Muller Message-Passing Automata and Logics

Benedikt Bollig[1] Dietrich Kuske[2]

[1]LSV, ENS Cachan, CNRS, France

[2]Institut für Informatik, Universität Leipzig, Germany

LATA, April 4, 2007

1. Muller Message-Passing Automata and MSCs

2. Monadic Second-Order Logic over MSCs

3. Ehrenfeucht-Fraïssé Game and Hanf's Theorem for FO^{∞} logic

4. Muller MPA vs. MSO Logic

Presentation outline

1. Muller Message-Passing Automata and MSCs
2. Monadic Second-Order Logic over MSCs
3. Ehrenfeucht-Fraïssé Game and Hanf's Theorem for FO^∞ logic
4. Muller MPA vs. MSO Logic

The architecture of a message-passing system

Definition

We fix the following parameters:

- *Proc* a finite set of at least two processes
- *Msg* a finite set of message contents

Definition

- $\Sigma_p := \{p!q(a) \mid q \in Proc \setminus \{p\},\ a \in Msg\} \cup \{p?q(a) \mid q \in Proc \setminus \{p\},\ a \in Msg\}$
- $\Sigma := \bigcup_{p \in Proc} \Sigma_p$

The architecture of a message-passing system

Definition

We fix the following parameters:

- $Proc$ a finite set of at least two processes
- Msg a finite set of message contents

Definition

- $\Sigma_p := \{p!q(a) \mid q \in Proc \setminus \{p\},\ a \in Msg\} \cup \{p?q(a) \mid q \in Proc \setminus \{p\},\ a \in Msg\}$
- $\Sigma := \bigcup_{p \in Proc} \Sigma_p$

Message-passing automata

Definition

A message-passing automaton (MPA) over $Proc$ and Msg is a structure

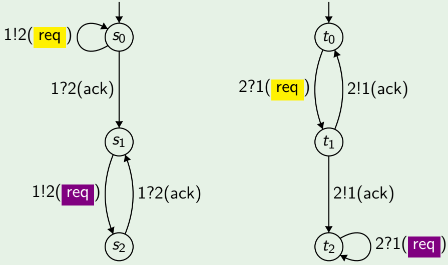
$$\mathcal{A} = ((\mathcal{A}_p)_{p \in Proc}, \mathcal{D}, \imath, Acc)$$

where

- $\mathcal{D}$ is a nonempty finite set of synchronization data
- for each $p \in Proc$, $\mathcal{A}_p = (S_p, \Delta_p)$ with
  - S_p is a nonempty finite set of local states
  - $\Delta_p \subseteq S_p \times \Sigma_p \times \mathcal{D} \times S_p$ is a set of local transitions
- $\imath \in \prod_{p \in Proc} S_p$ is the global initial state
- Acc is an acceptance condition ...

Message-passing automata

Example

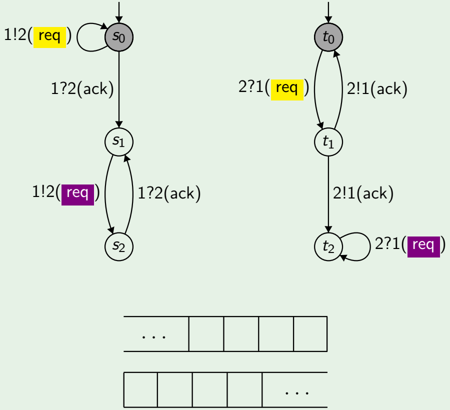


MPA $\mathcal{A}$ over $\{1, 2\}$ and $\{\text{req}, \text{ack}\}$

- $\mathcal{D} = \{\ ,\ ,\ \}$
- $S_1 = \{s_0, s_1, s_2\}$
  $S_2 = \{t_0, t_1, t_2\}$
- Δ_1: $s_0 \xrightarrow{1!2(\text{req})}_1 s_0 \ \ldots$
  Δ_2: $t_0 \xrightarrow{2?1(\text{req})}_2 t_1 \ \ldots$
- $\imath = (s_0, t_0)$
- $Acc = \ldots$

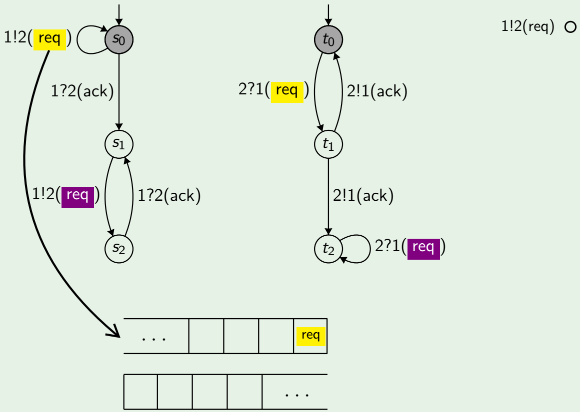
Message-passing automata

Example



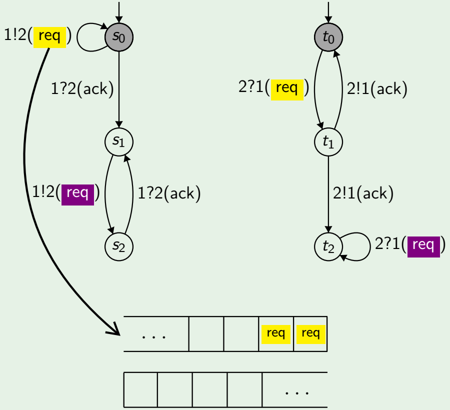
Message-passing automata

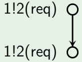
Example



Message-passing automata

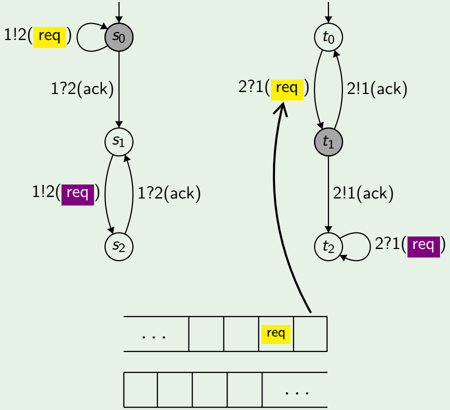
Example





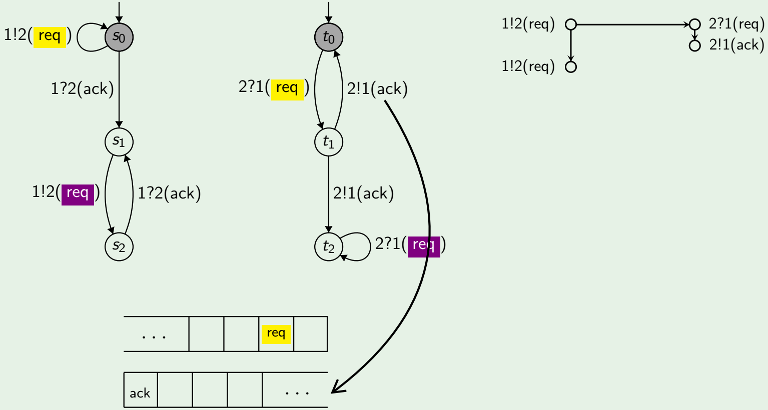
Message-passing automata

Example



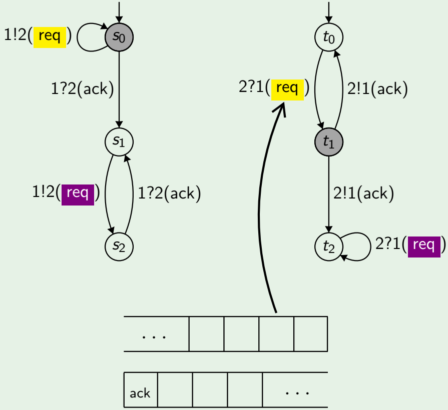
Message-passing automata

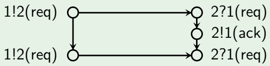
Example



Message-passing automata

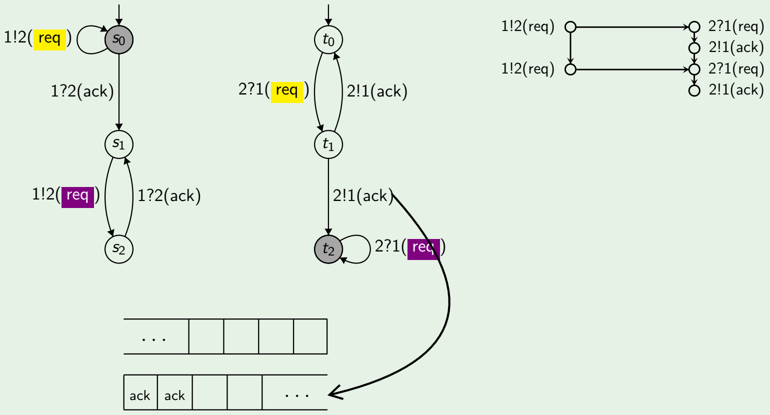
Example





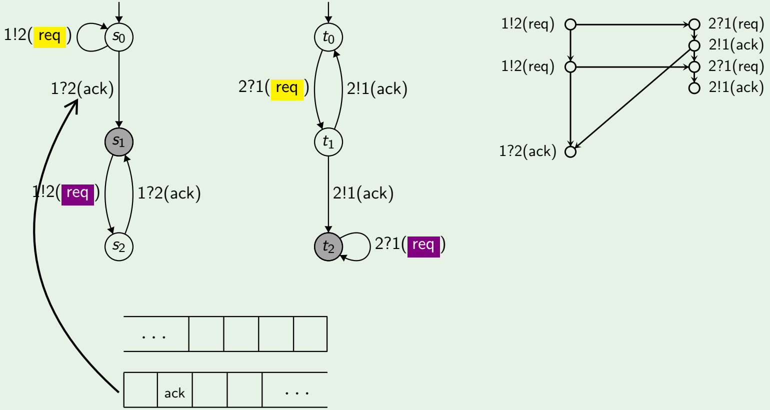
Message-passing automata

Example



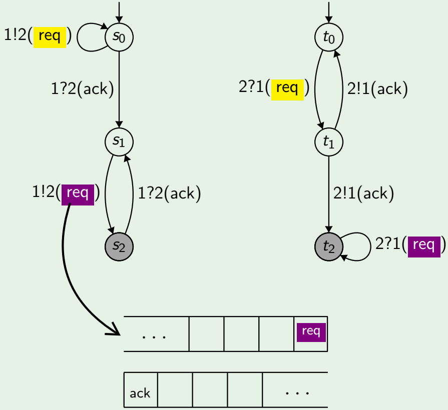
Message-passing automata

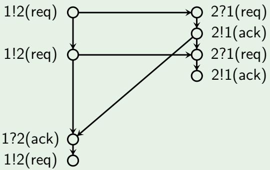
Example



Message-passing automata

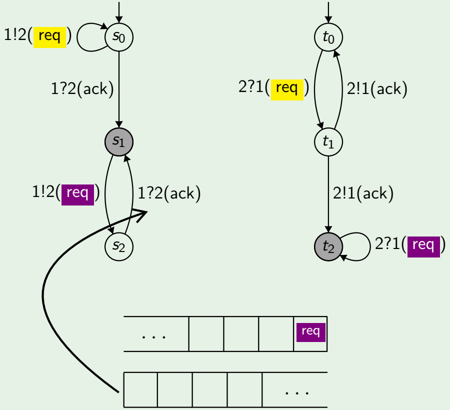
Example

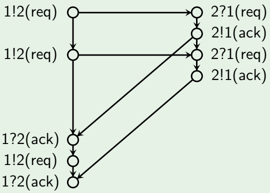




Message-passing automata

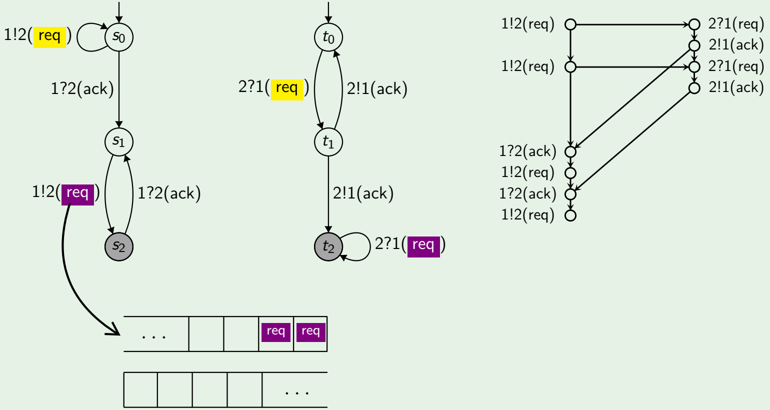
Example





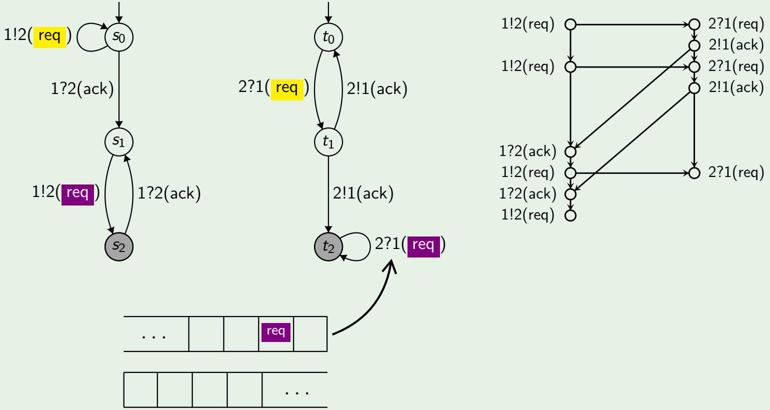
Message-passing automata

Example



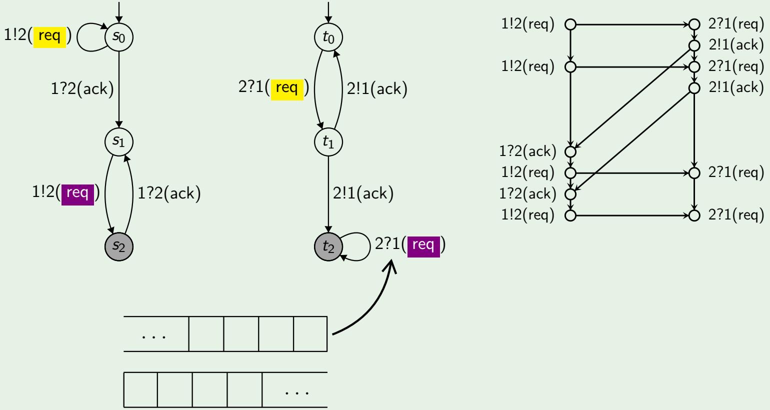
Message-passing automata

Example



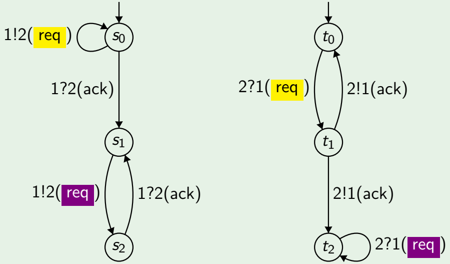
Message-passing automata

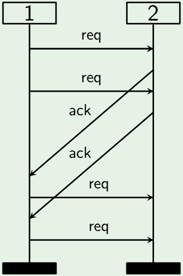
Example

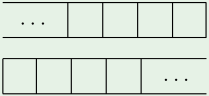


Message-passing automata

Example







Message sequence charts

Definition

A message sequence chart (MSC) is a structure $(E, \lessdot_{proc}, <_{msg}, \lambda)$ such that:

- E is the set of events
- $\lambda : E \to \Sigma$
- $\lessdot_{proc} \subseteq E \times E$ relates successors on a process line
- $<_{msg} \subseteq E \times E$ relates messages (FIFO and complete)
- $\leq = (\lessdot_{proc} \cup <_{msg})^*$ is a partial oder
- $\{e' \in E \mid e' \leq e\}$ is a finite set for any $e \in E$

Semantics of MPA in terms of MSCs

Let $\mathcal{A} = ((\mathcal{A}_p)_{p \in Proc}, \mathcal{D}, \imath, Acc)$ be an MPA and $\mathcal{M} = (E, \lessdot_{\text{proc}}, <_{\text{msg}}, \lambda)$ an MSC.

Definition

A run of $\mathcal{A}$ on $\mathcal{M}$ is a mapping $\rho : E \to \bigcup_{p \in Proc} S_p$ such that, for any $(e, e') \in <_{\text{msg}}$, there is $m \in \mathcal{D}$ such that:

- $(\rho^{-}(e), \lambda(e), m, \rho(e)) \in \Delta_{Proc(e)}$
- $(\rho^{-}(e'), \lambda(e'), m, \rho(e')) \in \Delta_{Proc(e')}$

where:

$$\rho^{-}(e) := \begin{cases} \imath[p] & \text{if } e \text{ is the first } p\text{-event} \\ \rho(p\text{-pred}(e)) & \text{otherwise} \end{cases}$$

Acceptance conditions for MPA

Let $\mathcal{A} = ((\mathcal{A}_p)_{p \in Proc}, \mathcal{D}, \imath, Acc)$ be an MPA and ρ be a run of $\mathcal{A}$ on $\mathcal{M}$.

Definition

$\mathcal{A}$ is called a Büchi / Muller / Staiger-Wagner MPA if $Acc \subseteq \prod_{p \in Proc} 2^{S_p}$

- Büchi: ρ is accepting if there is $\overline{s} \in Acc$ such that $\overline{s}[p] \cap \mathrm{Inf}_p(\rho) \neq \emptyset$
- Muller: ρ is accepting if $(\mathrm{Inf}_p(\rho))_{p \in Proc} \in Acc$
- Staiger-Wagner: ρ is accepting if $(\mathrm{Occ}_p(\rho))_{p \in Proc} \in Acc$

Acceptance conditions for MPA

Let $\mathcal{A} = ((\mathcal{A}_p)_{p \in Proc}, \mathcal{D}, \imath, Acc)$ be an MPA and ρ be a run of $\mathcal{A}$ on $\mathcal{M}$.

Definition

$\mathcal{A}$ is called a Büchi / Muller / Staiger-Wagner MPA if $Acc \subseteq \prod_{p \in Proc} 2^{S_p}$

Büchi: ρ is accepting if there is $\overline{s} \in Acc$ such that $\overline{s}[p] \cap \mathrm{Inf}_p(\rho) \neq \emptyset$

Muller: ρ is accepting if $(\mathrm{Inf}_p(\rho))_{p \in Proc} \in Acc$

Staiger-Wagner: ρ is accepting if $(\mathrm{Occ}_p(\rho))_{p \in Proc} \in Acc$

Acceptance conditions for MPA

Let $\mathcal{A} = ((\mathcal{A}_p)_{p \in Proc}, \mathcal{D}, \imath, Acc)$ be an MPA and ρ be a run of $\mathcal{A}$ on $\mathcal{M}$.

Definition

$\mathcal{A}$ is called a Büchi / Muller / Staiger-Wagner MPA if $Acc \subseteq \prod_{p \in Proc} 2^{S_p}$

Büchi: ρ is accepting if there is $\overline{s} \in Acc$ such that $\overline{s}[p] \cap \mathrm{Inf}_p(\rho) \neq \emptyset$

Muller: ρ is accepting if $(\mathrm{Inf}_p(\rho))_{p \in Proc} \in Acc$

Staiger-Wagner: ρ is accepting if $(\mathrm{Occ}_p(\rho))_{p \in Proc} \in Acc$

Acceptance conditions for MPA

Let $\mathcal{A} = ((\mathcal{A}_p)_{p \in Proc}, \mathcal{D}, \imath, Acc)$ be an MPA and ρ be a run of $\mathcal{A}$ on $\mathcal{M}$.

Definition

$\mathcal{A}$ is called a Büchi / Muller / Staiger-Wagner MPA if $Acc \subseteq \prod_{p \in Proc} 2^{S_p}$

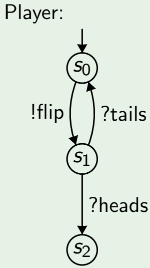
Büchi: ρ is accepting if there is $\overline{s} \in Acc$ such that $\overline{s}[p] \cap \mathrm{Inf}_p(\rho) \neq \emptyset$

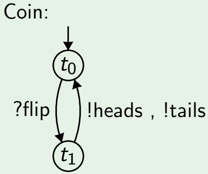
Muller: ρ is accepting if $(\mathrm{Inf}_p(\rho))_{p \in Proc} \in Acc$

Staiger-Wagner: ρ is accepting if $(\mathrm{Occ}_p(\rho))_{p \in Proc} \in Acc$

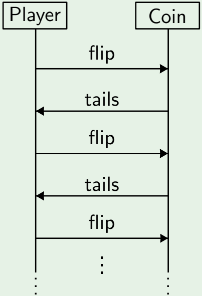
Example MPA

Example





$$Acc = \{ \; (\{s_0, s_1\}, \{t_0, t_1\}) \quad (\{s_2\}, \{t_0\}) \; \}$$



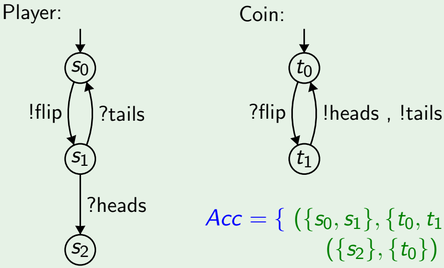
$$L_{\text{Muller}}(\mathcal{A}) = (\text{flip tails})^\omega + (\text{flip tails})^* \text{ flip heads}$$

$$L_{\text{Büchi}}(\mathcal{A}) = (\text{flip tails})^\omega + (\text{flip tails})^* \, (\varepsilon + \text{flip } (\varepsilon + \text{heads}))$$

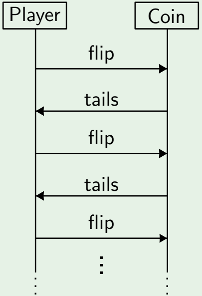
$$L_{\text{SW}}(\mathcal{A}) = (\text{flip tails})^\omega + (\text{flip tails})^* \text{ flip } (\varepsilon + \text{tails})$$

Example MPA

Example



$$Acc = \{\ (\{s_0, s_1\}, \{t_0, t_1\}) \quad (\{s_2\}, \{t_0\})\ \}$$



$$L_{\text{Muller}}(\mathcal{A}) = (\text{flip tails})^{\omega} + (\text{flip tails})^{*}\ \text{flip heads}$$

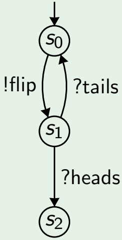
$$L_{\text{Büchi}}(\mathcal{A}) = (\text{flip tails})^{\omega} + (\text{flip tails})^{*}\ (\varepsilon + \text{flip}\ (\varepsilon + \text{heads}))$$

$$L_{\text{SW}}(\mathcal{A}) = (\text{flip tails})^{\omega} + (\text{flip tails})^{*}\ \text{flip}\ (\varepsilon + \text{tails})$$

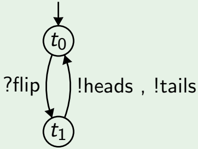
Example MPA

Example

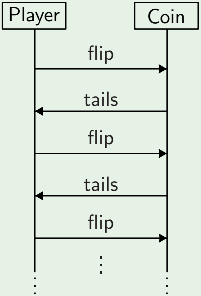
Player:



Coin:



$$Acc = \{\ (\{s_0, s_1\}, \{t_0, t_1\})\ \ (\{s_2\}, \{t_0\})\ \}$$



$$L_{\text{Muller}}(\mathcal{A}) = (\text{flip tails})^\omega + (\text{flip tails})^* \text{ flip heads}$$

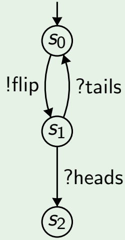
$$L_{\text{Büchi}}(\mathcal{A}) = (\text{flip tails})^\omega + (\text{flip tails})^* (\varepsilon + \text{flip } (\varepsilon + \text{heads}))$$

$$L_{\text{SW}}(\mathcal{A}) = (\text{flip tails})^\omega + (\text{flip tails})^* \text{ flip } (\varepsilon + \text{tails})$$

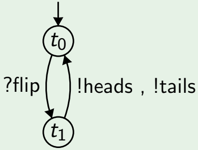
Example MPA

Example

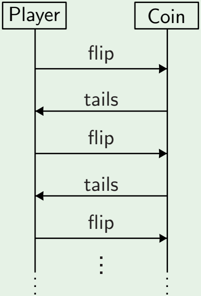
Player:



Coin:



$$Acc = \{\ (\{s_0, s_1\}, \{t_0, t_1\})\ \ (\{s_2\}, \{t_0\})\ \}$$



$$L_{\text{Muller}}(\mathcal{A}) = (\text{flip tails})^\omega + (\text{flip tails})^* \text{ flip heads}$$

$$L_{\text{Büchi}}(\mathcal{A}) = (\text{flip tails})^\omega + (\text{flip tails})^* \ (\varepsilon + \text{flip } (\varepsilon + \text{heads}))$$

$$L_{\text{SW}}(\mathcal{A}) = (\text{flip tails})^\omega + (\text{flip tails})^* \text{ flip } (\varepsilon + \text{tails})$$

Termination-detecting MPA

Definition

Informally, a Büchi / Muller / Staiger-Wagner MPA

$$\mathcal{A} = ((\mathcal{A}_p)_{p \in Proc}, \mathcal{D}, \imath, Acc)$$

is termination-detecting if any $\overline{s} \in Acc$ is equipped with a flag

$$flag : Proc \to \{0, 1\}$$

to detect if a process executes finitely (0) or infinitely (1) many actions.

Büchi MPA vs. Muller MPA

Theorem

Let L be a set of MSCs. Then the following are equivalent:

- *there is a termination-detecting Muller MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$*
- *there is a Muller MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$*
- *there is a Büchi MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$*

Lemma

Termination-detecting Staiger-Wagner MPA are strictly more expressive than Staiger-Wagner MPA.

Büchi MPA vs. Muller MPA

Theorem

Let L be a set of MSCs. Then the following are equivalent:

- *there is a termination-detecting Muller MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$*
- *there is a Muller MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$*
- *there is a Büchi MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$*

Lemma

Termination-detecting Staiger-Wagner MPA are strictly more expressive than Staiger-Wagner MPA.

Presentation outline

1. Muller Message-Passing Automata and MSCs
2. Monadic Second-Order Logic over MSCs
3. Ehrenfeucht-Fraïssé Game and Hanf's Theorem for FO^{∞} logic
4. Muller MPA vs. MSO Logic

Monadic second-order (MSO) logic over MSCs

Let $\mathcal{R}$ a set of binary relation symbols.

Definition

Set $\mathrm{MSO}(\mathcal{R})$ of MSO formulas over $\mathcal{R}$:

$$\varphi ::= \lambda(x) = \sigma \mid x = y \mid R(x, y) \mid x \in X \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x \varphi \mid \exists X \varphi$$

where $\sigma \in \Sigma$, $R \in \mathcal{R}$, x, y are first-order variables, and X is a second-order variable

- $\mathrm{FO}(\mathcal{R})$ is the first-order fragment of $\mathrm{MSO}(\mathcal{R})$
- $\mathrm{EMSO}(\mathcal{R})$ is the existential fragment of $\mathrm{MSO}(\mathcal{R})$ containing formulas $\exists X_1 \ldots \exists X_n\, \varphi \in \mathrm{MSO}(\mathcal{R})$ such that $\varphi \in \mathrm{FO}(\mathcal{R})$.

$\mathrm{MSO}(\{\leq, \lessdot_{\mathrm{proc}}, <_{\mathrm{msg}}\})$-formulas can be interpreted over MSCs.

Monadic second-order (MSO) logic over MSCs

Let $\mathcal{R}$ a set of binary relation symbols.

Definition

Set $\mathrm{MSO}(\mathcal{R})$ of MSO formulas over $\mathcal{R}$:

$$\varphi ::= \lambda(x) = \sigma \mid x = y \mid R(x, y) \mid x \in X \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x \varphi \mid \exists X \varphi$$

where $\sigma \in \Sigma$, $R \in \mathcal{R}$, x, y are first-order variables, and X is a second-order variable

- $\mathrm{FO}(\mathcal{R})$ is the first-order fragment of $\mathrm{MSO}(\mathcal{R})$
- $\mathrm{EMSO}(\mathcal{R})$ is the existential fragment of $\mathrm{MSO}(\mathcal{R})$ containing formulas $\exists X_1 \ldots \exists X_n\, \varphi \in \mathrm{MSO}(\mathcal{R})$ such that $\varphi \in \mathrm{FO}(\mathcal{R})$.

$\mathrm{MSO}(\{\leq, \lessdot_{\mathrm{proc}}, <_{\mathrm{msg}}\})$-formulas can be interpreted over MSCs.

Monadic second-order (MSO) logic over MSCs

Let $\mathcal{R}$ a set of binary relation symbols.

Definition

Set $\mathrm{MSO}(\mathcal{R})$ of MSO formulas over $\mathcal{R}$:

$$\varphi ::= \lambda(x) = \sigma \mid x = y \mid R(x,y) \mid x \in X \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x \varphi \mid \exists X \varphi$$

where $\sigma \in \Sigma$, $R \in \mathcal{R}$, x, y are first-order variables, and X is a second-order variable

- $\mathrm{FO}(\mathcal{R})$ is the first-order fragment of $\mathrm{MSO}(\mathcal{R})$
- $\mathrm{EMSO}(\mathcal{R})$ is the existential fragment of $\mathrm{MSO}(\mathcal{R})$ containing formulas $\exists X_1 \ldots \exists X_n\, \varphi \in \mathrm{MSO}(\mathcal{R})$ such that $\varphi \in \mathrm{FO}(\mathcal{R})$.

$\mathrm{MSO}(\{\leq, \lessdot_{\mathrm{proc}}, <_{\mathrm{msg}}\})$-formulas can be interpreted over MSCs.

Muller MPA vs. MSO logic

Theorem

MSO($\{\leq, \lessdot_{\text{proc}}, <_{\text{msg}}\}$)

$$\varphi ::= \; x \leq y \;|\; x \lessdot_{\text{proc}} y \;|\; x <_{\text{msg}} y \;|$$
$$\lambda(x) = \sigma \;|\; x = y \;|\; x \in X \;|\; \neg\varphi \;|\; \varphi_1 \vee \varphi_2 \;|\; \exists x \varphi \;|\; \exists X \varphi$$

- any implementation has a specification
  for any Muller MPA $\mathcal{A}$, there is an MSO formula φ such that $L(\mathcal{A}) = \{\mathcal{M} \mid \mathcal{M} \models \varphi\}$
- not every specification is implementable [B. & Leucker 2004]
  (even if we restrict to finite MSCs)

Muller MPA vs. MSO logic

Theorem

MSO($\{\leq, \not\lessdot_{\text{proc}}, \not<_{\text{msg}}\}$)

$$\varphi ::= x \leq y \mid \cancel{x \lessdot_{\text{proc}} y} \mid \cancel{x <_{\text{msg}} y} \mid \lambda(x) = \sigma \mid x = y \mid x \in X \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x \varphi \mid \exists X \varphi$$

- not every implementation has a specification
- not every specification is implementable

(even if we restrict to finite MSCs)

Muller MPA vs. MSO logic

Theorem

$\mathsf{MSO}(\{\leq, \lessdot_{\mathrm{proc}}, <_{\mathrm{msg}}\})$

$$\varphi ::= \; x \leq y \mid x \lessdot_{\mathrm{proc}} y \mid x <_{\mathrm{msg}} y \mid \lambda(x) = \sigma \mid x = y \mid x \in X \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x \varphi \mid \exists X \varphi$$

If we restrict to

$\forall$-bounded channels [Kuske 2003]
or $\exists$-bounded channels [Genest & Kuske & Muscholl 2004]:

- every implementation has a specification
- every specification is implementable

Muller MPA vs. MSO logic

Theorem

EMSO($\{\not\leq, \lessdot_{proc}, <_{msg}\}$)

$$\varphi ::= \cancel{x \leq y} \mid x \lessdot_{proc} y \mid x <_{msg} y \mid$$
$$\lambda(x) = \sigma \mid x = y \mid x \in X \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x \varphi \mid \exists X \varphi$$

If we restrict to finite MSCs [B. & Leucker 2004]:

- every implementation has a specification
- every specification is implementable
  (inherently nondeterministic and of elementary size)

Muller MPA vs. MSO logic

Theorem

EMSO($\{\cancel{\leq}, \lessdot_{\text{proc}}, <_{\text{msg}}\}$)

$$\varphi ::= \cancel{x \leq y} \mid x \lessdot_{\text{proc}} y \mid x <_{\text{msg}} y \mid \lambda(x) = \sigma \mid x = y \mid x \in X \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x \varphi \mid \exists X \varphi$$

If we restrict to finite MSCs [B. & Leucker 2004]:

- every implementation has a specification
- every specification is implementable (inherently nondeterministic and of elementary size)

$\rightsquigarrow$ Hanf's Theorem: connection between FO logic and automata

Muller MPA vs. MSO logic

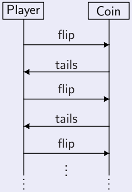
Theorem

EMSO($\{\not\leq, \lessdot_{\text{proc}}, <_{\text{msg}}\}$)

$$\varphi ::= \sout{x \leq y} \mid x \lessdot_{\text{proc}} y \mid x <_{\text{msg}} y \mid \lambda(x) = \sigma \mid x = y \mid x \in X \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x \varphi \mid \exists X \varphi$$

In the setting of Muller MPA and finite or infinite MSCs:

- not every implementation has a specification
  formulas cannot express: "there are infinitely many flips"
- every specification is implementable



Muller MPA vs. MSO logic

Theorem

$\mathrm{EMSO}^{\infty}(\{\not\leq, \lessdot_{\mathrm{proc}}, <_{\mathrm{msg}}\}) = \mathrm{EMSO} +$ "there are infinitely many ..."

$$\varphi ::= \exists^{\infty} x \varphi \quad | \quad \cancel{x \leq y} \ | \ x \lessdot_{\mathrm{proc}} y \ | \ x <_{\mathrm{msg}} y \ | \ \lambda(x) = \sigma \ | \ x = y \ | \ x \in X \ | \ \neg\varphi \ | \ \varphi_1 \vee \varphi_2 \ | \ \exists x \varphi \ | \ \exists X \varphi$$

- every implementation has a specification
- every specification is implementable

Presentation outline

Ehrenfeucht-Fraïssé game

The classical Ehrenfeucht-Fraïssé game characterizes FO.

- Played on structures $\mathfrak{A} = (A, \ldots)$ and $\mathfrak{B} = (B, \ldots)$ over a finite and function-free signature
- Two players: Spoiler and Duplicator
- Game position: $((\mathfrak{A}, \bar{a}), (\mathfrak{B}, \bar{b}), k)$
  winning if $k = 0$ and $(\bar{a}, \bar{b})$ is a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$

If $k > 0$:

(1) Spoiler chooses $a \in A$ or $b \in B$.

(2) Duplicator chooses an element in the other structure (i.e., $b \in B$ or $a \in A$).

(3) The game proceeds with $((\mathfrak{A}, \bar{a}a), (\mathfrak{B}, \bar{b}b), k - 1)$.

Theorem (Ehrenfeucht-Fraïssé)

$\mathfrak{A}$ and $\mathfrak{B}$ agree on FO[k] iff Duplicator wins the game $(\mathfrak{A}, \mathfrak{B}, k)$.

Ehrenfeucht-Fraïssé game

The classical Ehrenfeucht-Fraïssé game characterizes FO.

- Played on structures $\mathfrak{A} = (A, \ldots)$ and $\mathfrak{B} = (B, \ldots)$ over a finite and function-free signature
- Two players: Spoiler and Duplicator
- Game position: $((\mathfrak{A}, \overline{a}), (\mathfrak{B}, \overline{b}), k)$ winning if $k = 0$ and $(\overline{a}, \overline{b})$ is a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$

If $k > 0$:

(1) Spoiler chooses $a \in A$ or $b \in B$.

(2) Duplicator chooses an element in the other structure (i.e., $b \in B$ or $a \in A$).

(3) The game proceeds with $((\mathfrak{A}, \overline{a}a), (\mathfrak{B}, \overline{b}b), k - 1)$.

Theorem (Ehrenfeucht-Fraïssé)

$\mathfrak{A}$ and $\mathfrak{B}$ agree on $\mathrm{FO}[k]$ iff Duplicator wins the game $(\mathfrak{A}, \mathfrak{B}, k)$.

Ehrenfeucht-Fraïssé game

The classical Ehrenfeucht-Fraïssé game characterizes FO.

- Played on structures $\mathfrak{A} = (A, \ldots)$ and $\mathfrak{B} = (B, \ldots)$ over a finite and function-free signature
- Two players: Spoiler and Duplicator
- Game position: $((\mathfrak{A}, \overline{a}), (\mathfrak{B}, \overline{b}), k)$
  winning if $k = 0$ and $(\overline{a}, \overline{b})$ is a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$

If $k > 0$:

(1) Spoiler chooses $a \in A$ or $b \in B$.

(2) Duplicator chooses an element in the other structure (i.e., $b \in B$ or $a \in A$).

(3) The game proceeds with $((\mathfrak{A}, \overline{a}a), (\mathfrak{B}, \overline{b}b), k - 1)$.

Theorem (Ehrenfeucht-Fraïssé)

$\mathfrak{A}$ and $\mathfrak{B}$ agree on FO$[k]$ iff Duplicator wins the game $(\mathfrak{A}, \mathfrak{B}, k)$.

Extended Ehrenfeucht-Fraïssé game

The classical Ehrenfeucht-Fraïssé game characterizes FO.

If $k > 0$:

(1)

(2) Spoiler chooses $a \in A$ or $b \in B$.

(3) Duplicator chooses an element in the other structure (i.e., $b \in B$ or $a \in A$).

(4) The game proceeds with $((\mathfrak{A}, a), (\mathfrak{B}, b), k - 1)$.

(2')

(3')

(4')

(5')

Theorem

$\mathfrak{A}$ and $\mathfrak{B}$ agree on $\mathrm{FO}^{\infty}[k]$ iff Duplicator wins the (new) game $(\mathfrak{A}, \mathfrak{B}, k)$.

Extended Ehrenfeucht-Fraïssé game

The extended Ehrenfeucht-Fraïssé game characterizes FO^∞.

If $k > 0$:

(1) Spoiler chooses to proceed with (2) or (2').

(2) Spoiler chooses $a \in A$ or $b \in B$.

(3) Duplicator chooses an element in the other structure (i.e., $b \in B$ or $a \in A$).

(4) The game proceeds with $((\mathfrak{A}, a), (\mathfrak{B}, b), k - 1)$.

(2') Spoiler chooses an infinite subset Z of A or of B.

(3') Duplicator chooses an infinite subset of the set Z and an infinite subset of the other structure ($\rightsquigarrow A'$ and B').

(4') Spoiler chooses elements $a \in A'$ and $b \in B'$.

(5') The game proceeds with $((\mathfrak{A}, a), (\mathfrak{B}, b), k - 1)$.

Theorem

$\mathfrak{A}$ and $\mathfrak{B}$ agree on $\mathrm{FO}^\infty[k]$ iff Duplicator wins the (new) game $(\mathfrak{A}, \mathfrak{B}, k)$.

Extended Ehrenfeucht-Fraïssé game

The extended Ehrenfeucht-Fraïssé game characterizes FO^∞.

If $k > 0$:

(1) Spoiler chooses to proceed with (2) or (2').

(2) Spoiler chooses $a \in A$ or $b \in B$.

(3) Duplicator chooses an element in the other structure (i.e., $b \in B$ or $a \in A$).

(4) The game proceeds with $((\mathfrak{A}, a), (\mathfrak{B}, b), k-1)$.

(2') Spoiler chooses an infinite subset Z of A or of B.

(3') Duplicator chooses an infinite subset of the set Z and an infinite subset of the other structure ($\rightsquigarrow A'$ and B').

(4') Spoiler chooses elements $a \in A'$ and $b \in B'$.

(5') The game proceeds with $((\mathfrak{A}, a), (\mathfrak{B}, b), k-1)$.

Theorem

$\mathfrak{A}$ and $\mathfrak{B}$ agree on $\mathrm{FO}^\infty[k]$ iff Duplicator wins the (new) game $(\mathfrak{A}, \mathfrak{B}, k)$.

Threshold equivalence

In the context of structures of *bounded degree*, it is sufficient to count spheres of structures $\mathfrak{A}$ and $\mathfrak{B}$ up to some threshold to know if Duplicator wins.

Definition

Let $r \in \mathbb{N}$. The r-sphere of $\mathfrak{A}$ around $a \in A$ is the substructure of $\mathfrak{A}$ induced by $\{a' \in A \mid distance(a, a') \leq r\}$.

Example

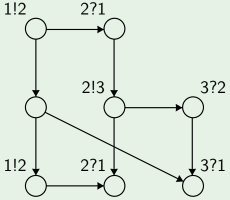
Threshold equivalence

In the context of structures of *bounded degree*, it is sufficient to count spheres of structures $\mathfrak{A}$ and $\mathfrak{B}$ up to some threshold to know if Duplicator wins.

Definition

Let $r \in \mathbb{N}$. The r-sphere of $\mathfrak{A}$ around $a \in A$ is the substructure of $\mathfrak{A}$ induced by $\{a' \in A \mid distance(a, a') \leq r\}$.

Example



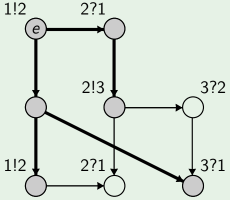
Threshold equivalence

In the context of structures of *bounded degree*, it is sufficient to count spheres of structures $\mathfrak{A}$ and $\mathfrak{B}$ up to some threshold to know if Duplicator wins.

Definition

Let $r \in \mathbb{N}$. The r-sphere of $\mathfrak{A}$ around $a \in A$ is the substructure of $\mathfrak{A}$ induced by $\{a' \in A \mid distance(a, a') \leq r\}$.

Example



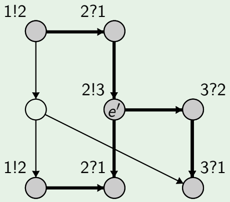
Threshold equivalence

In the context of structures of *bounded degree*, it is sufficient to count spheres of structures $\mathfrak{A}$ and $\mathfrak{B}$ up to some threshold to know if Duplicator wins.

Definition

Let $r \in \mathbb{N}$. The r-sphere of $\mathfrak{A}$ around $a \in A$ is the substructure of $\mathfrak{A}$ induced by $\{a' \in A \mid distance(a, a') \leq r\}$.

Example



Threshold equivalence

Definition

Let $r, t \in \mathbb{N}$. We write $\mathfrak{A} \leftrightarrows_{r,t} \mathfrak{B}$ if, for any r-sphere $\mathcal{S}$,

$$|\mathfrak{A}|_{\mathcal{S}} = |\mathfrak{B}|_{\mathcal{S}} \quad \text{or} \quad \text{both } t < |\mathfrak{A}|_{\mathcal{S}} \text{ and } t < |\mathfrak{B}|_{\mathcal{S}}.$$

Theorem (Hanf 1965)

For any $k, l \in \mathbb{N}$, there are $r, t \in \mathbb{N}$ such that

$$\mathfrak{A} \leftrightarrows_{r,t} \mathfrak{B} \quad \textit{implies} \quad \textit{Duplicator wins } (\mathfrak{A}, \mathfrak{B}, k) \ \textit{(classically)}$$

for any $\mathfrak{A}$ and $\mathfrak{B}$ of degree at most l.

Threshold equivalence

Definition

Let $r, t \in \mathbb{N}$. We write $\mathfrak{A} \leftrightarrows^{\infty}_{r,t} \mathfrak{B}$ if, for any r-sphere $\mathcal{S}$,

$$|\mathfrak{A}|_{\mathcal{S}} = |\mathfrak{B}|_{\mathcal{S}} \quad \text{or} \quad \text{both } t < |\mathfrak{A}|_{\mathcal{S}} < \infty \text{ and } t < |\mathfrak{B}|_{\mathcal{S}} < \infty.$$

Theorem

For any $k, l \in \mathbb{N}$, there are $r, t \in \mathbb{N}$ such that

$$\mathfrak{A} \leftrightarrows^{\infty}_{r,t} \mathfrak{B} \quad \textit{implies} \quad \textit{Duplicator wins } (\mathfrak{A}, \mathfrak{B}, k) \textit{ (extended)}$$

for any $\mathfrak{A}$ and $\mathfrak{B}$ of degree at most l.

Presentation outline

MPA vs. MSO logic

The key connection between MPA and FO logic:

"There is an MPA that computes the spheres around single events."

Proposition (B. & Leucker 2004)

Let $r \in \mathbb{N}$. There are Muller / termination-detecting Staiger-Wagner MPA $\mathcal{A}_r = ((\mathcal{A}_p)_{p \in Proc}, \mathcal{D}, \imath, F)$ and η mapping any local state to an r-sphere such that, for any MSC $\mathcal{M}$:

- *there exists an accepting run of $\mathcal{A}_r$ on $\mathcal{M}$*
- *for any accepting run ρ of $\mathcal{A}_r$ on $\mathcal{M}$ and any event e of $\mathcal{M}$,*

  $\eta(\rho(e))$ *is the r-sphere of $\mathcal{M}$ around e*

Muller MPA vs. MSO logic

Let $r \in \mathbb{N}$, $t \in \mathbb{N} \cup \{\infty\}$, and $\mathcal{S}$ be some r-sphere in some MSC.

Lemma

There exists a termination-detecting Muller MPA $\mathcal{A}$ such that $L(\mathcal{A})$ is the set of MSCs $\mathcal{M}$ satisfying

$$|\mathcal{M}|_{\mathcal{S}} = t \quad / \quad t < |\mathcal{M}|_{\mathcal{S}} < \infty$$

Let $r \in \mathbb{N}$, $t \in \mathbb{N}$, and $\mathcal{S}$ be some r-sphere in some MSC.

Lemma

There exists a termination-detecting Staiger-Wagner MPA $\mathcal{A}$ such that $L(\mathcal{A})$ is the set of MSCs $\mathcal{M}$ satisfying

$$|\mathcal{M}|_{\mathcal{S}} = t \quad / \quad t < |\mathcal{M}|_{\mathcal{S}}$$

Muller MPA vs. MSO logic

Let $r \in \mathbb{N}$, $t \in \mathbb{N} \cup \{\infty\}$, and $\mathcal{S}$ be some r-sphere in some MSC.

Lemma

There exists a termination-detecting Muller MPA $\mathcal{A}$ such that $L(\mathcal{A})$ is the set of MSCs $\mathcal{M}$ satisfying

$$|\mathcal{M}|_{\mathcal{S}} = t \quad / \quad t < |\mathcal{M}|_{\mathcal{S}} < \infty$$

Let $r \in \mathbb{N}$, $t \in \mathbb{N}$, and $\mathcal{S}$ be some r-sphere in some MSC.

Lemma

There exists a termination-detecting Staiger-Wagner MPA $\mathcal{A}$ such that $L(\mathcal{A})$ is the set of MSCs $\mathcal{M}$ satisfying

$$|\mathcal{M}|_{\mathcal{S}} = t \quad / \quad t < |\mathcal{M}|_{\mathcal{S}}$$

Muller MPA vs. MSO logic

Theorem

Let L be a set of MSCs. The following are equivalent:

- *there is a termination-detecting Muller MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$*
- *there is a sentence $\varphi \in \mathrm{EMSO}^{\infty}(\{\lessdot_{\mathrm{proc}}, <_{\mathrm{msg}}\})$ such that $L = L(\varphi)$*

Proof (⇐).

- It is sufficient to consider the case $\varphi \in \mathrm{FO}^{\infty}(\{\lessdot_{\mathrm{proc}}, <_{\mathrm{msg}}\})$.
- L is a finite union of $\leftrightarrows^{\infty}_{r,t}$-equivalence classes for some $r, t \in \mathbb{N}$.
- Any such equivalence class is an intersection of languages as in the lemma above.
- Radius is bounded by $3^{|\varphi|}$.
- t is bounded by $|\varphi| \cdot r$ or infinite.
- Size of the termination-detecting Muller MPA elementary in r.

□

Muller MPA vs. MSO logic

Theorem

Let L be a set of MSCs. The following are equivalent:

- *there is a termination-detecting Muller MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$*
- *there is a sentence $\varphi \in \mathrm{EMSO}^{\infty}(\{\lessdot_{\mathrm{proc}}, <_{\mathrm{msg}}\})$ such that $L = L(\varphi)$*

Proof ($\Leftarrow$).

- It is sufficient to consider the case $\varphi \in \mathrm{FO}^{\infty}(\{\lessdot_{\mathrm{proc}}, <_{\mathrm{msg}}\})$.
- L is a finite union of $\leftrightarrows^{\infty}_{r,t}$-equivalence classes for some $r, t \in \mathbb{N}$.
- Any such equivalence class is an intersection of languages as in the lemma above.
- Radius is bounded by $3^{|\varphi|}$.
- t is bounded by $|\varphi| \cdot r$ or infinite.
- Size of the termination-detecting Muller MPA elementary in r. □

Staiger-Wagner MPA vs. MSO logic

Similarly, one can show:

Theorem

Let L be a set of MSCs. The following are equivalent:

- *there is a termination-detecting Staiger-Wagner MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$*
- *there is a sentence $\varphi \in \mathrm{EMSO}(\{\lessdot_{\mathrm{proc}}, <_{\mathrm{msg}}\})$ such that $L = L(\varphi)$*

Summary

Theorem

Let L be a set of MSCs. The following are equivalent:

- *there is a termination-detecting Muller MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$*
- *there is a Muller MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$*
- *there is a Büchi MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$*
- *there is a sentence $\varphi \in \mathrm{EMSO}^{\infty}(\{\lessdot_{\mathrm{proc}}, <_{\mathrm{msg}}\})$ such that $L = L(\varphi)$*