THEORY OF AUTOMATA & FORMAL LANGUAGES

Operations on Languages

Combine, compare and transform sets of strings

Union → Intersection → Complement → Concatenation

Union of languages

Collect all strings that belong to either language.

Definition and input

$A = \{a, ab, b\}$

$B = \{b, ba, \varepsilon\}$

$A \cup B = \{w : w \in A \text{ or } w \in B\}$

Start with all members of A.

Add only new members of B.

Worked result

$A \cup B = \{a, ab, b, ba, \varepsilon\}$

b occurs in both; list it once.

ε belongs because $\varepsilon \in B$.

$A \cup B = B \cup A$

$A \cup \varnothing = A$

Union means OR, including strings belonging to both languages.

Intersection of languages

Keep only strings that occur in both languages.

Definition and input

$A = \{a, ab, b\}$

$B = \{b, ba, \varepsilon\}$

$A \cap B = \{w : w \in A \text{ and } w \in B\}$

Check a: not in B.

Check ab: not in B.

Worked result

Check b: present in both.

$A \cap B = \{b\}$

$A \cap B = B \cap A$

$A \cap \varnothing = \varnothing$

$A \cap A = A$

Intersection means AND: both membership tests must succeed.

Difference of languages

Remove from the first language every string also found in the second.

Compute A − B

$A = \{a, ab, b\}$

$B = \{b, ba, \varepsilon\}$

$A - B = \{w : w \in A \text{ and } w \notin B\}$

Remove b from A.

$A - B = \{a, ab\}$

Reverse the order

$B - A = \{ba, \varepsilon\}$

Remove b from B.

$A - B \neq B - A$ in this example.

$A - A = \emptyset$

$A - \emptyset = A$

Difference is directional: always identify which language comes first.

Complement of a language

Complement includes all strings over the chosen alphabet that are not in L.

Definition and finite preview

$\Sigma = \{a, b\}$; $L = \{a, ab\}$

$L^c = \Sigma^* - L$

Length 0: ε is included.

Length 1: b is included.

Length 2: aa, ba, bb are included.

Interpret the result

$L^c = \{\varepsilon, b, aa, ba, bb, \ldots\}$

Every string of length ≥ 3 is included.

a and ab are excluded.

$L \cup L^c = \Sigma^*$

$L \cap L^c = \varnothing$

Complement depends on the alphabet; the displayed list is only a preview.

Concatenation of languages

Join each string in the first language to each string in the second.

Compute AB

$A = \{a, ab\}$; $B = \{b, \varepsilon\}$

$AB = \{xy : x \in A, y \in B\}$

$a \cdot b = ab$; $a \cdot \varepsilon = a$

$ab \cdot b = abb$

$ab \cdot \varepsilon = ab$

Collect distinct results

$AB = \{a, ab, abb\}$

$BA = \{a, ab, ba, bab\}$

$AB \neq BA$ in this example.

$A\{\varepsilon\} = A$; $A\emptyset = \emptyset$

Duplicate results appear only once.

Concatenation joins strings in order; it is not set union.

Reversal of a language

Reverse each member independently, then collect the results.

Definition and example

$L = \{ab, aab, b, \varepsilon\}$

$L^R = \{w^R : w \in L\}$

$(ab)^R = ba$

$(aab)^R = baa$

$b^R = b$; $\varepsilon^R = \varepsilon$

Worked result and rules

$L^R = \{ba, baa, b, \varepsilon\}$

$(L^R)^R = L$

$(AB)^R = B^R A^R$

Reversal preserves string lengths.

It does not reverse the order of a set.

Reverse the symbols inside each string, not the listing of set members.

Powers of a language

Use concatenation repeatedly, counting whole strings as blocks.

Compute L^2

$L = \{a, b\}$

$L^0 = \{\varepsilon\}$; $L^1 = L$

$L^2 = LL$

$a \cdot a = aa$; $a \cdot b = ab$

$b \cdot a = ba$; $b \cdot b = bb$

Results and extension

$L^2 = \{aa, ab, ba, bb\}$

$L^3 = L^2L$ (eight strings here).

$L^{n+1} = L^nL$

For $L = \{a, aa\}$, $L^2 = \{aa, aaa, aaaa\}$

Different block choices can duplicate results.

A power counts blocks; the result is a set, so duplicates disappear.

Practice and explained answers

Use $A = \{a, ab\}$, $B = \{ab, b\}$ and $\Sigma = \{a, b\}$.

Try these questions

1. Find $A \cup B$.

2. Find $A \cap B$ and $A - B$.

3. Find AB.

4. Find A^R.

5. Does ε belong to A^c?

Answers and reasoning

1. $\{a, ab, b\}$

2. $\{ab\}$; $\{a\}$

3. $\{aab, ab, abab, abb\}$

4. $\{a, ba\}$

5. Yes: $\varepsilon \in \Sigma^*$ but $\varepsilon \notin A$.

Next: describe languages mathematically using conditions and set-builder notation.