Omega Regularity with Bounds

Thomas Colcombet (Cnrs, Irisa)
Joint work with Mikołaj Bojańczyk (Warsaw)

Séminaire automate au LIAFA le 2 mars 2007
An \( \omega \)-\textit{word} is an infinite word indexed by \( \omega \): \( a_1 a_2 a_3 \ldots \)
An \( \omega \)-word is an infinite word indexed by \( \omega: a_1a_2a_3 \ldots \)

**Monadic (second-order) logic:**

\[
\phi ::= \phi \lor \phi \mid \neg \phi \mid \exists X.\phi \mid \exists x.\phi \mid x \in X \mid x = S(y) \mid a(x)
\]
Omega Regular Languages

An $\omega$-word is an infinite word indexed by $\omega$: $a_1a_2a_3\ldots$

**Monadic (second-order) logic:**

$$\phi ::= \phi \lor \phi \mid \neg \phi \mid \exists X. \phi \mid \exists x. \phi \mid x \in X \mid x = S(y) \mid a(x)$$

E.g:

$$x \leq y \ ::= \ \forall X. (x \in X \land (\forall z. z \in X \rightarrow S(z) \in X)) \rightarrow y \in X$$
An $\omega$-word is an infinite word indexed by $\omega$: $a_1a_2a_3 \ldots$

**Monadic (second-order) logic:**

$$\phi ::= \phi \lor \phi \mid \neg \phi \mid \exists X. \phi \mid \exists x. \phi \mid x \in X \mid x = S(y) \mid a(x)$$
An \( \omega \)-word is an infinite word indexed by \( \omega: a_1 a_2 a_3 \ldots \)

**Monadic (second-order) logic:**

\[
\phi ::= \phi \lor \phi \mid \neg \phi \mid \exists X. \phi \mid \exists x. \phi \mid x \in X \mid x = S(y) \mid a(x)
\]

**THM (Buchi)**: Satisfaction of monadic logic is decidable over \( \omega \)-words (originally, monadic logic is decidable over \((\mathbb{N}, +1)\))
An \( \omega \)-word is an infinite word indexed by \( \omega: a_1 a_2 a_3 \ldots \)

**Monadic (second-order) logic:**

\[
\phi ::= \phi \lor \phi \mid \neg \phi \mid \exists X. \phi \mid \exists x. \phi \mid x \in X \mid x = S(y) \mid a(x)
\]

**THM (Buchi)** Satisfaction of monadic logic is decidable over \( \omega \)-words (originally, monadic logic is decidable over \((\mathbb{N}, +1)\))

**Idea:** Transform each monadic formula into an ‘equivalent’ automaton.
Step 1: Introduce a suitable family of automata.
**FROM MONADIC LOGIC TO AUTOMATA**

**Step 1:** Introduce a suitable family of automata.

**Step 2:** Prove closure under union, projection, complementation.

**Step 3 (generic):** Prove: for all $u; u; \neg X = \neg A \cup = \iff h u; \neg A i \in L(A)$

**Proof idea:** By induction on the structure of the formula.

In general, to each connective/quantifier and atomic predicate corresponds a language theoretic operation.

**Decidability:**

- **Buchi’s case**
  - Use the class of Buchi automata.
  - Only complementation is non-trivial (Based on Ramsey theorem).
FROM MONADIC LOGIC TO AUTOMATA

Step 1: Introduce a suitable family of automata.

Step 2: Prove closure under union, projection, complementation.

Step 3(generic): Prove: for all $\phi$, there exists an automaton $A_\phi$ such that:

$$\text{for all } u, \quad u, \vec{X} = \vec{A} \models \phi \quad \text{iff} \quad \langle u, \vec{A} \rangle \in L(A_\phi)$$
Step 1: Introduce a suitable family of automata.

Step 2: Prove closure under union, projection, complementation.

Step 3(generic): Prove: for all $\phi$, there exists an automaton $A_\phi$ such that:

$$\text{for all } u, \quad u, X = \vec{A} \models \phi \quad \iff \quad \langle u, \vec{A} \rangle \in L(A_\phi)$$
Step 1: Introduce a suitable family of automata.

Step 2: Prove closure under union, projection, complementation.

Step 3(generic): Prove: for all $\phi$, there exists an automaton $A_\phi$ such that:

\[
\text{for all } u, \quad u, \vec{X} = \vec{A} \models \phi \quad \text{iff} \quad \langle u, \vec{A} \rangle \in L(A_\phi)
\]

Proof idea: By induction on the structure of the formula.

\[
u, \vec{X} = \vec{A} \models \phi \land \psi \quad \text{iff} \quad \langle u, \vec{A} \rangle \in L(A_\phi) \cap L(A_\psi)
\]
Step 1: Introduce a suitable family of automata.

Step 2: Prove closure under union, projection, complementation.

Step 3(generic): Prove: for all $\phi$, there exists an automaton $A_\phi$ such that:

$$\text{for all } u, \; u, \bar{X} = \bar{A} \models \phi \iff \langle u, \bar{A} \rangle \in L(A_\phi)$$

Proof idea: By induction on the structure of the formula.

$$u, \bar{X} = \bar{A} \models \phi \land \psi \iff \langle u, \bar{A} \rangle \in L(A_\phi) \cap L(A_\psi)$$

$$u, \bar{X} = \bar{A} \models \neg \phi \iff \langle u, \bar{A} \rangle \in \overline{L(A_\phi)}$$
Step 1: Introduce a suitable family of automata.

Step 2: Prove closure under union, projection, complementation.

Step 3(generic): Prove: for all $\phi$, there exists an automaton $A_\phi$ such that:

$$\text{for all } u, \langle u, A \rangle \models \phi \iff \langle u, A \rangle \in L(A_\phi)$$

Proof idea: By induction on the structure of the formula.

$$\langle u, A \rangle \models \phi \land \psi \iff \langle u, A \rangle \in L(A_\phi) \cap L(A_\psi)$$

$$\langle u, A \rangle \models \neg \phi \iff \langle u, A \rangle \in \overline{C}(L(A_\phi))$$

$$\langle u, A \rangle \models \exists Y. \phi \iff \langle u, A \rangle \in \pi(L(A_\phi)) \quad (\pi \text{ projection cancelling } Y)$$
FROM MONADIC LOGIC TO AUTOMATA

Step 1: Introduce a suitable family of automata.

Step 2: Prove closure under union, projection, complementation.

Step 3(generic): Prove: for all \( \phi \), there exists an automaton \( A_\phi \) such that:

\[
\text{for all } u, \quad u, \vec{X} = \vec{A} \models \phi \quad \text{iff} \quad \langle u, \vec{A} \rangle \in L(A_\phi)
\]

Proof idea: By induction on the structure of the formula.

\[
\begin{align*}
  u, \vec{X} = \vec{A} \models \phi \land \psi & \quad \text{iff} \quad \langle u, \vec{A} \rangle \in L(A_\phi) \cap L(A_\psi) \\
  u, \vec{X} = \vec{A} \models \neg \phi & \quad \text{iff} \quad \langle u, \vec{A} \rangle \in C(L(A_\phi)) \\
  u, \vec{X} = \vec{A} \models \exists Y.\phi & \quad \text{iff} \quad \langle u, \vec{A} \rangle \in \pi(L(A_\phi)) \quad (\pi \text{ projection cancelling } Y)
\end{align*}
\]

In general, to each connective/quantifier and atomic predicate corresponds a language theoretic operation.
From monadic logic to automata

**Step 1:** Introduce a suitable family of automata.

**Step 2:** Prove closure under union, projection, complementation.

**Step 3(generic):** Prove: for all \( \phi \), there exists an automaton \( A_\phi \) such that:

\[
\text{for all } u, \quad u, \vec{X} = \vec{A} \models \phi \quad \text{iff} \quad \langle u, \vec{A} \rangle \in L(A_\phi)
\]

Decidability:
Prove decidability of the emptyness for the automata.

Buchi’s case
Use the class of Buchi automata.
Only complementation is non-trivial (Based on Ramsey theorem).
FROM MONADIC LOGIC TO AUTOMATA

Step 1: Introduce a suitable family of automata.

Step 2: Prove closure under union, projection, complementation.

Step 3(generic): Prove: for all $\phi$, there exists an automaton $A_\phi$ such that:

$$\text{for all } u, \quad u, \bar{X} = \bar{A} \models \phi \quad \text{iff} \quad \langle u, \bar{A} \rangle \in L(A_\phi)$$

Decidability: Prove decidability of the emptyness for the automata.
**FROM MONADIC LOGIC TO AUTOMATA**

**Step 1:** Introduce a suitable family of automata.

**Step 2:** Prove closure under union, projection, complementation.

**Step 3(generic):** Prove: for all \( \phi \), there exists an automaton \( A_\phi \) such that:

\[
\text{for all } u, \quad u, \overline{X} = \overline{A} \models \phi \quad \text{iff} \quad \langle u, \overline{A} \rangle \in L(A_\phi)
\]

**Decidability:** Prove decidability of the emptyness for the automata.

**Buchi’s case**

Use the class of **Buchi automata**.
FROM MONADIC LOGIC TO AUTOMATA

Step 1: Introduce a suitable family of automata.
Step 2: Prove closure under union, projection, complementation.
Step 3(generic): Prove: for all $\phi$, there exists an automaton $A_\phi$ such that:

$$\text{for all } u, \quad u, \vec{X} = \vec{A} \models \phi \iff \langle u, \vec{A} \rangle \in L(A_\phi)$$

Decidability: Prove decidability of the emptyness for the automata.

Buchi's case

Use the class of Buchi automata.

Only complementation is non-trivial
(Based on Ramsey theorem).
The following properties are equivalent:

- $L$ is the set of $\omega$-words models of a \textit{monadic formula}

  $$\forall x. \exists y. x < y \land b(y)$$
The following properties are equivalent:

- $L$ is the set of $\omega$-words models of a monadic formula
  \[ \forall x. \exists y. x < y \wedge b(y) \]

- $L$ is accepted by a nondeterministic Buchi automaton

- $L$ is the evaluation of a $\omega$-regular expression

- $L$ is recognized by a $\omega$-semigroup morphism

- $L$ is accepted by a Rabin/Street/Muller/Parity automaton (deterministic, nondeterministic or alternating).

- $L$ is modelling a $\omega$-calculus formula.
The following properties are equivalent:

- $L$ is the set of $\omega$-words models of a monadic formula
  
  $$\forall x. \exists y. x < y \land b(y)$$

- $L$ is accepted by a nondeterministic Buchi automaton

- $L$ is the evaluation of an $\omega$-regular expression
  
  $$(a^*b)^\omega$$
The following properties are equivalent:

- \( L \) is the set of \( \omega \)-words models of a monadic formula
  \[
  \forall x. \exists y. x < y \land b(y)
  \]

- \( L \) is accepted by a nondeterministic Buchi automaton

- \( L \) is the evaluation of an \( \omega \)-regular expression
  \[ (a^*b)^\omega \]

- \( L \) is recognized by an \( \omega \)-semigroup morphism
The following properties are equivalent:

- $L$ is the set of $\omega$-words models of a monadic formula
  \[ \forall x. \exists y. x < y \land b(y) \]

- $L$ is accepted by a nondeterministic Buchi automaton

- $L$ is the evaluation of an $\omega$-regular expression
  \[ (a^*b)^\omega \]

- $L$ is recognized by an $\omega$-semigroup morphism

- $L$ is accepted by a Rabin/Street/Muller/Parity automaton (deterministic, nondeterministic or alternating).
The following properties are equivalent:

- $L$ is the set of $\omega$-words models of a monadic formula

$$\forall x. \exists y. x < y \land b(y)$$

- $L$ is accepted by a nondeterministic Buchi automaton

- $L$ is the evaluation of an $\omega$-regular expression

$$(a^*b)^\omega$$

- $L$ is recognized by an $\omega$-semigroup morphism

- $L$ is accepted by a Rabin/Street/Muller/Parity automaton (deterministic, nondeterministic or alternating).

- $L$ is modelling a $\mu$-calculus formula.
EXTENSIONS

Extension of the equivalence logic/automata

- to countable ordinals (Buchi)
- to trees (Rabin)
- to iteration structures (Muchnik, Walukiewicz)
- to countable scattered linear orderings (Carton & Rispal)
EXTENSIONS

Extension of the equivalence logic/automata

- to countable ordinals (Buchi)
- to trees (Rabin)
- to iteration structures (Muchnik, Walukiewicz)
- to countable scattered linear orderings (Carton & Rispal)

Extension of models of decidable monadic theory

- pushdown graphs (Muller & Shupp)
- HR-equational graphs/structures (Courcelle)
- prefix-recognizable graphs/structures (Caucal)
- terms solution of safe HOP schemes (Knapik, Niwinski, Urzyczyn)
- graph of higher-order pushdown systems (Caucal)
- terms solution of HOP schemes (Ong)
EXTENSIONS

Extension of the equivalence logic/automata

- to countable ordinals (Buchi)
- to trees (Rabin)
- to iteration structures (Muchnik, Walukiewicz)
- to countable scattered linear orderings (Carton & Rispal)

Extension of models of decidable monadic theory

- pushdown graphs (Muller & Shupp)
- HR-equational graphs/structures (Courcelle)
- prefix-recognizable graphs/structures (Caucal)
- terms solution of safe HOP schemes (Knapik, Niwinski, Urzyczyn)
- graph of higher-order pushdown systems (Caucal)
- terms solution of HOP schemes (Ong)

Extension of the expressivity of the logic: None going beyond $\omega$-regularity
Extending $\omega$-regularity with bounds
EXTENDING $\omega$-REGULAR EXPRESSIONS

Syntax of $\omega$-regular expressions:

$$R = \emptyset \mid \varepsilon \mid a \mid R + R \mid R.R \mid R^* \quad O = R^\omega \mid R.O \mid O + O$$
Syntax of $\omega BS$-regular expressions:

$$R = \emptyset \mid \varepsilon \mid a \mid R + R \mid R.R \mid R^* \mid R^B \mid R^S \quad O = R^\omega \mid R.O \mid O + O$$
Extending $\omega$-regular expressions:

Syntax of $\omega_{BS}$-regular expressions:

$$R = \emptyset | \varepsilon | a | R + R | R.R | R^* | R^B | R^S \quad O = R^\omega | R.O | O + O$$

Intuitive meaning:

- without $B, S, \omega_{BS}$-regular coincides with $\omega$-regular
EXTENDING $\omega$-REGULAR EXPRESSIONS

Syntax of $\omega BS$-regular expressions:

$$R = \emptyset \ | \ \varepsilon \ | \ a \ | \ R + R \ | \ R.R \ | \ R^* \ | \ R^B \ | \ R^S \quad O = R^\omega \ | \ R.O \ | \ O + O$$

Intuitive meaning:

- without $B, S, \omega BS$-regular coincides with $\omega$-regular
- $B$ and $S$ are variants of the Kleene-star $*$
EXTENDING $\omega$-REGULAR EXPRESSIONS

Syntax of $\omega BS$-regular expressions:

$$R = \emptyset \ | \ \varepsilon \ | \ a \ | \ R + R \ | \ R.R \ | \ R^* \ | \ R^B \ | \ R^S \quad O = R^\omega \ | \ R.O \ | \ O + O$$

Intuitive meaning:

- without $B$, $S$, $\omega BS$-regular coincides with $\omega$-regular
- $B$ and $S$ are variants of the Kleene-star $*$
- $B$ constrains the number of iteration to be bounded
  ($B$ stands for ‘bounded’)
  $$(a^B b)^\omega$$  there is an infinite number of $b$’s and,
  the number of $a$’s between consecutive $b$’s is bounded

$$ababab \cdots \in (a^B b)^\omega$$

$$aba^2baba^3baba^4b \cdots \notin (a^B b)^\omega$$

$$aba^2ba^3ba^4b \cdots \notin (a^B b)^\omega$$
EXTENDING $\omega$-REGULAR EXPRESSIONS

Syntax of $\omega B S$-regular expressions:

$$R = \emptyset \mid \varepsilon \mid a \mid R + R \mid R.R \mid R^* \mid R^B \mid R^S \quad O = R^\omega \mid R.O \mid O + O$$

Intuitive meaning:

- without $B$, $S$, $\omega B S$-regular coincides with $\omega$-regular
- $B$ and $S$ are variants of the Kleene-star $*$
- $B$ constrains the number of iteration to be bounded ($B$ stands for ‘bounded’)
- $S$ constrains the number of iterations to tend toward the infinite ($S$ stands for ‘strictly unbounded’)

$(a^S b)^\omega$ there is an infinite number of $b$’s and,

number of $a$’s between consecutive $b$’s tends toward the infinite

$$ababab\cdots \notin (a^S b)^\omega \quad aba^2baba^3baba^4b\cdots \notin (a^S b)^\omega$$

$$aba^2ba^3ba^4b\cdots \in (a^S b)^\omega$$
SEMANTIC OF $\omega BS$-REGULAR LANGUAGES

$R = \emptyset \mid \epsilon \mid a \mid R + R \mid R.R \mid R^* \mid R^B \mid R^S \quad O = R^\omega \mid R.O \mid O + O$
SEMANTIC OF $\omega BS$-REGULAR LANGUAGES

$R = \emptyset \mid \varepsilon \mid a \mid R + R \mid R.R \mid R^* \mid R^B \mid R^S \quad O = R^\omega \mid R.O \mid O + O$

**Rq:** Standard evaluation of $R$ as a language of finite words. This cannot capture the semantic of $B$ and $S$ exponents.
SEMANTIC OF $\omega BS$-REGULAR LANGUAGES

$$R = \emptyset \mid \varepsilon \mid a \mid R + R \mid R.R \mid R^* \mid R^B \mid R^S \quad O = R^\omega \mid R.O \mid O + O$$

**Rq:** Standard evaluation of $R$ as a language of finite words. This cannot capture the semantic of $B$ and $S$ exponents.

**DEF:** $R$ is evaluated into a language of sequence of finite words. I.e. a subset of $(\Sigma^*)^\omega$
**Semantic of \( \omega BS \)-Regular Languages**

\[
R = \emptyset \mid \varepsilon \mid a \mid R + R \mid R.R \mid R^* \mid R^B \mid R^S \quad O = R^\omega \mid R.O \mid O + O
\]

**Rq:** Standard evaluation of \( R \) as a language of finite words. This cannot capture the semantic of \( B \) and \( S \) exponents.

**Def:** \( R \) is evaluated into a language of sequence of finite words. I.e. a subset of \((\Sigma^*)^\omega\)

- \( a \mapsto \{(a, a, a, a, a, \ldots)\} \)
**Semantic of \( \omega BS \)-Regular Languages**

\[
R = \emptyset \mid \varepsilon \mid a \mid R + R \mid R.R \mid R^* \mid R^B \mid R^S \quad O = R^\omega \mid R.O \mid O + O
\]

**Rq:** Standard evaluation of \( R \) as a language of finite words. This cannot capture the semantic of \( B \) and \( S \) exponents.

**Def:** \( R \) is evaluated into a language of sequence of finite words. I.e. a subset of \((\Sigma^*)^\omega\)

- \( a \mapsto \{(a, a, a, a, a, \ldots)\} \)
- \( U.V \mapsto \{(u_1 v_1, u_2 v_2, \ldots) : \bar{u} \in U, \bar{v} \in V\} \)
**Semantic of \( \omega BS \)-Regular Languages**

\[
R = \emptyset \mid \varepsilon \mid a \mid R + R \mid R.R \mid R^* \mid R^B \mid R^S \quad O = R^\omega \mid R.O \mid O + O
\]

**Rq:** Standard evaluation of \( R \) as a language of finite words. This cannot capture the semantic of \( B \) and \( S \) exponents.

**DEF:** \( R \) is evaluated into a language of sequence of finite words. I.e. a subset of \( (\Sigma^*)^\omega \)

- \( a \mapsto \{(a, a, a, a, a, \ldots )\} \)
- \( U.V \mapsto \{(u_1v_1, u_2v_2, \ldots ) : \bar{u} \in U, \; \bar{v} \in V\} \)
- \( U + V \mapsto \{\bar{w} : \bar{u}, \bar{v} \in U \cup V, \; \forall i. \; w_i \in \{u_i, v_i\}\} \)
SEMANTIC OF $\omega BS$-REGULAR LANGUAGES

\[ R = \emptyset \mid \varepsilon \mid a \mid R + R \mid R.R \mid R^* \mid R^B \mid R^S \quad O = R^\omega \mid R.O \mid O + O \]

Rq: Standard evaluation of $R$ as a language of finite words.
This cannot capture the semantic of $B$ and $S$ exponents.

DEF: $R$ is evaluated into a language of sequence of finite words.
I.e. a subset of $(\Sigma^*)^\omega$

- $a \mapsto \{(a, a, a, a, a, \ldots)\}$
- $U.V \mapsto \{(u_1v_1, u_2v_2, \ldots) : \; \bar{u} \in U, \; \bar{v} \in V\}$
- $U + V \mapsto \{\bar{w} : \; \bar{u}, \bar{v} \in U \cup V, \; \forall i. \; w_i \in \{u_i, v_i\}\}$
- $U^* \mapsto \{(u_0u_1 \ldots u_{\pi(1)-1}, u_{\pi(1)} \ldots u_{\pi(2)} - 1, \ldots) : \; \bar{u} \in U, \; \pi \text{ nondecreasing}\}$
**Semantic of \( \omega B S \)-Regular Languages**

\[
R = \emptyset \mid \varepsilon \mid a \mid R + R \mid R.R \mid R^* \mid R^B \mid R^S \quad O = R^\omega \mid R.O \mid O + O
\]

**Rq:** Standard evaluation of \( R \) as a language of finite words. This cannot capture the semantic of \( B \) and \( S \) exponents.

**DEF:** \( R \) is evaluated into a language of sequence of finite words. I.e. a subset of \((\Sigma^*)^\omega\)

- \( a \mapsto \{(a, a, a, a, a, \ldots)\}\)
- \( U.V \mapsto \{(u_1v_1, u_2v_2, \ldots) : \vec{u} \in U, \vec{v} \in V\}\)
- \( U + V \mapsto \{\vec{w} : \vec{u}, \vec{v} \in U \cup V, \forall i. w_i \in \{u_i, v_i\}\}\)
- \( U^* \mapsto \{(u_0u_1 \ldots u_{\pi(1)} - 1, u_{\pi(1)} \ldots u_{\pi(2)} - 1, \ldots) : \vec{u} \in U, \pi \text{ nondecreasing}\}\)
- \( U^B \mapsto \text{idem} + (\pi(i + 1) - \pi(i)) \text{ is bounded} \)
Rq: Standard evaluation of $R$ as a language of finite words. This cannot capture the semantic of $B$ and $S$ exponents.

DEF: $R$ is evaluated into a language of sequence of finite words. I.e. a subset of $(\Sigma^*)^\omega$

- $a \mapsto \{(a, a, a, a, a, \ldots)\}$
- $U.V \mapsto \{(u_1 v_1, u_2 v_2, \ldots) : \vec{u} \in U, \vec{v} \in V\}$
- $U + V \mapsto \{\vec{w} : \vec{u}, \vec{v} \in U \cup V, \forall i. w_i \in \{u_i, v_i\}\}$
- $U^* \mapsto \{(u_0 u_1 \ldots u_{\pi(1) - 1}, u_{\pi(1)} \ldots u_{\pi(2) - 1}, \ldots) : \vec{u} \in U, \pi \text{ nondecreasing}\}$
- $U^B \mapsto \text{idem} + (\pi(i + 1) - \pi(i)) \text{ is bounded}$
- $U^S \mapsto \text{idem} + (\pi(i + 1) - \pi(i)) \to +\infty$
**Semantic of \( \omega B S \)-Regular Languages**

\[
R = \emptyset \mid \varepsilon \mid a \mid R + R \mid R.R \mid R^* \mid R^B \mid R^S \quad \quad O = R^\omega \mid R.O \mid O + O
\]

**Rq:** Standard evaluation of \( R \) as a language of finite words. This cannot capture the semantic of \( B \) and \( S \) exponents.

**Def:** \( R \) is evaluated into a language of sequence of finite words. I.e. a subset of \( (\Sigma^*)^\omega \)

- \( a \mapsto \{(a, a, a, a, a, \ldots)\} \)
- \( U.V \mapsto \{(u_1v_1, u_2v_2, \ldots) : \, \vec{u} \in U, \, \vec{v} \in V\} \)
- \( U + V \mapsto \{\vec{w} : \, \vec{u}, \vec{v} \in U \cup V, \, \forall i. \, w_i \in \{u_i, v_i\}\} \)
- \( U^* \mapsto \{(u_0 u_1 \ldots u_{\pi(1)} - 1, u_{\pi(1)} \ldots u_{\pi(2)} - 1, \ldots) : \, \vec{u} \in U, \, \pi \text{ nondecreasing}\} \)
- \( U^B \mapsto \text{idem} + (\pi(i+1) - \pi(i)) \text{ is bounded} \)
- \( U^S \mapsto \text{idem} + (\pi(i+1) - \pi(i)) \to +\infty \)
- \( U^\omega \mapsto \{u_1 u_2 \ldots : \, \vec{u} \in U\} \)
OTHER EXAMPLES

$((a^*b)^*a^Sb)^\omega$
Other Examples

\[((a^* b)^* a^S b)^\omega\]

The languages with an infinite number of $b$'s such that the size of segments of $a$ is not bounded.
E.g. $aba^2 baba^3 baba^4 b \ldots \in ((a^* b)a^S b)^\omega$
The languages with an infinite number of $b$'s such that the size of segments of $a$ is not bounded.
E.g. $aba^2bababa^3bababa^4b \cdots \in ((a^*b)a^Sb)\omega$

\[ ((a^*b)^*a^Sb)^\omega \]

\[ ((a^Sb)^S \alpha^S \beta)^\omega \]
((a*b)^* a^S b)^\omega

The languages with an infinite number of b’s such that the size of segments of a is not bounded.
E.g. \(aba^2 baba^3 baba^4 b \cdots \in ((a*b)a^S b)^\omega\)

\(((a^S b)^S a^S c)^\omega\)

There is an infinite number of c’s. The number of b’s between two c’s tends toward the infinite. The number of a’s between two b or c’s tends toward the infinite.
THE DIAMOND

\[ \omega B S \text{-regular expressions} \]

- e.g. \( (a^B b^S c)^\omega \)

\[ \omega S \text{-regular expressions} \]

- e.g. \( (a^S b)^\omega \)

\[ \omega B \text{-regular expressions} \]

- e.g. \( (a^B b)^\omega \)

\[ \omega \text{-regular expressions} \]

- e.g. \( (a^* b)^\omega \)

\text{PROP: Emptyness of } \omega B S \text{-regular languages is decidable}

\text{PROP: The inclusions in the diamond are strict.}
THE DIAMOND

\[ \omega BS\text{-regular expressions} \]
\[ \text{e.g. } (a^B b^S c) \omega \]
\[ \subset \]

\[ \omega S\text{-regular expressions} \]
\[ \text{e.g. } (a^S b) \omega \]
\[ \subset \]

\[ \omega B\text{-regular expressions} \]
\[ \text{e.g. } (a^B b) \omega \]
\[ \subset \]

\[ \omega\text{-regular expressions} \]
\[ \text{e.g. } (a^* b) \omega \]

**PROP:** Emptyness of \( \omega BS \)-regular languages is decidable
THE DIAMOND

\[ \omega BS\text{-regular expressions} \]
\[ \text{e.g. } (a^B b^S c)^\omega \]

\[ \omega S\text{-regular expressions} \]
\[ \text{e.g. } (a^S b)^\omega \]

\[ \omega B\text{-regular expressions} \]
\[ \text{e.g. } (a^B b)^\omega \]

\[ \omega \text{-regular expressions} \]
\[ \text{e.g. } (a^* b)^\omega \]

**PROP:** Emptyness of \( \omega BS \text{-regular languages} \) is decidable

**PROP:** The inclusions in the diamond are strict.
**DEF:** One can define $\omega B$, $\omega S$, $\omega BS$-automata. Essentially: finite state automata with modified accepting condition (more expressive than Buchi). They come in two variants hierarchical or not.
**DEF:** One can define $\omega B$, $\omega S$, $\omega BS$-automata. Essentially: finite state automata with modified accepting condition (more expressive than Buchi). They come in two variants hierarchical or not.

$\omega BS$-automaton =
- finite automaton
- finite set of counters, of kind $B$ or $S$
- each counter can be left unchanged or reset or incremented

**THM:** The following are equivalent:
- $L$ is evaluation of an $\omega BS$-regular expression
- $L$ is accepted by an $\omega BS$-automaton
- $L$ is accepted by a hierarchical $\omega BS$-automaton

And the same holds for $\omega B$ and $\omega S$ regular languages.
DEF: One can define $\omega B$, $\omega S$, $\omega BS$-automata. Essentially: finite state automata with modified accepting condition (more expressive than Buchi). They come in two variants hierarchical or not.

$\omega BS$-automaton =
- finite automaton
- finite set of counters, of kind B or S
- each counter can be left unchanged or reset or incremented

A run is accepting if
- every counter is reset infinitely many times
- every $B$-counter is incremented a bounded number of times between resets
- the number of increments of an $S$-counter between reset tends toward the infinite
**DEF:** One can define \( \omega B, \omega S, \omega BS \)-automata. Essentially: finite state automata with modified accepting condition (more expressive than Buchi). They come in two variants hierarchical or not.

**THM:** The following are equivalent:
- \( L \) is evaluation of an \( \omega BS \)-regular expression
- \( L \) is accepted by an \( \omega BS \)-automaton
- \( L \) is accepted by a hierarchical \( \omega BS \)-automaton

And the same holds for \( \omega B \) and \( \omega S \) regular languages.
THE DIAMOND (2: AUTOMATA)

ωBS-regular expressions
hierarchical ωBS-automata
ωBS-automata

ωS-regular expressions
hierarchical ωS-automata
ωS-automata

ωB-regular expressions
hierarchical ωB-automata
ωB-automata

ω-regular expressions
Büchi automata
**THM:** The $\omega BS$-regular languages are closed under union, intersection, projection.

**PROP:** $\omega BS$-regular languages are not closed under complementation.

**MP:** The complement of $L = ((a B a)^* B a)^*$ is not $\omega BS$-regular.

**E.g.** The language $\{L | L$ contains $a f(1) b a f(2) b a f(3) b a \}$ iff there exists infinitely many values appearing infinitely often in $f$.
**THM:** The $\omega BS$-regular languages are closed under union, intersection, projection.

**Proof:** Union and projection are syntactic on $\omega BS$-regular expressions. Intersection is obtained by product of $\omega BS$-automata.
**THM:** The \( \omega BS \)-regular languages are closed under union, intersection, projection.

**Proof:** Union and projection are syntactic on \( \omega BS \)-regular expressions. Intersection is obtained by product of \( \omega BS \)-automata.

**Example of intersections:**

\[
((b^*ab^*)B \#)^\omega \cap ((a^*ba^*)B \#)^\omega = ((a + b)^B \#)^\omega
\]
**THM:** The $\omega BS$-regular languages are closed under union, intersection, projection.

**Proof:** Union and projection are syntactic on $\omega BS$-regular expressions. Intersection is obtained by product of $\omega BS$-automata.

**Example of intersections:**

\[
\begin{align*}
((b^* a b^*)^B \#)^\omega \cap ((a^* b a^*)^B \#)^\omega &= ((a + b)^B \#)^\omega \\
((b^* a b^*)^S \#)^\omega \cap ((a^* b a^*)^B \#)^\omega &= ((a^* b a^*)^B a^S (a^* b a^*)^B \#)^\omega
\end{align*}
\]
**THM:** The $\omega BS$-regular languages are closed under union, intersection, projection.

**Proof:** Union and projection are syntactic on $\omega BS$-regular expressions. Intersection is obtained by product of $\omega BS$-automata.

**Example of intersections:**

\[
((b^* ab^*)^B \#)^\omega \cap ((a^* ba^*)^B \#)^\omega = ((a + b)^B \#)^\omega
\]

\[
((b^* ab^*)^S \#)^\omega \cap ((a^* ba^*)^B \#)^\omega = ((a^* ba^*)^B a^S (a^* ba^*)^B \#)^\omega
\]

\[
((b^* ab^*)^S \#)^\omega \cap ((a^* ba^*)^S \#)^\omega = \left( \begin{array}{c} (a + b)^* a^S (a + b)^* b^S (a + b)^* \\ + \ (a + b)^* b^S (a + b)^* a^S (a + b)^* \\ + \ b^* (a^+ b^+)^S a^* \end{array} \right)^\#
\]
**THM:** The $\omega BS$-regular languages are closed under union, intersection, projection.

**PROP:** $\omega BS$-regular languages are not closed under complementation.
**THM:** The $\omega BS$-regular languages are closed under union, intersection, projection.

**PROP:** $\omega BS$-regular languages are not closed under complementation.

**Mp:** The complement of $L = ((a^B + a^S)^\#)^\omega$ is not $\omega BS$-regular.

The language $CL$ contains $a^{f(1)}ba^{f(2)}b \cdots$ iff there exists infinitely many values appearing infinitely often in $f$.

**E.g:** $a^1b$, $a^1ba^2b$, $a^1ba^2ba^3b$, $\cdots \in CL$
**CLOSURE**

**THM:** The $\omega BS$-regular languages are closed under union, intersection, projection.

**PROP:** $\omega BS$-regular languages are not closed under complementation.

**THM:** The complement of an $\omega B$-regular language is $\omega S$-regular
The complement of an $\omega S$-regular language is $\omega B$-regular
THEM: The $\omega BS$-regular languages are closed under union, intersection, projection.

PROP: $\omega BS$-regular languages are not closed under complementation.

THEM: The complement of an $\omega B$-regular language is $\omega S$-regular

The complement of an $\omega S$-regular language is $\omega B$-regular

Example:

$$\mathcal{C}((a^B b)^\omega) = (a + b)^* a^\omega + ((a^* b)^* a^S b)^\omega$$
**THM:** The $\omega BS$-regular languages are closed under union, intersection, projection.

**PROP:** $\omega BS$-regular languages are not closed under complementation.

**THM:** The complement of an $\omega B$-regular language is $\omega S$-regular

The complement of an $\omega S$-regular language is $\omega B$-regular

**Example:**

\[
\mathcal{C}((a^S b)\omega) = (a + b)^* a^\omega + ((a^* b)^* a^S b)^\omega
\]

\[
\mathcal{C}((a^B b)\omega) = (a + b)^* a^\omega + ((a^* b)^* a^B b)^\omega
\]
THE DIAMOND (3: CLOSURE)

union, intersection, projection

$\omega B S$-regular

union, intersection, projection

$\omega S$-regular

complementation

$\omega B$-regular

union, intersection, projection

$\omega$-regular

union, intersection, complementation, projection
The logic MSOL:

\[ \phi = \phi \lor \phi \mid \phi \land \phi \mid \neg \phi \mid \exists X . \phi \mid \exists x . \phi \mid x \in X \mid x = S(y) \mid a(x) \]
The logic **MSOLB** (Bojanczyk05):

\[
\phi = \phi \lor \phi \mid \phi \land \phi \mid \neg \phi \mid \exists X. \phi \mid \exists x. \phi \mid x \in X \mid x = S(y) \mid a(x) \mid \exists X. \phi
\]

With \( \exists X. \phi \equiv \forall n. \exists X. (|X| > n) \land \phi \)

"there exists a set as big as I want"
The logic **MSOLB** (Bojanczyk05):

\[
\phi = \phi \lor \phi \mid \phi \land \phi \mid \neg \phi \mid \exists X. \phi \mid \exists x. \phi \mid x \in X \mid x = S(y) \mid a(x) \mid \bigcup X. \phi
\]

With \( \bigcup X. \phi \equiv \forall n. \exists X. (|X| > n) \land \phi \)

“there exists a set as big as I want”

And also \( \Delta X. \phi \equiv \neg \bigcup X. \neg \phi \equiv \exists n. \forall X. (|X| > n) \rightarrow \phi \)

“for all sets above a certain size”
The logic MSOLB (Bojanczyk05):

\[ \phi = \phi \lor \phi \mid \phi \land \phi \mid \neg \phi \mid \exists X. \phi \mid \exists x. \phi \mid x \in X \mid x = S(y) \mid a(x) \mid \forall X. \phi \]

With \[ \forall X. \phi \equiv \forall n. \exists X. (|X| > n) \land \phi \]

“there exists a set as big as I want”

And also \[ \forall X. \phi \equiv \neg \forall X. \neg \phi \equiv \exists n. \forall X. (|X| > n) \rightarrow \phi \]

“for all sets above a certain size”

And also \[ \exists X. \phi \equiv \neg \exists X. \phi \equiv \exists n. \forall X. \phi \rightarrow (|X| < n) \]

“there is a bound on the size of sets satisfying”
The logic **MSOLB** (Bojanczyk05):

\[ \phi = \phi \lor \phi \mid \phi \land \phi \mid \neg \phi \mid \exists X. \phi \mid \exists x. \phi \mid x \in X \mid x = S(y) \mid a(x) \mid \bigcup X. \phi \]

With \( \bigcup X. \phi \equiv \forall n. \exists X. (|X| > n) \land \phi \)

“there exists a set as big as I want”

And also \( \bigtriangleup X. \phi \equiv \neg \bigcup X. \neg \phi \equiv \exists n. \forall X. (|X| > n) \rightarrow \phi \)

“for all sets above a certain size”

And also \( \bigtriangledown X. \phi \equiv \neg \bigcup X. \phi \equiv \exists n. \forall X. \phi \rightarrow (|X| < n) \)

“there is a bound on the size of sets satisfying”

**THM(Buchi):** SAT of MSOL is decidable over infinite words.
The logic **MSOLB** (Bojanczyk05): 

\[ \phi = \phi \lor \phi \mid \phi \land \phi \mid \neg \phi \mid \exists X. \phi \mid \exists x. \phi \mid x \in X \mid x = S(y) \mid a(x) \mid \mathcal{U}X. \phi \]

With \( \mathcal{U}X. \phi \equiv \forall n. \exists X. (|X| > n) \land \phi \)

"there exists a set as big as I want"

And also \( \mathcal{A}X. \phi \equiv \neg \mathcal{U}X. \neg \phi \equiv \exists n. \forall X. (|X| > n) \rightarrow \phi \)

"for all sets above a certain size"

And also \( \mathcal{B}X. \phi \equiv \neg \mathcal{U}X. \phi \equiv \exists n. \forall X. \phi \rightarrow (|X| < n) \)

"there is a bound on the size of sets satisfying"

**THM(Buchi):** SAT of MSOL is decidable over infinite words.

**Rq:** Adding equality/comparison of cardinality to MSOL leads to undecidability of SAT (already for finite words).
The logic **MSOLB** (Bojanczyk05):

\[ \phi = \phi \lor \phi \mid \phi \land \phi \mid \neg \phi \mid \exists X . \phi \mid \exists x . \phi \mid x \in X \mid x = S(y) \mid a(x) \mid \mathsf{U} X . \phi \]

With \( \mathsf{U} X . \phi \equiv \forall n . \exists X . (|X| > n) \land \phi \)

"there exists a set as big as I want"

And also \( \mathsf{A} X . \phi \equiv \neg \mathsf{U} X . \neg \phi \equiv \exists n . \forall X . (|X| > n) \rightarrow \phi \)

"for all sets above a certain size"

And also \( \mathsf{B} X . \phi \equiv \neg \mathsf{U} X . \phi \equiv \exists n . \forall X . \phi \rightarrow (|X| < n) \)

"there is a bound on the size of sets satisfying"

**THM(Buchi):** SAT of MSOL is decidable over infinite words.

**Rq:** Adding equality/comparison of cardinality to MSOL leads to undecidability of SAT (already for finite words).

**QUESTION:** Is SAT of MSOLB decidable over \( \omega \)-words?
The logic **MSOLB** (Bojanczyk05):

\[
\phi = \phi \lor \phi \mid \phi \land \phi \mid \neg \phi \mid \exists X. \phi \mid \exists x. \phi \mid x \in X \mid x = S(y) \mid a(x) \mid \forall X. \phi
\]

With \( \forall X. \phi \equiv \forall n. \exists X. (|X| > n) \land \phi \)

"there exists a set as big as I want"

And also \( \exists X. \phi \equiv \neg \forall X. \neg \phi \equiv \exists n. \forall X. (|X| > n) \rightarrow \phi \)

"for all sets above a certain size"

And also \( \exists X. \phi \equiv \neg \forall X. \phi \equiv \exists n. \forall X. \phi \rightarrow (|X| < n) \)

"there is a bound on the size of sets satisfying"

**THM(Buchi):** SAT of MSOL is decidable over infinite words.

**Rq:** Adding equality/comparison of cardinality to MSOL leads to undecidability of SAT (already for finite words).

**QUESTION:** Is SAT of MSOLB decidable over \( \omega \)-words? The question is open.

But \( \omega BS \)-regularity provides a partial answer.
and $\omega BS$-regular languages are closed under $\cup$.
Proposition: \( \omegaBS\text{-regular}\) languages are closed under intersection and union.
The Diamond (3:Logic)

\[ \forall, \land, \exists \]

- \( \omega BS \)-regular

- \( \land, \exists, \forall \)

- \( \omega S \)-regular

- \( \neg \)

- \( \omega B \)-regular

- \( \forall, \land, \exists \)

- \( \omega \)-regular

- \( \forall, \land, \neg, \exists, \forall \)

Prop: \( BS \)-regular languages are closed under \( U \).

Omega Regularity with Bounds – p.16
\( \forall, \land, \exists \)

\( \omega BS \)-regular

\( \forall, \land, \exists, \forall \)

\( \omega S \)-regular

\( \neg \)

\( \omega B \)-regular

\( \forall, \land, \neg, \exists, \forall \)

\( \forall, \land, \exists, \forall \)

\( \omega \)-regular

\( \forall, \land, \exists, \forall \)
PROP: $\omega S$ and $\omega BS$-regular languages are closed under $\bigcup$. 

The Diamond (3:logic)
PROP: $\omega S$ and $\omega BS$-regular languages are closed under $\cup$.
SAT OF MSOLB

DEF:

MSOLB+ = MSOLB where \( \uparrow \) appears only \textbf{positively}

MSOLB- = MSOLB where \( \uparrow \) appears only \textbf{negatively}
**SAT OF MSOLB**

**DEF:**
- $\text{MSOLB}^+$ = MSOLB where $\exists$ appears only **positively**
- $\text{MSOLB}^-$ = MSOLB where $\exists$ appears only **negatively**

**COR:**
- MSOL is equivalent to $\omega$-regular languages
- $\text{MSOLB}^+$ is equivalent to $\omega S$-regular languages
- $\text{MSOLB}^-$ is equivalent to $\omega B$-regular languages
- Boolean comb. of $\text{MSOLB}^+$ are contained in $\omega BS$-regular languages

$\Rightarrow$ **SAT is decidable for those fragments of MSOLB**
We have:

- Introduced an extension of $\omega$-regular expressions.
We have:

-Introduced an extension of $\omega$-regular expressions.
-Introduced corresponding class of automata.

Open questions:
- Solve the full logic MSOLB over $\omega$-words.
- Find equivalent class of languages over trees.
- What about the logic GMSOLB?
CONCLUSION

We have:

- Introduced an extension of $\omega$-regular expressions.
- Introduced corresponding class of automata.
- Shown decidability and closure properties.

Open questions.

- Solve the full logic MSOLB over $\omega$-words.
- Find equivalent class of languages over trees.
- What about the logic GMSOLB?
CONCLUSION

We have:
- Introduced an extension of $\omega$-regular expressions.
- Introduced corresponding class of automata.
- Shown decidability and closure properties.
- Used it for solving a fragment of the logic MSOLB over $\omega$-words.

Open questions.
- Solve the full logic MSOLB over $\omega$-words.
- Find equivalent class of languages over trees.
- What about the logic GMSOLB?
CONCLUSION

We have:

- Introduced an extension of $\omega$-regular expressions.
- Introduced corresponding class of automata.
- Shown decidability and closure properties.
- Used it for solving a fragment of the logic MSOLB over $\omega$-words.

Open questions.

- Solve the full logic MSOLB over $\omega$-words.
CONCLUSION

We have:

- Introduced an extension of $\omega$-regular expressions.
- Introduced corresponding class of automata.
- Shown decidability and closure properties.
- Used it for solving a fragment of the logic MSOLB over $\omega$-words.

Open questions.

- Solve the full logic MSOLB over $\omega$-words.
- Find equivalent class of languages over trees.
CONCLUSION

We have:

- Introduced an extension of $\omega$-regular expressions.
- Introduced corresponding class of automata.
- Shown decidability and closure properties.
- Used it for solving a fragment of the logic MSOLB over $\omega$-words.

Open questions.

- Solve the full logic MSOLB over $\omega$-words.
- Find equivalent class of languages over trees.
- What about the logic **GMSOLB**?:

\[
\phi = \phi \lor \phi \mid \phi \land \phi \mid \neg \phi \mid \exists X.\phi \mid \exists x.\phi \mid x \in X \mid x = S(y) \mid a(x) \\
\mid \exists M.\phi \\
\mid |X| < M \text{ (positively below } \exists M)\]

Omega Regularity with Bounds – p.18
Q?: Given a regular language of finite words, how many nesting of kleene star are required in a regular expression for describing it? This number is called the (restricted) star height of the language. Among the most important decision problems in language theory.

Rq: There are infinitely many languages of star height $k$. 
**EPILOGUE: RESTRICTED STAR-HEIGHT PROBLEM**

**Q?:** Given a regular language of finite words, how many nesting of kleene star are required in a regular expression for describing it? This number is called the (restricted) star height of the language. Among the most important decision problems in language theory.

**Rq:** There are infinitely many languages of star height \( k \).

**Solution 1:** Hashigushi gave a first algorithm, complex and difficult.
Q?: Given a regular language of finite words, how many nesting of kleene star are required in a regular expression for describing it? This number is called the (restricted) star height of the language. Among the most important decision problems in language theory.

Rq: There are infinitely many languages of star height \( k \).

Solution 1: Hashigushi gave a first algorithm, complex and difficult.

Solution 2: Kirsten gave an optimal solution by reduction to the limitedness problem for nested distance desert automata.
**Q?:** Given a regular language of finite words, how many nesting of kleene star are required in a regular expression for describing it? This number is called the (restricted) star height of the language. Among the most important decision problems in language theory.

**Rq:** There are infinitely many languages of star height \( k \).

**Solution 1:** Hashigushi gave a first algorithm, complex and difficult.

**Solution 2:** Kirsten gave an optimal solution by reduction to the limitedness problem for nested distance desert automata.

**In our framework**

Nested distance desert automata \( \simeq \) hierarchical \( \omega B \)-automata
**EPILOGUE: RESTRICTED STAR-HEIGHT PROBLEM**

**Q?:** Given a regular language of finite words, how many nesting of kleene star are required in a regular expression for describing it? This number is called the (restricted) star height of the language. Among the most important decision problems in language theory.

**Rq:** There are infinitely many languages of star height $k$.

**Solution 1:** Hashigushi gave a first algorithm, complex and difficult.

**Solution 2:** Kirsten gave an optimal solution by reduction to the limitedness problem for nested distance desert automata.

**In our framework**

Nested distance desert automata $\simeq$ hierarchical $\omega B$-automata

**Limitedness problem:** $L(A) = L(A[*/B])$?
This is decidable by our results.
Thank you.