The Algebraic Counterpart of the Wagner Hierarchy

Jérémie Cabessa

University of Lausanne

November 29, 2007

Outline

Outline

Outline

Outline

Outline

Outline

Classifying ω-rational languages

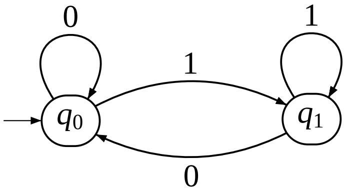
Automata
The Wagner Hierarchy
[Wag79]

Descriptive set theory
The Wadge hierarchy
of ω-rational languages
[Sel98]

Algebra
[CarPer96,99], [DupRis06]
The FSG-hierarchy

Muller automaton (deterministic)

$\mathcal{A} = (Q, A, \delta, \{q_0\}, \mathcal{T})$, with $\mathcal{T} \subseteq \mathcal{P}^Q$.



If $\mathcal{T} = \{\{q_0\}, \{q_1\}\}$, then $L^\omega(\mathcal{A}) = A^*(0^\omega \cup 1^\omega)$.

Theorem

The following conditions are equivalent:

1. L *is* ω*-rational*
2. L *is recognizable by a finite Muller automaton.*

The Wagner Hierarchy

(classification of ω-rational languages)

Given two ω-rational languages K and L, we set

$$
\begin{aligned}
K \leq_W L \quad &\text{iff} \quad K = f^{-1}(L) \text{ for some continuous function } f \\
&\text{iff} \quad w \in K \Leftrightarrow f(w) \in L. \\
K <_W L \quad &\text{iff} \quad K \leq_W L \not\leq_W K \\
K \equiv_W L \quad &\text{iff} \quad K \leq_W L \leq_W K.
\end{aligned}
$$

Definition (Wagner hierarchy)

The collection of all ω-rational languages ordered by the relation $\leq_W$ is called *the Wagner hierarchy*.

The Wagner Hierarchy
(classification of ω-rational languages)

Given two ω-rational languages K and L, we set

$$K \leq_W L \quad \text{iff} \quad K = f^{-1}(L) \text{ for some continuous function } f$$
$$\text{iff} \quad w \in K \Leftrightarrow f(w) \in L.$$

$$K <_W L \quad \text{iff} \quad K \leq_W L \not\leq_W K$$
$$K \equiv_W L \quad \text{iff} \quad K \leq_W L \leq_W K.$$

Definition (Wagner hierarchy)

The collection of all ω-rational languages ordered by the relation $\leq_W$ is called *the Wagner hierarchy*.

The Wagner Hierarchy
(classification of ω-rational languages)

Given two ω-rational languages K and L, we set

$$K \leq_W L \quad \text{iff} \quad K = f^{-1}(L) \text{ for some continuous function } f$$
$$\text{iff} \quad w \in K \Leftrightarrow f(w) \in L.$$
$$K <_W L \quad \text{iff} \quad K \leq_W L \not\leq_W K$$
$$K \equiv_W L \quad \text{iff} \quad K \leq_W L \leq_W K.$$

Definition (Wagner hierarchy)

The collection of all ω-rational languages ordered by the relation $\leq_W$ is called *the Wagner hierarchy*.

The Wagner Hierarchy
(classification of ω-rational languages)

Given two ω-rational languages K and L, we set

$$
\begin{aligned}
K \leq_W L \quad &\text{iff} \quad K = f^{-1}(L) \text{ for some continuous function } f \\
&\text{iff} \quad w \in K \Leftrightarrow f(w) \in L. \\
K <_W L \quad &\text{iff} \quad K \leq_W L \not\leq_W K \\
K \equiv_W L \quad &\text{iff} \quad K \leq_W L \leq_W K.
\end{aligned}
$$

Definition (Wagner hierarchy)

The collection of all ω-rational languages ordered by the relation $\leq_W$ is called *the Wagner hierarchy*.

The Wagner Hierarchy
(classification of ω-rational languages)

Given two ω-rational languages K and L, we set

$$\begin{array}{lll} K \leq_W L & \text{iff} & K = f^{-1}(L) \text{ for some continuous function } f \\ & \text{iff} & w \in K \Leftrightarrow f(w) \in L. \\ K <_W L & \text{iff} & K \leq_W L \not\leq_W K \\ K \equiv_W L & \text{iff} & K \leq_W L \leq_W K. \end{array}$$

Definition (Wagner hierarchy)

The collection of all ω-rational languages ordered by the relation $\leq_W$ is called *the Wagner hierarchy*.

Theorem

The Wagner hierarchy has height ω^ω and it is decidable.

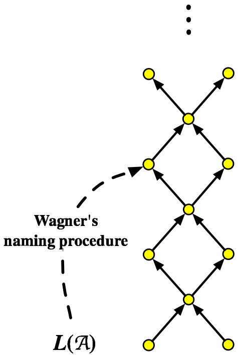


Illustration of Wagner's decidability procedure

Classifying ω-rational language is equivalent to classifying their underlying Muller automata (Kleene).

$$\mathcal{T} = \{\{q_0\}, \{q_2, q_3\}, \{q_4\}, \{q_6, q_7\}, \{q_8\}, \{q_{10}, q_{11}\}, \{q_{12}\}\}$$

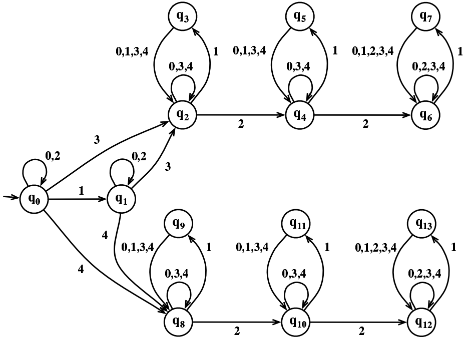
Illustration of Wagner's decidability procedure

Classifying ω-rational language is equivalent to classifying their underlying Muller automata (Kleene).

$$\mathcal{T} = \{\{q_0\}, \{q_2, q_3\}, \{q_4\}, \{q_6, q_7\}, \{q_8\}, \{q_{10}, q_{11}\}, \{q_{12}\}\}$$

Illustration of Wagner's decidability procedure

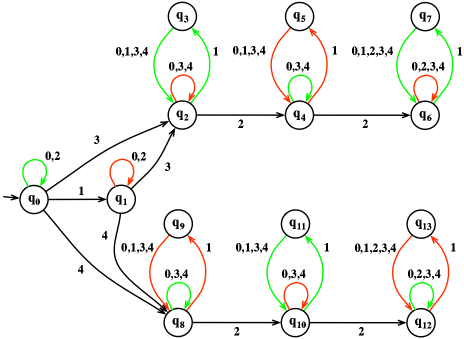
Classifying ω-rational language is equivalent to classifying their underlying Muller automata (Kleene).



$$\mathcal{T} = \{\{q_0\}, \{q_2, q_3\}, \{q_4\}, \{q_6, q_7\}, \{q_8\}, \{q_{10}, q_{11}\}, \{q_{12}\}\}$$

Illustration of Wagner's decidability procedure

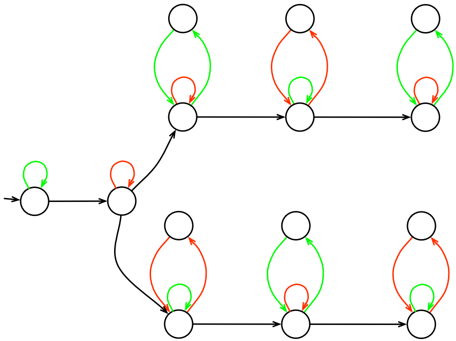
Classifying ω-rational language is equivalent to classifying their underlying Muller automata (Kleene).



$$\mathcal{T} = \{\{q_0\}, \{q_2, q_3\}, \{q_4\}, \{q_6, q_7\}, \{q_8\}, \{q_{10}, q_{11}\}, \{q_{12}\}\}$$

Illustration of Wagner's decidability procedure

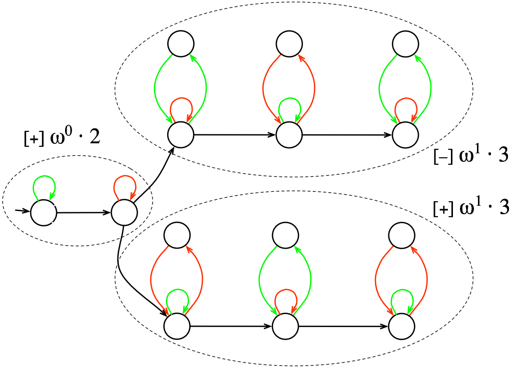
Classifying ω-rational language is equivalent to classifying their underlying Muller automata (Kleene).



$$\mathcal{T} = \{\{q_0\}, \{q_2, q_3\}, \{q_4\}, \{q_6, q_7\}, \{q_8\}, \{q_{10}, q_{11}\}, \{q_{12}\}\}$$

Illustration of Wagner's decidability procedure

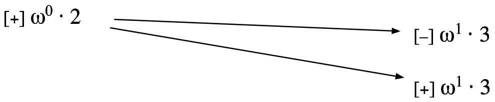
Classifying ω-rational language is equivalent to classifying their underlying Muller automata (Kleene).



$$\mathcal{T} = \{\{q_0\}, \{q_2, q_3\}, \{q_4\}, \{q_6, q_7\}, \{q_8\}, \{q_{10}, q_{11}\}, \{q_{12}\}\}$$

Illustration of Wagner's decidability procedure

Classifying ω-rational language is equivalent to classifying their underlying Muller automata (Kleene).



$$\mathcal{T} = \{\{q_0\}, \{q_2, q_3\}, \{q_4\}, \{q_6, q_7\}, \{q_8\}, \{q_{10}, q_{11}\}, \{q_{12}\}\}$$

Illustration of Wagner's decidability procedure

Classifying ω-rational language is equivalent to classifying their underlying Muller automata (Kleene).

$$\begin{tikzcd}[column sep=huge] {[+]\ \omega^0 \cdot 2} \arrow[r] \arrow[dr] & {[-]\ \omega^1 \cdot 3} \\ & {[+]\ \omega^1 \cdot 3} \end{tikzcd}$$

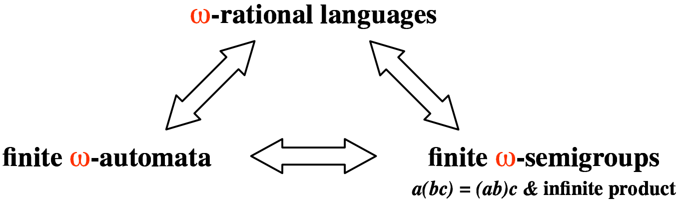
$$d_W(\mathcal{A}) = d_W(L(\mathcal{A})) = \omega \cdot 3 + 2$$

and $L(\mathcal{A})$ is non-self-dual

$$\mathcal{T} = \{\{q_0\}, \{q_2, q_3\}, \{q_4\}, \{q_6, q_7\}, \{q_8\}, \{q_{10}, q_{11}\}, \{q_{12}\}\}$$

$$
\begin{tikzcd}
 & \textbf{rational languages} \arrow[dl, leftrightarrow] \arrow[dr, leftrightarrow] & \\
\textbf{finite automata} \arrow[rr, leftrightarrow] & & \textbf{finite semigroups}
\end{tikzcd}
$$

$$a(bc) = (ab)c$$


ω-rational languages
finite ω-automata
finite ω-semigroups
$a(bc) = (ab)c$ & infinite product

An ω-semigroup is a semigroup equipped with a suitable infinite product.

Definition (ω-semigroup)

An ω-semigroup S is a pair (S_+, S_ω), where S_+ is a semigroup, S_ω is a set, and equipped with

- an associative mixed product : $S_+ \times S_\omega \longrightarrow S_\omega$
- a surjective infinite product $\pi_S : S_+^\omega \longrightarrow S_\omega$ satisfying

$$\pi_S(s_0 s_1 \cdots s_{k_1 - 1}, s_{k_1} \cdots s_{k_2 - 1}, \ldots) = \pi_S(s_0, s_1, s_2, \ldots),$$

$$s\pi_S(s_0, s_1, s_2, \ldots) = \pi_S(s, s_0, s_1, s_2, \ldots).$$

An ω-semigroup is a semigroup equipped with a suitable infinite product.

Definition (ω-semigroup)

An ω-semigroup S is a pair (S_+, S_ω), where S_+ is a semigroup, S_ω is a set, and equipped with

- an associative mixed product : $S_+ \times S_\omega \longrightarrow S_\omega$
- a surjective infinite product $\pi_S : S_+^\omega \longrightarrow S_\omega$ satisfying

$$\pi_S(s_0 s_1 \cdots s_{k_1 - 1}, s_{k_1} \cdots s_{k_2 - 1}, \ldots) = \pi_S(s_0, s_1, s_2, \ldots),$$

$$s\pi_S(s_0, s_1, s_2, \ldots) = \pi_S(s, s_0, s_1, s_2, \ldots).$$

An ω-semigroup is a semigroup equipped with a suitable infinite product.

Definition (ω-semigroup)

An ω-semigroup S is a pair (S_+, S_ω), where S_+ is a semigroup, S_ω is a set, and equipped with

- an associative mixed product : $S_+ \times S_\omega \longrightarrow S_\omega$
- a surjective infinite product $\pi_S : S_+^\omega \longrightarrow S_\omega$ satisfying

$$\pi_S(s_0 s_1 \cdots s_{k_1 - 1}, s_{k_1} \cdots s_{k_2 - 1}, \ldots) = \pi_S(s_0, s_1, s_2, \ldots),$$

$$s\pi_S(s_0, s_1, s_2, \ldots) = \pi_S(s, s_0, s_1, s_2, \ldots).$$

An ω-semigroup is a semigroup equipped with a suitable infinite product.

Definition (ω-semigroup)

An ω-semigroup S is a pair (S_+, S_ω), where S_+ is a semigroup, S_ω is a set, and equipped with

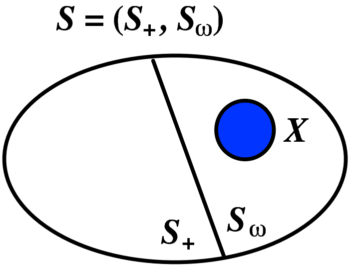
- an associative mixed product : $S_+ \times S_\omega \longrightarrow S_\omega$
- a surjective infinite product $\pi_S : S_+^\omega \longrightarrow S_\omega$ satisfying

$$\pi_S(s_0 s_1 \cdots s_{k_1-1}, s_{k_1} \cdots s_{k_2-1}, \ldots) = \pi_S(s_0, s_1, s_2, \ldots),$$
$$s\pi_S(s_0, s_1, s_2, \ldots) = \pi_S(s, s_0, s_1, s_2, \ldots).$$

Notation: $\pi_S(x, x, x, x, \ldots)$ will be written x^ω.

A *pointed ω-semigroup* is a pair (S, X), where

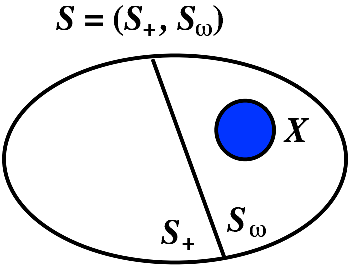
- $S = (S_+, S_\omega)$ is an ω-semigroup,
- $X \subseteq S_\omega$.



Notation: $\pi_S(x, x, x, x, \ldots)$ will be written x^ω.

A *pointed ω-semigroup* is a pair (S, X), where

- $S = (S_+, S_\omega)$ is an ω-semigroup,
- $X \subseteq S_\omega$.



Finite pointed ω-semigroups are the algebraic counterparts of Büchi automata.

Theorem

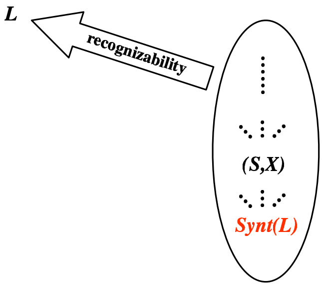
An ω-language is recognizable by a finite pointed ω-semigroup iff it is recognizable by a finite Büchi automaton (iff it is ω-rational).

Finite pointed ω-semigroups are the algebraic counterparts of Büchi automata.

Theorem

An ω-language is recognizable by a finite pointed ω-semigroup iff it is recognizable by a finite Büchi automaton (iff it is ω-rational).

Among all finite pointed ω-semigroups recognizing a given ω-rational language L, there exists a minimal one, **the syntactic ω-semigroup** of L, denoted by $\boldsymbol{Synt(L)}$.



Example

Consider the language $K = ((a+b)^*a)^\omega$. Then $Synt(K) = (S, X)$, where

- $S = (\{0,1\}, \{0^\omega, 1^\omega\})$ defined by the relations

$$0 \cdot 0 = 0 \qquad 0 \cdot 1 = 0 \qquad 1 \cdot 0 = 0 \qquad 1 \cdot 1 = 1$$
$$00^\omega = 0^\omega \qquad 10^\omega = 0^\omega \qquad 01^\omega = 1^\omega \qquad 11^\omega = 1^\omega$$

- $X = \{0^\omega\}$.

Example

Consider the language $K = ((a+b)^* a)^\omega$. Then $Synt(K) = (S, X)$, where

- $S = (\{0,1\}, \{0^\omega, 1^\omega\})$ defined by the relations

$$0 \cdot 0 = 0 \quad 0 \cdot 1 = 0 \quad 1 \cdot 0 = 0 \quad 1 \cdot 1 = 1$$

$$00^\omega = 0^\omega \quad 10^\omega = 0^\omega \quad 01^\omega = 1^\omega \quad 11^\omega = 1^\omega$$

- $X = \{0^\omega\}$.

Example

Consider the language $L = (a\{b,c\}^* \cup \{b\})^\omega$. Then $Synt(L) = (T, Y)$, where

- $T = (\{a, b, c, ca\}, \{a^\omega, (ca)^\omega, 0\})$ is defined by

$$\begin{array}{cccc}
a^2 = a & ab = a & ac = a & ba = a \\
b^2 = b & bc = c & cb = c & c^2 = c \\
b^\omega = a^\omega & c^\omega = 0 & aa^\omega = a^\omega & a(ca)^\omega = a^\omega \\
ba^\omega = a^\omega & b(ca)^\omega = (ca)^\omega & ca^\omega = (ca)^\omega & c(ca)^\omega = (ca)^\omega
\end{array}$$

- $Y = \{a^\omega\}$.

Example

Consider the language $L = (a\{b, c\}^* \cup \{b\})^\omega$. Then $Synt(L) = (T, Y)$, where

- $T = (\{a, b, c, ca\}, \{a^\omega, (ca)^\omega, 0\})$ is defined by

$$
\begin{array}{cccc}
a^2 = a & ab = a & ac = a & ba = a \\
b^2 = b & bc = c & cb = c & c^2 = c \\
b^\omega = a^\omega & c^\omega = 0 & aa^\omega = a^\omega & a(ca)^\omega = a^\omega \\
ba^\omega = a^\omega & b(ca)^\omega = (ca)^\omega & ca^\omega = (ca)^\omega & c(ca)^\omega = (ca)^\omega
\end{array}
$$

- $Y = \{a^\omega\}$.

We aim to classify finite pointed ω-semigroups. We adopt a hierarchical game approach.

1. **We transpose Wadge games on ω-semigroups...**
2. We define the corresponding reduction relation.
3. We then describe the resulting hierarchy.

We aim to classify finite pointed ω-semigroups. We adopt a hierarchical game approach.

1. **We transpose Wadge games on ω-semigroups...**
2. We define the corresponding reduction relation.
3. We then describe the resulting hierarchy.

We aim to classify finite pointed ω-semigroups. We adopt a hierarchical game approach.

1. **We transpose Wadge games on ω-semigroups...**
2. We define the corresponding reduction relation.
3. We then describe the resulting hierarchy.

We aim to classify finite pointed ω-semigroups. We adopt a hierarchical game approach.

1. **We transpose Wadge games on ω-semigroups...**
2. We define the corresponding reduction relation.
3. We then describe the resulting hierarchy.

Let $S = (S_+, S_\omega)$ and $T = (T_+, T_\omega)$ be two ω-semigroups, and let also $X \subseteq S_\omega$ and $Y \subseteq T_\omega$.

The infinite two-player game $\mathbb{SG}(\boldsymbol{X}, \boldsymbol{Y})$ is defined as follows:

- Player I plays elements from S_+,
- Player II plays elements from T_+,
- players I and II play alternately, Player I begins,
- Player II may skip her turn, Player I is not allowed,

Let $S = (S_+, S_\omega)$ and $T = (T_+, T_\omega)$ be two ω-semigroups, and let also $X \subseteq S_\omega$ and $Y \subseteq T_\omega$.

The infinite two-player game $\mathbb{SG}(\boldsymbol{X}, \boldsymbol{Y})$ is defined as follows:

- Player I plays elements from S_+,
- Player II plays elements from T_+,
- players I and II play alternately, Player I begins,
- Player II may skip her turn, Player I is not allowed,

Let $S = (S_+, S_\omega)$ and $T = (T_+, T_\omega)$ be two ω-semigroups, and let also $X \subseteq S_\omega$ and $Y \subseteq T_\omega$.

The infinite two-player game $\mathbb{SG}(\boldsymbol{X}, \boldsymbol{Y})$ is defined as follows:

- Player I plays elements from S_+,
- Player II plays elements from T_+,
- players I and II play alternately, Player I begins,
- Player II may skip her turn, Player I is not allowed,

Let $S = (S_+, S_\omega)$ and $T = (T_+, T_\omega)$ be two ω-semigroups, and let also $X \subseteq S_\omega$ and $Y \subseteq T_\omega$.

The infinite two-player game $\mathbb{SG}(\boldsymbol{X}, \boldsymbol{Y})$ is defined as follows:

- Player I plays elements from S_+,
- Player II plays elements from T_+,
- players I and II play alternately, Player I begins,
- Player II may skip her turn, Player I is not allowed,

Let $S = (S_+, S_\omega)$ and $T = (T_+, T_\omega)$ be two ω-semigroups, and let also $X \subseteq S_\omega$ and $Y \subseteq T_\omega$.

The infinite two-player game $\mathbb{SG}(\boldsymbol{X}, \boldsymbol{Y})$ is defined as follows:

- Player I plays elements from S_+,
- Player II plays elements from T_+,
- players I and II play alternately, Player I begins,
- Player II may skip her turn, Player I is not allowed,

$$
\begin{array}{lcccccccc}
\mathrm{I}: & s_0 & & & s_1 & & \cdots\cdots & \xrightarrow{\text{after } \omega \text{ moves}} & (s_0, s_1, s_2, \cdots) \\
 & & \searrow & \nearrow & & \searrow & & & \\
\mathrm{II}: & & - & & & t_0 & \cdots\cdots & \xrightarrow{\text{after } \omega \text{ moves}} & (t_0, t_1, t_2, \cdots)
\end{array}
$$

- Player II wins the game $\mathbb{SG}(X, Y)$ iff

$$\pi_S(s_0, s_1, \ldots) \in X \Leftrightarrow \pi_T(t_0, t_1, \ldots) \in Y.$$

Definition ($\mathbb{SG}$-reduction)

We write $X \leq_{SG} Y$ iff Player II has a winning strategy in $\mathbb{SG}(X, Y)$. And as usual $X \equiv_{SG} Y$ iff $X \leq_{SG} Y \leq_{SG} X$.

$$\begin{array}{lccccccccc}
\mathrm{I}: & s_0 & & & & s_1 & & \cdots\cdots & \xrightarrow{\text{after } \omega \text{ moves}} & (s_0, s_1, s_2, \cdots) \\
& & \searrow & & \nearrow & & \searrow & & & \\
\mathrm{II}: & & & - & & & t_0 & \cdots\cdots & \xrightarrow{\text{after } \omega \text{ moves}} & (t_0, t_1, t_2, \cdots)
\end{array}$$

- Player II wins the game $\mathbb{SG}(X, Y)$ iff

$$\pi_S(s_0, s_1, \ldots) \in X \Leftrightarrow \pi_T(t_0, t_1, \ldots) \in Y.$$

Definition ($\mathbb{SG}$-reduction)

We write $X \leq_{SG} Y$ iff Player II has a winning strategy in $\mathbb{SG}(X, Y)$. And as usual $X \equiv_{SG} Y$ iff $X \leq_{SG} Y \leq_{SG} X$.

$$
\begin{array}{lccccccccc}
\mathrm{I}: & s_0 & & & & s_1 & & & \cdots\cdots & \xrightarrow{\text{after } \omega \text{ moves}} & (s_0, s_1, s_2, \cdots) \\
 & & \searrow & & \nearrow & & \searrow & & & & \\
\mathrm{II}: & & & - & & & & t_0 & \cdots\cdots & \xrightarrow{\text{after } \omega \text{ moves}} & (t_0, t_1, t_2, \cdots)
\end{array}
$$

- Player II wins the game $\mathbb{SG}(X, Y)$ iff

$$\pi_S(s_0, s_1, \ldots) \in X \Leftrightarrow \pi_T(t_0, t_1, \ldots) \in Y.$$

Definition ($\mathbb{SG}$-reduction)

We write $X \leq_{SG} Y$ iff Player II has a winning strategy in $\mathbb{SG}(X, Y)$. And as usual $X \equiv_{SG} Y$ iff $X \leq_{SG} Y \leq_{SG} X$.

$$
\begin{array}{llllllll}
\mathrm{I}: & s_0 & & & s_1 & & \cdots\cdots & \xrightarrow{\text{after } \omega \text{ moves}} \; (s_0, s_1, s_2, \cdots) \\
 & & \searrow & \nearrow & & \searrow & & \\
\mathrm{II}: & & & - & & t_0 & \cdots\cdots & \xrightarrow{\text{after } \omega \text{ moves}} \; (t_0, t_1, t_2, \cdots)
\end{array}
$$

- Player II wins the game $\mathbb{SG}(X, Y)$ iff

$$\pi_S(s_0, s_1, \ldots) \in X \Leftrightarrow \pi_T(t_0, t_1, \ldots) \in Y.$$

Definition ($\mathbb{SG}$-reduction)

We write $X \leq_{SG} Y$ iff Player II has a winning strategy in $\mathbb{SG}(X, Y)$. And as usual $X \equiv_{SG} Y$ iff $X \leq_{SG} Y \leq_{SG} X$.

Definition ($\mathbb{FSG}$-hierarchy)

The collection of finite pointed ω-semigroups ordered by the $\leq_{SG}$-relation is called *the* $\mathbb{FSG}$*-hierarchy*.

Proposition ($\mathbb{SG}$-Borel Determinacy)

Let (S, X) and (T, Y) be two finite pointed ω-semigroups. Then the game $\mathbb{SG}(X, Y)$ is determined.

Proof.

A consequence of Borel Wadge determinacy. A given player has a w. s. in $\mathbb{SG}(X, Y)$ iff this same player has a w. s. in the $\mathbb{W}(\pi_S^{-1}(X), \pi_T^{-1}(Y))$. □

Definition ($\mathbb{FSG}$-hierarchy)

The collection of finite pointed ω-semigroups ordered by the $\leq_{SG}$-relation is called *the $\mathbb{FSG}$-hierarchy*.

Proposition ($\mathbb{SG}$-Borel Determinacy)

Let (S, X) and (T, Y) be two finite pointed ω-semigroups. Then the game $\mathbb{SG}(X, Y)$ is determined.

Proof.

A consequence of Borel Wadge determinacy. A given player has a w. s. in $\mathbb{SG}(X, Y)$ iff this same player has a w. s. in the $\mathbb{W}(\pi_S^{-1}(X), \pi_T^{-1}(Y))$. □

Definition ($\mathbb{FSG}$-hierarchy)

The collection of finite pointed ω-semigroups ordered by the $\leq_{SG}$-relation is called *the* $\mathbb{FSG}$*-hierarchy*.

Proposition ($\mathbb{SG}$-Borel Determinacy)

Let (S, X) *and* (T, Y) *be two finite pointed* ω*-semigroups. Then the game* $\mathbb{SG}(X, Y)$ *is determined.*

Proof.

A consequence of Borel Wadge determinacy. A given player has a w. s. in $\mathbb{SG}(X, Y)$ iff this same player has a w. s. in the $\mathbb{W}(\pi_S^{-1}(X), \pi_T^{-1}(Y))$. □

Corollary

- *The $\leq_{SG}$-antichains have length at most two.*
- *If X and Y are incomparable, then $X \equiv_{SG} Y^c$.*
- *The partial ordering $\leq_{SG}$ is wellfounded on finite pointed ω-semigroups.*

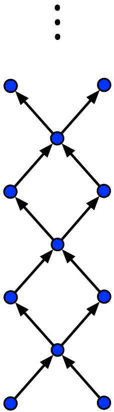
Corollary

- *The $\leq_{SG}$-antichains have length at most two.*
- *If X and Y are incomparable, then $X \equiv_{SG} Y^c$.*
- *The partial ordering $\leq_{SG}$ is wellfounded on finite pointed ω-semigroups.*

Corollary

- *The $\leq_{SG}$-antichains have length at most two.*
- *If X and Y are incomparable, then $X \equiv_{SG} Y^c$.*
- *The partial ordering $\leq_{SG}$ is wellfounded on finite pointed ω-semigroups.*

The $\mathbb{FSG}$-hierarchy



Theorem

The $\mathbb{FSG}$-hierarchy and the Wagner hierarchy are isomorphic.

Proof.

The mapping which associates every ω-rational language with its syntactic image is an isomorphism between the Wagner and the $\mathbb{FSG}$-hierarchies. $\square$

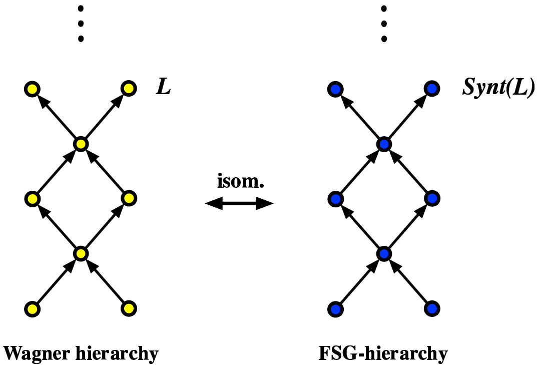
Theorem

The $\mathbb{FSG}$-hierarchy and the Wagner hierarchy are isomorphic.

Proof.

The mapping which associates every ω-rational language with its syntactic image is an isomorphism between the Wagner and the $\mathbb{FSG}$-hierarchies. □

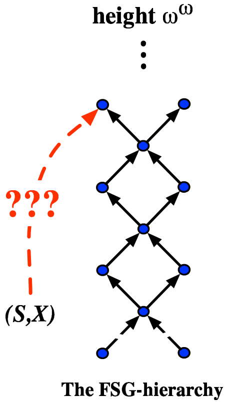
An ω-rational language and its syntactic image have the same degrees.



Corollary

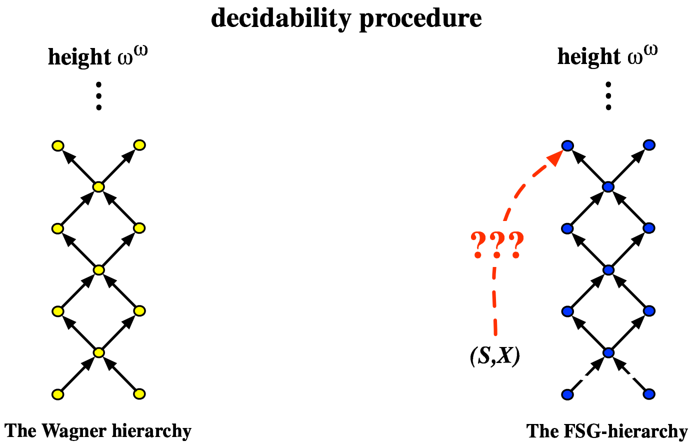
The $\mathbb{FSG}$-hierarchy has height ω^ω, and it is decidable.

decidability procedure



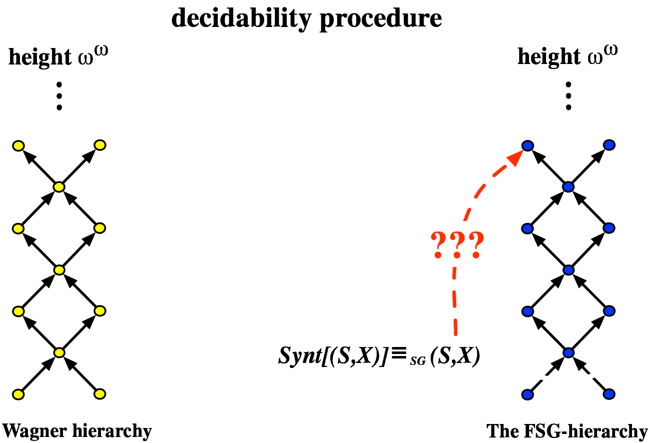
Corollary

The $\mathbb{FSG}$-hierarchy has height ω^ω, and it is decidable.



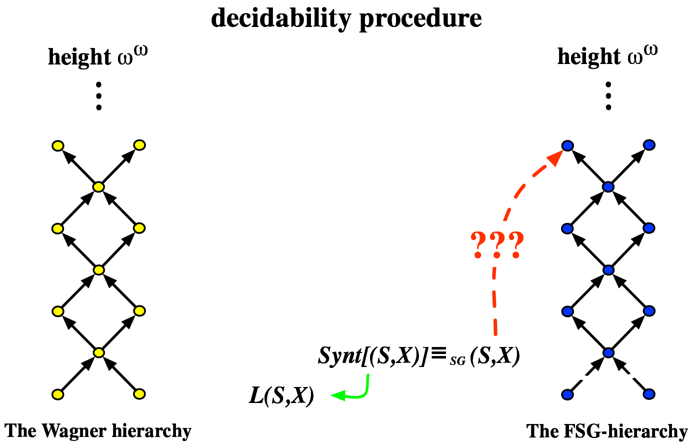
Corollary

The $\mathbb{FSG}$-hierarchy has height ω^ω, and it is decidable.



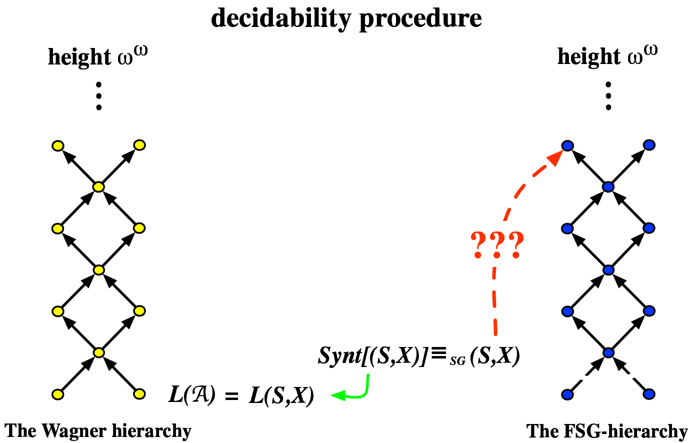
Corollary

The $\mathbb{FSG}$-hierarchy has height ω^ω, and it is decidable.



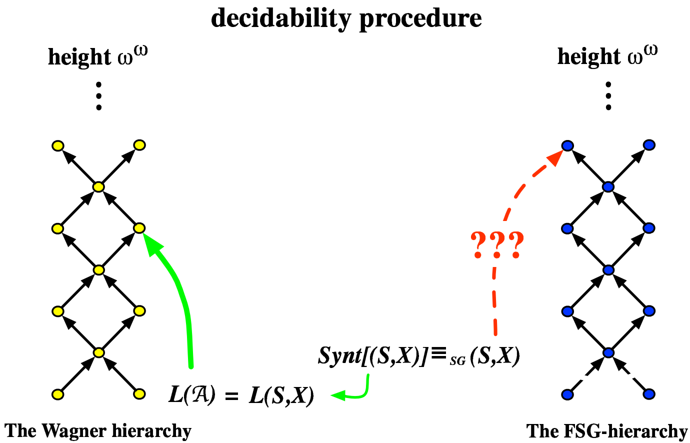
Corollary

The $\mathbb{FSG}$-hierarchy has height ω^ω, and it is decidable.



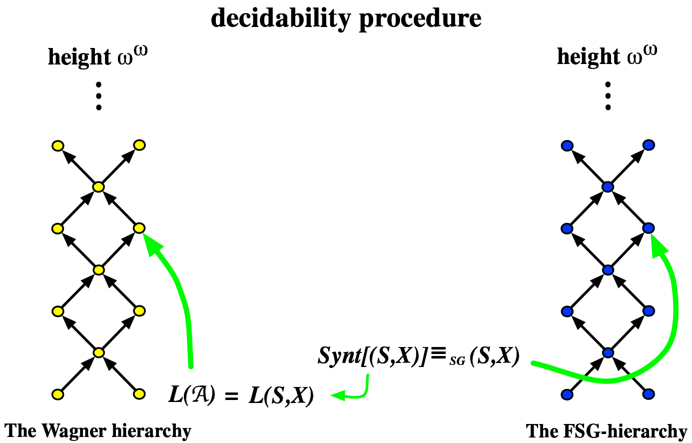
Corollary

The $\mathbb{FSG}$-hierarchy has height ω^ω, and it is decidable.

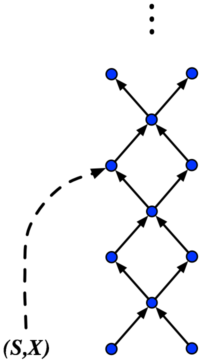


Corollary

The $\mathbb{FSG}$-hierarchy has height ω^ω, and it is decidable.

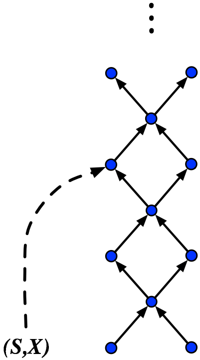


We also have a **direct** decidability procedure...



One can compute the Wagner degree of an ω-rational language directly on its syntactic image.

We also have a **direct** decidability procedure...



One can compute the Wagner degree of an ω-rational language directly on its syntactic image.

Summary

- Characterization of the algebraic counterpart of the Wagner hierarchy.
- Decidability procedure of this hierarchy.
- One can compute the Wagner degree of an ω-rational language directly on its syntactic image.

Summary

- Characterization of the algebraic counterpart of the Wagner hierarchy.
- Decidability procedure of this hierarchy.
- One can compute the Wagner degree of an ω-rational language directly on its syntactic image.

Summary

- Characterization of the algebraic counterpart of the Wagner hierarchy.
- Decidability procedure of this hierarchy.
- One can compute the Wagner degree of an ω-rational language directly on its syntactic image.

Summary

- Characterization of the algebraic counterpart of the Wagner hierarchy.
- Decidability procedure of this hierarchy.
- One can compute the Wagner degree of an ω-rational language directly on its syntactic image.