

Simulation Theorems in μ -calculi

André Arnold

Simulation TheoremS in μ -calculi

Several

Simulation TheoremS in μ -calculi

Several
or ONE ?

First Simulation Theorem

Rabin (1969)

Every formula of S2S is (effectively) equivalent to a nondeterministic Rabin tree automaton.

First Simulation Theorem

Rabin (1969)

Every formula of S2S is (effectively) equivalent to a nondeterministic Rabin tree automaton.

Muller & Schupp (1995)

Every (Streett) alternating tree automaton is (effectively) equivalent to a (Streett) nondeterministic one.

First Simulation Theorem

Rabin (1969)

Every formula of S2S is (effectively) equivalent to a nondeterministic Rabin tree automaton.

Muller & Schupp (1995)

Every (Streett) alternating tree automaton is (effectively) equivalent to a (Streett) nondeterministic one.

Muller & Schupp's Theorem belongs to the μ -calculus

- Rabin $\equiv$ Streett $\equiv$ parity $\equiv$ Muller.
- parity automata $\equiv$ μ -calculus.

Parity automata and μ -calculus

Let F be a set of monotononic functions over a lattice L
(among them, possibly, $\vee$, $\wedge$, $\top$ and $\perp$).

Every μ -term is (effectively) equivalent to a parity automaton (Q, q_*, ρ, Δ) (and vice-versa) where

- Q is a finite set of states and q_* is the initial one,
- $\rho : Q \rightarrow \mathbb{N}$ is the ranking function,
- $\Delta : Q \rightarrow T(F, Q)$ is the transition function.

The notion of a Simulation Theorem

Every automaton is effectively equivalent to an automaton in **normal form**

i.e., $\Delta(q)$ is in “normal” form

- for which satisfiability is decidable,

The notion of a Simulation Theorem

Every automaton is effectively equivalent to an automaton in **normal form**
i.e., $\Delta(q)$ is in “normal” form

- for which satisfiability is decidable,
- or ??????

The μ -calculus for trees

$$F = \{\vee, \wedge, \perp\} \cup \{m(x, y) \mid m \in M\}.$$

Normal form of $\Delta(q)$ for nondeterministic automata:

$$\bigvee_i m_i(q_i, q'_i).$$

Ingredients of the proof of Simulation Theorem

Distributivity rules

- $(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z)$
- $m(x, x') \wedge m(y, y') = m(x \wedge y, x' \wedge y')$

Ingredients of the proof of Simulation Theorem

Distributivity rules

- $(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z)$
- $m(x, x') \wedge m(y, y') = m(x \wedge y, x' \wedge y')$
- $m(x, x') \wedge m'(y, y') = \perp$
- $m(\perp, x') = m(y, \perp) = \perp$
- $x \wedge \perp = \perp, x \vee \perp = x$

Ingredients of the proof of Simulation Theorem

Distributivity rules

- $(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z)$
- $m(x, x') \wedge m(y, y') = m(x \wedge y, x' \wedge y')$
- $m(x, x') \wedge m'(y, y') = \perp$
- $m(\perp, x') = m(y, \perp) = \perp$
- $x \wedge \perp = \perp, x \vee \perp = x$

McNaughton's Theorem

Indeed, its dual.

Dual McNaughton Theorem

Let $\rho : Q \rightarrow \mathbb{N}$ and $\mathcal{R} = \mathcal{P}(Q \times Q)$.

There exists a deterministic parity (word) automaton which accepts all words $u = R_0 R_1 \cdots R_n \cdots \in \mathcal{R}^\omega$ such that every infinite path in the graph u satisfies the parity condition.

Dual McNaughton Theorem

Let $\rho : Q \rightarrow \mathbb{N}$ and $\mathcal{R} = \mathcal{P}(Q \times Q)$.

There exists a deterministic parity (word) automaton which accepts all words $u = R_0 R_1 \cdots R_n \cdots \in \mathcal{R}^\omega$ such that every infinite path in the graph u satisfies the parity condition.

$$q_i = \cdots \wedge m(\bigwedge Y_i^{(1)}, \bigwedge Y_i^{(2)}) \wedge \cdots$$

$$\bigwedge_{i \in I} q_i = \cdots m(\bigwedge (\bigvee_{i \in I} Y_i^{(1)}), \bigwedge (\bigvee_{i \in I} Y_i^{(2)})) \wedge \cdots$$

Dual McNaughton Theorem

Let $\rho : Q \rightarrow \mathbb{N}$ and $\mathcal{R} = \mathcal{P}(Q \times Q)$.

There exists a deterministic parity (word) automaton which accepts all words $u = R_0 R_1 \cdots R_n \cdots \in \mathcal{R}^\omega$ such that every infinite path in the graph u satisfies the parity condition.

$$q_i = \cdots \wedge m(\bigwedge Y_i^{(1)}, \bigwedge Y_i^{(2)}) \wedge \cdots$$

$$\bigwedge_{i \in I} q_i = \cdots m(\bigwedge (\bigvee_{i \in I} Y_i^{(1)}), \bigwedge (\bigvee_{i \in I} Y_i^{(2)})) \wedge \cdots$$

$$\delta(\langle \{q_i \mid i \in I\}, s \rangle) = \cdots m(\langle \bigvee_{i \in I} Y_i^{(1)}, s_1 \rangle, \langle \bigvee_{i \in I} Y_i^{(2)}, s_2 \rangle) \wedge \cdots$$

with $s_j = \delta(s, R_j)$ and $R_j = \bigvee_{i \in I} \{q_i\} \times Y_i^{(j)}$

Simulation Theorem for the modal μ -calculus

Janin & Walukiewicz (1995)

Every modal automaton is (effectively) equivalent to a disjunctive one.

$$\Delta(q) = \bigvee_i m_i \wedge \bigwedge_{a \in A} a \rightarrow X_{i,a}$$

with $X_a \subseteq Q \cup \top$ and $a \rightarrow X = (\bigwedge_{q \in X} \langle a \rangle q) \wedge [a] \bigvee X$.

Simulation Theorem for the modal μ -calculus

Janin & Walukiewicz (1995)

Every modal automaton is (effectively) equivalent to a disjunctive one.

$$\Delta(q) = \bigvee_i m_i \wedge \bigwedge_{a \in A} a \rightarrow X_{i,a}$$

with $X_a \subseteq Q \cup \top$ and $a \rightarrow X = (\bigwedge_{q \in X} \langle a \rangle q) \wedge [a] \bigvee X$.

$$\langle a \rangle q = a \rightarrow \{q, \top\}, \quad [a]q = (a \rightarrow \emptyset) \vee (a \rightarrow \{q\})$$

Simulation Theorem for the modal μ -calculus

Janin & Walukiewicz (1995)

Every modal automaton is (effectively) equivalent to a disjunctive one.

$$\Delta(q) = \bigvee_i m_i \wedge \bigwedge_{a \in A} a \rightarrow X_{i,a}$$

with $X_a \subseteq Q \cup \top$ and $a \rightarrow X = (\bigwedge_{q \in X} \langle a \rangle q) \wedge [a] \bigvee X$.

$$\langle a \rangle q = a \rightarrow \{q, \top\}, \quad [a]q = (a \rightarrow \emptyset) \vee (a \rightarrow \{q\})$$

Distributivity rules

$$(a \rightarrow X) \wedge (a \rightarrow Y) = \bigvee_{R \subseteq X \times Y, R \text{ bicomplete}} a \rightarrow \{x \wedge y \mid x \in X, y \in Y\}$$

Simulation Theorem for the modal μ -calculus

Janin & Walukiewicz (1995)

Every modal automaton is (effectively) equivalent to a disjunctive one.

$$\Delta(q) = \bigvee_i m_i \wedge \bigwedge_{a \in A} a \rightarrow X_{i,a}$$

with $X_a \subseteq Q \cup \top$ and $a \rightarrow X = (\bigwedge_{q \in X} \langle a \rangle q) \wedge [a] \bigvee X$.

$$\langle a \rangle q = a \rightarrow \{q, \top\}, \quad [a]q = (a \rightarrow \emptyset) \vee (a \rightarrow \{q\})$$

Distributivity rules

$$(a \rightarrow X) \wedge (a \rightarrow Y) = \bigvee_{R \subseteq X \times Y, R \text{ bicomplete}} a \rightarrow \{x \wedge y \mid x \in X, y \in Y\}$$

JW $\Rightarrow$ MS

If X and Y are singletons then R is a singleton.

$$m(x, y) = m \wedge l \rightarrow \{x\} \wedge r \rightarrow \{y\}$$

A general Simulation Theorem

Arnold & Niwinski (2001)

Every automaton (over F) is equivalent to an intersection-free one

whenever $\forall f, g \in F - \{\wedge\}, \exists t \in T_+(F - \{\wedge\}, X)$ such that

$$f(x_1, \dots, x_n) \wedge g(y_1, \dots, y_m) = t(c_1, \dots, c_k)$$

where c_i is a conjunction of variables in $\{x_1, \dots, x_n, y_1, \dots, y_m\}$.

A general Simulation Theorem

Arnold & Niwinski (2001)

Every automaton (over F) is equivalent to an intersection-free one

whenever $\forall f, g \in F - \{\wedge\}, \exists t \in T_+(F - \{\wedge\}, X)$ such that

$$f(x_1, \dots, x_n) \wedge g(y_1, \dots, y_m) = t(c_1, \dots, c_k)$$

where c_i is a conjunction of variables in $\{x_1, \dots, x_n, y_1, \dots, y_m\}$.

! Not that general ! It does not encompass JW !

A general Simulation Theorem

Arnold & Niwinski (2001)

Every automaton (over F) is equivalent to an intersection-free one

whenever $\forall f, g \in F - \{\wedge\}, \exists t \in T_+(F - \{\wedge\}, X)$ such that

$$f(x_1, \dots, x_n) \wedge g(y_1, \dots, y_m) = t(c_1, \dots, c_k)$$

where c_i is a conjunction of variables in $\{x_1, \dots, x_n, y_1, \dots, y_m\}$.

! Not that general ! It does not encompass JW !

- $m \rightarrow X$ has not a fixed arity.
- $(x \vee y) \wedge f(\vec{z}) = (x \wedge f(\vec{z})) \vee (y \wedge f(\vec{z}))$ has not the required form.
- $(a \rightarrow X) \wedge (b \rightarrow Y)$ is irreducible.

A general Simulation Theorem

Arnold & Niwinski (2001)

Every automaton (over F) is equivalent to an intersection-free one

whenever $\forall f, g \in F - \{\wedge\}, \exists t \in T_+(F - \{\wedge\}, X)$ such that

$$f(x_1, \dots, x_n) \wedge g(y_1, \dots, y_m) = t(c_1, \dots, c_k)$$

where c_i is a conjunction of variables in $\{x_1, \dots, x_n, y_1, \dots, y_m\}$.

! Not that general ! It does not encompass JW !

- $m \rightarrow X$ has not a fixed arity. Not a real trouble
- $(x \vee y) \wedge f(\vec{z}) = (x \wedge f(\vec{z})) \vee (y \wedge f(\vec{z}))$ has not the required form.
- $(a \rightarrow X) \wedge (b \rightarrow Y)$ is irreducible.

A general Simulation Theorem

Arnold & Niwinski (2001)

Every automaton (over F) is equivalent to an intersection-free one

whenever $\forall f, g \in F - \{\wedge\}, \exists t \in T_+(F - \{\wedge\}, X)$ such that

$$f(x_1, \dots, x_n) \wedge g(y_1, \dots, y_m) = t(c_1, \dots, c_k)$$

where c_i is a conjunction of variables in $\{x_1, \dots, x_n, y_1, \dots, y_m\}$.

! Not that general ! It does not encompass JW !

- $m \rightarrow X$ has not a fixed arity. Not a real trouble
- $(x \vee y) \wedge f(\vec{z}) = (x \wedge f(\vec{z})) \vee (y \wedge f(\vec{z}))$ has not the required form.
- $(a \rightarrow X) \wedge (b \rightarrow Y)$ is irreducible.
Use $C(\dots, X_a, \dots) = \bigwedge_{a \in A} (a \rightarrow X_a)$

A more general Simulation Theorem

Exercise

If $\forall (f, g) \in R \subseteq (F - \{\wedge\})^2$ (R symmetrical) there exists $t \in T_+(F - \{\wedge\}, X)$ such that $f(\vec{x}) \wedge g(\vec{y}) = t(\vec{c})$ then every automaton has an equivalent in the following irreducible form:

$$\Delta(q) = \bigwedge_{i \in I} f_i(\vec{q}_i)$$

with $\forall i \neq j \in I, (f_i, f_j) \notin R$.

A more general Simulation Theorem

Exercise

If $\forall (f, g) \in R \subseteq (F - \{\wedge\})^2$ (R symmetrical) there exists $t \in T_+(F - \{\wedge\}, X)$ such that $f(\vec{x}) \wedge g(\vec{y}) = t(\vec{c})$ then every automaton has an equivalent in the following irreducible form:

$$\Delta(q) = \bigwedge_{i \in I} f_i(\vec{q}_i)$$

with $\forall i \neq j \in I, (f_i, f_j) \notin R$.

$(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z)$ does not satisfy the hypothesis, but $(x \vee y) \wedge (z \vee z) = (x \wedge z) \vee (y \wedge z)$ does.

μ -calculus for processes with loops (1)

The intended models are *deterministic* transition systems over an alphabet A .

Basic operators (for all $a \in A$): $\langle a \rangle x$ and $[a]x$ interpreted as usual and $\circlearrowleft_a$ is defined by $s \models \circlearrowleft_a$ iff $\delta(s, a) = s$.

μ -calculus for processes with loops (1)

The intended models are *deterministic* transition systems over an alphabet A .

Basic operators (for all $a \in A$): $\langle a \rangle x$ and $[a]x$ interpreted as usual and $\circlearrowleft_a$ is defined by $s \models \circlearrowleft_a$ iff $\delta(s, a) = s$.

Arnold, Vincent, Walukiewicz (2003)

Every modal automaton is equivalent to a simple automaton:

$\delta(q) = \bigvee (\bigwedge_{a \in L} \circlearrowleft_a \wedge \bigwedge_{a \in E} \langle a \rangle q_a \wedge \bigwedge_{a \in N} [a] \perp)$ (L, E, N partition of A).

μ -calculus for processes with loops (2)

(1) Get the normal form

$\delta(q) = \bigvee (\bigwedge_{a \in L} \circlearrowleft_a \wedge \bigwedge_{a \in L \cup E} \langle a \rangle q_a \wedge \bigwedge_{a \in N} [a] \perp)$ (L, E, N partition of A) by

- substituting $[a] \perp \vee \langle a \rangle q$ for $[a] q$
- applying the Simulation Theorem with the rules
 - $\langle a \rangle x \wedge \langle a \rangle y = \langle a \rangle (x \wedge y)$
 - $\langle a \rangle x \wedge [a] \perp = \circlearrowleft_a \wedge [a] \perp = \perp$

μ -calculus for processes with loops (2)

(1) Get the normal form

$\delta(q) = \bigvee (\bigwedge_{a \in L} \circlearrowleft_a \wedge \bigwedge_{a \in L \cup E} \langle a \rangle q_a \wedge \bigwedge_{a \in N} [a] \perp)$ (L, E, N partition of A) by

- substituting $[a] \perp \vee \langle a \rangle q$ for $[a] q$
- applying the Simulation Theorem with the rules
 - $\langle a \rangle x \wedge \langle a \rangle y = \langle a \rangle (x \wedge y)$
 - $\langle a \rangle x \wedge [a] \perp = \circlearrowleft_a \wedge [a] \perp = \perp$

(2) Use the rule $\circlearrowleft_a \wedge \langle a \rangle x = \circlearrowleft_a \wedge x$ to get a **non guarded** automaton.
Transform it into a guarded one

(3) Iterate (1) and (2) until termination.

Yet more general (1)

If $\forall (f, g) \in R \subseteq (F - \{\wedge\})^2$ (R symmetrical) there exists $t \in T_+(F - \{\wedge\}, X)$ such that $f(\vec{x}) \wedge g(\vec{y}) = \textcolor{red}{c} \wedge t(\vec{c})$ then every automaton has an equivalent in the following irreducible form:

$$\Delta(q) = \bigwedge_{i \in I} f_i(\vec{q}_i)$$

with $\forall i \neq j \in I, (f_i, f_j) \notin R$.

under some termination conditions *to be defined*.

Yet more general (1)

If $\forall (f, g) \in R \subseteq (F - \{\wedge\})^2$ (R symmetrical) there exists $t \in T_+(F - \{\wedge\}, X)$ such that $f(\vec{x}) \wedge g(\vec{y}) = \textcolor{red}{c} \wedge t(\vec{c})$ then every automaton has an equivalent in the following irreducible form:

$$\Delta(q) = \bigwedge_{i \in I} f_i(\vec{q}_i)$$

with $\forall i \neq j \in I, (f_i, f_j) \notin R$.

under some termination conditions *to be defined*.

The transformation of a non guarded automaton into a guarded one consists in substituting $\Delta(q)$ for q (together with a lot of technicalities, such as duplication of variables, for preserving the parity condition).

Yet more general (2)

The previous theorem does not apply to

$$\begin{aligned}(x \vee y) \wedge f &= (\textcolor{red}{x} \wedge f) \vee (\textcolor{red}{y} \wedge f) \\ (x \mathcal{U} y) \wedge (x' \mathcal{U} y') &= (x \wedge x') \mathcal{U} (\textcolor{red}{y} \wedge (x' \mathcal{U} y')) \\ &\vee (x \wedge x') \mathcal{U} (\textcolor{red}{y}' \wedge (x \mathcal{U} y))\end{aligned}$$

Yet more general (2)

The previous theorem does not apply to

$$\begin{aligned}
 (x \vee y) \wedge f &= (\textcolor{red}{x} \wedge f) \vee (\textcolor{red}{y} \wedge f) \\
 (x \mathcal{U} y) \wedge (x' \mathcal{U} y') &= (x \wedge x') \mathcal{U} (\textcolor{red}{y} \wedge (x' \mathcal{U} y')) \\
 &\vee (x \wedge x') \mathcal{U} (\textcolor{red}{y}' \wedge (x \mathcal{U} y))
 \end{aligned}$$

Conjecture

If $\forall (f, g) \in R \subseteq (F - \{\wedge\})^2$ (R symmetrical) there exists $t \in T(F, X)$ such that $f(\vec{x}) \wedge g(\vec{y}) = t(\vec{x}, \vec{y})$ then every automaton has an equivalent in the following irreducible form:

$$\Delta(q) = \bigwedge_{i \in I} f_i(\vec{q}_i)$$

with $\forall i \neq j \in I, (f_i, f_j) \notin R$.

under some termination conditions *to be defined*.

$\mu(\text{LTL-Next})$

$$\mathcal{L}_1 \quad F \quad :: \quad a|x|\top|\perp|F \vee F|F \wedge F|\mu x.F|\nu x.F|F \mathcal{U} F|\mathcal{A}F$$

$$\mathcal{L}_2 \quad F \quad :: \quad a|x|\top|\perp|F \vee F|F \wedge F|\mu x.F|\nu x.F|a \mathcal{U} F|\mathcal{A}a$$

$$\mathcal{L}_2 \subseteq \mathcal{L}_1$$

$\mu(\text{LTL-Next})$

$$\mathcal{L}_1 \quad F \quad :: \quad a|x|\top|\perp|F \vee F|F \wedge F|\mu x.F|\nu x.F|F \mathcal{U} F|\mathcal{A}F$$

$$\mathcal{L}_2 \quad F \quad :: \quad a|x|\top|\perp|F \vee F|F \wedge F|\mu x.F|\nu x.F|a \mathcal{U} F|\mathcal{A}a$$

$$\mathcal{L}_2 \subseteq \mathcal{L}_1$$

Fifth year examination, Bordeaux, 2005

Every formula of $\mathcal{L}_1$ is equivalent to a formula of $\mathcal{L}_2$.

$\mu(\text{LTL-Next})$

$$\mathcal{L}_1 \quad F \quad :: \quad a|x|\top|\perp|F \vee F|F \wedge F|\mu x.F|\nu x.F|F \mathcal{U} F|\mathcal{A}F$$

$$\mathcal{L}_2 \quad F \quad :: \quad a|x|\top|\perp|F \vee F|F \wedge F|\mu x.F|\nu x.F|a \mathcal{U} F|\mathcal{A}a$$

$$\mathcal{L}_2 \subseteq \mathcal{L}_1$$

Fifth year examination, Bordeaux, 2005

Every formula of $\mathcal{L}_1$ is equivalent to a formula of $\mathcal{L}_2$.

- $\mathcal{L}_1$ is closed under stuttering ($u \models F$ iff $\min_{st}(u) \models F$).
- $\mathcal{L}_1 \subseteq \mu(\text{Next})$.
- $\min_{st}(u) \models \text{Next}F$ iff $\min_{st}(u) \models \bigvee_a ((F \wedge \mathcal{A}a) \vee (a \mathcal{U} (F \wedge \bigvee_{b \neq a} b)))$ (in $\mathcal{L}_2$).

Simplification by rewriting

- $\mathcal{A}(x \wedge y) = \mathcal{A}x \wedge \mathcal{A}y$
- $\mathcal{A}(x \vee y) = ((x \vee y) \mathcal{U}(\mathcal{A}x \vee \mathcal{A}y)) \vee \nu z.((x \mathcal{U}z \wedge y \wedge \neg x) \vee (y \mathcal{U}z \wedge x \wedge \neg y))$
- $\mathcal{A}(\mathcal{A}x) = \mathcal{A}x$
- $\mathcal{A}(x \mathcal{U}y) = ((x \vee y) \mathcal{U}\mathcal{A}y) \vee \nu z.((x \mathcal{U}z \wedge y \wedge \neg x) \vee (y \mathcal{U}z \wedge x \wedge \neg y))$
- $(x \wedge y) \mathcal{U}z = (x \mathcal{U}z) \wedge (y \mathcal{U}z)$
- $(x \vee y) \mathcal{U}z = \mu t.(z \vee (x \mathcal{U}t) \vee (y \mathcal{U}t))$
- $(\mathcal{A}x) \mathcal{U}y = x \mathcal{U}(y \wedge \mathcal{A}x)$
- $(x \mathcal{U}y) \mathcal{U}z = (x \vee y) \mathcal{U}(z \wedge (x \mathcal{U}y)) \quad \vee \quad (x \vee y) \mathcal{U}(y \wedge (y \mathcal{U}z))$

A conjecture

Let $\Sigma = \{(t_i, t'_i) \mid i \in I\}$ be a (noetherian up to AC ?) rewriting system over $T(F, X)$ **such that** $\text{height}(t_i) \leq 2$.

Then every automaton is equivalent to an irreducible one:

$\Delta(q) = f(q_1, \dots, q_n)$ such that $f(\Delta(q_1), \dots, \Delta(q_n))$ contains no redex.

A conjecture

Let $\Sigma = \{(t_i, t'_i) \mid i \in I\}$ be a (noetherian up to AC ?) rewriting system over $T(F, X)$ **such that** $\text{height}(t_i) \leq 2$.

Then every automaton is equivalent to an irreducible one:

$\Delta(q) = f(q_1, \dots, q_n)$ such that $f(\Delta(q_1), \dots, \Delta(q_n))$ contains no redex.

Even if t'_i is a μ -term?

A conjecture

Let $\Sigma = \{(t_i, t'_i) \mid i \in I\}$ be a (noetherian up to AC ?) rewriting system over $T(F, X)$ **such that** $\text{height}(t_i) \leq 2$.

Then every automaton is equivalent to an irreducible one:

$\Delta(q) = f(q_1, \dots, q_n)$ such that $f(\Delta(q_1), \dots, \Delta(q_n))$ contains no redex.

Even if t'_i is a μ -term?

What if $\text{height}(t_i) > 2$?

The Xavier Briand's μ -calculus

The modal μ -calculus for deterministic transition systems over A extended by cycle propositions $\bigcirc_u$ ($u \in A^+$): $s \models \bigcirc_u$ iff $\delta(s, u) = s$.

Arnold, Vincent, Walukiewicz (2003) Satisfiability is decidable for cycles of length 1.

The Xavier Briand's μ -calculus

The modal μ -calculus for deterministic transition systems over A extended by cycle propositions $\odot_u$ ($u \in A^+$): $s \models \odot_u$ iff $\delta(s, u) = s$.

Arnold, Vincent, Walukiewicz (2003) Satisfiability is decidable for cycles of length 1.

Briand, (2005) Satisfiability is undecidable for cycles of length 3.

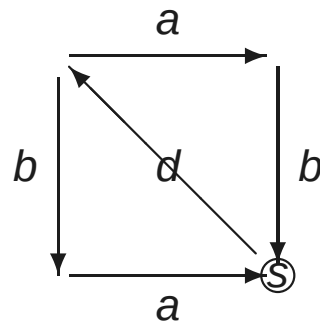
The Xavier Briand's μ -calculus

The modal μ -calculus for deterministic transition systems over A extended by cycle propositions $\bigcirc_u$ ($u \in A^+$): $s \models \bigcirc_u$ iff $\delta(s, u) = s$.

Arnold, Vincent, Walukiewicz (2003) Satisfiability is decidable for cycles of length 1.

Briand, (2005) Satisfiability is undecidable for cycles of length 3.

$$s \models \bigcirc_{dab} \wedge \bigcirc_{dba} \Rightarrow$$



And beyond ?

Reduction: $\circlearrowleft_{abc} \wedge \langle a \rangle (x \wedge \langle b \rangle (y \wedge \langle c \rangle z)) = \circlearrowleft_{abc} \wedge z \wedge \langle a \rangle (x \wedge \langle b \rangle (y \wedge \langle c \rangle \top))$

And beyond ?

Reduction: $\circlearrowleft_{abc} \wedge \langle a \rangle (x \wedge \langle b \rangle (y \wedge \langle c \rangle z)) = \circlearrowleft_{abc} \wedge z \wedge \langle a \rangle (x \wedge \langle b \rangle (y \wedge \langle c \rangle \top))$

Irreducible form: Let $\Delta(q) = \bigvee (\bigwedge \circlearrowleft_{abc} \wedge \bigwedge_{d \in E} \langle d \rangle q_d \wedge \bigwedge_{d \notin E} [d] \perp)$.

Let σ be a positional strategy : $\Delta_\sigma(q) = \bigwedge \circlearrowleft_{abc} \wedge \bigwedge_{d \in E} \langle d \rangle q_d \wedge \bigwedge_{d \notin E} [d] \perp$

Let G_σ be the graph whose set of nodes is $Q \cup \{\top, \perp\}$ and edges are labelled by A defined by Δ_σ .

Δ is irreducible if for any σ and any node q of G_σ which has the property $\circlearrowleft_{abc}$ there is a path abc from q and this path leads to $\top$.

And beyond ?

Reduction: $\odot_{abc} \wedge \langle a \rangle (x \wedge \langle b \rangle (y \wedge \langle c \rangle z)) = \odot_{abc} \wedge z \wedge \langle a \rangle (x \wedge \langle b \rangle (y \wedge \langle c \rangle \top))$

Irreducible form: Let $\Delta(q) = \bigvee (\bigwedge \odot_{abc} \wedge \bigwedge_{d \in E} \langle d \rangle q_d \wedge \bigwedge_{d \notin E} [d] \perp)$.

Let σ be a positional strategy : $\Delta_\sigma(q) = \bigwedge \odot_{abc} \wedge \bigwedge_{d \in E} \langle d \rangle q_d \wedge \bigwedge_{d \notin E} [d] \perp$

Let G_σ be the graph whose set of nodes is $Q \cup \{\top, \perp\}$ and edges are labelled by A defined by Δ_σ .

Δ is irreducible if for any σ and any node q of G_σ which has the property $\odot_{abc}$ there is a path abc from q and this path leads to $\top$.

Guess Satisfiability is decidable for irreducible automata.

And beyond ?

Reduction: $\odot_{abc} \wedge \langle a \rangle (x \wedge \langle b \rangle (y \wedge \langle c \rangle z)) = \odot_{abc} \wedge z \wedge \langle a \rangle (x \wedge \langle b \rangle (y \wedge \langle c \rangle \top))$

Irreducible form: Let $\Delta(q) = \bigvee (\bigwedge \odot_{abc} \wedge \bigwedge_{d \in E} \langle d \rangle q_d \wedge \bigwedge_{d \notin E} [d] \perp)$.

Let σ be a positional strategy : $\Delta_\sigma(q) = \bigwedge \odot_{abc} \wedge \bigwedge_{d \in E} \langle d \rangle q_d \wedge \bigwedge_{d \notin E} [d] \perp$

Let G_σ be the graph whose set of nodes is $Q \cup \{\top, \perp\}$ and edges are labelled by A defined by Δ_σ .

Δ is irreducible if for any σ and any node q of G_σ which has the property $\odot_{abc}$ there is a path abc from q and this path leads to $\top$.

Guess Satisfiability is decidable for irreducible automata.

Consequence There is no effective transformation of an automaton into an irreducible form.