

Laurea Specialistica in Informatica a.a. 2006-2007

Interazione Uomo-Macchina II:

Interfacce Intelligenti

Nicole Novielli e Fiorella de Rosis

novielli_at_di.uniba.it

Introduzione

Prima parte: Formalizzazione e Ragionamento

- 1.1. Ragionamento logico:
 - Formalizzazione
 - Risoluzione
- 1.2. Ragionamento incerto
 - Reti Causali Probabilistiche
 - Reti dinamiche
 - Apprendimento di Reti

Seconda parte: Modelli di Utente

- 2.1. Modelli logici
- 2.2. Modelli con incertezza

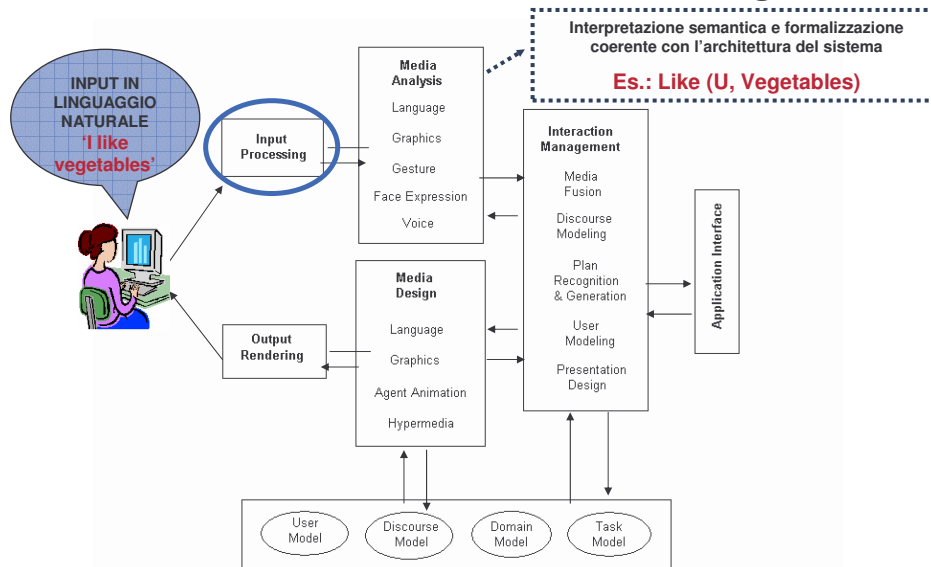
Terza parte: Interazione in linguaggio naturale

- 3.1. Generazione di messaggi
 - Introduzione
 - Teorie
 - Metodi

3.2. *Comprensione di messaggi*

Quarta parte: Simulazione di dialoghi

Architettura di una interfaccia intelligente



Natural Language Understanding

(Comprensione e interpretazione del linguaggio naturale)

Nell'analisi del linguaggio si distingue, classicamente, fra:

- *Morfologia*: studio della **struttura delle parole**
- *Sintassi*: studio della **struttura delle frasi**
- *Semantica*: studio del **significato delle frasi**
- *Pragmatica*: studio degli **obiettivi comunicativi** delle frasi

Morfologia: Le parole possono essere categorizzate in 'Parts of Speech':

Parts of speech	Esempio
Nouns (noun)	dog, equation, concert
Pronouns (pro)	I, you, it, they, them
Possessive (pos)	my, your
Verb (verb)	is, had, went, loves
Adjective (adj)	red, large, lovely
Determiner (det)	the, a, some
Proper noun (prop)	Alice, John
Conjunction (conj)	and, but, since
Preposition (prep)	In, to, into
Auxiliary verb (aux)	be, have
Modal verb (modal)	will, can, must, should
Adverb (adv)	closely, quickly,
Wh-word (wh)	who, what, where,
Final punctuation (fpunct)	. , ? !

Esempio per l'inglese

Le 'parts of speech' si chiamano anche 'simboli lessicali'

Sintassi:

La struttura delle frasi in un determinato linguaggio può essere definita con una Grammatica

Le Grammatiche 'context-free' (CFG) permettono di definire la struttura delle frasi 'corrette' in un linguaggio naturale semplice.

Una CFG è composta da:

- un insieme di *simboli terminali*
- un insieme di *simboli non terminali*
- un non terminale designato come '*simbolo iniziale*' (o '*seme*')
- un insieme di *regole di produzione* che hanno un unico non-terminale a sinistra ed una stringa di simboli terminali e/o non terminali a destra

Un esempio semplice di CFG

Simboli non terminali: le parts of speech che abbiamo visto prima, oltre a categorie più ampie: np, vp, s, s-maj e al *Seme:* s-maj

Regole:

1. s-maj → s fpunc
2. s → np vp
3. np → noun
4. np → det noun
5. vp → verb
6. vp → verb np
7. vp → verb np np
8. det → the
9. noun → salespeople
10. noun → dog
11. noun → biscuits
12. verb → ate
13. verb → sold
14. fpunc → .

*Questa grammatica permette, ad
Esempio, di generare la frase:*

“Salespeople sold the dog biscuits”

Ma anche (estendendo il lessico):

“John sent Mary an email”

e simili

Usi delle Grammatiche

Le grammatiche possono essere utilizzate:

In modo generativo, per generare tutte e sole le stringhe che appartengono ad un linguaggio

In modo riconoscitivo, per riconoscere e classificare le frasi appartenenti a un linguaggio.

Uso generativo: Derivazione di una frase

a partire dal seme, si applicano le regole della grammatica una alla volta e in una sequenza opportuna, fino a ottenere la frase.

Un esempio di derivazione

“Salespeople sold the dog biscuits”

1. s-maj → s fpunc
2. s → np vp
3. np → noun
4. np → det noun
5. vp → verb
6. vp → verb np
7. vp → verb np np
8. det → the
9. noun → Salespeople
10. noun → dog
11. noun → biscuits
12. verb → ate
13. verb → sold
14. fpunc → .

*Applico le regole una alla volta,
a partire dal seme,
fino ad ottenere la stringa:*

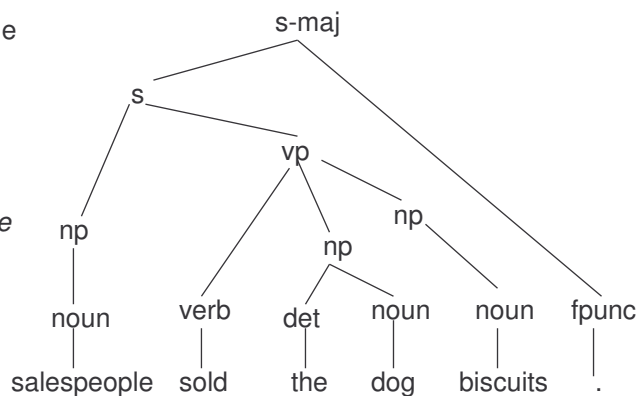
s-maj (1) ⇒ s fpunc (14) ⇒ s . (2) ⇒
np vp . (3) ⇒
noun vp fpunc (9) ⇒ Salespeople vp . (7) ⇒
Salespeople verb np np . (13) ⇒
Salespeople sold np np . (4) ⇒
Salespeople sold det noun np . (8) ⇒
Salespeople sold the noun np . (10) ⇒
Salespeople sold the dog np . (11) ⇒
Salespeople sold the dog biscuits.

La struttura di una frase può essere rappresentata in un albero sintattico

“Salespeople sold the dog biscuits”

np: predicato nominale
vp: predicato verbale
s: sentence

*(il seme della
grammatica è la radice
dell'albero sintattico)*



(s-maj s (np (noun salespeople)) vp ((verb sold) np ((det the) (noun dog))
(np (noun biscuits))) (fpunc .))

Esercizio 1

Aggiungi, alle regole della CFG del lucido n 4, la seguente:

np → det noun noun

E prova a disegnare l'albero sintattico della frase:
"Salespeople sold the dog biscuits."

Questo albero è unico?

Le Grammatiche possono essere 'ambigue'

Una *Grammatica* è *ambigua* quando ammette **diverse interpretazioni della stessa frase** (cioè diversi alberi sintattici che la rappresentano).

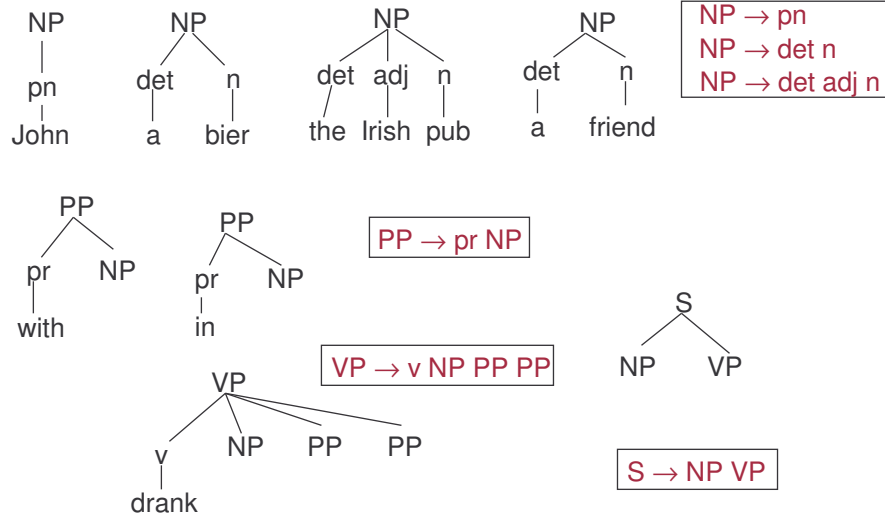
Le Grammatiche devono essere definite in modo da evitare le ambiguità di interpretazione!!! (vedi ad es le Grammatiche dei linguaggi di programmazione).

Tuttavia, alcuni *linguaggi* sono **inerentemente ambigui**: Non è possibile, cioè, definire delle grammatiche non ambigue che li rappresentino.

Tutti i linguaggi naturali hanno questa caratteristica; vedi l'esempio della frase precedente (comprare 'biscotti per cani' vs 'i biscotti per il cane').

Impariamo a scrivere una Grammatica

Consideriamo la frase: “John drank a bier in the Irish pub with a friend”.
Analizziamola a partire dalle ‘parts of speech’.



Esercizio 2

Definisci una Grammatica per la generazione delle seguenti frasi relative allo scenario di un dialogo tra un utente ed un sistema che svolge il ruolo di agenzia di viaggio:

I want to go to London

I would like a hotel room

I am leaving from Bari in the morning

Utilità delle Grammatiche

Le Grammatiche sono un modo tipicamente 'generativo' di definire i linguaggi. Possono essere utilizzate, quindi, per definire, in un simulatore di dialogo, il linguaggio in cui si esprime il sistema (generazione 'superficiale' delle singole frasi).

Sono, però, anche alla base dei **sistemi di parsing**.

Possono essere utilizzate, quindi, anche per definire il linguaggio con cui l'utente può interagire con il sistema
(NB: ponendo delle limitazioni al suo modo di esprimersi!!!)

Esempio

Un frammento di grammatica per l'Es. sull'agenzia di viaggi (esercizio 2):

HOTEL_REQUEST:= WANT HOTEL

WANT:= PRON VERB

PRON:= i / we / ...

VERB:= want/ need / would like / ...

HOTEL:= hotel / motel / accomodation / place to stay / ...

Quali frasi permette di riconoscere questa grammatica?

Esercizio 3

Modifica la grammatica descritta nel lucido precedente per riconoscere anche frasi del tipo:

I would like a nice, quiet hotel in the central part of the city.

I need a cheap room in a hotel for young people.

Please find me a room in a hotel close to Piazza di Spagna!

Hey baby, would you please find me a room in a cheap hotel?

... e simili ...

Transition Network (TN)

Le Grammatiche sono un metodo *generativo* per la definizione dei linguaggi.

Permettono cioè di definire *come generare* tutte e sole le stringhe che appartengono ad un linguaggio.

Gli Automi sono, invece, un metodo *riconoscitivo* per la definizione dei linguaggi.

Permettono cioè di definire *come riconoscere* tutte e sole le stringhe che appartengono ad un linguaggio.

Per il Teorema di equivalenza fra Grammatiche e Automi (nei linguaggi di tipo 3), una Grammatica molto semplice può essere rappresentata, graficamente, con un *Diagramma di Transizione fra gli Stati (TN)*

Transition Network (TN): un esempio

La grammatica G1:

np → art np1

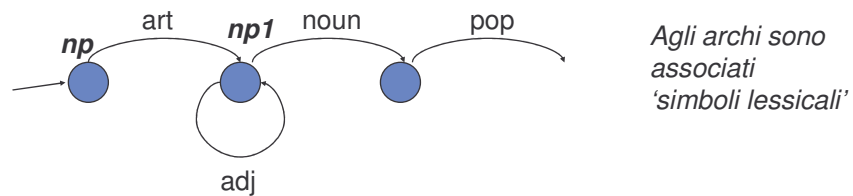
np1 → adj np1

np1 → noun

Permette di generare **frammenti di frasi molto semplici** (predicati nominali) come:

“a purple cow”, “the large American city”,...

Il suo TN equivalente è il seguente:



Limiti dei Transition Network (TN)

I TN sono automi a stati finiti, equivalenti a **'grammatiche regolari'**.

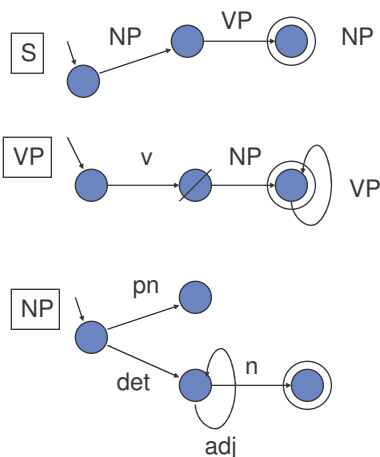
Permettono di fare analisi lessicali o di riconoscere frasi semplici, ma **non** di riconoscere **linguaggi più complessi** (come quelli definibili con CFG).

Per questi linguaggi, occorre introdurre una *memoria a stack* e associare ad ogni arco, oltre ad un carattere letto, operazioni di lettura/scrittura in questa memoria.

Occorre cioè passare da 'Transition Network' a **'Recursive Transition Network'**

Recursive Transition Networks (RTN)

S'introduce uno stack che permette di saltare dal TN principale ad uno secondario (innestamento).



*John drank
John went to the pub
John went to the Irish pub
John drank a beer in the new Irish
pub with a friend*

...

Agli archi possono essere associati simboli lessicali oppure non terminali. Quando si attraversa un arco a cui è associato un simbolo non terminale, si memorizza il nome del simbolo in un registro e si salta ad un sub-RTN che ha questo nome. Alla fine dell'esecuzione del sub-RTN, si torna all'RTN principale.

Augmented Transition Networks (ATN)

Una ATN è un diagramma di transizione fra gli stati, ai cui archi possono essere associati una label e una *condizione*.

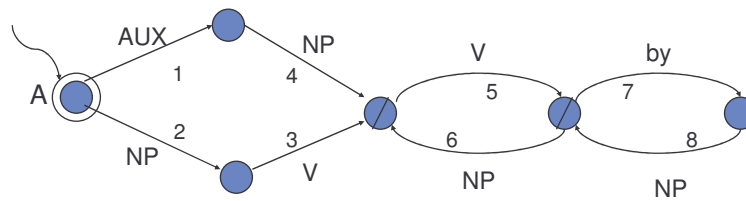
Le *label* possono denotare **categorie di parole** oppure **altre ATN**: si tratta quindi di diagrammi ricorsivi.

Le **condizioni** devono essere soddisfatte perché l'arco possa essere attraversato.

Un insieme di **registri** permette di memorizzare risultati intermedi oppure lo stato dell'esplorazione dell'ATN.

Un esempio di ATN

“The rice was eaten by the cat”

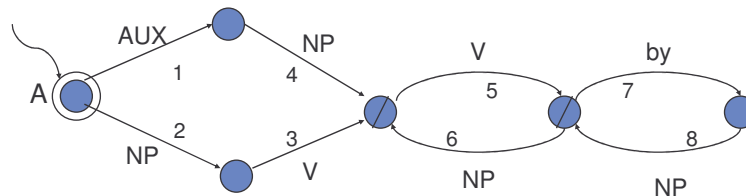


<i>test</i>	<i>action</i>
1 T	SETR V*
2 T	SETR TYPE'QUESTION SETR SUBJ*
3 agrees SUBJ*	SETR TYPE'DECLARATIVE SETR V*
4 agrees* V	SETR SUBJ*
5 AND(GETF PPRT)(=V'BE)	SETR OBJ SUBJ SETR V*
6 TRANS V	SETR AGFLAG TRUE SETR SUBJ'SOMEONE SETR OBJ*
7 AGFLAG	SETR AGFLAG FALSE
8 T	SETR SUBJ*

**Tabella
Condizioni-
azioni**

Riconosciamo la frase

“The rice was eaten by the cat”



Iniziamo dallo start node A.

Ci sono due cammini possibili; ma poiché la prima parte della frase ('the rice') è un NP, si sceglie l'arco 2. Prima di attraversarlo, però, si controlla se ci sono altre condizioni o azioni da soddisfare (vedi tabella). 'T' significa che non ci sono altri test.

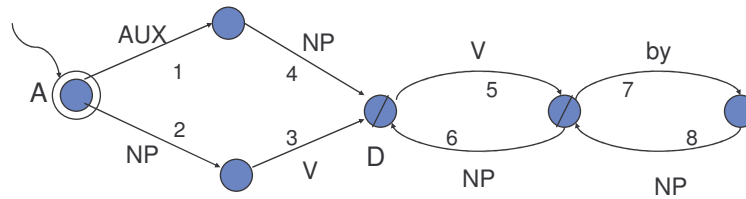
Si esegue l'azione che consiste nel memorizzare 'rice' nel registro SUBJECT e TYPE come 'declarative'.

Si passa all'esame di 'was'.

È un verbo: quindi si traversa l'arco 3. Il test indica di controllare che questo verbo si accordi con il soggetto. Se questa condizione è verificata, si memorizza 'was' nel registro VERB.

Riconosciamo la frase

"The rice was eaten by the cat"



Siamo ora al nodo D.

'Eaten' è un verbo; il test sull'arco 5 indica di controllare che questo verbo si accordi con l'ausiliario.

Si compiono ora 4 azioni: il contenuto del registro SUBJ ('rice') viene trasferito nel registro OBJ. Il contenuto del registro VERB viene sostituito con 'was eaten'. Un flag indica che la frase è in forma passiva. Il registro SUBJ viene settato al valore 'someone'.

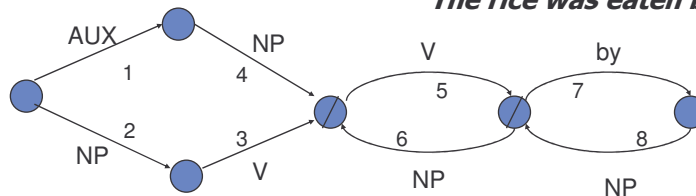
By: si sceglie l'arco 7.

'The cat': un'altra 'noun phrase': si traversa l'arco 8 e si memorizza il frammento di frase in SUBJECT.

Il processo termina.

Riassumiamo il processo:

"The rice was eaten by the cat"



test	action
• T	SETR V* SETR TYPE:QUESTION
2 T	SETR SUBJ* SETR TYPE:DECLARATIVE
3 agrees SUBJ*	SETR V*
4 agrees* V	SETR SUBJ*
5 AND(GETF PPRT)=(V'BE)	SETR OBJ SUBJ SETR V* SETR AGFLAG TRUE SETR SUBJ'SOMEONE
6 TRANS V	SETR OBJ*
7 AGFLAG	SETR AGFLAG FALSE
8 T	SETR SUBJ*

The rice: NP; arc 2
(SUBJ = the rice;
DECLARATIVE)

was: V 3

eaten: V 5
(OBJ=rice; V=was eaten;
AGFLAG=T; SUBJ=SOMEONE)

by: 7
AGFLAG=F

the cat: NP 8
SUBJ=the cat

Esercizio 4 (non semplice!)

Interpreta, con l'ATN precedente, le frasi seguenti:

“The cat ate the rice”

“Did the cat eat?”

“The cat saw the TV”

“Was the fish eaten by the cat?”

Fai qualche esempio di frasi non interpretabili con quella ATN.

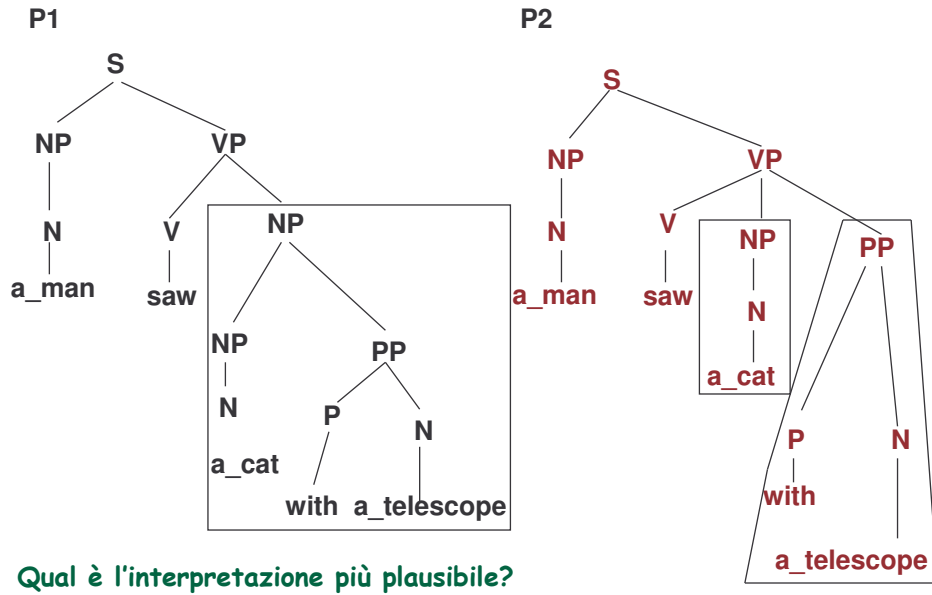
Parsing probabilistico

Non tutte le sequenze di parole si presentano, in un linguaggio, con la stessa frequenza.

Questa conoscenza può essere utilizzata per riconoscere e interpretare frasi in condizioni di incertezza.

In questo caso, il processo di riconoscimento produce, come risultato, un **valore di probabilità** per ciascuna delle interpretazioni possibili della stringa di input.

Esempio: "A man saw a cat with a telescope"



Quale grammatica

$S \rightarrow NP VP$

$VP \rightarrow V NP$

$VP \rightarrow V NP PP$

$NP \rightarrow NP PP$

$NP \rightarrow N$

$PP \rightarrow P N$

$N \rightarrow a_man$

$N \rightarrow a_cat$

$N \rightarrow a_telescope$

$P \rightarrow with$

$V \rightarrow saw$

Ad es. il non terminale VP può essere riscritto in due modi differenti:

Quale tra i due è il più plausibile per 'The man saw a cat with a telescope'?

SOLUZIONE:

grammatiche probabilistiche

Grammatiche Probabilistiche

Si propongono di assegnare un valore di probabilità per ogni frase generabile con la grammatica.

Formalmente, la definizione di grammatica probabilistica differisce da quella delle grammatiche tradizionali soltanto nel senso che:

- *ad ogni produzione viene assegnato un valore di probabilità e*
- *la somma dei valori di probabilità associati alle produzioni che hanno, a sinistra, lo stesso non terminale è = 1.*

Esempio:

S → NP VP [.80]	Nome → Noun [.75]
S → AuxNP VP [.15]	Nome → Noun Nome [.20]
S → VP [.05]	Nome → NomeProprio Nome [.05]
NP → Det Nome [.20]	VP → Verbo [.55]
NP → NomeProprio [.35]	VP → Verbo NP [.40]
NP → Nome [.05]	VP → Verbo NP NP [.05]
NP → ProNome [.40]	Det → that [.05] the [.80] a [.15]
	...

Come usare l'incertezza nelle produzioni

La probabilità di un albero di parsing T per una frase S si può stimare come

$$P(T,S) = \prod_{n \in T} p(r(n))$$

Se diverse interpretazioni sono possibili, l'algoritmo di parsing sceglierà, tra di esse, l'albero con probabilità massima.

Riprendiamo l'esempio precedente

associando ad ogni produzione un valore di probabilità

Qual è l'interpretazione più probabile, per
'A men saw a cat with a telescope'?

S → NP VP [1]
 VP → V NP [0.25]
 VP → V NP PP [0.75]
 NP → NP PP [0.5]
 NP → N [0.5]
 PP → P N [1]

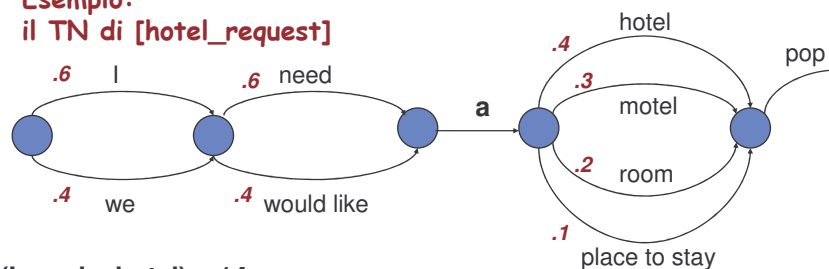
N → a_man
 N → a_cat
 N → a_telescope
 P → with
 V → saw

P1 S ⇒ NP VP [1] ⇒ N [0.5] V NP [0.25] ⇒
 a_man saw NP PP [1] ⇒ a_man saw N [0.50] P N
 [1] ⇒ a_man saw a cat with a_telescope
 $P(P1) = 1 * .5 * .25 * .50 * 1 = .0625$

P2 S ⇒ NP VP [1] ⇒ N [0.5] V NP PP [0.75] ⇒
 a_man saw N [0.50] P N [1] ⇒ a_man saw a cat
 with a_telescope
 $P(P2) = 1 * .5 * .75 * .50 * 1 = .1875$

E' possibile rappresentare l'incertezza associata a processi di interpretazione di frasi associando valori di probabilità agli archi dei diagrammi di transizione fra gli stati anziché alle produzioni

Esempio:
 il TN di [hotel_request]



$P(I \text{ need a hotel}) = .14$

$P(I \text{ need a place to stay}) = .04$

$P(\text{we would like a room}) = .03$

...

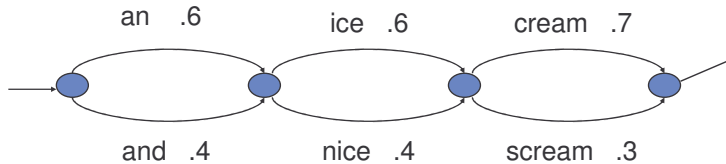
Associamo agli archi delle ATN un valore di probabilità, che indica
 "qual è la probabilità di attraversare quell'arco,
 se ci si trova nello stato da cui l'arco inizia".

La somma delle probabilità associate agli
 archi che escono da uno stato deve essere = 1.

Usi delle Probabilistic FSTN (o Markov Chain)

Nella interpretazione dello speech, i Probabilistic FSTN possono essere applicati al riconoscimento di stringhe 'poco chiare' (es. tool di speech recognition)

Esempio:

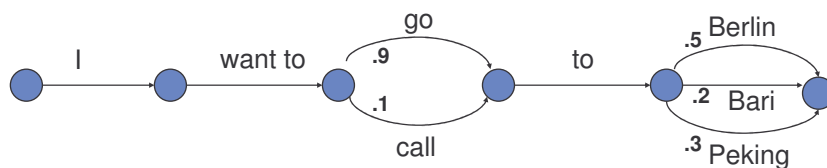


La probabilità di riconoscere una stringa è data dal prodotto delle probabilità associate agli archi attraversati.

$P(\text{an ice cream}) = .25$
 $P(\text{and nice scream}) = .048$
 $P(\text{an nice cream}) = .17$
 ...

Un altro esempio

Sistema automatico di prenotazioni di viaggi aerei: le probabilità in questo caso dipendono anche dal contesto in cui la frase viene pronunciata



"I want to go to Berlin": $P=.45$
"I want to go to Bari": $P=.18$
"I want to go to Peking": $P=.27$

"I want to call to Berlin": $P=.05$
"I want to call to Bari": $P=.02$
"I want to call to Peking": $P=.03$

**Le Catene Markoviane
aiutano a individuare
l'interpretazione più probabile
di frasi 'incomplete'**

**Esempio: un estratto di dialogo speech-based
In un Call Center.**

U: "I want to ... to ...!"

S: "Do you want to go to Berlin?"

U: "No, to ...!"

S: "Oh sorry, it's Peking where you want to go, right?"

Esercizio 5

**Fai un altro esempio di Markov Chain
per un Call Center per prenotazioni aeree.**

**Disegna la Catena di Markov e fai un esempio
di dialogo.**

Semantica

L'**obiettivo** del riconoscimento in un sistema di dialogo è quello di **formalizzare le frasi in linguaggio naturale tramite un linguaggio ad hoc** (ad es. linguaggio logico), allo scopo di 'ragionare' su di esse.

The dog ate.

Ate(dog)

A man saw a cat with a telescope

Saw(man, CatWithTelescope) gatto dotato di telescopio ☺

Oppure: Saw(man, Cat, UsingTelescope) usando un telescopio
nelle due interpretazioni della frase che abbiamo visto.

Jack downloaded the computer game

Downloaded(Jack, ComputerGame)