

# 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  $\omega$ -word is an infinite word indexed by  $\omega$ :  $a_1 a_2 a_3 \dots$

# OMEGA REGULAR LANGUAGES

---

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

Monadic (second-order) logic:

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

# OMEGA REGULAR LANGUAGES

---

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

Monadic (second-order) logic:

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

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

Monadic (second-order) logic:

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

# OMEGA REGULAR LANGUAGES

---

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

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  $\omega$ -words  
(originally, monadic logic is decidable over  $(\mathbb{N}, +1)$ )

# OMEGA REGULAR LANGUAGES

---

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**Decidability:** Prove decidability of the emptiness 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  $\mathcal{A}_\phi$  such that:

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

**Decidability:** Prove decidability of the emptiness 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  $\mathcal{A}_\phi$  such that:

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

**Decidability:** Prove decidability of the emptiness 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  $\omega$ -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  $\omega$ -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  $\omega$ -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 an  **$\omega$ -regular expression**

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

# A MORE GENERAL VIEW OF THE PICTURE

---

**THM**(Buchi,...) 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 an  **$\omega$ -regular expression**

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

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

# A MORE GENERAL VIEW OF THE PICTURE

---

**THM**(Buchi,...) 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 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).

# A MORE GENERAL VIEW OF THE PICTURE

---

**THM**(Buchi,...) 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 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:

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

$$O = R^\omega \mid R.O \mid O + O$$

# EXTENDING $\omega$ -REGULAR EXPRESSIONS

---

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:

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

## Intuitive meaning:

- without  $B, S$ ,  $\omega BS$ -regular coincides with  $\omega$ -regular

# EXTENDING $\omega$ -REGULAR EXPRESSIONS

---

Syntax of  $\omega BS$ -regular expressions:

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

## 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:

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

## Intuitive meaning:

- without  $B, S$ ,  $\omega BS$ -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

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

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

$$aba^2 ba^3 ba^4 b \dots \notin (a^B b)^\omega$$

# EXTENDING $\omega$ -REGULAR EXPRESSIONS

---

Syntax of  $\omega BS$ -regular expressions:

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

## 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')
- $S$  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

$$ababab \dots \notin (a^S b)^\omega$$

$$aba^2baba^3baba^4b \dots \notin (a^S b)^\omega$$

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

# SEMANTIC OF $\omega BS$ -REGULAR LANGUAGES

---

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

# 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  $(\Sigma^*)^\omega$

- $a \mapsto \{(a, a, a, a, a, \dots)\}$

# 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  $(\Sigma^*)^\omega$

- $a \mapsto \{(a, a, a, a, a, \dots)\}$
- $U.V \mapsto \{(u_1v_1, u_2v_2, \dots) : \vec{u} \in U, \vec{v} \in V\}$

# 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  $(\Sigma^*)^\omega$

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

# SEMANTIC OF $\omega BS$ -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  $(\Sigma^*)^\omega$

- $a \mapsto \{(a, a, a, a, a, \dots)\}$
- $U.V \mapsto \{(u_1v_1, u_2v_2, \dots) : \vec{u} \in U, \vec{v} \in V\}$
- $U + V \mapsto \{\vec{w} : \vec{u}, \vec{v} \in U \cup V, \forall i. w_i \in \{u_i, v_i\}\}$
- $U^* \mapsto \{(u_0u_1 \dots u_{\pi(1)-1}, u_{\pi(1)} \dots u_{\pi(2)} - 1, \dots) : \vec{u} \in U, \pi \text{ nondecreasing}\}$

# 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  $(\Sigma^*)^\omega$

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

# 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  $(\Sigma^*)^\omega$

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

# 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  $(\Sigma^*)^\omega$

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

# OTHER EXAMPLES

---

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

# OTHER EXAMPLES

---

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

The languages with an infinite number of  $b$ 's such that the size of segments of  $a$  is not bounded.

E.g.  $aba^2baba^3baba^4b \dots \in ((a^*b)^*a^Sb)^\omega$

# OTHER EXAMPLES

---

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

The languages with an infinite number of  $b$ 's such that the size of segments of  $a$  is not bounded.

E.g.  $aba^2baba^3baba^4b \dots \in ((a^*b)a^Sb)^\omega$

$$((a^Sb)^S a^S c)^\omega$$

# OTHER EXAMPLES

---

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

The languages with an infinite number of  $b$ 's such that the size of segments of  $a$  is not bounded.

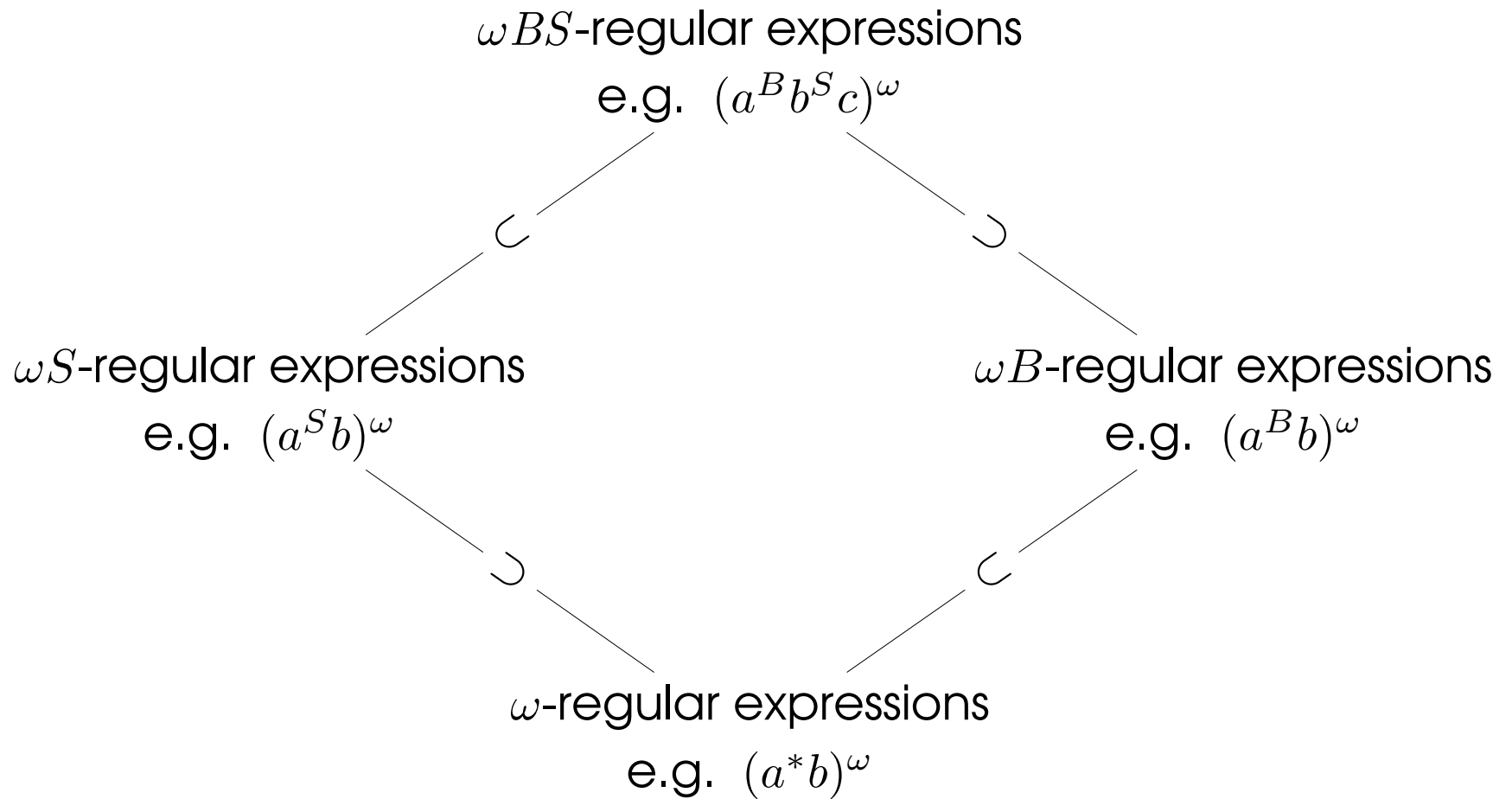
E.g.  $aba^2baba^3baba^4b \dots \in ((a^*b)a^Sb)^\omega$

$$((a^Sb)^S a^S c)^\omega$$

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.

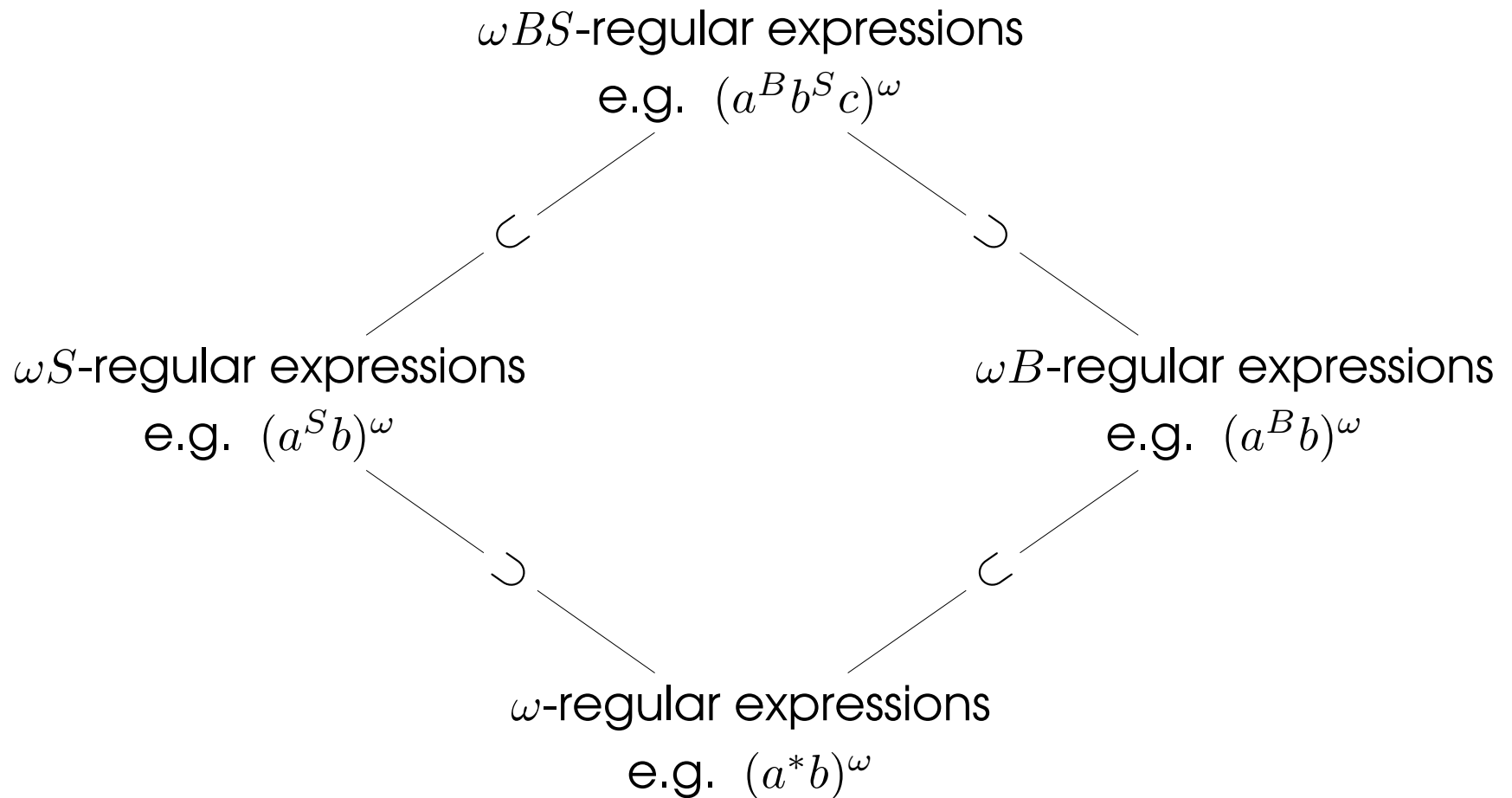
# THE DIAMOND

---



# THE DIAMOND

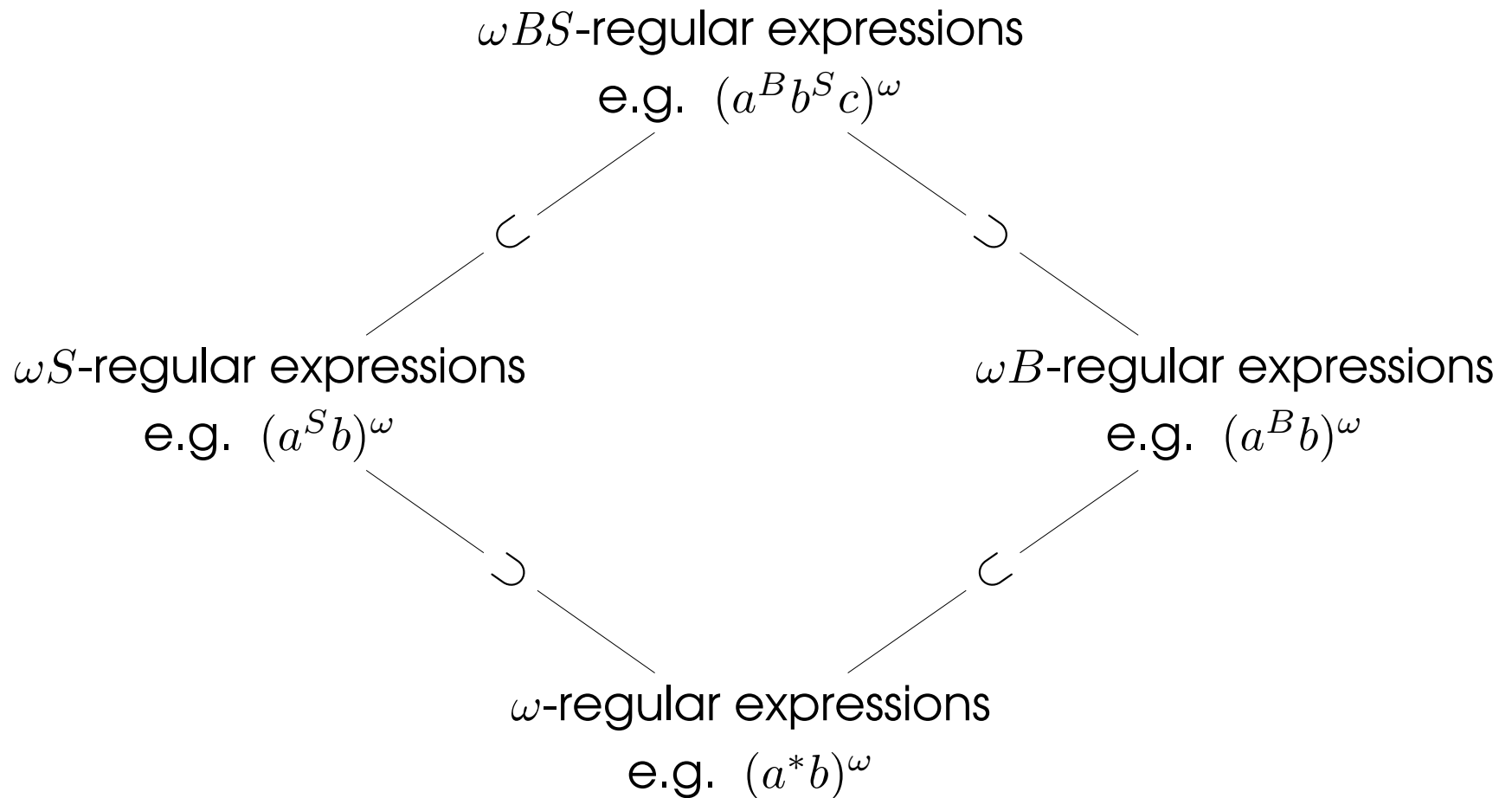
---



**PROP:** Emptiness of  $\omega BS$ -regular languages is decidable

# THE DIAMOND

---



**PROP:** Emptiness of  $\omega BS$ -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  $\omega B$ ,  $\omega S$ ,  $\omega BS$ -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**

# 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.

$\omega BS$ -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

# 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.

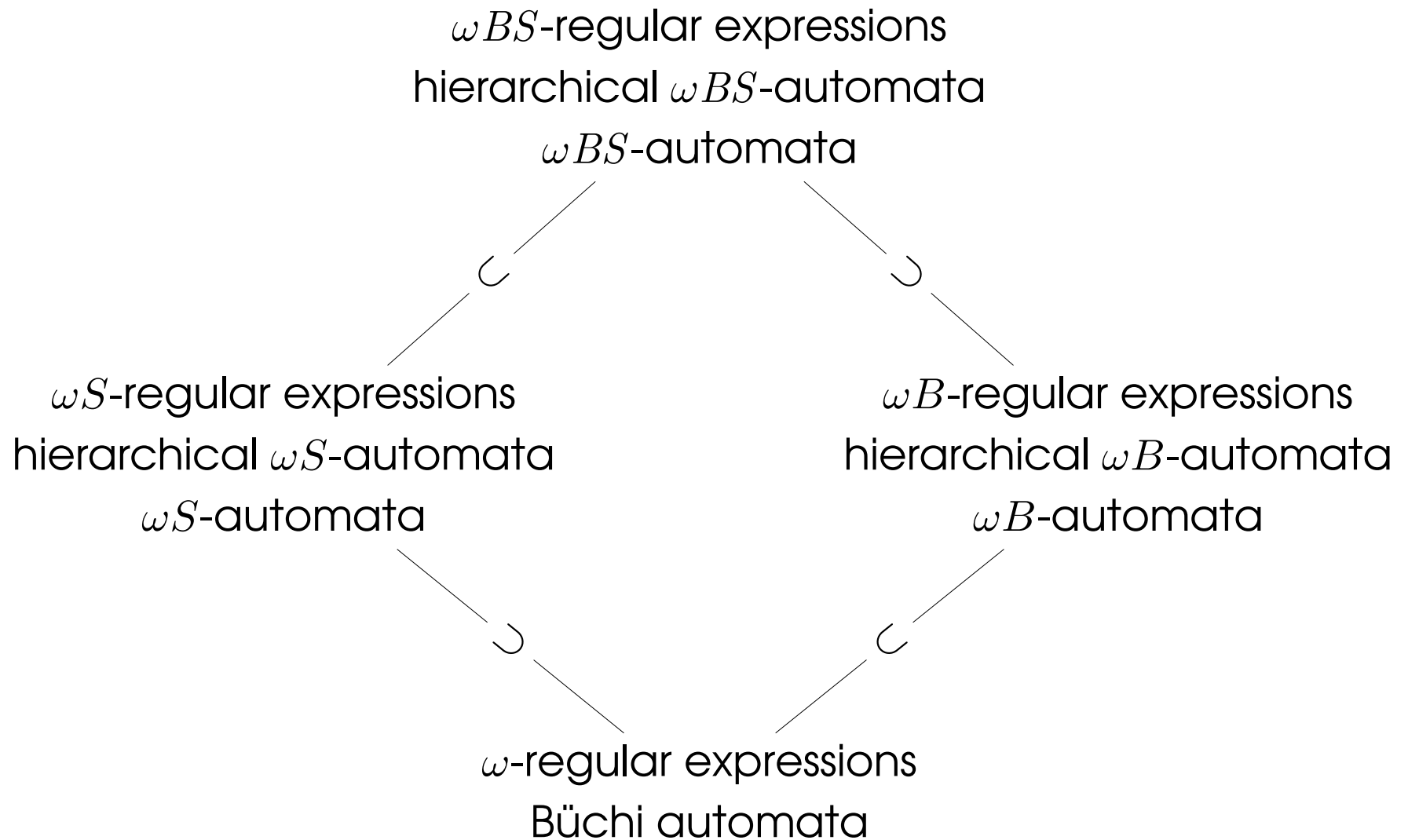
**THM:** The following are equivalent:

- $L$  is evaluation of an  $\omega BS$ -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)

---



# CLOSURE

---

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

# CLOSURE

---

**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.

# CLOSURE

---

**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:**

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

# CLOSURE

---

**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:**

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

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

# CLOSURE

---

**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:

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

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

$$((b^*ab^*)^S \#)^\omega \cap ((a^*ba^*)^S \#)^\omega = \left( \left( \begin{array}{c} (a+b)^*a^S(a+b)^*b^S(a+b)^* \\ + \\ (a+b)^*b^S(a+b)^*a^S(a+b)^* \\ + \\ b^*(a^+b^+)^S a^* \end{array} \right) \# \right)^\omega$$

# CLOSURE

---

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

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

# CLOSURE

---

**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  $\mathbb{C}L$  contains  $a^{f(1)}ba^{f(2)}b \dots$  iff there exists infinitely many values appearing infinitely often in  $f$ .

**E.g:**  $a^1b \ a^1ba^2b \ a^1ba^2ba^3b \ \dots \in \mathbb{C}L$

# 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

# 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

**Example:**

$$\mathbb{C}((a^B b)^\omega) = (a + b)^* a^\omega + ((a^* b)^* a^S b)^\omega$$

# CLOSURE

---

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

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

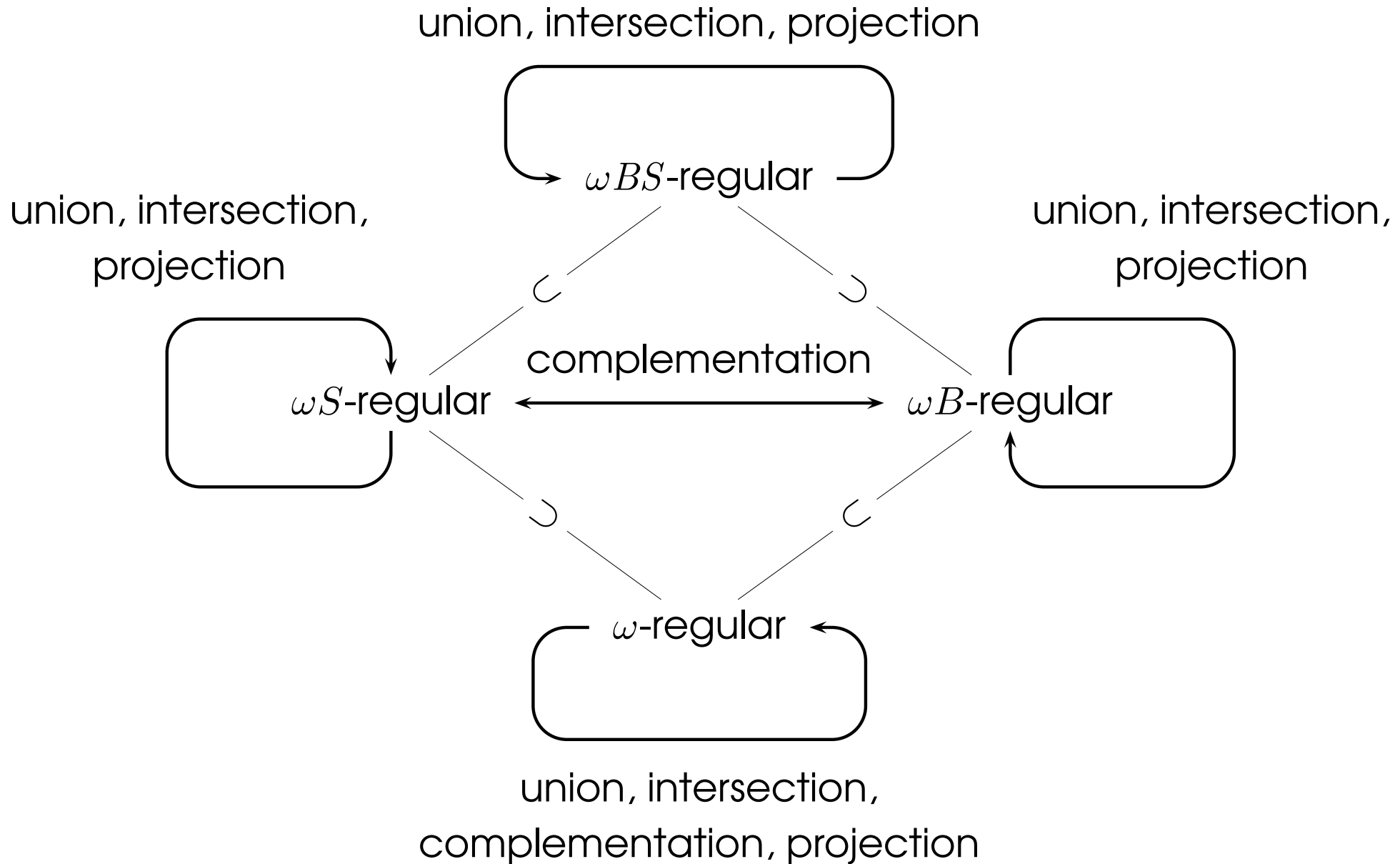
**THM:** The complement of an  $\omega B$ -regular language is  $\omega S$ -regular  
The complement of an  $\omega S$ -regular language is  $\omega B$ -regular

**Example:**

$$\mathbb{C}((a^B b)^\omega) = (a + b)^* a^\omega + ((a^* b)^* a^S b)^\omega$$

$$\mathbb{C}((a^S b)^\omega) = (a + b)^* a^\omega + ((a^* b)^* a^B b)^\omega$$

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

“there exists a set as big as I want”

# MONADIC LOGIC WITH BOUNDS (MSOLB)

---

The logic **MSOLB** (Bojanczyk05) :

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

With  $\mathbf{UX}.\phi \equiv \forall n. \exists X. (|X| > n) \wedge \phi$

“there exists a set as big as I want”

And also  $\mathbf{AX}.\phi \equiv \neg \mathbf{UX}.\neg \phi \equiv \exists n. \forall X. (|X| > n) \rightarrow \phi$

“for all sets above a certain size”

# MONADIC LOGIC WITH BOUNDS (MSOLB)

---

The logic **MSOLB** (Bojanczyk05) :

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

With  $\mathbf{UX}.\phi \equiv \forall n. \exists X. (|X| > n) \wedge \phi$

“there exists a set as big as I want”

And also  $\mathbf{AX}.\phi \equiv \neg\mathbf{UX}.\neg\phi \equiv \exists n. \forall X. (|X| > n) \rightarrow \phi$

“for all sets above a certain size”

And also  $\mathbf{BX}.\phi \equiv \neg\mathbf{UX}.\phi \equiv \exists n. \forall X. \phi \rightarrow (|X| < n)$

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

# MONADIC LOGIC WITH BOUNDS (MSOLB)

---

The logic **MSOLB** (Bojanczyk05) :

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

With  $\mathbf{UX. \phi} \equiv \forall n. \exists X. (|X| > n) \wedge \phi$

“there exists a set as big as I want”

And also  $\mathbf{AX. \phi} \equiv \neg \mathbf{UX. \neg \phi} \equiv \exists n. \forall X. (|X| > n) \rightarrow \phi$

“for all sets above a certain size”

And also  $\mathbf{BX. \phi} \equiv \neg \mathbf{UX. \phi} \equiv \exists n. \forall X. \phi \rightarrow (|X| < n)$

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

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

# MONADIC LOGIC WITH BOUNDS (MSOLB)

---

The logic **MSOLB** (Bojanczyk05) :

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

With  $\mathbf{UX}.\phi \equiv \forall n. \exists X. (|X| > n) \wedge \phi$

“there exists a set as big as I want”

And also  $\mathbf{AX}.\phi \equiv \neg \mathbf{UX}.\neg \phi \equiv \exists n. \forall X. (|X| > n) \rightarrow \phi$

“for all sets above a certain size”

And also  $\mathbf{BX}.\phi \equiv \neg \mathbf{UX}.\phi \equiv \exists n. \forall X. \phi \rightarrow (|X| < n)$

“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).

# MONADIC LOGIC WITH BOUNDS (MSOLB)

---

The logic **MSOLB** (Bojanczyk05) :

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

With  $\mathbf{UX}.\phi \equiv \forall n. \exists X. (|X| > n) \wedge \phi$

“there exists a set as big as I want”

And also  $\mathbf{AX}.\phi \equiv \neg\mathbf{UX}.\neg\phi \equiv \exists n. \forall X. (|X| > n) \rightarrow \phi$

“for all sets above a certain size”

And also  $\mathbf{BX}.\phi \equiv \neg\mathbf{UX}.\phi \equiv \exists n. \forall X. \phi \rightarrow (|X| < n)$

“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?

# MONADIC LOGIC WITH BOUNDS (MSOLB)

---

The logic **MSOLB** (Bojanczyk05) :

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

With  $\mathbf{UX}.\phi \equiv \forall n. \exists X. (|X| > n) \wedge \phi$

“there exists a set as big as I want”

And also  $\mathbf{AX}.\phi \equiv \neg\mathbf{UX}.\neg\phi \equiv \exists n. \forall X. (|X| > n) \rightarrow \phi$

“for all sets above a certain size”

And also  $\mathbf{BX}.\phi \equiv \neg\mathbf{UX}.\phi \equiv \exists n. \forall X. \phi \rightarrow (|X| < n)$

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

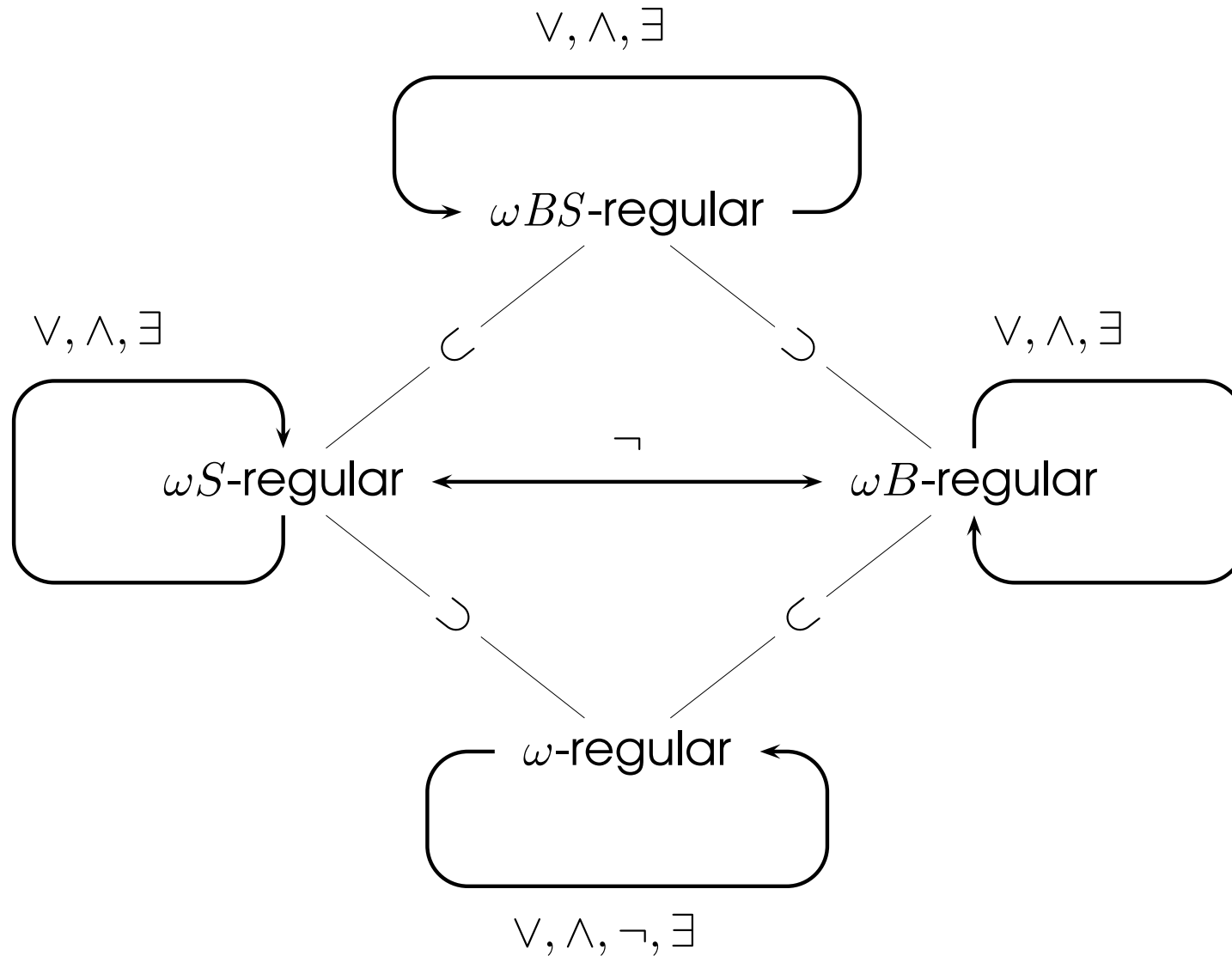
**THM(Buchi):** SAT of MSOL is decidable over infinite words.

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

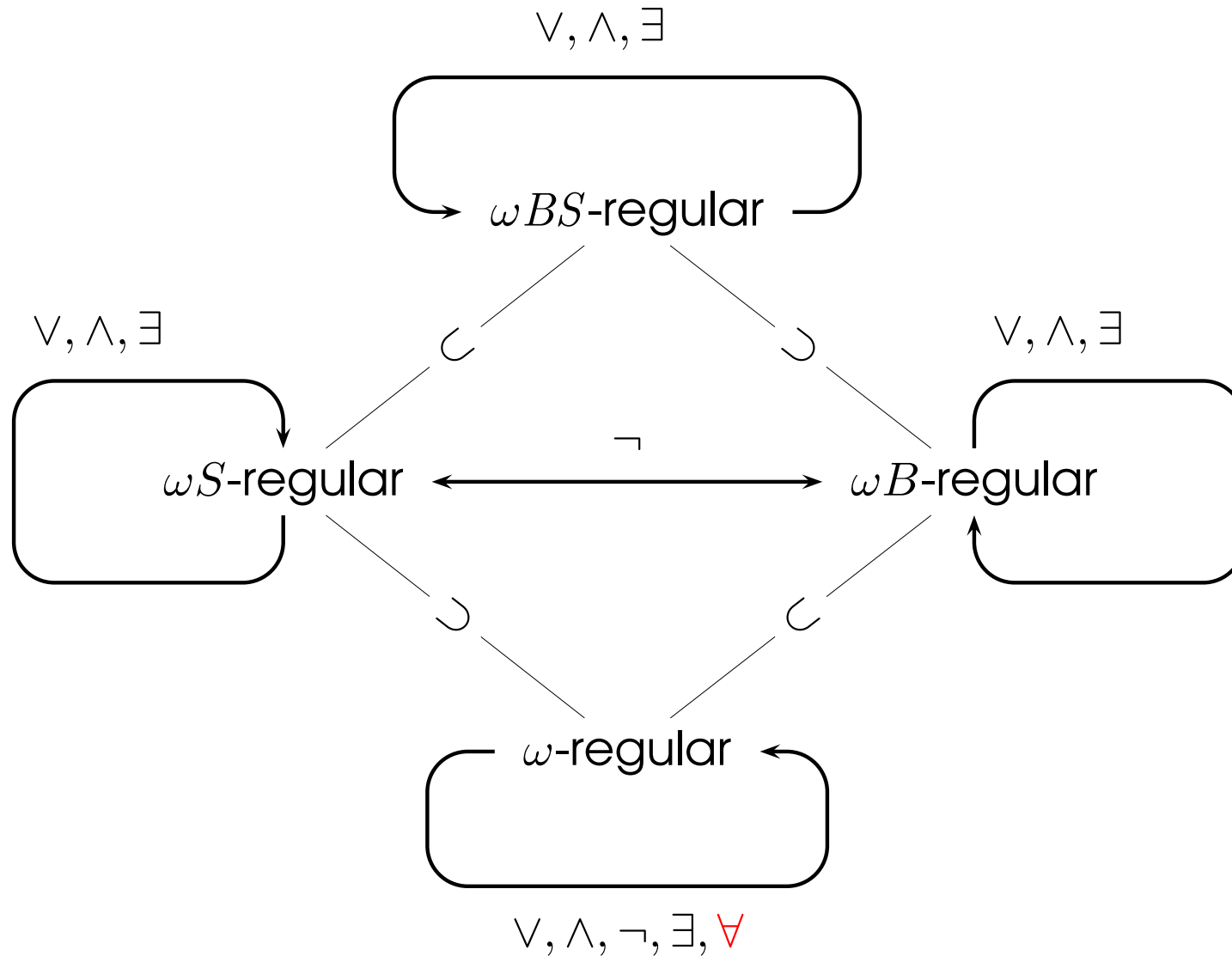
**QUESTION:** Is SAT of MSOLB decidable over  $\omega$ -words? **The question is open.**

**But  $\omega BS$ -regularity provides a partial answer.**

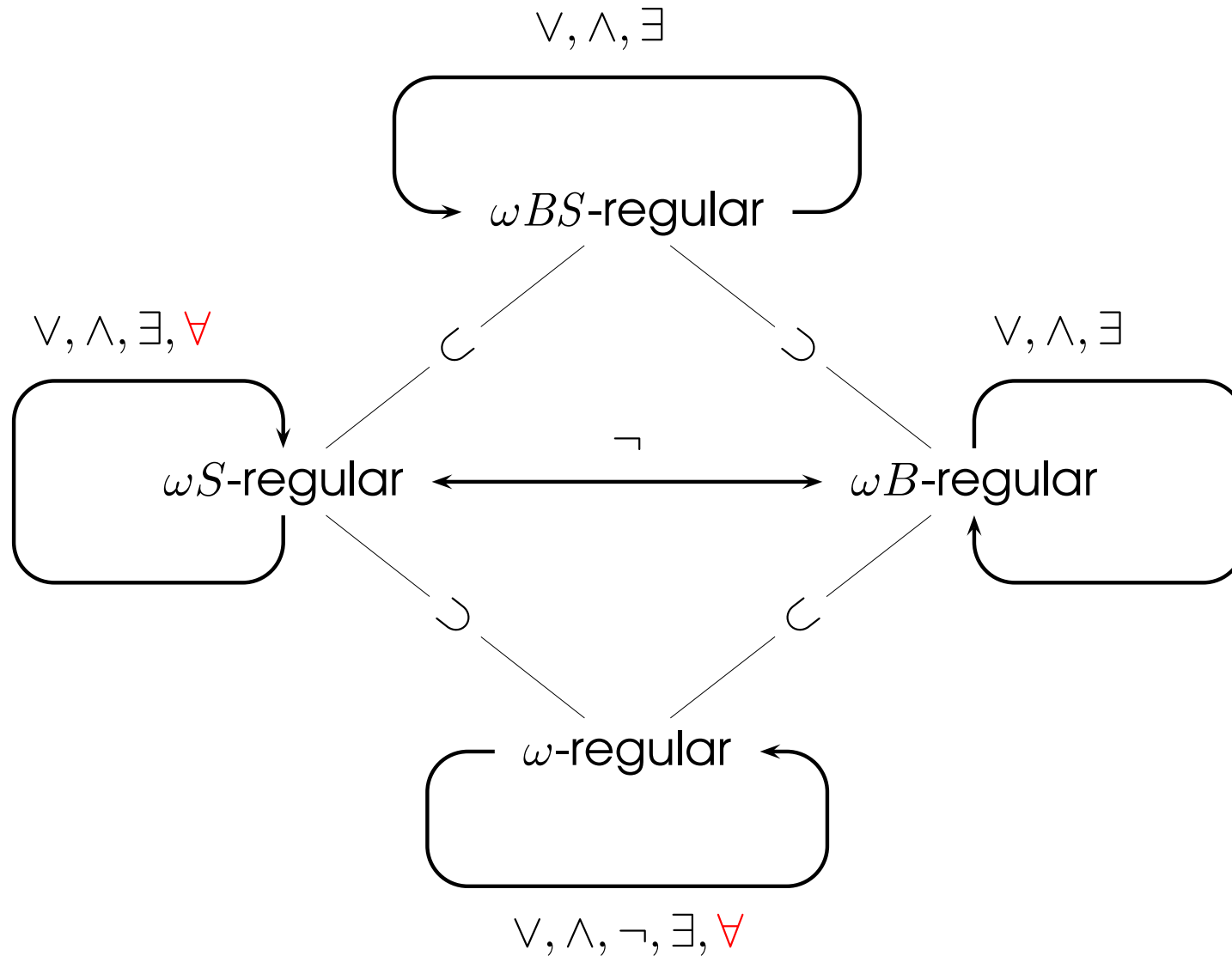
# THE DIAMOND (3:LOGIC)



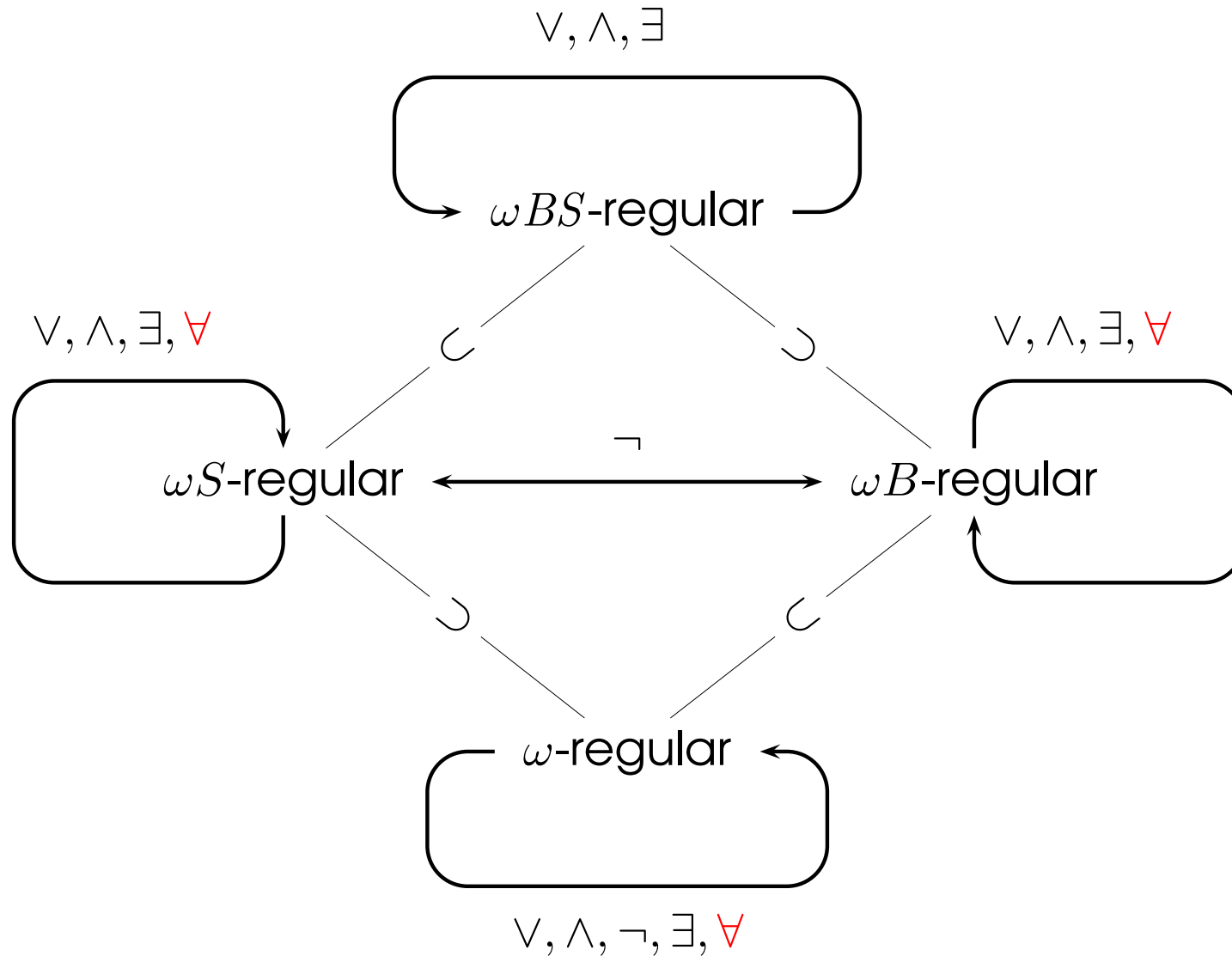
# THE DIAMOND (3:LOGIC)



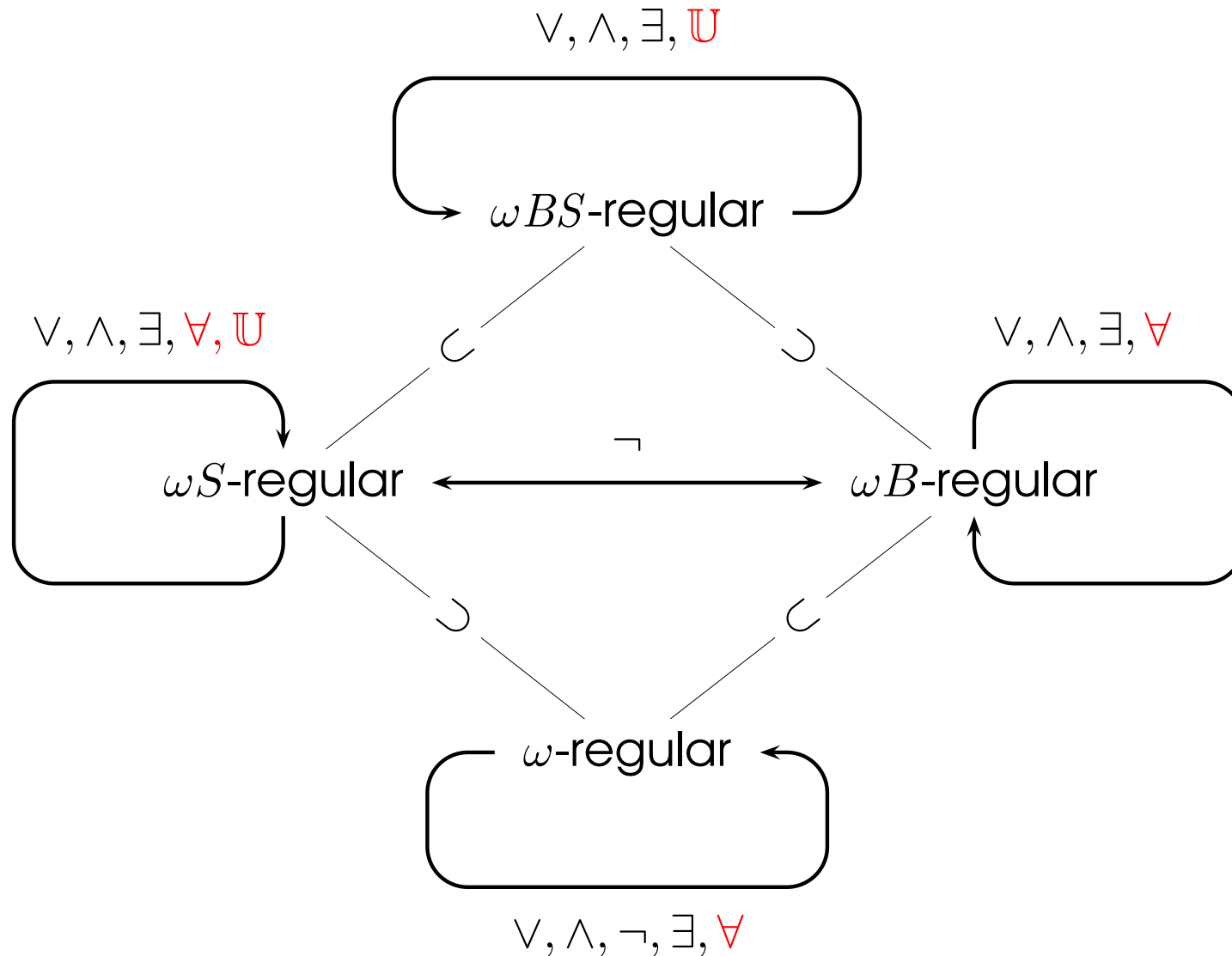
# THE DIAMOND (3:LOGIC)



# THE DIAMOND (3:LOGIC)

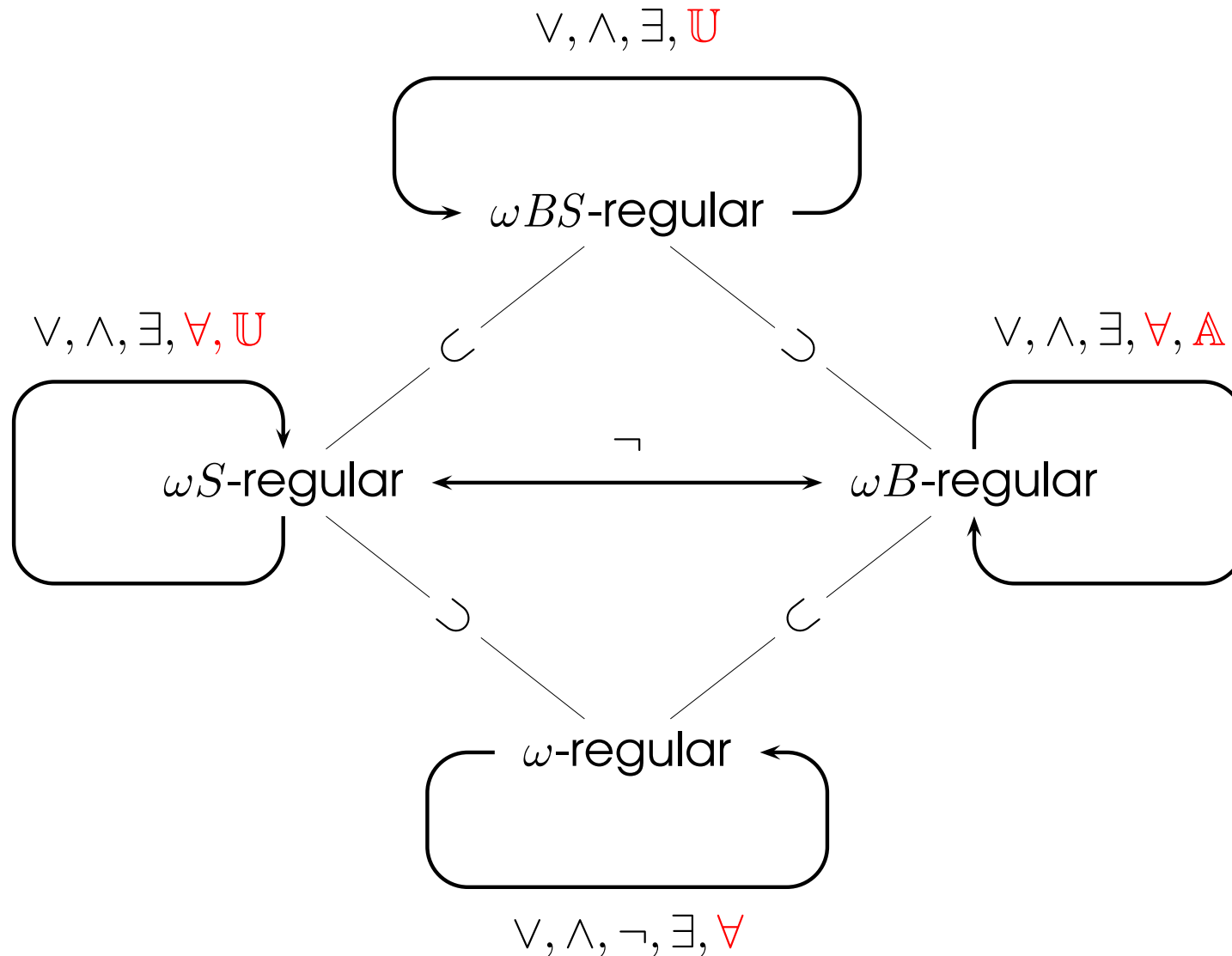


# THE DIAMOND (3:LOGIC)



**PROP:**  $\omega S$  and  $\omega BS$ -regular languages are closed under  $\cup$ .

# THE DIAMOND (3:LOGIC)



**PROP:**  $\omega S$  and  $\omega BS$ -regular languages are closed under  $\cup$ .

# SAT OF MSOLB

---

**DEF:** MSOLB+ = MSOLB where  $\cup$  appears only **positively**  
MSOLB- = MSOLB where  $\cup$  appears only **negatively**

# SAT OF MSOLB

---

**DEF:** MSOLB+ = MSOLB where  $\cup$  appears only **positively**  
MSOLB- = MSOLB where  $\cup$  appears only **negatively**

**COR:** MSOL is equivalent to  $\omega$ -regular languages  
MSOLB+ is equivalent to  $\omega S$ -regular languages  
MSOLB- is equivalent to  $\omega B$ -regular languages  
Boolean comb. of MSOLB+ are contained in  $\omega BS$ -regular language  
 $\Rightarrow$  **SAT is decidable for those fragments of MSOLB**

# CONCLUSION

---

We have:

- Introduced an extension of  $\omega$ -regular expressions.

# CONCLUSION

---

We have:

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

# CONCLUSION

---

We have:

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

# 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.

# 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**?:

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

# 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  $\omega B$ -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:**  $L(\mathcal{A}) = L(\mathcal{A}[* / B])$ ?

This is decidable by our results.

---

**Thank you.**