Languages Defined by Recurrent Circuits

Christian Reitwießner

Department of Computer Science
University of Würzburg

Walkshop, 1. Oktober 2007

Outline

Examples of Regular Expression

- $L_2 \stackrel{\text{df}}{=} (aa \cup b)^*$
  (words containing *a*-blocks of even length)
- $L_3 \stackrel{\text{df}}{=} (aaa \cup b)^*$
  (. . . of length multiple of three)
- $L_6 \stackrel{\text{df}}{=} L_2 \cap L_3 = (aa \cup b)^* \cap (aaa \cup b)^*$
  (. . . of length multiple of six)

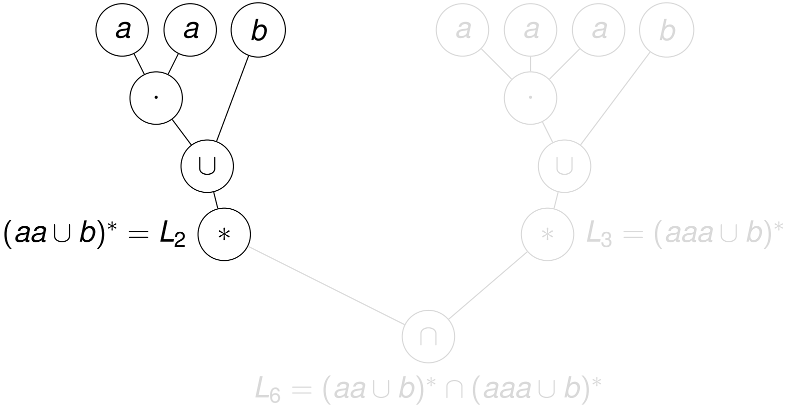
Examples of Regular Expression

- $L_2 \stackrel{\text{df}}{=} (aa \cup b)^*$
  (words containing *a*-blocks of even length)
- $L_3 \stackrel{\text{df}}{=} (aaa \cup b)^*$
  (... of length multiple of three)
- $L_6 \stackrel{\text{df}}{=} L_2 \cap L_3 = (aa \cup b)^* \cap (aaa \cup b)^*$
  (... of length multiple of six)

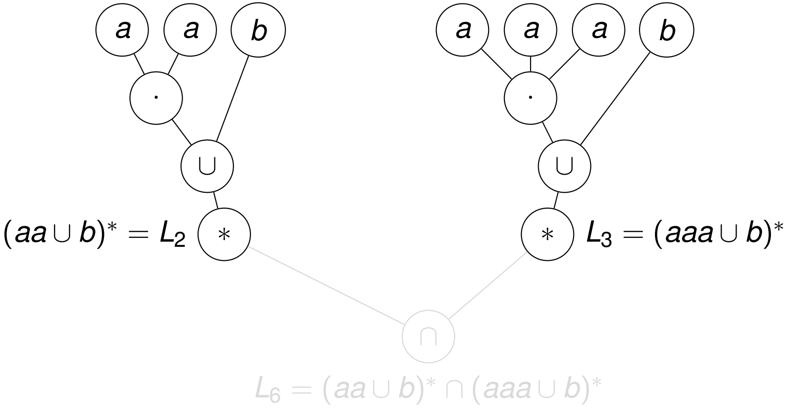
Examples of Regular Expression

- $L_2 \overset{\text{df}}{=} (aa \cup b)^*$
  (words containing *a*-blocks of even length)
- $L_3 \overset{\text{df}}{=} (aaa \cup b)^*$
  (... of length multiple of three)
- $L_6 \overset{\text{df}}{=} L_2 \cap L_3 = (aa \cup b)^* \cap (aaa \cup b)^*$
  (... of length multiple of six)

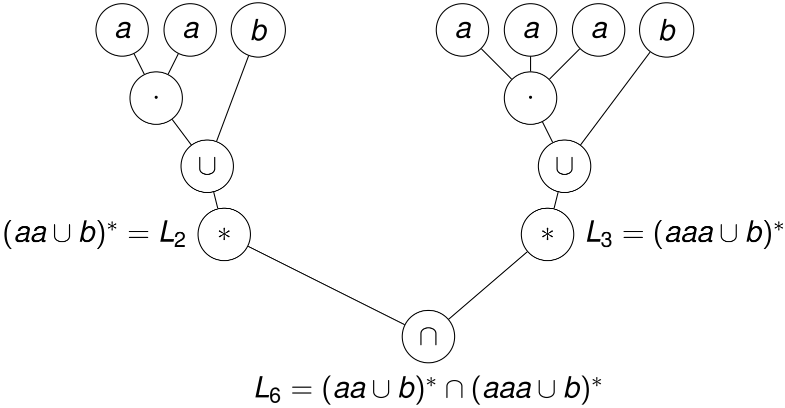
Parse Tree of a Regular Expression



Parse Tree of a Regular Expression



Parse Tree of a Regular Expression



From Regular Expression to Recurrent Circuits

- Regular expressions can be seen as combinatoric circuits with
  - letters as inputs and
  - language-operations in the computing gates.
- What class of languages do we get if we allow non-combinatoric circuits?

From Regular Expression to Recurrent Circuits

- Regular expressions can be seen as combinatoric circuits with
  - letters as inputs and
  - language-operations in the computing gates.
- What class of languages do we get if we allow non-combinatoric circuits?

Definition of the Syntax

For $\mathcal{O} \subseteq \{\cup, \cap, \cdot\}$ a recurrent $\mathcal{O}$-circuit C over the alphabet Σ is

- a directed graph (V, E) with ordered edges
- where to each gate in V is assigned:
  - one operation from $\mathcal{O}$ and
  - an initial set that can be
- output gates $V' \subseteq V$

Definition of the Syntax

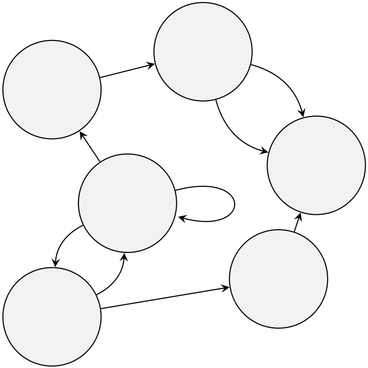
For $\mathcal{O} \subseteq \{\cup, \cap, \cdot\}$ a recurrent $\mathcal{O}$-circuit C over the alphabet Σ is

- a directed graph (V, E) with ordered edges
- where to each gate in V is assigned:
  - one operation from $\mathcal{O}$ and
  - an initial set that can be
    - $\{w\}$ for some $w \in \Sigma^*$,
    - $\emptyset$
- output gates $V' \subseteq V$

Definition of the Syntax

For $\mathcal{O} \subseteq \{\cup, \cap, \cdot\}$ a recurrent $\mathcal{O}$-circuit C over the alphabet Σ is

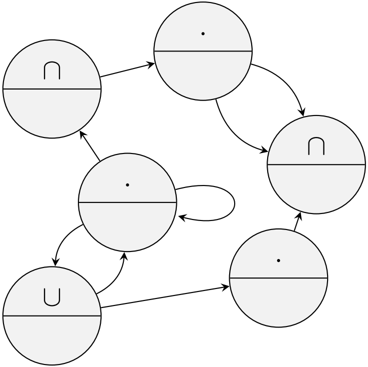
- a directed graph (V, E) with ordered edges
- where to each gate in V is assigned:
  - one operation from $\mathcal{O}$ and
  - an initial set that can be
    - $\{w\}$ for some $w \in \Sigma^*$,
    - $\emptyset$
- output gates $V' \subseteq V$



Definition of the Syntax

For $\mathcal{O} \subseteq \{\cup, \cap, \cdot\}$ a recurrent $\mathcal{O}$-circuit C over the alphabet Σ is

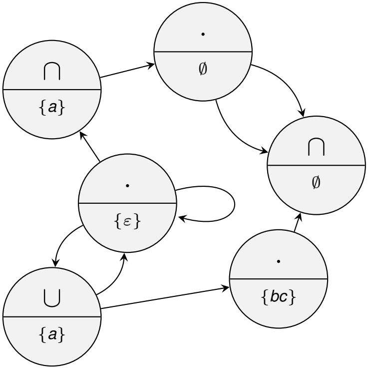
- a directed graph (V, E) with ordered edges
- where to each gate in V is assigned:
  - one operation from $\mathcal{O}$ and
  - an initial set that can be
    - $\{w\}$ for some $w \in \Sigma^*$,
    - $\emptyset$
- output gates $V' \subseteq V$



Definition of the Syntax

For $\mathcal{O} \subseteq \{\cup, \cap, \cdot\}$ a recurrent $\mathcal{O}$-circuit C over the alphabet Σ is

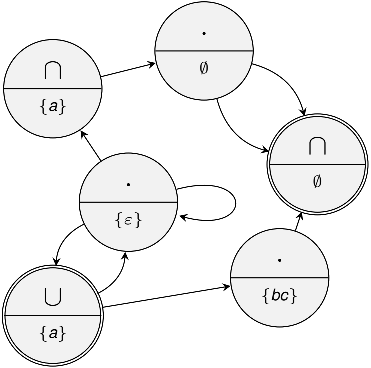
- a directed graph (V, E) with ordered edges
- where to each gate in V is assigned:
  - one operation from $\mathcal{O}$ and
  - an initial set that can be
    - $\{w\}$ for some $w \in \Sigma^*$,
    - $\emptyset$
- output gates $V' \subseteq V$



Definition of the Syntax

For $\mathcal{O} \subseteq \{\cup, \cap, \cdot\}$ a recurrent $\mathcal{O}$-circuit C over the alphabet Σ is

- a directed graph (V, E) with ordered edges
- where to each gate in V is assigned:
  - one operation from $\mathcal{O}$ and
  - an initial set that can be
    - $\{w\}$ for some $w \in \Sigma^*$,
    - $\emptyset$
- output gates $V' \subseteq V$



Definition of the Semantics

Let C be a recurrent circuit. For each node v we define:

- $C(v, 0) \stackrel{\text{df}}{=}$ initial set of v
- If $u_1, u_2, \ldots, u_n$ are the ordered predecessors of v and o is the operation of v then

$$C(v, t+1) \stackrel{\text{df}}{=} C(v, t) \cup \begin{cases} \bigcup_{i=1}^{n} C(u_i, t) & \text{if } o = \cup \\ \bigcap_{i=1}^{n} C(u_i, t) & \text{if } o = \cap \\ C(u_1, t) \cdot \ldots \cdot C(u_n, t) & \text{if } o = \cdot \end{cases}$$

- $C(v) \stackrel{\text{df}}{=} \bigcup_{t=0}^{\infty} C(v, t)$

Definition of the Semantics

Let C be a recurrent circuit. For each node v we define:

- $C(v,0) \stackrel{\text{df}}{=}$ initial set of v
- If $u_1, u_2, \ldots, u_n$ are the ordered predecessors of v and o is the operation of v then

$$C(v,t+1) \stackrel{\text{df}}{=} C(v,t) \cup \begin{cases} \bigcup_{i=1}^{n} C(u_i,t) & \text{if } o = \cup \\ \bigcap_{i=1}^{n} C(u_i,t) & \text{if } o = \cap \\ C(u_1,t) \cdot \ldots \cdot C(u_n,t) & \text{if } o = \cdot \end{cases}$$

- $C(v) \stackrel{\text{df}}{=} \bigcup_{t=0}^{\infty} C(v,t)$

Definition of the Semantics

Let C be a recurrent circuit. For each node v we define:

- $C(v,0) \stackrel{\text{df}}{=}$ initial set of v
- If $u_1, u_2, \ldots, u_n$ are the ordered predecessors of v and o is the operation of v then

$$C(v,t+1) \stackrel{\text{df}}{=} C(v,t) \cup \begin{cases} \bigcup_{i=1}^{n} C(u_i,t) & \text{if } o = \cup \\ \bigcap_{i=1}^{n} C(u_i,t) & \text{if } o = \cap \\ C(u_1,t)\cdot \ldots \cdot C(u_n,t) & \text{if } o = \cdot \end{cases}$$

- $C(v) \stackrel{\text{df}}{=} \bigcup_{t=0}^{\infty} C(v,t)$

Definition of Recurrent Circuit Classes

Definition

- For a recurrent circuit C with output gates V' we define $L(C) \overset{\mathrm{df}}{=} \bigcup_{v \in V'} C(v)$.
- For any $\mathcal{O} \subseteq \{\cup, \cap, \cdot\}$ we define $\mathrm{RC}(\mathcal{O}) \overset{\mathrm{df}}{=} \{L \mid L = L(C)$ for some recurrent $\mathcal{O}$-circuit $C\}$.

Overview of the Classes

$$\mathrm{CSL}$$

$$\mathrm{RC}(\cup,\cap,\cdot) = \mathrm{CCFL}$$

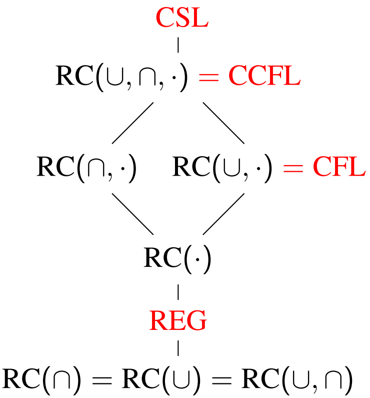
$$\mathrm{RC}(\cap,\cdot) \qquad \mathrm{RC}(\cup,\cdot) = \mathrm{CFL}$$

$$\mathrm{RC}(\cdot)$$

$$\mathrm{REG}$$

$$\mathrm{RC}(\cap) = \mathrm{RC}(\cup) = \mathrm{RC}(\cup,\cap)$$

Overview of the Classes



First Observations

Theorem

For every gate v of the circuit C with operation o and predecessors $u_1, u_2, \ldots, u_n$:

$$C(v) = C(v,0) \cup \begin{cases} \bigcup_{i=1}^{n} C(u_i) & \text{if } o = \cup \\ \bigcap_{i=1}^{n} C(u_i) & \text{if } o = \cap \\ C(u_1) \cdot \ldots \cdot C(u_n) & \text{if } o = \cdot \end{cases}$$

Theorem

$\mathrm{RC}(\mathcal{O})$ *is closed under any operation in* $\mathcal{O}$ *for* $\mathcal{O} \subseteq \{\cup, \cap, \cdot\}$.

First Observations

Theorem

For every gate v of the circuit C with operation o and predecessors $u_1, u_2, \ldots, u_n$:

$$C(v) = C(v,0) \cup \begin{cases} \bigcup_{i=1}^{n} C(u_i) & \text{if } o = \cup \\ \bigcap_{i=1}^{n} C(u_i) & \text{if } o = \cap \\ C(u_1) \cdot \ldots \cdot C(u_n) & \text{if } o = \cdot \end{cases}$$

Theorem

$\mathrm{RC}(\mathcal{O})$ *is closed under any operation in* $\mathcal{O}$ *for* $\mathcal{O} \subseteq \{\cup, \cap, \cdot\}$.

Equivalence to Context Free Languages

Theorem

$\mathrm{RC}(\cup, \cdot) = \mathrm{CFL}$

Sketch of Construction.

Gates and nonterminals correspond in a natural way:
Concatenation gate A with predecessors $A_1, A_2, \ldots, A_n$ corresponds to production $A \to A_1 A_2 \ldots A_n$.
Union gate B with predecessors $B_1, B_2, \ldots, B_n$ corresponds to production $B \to B_1 \mid B_2 \mid \cdots \mid B_n$. □

Equivalence to Context Free Languages

Theorem

$\mathrm{RC}(\cup, \cdot) = \mathrm{CFL}$

Sketch of Construction.

Gates and nonterminals correspond in a natural way:
Concatenation gate A with predecessors $A_1, A_2, \ldots, A_n$ corresponds to production $A \to A_1 A_2 \ldots A_n$.
Union gate B with predecessors $B_1, B_2, \ldots, B_n$ corresponds to production $B \to B_1 \mid B_2 \mid \cdots \mid B_n$. □

Conjunctive Context Free Languages (CCFL)

- Defined by Okhotin in 2001.
- Context free grammars with intersection.
- Contain productions of the form $A \to w_1 \& w_2 \& \ldots \& w_n$ with $w_i \in (N \cup \Sigma)^*$.

Definition

$x \in (N \cup \Sigma)^*$ can be derived from $w_1 \& w_2 \& \ldots \& w_n$ iff it can be derived from all w_i, $i = 1, 2, \ldots, n$.

$\mathrm{RC}(\cup, \cap, \cdot) = \mathrm{CCFL}$

Theorem

$\mathrm{RC}(\cup, \cap, \cdot) = \mathrm{CCFL}$

Sketch of Construction.

Additionally to proof of $\mathrm{RC}(\cup, \cdot) = \mathrm{CFL}$:
Intersection gate C with predecessors $C_1, C_2, \ldots, C_n$ corresponds to production $C \to C_1 \& C_2 \& \ldots \& C_n$. □

$\mathrm{RC}(\cup, \cap, \cdot) = \mathrm{CCFL}$

Theorem

$\mathrm{RC}(\cup, \cap, \cdot) = \mathrm{CCFL}$

Sketch of Construction.

Additionally to proof of $\mathrm{RC}(\cup, \cdot) = \mathrm{CFL}$:
Intersection gate C with predecessors $C_1, C_2, \ldots, C_n$ corresponds to production $C \to C_1 \& C_2 \& \ldots \& C_n$. □

Facts about CCFL by Okhotin

- Efficient parsing algorithm: Adaption of CYK, also $O(n^3)$.
- This implies $\mathrm{CCFL} \subsetneq \mathrm{P}$.
- There is a P-complete language in CCFL.
- $\Gamma_\cap(\mathrm{CFL}) \subsetneq \mathrm{CCFL} \subseteq \mathrm{CSL}$
- There are non-regular unary CCFL [Jeż, 2007], as opposed to unary CFL.

Facts about CCFL by Okhotin

- Efficient parsing algorithm: Adaption of CYK, also $O(n^3)$.
- This implies $\mathrm{CCFL} \subsetneq \mathrm{P}$.
- There is a P-complete language in CCFL.
- $\Gamma_{\cap}(\mathrm{CFL}) \subsetneq \mathrm{CCFL} \subseteq \mathrm{CSL}$
- There are non-regular unary CCFL [Jeż, 2007], as opposed to unary CFL.

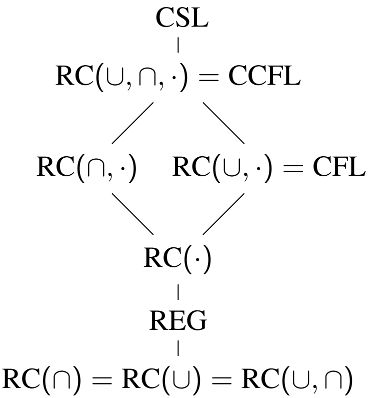
Facts about CCFL by Okhotin

- Efficient parsing algorithm: Adaption of CYK, also $O(n^3)$.
- This implies $\mathrm{CCFL} \subsetneq \mathrm{P}$.
- There is a P-complete language in CCFL.
- $\Gamma_\cap(\mathrm{CFL}) \subsetneq \mathrm{CCFL} \subseteq \mathrm{CSL}$
- There are non-regular unary CCFL [Jeż, 2007], as opposed to unary CFL.

Facts about CCFL by Okhotin

- Efficient parsing algorithm: Adaption of CYK, also $O(n^3)$.
- This implies $\mathrm{CCFL} \subsetneq \mathrm{P}$.
- There is a P-complete language in CCFL.
- $\Gamma_\cap(\mathrm{CFL}) \subsetneq \mathrm{CCFL} \subseteq \mathrm{CSL}$
- There are non-regular unary CCFL [Jeż, 2007], as opposed to unary CFL.

Overview of the Classes



$\mathrm{RC}(\cdot)$

$\mathrm{RC}(\cdot)$

- can be defined by restriction of PDAs.
- is incomparable to deterministic PDAs.
- is neither closed under $\cap$ nor under $\overline{}$.
- is closed under $\cup, \cdot, ^*$ and homomorphisms.

$\mathrm{RC}(\cdot)$

$\mathrm{RC}(\cdot)$

- can be defined by restriction of PDAs.
- is incomparable to deterministic PDAs.
- is neither closed under $\cap$ nor under $\overline{}$.
- is closed under $\cup, \cdot, ^*$ and homomorphisms.

$\mathrm{RC}(\cdot)$

$\mathrm{RC}(\cdot)$

- can be defined by restriction of PDAs.
- is incomparable to deterministic PDAs.
- is neither closed under $\cap$ nor under $\overline{}$.
- is closed under $\cup, \cdot, ^*$ and homomorphisms.

$\mathrm{RC}(\cdot)$

$\mathrm{RC}(\cdot)$

- can be defined by restriction of PDAs.
- is incomparable to deterministic PDAs.
- is neither closed under $\cap$ nor under $^{-}$.
- is closed under $\cup, \cdot, ^{*}$ and homomorphisms.

Nice (Non-)Closure Proofs for $\mathrm{RC}(\mathcal{O})$

Theorem

$\mathrm{RC}(\cdot)$ *is not closed under intersection with regular languages.*

Sketch of Proof.

Theorem of Chomsky and Schützenberger:
$L \in \mathrm{CFL} \iff L = h(D_n^* \cap R)$, for some homomorphism h, $n \in \mathbb{N}$ and $R \in \mathrm{REG}$
(D_n^* is the (one-sided) Dyck language with n bracket-types).
Since $D_n^* \in \mathrm{RC}(\cdot)$, this would imply $\mathrm{RC}(\cdot) = \mathrm{CFL}$, but $\mathrm{RC}(\cdot) \subsetneq \mathrm{CFL}$. □

Nice (Non-)Closure Proofs for $\mathrm{RC}(\mathcal{O})$

Theorem

$\mathrm{RC}(\cdot)$ *is not closed under intersection with regular languages.*

Sketch of Proof.

Theorem of Chomsky and Schützenberger:
$L \in \mathrm{CFL} \iff L = h(D_n^* \cap R)$, for some homomorphism h, $n \in \mathbb{N}$ and $R \in \mathrm{REG}$
(D_n^* is the (one-sided) Dyck language with n bracket-types).
Since $D_n^* \in \mathrm{RC}(\cdot)$, this would imply $\mathrm{RC}(\cdot) = \mathrm{CFL}$, but $\mathrm{RC}(\cdot) \subsetneq \mathrm{CFL}$. □

Other Nice (Non-)Closure Proofs for $\mathrm{RC}(\mathcal{O})$

Theorem

$\mathrm{RC}(\cdot)$ *is not closed under inverse homomorphisms.*

Sketch of Proof.

Theorem from AFL-Theory:
Let $\mathcal{K}$ be an ε-free language class.
$\mathcal{K}$ closed under $\cdot$, ε-free h and h^{-1} $\Rightarrow$ $\mathcal{K}$ closed under $\cap\mathrm{REG}$.
But the conclusion has just been disproved. □

Other Nice (Non-)Closure Proofs for $\mathrm{RC}(\mathcal{O})$

Theorem

$\mathrm{RC}(\cdot)$ *is not closed under inverse homomorphisms.*

Sketch of Proof.

Theorem from AFL-Theory:
Let $\mathcal{K}$ be an ε-free language class.
$\mathcal{K}$ closed under $\cdot$, ε-free h and h^{-1} $\Rightarrow$ $\mathcal{K}$ closed under $\cap\mathrm{REG}$.
But the conclusion has just been disproved. □

$\mathrm{RC}(\cap, \cdot)$ is Not Closed Under Homomorphisms

Theorem

$\mathrm{RC}(\cap, \cdot)$ *is not closed under homomorphisms.*

Sketch of Proof.

Chomsky-Schützenberger: $\mathrm{CFL} = h(D^* \wedge \mathrm{REG})$
Thus $\mathrm{CFL} \subseteq \Gamma_{h,\cap}(\mathrm{RC}(\cdot)) \Rightarrow \Gamma_{h,\cap}(\mathrm{CFL}) \subseteq \Gamma_{h,\cap}(\mathrm{RC}(\cdot))$
Theorem of Ginsburg, Greibach, Harrison: $\mathrm{RE} = \Gamma_{h,\cap}(\mathrm{CFL})$
$\Rightarrow \mathrm{RE} \subseteq \mathrm{RC}(\cap, \cdot)$, contradiction. □

$\mathrm{RC}(\cap, \cdot)$ is Not Closed Under Homomorphisms

Theorem

$\mathrm{RC}(\cap, \cdot)$ *is not closed under homomorphisms.*

Sketch of Proof.

Chomsky-Schützenberger: $\mathrm{CFL} = h(D^* \wedge \mathrm{REG})$

Thus $\mathrm{CFL} \subseteq \Gamma_{h,\cap}(\mathrm{RC}(\cdot)) \Rightarrow \Gamma_{h,\cap}(\mathrm{CFL}) \subseteq \Gamma_{h,\cap}(\mathrm{RC}(\cdot))$

Theorem of Ginsburg, Greibach, Harrison: $\mathrm{RE} = \Gamma_{h,\cap}(\mathrm{CFL})$

$\Rightarrow \mathrm{RE} \subseteq \mathrm{RC}(\cap, \cdot)$, contradiction. □

Summary and Open Problems

- Interesting language classes characterized by recurrent circuits:
  - $\mathrm{RC}(\cup, \cap, \cdot)$, an extension of context free grammars.
  - $\mathrm{RC}(\cup, \cdot)$, context free grammars.
  - $\mathrm{RC}(\cap, \cdot)$, incomparable to context free grammars.
  - $\mathrm{RC}(\cdot)$, a restriction of context free grammars.
- Open problems:
  - Which of the inclusions $\Gamma_{\cap}(\mathrm{RC}(\cdot)) \subseteq \mathrm{RC}(\cap, \cdot) \subseteq \mathrm{RC}(\cup, \cap, \cdot)$ is strict?
  - Closure of $\mathrm{RC}(\cap, \cdot)$ and $\mathrm{RC}(\cap, \cup, \cdot)$ under ε-free homomorphisms and complementation.
- Are there other reasonable possibilities for the initial sets (finite sets, regular languages, …)?

Summary and Open Problems

- Interesting language classes characterized by recurrent circuits:
  - $\mathrm{RC}(\cup, \cap, \cdot)$, an extension of context free grammars.
  - $\mathrm{RC}(\cup, \cdot)$, context free grammars.
  - $\mathrm{RC}(\cap, \cdot)$, incomparable to context free grammars.
  - $\mathrm{RC}(\cdot)$, a restriction of context free grammars.
- Open problems:
  - Which of the inclusions $\Gamma_\cap(\mathrm{RC}(\cdot)) \subseteq \mathrm{RC}(\cap, \cdot) \subseteq \mathrm{RC}(\cup, \cap, \cdot)$ is strict?
  - Closure of $\mathrm{RC}(\cap, \cdot)$ and $\mathrm{RC}(\cap, \cup, \cdot)$ under ε-free homomorphisms and complementation.
- Are there other reasonable possibilities for the initial sets (finite sets, regular languages, ...)?

Summary and Open Problems

- Interesting language classes characterized by recurrent circuits:
  - $\mathrm{RC}(\cup, \cap, \cdot)$, an extension of context free grammars.
  - $\mathrm{RC}(\cup, \cdot)$, context free grammars.
  - $\mathrm{RC}(\cap, \cdot)$, incomparable to context free grammars.
  - $\mathrm{RC}(\cdot)$, a restriction of context free grammars.
- Open problems:
  - Which of the inclusions $\Gamma_\cap(\mathrm{RC}(\cdot)) \subseteq \mathrm{RC}(\cap, \cdot) \subseteq \mathrm{RC}(\cup, \cap, \cdot)$ is strict?
  - Closure of $\mathrm{RC}(\cap, \cdot)$ and $\mathrm{RC}(\cap, \cup, \cdot)$ under ε-free homomorphisms and complementation.
- Are there other reasonable possibilities for the initial sets (finite sets, regular languages, . . .)?

Summary and Open Problems

- Interesting language classes characterized by recurrent circuits:
  - $\mathrm{RC}(\cup, \cap, \cdot)$, an extension of context free grammars.
  - $\mathrm{RC}(\cup, \cdot)$, context free grammars.
  - $\mathrm{RC}(\cap, \cdot)$, incomparable to context free grammars.
  - $\mathrm{RC}(\cdot)$, a restriction of context free grammars.
- Open problems:
  - Which of the inclusions $\Gamma_\cap(\mathrm{RC}(\cdot)) \subseteq \mathrm{RC}(\cap, \cdot) \subseteq \mathrm{RC}(\cup, \cap, \cdot)$ is strict?
  - Closure of $\mathrm{RC}(\cap, \cdot)$ and $\mathrm{RC}(\cap, \cup, \cdot)$ under ε-free homomorphisms and complementation.
- Are there other reasonable possibilities for the initial sets (finite sets, regular languages, . . .)?