Omega Regularity with Bounds

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

Séminaire automate au LIAFA le 2 mars 2007

Omega Regular languages

An ω-word is an infinite word indexed by ω: $a_1 a_2 a_3 \ldots$

Omega Regular languages

An ω-word is an infinite word indexed by ω: $a_1 a_2 a_3 \ldots$

Monadic (second-order) logic:

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

Omega Regular languages

An ω-word is an infinite word indexed by ω: $a_1 a_2 a_3 \ldots$

Monadic (second-order) logic:

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

E.g:

$$x \leq y \quad := \quad \forall X.(x \in X \wedge (\forall z.z \in X \rightarrow S(z) \in X)) \rightarrow y \in X$$

Omega Regular languages

An ω-word is an infinite word indexed by ω: $a_1 a_2 a_3 \ldots$

Monadic (second-order) logic:

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

Omega Regular languages

An ω-word is an infinite word indexed by ω: $a_1 a_2 a_3 \dots$

Monadic (second-order) logic:

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

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

OMEGA REGULAR LANGUAGES

An ω-word is an infinite word indexed by ω: $a_1 a_2 a_3 \ldots$

Monadic (second-order) logic:

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

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

Idea: Transform each monadic formula into an ‘equivalent’ automaton.

From monadic logic to automata

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.

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 ϕ, there exists an automaton $\mathcal{A}_\phi$ such that:

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

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 ϕ, there exists an automaton $\mathcal{A}_\phi$ such that:

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

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 ϕ, there exists an automaton $\mathcal{A}_\phi$ such that:

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

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

$$u, \vec{X} = \vec{A} \models \phi \wedge \psi \qquad \text{iff } \langle u, \vec{A} \rangle \in L(\mathcal{A}_\phi) \cap L(\mathcal{A}_\psi)$$

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 ϕ, there exists an automaton $\mathcal{A}_\phi$ such that:

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

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

$$u, \vec{X} = \vec{A} \models \phi \wedge \psi \qquad \text{iff } \langle u, \vec{A} \rangle \in L(\mathcal{A}_\phi) \cap L(\mathcal{A}_\psi)$$

$$u, \vec{X} = \vec{A} \models \neg\phi \qquad \text{iff } \langle u, \vec{A} \rangle \in \mathtt{C}(L(\mathcal{A}_\phi))$$

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 ϕ, there exists an automaton $\mathcal{A}_\phi$ such that:

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

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

$$u, \vec{X} = \vec{A} \models \phi \wedge \psi \qquad \text{iff } \langle u, \vec{A} \rangle \in L(\mathcal{A}_\phi) \cap L(\mathcal{A}_\psi)$$

$$u, \vec{X} = \vec{A} \models \neg\phi \qquad \text{iff } \langle u, \vec{A} \rangle \in \complement(L(\mathcal{A}_\phi))$$

$$u, \vec{X} = \vec{A} \models \exists Y.\phi \qquad \text{iff } \langle u, \vec{A} \rangle \in \pi(L(\mathcal{A}_\phi)) \quad (\pi \text{ projection cancelling } Y)$$

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 ϕ, there exists an automaton $\mathcal{A}_\phi$ such that:

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

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

$$u, \vec{X} = \vec{A} \models \phi \wedge \psi \qquad \text{iff } \langle u, \vec{A} \rangle \in L(\mathcal{A}_\phi) \cap L(\mathcal{A}_\psi)$$

$$u, \vec{X} = \vec{A} \models \neg\phi \qquad \text{iff } \langle u, \vec{A} \rangle \in \mathtt{C}(L(\mathcal{A}_\phi))$$

$$u, \vec{X} = \vec{A} \models \exists Y.\phi \qquad \text{iff } \langle u, \vec{A} \rangle \in \pi(L(\mathcal{A}_\phi)) \quad (\pi \text{ projection cancelling } Y)$$

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 ϕ, there exists an automaton $\mathcal{A}_\phi$ such that:

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

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 ϕ, there exists an automaton $\mathcal{A}_\phi$ such that:

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

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 ϕ, there exists an automaton $\mathcal{A}_\phi$ such that:

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

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 ϕ, there exists an automaton $\mathcal{A}_\phi$ such that:

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

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).

A MORE GENERAL VIEW OF THE PICTURE

THM(Buchi,...)The following properties are equivalent:

- L is the set of ω-words models of a monadic formula

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

A MORE GENERAL VIEW OF THE PICTURE

THM(Buchi,. . .)The following properties are equivalent:

- L is the set of ω-words models of a monadic formula

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

- L is accepted by a nondeterministic Buchi automaton

A more general view of the picture

THM(Buchi,...)The following properties are equivalent:

- L is the set of ω-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 an ω-regular expression

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

A more general view of the picture

THM(Buchi,. . .)The following properties are equivalent:

- L is the set of ω-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 an ω-regular expression

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

- L is recognized by an ω-semigroup morphism

A MORE GENERAL VIEW OF THE PICTURE

THM(Buchi,. . .)The following properties are equivalent:

- L is the set of ω-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 an ω-regular expression

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

- L is recognized by an ω-semigroup morphism
- L is accepted by a Rabin/Street/Muller/Parity automaton (deterministic, nondeterministic or alternating).

A MORE GENERAL VIEW OF THE PICTURE

THM(Buchi,. . .)The following properties are equivalent:

- L is the set of ω-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 an ω-regular expression

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

- L is recognized by an ω-semigroup morphism
- L is accepted by a Rabin/Street/Muller/Parity automaton (deterministic, nondeterministic or alternating).
- L is modelling a μ-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 beyong ω-regularity

Extending ω-regularity with bounds

EXTENDING ω-REGULAR EXPRESSIONS

Syntax of ω-regular expressions:

$$R = \emptyset \ \mid \varepsilon \ \mid a \ \mid R + R \mid R.R \ \mid R^* \qquad\qquad O = R^\omega \ \mid R.O \ \mid O + O$$

EXTENDING ω-REGULAR EXPRESSIONS

Syntax of ωBS-regular expressions:

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

EXTENDING ω-REGULAR EXPRESSIONS

Syntax of ωBS-regular expressions:

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

Intuitive meaning:

- without B, S, ωBS-regular coincides with ω-regular

Extending ω-regular expressions

Syntax of ωBS-regular expressions:

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

Intuitive meaning:

- without B, S, ωBS-regular coincides with ω-regular
- B and S are variants of the Kleene-star $*$

EXTENDING ω-REGULAR EXPRESSIONS

Syntax of ωBS-regular expressions:

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

Intuitive meaning:

- without B, S, ωBS-regular coincides with ω-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 ω-REGULAR EXPRESSIONS

Syntax of ωBS-regular expressions:

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

Intuitive meaning:

- without B, S, ωBS-regular coincides with ω-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 infir

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

$$aba^2ba^3ba^4b\cdots \in (a^S b)^\omega$$

SEMANTIC OF ωBS-REGULAR LANGUAGES

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

SEMANTIC OF ωBS-REGULAR LANGUAGES

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

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

SEMANTIC OF ωBS-REGULAR LANGUAGES

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

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 ωBS-REGULAR LANGUAGES

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

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 ωBS-REGULAR LANGUAGES

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

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 $(\Sigma^*)^\omega$

- $a \mapsto \{(a, a, a, a, a, \dots)\}$
- $U.V \mapsto \{(u_1 v_1, u_2 v_2, \dots) \;:\; \vec{u} \in U,\;\; \vec{v} \in V\}$

SEMANTIC OF ωBS-REGULAR LANGUAGES

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

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 $(\Sigma^*)^\omega$

- $a \mapsto \{(a,a,a,a,a,\dots)\}$
- $U.V \mapsto \{(u_1v_1, u_2v_2, \dots) : \vec{u} \in U,\ \vec{v} \in V\}$
- $U+V \mapsto \{\vec{w} : \vec{u}, \vec{v} \in U \cup V,\ \forall i.\ w_i \in \{u_i, v_i\})$

SEMANTIC OF ωBS-REGULAR LANGUAGES

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

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 $(\Sigma^*)^\omega$

- $a \mapsto \{(a, a, a, a, a, \ldots)\}$
- $U.V \mapsto \{(u_1 v_1, u_2 v_2, \ldots) \;:\; \vec{u} \in U,\;\; \vec{v} \in V\}$
- $U + V \mapsto \{\vec{w} \;:\; \vec{u}, \vec{v} \in U \cup V,\; \forall i.\; w_i \in \{u_i, v_i\})$
- $U^* \mapsto \{(u_0 u_1 \ldots u_{\pi(1)-1}, u_{\pi(1)} \ldots u_{\pi(2)} - 1, \ldots) \;:\; \vec{u} \in U,\; \pi \; nondecreasing\}$

Semantic of ωBS-regular languages

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

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 $(\Sigma^*)^\omega$

- $a \mapsto \{(a, a, a, a, a, \dots)\}$
- $U.V \mapsto \{(u_1 v_1, u_2 v_2, \dots) \;:\; \vec{u} \in U,\;\; \vec{v} \in V\}$
- $U + V \mapsto \{\vec{w} \;:\; \vec{u}, \vec{v} \in U \cup V,\ \forall i.\ w_i \in \{u_i, v_i\})$
- $U^* \mapsto \{(u_0 u_1 \dots u_{\pi(1)-1}, u_{\pi(1)} \dots u_{\pi(2)} - 1, \dots) \;:\; \vec{u} \in U,\ \pi\ nondecreasing\}$
- $U^B \mapsto$ idem + $(\pi(i+1) - \pi(i))$ is bounded

Semantic of ωBS-regular languages

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

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 $(\Sigma^*)^\omega$

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

SEMANTIC OF ωBS-REGULAR LANGUAGES

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

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 $(\Sigma^*)^\omega$

- $a \mapsto \{(a,a,a,a,a,\ldots)\}$
- $U.V \mapsto \{(u_1v_1, u_2v_2,\ldots) \;:\; \vec{u}\in U,\;\; \vec{v}\in V\}$
- $U+V \mapsto \{\vec{w} \;:\; \vec{u},\vec{v}\in U\cup V,\; \forall i.\; w_i\in\{u_i,v_i\}\})$
- $U^* \mapsto \{(u_0u_1\ldots u_{\pi(1)-1}, u_{\pi(1)}\ldots u_{\pi(2)}-1,\ldots) \;:\; \vec{u}\in U,\; \pi\; nondecreasing\}$
- $U^B \mapsto$ idem + $(\pi(i+1)-\pi(i))$ is bounded
- $U^S \mapsto$ idem + $(\pi(i+1)-\pi(i)) \to +\infty$
- $U^\omega \mapsto \{u_1u_2\ldots \;:\; \vec{u}\in U\}$

Other examples

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

OTHER EXAMPLES

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

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

OTHER EXAMPLES

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

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

$$((a^Sb)^Sa^Sc)^\omega$$

OTHER EXAMPLES

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

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

$$((a^Sb)^Sa^Sc)^\omega$$

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

ωBS-regular expressions
e.g. $(a^B b^S c)^\omega$

ωS-regular expressions $\subset$ ωBS-regular expressions

ωB-regular expressions $\subset$ ωBS-regular expressions

ωS-regular expressions
e.g. $(a^S b)^\omega$

ωB-regular expressions
e.g. $(a^B b)^\omega$

ω-regular expressions $\subset$ ωS-regular expressions

ω-regular expressions $\subset$ ωB-regular expressions

ω-regular expressions
e.g. $(a^* b)^\omega$

THE DIAMOND

ωBS-regular expressions
e.g. $(a^B b^S c)^\omega$

$\subseteq$ $\supseteq$

ωS-regular expressions
e.g. $(a^S b)^\omega$

ωB-regular expressions
e.g. $(a^B b)^\omega$

$\supseteq$ $\subseteq$

ω-regular expressions
e.g. $(a^* b)^\omega$

PROP: Emptyness of ωBS-regular languages is decidable

The diamond

ωBS-regular expressions
e.g. $(a^B b^S c)^\omega$

$\subset$ $\supset$

ωS-regular expressions
e.g. $(a^S b)^\omega$

ωB-regular expressions
e.g. $(a^B b)^\omega$

$\supset$ $\subset$

ω-regular expressions
e.g. $(a^* b)^\omega$

PROP: Emptyness of ωBS-regular languages is decidable

PROP: The inclusions in the diamond are strict.

AUTOMATA

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.

AUTOMATA

DEF: One can define ωB, ωS, ωBS-automata.
Essentially: finite state automata with modified accepting condition (more expressive than Buchi).
They come in two variants hierarchical or not.

ωBS-automaton =

finite automaton

\+ finite set of counters, of kind B or S

\+ each counter can be left unchanged or reset or incremented

AUTOMATA

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.

ωBS-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

AUTOMATA

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 ωBS-regular expression
- L is accepted by an ωBS-automaton
- L is accepted by a hierarchical ωBS-automaton

And the same holds for ωB and ωS regular languages.

THE DIAMOND (2:AUTOMATA)

ωBS-regular expressions
hierarchical ωBS-automata
ωBS-automata

$\cup$ $\cup$

ωS-regular expressions
hierarchical ωS-automata
ωS-automata

ωB-regular expressions
hierarchical ωB-automata
ωB-automata

$\cup$ $\cup$

ω-regular expressions
Büchi automata

CLOSURE

THM: The ωBS-regular languages are closed under union, intersection, projection.

Closure

THM: The ωBS-regular languages are closed under union, intersection, projection.
Proof: Union and projection are syntactic on ωBS-regular expressions. Intersection is obtained by product of ωBS-automata.

CLOSURE

THM: The ωBS-regular languages are closed under union, intersection, projection.
Proof: Union and projection are syntactic on ωBS-regular expressions.
Intersection is obtained by product of ωBS-automata.

Example of intersections:

$$((b^*ab^*)^B\sharp)^\omega \;\cap\; ((a^*ba^*)^B\sharp)^\omega = ((a+b)^B\sharp)^\omega$$

CLOSURE

THM: The ωBS-regular languages are closed under union, intersection, projection.
Proof: Union and projection are syntactic on ωBS-regular expressions.
Intersection is obtained by product of ωBS-automata.

Example of intersections:

$$((b^*ab^*)^B \sharp)^\omega \; \cap \; ((a^*ba^*)^B \sharp)^\omega = ((a+b)^B \sharp)^\omega$$

$$((b^*ab^*)^S \sharp)^\omega \; \cap \; ((a^*ba^*)^B \sharp)^\omega = ((a^*ba^*)^B \, a^S \, (a^*ba^*)^B \sharp)^\omega$$

CLOSURE

THM: The ωBS-regular languages are closed under union, intersection, projection.
Proof: Union and projection are syntactic on ωBS-regular expressions. Intersection is obtained by product of ωBS-automata.

Example of intersections:

$$((b^*ab^*)^B\,\sharp)^\omega \;\cap\; ((a^*ba^*)^B\,\sharp)^\omega = ((a+b)^B\,\sharp)^\omega$$

$$((b^*ab^*)^S\,\sharp)^\omega \;\cap\; ((a^*ba^*)^B\,\sharp)^\omega = ((a^*ba^*)^B\,a^S\,(a^*ba^*)^B\,\sharp)^\omega$$

$$((b^*ab^*)^S\,\sharp)^\omega \;\cap\; ((a^*ba^*)^S\,\sharp)^\omega = \left(\left(\begin{array}{cc} & (a+b)^*a^S(a+b)^*b^S(a+b)^* \\ + & (a+b)^*b^S(a+b)^*a^S(a+b)^* \\ + & b^*(a^+b^+)^Sa^* \end{array}\right)\sharp\right)^\omega$$

CLOSURE

THM: The ωBS-regular languages are closed under union, intersection, projection.

PROP: ωBS-regular languages are not closed under complementation.

Closure

THM: The ωBS-regular languages are closed under union, intersection, projection.

PROP: ωBS-regular languages are not closed under complementation.

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

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

E.g: $a^1b \;\; a^1ba^2b \;\; a^1ba^2ba^3b \;\; \cdots \in \complement L$

Closure

THM: The ωBS-regular languages are closed under union, intersection, projection.

PROP: ωBS-regular languages are not closed under complementation.

THM: The complement of an ωB-regular language is ωS-regular
The complement of an ωS-regular language is ωB-regular

Closure

THM: The ωBS-regular languages are closed under union, intersection, projection.

PROP: ωBS-regular languages are not closed under complementation.

THM: The complement of an ωB-regular language is ωS-regular
The complement of an ωS-regular language is ωB-regular

Example:

$$\complement((a^B b)^\omega) = (a+b)^* a^\omega + ((a^* b)^* a^S b)^\omega$$

CLOSURE

THM: The ωBS-regular languages are closed under union, intersection, projection.

PROP: ωBS-regular languages are not closed under complementation.

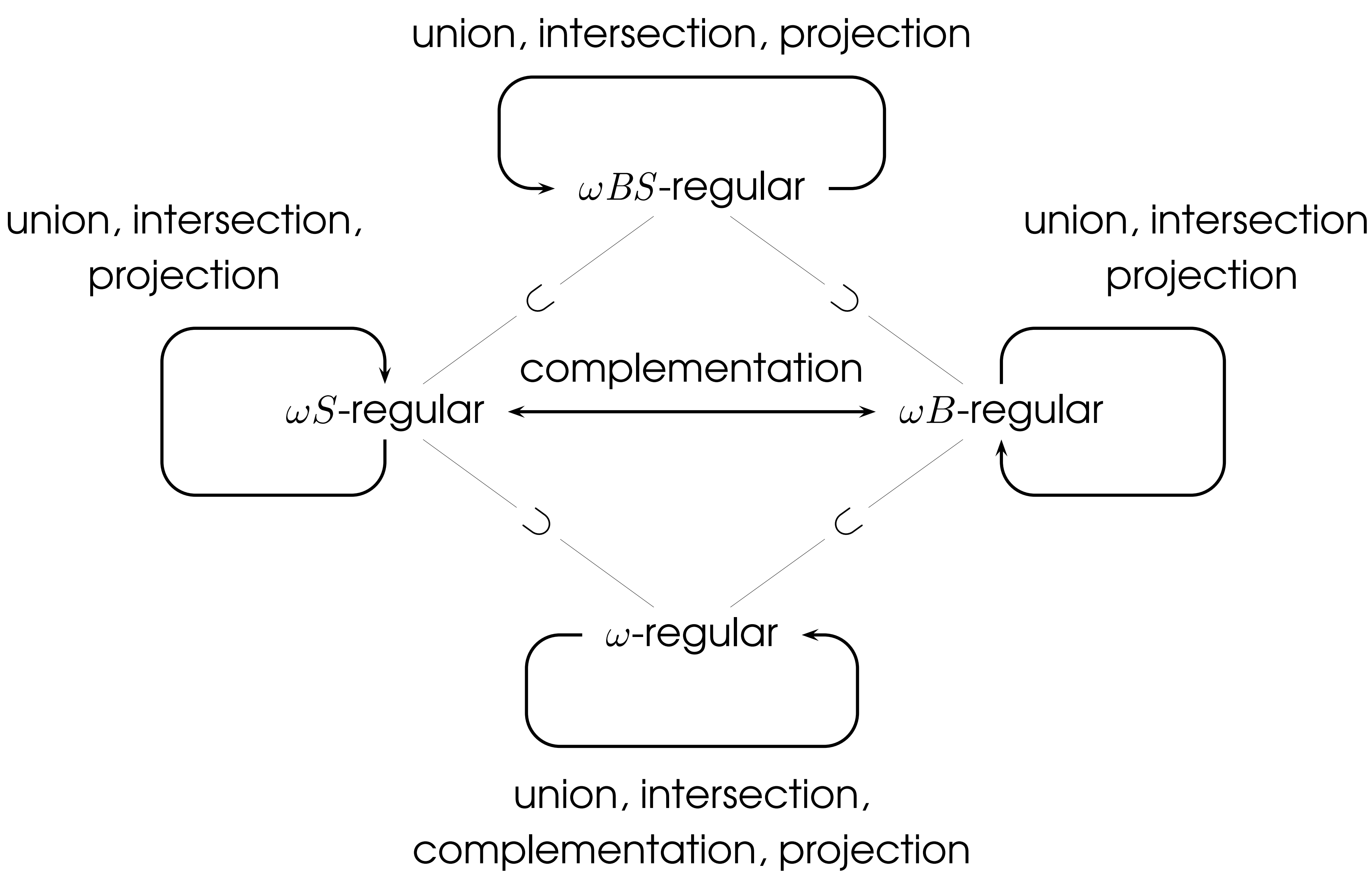
THM: The complement of an ωB-regular language is ωS-regular
The complement of an ωS-regular language is ωB-regular

Example:

$$\complement((a^B b)^\omega) = (a+b)^* a^\omega + ((a^* b)^* a^S b)^\omega$$

$$\complement((a^S b)^\omega) = (a+b)^* a^\omega + ((a^* b)^* a^B b)^\omega$$

THE DIAMOND (3:CLOSURE)



Monadic logic with bounds (MSOLB)

The logic MSOL :

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

MONADIC LOGIC WITH BOUNDS (MSOLB)

The logic **MSOLB** (Bojanczyk05) :

$$\phi = \ \phi \vee \phi \ \mid \phi \wedge \phi \ \mid \neg \phi \ \mid \exists X . \phi \ \mid \exists x . \phi \ \mid x \in X \ \mid x = S(y) \ \mid a(x) \ \ \mid \mathbb{U} X . \phi$$

With $\mathbb{U} X . \phi \quad \equiv \quad \forall n . \ \exists X . \ (|X| > n) \wedge \phi$

"there exists a set as big as I want"

MONADIC LOGIC WITH BOUNDS (MSOLB)

The logic **MSOLB** (Bojanczyk05) :

$$\phi = \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \exists X.\phi \mid \exists x.\phi \mid x \in X \mid x = S(y) \mid a(x) \mid \mathbb{U}X.\phi$$

With $\mathbb{U}X.\phi \quad \equiv \quad \forall n.\ \exists X.\ (|X| > n) \wedge \phi$

"there exists a set as big as I want"

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

"for all sets above a certain size"

MONADIC LOGIC WITH BOUNDS (MSOLB)

The logic **MSOLB** (Bojanczyk05) :

$$\phi = \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \exists X.\phi \mid \exists x.\phi \mid x \in X \mid x = S(y) \mid a(x) \mid \mathbb{U}X.\phi$$

With $\mathbb{U}X.\phi \quad \equiv \quad \forall n.\ \exists X.\ (|X| > n) \wedge \phi$

"there exists a set as big as I want"

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

"for all sets above a certain size"

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

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

Monadic logic with bounds (MSOLB)

The logic **MSOLB** (Bojanczyk05) :

$$\phi = \ \phi \vee \phi \ | \ \phi \wedge \phi \ | \neg \phi \ | \ \exists X. \phi \ | \ \exists x. \phi \ | \ x \in X \ \ | \ x = S(y) \ \ | \ a(x) \ \ \ | \ \mathbb{U} X. \phi$$

With $\mathbb{U}X.\phi \quad \equiv \quad \forall n.\ \exists X.\ (|X| > n) \wedge \phi$

"there exists a set as big as I want"

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

"for all sets above a certain size"

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

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

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

Monadic logic with bounds (MSOLB)

The logic **MSOLB** (Bojanczyk05) :

$$\phi = \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \exists X.\phi \mid \exists x.\phi \mid x \in X \mid x = S(y) \mid a(x) \quad \mid \mathbb{U}X.\phi$$

With $\mathbb{U}X.\phi \quad \equiv \quad \forall n.\ \exists X.\ (|X| > n) \wedge \phi$

"there exists a set as big as I want"

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

"for all sets above a certain size"

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

"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).

Monadic logic with bounds (MSOLB)

The logic **MSOLB** (Bojanczyk05) :

$$\phi = \ \phi \vee \phi \ | \ \phi \wedge \phi \ | \ \neg \phi \ | \ \exists X.\phi \ | \ \exists x.\phi \ | \ x \in X \ | \ x = S(y) \ | \ a(x) \ \ | \ \mathbb{U}X.\phi$$

With $\mathbb{U}X.\phi \quad \equiv \quad \forall n.\ \exists X.\ (|X| > n) \wedge \phi$

"there exists a set as big as I want"

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

"for all sets above a certain size"

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

"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).

QESTION: Is SAT of MSOLB decidable over ω-words?

Monadic logic with bounds (MSOLB)

The logic **MSOLB** (Bojanczyk05) :

$$\phi = \ \phi \vee \phi \mid \phi \wedge \phi \mid \neg \phi \mid \exists X.\phi \mid \exists x.\phi \mid x \in X \mid x = S(y) \mid a(x) \quad \mid \mathbb{U}X.\phi$$

With $\mathbb{U}X.\phi \quad \equiv \quad \forall n.\ \exists X.\ (|X| > n) \wedge \phi$

"there exists a set as big as I want"

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

"for all sets above a certain size"

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

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

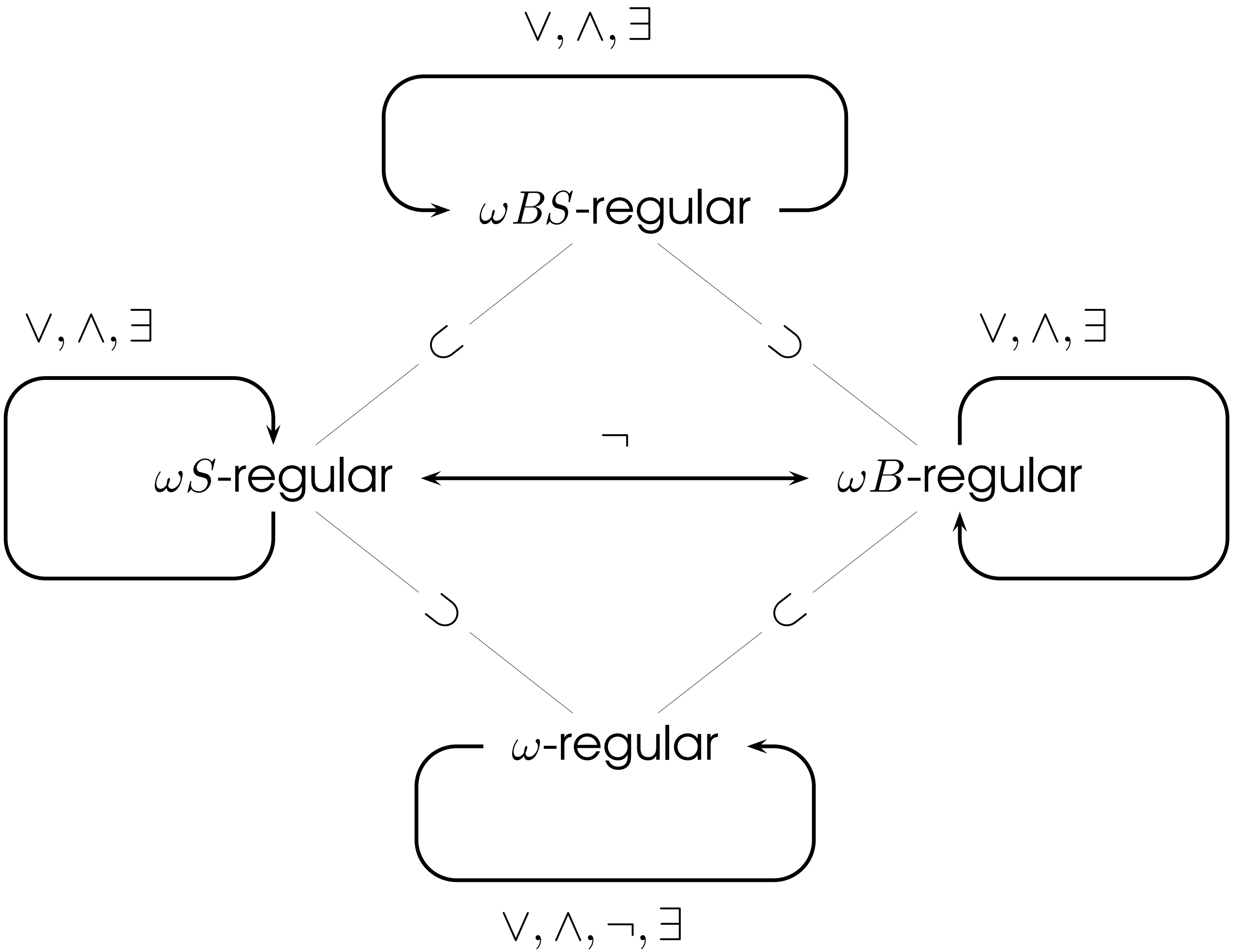
THM(Buchi): SAT of MSOL is decidable over infinite words.

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

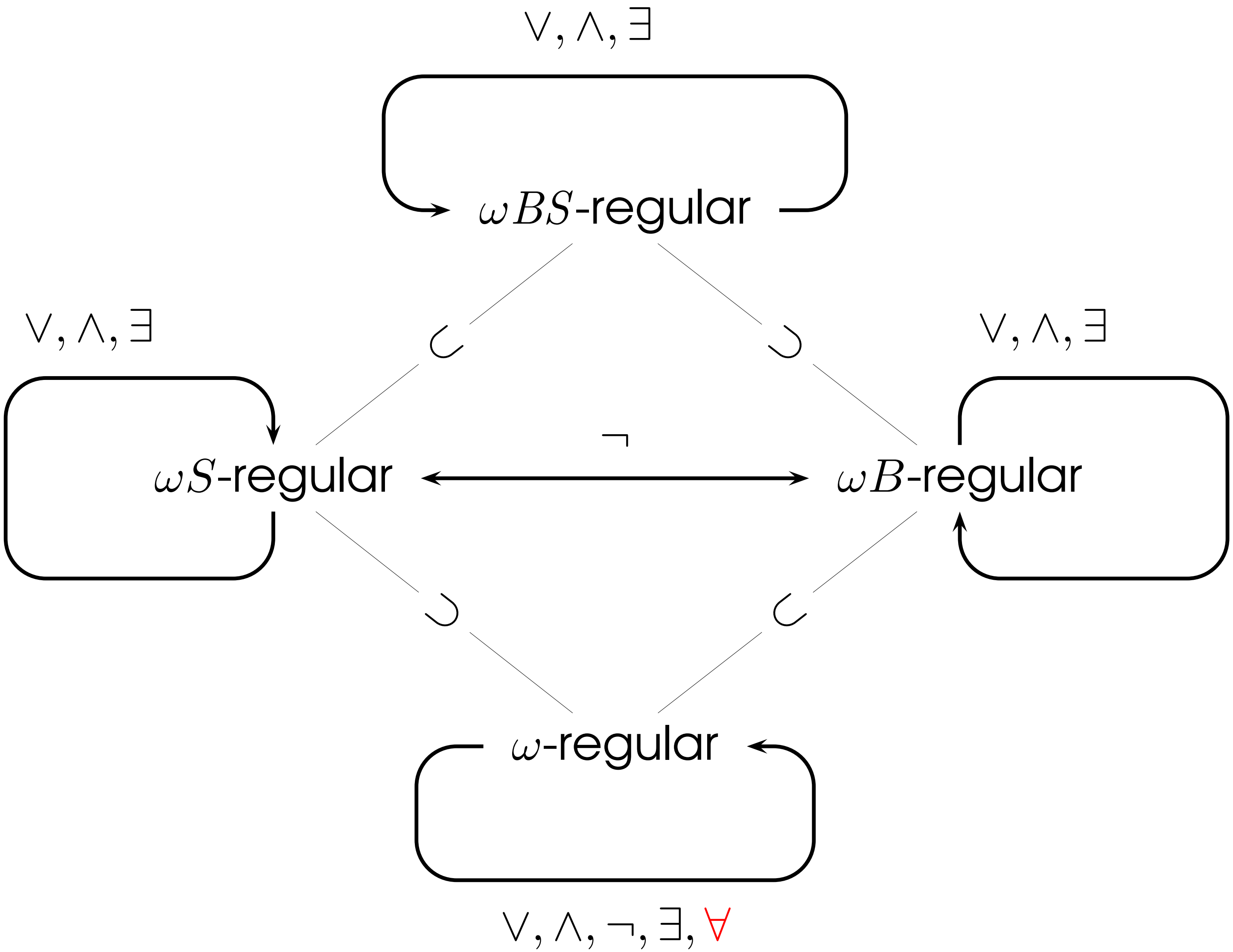
QESTION: Is SAT of MSOLB decidable over ω-words? **The question is open.**

But ωBS-regularity provides a partial answer.

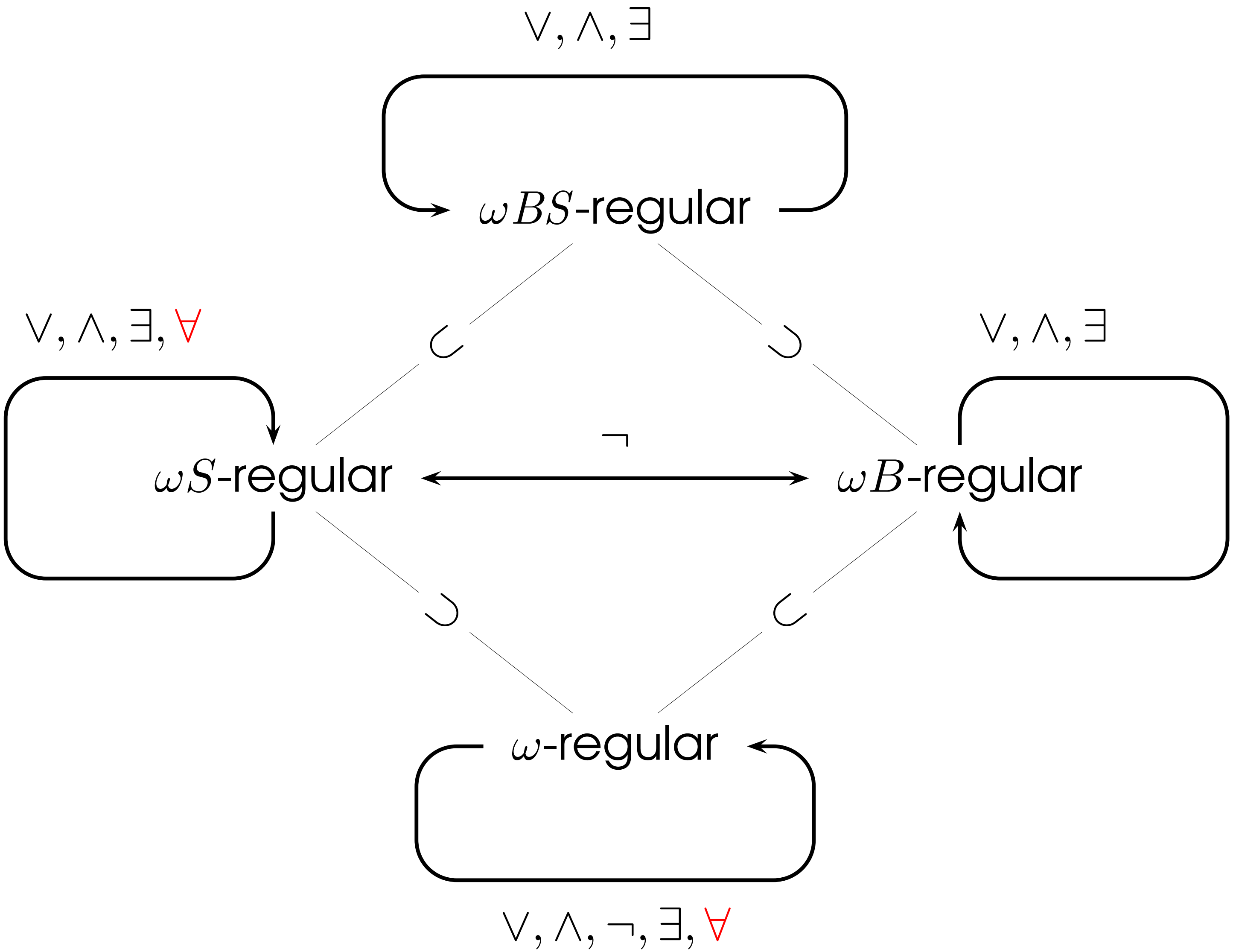
THE DIAMOND (3:LOGIC)



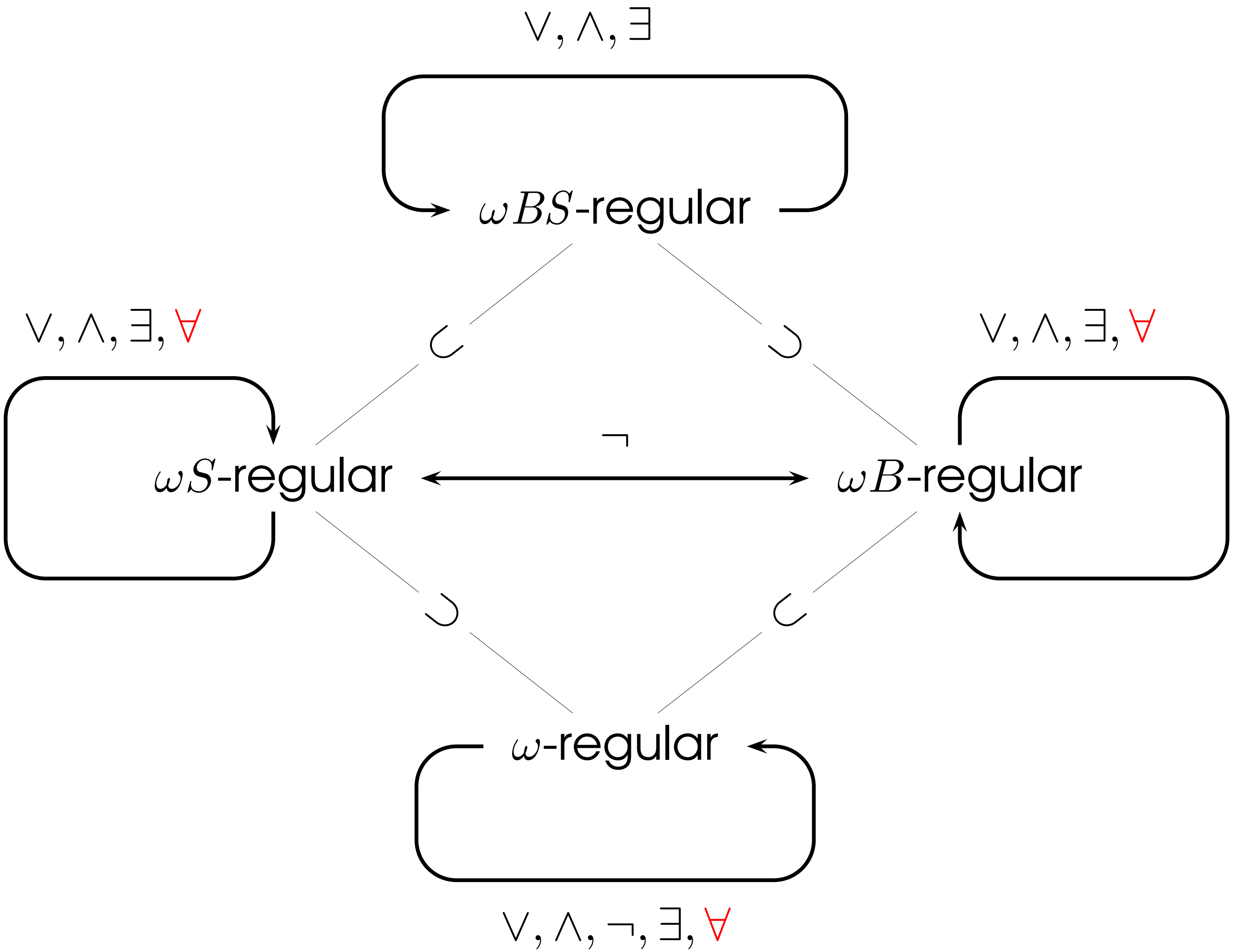
The diamond (3:logic)



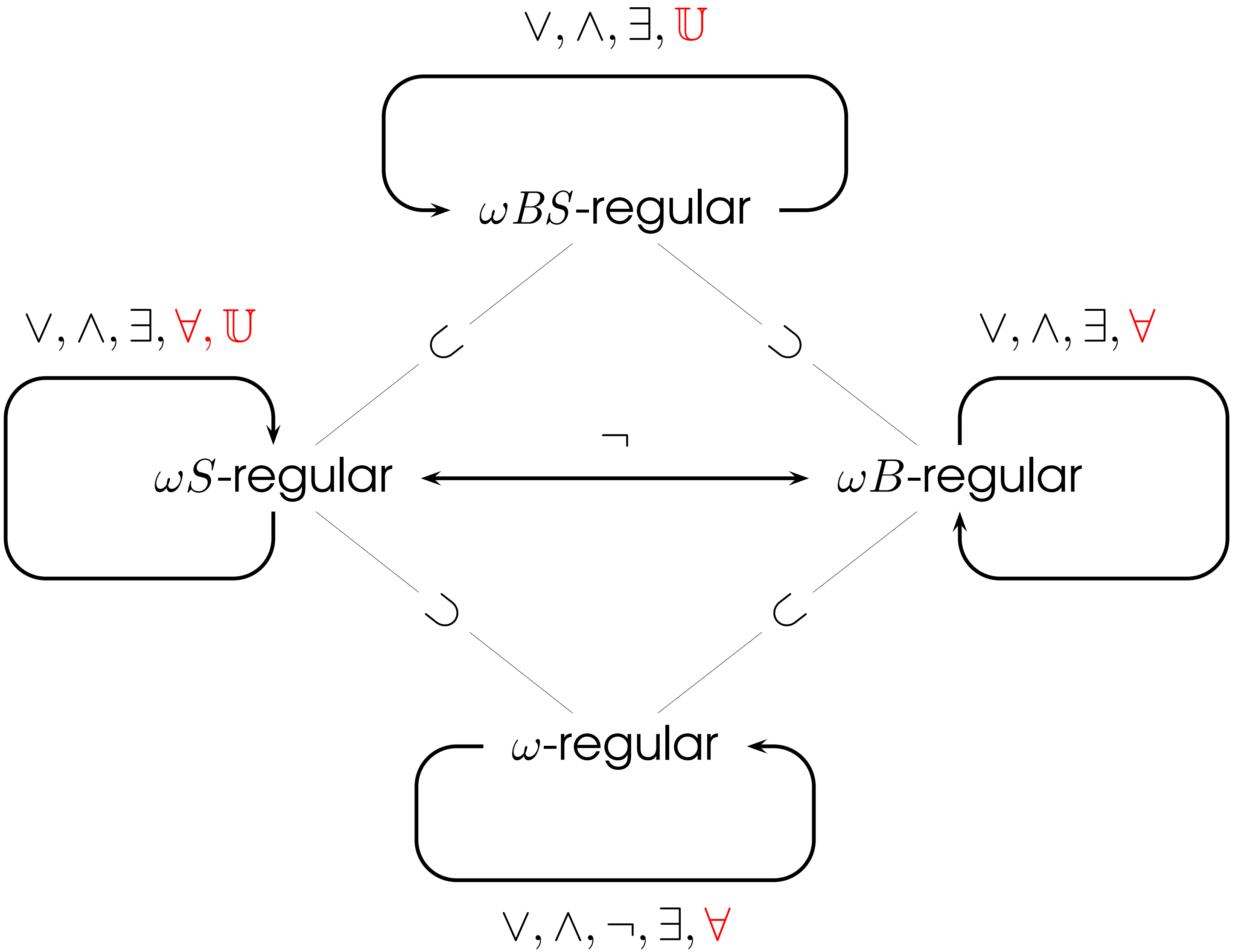
THE DIAMOND (3:LOGIC)



THE DIAMOND (3:LOGIC)

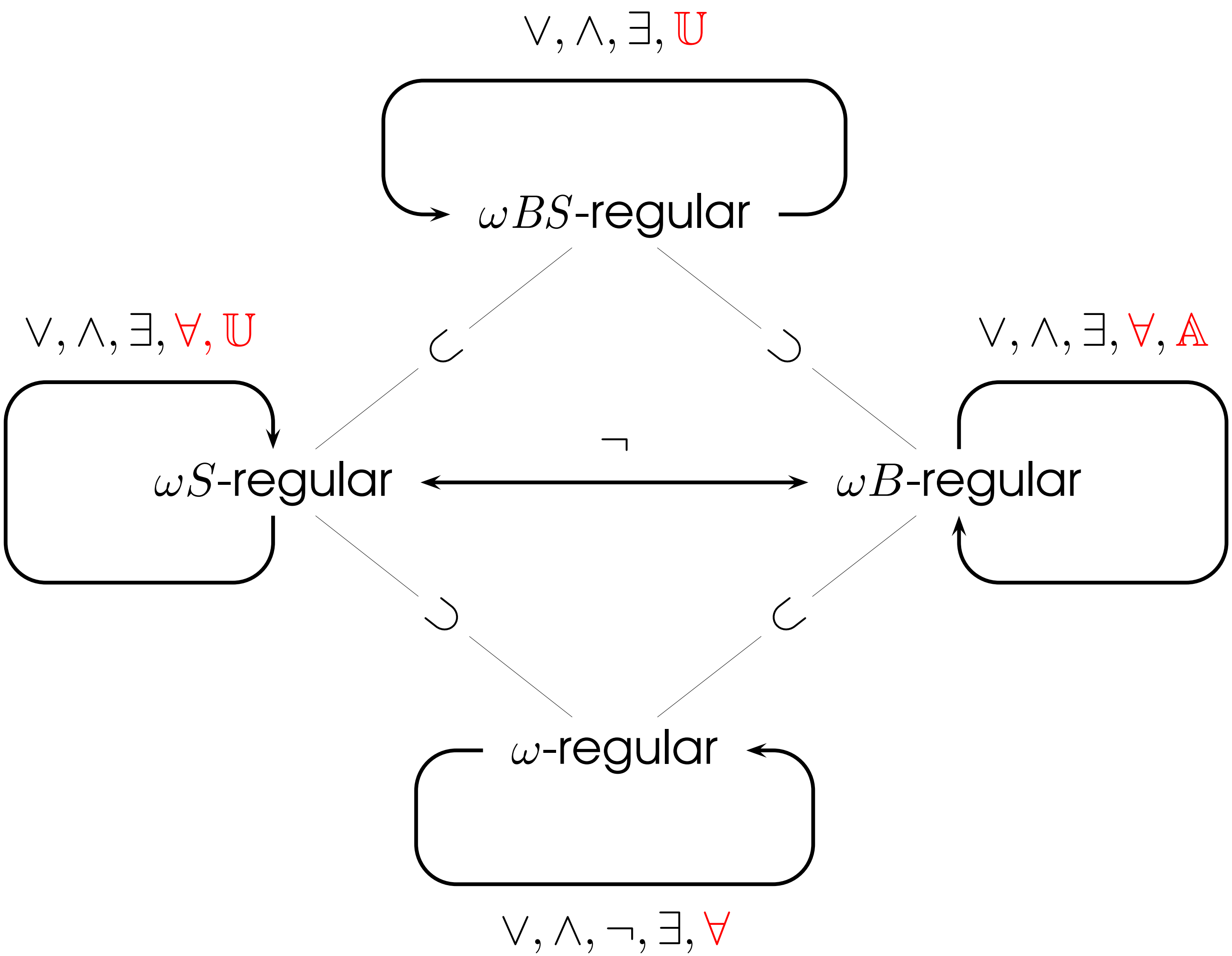


THE DIAMOND (3:LOGIC)



PROP: ωS and ωBS-regular langugues are closed under $\mathbb{U}$.

THE DIAMOND (3:LOGIC)



PROP: ωS and ωBS-regular langugaes are closed under $\mathbb{U}$.

SAT OF MSOLB

DEF: MSOLB+ = MSOLB where $\mathbb{U}$ appears only positively
MSOLB- = MSOLB where $\mathbb{U}$ appears only negatively

SAT of MSOLB

DEF: MSOLB+ = MSOLB where $\mathbb{U}$ appears only positively
MSOLB- = MSOLB where $\mathbb{U}$ appears only negatively

COR: MSOL is equivalent to ω-regular languages
MSOLB+ is equivalent to ωS-regular languages
MSOLB- is equivalent to ωB-regular languages
Boolean comb. of MSOLB+ are contained in ωBS-regular language
$\Rightarrow$ **SAT is decidable for those fragments of MSOLB**

CONCLUSION

We have:

- Introduced an extension of ω-regular expressions.

Conclusion

We have:

- Introduced an extension of ω-regular expressions.
- Introduced corresponding class of automata.

Conclusion

We have:

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

CONCLUSION

We have:

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

Conclusion

We have:

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

Open questions.

- Solve the full logic MSOLB over ω-words.

Conclusion

We have:

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

Open questions.

- Solve the full logic MSOLB over ω-words.
- Find equivalent class of languages over trees.

Conclusion

We have:

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

Open questions.

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

$$\begin{aligned} \phi = \ & \phi \vee \phi \mid \phi \wedge \phi \mid \neg \phi \mid \exists X.\phi \mid \exists x.\phi \mid x \in X \mid x = S(y) \mid a(x) \\ \mid \ & \exists M.\phi \\ \mid \ & |X| < M \text{ (positively below } \exists M) \end{aligned}$$

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.

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.

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.

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 ωB-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 ωB-automata

Limitedness problem: $L(\mathcal{A}) = L(\mathcal{A}[*/B])$?

This is decidable by our results.

Thank you.