**TECNOLOGIE DEL LINGUAGGIO NATURALE**

**RELAZIONE PROGETTO: TRADUTTORE DA ITALIANO A ITALIANO-YODA**

**UNIVERSITÀ DEGLI STUDI DI TORINO A.A 2018/19**

**STUDENTI DOCENTE**

Donadio Gaetano 899187 prof. Mazzei Alessandro

Marrazzo Alessandro 898842

**SISTEMA DI TRADUZIONE ITALIANO -> ITALIANO-YODISH**

**Introduzione**

L'algoritmo Cocke-Younger-Kasami (CKY) è un algoritmo di analisi per grammatiche context-free. Impiega un procedimento bottom-up e la programmazione dinamica per generare l’albero sintattico di una frase. La versione standard di CKY funziona solo su grammatiche context-free fornite in Chomsky Normal Form (CNF), cioè con regole grammaticali binarie (del tipo **A -> B C)** e regole lessicali unarie (del tipo **A -> a**).

Qualsiasi grammatica context-free, che non genera la stringa vuota, può essere rappresentata in Chomsky Normal Form, aggiungendo alcune regole o modificando opportunamente le regole della grammatica in uso.

Nel nostro caso è stato implementato tale algoritmo nel sistema di traduzione di una frase dall’italiano all’italiano-yodish. La differenza principale tra queste due lingue è che, mentre in italiano le frasi si generano nella forma soggetto - verbo – oggetto, in italiano-yodish si antepone l’oggetto al soggetto e al verbo.

**Strutture dati utilizzate**

La grammatica italiana viene memorizzata all’interno di un hashmap. È stata scelta questa tipologia di struttura dati per indicizzare al meglio le regole, cosi da avere un accesso diretto ad esse. Le regole sono salvate nell’hashmap in modo che la coppia <chiave, valore> sia formata, rispettivamente, da corpo e testa della regola stessa (se, ad esempio, abbiamo una regola **S -> NP VP**, essa verrà salvata nell’hashmap come una tupla, avente come chiave la stringa “**NP VP**”e come valore un sintagma, rappresentato dalla stringa “**S**”). In questo modo, durante l’esecuzione dell’algoritmo CKY, si potrà controllare direttamente se esiste un sintagma che soddisfi la regola costituita dalla coppia di valori.

Ogni sintagma è rappresentato da una struttura dati, denominata TREE, che ha al suo interno una stringa, rappresentante il valore del sintagma e una lista di alberi figli, rappresentanti i sintagmi che lo generano.

Tree implementa anche alcuni metodi che agevolano l’accesso, l’aggiunta e la rimozione di un albero dalla lista dei figli.

Tale struttura dati è stata scelta, in quanto è quella che rappresenta al meglio la relazione tra sintagmi.

L’algoritmo sfrutta una tabella per la generazione dell’albero sintattico. Tale tabella è una matrice di celle.

Ogni cella è multislot, ovvero può contenere più elementi, ed è rappresentata con un oggetto, denominato CELL, al cui interno è presente una lista di alberi, ognuno dei quali rappresenta un elemento presenta nella cella.

La classe Cell implementa anche un metodo che agevola l’aggiunta di alberi nella cella e un metodo che agevola l’accesso all’albero i della lista.

Durante l’esecuzione dell’algoritmo, l’albero sintattico della frase si costruisce parallelamente al popolamento della matrice. Ogni qualvolta una cella

L’albero sintattico della frase è corretto se, al completamento dell’algoritmo in posizione **[0, n- 1]** è presente l’albero avente come valore il sintagma **S**.

**Descrizione** **del sistema**

Il sistema di traduzione può essere suddiviso in diverse fasi, quali, **parsificazione** delle grammatiche, **derivazione** dell’albero sintattico della frase in italiano e **traduzione** nell’albero sintattico in italiano-yodish, con successiva formulazione della frase finale.

Per la realizzazione del sistema, sono state scritte una grammatica per l’Italiano e una per l’Italiano-Yodish che permettono di parsificare alcune frasi d’esempio nelle rispettive lingue. Le grammatiche sono state costruite in Chomsky Normal Form (CNF), in cui abbiamo solamente regole non lessicali binarie, mentre le regole lessicali sono unarie.

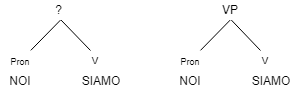
La fase di parsificazione delle grammatiche è effettuata da una classe parser che legge da file appositi le grammatiche contenenti le varie regole e le memorizza nel sistema così da poter essere utilizzate nelle fasi successive.

La fase di derivazione dell’albero sintattico viene eseguita implementando l’algoritmo CKY. L’algoritmo prende in input una sequenza di parole e inizia a riempire la matrice posizionando i POS tag riferiti ad ognuna di esse lungo la diagonale principale. In seguito, per ogni cella **[j, i]** sopra la diagonale principale, si verifica se nella grammatica sia presente una regola della forma **A -> B C** tale che **B** è rappresentata dal valore nella cella **[j, k-1]** e **C** dal valore nella cella **[k, i]** con k compreso tra **j+1** e **i.** In caso tale regola sia presente nella grammatica si riempie la cella **[j, i]** con il valore **A.** La porzione della matrice al di sotto della diagonale è ininfluente ai fini dell’algoritmo, dunque non viene utilizzata.

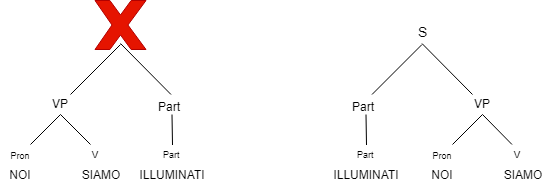
Se, al termine dell’algoritmo, nella posizione **[0, n-1],** dove **n** è il numero di parole nella frase, viene generato l’albero avente valore **S**, rappresentante la frase completa, l’algoritmo ha avuto successo e restituirà l’albero sintattico.

Nella fase di traduzione, l’albero sintattico generato precedentemente, viene trasformato in modo tale da rispettare la grammatica finale ovvero quella italiano-yodish. L’algoritmo di traduzione è basato su un approccio bottom-up. Esso riceve in input una sequenza di tag che inizialmente corrisponde a quelli delle parole nella frase in italiano e attraverso chiamate ricorsive, genera l’albero sintattico corrispondente alla frase tradotta in lingua italiano-yodish.

Durante ogni chiamata ricorsiva, vengono analizzate le coppie adiacenti all’interno della sequenza e, per ognuna di esse si verifica l’esistenza di una regola grammaticale. Se questa esiste la coppia di tag viene sostituita dal tag che soddisfa la regola.



Qualora la regola non esista, viene effettuato un nuovo controllo invertendo l’ordine dei termini nella coppia. Qualora anche questo controllo dia esito negativo si passa ad analizzare la coppia successiva.



L’algoritmo termina quando nella sequenza di tag è presente solo il tag **S.**

Durante l’esecuzione è possibile che si generino tag **S** in corrispondenza di frasi parziali di senso compiuto, ma ancora non è stato trasformato completamente l’albero. In tal caso, si ignora la regola e si invertono i due termini all’interno della sequenza.

Al termine dell’esecuzione, le foglie dell’albero appena generato rappresentano le parole della frase tradotta e, con una ricerca in profondità, si va a costruire la frase finale che è fornita in output dal sistema.