ABSTRACT

La teoria dei codici, nata negli anni '40, costituisce una branca dell'informatica teorica in continuo sviluppo. Ha suscitato grande interesse sia da un punto di vista pratico, per le numerose e importanti sue applicazioni, sia da un punto di vista teorico, per le ricche e suggestive strutture algebriche che in essa intervengono.

Un codice è un insieme di parole su un alfabeto finito che è accettabile nella trasmissione dell'informazione: abbinando tali parole si ottengono messaggi che possono essere univocamente decifrati.
L'univoca decifrabilità formalizza l'idea di non ambiguità di un messaggio codificato.
Tipici esempi sono il codice Morse, il codice ASCII.

Nel 1950 Sardinas e Patterson fornirono una condizione necessaria e sufficiente perché un insieme sia un codice, usata successivamente da Capocelli e Hoffmann (1985) che idearono un algoritmo per testare l'univoca decifrabilità di un insieme finito di parole.

Nel 1982 Rodeh disegnò un algoritmo per testare l'univoca decifrabilità di un codice C basato sull'algoritmo di McCreight per

la costruzione dei suffix-tree con complessità pari a $O(nm)$ dove n rappresenta il numero di parole di codice da processare e m la loro lunghezza totale.

L'algoritmo si basa sulla definizione di coda (particolari parole costruite a partire dall'insieme C) e consente di trasportare l'univoca decifrabilità di un insieme finito C sull'insieme delle code di C. Se nessuna coda è una parola di codice allora C è un codice (e viceversa).

Dunque l'algoritmo di Rodeh genera tutte le code possibili e le confronta con tutte le parole di codice terminando negativamente nel caso il confronto risulti positivo.

Cruciale è quindi la definizione dell'insieme delle code in modo efficiente. Per tener traccia dell'insieme delle code, Rodeh usa il suffix-tree , come implementato dall'algoritmo di McCreight (1976).

I suffix-tree (o albero dei suffissi) è una struttura dati che evidenzia la struttura interna di una stringa in modo da facilitare operazioni comuni come la ricerca di sottostringhe. Un esempio d'uso potrebbe essere la ricerca di un testo in un dizionario: le parole presenti nel dizionario verrebbero rappresentate da un albero dei suffissi che agirebbe da indice per il testo da cercare. La particolarità è che la ricerca verrà effettuata in tempo lineare. Di nostro interesse è sicuramente la possibilità di verificare tramite un suffix-tree se un linguaggio è un codice, ovvero se risulta univocamente decifrabile.

Esistono diversi algoritmi per la costruzione dei suffix-tree, ma il più interessante, almeno ai fini di questo lavoro di tesi, è quello di McCreight che nel 1976 disegnò un algoritmo in tempo lineare più efficiente in spazio rispetto ai precedenti.

In particolare, supponiamo che $C = \{\alpha_1, \dots, \alpha_n\}$ sia l'insieme di cui vogliamo testare l'univoca decifrabilità.
Inizialmente viene costruito il suffix tree T sul testo $w = \alpha_1 \$_1 \alpha_2 \$_2 \dots \alpha_n \$$ dove $\$_i$, $\$$ sono simboli speciali, diversi tra loro e da tutte le lettere degli α_1.
Tale suffix tree T è costruito usando l'algoritmo di McCreight, come implementato nella tesi di C. Giordano "Implementazione in Java dell'algoritmo di McCreight per la costruzione dei suffix tree ".
In tal modo si ottengono rappresentate nelle foglie di T i suffissi di w.
Oggetto di questo lavoro di tesi è stata la modifica di T per consentire poi l'implementazione dell'algoritmo di Rodeh.
In particolare è stato aggiornata la struttura dei nodi dell'albero aggiungendo altre informazioni indispensabili alla riuscita dell'algoritmo di Rodeh (i valori p e d). Inoltre è stato aggiunto un metodo per consentire la potatura di alcuni rami contenenti suffissi obsoleti creati con l'uso di separatori all'interno della stringa in input. In tal modo sarà poi possibile determinare le code da utilizzare per l'algoritmo di Rodeh e in futuro, quindi, completare l'implementazione dell'algoritmo.

La prima parte della tesi è completamente dedicata alle definizioni di base in modo da consentire una maggiore comprensione ai capitoli successivi nonché alla presentazione dei suffix-tree con una ampia descrizione e alcuni esempi. L'ultima parte è dedicata alla descrizione e all'implementazione dell'algoritmo di Rodeh nonché ai dettagli sulle strutture dati utilizzate.

In particolare il primo capitolo presenta una panoramica sulla teoria dei codici evidenziando alcune nozioni di base essenziali alla comprensione del lavoro svolto: definizione di parole, linguaggi e codici, Test di Sardinas e Patterson , descrizione dell'algoritmo di Capocelli e Hoffmann e più in particolare descrizione dell'algoritmo di Rodeh.

Il secondo capitolo tratta il test sull'univoca decifrabilità di Sardinas e Patterson e gli algoritmi basati su di esso, fornendo particolare attenzione all'algoritmo di Rodeh.

Il terzo capitolo fornisce una ampia descrizione degli alberi dei suffissi soffermandosi sulla loro importanza e sugli ambiti di utilizzo proseguendo poi con la struttura e le proprietà fino a giungere a una breve presentazione sull'algoritmo di McCreight.

Nel quarto capitolo vengono presentate le modifiche al suffix tree di McCreight effettuate da Rodeh per il suo algoritmo che sono state oggetto dell'implementazione.

Il quinto capitolo è il cuore del lavoro e riguarda l'implementazione dell'algoritmo di Rodeh passando alla presentazione di alcune strutture dati utilizzate, alla modifica

dell'STNode() del Dott. Giordano, alla descrizione e alla costruzione di metodi fondamentali come il Pruning() e l'aggiornamento dei nodi dell'albero.