Finite-state automata

is a new relation, defined as follows:

The reflexive transitive extension of the transition relation $R$

Example: Finite-state automaton

Finite-state automata

and alphabet symbols to states.

A finite-state automaton is a five-tuple $(Q, \Sigma, \delta, q_0, F)$, where

Finite-state automata are models of computation. They compute

 Languages.
Finite-state automata

Example: Some finite-state automata

Final language process

Example: Some finite-state automata

Final language process
Finite-state automata

Example: Some finite-state automata

\[ q_0 \overset{a}{\rightarrow} q_1 \]

\[ \{a, aa, aaa, aaaa, \ldots\} \]
Operations on Finite-State Automata

Finite-State Automata

Example: Finite-state automata and regular expressions

Finite-State Automata

Example: Automata with "c" moves

Finite-State Automata

Theorem (Kleene, 1956): The class of languages recognized by finite-state automata is the class of regular languages.
Regular Relations

A naïve morphological analyzer:

Applications of Finite-State Automata in Language Processing

Minimization and Determinization

Lexicon:

Example: Equivalent automata
Finite-state transducers

Regular Relations

Morphological analysis:

Regular Relations
Given a transducer \( \mathcal{T} \), its lower automaton is \( (Q', q_0', \delta', \gamma') \), where \( (Q, q_0, \delta, \gamma) \) is \( \mathcal{T} \).

\[
\text{Example: English-to-French}
\]

\[
\mathcal{T} : A \rightarrow B, c : C
\]

\[
\text{Example: The uppercase transducer}
\]

\[
\mathcal{T}_U : A \rightarrow B, c : C
\]

**Properties of Finite-state Transducers**

The language of a finite-state transducer is a language of pairs: a binary relation over \( L \times L \). The language of a finite-state transducer is defined analogously to how the language of an automaton is defined.

**Finite-state Transducers**
Properties of Finite-state Transducers

Weights
Closed under composition
Not closed under intersection (and hence complementation)
Closed under concatenation, Kleene-star, union
Denote regular relations
Computationally efficient

Transducers with no ε-moves are closed under intersection.

\[
\{ w Q w D \} = (\varepsilon) L \cup L
\]
\[
\{ 0 \leq w \mid w Q w D \} = (\varepsilon) L
\]
\[
\{ 0 \leq w \mid w Q w D \} = (\varepsilon) L
\]

Transducers are not closed under intersection.

Properties of Finite-state Transducers

\[
|\varepsilon L \times \varepsilon L|
\]

The number of states in the composition transducer might be

\[
((m) L) L = (m) L
\]

m \in L

Transducers, there exists a transducer L such that for every m \in L

Transducers are closed under composition: if L and L are

Transducers, there exists a transducer L such that for every m \in L

A transducer L is functional if for every m \in L

Transducers are not closed under intersection.