On expressive power of regular expressions with subroutine calls and lookaround assertions

Ondřej Guth
ondrej.guth@fit.cvut.cz

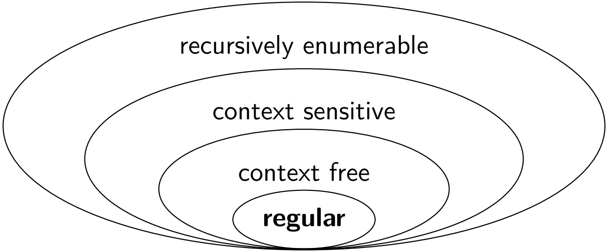
Department of Theoretical Computer Science
Faculty of Information Technology
Czech Technical University in Prague

The Prague Stringology Conference 2023

Regular expressions[1] match regular languages[2]

Regular expression is the minimal set of operations to express regular language

$(\texttt{my} + \varepsilon)(\texttt{great})^*\texttt{grandfather}$



[1] Stephen Cole Kleene. "Representation of Events in Nerve Nets and Finite Automata". In: *Automata Studies. (AM-34)*. Ed. by C. E. Shannon and J. McCarthy. Princeton University Press, Dec. 1956, pp. 3–42. ISBN: 978-1-4008-8261-8. DOI: 10.1515/9781400882618-002. (Visited on 08/20/2021).

[2] Noam Chomsky. "On Certain Formal Properties of Grammars". In: *Information and Control* (1959). ISSN: 00199958. DOI: 10.1016/S0019-9958(59)90362-6.

Practical "regular" expressions are different and have various flavours

PCRE2: $\texttt{a}^n\texttt{b}^n$

$(\texttt{a}{\color{red}(?1)}?\texttt{b})$

What language class can be expressed by a particular regex flavour?

?

Expressive power of certain combinations of features was already known

Character class, interval quantifier, concatenation, alternative, iteration: RLs

`[Mm]y ((great){1,3}-grand)(father|mother)`

Expressive power of certain combinations of features was already known

Character class, interval quantifier, concatenation, alternative, iteration: RLs

`[Mm]y ((great){1,3}-grand)(father|mother)`

Zero-width lookaround assertions: RLs

`girls? need((?<=s need)|(?<=l need)s)`

Expressive power of certain combinations of features was already known

Character class, interval quantifier, concatenation, alternative, iteration: RLs

`[Mm]y ((great){1,3}-grand)(father|mother)`

Zero-width lookaround assertions: RLs

`girls? need((?<=s need)|(?<=l need)s)`

Backreferences: some CSLs and not some CFLs

`([a-z]{10})\1`

Formalisation of regex: matching relation[3]

Other formalisms exist

Definition

A matching relation $\rightsquigarrow$ is of the form $(\boldsymbol{r}, \boldsymbol{x}, i) \rightsquigarrow \mathcal{R}$ where $\mathcal{R} = \{i : i \in \mathbb{N} \wedge i \leq |\boldsymbol{x}| + 1\}$ (matching result).

Definition

The language of a regex $\boldsymbol{r} \in \mathbb{E}_{\mathrm{LS},\mathcal{A},\mathcal{X}}$ is $\mathrm{L}(\boldsymbol{r}) = \{\boldsymbol{x} : (\boldsymbol{r}, \boldsymbol{x}, 1) \rightsquigarrow \mathcal{R} \wedge |\boldsymbol{x}| + 1 \in \mathcal{R}\}$.

[3] Nariyoshi Chida and Tachio Terauchi. "On Lookaheads in Regular Expressions with Backreferences". In: *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*. Ed. by Amy P. Felty. Vol. 228. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 15:1–15:18. ISBN: 978-3-95977-233-4. DOI: 10.4230/LIPIcs.FSCD.2022.15.

Formalisation of regex: matching relation[3]

Other formalisms exist

Definition

A matching relation $\rightsquigarrow$ is of the form $(\boldsymbol{r}, \boldsymbol{x}, i) \rightsquigarrow \mathcal{R}$ where $\mathcal{R} = \{i : i \in \mathbb{N} \land i \leq |\boldsymbol{x}| + 1\}$ (matching result).

Definition

The language of a regex $\boldsymbol{r} \in \mathbb{E}_{\mathrm{LS},\mathcal{A},\mathcal{X}}$ is $\mathrm{L}(\boldsymbol{r}) = \{\boldsymbol{x} : (\boldsymbol{r}, \boldsymbol{x}, 1) \rightsquigarrow \mathcal{R} \land |\boldsymbol{x}| + 1 \in \mathcal{R}\}$.

Example

$$\frac{a \in \mathcal{A} \land i \leq |\boldsymbol{x}| \land \boldsymbol{x}[i] = a}{(a, \boldsymbol{x}, i) \rightsquigarrow \{i + 1\}}$$

Example (Lookahead)

$$\frac{(\boldsymbol{r}, \boldsymbol{x}, i) \rightsquigarrow \mathcal{R}}{((?{=}\boldsymbol{r}), \boldsymbol{x}, i) \rightsquigarrow \{i \land \mathcal{R} \neq \emptyset\}}$$

---

[3] Nariyoshi Chida and Tachio Terauchi. "On Lookaheads in Regular Expressions with Backreferences". In: *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*. Ed. by Amy P. Felty. Vol. 228. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 15:1–15:18. ISBN: 978-3-95977-233-4. DOI: 10.4230/LIPIcs.FSCD.2022.15.

Formalisation of subroutine calls

Definition (Numbered subroutine call)

$$\frac{(\sigma(l), \boldsymbol{x}, i) \rightsquigarrow \mathcal{R}}{((?l), \boldsymbol{x}, i) \rightsquigarrow \mathcal{R}}$$

Definition (Named subroutine call)

$$\frac{(\sigma(\nu(N)), \boldsymbol{x}, i) \rightsquigarrow \mathcal{R}}{((?\mathrm{P}{>}N), \boldsymbol{x}, i) \rightsquigarrow \mathcal{R}}$$

Formalisation of subroutine calls

Definition (Numbered subroutine call)

$$\frac{(\sigma(l), \boldsymbol{x}, i) \leadsto \mathcal{R}}{((?l), \boldsymbol{x}, i) \leadsto \mathcal{R}}$$

Definition (Named subroutine call)

$$\frac{(\sigma(\nu(N)), \boldsymbol{x}, i) \leadsto \mathcal{R}}{((?\mathrm{P}{>}N), \boldsymbol{x}, i) \leadsto \mathcal{R}}$$

Example

$$\dfrac{\dfrac{}{((?(\mathrm{DEFINE})(?{<}S{>}\texttt{a}((?1) \mid \varepsilon)\texttt{b}), \texttt{ab}, 1) \leadsto \{1\}} \qquad \dfrac{\dfrac{\dfrac{}{(\texttt{a}, \texttt{ab}, 1) \leadsto \{2\}} \quad \dfrac{\dfrac{\dfrac{\cdots}{(\texttt{a}((?1) \mid \varepsilon)\texttt{b}, \texttt{ab}, 2) \leadsto \emptyset}}{((?1), \texttt{ab}, 2) \leadsto \emptyset} \quad \dfrac{}{(\varepsilon, \texttt{ab}, 2) \leadsto \{2\}}}{(((?1) \mid \varepsilon), \texttt{ab}, 2) \leadsto \{2\}} \quad \dfrac{}{(\texttt{b}, \texttt{ab}, 2) \leadsto \{3\}}}{(\texttt{a}((?1) \mid \varepsilon)\texttt{b}, \texttt{ab}, 1) \leadsto \{3\}}}{((?\mathrm{P}{>}S), \texttt{ab}, 1) \leadsto \{3\}}}{((?(\mathrm{DEFINE})(?{<}S{>}\texttt{a}((?1) \mid \varepsilon)\texttt{b})(?\mathrm{P}{>}S), \texttt{ab}, 1) \leadsto \{3\}}$$

Example (Context-free languages)

$\{\mathtt{a}^n\mathtt{b}^n\}$, $\{\boldsymbol{w}\boldsymbol{w}^{\mathrm{R}}\}$

Example (Context-free languages)

$\{\mathtt{a}^n\mathtt{b}^n\}$, $\{\boldsymbol{w}\boldsymbol{w}^{\mathrm{R}}\}$

Definition (Context-free grammar)

A quadruple $(\mathcal{V}, \mathcal{A}, \mathcal{R}, S)$ where every member of $\mathcal{R}$ is in the form of $N \to \boldsymbol{v}, \boldsymbol{v} \in (\mathcal{A} \cup \mathcal{V})^*$.

Definition (Derivation step in CFG)

If $\boldsymbol{v}_1 \to \boldsymbol{v}_2 \in \mathcal{R}$ then $\boldsymbol{p}\boldsymbol{v}_1\boldsymbol{s} \Rightarrow \boldsymbol{p}\boldsymbol{v}_2\boldsymbol{s}$ is possible.

Definition (Language generated by grammar)

$\{\boldsymbol{x} \in \mathcal{A}^* : S \Rightarrow^* \boldsymbol{x}\}$

Example (Context-free languages)

$\{\mathtt{a}^n\mathtt{b}^n\}$, $\{\boldsymbol{w}\boldsymbol{w}^{\mathrm{R}}\}$

Definition (Context-free grammar)

A quadruple $(\mathcal{V}, \mathcal{A}, \mathcal{R}, S)$ where every member of $\mathcal{R}$ is in the form of $N \to \boldsymbol{v}, \boldsymbol{v} \in (\mathcal{A} \cup \mathcal{V})^*$.

Definition (Derivation step in CFG)

If $\boldsymbol{v}_1 \to \boldsymbol{v}_2 \in \mathcal{R}$ then $\boldsymbol{p}\boldsymbol{v}_1\boldsymbol{s} \Rightarrow \boldsymbol{p}\boldsymbol{v}_2\boldsymbol{s}$ is possible.

Definition (Language generated by grammar)

$\{\boldsymbol{x} \in \mathcal{A}^* : S \Rightarrow^* \boldsymbol{x}\}$

Example

$$(\{S\}, \{\mathtt{a}, \mathtt{b}\}, \{S \to \mathtt{a}S\mathtt{b} \mid \varepsilon\}, S)$$

$$S$$

Example (Context-free languages)

$\{\mathtt{a}^n\mathtt{b}^n\}$, $\{\boldsymbol{w}\boldsymbol{w}^{\mathrm{R}}\}$

Definition (Context-free grammar)

A quadruple $(\mathcal{V}, \mathcal{A}, \mathcal{R}, S)$ where every member of $\mathcal{R}$ is in the form of $N \to \boldsymbol{v}, \boldsymbol{v} \in (\mathcal{A} \cup \mathcal{V})^*$.

Definition (Derivation step in CFG)

If $\boldsymbol{v}_1 \to \boldsymbol{v}_2 \in \mathcal{R}$ then $\boldsymbol{p}\boldsymbol{v}_1\boldsymbol{s} \Rightarrow \boldsymbol{p}\boldsymbol{v}_2\boldsymbol{s}$ is possible.

Definition (Language generated by grammar)

$\{\boldsymbol{x} \in \mathcal{A}^* : S \Rightarrow^* \boldsymbol{x}\}$

Example

$$(\{S\}, \{\mathtt{a}, \mathtt{b}\}, \{S \to \mathtt{a}S\mathtt{b} \mid \varepsilon\}, S)$$

$$S \Rightarrow \mathtt{a}S\mathtt{b}$$

Example (Context-free languages)

$\{\mathtt{a}^n\mathtt{b}^n\}$, $\{\boldsymbol{w}\boldsymbol{w}^{\mathrm{R}}\}$

Definition (Context-free grammar)

A quadruple $(\mathcal{V}, \mathcal{A}, \mathcal{R}, S)$ where every member of $\mathcal{R}$ is in the form of $N \to \boldsymbol{v}, \boldsymbol{v} \in (\mathcal{A} \cup \mathcal{V})^*$.

Definition (Derivation step in CFG)

If $\boldsymbol{v}_1 \to \boldsymbol{v}_2 \in \mathcal{R}$ then $\boldsymbol{p}\boldsymbol{v}_1\boldsymbol{s} \Rightarrow \boldsymbol{p}\boldsymbol{v}_2\boldsymbol{s}$ is possible.

Definition (Language generated by grammar)

$\{\boldsymbol{x} \in \mathcal{A}^* : S \Rightarrow^* \boldsymbol{x}\}$

Example

$$(\{S\}, \{\mathtt{a}, \mathtt{b}\}, \{S \to \mathtt{a}S\mathtt{b} \mid \varepsilon\}, S)$$

$$S \Rightarrow \mathtt{a}S\mathtt{b} \Rightarrow \mathtt{aa}S\mathtt{bb}$$

Example (Context-free languages)

$\{\mathtt{a}^n\mathtt{b}^n\}$, $\{\boldsymbol{w}\boldsymbol{w}^{\mathrm{R}}\}$

Definition (Context-free grammar)

A quadruple $(\mathcal{V}, \mathcal{A}, \mathcal{R}, S)$ where every member of $\mathcal{R}$ is in the form of $N \to \boldsymbol{v}, \boldsymbol{v} \in (\mathcal{A} \cup \mathcal{V})^*$.

Definition (Derivation step in CFG)

If $\boldsymbol{v}_1 \to \boldsymbol{v}_2 \in \mathcal{R}$ then $\boldsymbol{p}\boldsymbol{v}_1\boldsymbol{s} \Rightarrow \boldsymbol{p}\boldsymbol{v}_2\boldsymbol{s}$ is possible.

Definition (Language generated by grammar)

$\{\boldsymbol{x} \in \mathcal{A}^* : S \Rightarrow^* \boldsymbol{x}\}$

Example

$$(\{S\}, \{\mathtt{a}, \mathtt{b}\}, \{S \to \mathtt{a}S\mathtt{b} \mid \varepsilon\}, S)$$

$$S \Rightarrow \mathtt{a}S\mathtt{b} \Rightarrow \mathtt{aa}S\mathtt{bb} \Rightarrow \mathtt{aa}\varepsilon\mathtt{bb}$$

Every context-free grammar can be expressed by a regex with subroutine calls

Conversion of a CFG to an equivalent regex

$$(\{S\}, \{\mathtt{a}, \mathtt{b}\}, \{S \to \mathtt{a}S\mathtt{b} \mid \varepsilon\}, S)$$

1. $\boldsymbol{r}_1 = \mathrm{rx}(\mathtt{a}S\mathtt{b}) = \mathtt{a}(\mathrm{?P}{>}S)\mathtt{b}$

Every context-free grammar can be expressed by a regex with subroutine calls

Conversion of a CFG to an equivalent regex

$$(\{S\}, \{\mathtt{a}, \mathtt{b}\}, \{S \to \mathtt{a}S\mathtt{b} \mid \varepsilon\}, S)$$

1. $\boldsymbol{r}_1 = \mathrm{rx}(\mathtt{a}S\mathtt{b}) = \mathtt{a}(?\mathrm{P}{>}S)\mathtt{b}$
2. $\boldsymbol{r}_2 = \varepsilon$

Every context-free grammar can be expressed by a regex with subroutine calls

Conversion of a CFG to an equivalent regex

$$(\{S\}, \{\mathtt{a}, \mathtt{b}\}, \{S \to \mathtt{a}S\mathtt{b} \mid \varepsilon\}, S)$$

1. $\boldsymbol{r}_1 = \mathrm{rx}(\mathtt{a}S\mathtt{b}) = \mathtt{a}(?\mathrm{P}{>}S)\mathtt{b}$
2. $\boldsymbol{r}_2 = \varepsilon$
3. $\boldsymbol{r} = (?(\mathrm{DEFINE})(?{<}S{>}\boldsymbol{r}_1 \mid \boldsymbol{r}_2))$

Every context-free grammar can be expressed by a regex with subroutine calls

Conversion of a CFG to an equivalent regex

$$(\{S\}, \{\texttt{a}, \texttt{b}\}, \{S \to \texttt{a}S\texttt{b} \mid \varepsilon\}, S)$$

1. $\boldsymbol{r}_1 = \mathrm{rx}(\texttt{a}S\texttt{b}) = \texttt{a}(\text{?P}{>}S)\texttt{b}$
2. $\boldsymbol{r}_2 = \varepsilon$
3. $\boldsymbol{r} = (\text{?(DEFINE)}(\text{?}{<}S{>}\boldsymbol{r}_1 \mid \boldsymbol{r}_2))$
4. $\boldsymbol{r} = (\text{?(DEFINE)}(\text{?}{<}S{>}\boldsymbol{r}_1 \mid \boldsymbol{r}_2))(\text{?P}{>}S)$

Every regex with concatenation, alternative, and subroutine call can be expressed by a context-free grammar

Conversion of a regex to an equivalent CFG

$$(?\mathrm{P}{>}N)\mathtt{a}(?(\mathrm{DEFINE})(?{<}N{>}\varepsilon \mid \mathtt{b}(?\mathrm{P}{>}N)\mathtt{b} \mid \mathtt{c}(?\mathrm{P}{>}N)\mathtt{c}))$$

- $\mathfrak{G}_\varepsilon = (\{S_\varepsilon\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_\varepsilon \to \varepsilon\}, S_\varepsilon\})$, $\mathfrak{G}_a = (\{S_\mathtt{a}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_\mathtt{a} \to \mathtt{a}\}, S_\mathtt{a})\ldots$

Every regex with concatenation, alternative, and subroutine call can be expressed by a context-free grammar

Conversion of a regex to an equivalent CFG

$$(?\mathrm{P}{>}N)\mathtt{a}(?(\mathrm{DEFINE})(?{<}N{>}\varepsilon \mid \mathtt{b}(?\mathrm{P}{>}N)\mathtt{b} \mid \mathtt{c}(?\mathrm{P}{>}N)\mathtt{c}))$$

- $\mathfrak{G}_\varepsilon = (\{S_\varepsilon\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_\varepsilon \to \varepsilon\}, S_\varepsilon\})$, $\mathfrak{G}_a = (\{S_\mathtt{a}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_\mathtt{a} \to \mathtt{a}\}, S_\mathtt{a})\ldots$
- $\mathfrak{G}_{(?\mathrm{P}>N)} = (\{S_{(?\mathrm{P}>N)}, N\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_{(?\mathrm{P}>N)} \to N\}, S_{(?\mathrm{P}>N)})$

Every regex with concatenation, alternative, and subroutine call can be expressed by a context-free grammar

Conversion of a regex to an equivalent CFG

$$(?\mathrm{P}{>}N)\mathtt{a}(?(\mathrm{DEFINE})(?{<}N{>}\varepsilon \mid \mathtt{b}(?\mathrm{P}{>}N)\mathtt{b} \mid \mathtt{c}(?\mathrm{P}{>}N)\mathtt{c}))$$

- $\mathfrak{G}_\varepsilon = (\{S_\varepsilon\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_\varepsilon \to \varepsilon\}, S_\varepsilon\})$, $\mathfrak{G}_a = (\{S_\mathtt{a}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_\mathtt{a} \to \mathtt{a}\}, S_\mathtt{a})\ldots$
- $\mathfrak{G}_{(?\mathrm{P}>N)} = (\{S_{(?\mathrm{P}>N)}, N\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_{(?\mathrm{P}>N)} \to N\}, S_{(?\mathrm{P}>N)})$
- after constructing grammars for elementary expressions. . .

Every regex with concatenation, alternative, and subroutine call can be expressed by a context-free grammar

Conversion of a regex to an equivalent CFG

$$(?\mathrm{P}{>}N)\mathtt{a}(?(\mathrm{DEFINE})(?{<}N{>}\varepsilon \mid \mathtt{b}(?\mathrm{P}{>}N)\mathtt{b} \mid \mathtt{c}(?\mathrm{P}{>}N)\mathtt{c}))$$

- $\mathfrak{G}_\varepsilon = (\{S_\varepsilon\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_\varepsilon \to \varepsilon\}, S_\varepsilon\})$, $\mathfrak{G}_a = (\{S_\mathtt{a}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_\mathtt{a} \to \mathtt{a}\}, S_\mathtt{a})\ldots$
- $\mathfrak{G}_{(?\mathrm{P}>N)} = (\{S_{(?\mathrm{P}>N)}, N\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_{(?\mathrm{P}>N)} \to N\}, S_{(?\mathrm{P}>N)})$
- after constructing grammars for elementary expressions...
- $\mathfrak{G}_{\mathtt{b}(?\mathrm{P}>N)} = (\mathcal{V}_\mathtt{b} \cup \mathcal{V}_{(?\mathrm{P}>N)} \cup \{S_{\mathtt{b}(?\mathrm{P}>N)}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \mathcal{R}_\mathtt{b} \cup \mathcal{R}_{(?\mathrm{P}>N)} \cup \{S_{\mathtt{b}(?\mathrm{P}>N)} \to S_\mathtt{b} S_{(?\mathrm{P}>N)}\}, S_{\mathtt{b}(?\mathrm{P}>N)}) : S_{\mathtt{b}(?\mathrm{P}>N)} \notin \mathcal{V}_\mathtt{b} \cup \mathcal{V}_{(?\mathrm{P}>N)}\ldots$

Every regex with concatenation, alternative, and subroutine call can be expressed by a context-free grammar

Conversion of a regex to an equivalent CFG

$$(?\mathrm{P}>N)\mathtt{a}(?(\mathrm{DEFINE})(?<N>\varepsilon \mid \mathtt{b}(?\mathrm{P}>N)\mathtt{b} \mid \mathtt{c}(?\mathrm{P}>N)\mathtt{c}))$$

- $\mathfrak{G}_\varepsilon = (\{S_\varepsilon\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_\varepsilon \to \varepsilon\}, S_\varepsilon\})$, $\mathfrak{G}_a = (\{S_\mathtt{a}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_\mathtt{a} \to \mathtt{a}\}, S_\mathtt{a})\ldots$
- $\mathfrak{G}_{(?\mathrm{P}>N)} = (\{S_{(?\mathrm{P}>N)}, N\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{S_{(?\mathrm{P}>N)} \to N\}, S_{(?\mathrm{P}>N)})$
- after constructing grammars for elementary expressions. . .
- $\mathfrak{G}_{\mathtt{b}(?\mathrm{P}>N)} = (\mathcal{V}_\mathtt{b} \cup \mathcal{V}_{(?\mathrm{P}>N)} \cup \{S_{\mathtt{b}(?\mathrm{P}>N)}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \mathcal{R}_\mathtt{b} \cup \mathcal{R}_{(?\mathrm{P}>N)} \cup \{S_{\mathtt{b}(?\mathrm{P}>N)} \to S_\mathtt{b} S_{(?\mathrm{P}>N)}\}, S_{\mathtt{b}(?\mathrm{P}>N)}) : S_{\mathtt{b}(?\mathrm{P}>N)} \notin \mathcal{V}_\mathtt{b} \cup \mathcal{V}_{(?\mathrm{P}>N)}\ldots$
- $\mathfrak{G}_{\varepsilon|\mathtt{b}(?\mathrm{P}>N)\mathtt{b}} = (\mathcal{V}_\varepsilon \cup \mathcal{V}_{\mathtt{b}(?\mathrm{P}>N)\mathtt{b}} \cup \{S_{\varepsilon|\mathtt{b}(?\mathrm{P}>N)\mathtt{b}}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \mathcal{R}_\varepsilon \cup \mathcal{R}_{\mathtt{b}(?\mathrm{P}>N)\mathtt{b}} \cup \{S_{\varepsilon|\mathtt{b}(?\mathrm{P}>N)\mathtt{b}} \to S_\varepsilon \mid S_{\mathtt{b}(?\mathrm{P}>N)\mathtt{b}}\}, S_{\varepsilon|\mathtt{b}(?\mathrm{P}>N)\mathtt{b}}) : S_{\varepsilon|\mathtt{b}(?\mathrm{P}>N)\mathtt{b}} \notin \mathcal{V}_\varepsilon \cup \mathcal{V}_{\mathtt{b}(?\mathrm{P}>N)\mathtt{b}}$

Every regex with concatenation, alternative, and subroutine call can be expressed by a context-free grammar

Conversion of a regex to an equivalent CFG

$$(?\mathrm{P}{>}N)\mathtt{a}(?(\mathrm{DEFINE})(?{<}N{>}\varepsilon \mid \mathtt{b}(?\mathrm{P}{>}N)\mathtt{b} \mid \mathtt{c}(?\mathrm{P}{>}N)\mathtt{c}))$$

- $\mathfrak{G}_\varepsilon = (\{S_\varepsilon\}, \{\mathtt{a},\mathtt{b},\mathtt{c}\}, \{S_\varepsilon \to \varepsilon\}, S_\varepsilon\})$, $\mathfrak{G}_a = (\{S_\mathtt{a}\}, \{\mathtt{a},\mathtt{b},\mathtt{c}\}, \{S_\mathtt{a} \to \mathtt{a}\}, S_\mathtt{a})\ldots$
- $\mathfrak{G}_{(?\mathrm{P}>N)} = (\{S_{(?\mathrm{P}>N)}, N\}, \{\mathtt{a},\mathtt{b},\mathtt{c}\}, \{S_{(?\mathrm{P}>N)} \to N\}, S_{(?\mathrm{P}>N)})$
- after constructing grammars for elementary expressions. . .
- $\mathfrak{G}_{\mathtt{b}(?\mathrm{P}>N)} = (\mathcal{V}_\mathtt{b} \cup \mathcal{V}_{(?\mathrm{P}>N)} \cup \{S_{\mathtt{b}(?\mathrm{P}>N)}\}, \{\mathtt{a},\mathtt{b},\mathtt{c}\}, \mathcal{R}_\mathtt{b} \cup \mathcal{R}_{(?\mathrm{P}>N)} \cup \{S_{\mathtt{b}(?\mathrm{P}>N)} \to S_\mathtt{b} S_{(?\mathrm{P}>N)}\}, S_{\mathtt{b}(?\mathrm{P}>N)}) : S_{\mathtt{b}(?\mathrm{P}>N)} \notin \mathcal{V}_\mathtt{b} \cup \mathcal{V}_{(?\mathrm{P}>N)}\ldots$
- $\mathfrak{G}_{\varepsilon|\mathtt{b}(?\mathrm{P}>N)\mathtt{b}} = (\mathcal{V}_\varepsilon \cup \mathcal{V}_{\mathtt{b}(?\mathrm{P}>N)\mathtt{b}} \cup \{S_{\varepsilon|\mathtt{b}(?\mathrm{P}>N)\mathtt{b}}\}, \{\mathtt{a},\mathtt{b},\mathtt{c}\}, \mathcal{R}_\varepsilon \cup \mathcal{R}_{\mathtt{b}(?\mathrm{P}>N)\mathtt{b}} \cup \{S_{\varepsilon|\mathtt{b}(?\mathrm{P}>N)\mathtt{b}} \to S_\varepsilon \mid S_{\mathtt{b}(?\mathrm{P}>N)\mathtt{b}}\}, S_{\varepsilon|\mathtt{b}(?\mathrm{P}>N)\mathtt{b}}) : S_{\varepsilon|\mathtt{b}(?\mathrm{P}>N)\mathtt{b}} \notin \mathcal{V}_\varepsilon \cup \mathcal{V}_{\mathtt{b}(?\mathrm{P}>N)\mathtt{b}}$
- . . .

Thus it is shown that subroutine calls match exactly context-free languages

Theorem

$\mathbb{L}_{\mathrm{E_S}} = \mathbb{L}_{\mathrm{CF}}$

Adding lookahead to a regex with subroutine calls extends it expressive power

Theorem

$\mathbb{L}_{\mathrm{E_S}} \subsetneq \mathbb{L}_{\mathrm{E_{LS}}}$

Proof.

$$\{\mathtt{a}^g\mathtt{b}^g\mathtt{c}^g : g \in \mathbb{N}\} =$$

$$\mathrm{L}((?{=}(?{<}N_1{>}\mathtt{a}(\varepsilon \mid (?\mathrm{P}{>}N_1))\mathtt{b})\,\mathtt{c})\mathtt{aa}^*(?{<}N_2{>}\mathtt{b}(\varepsilon \mid (?\mathrm{P}{>}N_2))\mathtt{c}))$$

□

Summary

- expressive power of expressions with concatenation, alternative, and subroutine call is equivalent to the class of context-free languages
  - algorithms to convert between regex and context-free grammar
- expressive power of expressions with concatentation, alternative, subroutine call, and lookahead is beyond context-free languages

Future work: expressive power of regex with both subroutine call and lookaround.