Department of Mathematics, Computer Science and Physics, University of Udine

The Safety Fragment of Temporal Logics on Infinite Sequences

Lesson 2

Luca Geatti
luca.geatti@uniud.it
Angelo Montanari
angelo.montanari@uniud.it

April 8th, 2024

The *monadic second-order theory of one successor* (S1S, for short) is a fragment of second-order logic in which we fix this alphabet:

$$\underbrace{0}_{\text{constant}}, \underbrace{+1}_{\text{function}}, \underbrace{<, =}_{\substack{\text{binary}\\\text{predicates}}}, \underbrace{\{P\}_{P\in\Sigma}}_{\substack{\text{unary}\\\text{predicates}}}$$

The *monadic second-order theory of one successor* (S1S, for short) is a fragment of second-order logic in which we fix this alphabet:

$$\underbrace{0}_{\text{constant}}, \underbrace{+1}_{\text{function}}, \underbrace{<, =}_{\text{binary predicates}}, \underbrace{\{P\}_{P\in\Sigma}}_{\text{unary predicates}}$$

Its syntax is the following. Let $\mathcal{V} = \{x, y, z, \dots\}$ be a set of *first-order variables*. Let $\mathcal{V}' = \{X, Y, Z, \dots\}$ be a set of *second-order variables*.

$$\text{(terms)} \quad t := x \mid 0 \mid t+1$$

$$\text{(formulas)} \quad \phi := \underbrace{P(t)}_{\text{with } P\in\Sigma} \mid \underbrace{X(t)}_{\text{with } X \text{ monadic variable}} \mid t < t' \mid t = t' \mid \neg\phi \mid \phi \vee \phi \mid \underbrace{\exists x \,.\, \phi}_{\text{first-order quantifier}} \mid \underbrace{\exists X \,.\, \phi}_{\text{monadic second-order quantifier}}$$

The *monadic second-order theory of one successor* (S1S, for short) is a fragment of second-order logic in which we fix this alphabet:

$$\underbrace{0}_{\text{constant}}, \underbrace{+1}_{\text{function}}, \underbrace{<, =}_{\substack{\text{binary}\\ \text{predicates}}}, \underbrace{\{P\}_{P\in\Sigma}}_{\substack{\text{unary}\\ \text{predicates}}}$$

Semantics:

Words

$$\langle D, 0, +1, <, =, \{P\}_{P\in\Sigma}\rangle$$

ω-Words

$$\langle \mathbb{N}, 0, +1, <, =, \{P\}_{P\in\Sigma}\rangle$$

The *monadic second-order theory of one successor* (S1S, for short) is a fragment of second-order logic in which we fix this alphabet:

$$\underbrace{0}_{\text{constant}}, \underbrace{+1}_{\text{function}}, \underbrace{<, =}_{\substack{\text{binary}\\\text{predicates}}}, \underbrace{\{P\}_{P \in \Sigma}}_{\substack{\text{unary}\\\text{predicates}}}$$

- Let $\phi(x, y, z, X, Y, Z, \dots)$ be an S1S formula with free variables $x, y, z, X, Y, Z, \dots$ and let ρ be a variable evaluation function.
- We write $\langle D, 0, +1, <, =, \{P\}_{P \in \Sigma} \rangle, \rho \models \phi(x, y, z, X, Y, Z, \dots)$ to denote the fact that the finite word $\langle D, 0, +1, <, =, \{P\}_{P \in \Sigma} \rangle$ *satisfies* $\phi(x, y, z, X, Y, Z, \dots)$ under the evaluation ρ of the free variables.
- The same holds for ω-words $\langle \mathbb{N}, 0, +1, <, =, \{P\}_{P \in \Sigma} \rangle$.

Remark

We can free ourselves from the dependency from a specific alphabet of symbols by replacing Σ by $\{0,1\}^n$, where $n = \lceil log_2(|\Sigma|) \rceil$.

Example. Let $\Sigma = \{a, b, c\}$. We can encode a, b, and c as $(0,0)$, $(0,1)$, and $(1,0)$, respectively, and make use of two second-order variables X_1 and X_2 to represent (ω-)words. As an example, the word $w = abbcb$ is encoded by the word $w = (0,0)(0,1)(0,1)\ (1,0)(0,1)$, which interprets X_1 as the singleton $\{3\}$ and X_2 as the set of natural numbers $\{1,2,4\}$.

S1S

The Monadic Second-order Theory of One Successor is the set of sentences of such a language which are true over $\langle D, 0, +1, <, = \rangle$ (resp., $\langle \mathbb{N}, 0, +1, <, = \rangle$).

Example

There exists a position in which both P_1 and P_2 hold.

$$\exists x \,.\, (P_1(x) \wedge P_2(x+1))$$

Example

Each position where P_1 holds is followed by a position where P_2 holds (by using $+1$ and second-order quantification).

$$\forall x \,.\, \Big(P_1(x) \rightarrow \forall X \,.\, \big(X(x) \wedge \forall y \,.\, (X(y) \rightarrow X(y+1)) \rightarrow \exists z \,.\, (X(z) \wedge P_2(z)) \big) \Big)$$

- We call S1S[FO] (the *first-order* fragment of S1S) the fragment of S1S devoid of second-order quantifiers.
- We denote with $\mathsf{S1S_f}$ the logic S1S interpreted over *finite words*.

- We are interested on S1S[FO] formula $\phi(x)$ with *exactly one free variable* x.
  - x is meant to represent the initial time point.
- The *language over finite words* of $\phi(x)$, denoted with $\mathcal{L}^{<\omega}(\phi(x))$ is defined as:

$$\mathcal{L}^{<\omega}(\phi(x)) \coloneqq \left\{ \langle D, 0, +1, <, =, \{P\}_{P \in \Sigma} \rangle, x \mapsto 0 \models \phi(x) \right\}$$

- The *language over ω-words* of $\phi(x)$, denoted with $\mathcal{L}(\phi(x))$ is defined as:

$$\mathcal{L}(\phi(x)) \coloneqq \left\{ \langle \mathbb{N}, 0, +1, <, =, \{P\}_{P \in \Sigma} \rangle, x \mapsto 0 \models \phi(x) \right\}$$

Theorem (Büchi's Theorem over ω-words)

- *For each* S1S *formula* ϕ, *the language* $\mathcal{L}(\phi)$ *is an* ω*-regular language.*
- *For each* ω*-regular language* $\mathcal{L}$, *there exists an* S1S *formula* ϕ *such that* $\mathcal{L} = \mathcal{L}(\phi)$.

Theorem (Büchi's Theorem over finite words)

- *For each* $\mathsf{S1S_f}$ *formula* ϕ, *the language* $\mathcal{L}^{<\omega}(\phi)$ *is a regular language.*
- *For each regular language* $\mathcal{L}$, *there exists an* $\mathsf{S1S_f}$ *formula* ϕ *such that* $\mathcal{L} = \mathcal{L}^{<\omega}(\phi)$.

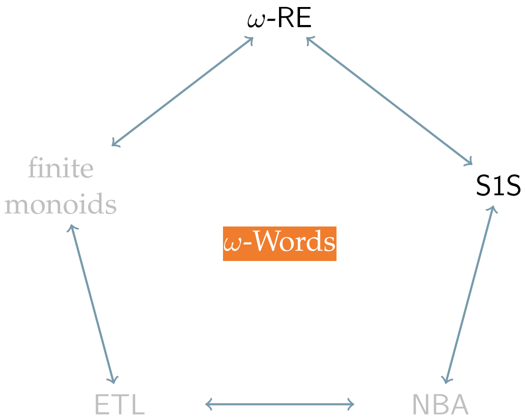
Reference:

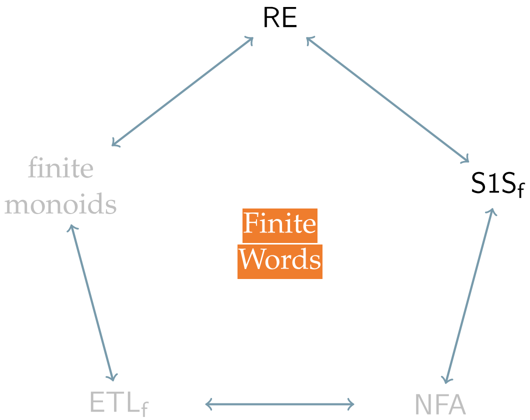
J. R. Buechi (1960). "On a decision method in restricted second-order arithmetics". In: *Proc. Internat. Congr. on Logic, Methodology and Philosophy of Science, 1960*

Reference:

Calvin C Elgot (1961). "Decision problems of finite automata design and related arithmetics". In: *Transactions of the American Mathematical Society* 98.1, pp. 21–51. DOI: 10.1090/S0002-9947-1961-0139530-9

Characterizations of ω-Regular Languages




RE
finite monoids
$S1S_f$
Finite Words
ETL_f
NFA

Theorem (Expressive Equivalence over ω-words)

- *For each* S1S[FO] *formula* ϕ*, the language* $\mathcal{L}(\phi)$ *is a star-free* ω*-language.*
- *For each star-free* ω*-language* $\mathcal{L}(\phi)$*, there exists an* S1S[FO] *formula* ϕ *such that* $\mathcal{L} = \mathcal{L}(\phi)$.

Theorem (Expressive Equivalence over finite words)

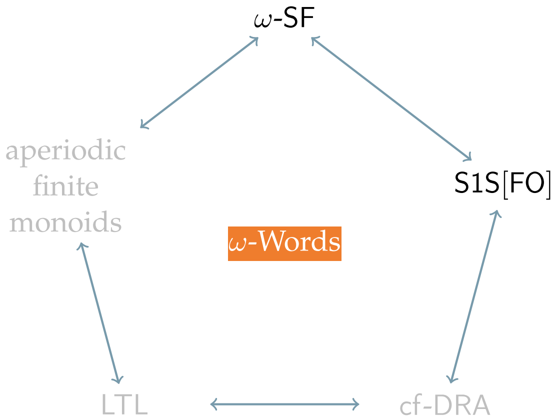
- *For each* $\mathsf{S1S[FO]_f}$ *formula* ϕ*, the language* $\mathcal{L}^{<\omega}(\phi)$ *is a star-free language.*
- *For each star-free language* $\mathcal{L}$*, there exists an* $\mathsf{S1S[FO]_f}$ *formula* ϕ *such that* $\mathcal{L} = \mathcal{L}^{<\omega}(\phi)$.

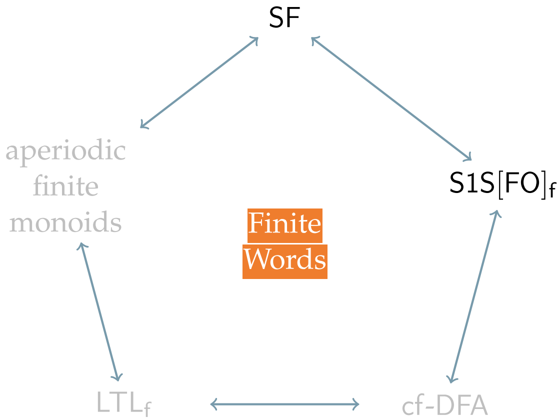
Reference:

Richard E Ladner (1977). "Application of model theoretic games to discrete linear orders and finite automata". In: *Information and Control* 33.4, pp. 281–303. DOI: 10.1016/S0019-9958(77)90443-0

Reference:

Wolfgang Thomas (1981). "A combinatorial approach to the theory of ω-automata". In: *Information and Control* 48.3, pp. 261–283. DOI: 10.1016/S0019-9958(81)90663-X

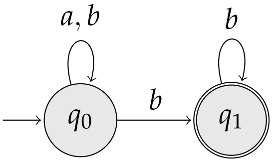

ω-SF
aperiodic
finite
monoids
S1S[FO]
ω-Words
LTL
cf-DRA


SF
aperiodic
finite
monoids
S1S[FO]f
Finite
Words
LTLf
cf-DFA

Definition (Nondeterministic Automaton)

A *nondeterministic automaton* $\mathcal{A}$ is a tuple $\langle Q, \Sigma, I, \Delta, F \rangle$ where:

- Q is the *set of states*;
- Σ is the *alphabet*;
- $I \subseteq Q$ is the set of *initial states*;
- $\Delta \subseteq Q \times \Sigma \times Q$ is the *transition relation*;
- $F \subseteq Q$ is the set of *final states*;



- $Q = \{q_0, q_1\}$;
- $\Sigma = \{a, b\}$;
- $I = \{q_0\}$;
- $\Delta = \{(q_0, a, q_0), (q_0, b, q_0), (q_0, b, q_1), (q_1, b, q_1)\}$;
- $F = \{q_1\}$;

Definition (Nondeterministic Automaton)

A *nondeterministic automaton* $\mathcal{A}$ is a tuple $\langle Q, \Sigma, I, \Delta, F\rangle$ where:

- Q is the *set of states*;
- Σ is the *alphabet*;
- $I \subseteq Q$ is the set of *initial states*;
- $\Delta \subseteq Q \times \Sigma \times Q$ is the *transition relation*;
- $F \subseteq Q$ is the set of *final states*;

A (complete) nondeterministic automaton is *deterministic* iff Δ is a *function*, that is:

$$|\Delta(q, a)| = 1 \qquad \text{for each } q \in Q, a \in \Sigma$$

Let $\mathcal{A}$ be a nondeterministic automaton with alphabet Σ.

Words

- Given a word $\sigma \in \Sigma^*$ with $\sigma = \langle \sigma_0, \sigma_1, \ldots, \sigma_n \rangle$, a *run* π *of* $\mathcal{A}$ *over* σ is a finite sequence of states $\langle q_0, q_1, \ldots, q_{n+1} \rangle \in Q^*$ such that:
  - $q_0 \in I$;
  - $(q_i, \sigma_i, q_{i+1}) \in \Delta$, for each $0 \leq i \leq n$

ω-Words

- Given an ω-word $\sigma \in \Sigma^\omega$ with $\sigma = \langle \sigma_0, \sigma_1, \ldots \rangle$, a *run* π *of* $\mathcal{A}$ *over* σ is an infinite sequence of states $\langle q_0, q_1, \ldots \rangle \in Q^\omega$ such that:
  - $q_0 \in I$;
  - $(q_i, \sigma_i, q_{i+1}) \in \Delta$, for each $i \geq 0$

Let $\mathcal{A}$ be a nondeterministic automaton with alphabet Σ.

Words

Definition (NFA)

A *Nondeterministic Finite Automaton* (NFA, for short) $\langle Q, \Sigma, I, \Delta, F\rangle$ is a nondeterministic automaton in which a run $\pi := \langle q_0, q_1, \ldots, q_{n+1}\rangle \in Q^*$ is said to be *accepting* iff $q_{n+1} \in F$.

ω-Words

Definition (NBA)

A *Nondeterministic Büchi Automaton* (NBA, for short) $\langle Q, \Sigma, I, \Delta, F\rangle$ is a nondeterministic automaton in which a run $\pi := \langle q_0, q_1, \ldots\rangle \in Q^\omega$ is said to be *accepting* iff $\mathsf{Inf}(\pi) \cap F \neq \varnothing$.

$\mathsf{Inf}(\pi)$ is the set of states that occur infinitely often in the infinite run π.

Let $\mathcal{A}$ be a nondeterministic automaton with alphabet Σ.

Words

Definition (NFA)

A *Nondeterministic Finite Automaton* (NFA, for short) $\langle Q, \Sigma, I, \Delta, F \rangle$ is a nondeterministic automaton in which a run $\pi := \langle q_0, q_1, \ldots, q_{n+1} \rangle \in Q^*$ is said to be *accepting* iff $q_{n+1} \in F$.

ω-Words

Definition (NBA)

A *Nondeterministic Büchi Automaton* (NBA, for short) $\langle Q, \Sigma, I, \Delta, F \rangle$ is a nondeterministic automaton in which a run $\pi := \langle q_0, q_1, \ldots \rangle \in Q^\omega$ is said to be *accepting* iff $\mathsf{Inf}(\pi) \cap F \neq \varnothing$.

A run is accepting for a Büchi automaton iff it reaches a final state infinitely often.

Let $\mathcal{A}$ be a nondeterministic automaton with alphabet Σ.

Words

Let $\mathcal{A} = \langle Q, \Sigma, I, \Delta, F \rangle$ be an NFA.

- A word $\sigma \in \Sigma^*$ is *accepted* by $\mathcal{A}$ iff there exists at least one accepting run of $\mathcal{A}$ over σ.
- The *language of* $\mathcal{A}$, denoted by $\mathcal{L}^{<\omega}(\mathcal{A})$, is the set of words in Σ^* accepted by $\mathcal{A}$.

ω-Words

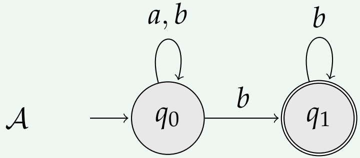
Let $\mathcal{A} = \langle Q, \Sigma, I, \Delta, F \rangle$ be an NBA.

- An ω-word $\sigma \in \Sigma^\omega$ is *accepted* by $\mathcal{A}$ iff there exists at least one accepting run of $\mathcal{A}$ over σ.
- The *language of* $\mathcal{A}$, denoted by $\mathcal{L}(\mathcal{A})$, is the set of ω-words in Σ^ω accepted by $\mathcal{A}$.

Let $\mathcal{A}$ be a nondeterministic automaton with alphabet Σ.

Words

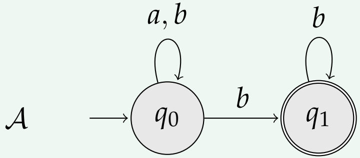
Example



$\mathcal{L}^{<\omega}(\mathcal{A}) = \{\sigma \in \Sigma^* \mid \sigma \neq \varepsilon \wedge$ the last letter of σ is b$\}$

ω-Words

Example



$\mathcal{L}(\mathcal{A}) = \{\sigma \in \Sigma^\omega \mid$ in σ there is a finite number of a$\}$

An important difference

NFA

- A DFA is a deterministic NFA
- NFA are closed under *determinization*: for each NFA $\mathcal{A}$ there exists a DFA $\mathcal{A}'$ such that $\mathcal{L}^{<\omega}(\mathcal{A}) = \mathcal{L}^{<\omega}(\mathcal{A}')$.
- Subset construction.

NBA

- A DBA is a deterministic NBA
- NBA are not closed under *determinization*: there exists a NBA $\mathcal{A}$ for which all DBA $\mathcal{A}'$ are such that $\mathcal{L}(\mathcal{A}) \neq \mathcal{L}(\mathcal{A}')$.

An important difference

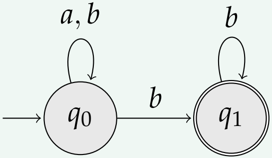
NFA

- A DFA is a deterministic NFA
- NFA are closed under *determinization*: for each NFA $\mathcal{A}$ there exists a DFA $\mathcal{A}'$ such that $\mathcal{L}^{<\omega}(\mathcal{A}) = \mathcal{L}^{<\omega}(\mathcal{A}')$.
- Subset construction.

NBA

Example

Let $\Sigma := \{a, b\}$. The language $\mathcal{L} = \{\sigma \in \Sigma^\omega \mid \exists^{<\omega} i \,.\, \sigma_i = a\}$ is not accepted by any DBA. However, it is accepted by the following NBA.



Theorem (Expressive Equivalence for NBA)

For each ω-language $\mathcal{L} \subseteq \Sigma^\omega$, it holds that:

$$\mathcal{L} \text{ is } \omega\text{-regular}$$
$$\text{iff}$$
$$\mathcal{L} = \mathcal{L}(\mathcal{A}) \text{ for some NBA } \mathcal{A}$$

Theorem (Expressive Equivalence for NFA/DFA)

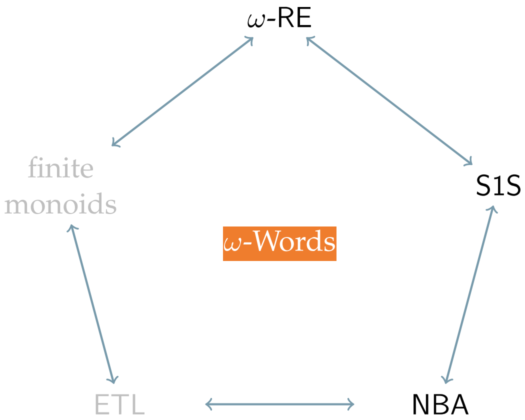
For each language $\mathcal{L} \subseteq \Sigma^*$, it holds that:

$$\mathcal{L} \text{ is regular}$$
$$\text{iff}$$
$$\mathcal{L} = \mathcal{L}^{<\omega}(\mathcal{A}) \text{ for some NFA/DFA } \mathcal{A}$$

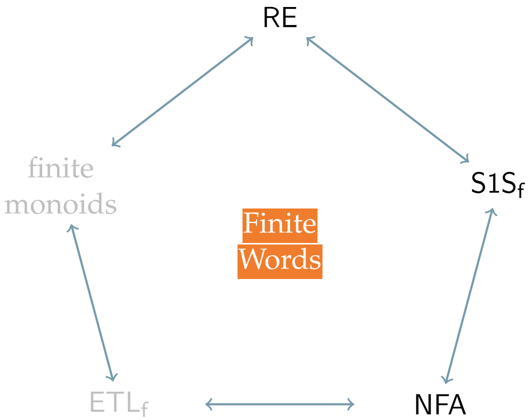
Reference:

Robert McNaughton (1966). "Testing and generating infinite sequences by a finite automaton". In: *Information and control* 9.5, pp. 521–530. DOI: 10.1016/S0019-9958(66)80013-X

Characterizations of ω-Regular Languages



Characterizations of Regular Languages



REFERENCES

Bibliography I

J. R. Buechi (1960). "On a decision method in restricted second-order arithmetics". In: *Proc. Internat. Congr. on Logic, Methodology and Philosophy of Science, 1960*.

Calvin C Elgot (1961). "Decision problems of finite automata design and related arithmetics". In: *Transactions of the American Mathematical Society* 98.1, pp. 21–51. DOI: 10.1090/S0002-9947-1961-0139530-9.

Richard E Ladner (1977). "Application of model theoretic games to discrete linear orders and finite automata". In: *Information and Control* 33.4, pp. 281–303. DOI: 10.1016/S0019-9958(77)90443-0.

Robert McNaughton (1966). "Testing and generating infinite sequences by a finite automaton". In: *Information and control* 9.5, pp. 521–530. DOI: 10.1016/S0019-9958(66)80013-X.

Bibliography II

Wolfgang Thomas (1981). "A combinatorial approach to the theory of ω-automata". In: *Information and Control* 48.3, pp. 261–283. DOI: 10.1016/S0019-9958(81)90663-X.