UNIVERSIDAD ADOLFO IBÁÑEZ

Procesamiento de Lenguaje Natural

DIPLOMA/MAGISTER EN INTELIGENCIA ARTIFICIAL
UNIVERSIDAD ADOLFO IBÁÑEZ

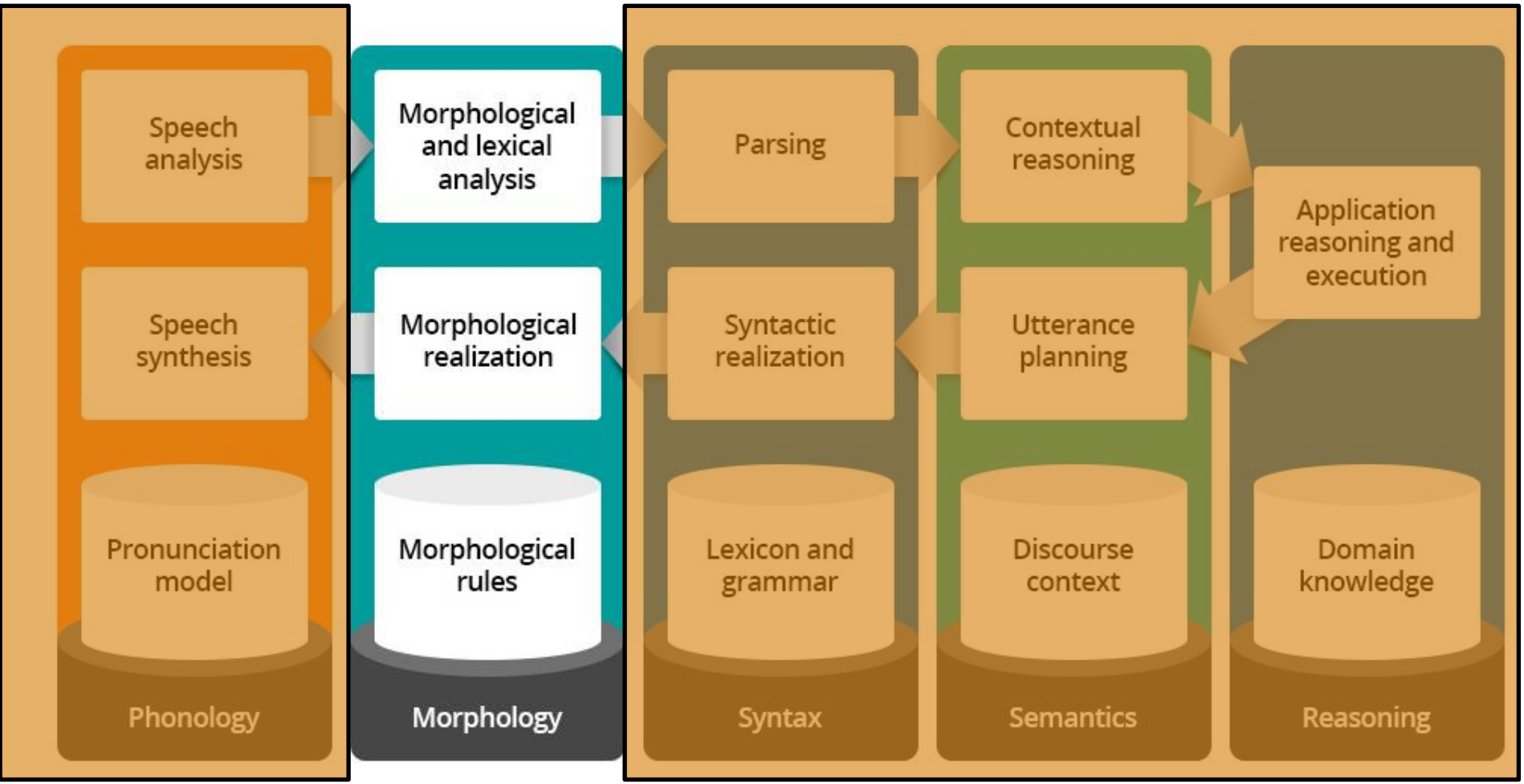
Profesor: Dr. John Atkinson

Análisis Morfológico

OBJETIVO

Entender la manera en que se forman las palabras a partir de sus unidades más básicas, una vez que se han identificado los *fonemas* básicos.

ETAPAS EN NLP



Morfología Computacional

Estudio de las formas (o partes) que se construyen las palabras a partir de unidades significativas más pequeñas denominadas *morfemas*.

Morfología Computacional

El significado del núcleo de las unidades (**stem**)

Piezas que se adhieren a los "troncos" para cambiar sus funciones gramaticales (***inflexión***)

Morfema = Tronco + Afijo

Ejemplo:

Empresas	Empresa **+ s**
Acéfalo	**A +** céfalo
Depositaron	Deposit **+ aron**

Tipos de Inflexión (1)

Stemming

Proceso de reducción de la inflexion en palabras a su forma raiz, incluso en un ***stem*** (tronco) no válido en el lenguaje

Form	Suffix	Stem
studies	-es	studi
studying	-ing	study
niñas	-as	niñ
niñez	-ez	niñ

Tipos de Inflexión (2)

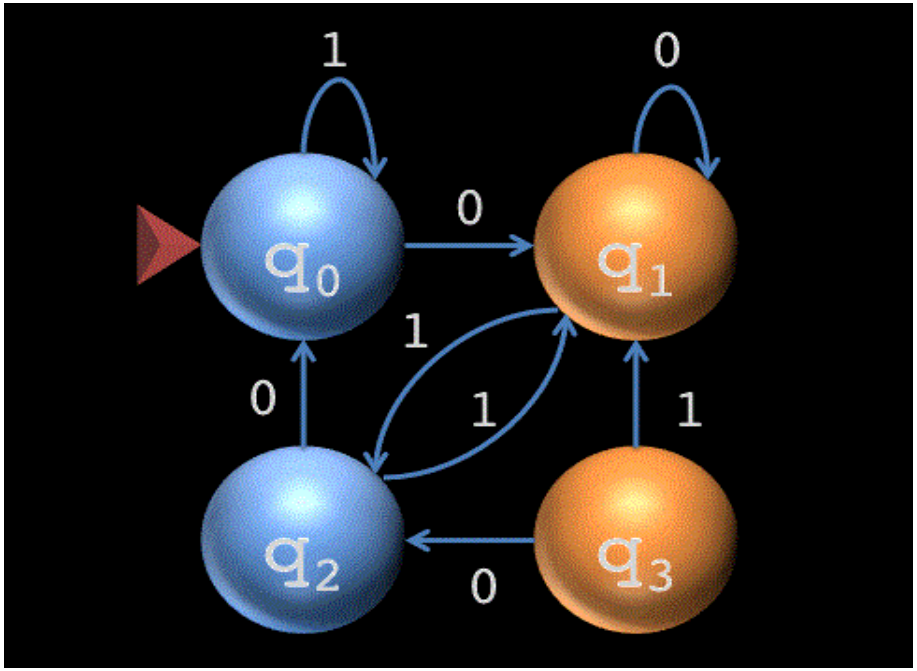
Lematización

Proceso que toma en consideración las reglas morfológicas y diccionario para generar el ***lema*** válido de una palabra.

Form	Morphological information	Lemma
studies	Third person, singular number, present tense of the verb study	study
studying	Gerund of the verb study	study
niñas	Feminine gender, plural number of the noun niño	niño
niñez	Singular number of the noun niñez	niñez

¿Cómo realizamos *Análisis Morfológico*?

Podemos utilizar modelos matemáticos conocidos como Autómatas de Estados Finitos (FA).



Autómatas (de Estados) Finitos

- Un FA es una *máquina* abstracta que puede utilizarse para computar funciones y para reconocer cierto tipo de lenguajes.

- Para una cierta entrada (*string de caracteres*), un FA determina si esta pertenece o no, a un cierto lenguaje previamente definido.

Muchas aplicaciones

- *Interacciones de un chatbot*
- *Búsqueda de patrones en un texto*
- *Diseño de lenguajes de programación*
- *Reconocimiento de lenguajes*
- *Verificación de circuitos*
- *Economía y teoría de juegos*
- *Biología*
- *Protocolos de comunicación*
- *etc*

Autómatas (de Estados) Finitos

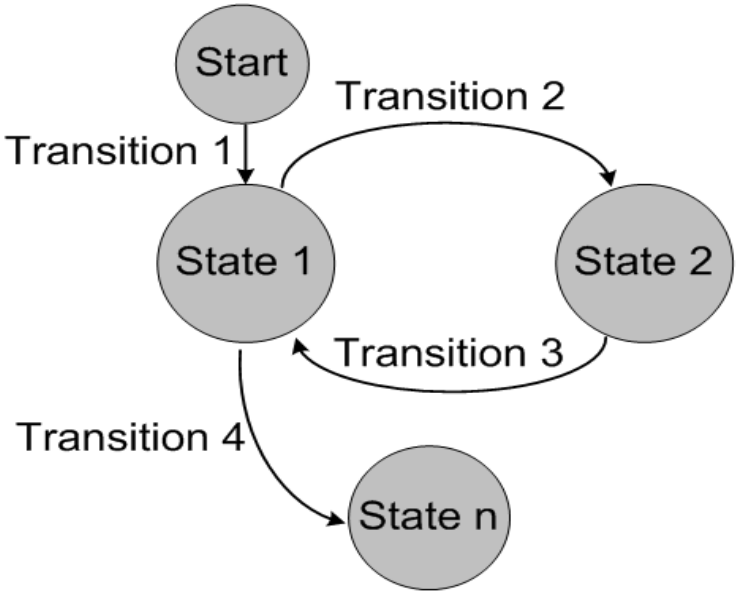
Un *autómata finito* A es una quíntupla:

$$A = (Q, \Sigma, \delta, q_0, F)$$

Q es un conjunto finito de estados en los que puede estar la máquina A
Σ es un alfabeto finito del lenguaje (*vocabulario*)
δ es una *función de transición* $(q,a) \rightarrow p$ con q,p estados y a el símbolo actual de la entrada.
$q_0 \in Q$ es el estado de comienzo
$F \subseteq Q$ es un conjunto de estados *finales* (de aceptación).

Función de Transición

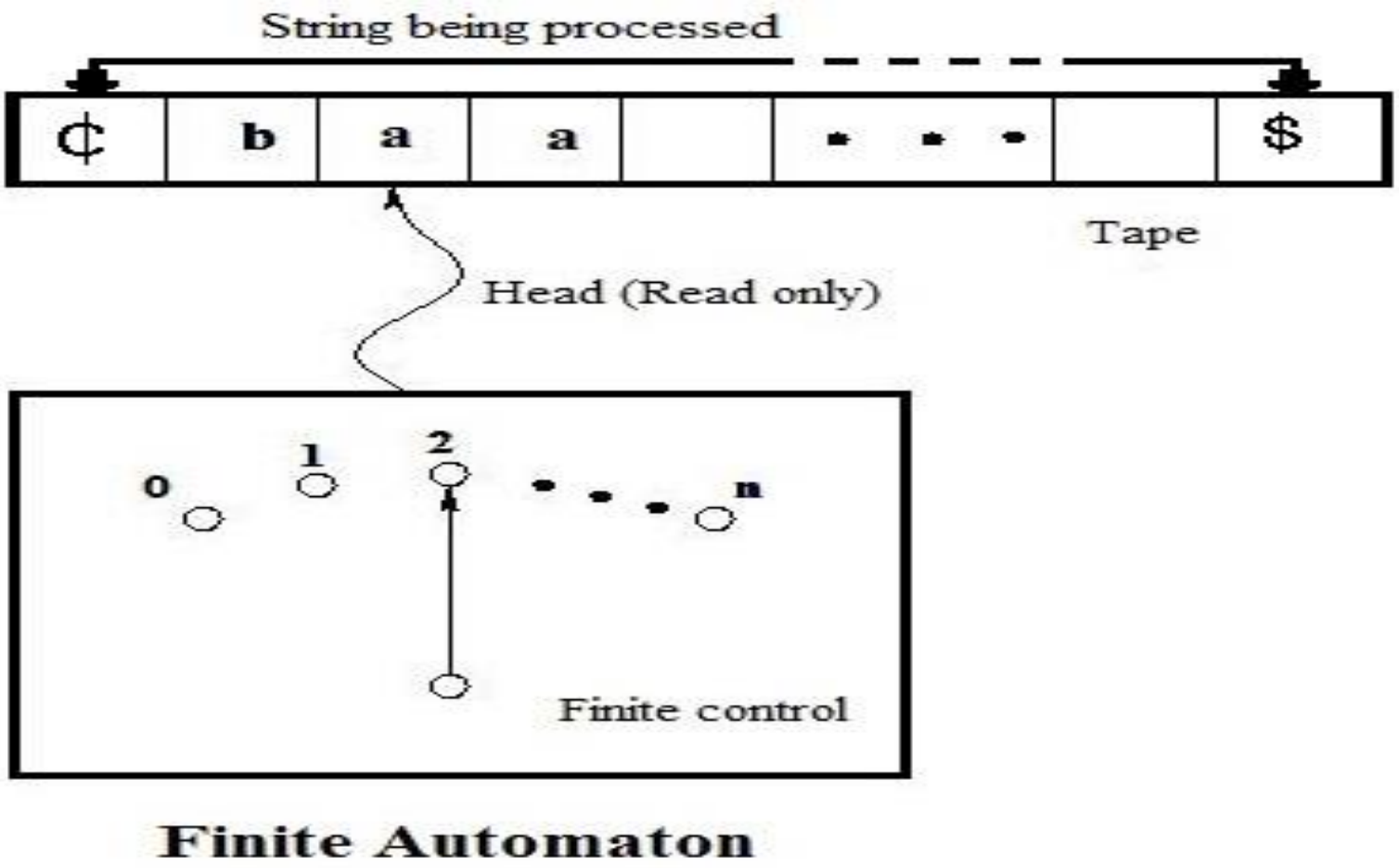
Representación como diagrama de estados



Representación como *tabla de transición*

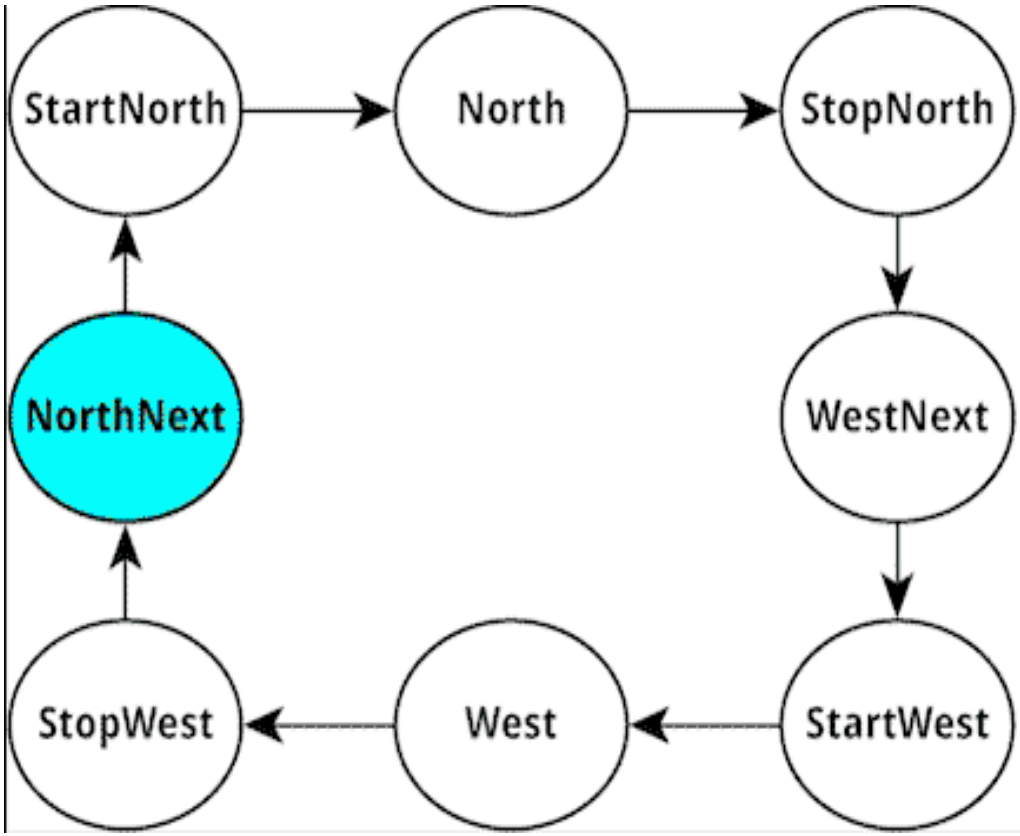
Current state / Input	State A	State B	State C
Input X	…	…	…
Input Y	…	State C	…
Input Z	…	…	…

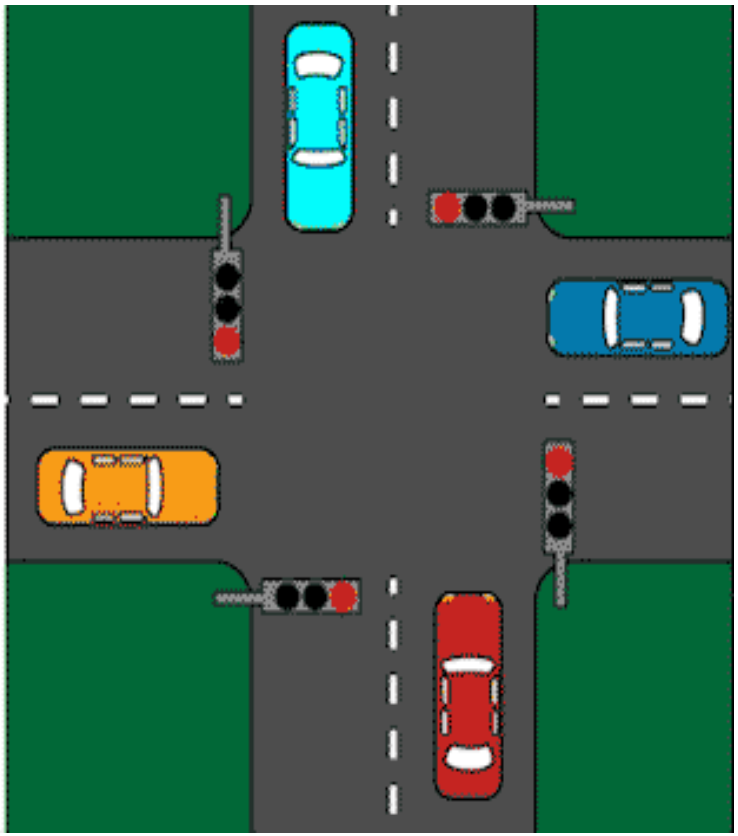
¿Cómo funciona un FA?



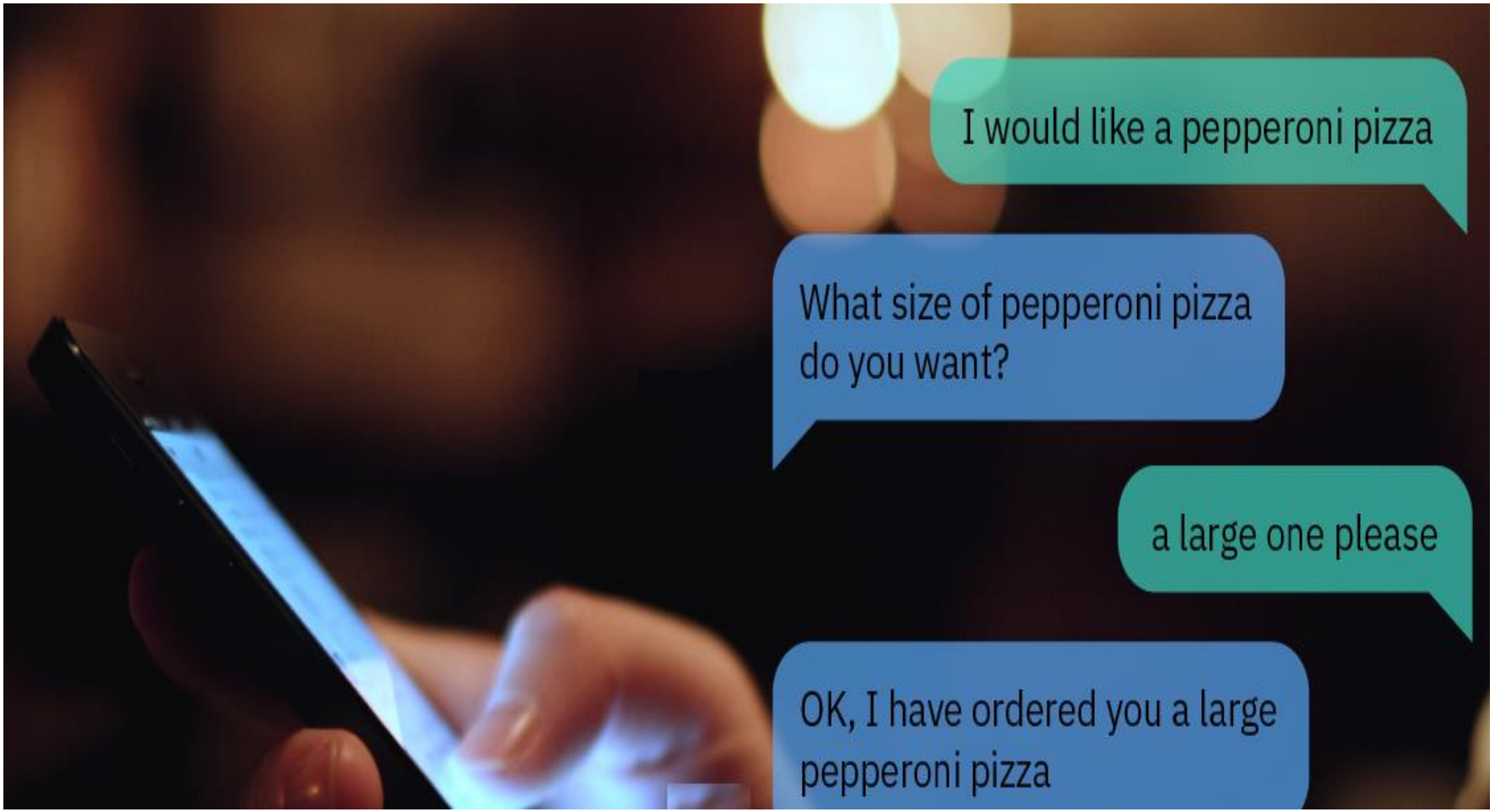
¿Qué son en realidad los *Estados* y *Transiciones*?

Coordinar tránsito vehicular..



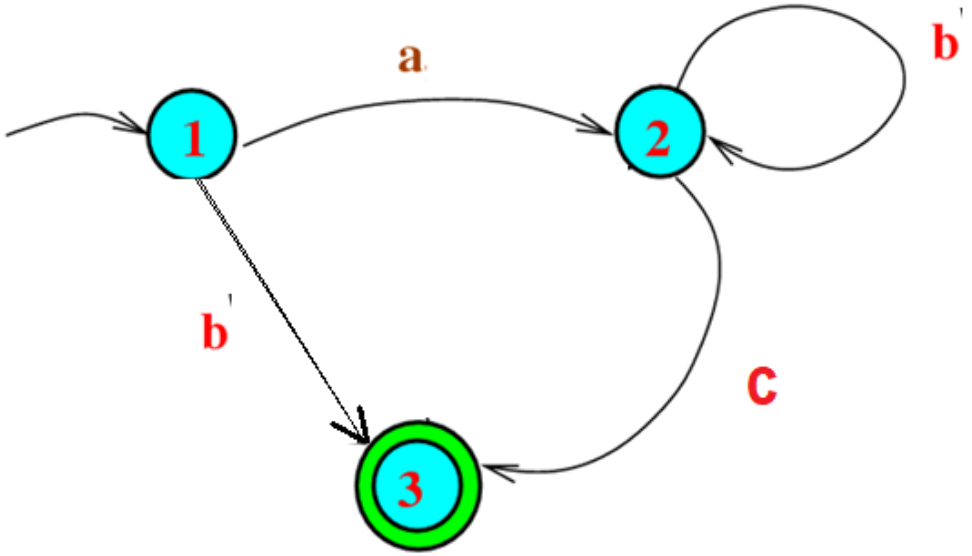


Solicitar pizza por teléfono…



EJEMPLO

Un FA que reconoce un cierto lenguaje:



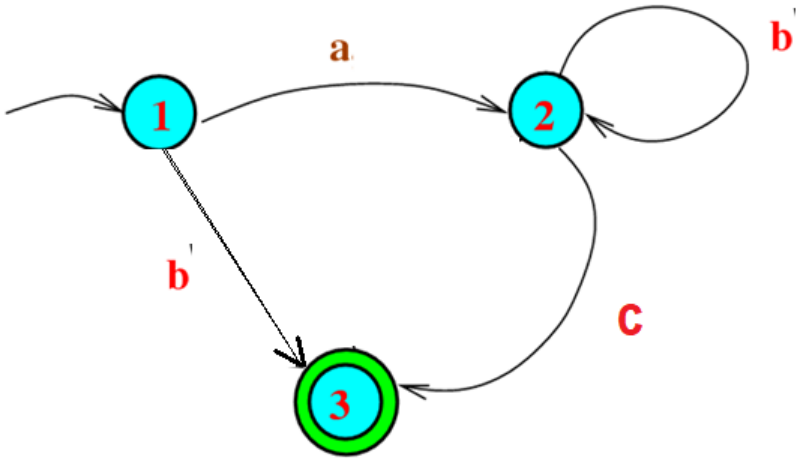
Q

	Σ a	b	c
1	2	3	ø
2	ø	2	3
3	ø	ø	ø

$Q_0 = 1$ $F = \{3\}$

EJEMPLO

¿Qué lenguaje reconoce la máquina (FA) previa?



- "a concatenado con varias repeticiones de b y concatenado con c"

o bien (unión)

- "una a"

Lenguajes Regulares

- El lenguaje reconocido por un FA se denomina un ***lenguaje regular*** o tipo 3 en la jerarquía de *Chomsky*.
- Un ***lenguaje regular*** se forma combinando símbolos en base a tres operaciones posibles: **concatenación, repetición, unión**.
- El algoritmo para reconocer un ***lenguaje regular*** es muy simple (sólo debe recorrer una tabla de transición).

Jerarquía de *Chomsky*

Gramática	Lenguaje	Modelo
Tipo 0: Irrestricta	Recursivamente enumerable (Nivel Pragmático)	Máquina de Turing (MT)
Tipo 1: Dependiente del Contexto	Dependiente del Contexto (Nivel Semántico)	Autómata Linealmente Limitado (ALL)
Tipo 2: Independiente del Contexto	Independiente del Contexto (Nivel Sintáctico)	Autómata de Pila (AP)
Tipo 3: Regular	Regular (Nivel Léxico)	Autómata Finito (AF)

Morfología para *Lenguajes Regulares*

- Podemos utilizar FA para implementar *analizadores morfológicos*.
- Sólo sirven para reconocer *lenguajes regulares*.
- Los términos *regular* e *irregular* se utilizan para referirse a las palabras que siguen las reglas y aquellas que no.

Morfología para *Lenguajes Regulares*

Ejemplos:

- Regulares: marcadores para plural de nombres (nouns): *cliente/clientes*.
- Irregulares: marcadores no son tan triviales para categorías como verbos irregulares: *fue/irá, write/wrote, go/went, was/were*.

EJEMPLO

Veamos las variaciones morfológicas posibles para: **computar**

Computador → computarizar → computarización

Computación → computacional

Computador → computarizar → computarizable

...

¿Porqué es importante?

- ✓ *Stemming* en recuperación de información
  - Uno podría buscar (Google) “ir a casa” y encontrar páginas con “fue a casa” e “irá a casa”
- ✓ *Morfología en Traducción*
  - Se requiere saber que la palabra *quiero* y *quieres* estan relacionadas a *querer*
- ✓ *Morfología en corrección ortográfica* (*MS Word*)
  - Se requiere saber que *señra* y *monopoliamente* no son palabras aunque esten construidas de partes de palabras.

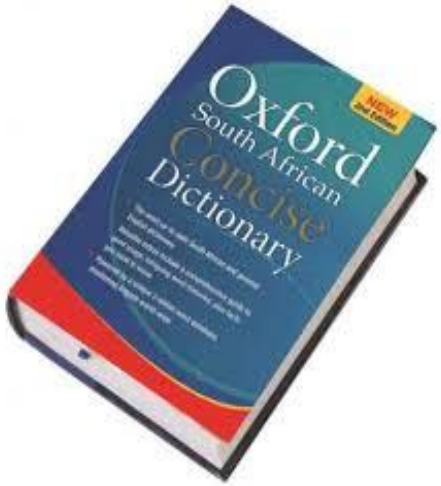
¿Cómo lo hacemos?

Un método algorítmico (FA) puede realizar automáticamente los siguientes tipos de transformaciones:

CLIENTES	→	CLIENTE + N + PL
CLIENTE	→	CLIENTE + N + SG
BANCOS	→	BANCO + N + PL

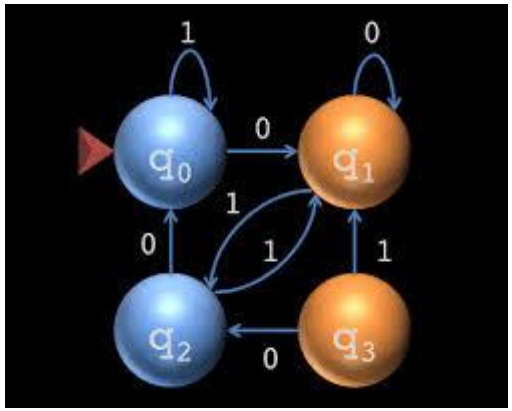
¿Qué necesitamos?

Lexicón



Reglas Morfológicas

Táctica Morfológica

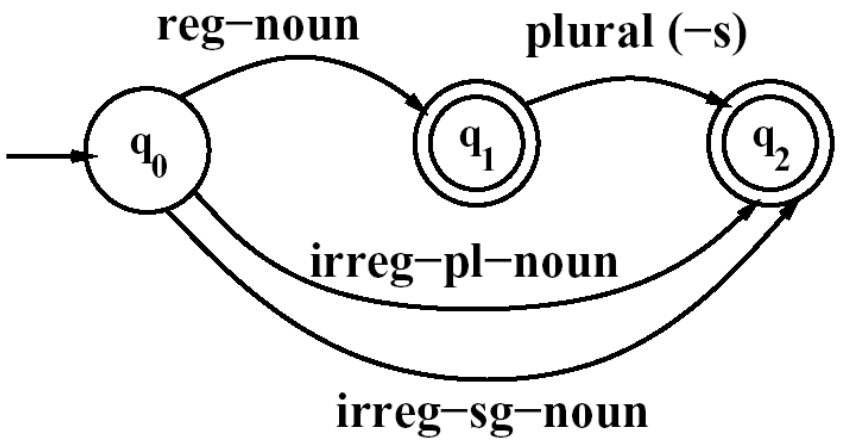


Morfología *Inflexional*

Lexicón

reg-noun	Irreg-pl-noun	Irreg-sg-noun	plural
fox cat dog	geese sheep mice	goose sheep mouse	-s

Regla (expresada por un FA)



RESUMEN

- El **análisis morfológico** permite separar y entender la naturaleza de las palabras de un texto.
- La inflexion morfológica produce dos métodos de analisis: ***Lematización*** y ***Stemming***.
- Los métodos usuales para implementar analizadores morfológicos se basan en reglas y **máquinas de estados finitos**.


CASE STUDY

Problema

- ✓ Una empresa de comunicaciones desea proveer de un sistema de auto-atención telefónica para clientes.
- ✓ Por ahora, sólo se puede “guiar” al cliente utilizando palabras aisladas (SIN comprensión de una oración).
- ✓ Clientes responden mediante voz, y el sistema los va “guiando” a través de un FA.

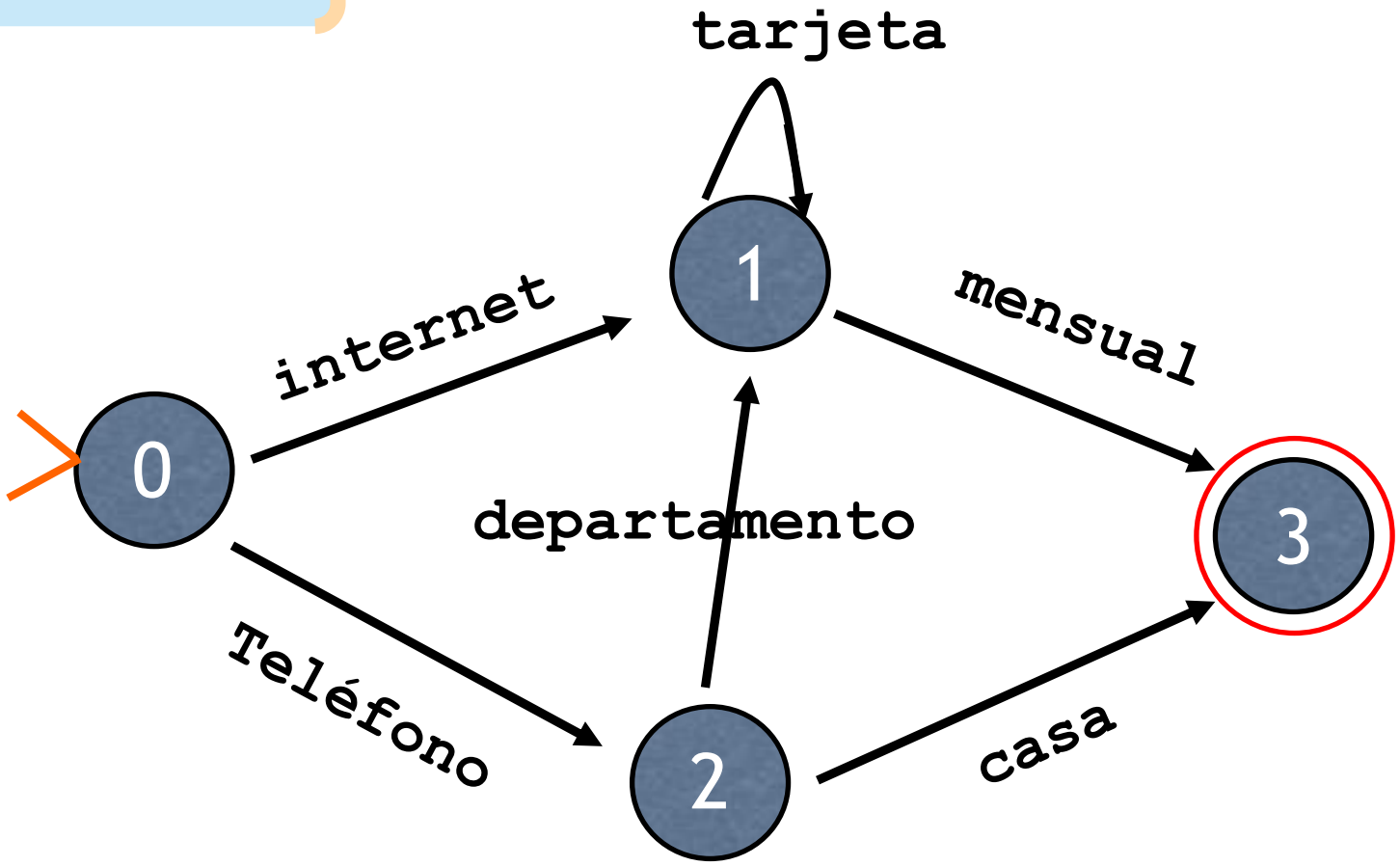
Posible Solución

1. Diseñar un *autómata* para la interacción con el cliente (los *símbolos* representan las *acciones* posibles).

2. (opcionalmente) Realizar *lematización* de la entrada del usuario para evitar variabilidad morfológica en la consulta.

3. Re-utilizar módulo para *reconocimiento hablado* de la orden (ASR), y otro para *síntesis de la respuesta (TTS)*.

Autómata:


tarjeta
internet
mensual
departamento
Teléfono
casa
0
1
2
3

Programas

Cargar en *Google Colab*, los siguientes programas:

- *Lematización*: morfo.
- *Reconocimiento y síntesis de voz*: asr-tts.
- *Autómata (FA)*: automata.

Notas

Utilizaremos dos bibliotecas o toolkits que permiten funcionalidades para diferentes tareas de NLP:

- **NLTK** (Natural-Language Toolkit): orientado a investigación, incluye métodos para varias tareas de NLP y datasets para entrenar modelos propios (www.nltk.org) para diferentes lenguas.

- **SpaCY**: orientado para construir sistemas industriales de NLP, e incluye modelos estadísticos y de aprendizaje automático pre-entrenados para más de 50 lenguajes (https://spacy.io).