle$ with

$$\delta = \{ \quad a \rightarrow q_1, \quad g(q_1) \rightarrow q_2, \quad b \rightarrow q_3, \quad f(q_2, q_3) \rightarrow q_f \quad \}$$

$q_f$

$f(g(q_1), b) \rightarrow^*_{\mathcal{A}} q_f$

**An example of solution**

$\sigma = \{x \mapsto q_1\}$ is a solution of $t$ on $q_f$

# Rewriting Approximations

Inputs

- $\mathcal{A}_0$ : an initial tree automaton
- $\mathcal{A}$ : an automaton such that $\mathcal{A}_0 \subseteq \mathcal{A}$
- $\mathcal{R}$ : a set of rewrite rules $l \rightarrow r$

# Rewriting Approximations

Inputs

- $\mathcal{A}_0$ : an initial tree automaton
- $\mathcal{A}$ : an automaton such that $\mathcal{A}_0 \subseteq \mathcal{A}$
- $\mathcal{R}$ : a set of rewrite rules $l \rightarrow r$

If for any $l \rightarrow r \in \mathcal{R}$, any $q$ of $\mathcal{A}$ one has

any solution $\sigma$ of $l$ on the state $q \Rightarrow \sigma$ is a solution of $r$ on $q$

then

# Rewriting Approximations

Inputs

- $\mathcal{A}_0$ : an initial tree automaton
- $\mathcal{A}$ : an automaton such that $\mathcal{A}_0 \subseteq \mathcal{A}$
- $\mathcal{R}$ : a set of rewrite rules $l \rightarrow r$

If for any $l \rightarrow r \in \mathcal{R}$, any $q$ of $\mathcal{A}$ one has

any solution $\sigma$ of $l$ on the state $q$ $\Rightarrow$ $\sigma$ is a solution of $r$ on $q$

then

# Rewriting Approximations

Inputs

- $\mathcal{A}_0$ : an initial tree automaton
- $\mathcal{A}$ : an automaton such that $\mathcal{A}_0 \subseteq \mathcal{A}$
- $\mathcal{R}$ : a set of rewrite rules $l \rightarrow r$

If for any $l \rightarrow r \in \mathcal{R}$, any $q$ of $\mathcal{A}$ one has

any solution $\sigma$ of $l$ on the state $q \Rightarrow \sigma$ is a solution of $r$ on $q$

then

# Rewriting Approximations

Inputs

- $\mathcal{A}_0$ : an initial tree automaton
- $\mathcal{A}$ : an automaton such that $\mathcal{A}_0 \subseteq \mathcal{A}$
- $\mathcal{R}$ : a set of rewrite rules $l \rightarrow r$

If for any $l \rightarrow r \in \mathcal{R}$, any $q$ of $\mathcal{A}$ one has

any solution $\sigma$ of $l$ on the state $q \Rightarrow \sigma$ is a solution of $r$ on $q$

then

$$\mathcal{R}(\mathcal{L}(\mathcal{A})) \subseteq \mathcal{L}(\mathcal{A}) \ (\mathcal{A} \ is \ \mathcal{R}-closed)$$

# Rewriting Approximations

Inputs

- $\mathcal{A}_0$ : an initial tree automaton
- $\mathcal{A}$ : an automaton such that $\mathcal{A}_0 \subseteq \mathcal{A}$
- $\mathcal{R}$ : a set of rewrite rules $l \rightarrow r$

If for any $l \rightarrow r \in \mathcal{R}$, any $q$ of $\mathcal{A}$ one has

any solution $\sigma$ of $l$ on the state $q \Rightarrow \sigma$ is a solution of $r$ on $q$

then

$$\mathcal{R}(\mathcal{L}(\mathcal{A})) \subseteq \mathcal{L}(\mathcal{A}) \ (\mathcal{A} \ is \ \mathcal{R} - closed)$$

In other words

$$\mathcal{R}^*(\mathcal{L}(\mathcal{A}_0)) \subseteq \mathcal{L}(\mathcal{A})$$

# Rewriting Approximations

Inputs

- $\mathcal{A}_0$ : an initial tree automaton
- $\mathcal{A}$ : an automaton such that $\mathcal{A}_0 \subseteq \mathcal{A}$
- $\mathcal{R}$ : a set of rewrite rules $l \rightarrow r$

If for any $l \rightarrow r \in \mathcal{R}$, any $q$ of $\mathcal{A}$ one has

any solution $\sigma$ of $l$ on the state $q \Rightarrow \sigma$ is a solution of $r$ on $q$

then

$$\mathcal{R}(\mathcal{L}(\mathcal{A})) \subseteq \mathcal{L}(\mathcal{A}) \; (\mathcal{A} \textit{ is } \mathcal{R} - \textit{closed})$$

In other words

$$\mathcal{R}^*(\mathcal{L}(\mathcal{A}_0)) \subseteq \mathcal{L}(\mathcal{A})$$

Given $\mathcal{A}_0$ and $\mathcal{R}$, $\mathcal{A}$ is an $(\mathcal{A}_0, \mathcal{R})$ over-approximation

# Example of Rewriting Approximation

Inputs

- $\mathcal{A}_0 = \langle\{q_f, q_a\}, \{f:1, s:1, a:0\}, \{q_f\}, \{a \rightarrow q_a, f(q_a) \rightarrow q_f\}\rangle$
- $\mathcal{A}_f = \langle\{q_f, q_a\}, \{f:1, s:1, a:0\}, \{q_f\}, \Delta_f\rangle$ with $\Delta_f = \{a \rightarrow q_a, f(q_a) \rightarrow q_f, s(q_a) \rightarrow q_a\}$
- $\mathcal{R} = \{f(x) \rightarrow f(s(s(x)))\}$

# Example of Rewriting Approximation

Inputs

- $\mathcal{A}_0 = \langle \{q_f, q_a\}, \{f:1, s:1, a:0\}, \{q_f\}, \{a \rightarrow q_a, f(q_a) \rightarrow q_f\} \rangle$
- $\mathcal{A}_f = \langle \{q_f, q_a\}, \{f:1, s:1, a:0\}, \{q_f\}, \Delta_f \rangle$ with $\Delta_f = \{a \rightarrow q_a, f(q_a) \rightarrow q_f, s(q_a) \rightarrow q_a\}$
- $\mathcal{R} = \{f(x) \rightarrow f(s(s(x)))\}$

$\sigma = \{x \rightarrow q_a\}$ is the unique solution of $f(x)$ on $q_f$ and $\sigma$ is also a solution for $f(s(s(x)))$

# Example of Rewriting Approximation

Inputs

- $\mathcal{A}_0 = \langle \{q_f, q_a\}, \{f:1, s:1, a:0\}, \{q_f\}, \{a \to q_a, f(q_a) \to q_f\} \rangle$
- $\mathcal{A}_f = \langle \{q_f, q_a\}, \{f:1, s:1, a:0\}, \{q_f\}, \Delta_f \rangle$ with $\Delta_f = \{a \to q_a, f(q_a) \to q_f, s(q_a) \to q_a\}$
- $\mathcal{R} = \{f(x) \to f(s(s(x)))\}$

$\sigma = \{x \to q_a\}$ is the unique solution of $f(x)$ on $q_f$ and $\sigma$ is also a solution for $f(s(s(x)))$

$$f(a) \to_{\mathcal{R}} f(s(s(a))) \to f(s^4(a)) \ldots$$

# Example of Rewriting Approximation

Inputs

- $\mathcal{A}_0 = \langle\{q_f, q_a\}, \{f:1, s:1, a:0\}, \{q_f\}, \{a \rightarrow q_a, f(q_a) \rightarrow q_f\}\rangle$
- $\mathcal{A}_f = \langle\{q_f, q_a\}, \{f:1, s:1, a:0\}, \{q_f\}, \Delta_f\rangle$ with $\Delta_f = \{a \rightarrow q_a, f(q_a) \rightarrow q_f, s(q_a) \rightarrow q_a\}$
- $\mathcal{R} = \{f(x) \rightarrow f(s(s(x)))\}$

$\sigma = \{x \rightarrow q_a\}$ is the unique solution of $f(x)$ on $q_f$ and $\sigma$ is also a solution for $f(s(s(x)))$

$$f(a) \rightarrow_{\mathcal{R}} f(s(s(a))) \rightarrow f(s^4(a)) \ldots$$

$$\mathcal{R}^*(\mathcal{L}(\mathcal{A}_0)) = \{f(s^{(2n)}(a)) | n \in \mathbb{N}\}$$

# Example of Rewriting Approximation

Inputs

- $\mathcal{A}_0 = \langle\{q_f, q_a\}, \{f:1, s:1, a:0\}, \{q_f\}, \{a \rightarrow q_a, f(q_a) \rightarrow q_f\}\rangle$
- $\mathcal{A}_f = \langle\{q_f, q_a\}, \{f:1, s:1, a:0\}, \{q_f\}, \Delta_f\rangle$ with $\Delta_f = \{a \rightarrow q_a, f(q_a) \rightarrow q_f, s(q_a) \rightarrow q_a\}$
- $\mathcal{R} = \{f(x) \rightarrow f(s(s(x)))\}$

$\sigma = \{x \rightarrow q_a\}$ is the unique solution of $f(x)$ on $q_f$ and $\sigma$ is also a solution for $f(s(s(x)))$

$$f(a) \rightarrow_{\mathcal{R}} f(s(s(a))) \rightarrow f(s^4(a)) \dots$$

$$\mathcal{R}^*(\mathcal{L}(\mathcal{A}_0)) = \{f(s^{(2n)}(a)) | n \in \mathbb{N}\} \text{ and } \mathcal{L}(\mathcal{A}) = \{f(s^n(a) | n \in \mathbb{N}\}$$

# Contents

# Symbolic Tree Automata

$$\mathcal{A} = \{ \quad \begin{array}{l} a \rightarrow q_a \\ f(q_{s(a)}) \rightarrow q_f \\ s(q_a) \rightarrow q_{s(a)} \end{array} \quad \}$$

# Symbolic Tree Automata

$$\mathcal{A} = \{ \quad a \to q_a \quad f(q_{s(a)}) \to q_f \quad s(q_a) \to q_{s(a)} \quad \}$$

$$\mathcal{A}_{\mathcal{S}} = \{ \quad a \to X_1 \quad f(X_2) \to X_0 \quad s(X_1) \to X_2$$

# Symbolic Tree Automata

$$\mathcal{A} = \{ \quad a \to q_a \quad f(q_{s(a)}) \to q_f \quad s(q_a) \to q_{s(a)} \quad \}$$

$$\mathcal{A}_{\mathcal{S}} = \{ \quad a \to X_1 \quad f(X_2) \to X_0 \quad s(X_1) \to X_2$$

$$\iota = \{ \quad X_0 \mapsto q_f, \quad X_1 \mapsto q_a, \quad X_2 \mapsto q_{s(a)} \quad \}$$

# Symbolic Tree Automata

$$\mathcal{A} = \{ \quad a \rightarrow q_a, \quad f(q_{s(a)}) \rightarrow q_f, \quad s(q_a) \rightarrow q_{s(a)} \quad \}$$

$$\mathcal{A}_\mathcal{S} = \{ \quad a \rightarrow X_1, \quad f(X_2) \rightarrow X_0, \quad s(X_1) \rightarrow X_2$$

$$\iota = \{ \quad X_0 \mapsto q_f, \quad X_1 \mapsto q_a, \quad X_2 \mapsto q_{s(a)} \quad \}$$

**Proposition**

*Let $\mathcal{A}_\mathcal{S}$ be an STA. If $\mathcal{A} \subseteq \mathcal{A}_\mathcal{S}$ modulo renaming then*

$$\forall \iota . \mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}_\mathcal{S}\iota)$$

# Reduction seen as Formula

$$f(s(a)) \xrightarrow{\phi} X_0$$

$$\mathcal{A}_{\mathcal{S}} = \{ \quad \begin{array}{l} a \rightarrow X_1 \\ f(X_2) \rightarrow X_0 \\ s(X_1) \rightarrow X_2 \\ s(X_3) \rightarrow X_4 \\ s(X_5) \rightarrow X_6 \end{array} \quad \}$$

# Reduction seen as Formula

$$f(s(a)) \xrightarrow{\phi} X_0$$

$$\mathcal{A}_{\mathcal{S}} = \{ \quad \begin{array}{l} a \to X_1 \\ f(X_2) \to X_0 \\ s(X_1) \to X_2 \\ s(X_3) \to X_4 \\ s(X_5) \to X_6 \end{array} \quad \}$$

# Reduction seen as Formula

$$f(s(a)) \xrightarrow{\phi} X_0$$

$$f(s(a)) \xrightarrow{\phi} X_0 \quad \text{if} \quad s(a) \xrightarrow{\phi_1} X_2 \text{ and } \phi = \phi_1 \wedge X_0 = X_0$$

$$\mathcal{A}_\mathcal{S} = \{ \quad a \to X_1, \quad f(X_2) \to X_0, \quad s(X_1) \to X_2, \quad s(X_3) \to X_4, \quad s(X_5) \to X_6 \quad \}$$

# Reduction seen as Formula

$$f(s(a)) \xrightarrow{\phi} X_0$$

$$f(s(a)) \xrightarrow{\phi} X_0 \quad \text{if} \quad s(a) \xrightarrow{\phi_1} X_2$$
$$\text{and } \phi = \phi_1 \wedge X_0 = X_0$$

$$\mathcal{A}_{\mathcal{S}} = \{ \quad a \to X_1$$
$$f(X_2) \to X_0$$
$$s(X_1) \to X_2$$
$$s(X_3) \to X_4$$
$$s(X_5) \to X_6 \quad \}$$

# Reduction seen as Formula

$$f(s(a)) \xrightarrow{\phi} X_0$$

$f(s(a)) \xrightarrow{\phi} X_0$ if $s(a) \xrightarrow{\phi_1} X_2$ and $\phi = \phi_1 \wedge X_0 = X_0$

$s(a) \xrightarrow{\phi_1} X_2$ if $a \xrightarrow{\phi_2} X_1$ and $\phi_1 = \phi_2 \wedge X_2 = X_2$

$$\mathcal{A}_{\mathcal{S}} = \{ \quad a \rightarrow X_1, \; f(X_2) \rightarrow X_0, \; s(X_1) \rightarrow X_2, \; s(X_3) \rightarrow X_4, \; s(X_5) \rightarrow X_6 \quad \}$$

# Reduction seen as Formula

$$f(s(a)) \xrightarrow{\phi} X_0$$

$f(s(a)) \xrightarrow{\phi} X_0$ if $s(a) \xrightarrow{\phi_1} X_2$ and $\phi = \phi_1 \wedge X_0 = X_0$

$s(a) \xrightarrow{\phi_1} X_2$ if $a \xrightarrow{\phi_2} X_1$ and $\phi_1 = \phi_2 \wedge X_2 = X_2$

$$\mathcal{A}_\mathcal{S} = \{ \quad a \to X_1, \quad f(X_2) \to X_0, \quad s(X_1) \to X_2, \quad s(X_3) \to X_4, \quad s(X_5) \to X_6 \quad \}$$

# Reduction seen as Formula

$$f(s(a)) \xrightarrow{\phi} X_0$$

$f(s(a)) \xrightarrow{\phi} X_0$ if $s(a) \xrightarrow{\phi_1} X_2$ and $\phi = \phi_1 \wedge X_0 = X_0$

$s(a) \xrightarrow{\phi_1} X_2$ if $a \xrightarrow{\phi_2} X_1$ and $\phi_1 = \phi_2 \wedge X_2 = X_2$
OR ($a \xrightarrow{\phi_3} X_3$ and $\phi_1 = \phi_3 \wedge X_4 = X_2$)
OR

$$\mathcal{A}_\mathcal{S} = \{ \quad a \to X_1,\ f(X_2) \to X_0,\ s(X_1) \to X_2,\ s(X_3) \to X_4,\ s(X_5) \to X_6 \quad \}$$

# Reduction seen as Formula

$$\mathcal{A}_{\mathcal{S}} = \{ \quad a \to X_1,\ f(X_2) \to X_0,\ s(X_1) \to X_2,\ s(X_3) \to X_4,\ s(X_5) \to X_6 \quad \}$$

$$f(s(a)) \xrightarrow{\phi} X_0$$

$f(s(a)) \xrightarrow{\phi} X_0$ if $s(a) \xrightarrow{\phi_1} X_2$
and $\phi = \phi_1 \wedge X_0 = X_0$

$s(a) \xrightarrow{\phi_1} X_2$ if $a \xrightarrow{\phi_2} X_1$ and
$\phi_1 = \phi_2 \wedge X_2 = X_2$
OR ($a \xrightarrow{\phi_3} X_3$ and
$\phi_1 = \phi_3 \wedge X_4 = X_2$)
OR

# Reduction seen as Formula

$$f(s(a)) \xrightarrow{\phi} X_0$$

$f(s(a)) \xrightarrow{\phi} X_0$ if $s(a) \xrightarrow{\phi_1} X_2$ and $\phi = \phi_1 \wedge X_0 = X_0$

$s(a) \xrightarrow{\phi_1} X_2$ if $a \xrightarrow{\phi_2} X_1$ and $\phi_1 = \phi_2 \wedge X_2 = X_2$
OR ($a \xrightarrow{\phi_3} X_3$ and $\phi_1 = \phi_3 \wedge X_4 = X_2$)
OR
($a \xrightarrow{\phi_4} X_5$ AND $\phi_1 = \phi_4 \wedge X_6 = X_2$)

$$\mathcal{A}_\mathcal{S} = \{ \quad a \rightarrow X_1, \quad f(X_2) \rightarrow X_0, \quad s(X_1) \rightarrow X_2, \quad s(X_3) \rightarrow X_4, \quad s(X_5) \rightarrow X_6 \quad \}$$

# Reduction seen as Formula

$$\mathcal{A}_{\mathcal{S}} = \{ \quad a \to X_1,\ f(X_2) \to X_0,\ s(X_1) \to X_2,\ s(X_3) \to X_4,\ s(X_5) \to X_6 \quad \}$$

$$f(s(a)) \xrightarrow{\phi} X_0$$

$f(s(a)) \xrightarrow{\phi} X_0$ if $s(a) \xrightarrow{\phi_1} X_2$ and $\phi = \phi_1 \wedge X_0 = X_0$

$s(a) \xrightarrow{\phi_1} X_2$ if $a \xrightarrow{\phi_2} X_1$ and $\phi_1 = \phi_2 \wedge X_2 = X_2$
OR ($a \xrightarrow{\phi_3} X_3$ and $\phi_1 = \phi_3 \wedge X_4 = X_2$)
OR
($a \xrightarrow{\phi_4} X_5$ AND $\phi_1 = \phi_4 \wedge X_6 = X_2$)

# Reduction seen as Formula

$$\mathcal{A}_{\mathcal{S}} = \{ \; a \rightarrow X_1,\; f(X_2) \rightarrow X_0,\; s(X_1) \rightarrow X_2,\; s(X_3) \rightarrow X_4,\; s(X_5) \rightarrow X_6 \; \}$$

$$f(s(a)) \xrightarrow{\phi} X_0$$

$f(s(a)) \xrightarrow{\phi} X_0$ if $s(a) \xrightarrow{\phi_1} X_2$ and $\phi = \phi_1 \wedge X_0 = X_0$

$s(a) \xrightarrow{\phi_1} X_2$ if $a \xrightarrow{\phi_2} X_1$ and $\phi_1 = \phi_2 \wedge X_2 = X_2$
OR ($a \xrightarrow{\phi_3} X_3$ and $\phi_1 = \phi_3 \wedge X_4 = X_2$)
OR
($a \xrightarrow{\phi_4} X_5$ AND $\phi_1 = \phi_4 \wedge X_6 = X_2$)

$a \xrightarrow{\phi_2} X_1$ if $\phi_2 = (X_1 = X_1)$

# Reduction seen as Formula

$$f(s(a)) \xrightarrow{\phi} X_0$$

$$\mathcal{A}_{\mathcal{S}} = \{ \quad a \rightarrow X_1, \quad f(X_2) \rightarrow X_0, \quad s(X_1) \rightarrow X_2, \quad s(X_3) \rightarrow X_4, \quad s(X_5) \rightarrow X_6 \quad \}$$

$f(s(a)) \xrightarrow{\phi} X_0$ if $s(a) \xrightarrow{\phi_1} X_2$ and $\phi = \phi_1 \wedge X_0 = X_0$

$s(a) \xrightarrow{\phi_1} X_2$ if $a \xrightarrow{\phi_2} X_1$ and $\phi_1 = \phi_2 \wedge X_2 = X_2$ OR ($a \xrightarrow{\phi_3} X_3$ and $\phi_1 = \phi_3 \wedge X_4 = X_2$) OR ($a \xrightarrow{\phi_4} X_5$ AND $\phi_1 = \phi_4 \wedge X_6 = X_2$)

$a \xrightarrow{\phi_2} X_1$ if $\phi_2 = (X_1 = X_1)$

$a \xrightarrow{\phi_3} X_3$ if $\phi_3 = (X_1 = X_3)$

# Reduction seen as Formula

$$\mathcal{A}_{\mathcal{S}} = \{ \quad a \to X_1,\ f(X_2) \to X_0,\ s(X_1) \to X_2,\ s(X_3) \to X_4,\ s(X_5) \to X_6 \quad \}$$

$$f(s(a)) \xrightarrow{\phi} X_0$$

$f(s(a)) \xrightarrow{\phi} X_0$ if $s(a) \xrightarrow{\phi_1} X_2$ and $\phi = \phi_1 \wedge X_0 = X_0$

$s(a) \xrightarrow{\phi_1} X_2$ if $a \xrightarrow{\phi_2} X_1$ and $\phi_1 = \phi_2 \wedge X_2 = X_2$
OR ($a \xrightarrow{\phi_3} X_3$ and $\phi_1 = \phi_3 \wedge X_4 = X_2$)
OR
($a \xrightarrow{\phi_4} X_5$ AND $\phi_1 = \phi_4 \wedge X_6 = X_2$)

$a \xrightarrow{\phi_2} X_1$ if $\phi_2 = (X_1 = X_1)$

$a \xrightarrow{\phi_3} X_3$ if $\phi_3 = (X_1 = X_3)$

$a \xrightarrow{\phi_4} X_5$ if $\phi_4 = (X_1 = X_5)$

# Reduction seen as Formula

To summarize,

$$f(s(a)) \xrightarrow{\phi} X_0 \ \textit{with}$$

$$\begin{array}{llll} \phi = & X_0 = X_0 \wedge ( & X_2 = X_2 \vee \\ & & (X_4 = X_2 \wedge X_3 = X_1) \vee \\ & & (X_6 = X_2 \wedge X_5 = X_1)) \end{array}$$

# Reduction seen as Formula

To summarize,

$$f(s(a)) \xrightarrow{\phi} X_0 \ with$$

$$\begin{aligned}\phi = \quad X_0 = X_0 \wedge ( \quad & X_2 = X_2 \vee \\ & (X_4 = X_2 \wedge X_3 = X_1) \vee \\ & (X_6 = X_2 \wedge X_5 = X_1))\end{aligned}$$

More generally, $t \xrightarrow{\phi}_{\mathcal{A}} X$ denotes the condition under which $t$ can be recognized into $X$

# Generalization to Recognition

**Definition ($Reco_{\mathcal{A}_S}(t, X)$)**

Let $\mathcal{A}_S$ be an STA such that $\mathcal{A}_S = \langle \mathcal{X}, \Sigma, \mathcal{X}_F, \Delta \rangle$ and $t$ be a term without variables.

$$Reco_{\mathcal{A}_S}(t, X) = \bigvee_{Y \in \mathcal{X}, t \xrightarrow{\phi} Y} \phi \wedge X = Y$$

# Generalization to Recognition

**Definition ($Reco_{\mathcal{A}_S}(t, X)$)**

Let $\mathcal{A}_S$ be an STA such that $\mathcal{A}_S = \langle \mathcal{X}, \Sigma, \mathcal{X}_F, \Delta \rangle$ and $t$ be a term without variables.

$$Reco_{\mathcal{A}_S}(t, X) = \bigvee_{Y \in \mathcal{X}, t \xrightarrow{\phi} Y} \phi \wedge X = Y$$

**Definition ($Reco_{\mathcal{A}_S}(t)$)**

Let $\mathcal{A}_S$ be an STA such that $\mathcal{A}_S = \langle \mathcal{X}, \Sigma, \mathcal{X}_F, \Delta \rangle$ and $t$ be a term without variables.

$$Reco_{\mathcal{A}_S}(t) = \bigvee_{X \in \mathcal{X}_F} Reco_{\mathcal{A}_S}(t, X)$$

# Generalization to Recognition

**Proposition**

*Let $\mathcal{A}_S$ be an STA and t be a term. Let $\iota : \mathcal{X} \mapsto \mathcal{Q}$ be an instantiation of $\mathcal{A}_S$.*

$$\iota \models \; Reco_{\mathcal{A}_\mathbf{S}}(t) \text{ iff } t \in \mathcal{L}(\mathcal{A}_S \iota)$$

# Solutions of a Pattern in STA

$$\mathcal{A}_{\mathcal{S}} = \{ \quad a \rightarrow X_1 \\ f(X_2) \rightarrow X_0 \\ s(X_1) \rightarrow X_2 \\ s(X_3) \rightarrow X_4 \\ s(X_5) \rightarrow X_6 \quad \}$$

*Does $f(s(x))$ has a solution on $X_0$?*

# Solutions of a Pattern in STA

$$\mathcal{A}_{\mathcal{S}} = \{ \quad a \to X_1, \quad f(X_2) \to X_0, \quad s(X_1) \to X_2, \quad s(X_3) \to X_4, \quad s(X_5) \to X_6 \quad \}$$

*Does $f(s(x))$ has a solution on $X_0$?*

$\sigma_1 = \{x \mapsto X_1\}$ if $X_2 = X_2$

# Solutions of a Pattern in STA

$$\mathcal{A}_{\mathcal{S}} = \{ \quad a \to X_1, \; f(X_2) \to X_0, \; s(X_1) \to X_2, \; s(X_3) \to X_4, \; s(X_5) \to X_6 \quad \}$$

*Does $f(s(x))$ has a solution on $X_0$?*

$\sigma_1 = \{x \mapsto X_1\}$ if $X_2 = X_2$

$\sigma_2 = \{x \mapsto X_3\}$ if $X_2 = X_4$

# Solutions of a Pattern in STA

$$\mathcal{A}_{\mathcal{S}} = \{ \quad a \to X_1, \quad f(X_2) \to X_0, \quad s(X_1) \to X_2, \quad s(X_3) \to X_4, \quad s(X_5) \to X_6 \quad \}$$

*Does $f(s(x))$ has a solution on $X_0$?*

$\sigma_1 = \{x \mapsto X_1\}$ if $X_2 = X_2$

$\sigma_2 = \{x \mapsto X_3\}$ if $X_2 = X_4$

$\sigma_3 = \{x \mapsto X_5\}$ if $X_2 = X_6$

# Solutions of a Pattern in STA

$$\mathcal{A}_{\mathcal{S}} = \{ \quad a \to X_1, \quad f(X_2) \to X_0, \quad s(X_1) \to X_2, \quad s(X_3) \to X_4, \quad s(X_5) \to X_6 \quad \}$$

*Does $f(s(x))$ has a solution on $X_0$?*

$\sigma_1 = \{x \mapsto X_1\}$ if $X_2 = X_2$
$\sigma_2 = \{x \mapsto X_3\}$ if $X_2 = X_4$
$\sigma_3 = \{x \mapsto X_5\}$ if $X_2 = X_6$

$S^t_X$ denotes the set of solutions $(\sigma, \phi)$ of $t$ on $X$

# Formula specifying $\mathcal{R}-$closed Automata

**Definition ($\phi^{RwA}_{\mathcal{R},\mathcal{A}_{\boldsymbol{S}}}$)**

Let $\mathcal{A}_S$ be an STA and $\mathcal{R}$ be a TRS such that $\mathcal{A}_S = \langle \mathcal{X}, \Sigma, \mathcal{X}_F, \Delta \rangle$.

$$\phi^{RwA}_{\mathcal{R},\mathcal{A}_{\boldsymbol{S}}} \stackrel{def}{=} \bigwedge_{l \to r \in \mathcal{R}} \bigwedge_{X \in \mathcal{X}_{\mathcal{Q}}} \bigwedge_{(\sigma,\alpha) \in S^{l}_{\boldsymbol{X}}} (\alpha \Rightarrow reco(r\sigma, X))$$

# Formula specifying $\mathcal{R}-$closed Automata

Definition ($\phi^{RwA}_{\mathcal{R},\mathcal{A}_{\mathbf{S}}}$)

Let $\mathcal{A}_S$ be an STA and $\mathcal{R}$ be a TRS such that $\mathcal{A}_S = \langle \mathcal{X}, \Sigma, \mathcal{X}_F, \Delta \rangle$.

$$\phi^{RwA}_{\mathcal{R},\mathcal{A}_{\mathbf{S}}} \stackrel{def}{=} \bigwedge_{l \to r \in \mathcal{R}} \bigwedge_{X \in \mathcal{X}_{\mathcal{Q}}} \bigwedge_{(\sigma,\alpha) \in S^{l}_{\mathbf{X}}} (\alpha \Rightarrow reco(r\sigma, X))$$

# Formula specifying $\mathcal{R}$−closed Automata

**Definition ($\phi^{RwA}_{\mathcal{R},\mathcal{A}_{\boldsymbol{S}}}$)**

Let $\mathcal{A}_S$ be an STA and $\mathcal{R}$ be a TRS such that $\mathcal{A}_S = \langle \mathcal{X}, \Sigma, \mathcal{X}_F, \Delta \rangle$.

$$\phi^{RwA}_{\mathcal{R},\mathcal{A}_{\boldsymbol{S}}} \stackrel{def}{=} \bigwedge_{l \to r \in \mathcal{R}} \bigwedge_{X \in \mathcal{X}_{\mathcal{Q}}} \bigwedge_{(\sigma,\alpha) \in S^{l}_{\boldsymbol{X}}} (\alpha \Rightarrow reco(r\sigma, X))$$

# Formula specifying $\mathcal{R}$−closed Automata

### Definition ($\phi^{RwA}_{\mathcal{R},\mathcal{A}_{\mathbf{S}}}$)

Let $\mathcal{A}_S$ be an STA and $\mathcal{R}$ be a TRS such that $\mathcal{A}_S = \langle \mathcal{X}, \Sigma, \mathcal{X}_F, \Delta \rangle$.

$$\phi^{RwA}_{\mathcal{R},\mathcal{A}_{\mathbf{S}}} \stackrel{def}{=} \bigwedge_{l \to r \in \mathcal{R}} \bigwedge_{X \in \mathcal{X}_{\mathcal{Q}}} \bigwedge_{(\sigma,\alpha) \in S^l_{\mathbf{X}}} (\alpha \Rightarrow reco(r\sigma, X))$$

### Proposition

*Let $\mathcal{A}_S$ and $\mathcal{R}$ be respectively an STA and a TRS. Let $\mathcal{Q}$ be a set of states and $\iota : \mathcal{X} \to \mathcal{Q}$ be an instantiation of $\mathcal{A}_S$. Thus,*

$$\iota \models \phi^{RwA}_{\mathcal{R},\mathcal{A}_{\mathbf{S}}} \text{ iff } \mathcal{A}_S\iota \text{ is } \mathcal{R}-closed$$

# Formula of Forbidden Terms

**Definition ($\phi^{Bad}_{\mathcal{A}_{\boldsymbol{S}}}$)**

Let $\mathcal{A}_S$ be an STA $\langle \mathcal{X}, \Sigma, \mathcal{X}_F, \Delta \rangle$ and *Bad* be a finite set of terms.

$$\phi^{Bad}_{\mathcal{A}_{\boldsymbol{S}}} \overset{def}{=} \bigwedge_{t \in Bad} (\neg(reco_{\mathcal{A}_{\boldsymbol{S}}}(t)))$$

# Formula of Forbidden Terms

### Definition ($\phi_{\mathcal{A}_S}^{Bad}$)

Let $\mathcal{A}_S$ be an STA $\langle \mathcal{X}, \Sigma, \mathcal{X}_F, \Delta \rangle$ and *Bad* be a finite set of terms.

$$\phi_{\mathcal{A}_S}^{Bad} \stackrel{def}{=} \bigwedge_{t \in Bad} (\neg(reco_{\mathcal{A}_S}(t)))$$

### Proposition

*Let $\mathcal{Q}$ be a set of states and $\iota$ be an instantiation $\mathcal{X} \to \mathcal{Q}$ of $\mathcal{A}_S$.*

$$\iota \models \phi_{\mathcal{A}_S}^{Bad} \text{ iff } \mathcal{L}(\mathcal{A}_S^{\iota}) \cap Bad = \emptyset$$

# Conclusive Analysis from Formula Satisfaction

### Theorem

*Let $\mathcal{A}_S$ be an STA and $\mathcal{A}$ be an automaton such that $\mathcal{A} \subseteq \mathcal{A}_S$. Let $\mathcal{R}$ be a TRS and Bad be a finite set of terms. Let $\iota$ be an instantiation of $\mathcal{A}_S$.*

$$\iota \models \phi_{\mathcal{A}_{\boldsymbol{S}}}^{Bad} \wedge \phi_{\mathcal{R},\mathcal{A}_{\boldsymbol{S}}}^{RwA} \text{ iff } \mathcal{A}_S\iota \text{ is a conclusive } (\mathcal{A},\mathcal{R}) \text{ over} - \text{approximation}$$

# Example of Formula characterizing a Conclusive Approximation

Inputs

- $\mathcal{R} = \{f(x) \to f(s(s(x))), even(f(s(s(x)))) \to even(f(x)), even(f(0)) \to true, even(f(s(0))) \to false\}$
- $Bad = \{false\}$
- $\mathcal{A} = \langle \mathcal{Q}, \Sigma, F, \delta \rangle$ with
  - $\mathcal{Q} = \{q_0, q_1, q_2\}$
  - $\Sigma = \{f : 1, s : 1, 0 : 0, even : 1, true : 0, false : 0\}$
  - $F = \{q_2\}$
  - $\delta = \{even(q_1) \to q_2, f(q_0) \to q_1, 0 \to q_0\}$
- $\mathcal{A}_S = \langle \mathcal{X}, \Sigma, \mathcal{X}_F, \Delta \rangle$ with
  - $\mathcal{X} = \{X_{q_0}, \ldots, X_{q_{11}}\}$
  - $\mathcal{X}_F = \{X_{q_2}\}$
  - $\Delta = \{true \to X_{q_{10}}, false \to X_{q_{11}}, s(X_{q_5}) \to X_{q_6}, s(X_{q_4}) \to X_{q_5}, 0 \to X_{q_0}, even(X_{q_9}) \to X_{q_7}, even(X_{q_1}) \to X_{q_2}, f(X_{q_8}) \to X_{q_9}, f(X_{q_6}) \to X_{q_3}, f(X_{q_0}) \to X_{q_1}\}$

# Example of Formula characterizing a Conclusive Approximation

Inputs

- $\mathcal{R} = \{f(x) \to f(s(s(x))), even(f(s(s(x)))) \to even(f(x)), even(f(0)) \to true,\ even(f(s(0))) \to false\}$
- $Bad = \{false\}$
- $\mathcal{A} = \langle \mathcal{Q}, \Sigma, F, \delta \rangle$ with
  - $\mathcal{Q} = \{q_0, q_1, q_2\}$
  - $\Sigma = \{f : 1, s : 1, 0 : 0, even : 1, true : 0, false : 0\}$
  - $F = \{q_2\}$
  - $\delta = \{even(q_1) \to q_2, f(q_0) \to q_1, 0 \to q_0\}$
- $\mathcal{A}_S = \langle \mathcal{X}, \Sigma, \mathcal{X}_F, \Delta \rangle$ with
  - $\mathcal{X} = \{X_{q_0}, \ldots, X_{q_{11}}\}$
  - $\mathcal{X}_F = \{X_{q_2}\}$
  - $\Delta = \{true \to X_{q_{10}}, false \to X_{q_{11}}, s(X_{q_5}) \to X_{q_6}, s(X_{q_4}) \to X_{q_5}, 0 \to X_{q_0}, even(X_{q_9}) \to X_{q_7}, even(X_{q_1}) \to X_{q_2}, f(X_{q_8}) \to X_{q_9}, f(X_{q_6}) \to X_{q_3}, f(X_{q_0}) \to X_{q_1}\}$

# Contents

# To summarize

+ High level expertise is kicked out
  - Characterization of conclusive approximations by a formula $\phi$
  - If $\exists \iota.\ \iota \models \phi$ then trivial construction of the resulting automaton
  - Automated semi-algorithm for proving unreachability

− Formula are huge. Needs of techniques for solving such formulae
  - Mona : OK until 15 variables and KO $>$15

# Work in progress & Future Works

- Solving formulae using Symbolic techniques (à la Mona) *to be published*
- Solving formulae using Constraint techniques
- Exploring other ways to represent rewriting approximations

# Questions

Thanks for your attention