Computación y Estructuras Discretas III

Andrés A. Aristizábal P.
aaaristizabal@icesi.edu.co
Ángela Villota
apvillota@icesi.edu.co

Departamento de Computación y Sistemas Inteligentes

2024-2

Agenda del día

1. **Regular languages and Automata theory**
   - Defining a Finite State Transducer
   - Regular Relations
   - Language of an FST
   - Exercises

1. **Regular languages and Automata theory**
   - Defining a Finite State Transducer
   - Regular Relations
   - Language of an FST
   - Exercises

Defining a Finite State Transducer

What are Finite State Transducers?

Defining a Finite State Transducer

What are Finite State Transducers?

- Automata with output.

Defining a Finite State Transducer

What are Finite State Transducers?

- Automata with output.
- Involves two alphabets Σ (input) and Γ (output).

Defining a Finite State Transducer

What are Finite State Transducers?

- Automata with output.
- Involves two alphabets Σ (input) and Γ (output).
- Edges have the form $u : v$ where $u \in \Sigma$ and $v \in \Gamma$ and mean replace u with v.

Defining a Finite State Transducer

What are Finite State Transducers?

- Automata with output.
- Involves two alphabets Σ (input) and Γ (output).
- Edges have the form $u : v$ where $u \in \Sigma$ and $v \in \Gamma$ and mean replace u with v.
- Finite transducers define not sets but relations between inputs and outputs.
- Automata can also be treated as transducers (that output exactly their input for the words accepted by automaton).

Defining a Finite State Transducer

What are Finite State Transducers?

- Automata with output.
- Involves two alphabets Σ (input) and Γ (output).
- Edges have the form $u : v$ where $u \in \Sigma$ and $v \in \Gamma$ and mean replace u with v.
- Finite transducers define not sets but relations between inputs and outputs.
- Automata can also be treated as transducers (that output exactly their input for the words accepted by automaton).
- We can treat finite transductions as sets of word pairs.

Defining a Finite State Transducer

How can we define a Deterministic Finite State Transducer (FST)?

Defining a Finite State Transducer

How can we define a Deterministic Finite State Transducer (FST)?

We define a Deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

Defining a Finite State Transducer

How can we define a Deterministic Finite State Transducer (FST)?

We define a Deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.

Defining a Finite State Transducer

How can we define a Deterministic Finite State Transducer (FST)?

We define a Deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.
2. Σ is the finite input alphabet.

Defining a Finite State Transducer

How can we define a Deterministic Finite State Transducer (FST)?

We define a Deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.
2. Σ is the finite input alphabet.
3. Γ is the finite output alphabet.

Defining a Finite State Transducer

How can we define a Deterministic Finite State Transducer (FST)?

We define a Deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.
2. Σ is the finite input alphabet.
3. Γ is the finite output alphabet.
4. $\delta : Q \times \Sigma \cup \{\lambda\} \longrightarrow Q$ is the transition function.

Defining a Finite State Transducer

How can we define a Deterministic Finite State Transducer (FST)?

We define a Deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.
2. Σ is the finite input alphabet.
3. Γ is the finite output alphabet.
4. $\delta : Q \times \Sigma \cup \{\lambda\} \longrightarrow Q$ is the transition function.
5. $\omega : Q \times \Sigma \cup \{\lambda\} \longrightarrow \Gamma$ is the output function.

Defining a Finite State Transducer

How can we define a Deterministic Finite State Transducer (FST)?

We define a Deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.
2. Σ is the finite input alphabet.
3. Γ is the finite output alphabet.
4. $\delta : Q \times \Sigma \cup \{\lambda\} \longrightarrow Q$ is the transition function.
5. $\omega : Q \times \Sigma \cup \{\lambda\} \longrightarrow \Gamma$ is the output function.
6. $q_0 \in Q$ is the start state.

Defining a Finite State Transducer

How can we define a Deterministic Finite State Transducer (FST)?

We define a Deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.
2. Σ is the finite input alphabet.
3. Γ is the finite output alphabet.
4. $\delta : Q \times \Sigma \cup \{\lambda\} \longrightarrow Q$ is the transition function.
5. $\omega : Q \times \Sigma \cup \{\lambda\} \longrightarrow \Gamma$ is the output function.
6. $q_0 \in Q$ is the start state.
7. $\emptyset \neq F \subseteq Q$ is the set of accepting states.

Defining a Finite State Transducer

How can we define a Non-deterministic Finite State Transducer (FST)?

Defining a Finite State Transducer

How can we define a Non-deterministic Finite State Transducer (FST)?

We define a Non-deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

Defining a Finite State Transducer

How can we define a Non-deterministic Finite State Transducer (FST)?

We define a Non-deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.

Defining a Finite State Transducer

How can we define a Non-deterministic Finite State Transducer (FST)?

We define a Non-deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.
2. Σ is the finite input alphabet.

Defining a Finite State Transducer

How can we define a Non-deterministic Finite State Transducer (FST)?

We define a Non-deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.
2. Σ is the finite input alphabet.
3. Γ is the finite output alphabet.

Defining a Finite State Transducer

How can we define a Non-deterministic Finite State Transducer (FST)?

We define a Non-deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.
2. Σ is the finite input alphabet.
3. Γ is the finite output alphabet.
4. $\delta : Q \times \Sigma \cup \{\lambda\} \longrightarrow \wp(Q)$ is the transition function.

Defining a Finite State Transducer

How can we define a Non-deterministic Finite State Transducer (FST)?

We define a Non-deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.
2. Σ is the finite input alphabet.
3. Γ is the finite output alphabet.
4. $\delta : Q \times \Sigma \cup \{\lambda\} \longrightarrow \wp(Q)$ is the transition function.
5. $\omega : Q \times \Sigma \cup \{\lambda\} \longrightarrow \Gamma \cup \{\lambda\}$ is the output function.

Defining a Finite State Transducer

How can we define a Non-deterministic Finite State Transducer (FST)?

We define a Non-deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.
2. Σ is the finite input alphabet.
3. Γ is the finite output alphabet.
4. $\delta : Q \times \Sigma \cup \{\lambda\} \longrightarrow \wp(Q)$ is the transition function.
5. $\omega : Q \times \Sigma \cup \{\lambda\} \longrightarrow \Gamma \cup \{\lambda\}$ is the output function.
6. $q_0 \in Q$ is the start state.

Defining a Finite State Transducer

How can we define a Non-deterministic Finite State Transducer (FST)?

We define a Non-deterministic FST as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ where:

1. Q is the finite set of states.
2. Σ is the finite input alphabet.
3. Γ is the finite output alphabet.
4. $\delta : Q \times \Sigma \cup \{\lambda\} \longrightarrow \wp(Q)$ is the transition function.
5. $\omega : Q \times \Sigma \cup \{\lambda\} \longrightarrow \Gamma \cup \{\lambda\}$ is the output function.
6. $q_0 \in Q$ is the start state.
7. $\emptyset \neq F \subseteq Q$ is the set of accepting states.

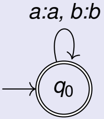
Defining a Finite State Transducer

What about this FST over the alphabet $\Sigma = \{a, b\}$?

Defining a Finite State Transducer

What about this FST over the alphabet $\Sigma = \{a, b\}$?

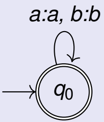
Example



Defining a Finite State Transducer

What about this FST over the alphabet $\Sigma = \{a, b\}$?

Example



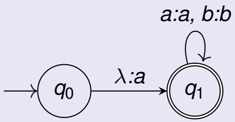
The simplest transducer, the identity relation.

Defining a Finite State Transducer

What about this FST over the alphabet $\Sigma = \{a, b\}$?

What about this FST over the alphabet $\Sigma = \{a, b\}$?

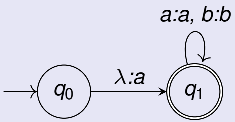
Example



Defining a Finite State Transducer

What about this FST over the alphabet $\Sigma = \{a, b\}$?

Example



Adds a to the beginning.

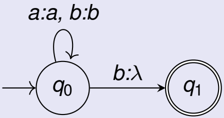
Defining a Finite State Transducer

What about this FST over the alphabet $\Sigma = \{a, b\}$?

Defining a Finite State Transducer

What about this FST over the alphabet $\Sigma = \{a, b\}$?

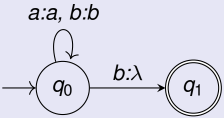
Example



Defining a Finite State Transducer

What about this FST over the alphabet $\Sigma = \{a, b\}$?

Example



Removes final b if it is present and rejects other words.

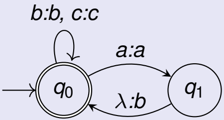
Defining a Finite State Transducer

What about this FST over the alphabet $\Sigma = \{a, b, c\}$?

Defining a Finite State Transducer

What about this FST over the alphabet $\Sigma = \{a, b, c\}$?

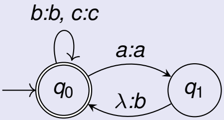
Example



Defining a Finite State Transducer

What about this FST over the alphabet $\Sigma = \{a, b, c\}$?

Example



Adds b after every a.

Defining a Finite State Transducer

What about FST's actions?

Defining a Finite State Transducer

What about FST's actions?

- The action of a Finite State Transducer can be viewed as computing a relation between two sets.

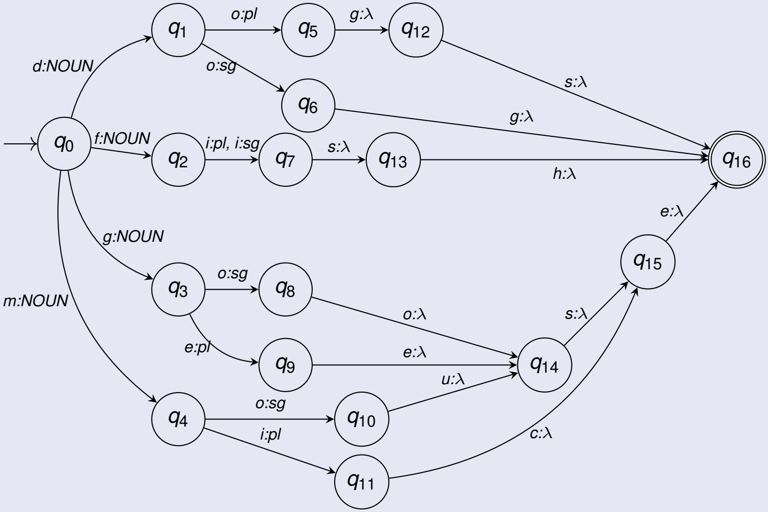
Defining a Finite State Transducer

What about this FST?

Defining a Finite State Transducer

What about this FST?

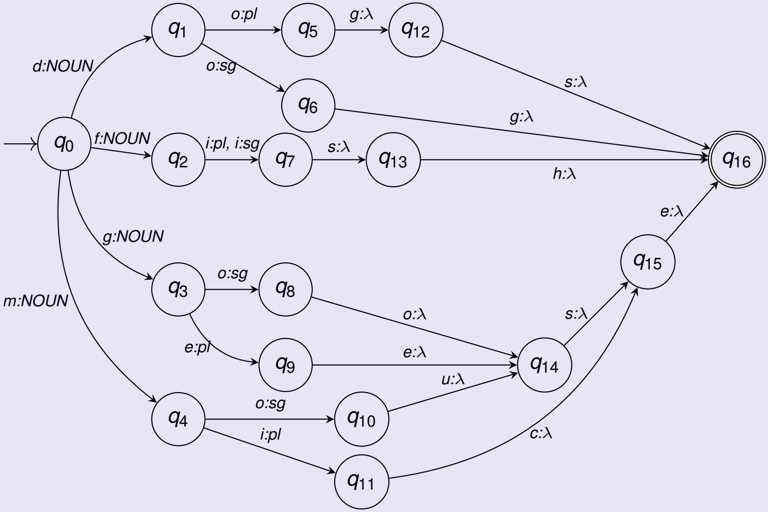
Example



Defining a Finite State Transducer

What about this FST?

Example



Maps words to morphological features.

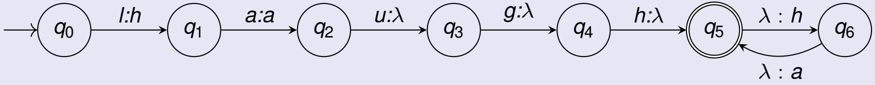
Defining a Finite State Transducer

What about this FST?

Defining a Finite State Transducer

What about this FST?

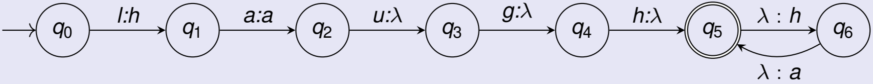
Example



Defining a Finite State Transducer

What about this FST?

Example



The laughing machine.

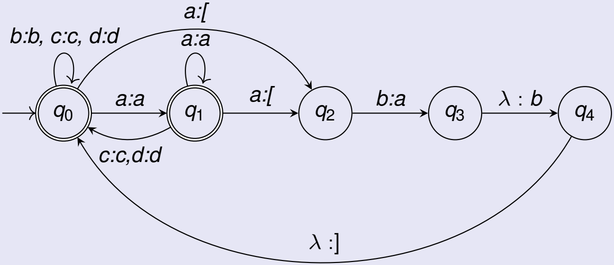
Defining a Finite State Transducer

What about this FST?

Defining a Finite State Transducer

What about this FST?

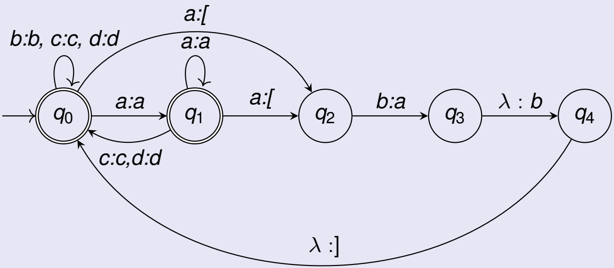
Example



Defining a Finite State Transducer

What about this FST?

Example



The bracketing machine. Every occurrence of ab is enclosed within brackets.

Agenda del día

1. **Regular languages and Automata theory**
   - Defining a Finite State Transducer
   - Regular Relations
   - Language of an FST
   - Exercises

Regular Relations

What is a relation?

Regular Relations

What is a relation?

A relation over the sets A and B is a subset of the Cartesian product, $A \times B$.
Ex: Let $A = \{dog, cat, cow\}$, $B = \{seven, \pi, octopus\}$.
Then $R = \{(dog, seven), (cow, \pi), (cow, octopus)\}$ is a relation.

Regular Relations

What is a relation?

A relation over the sets A and B is a subset of the Cartesian product, $A \times B$.
Ex: Let $A = \{dog, cat, cow\}$, $B = \{seven, \pi, octopus\}$.
Then $R = \{(dog, seven), (cow, \pi), (cow, octopus)\}$ is a relation.

What is a regular relation?

Regular Relations

What is a relation?

A relation over the sets A and B is a subset of the Cartesian product, $A \times B$.
Ex: Let $A = \{dog, cat, cow\}$, $B = \{seven, \pi, octopus\}$.
Then $R = \{(dog, seven), (cow, \pi), (cow, octopus)\}$ is a relation.

What is a regular relation?

A regular (or rational) relation over the alphabets Σ, Γ is formed from a finite combination of the following rules:

1. $\forall (x, y) \in \Sigma \cup \{\lambda\} \times \Gamma \cup \{\lambda\}$
2. $\emptyset$ is a regular relation
3. If R, S are regular relations, then so are $R \cdot S$, $R \cup S$, and R^*

Agenda del día

1. **Regular languages and Automata theory**
   - Defining a Finite State Transducer
   - Regular Relations
   - Language of an FST
   - Exercises

Language of an FST

What is the language of a FST?

Language of an FST

What is the language of a FST?

- The language $L(T)$ of a non-deterministic FST $T = (Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ is defined using the extended transition and output functions δ^*, ω^*.
- $L(T) = \{(u, v) \mid \delta^*(q_0, u) \cap F \neq \emptyset \wedge v \in \omega^*(q_0, u)\}$

1. **Regular languages and Automata theory**
   - Defining a Finite State Transducer
   - Regular Relations
   - Language of an FST
   - Exercises