Regular Expressions and Non-Regular Languages

Chanathip Namprempre

Computer Science
Reed College

Outline

Regular expressions: Examples

Language	Regular Expression
All strings starting with a 0 or a 1 followed by any number of 0s.	$(0 \cup 1)0^*$
All possible strings of 0s and 1s.	$(0 \cup 1)^*$
All strings ending with 1.	Σ^*1
All strings that either start with a 0 or end with a 1.	$(0\Sigma^*) \cup (\Sigma^*1)$

Precedence: parentheses, star, concatenation, union

Regular expressions: Examples

"A variable in C begins with a letter followed by any number of letters, digits, and underscore."

$$\text{letter}(\text{letter} \cup \text{digit} \cup \_)^*$$

"A real number (in mathematics) is some number of digits, optionally followed by a decimal point and more digits."

$$\text{digit}^*(. \cup \varepsilon)\text{digit}^+$$

These things can be described more precisely using regular expressions.

Formal definition of a regular expression

Definition

We say that R is a regular expression if R is

1. a for some a in the alphabet Σ,
2. ε,
3. $\emptyset$,
4. $(R_1 \cup R_2)$ where R_1 and R_2 are regular expressions,
5. $(R_1 \circ R_2)$ where R_1 and R_2 are regular expressions, or
6. (R_1^*) where R_1 is a regular expression.

- This is an inductive definition, aka a recursive definition.
- The notation $L(R)$ denotes the language defined by the regular expression R.

More Examples

Regular expression	Language
0^*10^*	$\{w \mid w$ has exactly a single 1 $\}$
$\Sigma^*1\Sigma^*$	$\{w \mid w$ has at least one 1 $\}$
$\Sigma^*001\Sigma^*$	$\{w \mid w$ contains the string 001 as a substring $\}$
$(\Sigma\Sigma)^*$	$\{w \mid w$ is a string of even length $\}$
$(\Sigma\Sigma\Sigma)^*$	$\{w \mid w$ has length a multiple of 3 $\}$

More Examples

Regular expression	Language
$01 \cup 10$	$\{01, 10\}$
$0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$	$\{w \mid w$ starts and ends with the same symbol $\}$
$(0 \cup \varepsilon)1^*$	$01^* \cup 1^*$
$(0 \cup \varepsilon)(1 \cup \varepsilon)$	$\{\varepsilon, 0, 1, 01\}$
$1^*\emptyset$	$\emptyset$
$\emptyset^*$	$\{\varepsilon\}$

More example

Let D be the set $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Consider the following regular expression.

$$\{+, -, \varepsilon\}(\mathrm{DD}^* \cup \mathrm{DD}^*.\mathrm{D}^* \cup \mathrm{D}^*.\mathrm{DD}^*)$$

What language do you think this regular expression describes?

Regular expressions and finite automata are EQUIVALENT.

Theorem

A language is regular if and only if some regular expression describes it.

There are two directions that need to be proved:

[$\Longleftarrow$] If a language is described by a regular expression, then it is a regular language.

[$\Longrightarrow$] If a language is a regular language, then there is a regular expression describing it.

Theorem

A language is regular if and only if some regular expression describes it.

1. If a language is described by a regular expression, then it is a regular language.

   Proof idea: Given a regular expression, use it to construct an NFA recognizing the same language.

2. If a language is a regular language, then there is a regular expression describing it.

   Proof idea: Given a DFA recognizing the language, construct a regular expression from the DFA.

Theorem

A language is regular if and only if some regular expression describes it.

1. If a language is described by a regular expression, then it is a regular language.

   Proof idea: Given a regular expression, use it to construct an NFA recognizing the same language.

2. If a language is a regular language, then there is a regular expression describing it.

   Proof idea: Given a DFA recognizing the language, construct a regular expression from the DFA.

Theorem

A language is regular if and only if some regular expression describes it.

1. If a language is described by a regular expression, then it is a regular language.

   Proof idea: Given a regular expression, use it to construct an NFA recognizing the same language.

2. If a language is a regular language, then there is a regular expression describing it.

   Proof idea: Given a DFA recognizing the language, construct a regular expression from the DFA.

1) Converting Regular Expression to NFA

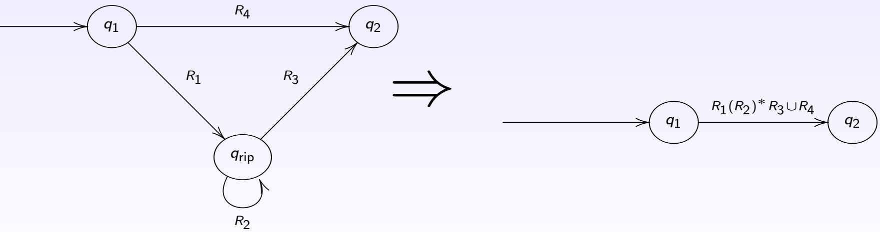
idea

Easy. Start with the recursive definition of regular expression. Construct an NFA for each case.

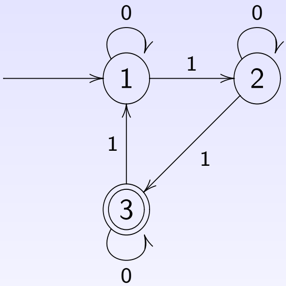
2) Converting DFA to Regular Expression

idea

1. Add a new start state S' and a new accept state F'. This gives us a GNFA.
2. Rip out one state (that isn't S' and F') at a time



2) Converting DFA to Regular Expression: Example



Try ripping in different orders, say, 1,2,3 and 2, 3, 1. Are the answers the same?

One gives you $0^*10^*1(0 \cup (10^*10^*1))^*$. The other gives you $(10^*10^*1 \cup 0)^*10^*10^*$.

Outline

Some languages are not regular! For example,

$$\{0^n1^n \mid n \geq 0\}$$

or the language

$$\{w \mid w \text{ has an equal number of 0s and 1s}\}$$

Intuitively, these language are problematic for FAs because they require infinite memory to count.

BUT the following language is regular!

$\{w \mid w$ has an equal number of occurrences of 01 and 10 as substrings$\}$

Q: How do we tell???

A: Use pumping lemma

Pumping Lemma: Intuition

Consider the following language:

$$L = \{0^n 1^n \mid n \geq 0\}$$

- Suppose towards a contradiction that L was regular.
- Suppose that L is recognized by a DFA D.
- Suppose that D has m states.
- Consider the string $w = 0^{m+1}1^{m+1}$.
- There must be a loop when D processes w.
- We can show that this means that D would accept a string $w' \notin L$.
- Thus, we have a contradiction. So L is not regular.

Pumping Lemma

Theorem (Pumping Lemma)

If L *is a regular language,* then
there is *a number* $p \geq 0$
so that for all $s \in L$ *with* $|s| \geq p$
there is *a parse of* $s = xyz$ *with* $|y| \geq 1$ *and* $|xy| \leq p$
such that for any $i \geq 0, xy^i z \in L$

We can use the Pumping Lemma to prove languages not regular.

Theorem (Contrapositive of Pumping Lemma)

If for any *number* $p \geq 0$
there exists *a string* $s \in L$ *with* $|s| \geq p$
so that for any *parse of* $s = xyz$ *with* $|y| \geq 1$ *and* $|xy| \leq p$
there exists *some* $i \geq 0$ *such that* $xy^i z \notin L$,
then L *is not regular.*

Examples

We can use the contrapositive of the Pumping Lemma to prove these languages non-regular.

1. $C = \{w \mid w \text{ has an equal number of 0s and 1s}\}$.
2. $F = \{ww \mid w \in \{0,1\}^*\}$
3. $E = \{0^i 1^j \mid i > j\}$

Notice that we cannot use the Pumping Lemma (or its contrapositive) to prove that a language is regular.