Domande teoria Traduttori 1

*1) Definire le grammatiche lineari destre.*

Le grammatiche lineari destre sono grammatiche lineari in cui il simbolo non terminale è a destra, di conseguenza l’albero di derivazione avrà un asse di crescita che crescerà appunto alla sua destra, esse sono nella forma (Σ\* U ε)(V U ε): vi è un numero arbitrario di terminali a sinistra e/o un non terminale a destra.

*2) Dimostrare che i linguaggi generati dalle grammatiche lineari sono contenuti da quelli generati da grammatiche context-free.*

Una grammatica lineare è per definizione una grammatica context-free, per dimostrare il contenimento si deve trovare un esempio che non appartenga alla prima grammatica: Dato il linguaggio L={a^n,b^n,c^m,d^m,...}, per generare lo stesso numero di a e b, c e d, eccetera devo avere più assi di crescita che risolvano questi bilanciamenti, questo va contro la definizione di grammatica lineare e quindi non corrispondono alle context-free ma sono contenute in esso.

*3) Definire formalmente la definizione di linguaggio denotato da una espressione regolare*

Data un’espressione regolare e, l’implicazione segue le seguenti regole:

* e1 U e2 U...U ek => ei con 1<=i<=k

l’unione di più espressioni regolari indica un’alternatica, infatti da esse si sceglierà l’espressione regolare i presente nell’insieme;

* e\*=> e U e k volte con k>=0

e\* indica una serie di unioni eseguita un numero arbitrario di volte che parte da 0 e tende a infinito, in poche parole è un unione dell’espressione regolare e con se stessa eseguita k volte.

* e+=> e U e k volte con k>0

come per e\* ma con k>0

* e^n=> e U e n volte

Unione di e con sé stessa n volte, è utilizzata nei bilanciamenti

queste regole sono strettamente contenute l’una dentro l’altra, le uniche eccezioni sono la prima e la quanta le quali possono coincidere.

Date due espressioni regolari e1 ed e2, si dice che e1 deriva da e2 se e1 è formata da ABC, e2 è formata da ADC e B deriva da D, la derivazione è ennesima quando compiuta in n passi, è incerta quando non si sa il numero di passi e sinistra quando parte da sinistra e si sposta a ogni passo verso destra, l’ordine di derivazione tuttavia non è importante.

*4) Dimostrare che i linguaggi generati da grammatiche lineari destre sono contenuti in quelli generati da grammatiche lineari.*

i linguaggi lineari destre sono per definizioni linguaggi lineari, per dimostrare il contenimento bisogna trovare un esempio che contraddice la definizione del primo: la regola X → aXb|c genera un numero n di “a” a sinistra, una c come marca di centro e un numero n di b a destra, per fare in modo che il numero di a sia uguale a quello di b devo per forza utilizzare un asse di crescita centrale, contraddicendo così l’ipotesi che le grammatiche lineari destre sono uguali a quelle lineari.

*5) Definire formalmente (per casi) le espressioni regolari*

Le espressioni regolari sono un linguaggio che permette la definizione di altri linguaggi, di questo linguaggio fanno parte:

* L’insieme nullo, da esso si denota il cosiddetto linguaggio nullo ovvero senza alfabeto;
* la stringa vuota, l’unico elemento che denota il suo linguaggio;
* elementi dell’alfabeto;
* concatenamenti tra espressioni, il linguaggio denotato sarà il concatenamento tra i rispettivi linguaggi;
* unione tra espressioni: il linguaggio denotato sarà l’unione dei rispettivi linguaggi;
* Chiusura di Kleene su un’espressione, in tal caso il linguaggio denotato è la chiusura del suo linguaggio.

*6) I linguaggi regolari sono tutti deterministici? Giustificare la risposta*

I linguaggi regolari sono tutti deterministici perchè sono finiti e possono essere riconosciuti dagli automi a stati finiti, per questi ultimi esiste sempre una versione deterministica.

*7) Definire le funzioni di transizione δ e δ\* per gli automi a stati finiti deterministici.*

La funzione di transizione δ ha tipatura δ:Qx∑→ Q ovvero che, dato uno stato e un elemento dell’alfabeto, l’automa passa a un altro stato oppure fallisce. Dal momento che δ opera solo sui singoli elementi dell’alfabeto, serve un funzione di transizione che utilizzi le stringhe, da δ si ottiene quindi δ\* ovvero, dato uno stato e una stringa, si passa a un altro stato dell’automa oppure si va in fail (la sua tipatura è δ\*:Qx∑\* → Q). δ\* è facilmente dimostrabile per induzione: dato uno stato q e un elemento a dell’alfabeto, δ\*(q,a)=δ(q,a), dal momento che la stringa “a” è formata da un solo elemento. Nel passo induttivo, data una stringa Y, si ha che δ\*(q,Ya) = qi, essa si può ancora scomporre in δ\*(δ\*(q,Y),a) dato che δ\* restituisce come output uno stato. Per ipotesi induttiva, δ\*(δ\*(q,Y),a)=δ(δ\*(q,Y),a) dato che nella transizione più esterna è applicabile il passo induttivo.

*8) Le grammatiche context free senza produzioni autoinclusive generano sempre linguaggi regolari? Se ci sono produzioni autoinclusive, il linguaggio generato è sempre non regolare?*

Le grammatiche context-free senza produzioni autoinclusive sono considerabili come grammatiche lineari destre o sinistre dal momento che non hanno forme di bilanciamento, di conseguenza una grammatica di questo tipo genera sempre un linguaggio regolare. Nel caso vi siano produzioni autoinclusive, il linguaggio generato può essere regolare solo l’alfabeto è unitario oppure le autoinclusioni della grammatica sono formate dallo stesso gruppo di elementi:

S→ aSa|a con ∑{a} //alfabeto unitario

S→ abSab|ab con ∑{a,b} // ab è presente sia a destra che a sinistra di S

*9) Data una grammatica strettamente lineare destra, definire formalmente l’automa a stati finiti equivalente.*

Data una grammatica G=<V,∑,P,S> strettamente lineare destra, costruire l’automa M=<Q,∑,δ,q0,F>:

l’insieme di stati Q è dato dall’insieme V, lo stato iniziale q0 è dato da S, invece per quanto riguarda δ, per ogni regola A→ E, se E=aB , si ha che δ(A,a)=B, se E=a, si aggiunge uno stato T finale, quindi δ(A,a)=T. F è formato da T e da tutti gli stati di Q aventi ε tra le possibili alternative.

*10) Definire formalmente l’automa a stati finiti M deterministico da quello N non deterministico e senza epsilon mosse.*

Dato un automa N non deterministico, definire un automa M deterministico: l'insieme di stati Q é dato da tutti gli stati raggiungibili da quello iniziale, essi raggruppano tutti gli stati destinazione di una a-mossa in un'unica stato. Di conseguenza la funzione di transizione non genera più un insieme di stati ma il rispettivo “ stato unione”, ovvero uno stato che rappresenta l'insieme di stati citato prima. L'insieme di stati finali é dato dagli stati finali raggiungibili da q0 e da tutti gli stati unione che contengono uno stato finale.

*11) Dimostrare che i linguaggi regolari sono tutti òdeterministici ( nota. la dimostrazione si ottiene “combinando” alcune dimostrazioni fatte nel corso)*

*13) Sia N = <Q,Σ,ઠ,q0,F> un automa a stati finiti non deterministico con ε mosse. Definire l’automa non deterministico senza ε mosse equivalente, N’ = <Q’,Σ’,ઠ’,q0’,F’> con particolare attenzione alla definizione di ઠ’.*

L'insieme di stati Q rimane lo stesso, così come qo e l'alfabeto. La funzione di transizione deve fare le stesse cose ma senza epsilon mosse, di conseguenza si introduce il concetto di epsilon chiusura: essa é il cammino formato da tutti gli stati connessi da sole epsilon mosse. Per mantenere lo stesso comportamento, se dopo le epsilon mosse vi sono a-mosse che partono dallo stesso in cui arriviamo, si aggiunge una a-mosse dallo stato di partenza a quello in cui arriva la a-mossa. La funzione di transizione é quindi descrivibile come epsilon chiusura di una transizione su a di una epsilon chiusura di uno stato q: quella più interna serve per trovare gli stati a cui portano le epsilon mosse, la transizione per trovare gli stati a cui portano le a-mosse e quella più esterna a trovare gli stati di destinazione della nuova a-mossa da inserire. F mantiene i suoi stati, viene anche aggiunto q0 solo se é connesso a uno stato finale tramite una epsilon mossa.

*14) Quando un linguaggio è deterministico? I linguaggi finiti sono deterministici? E i linguaggi regolari? E i linguaggi context-free? Motivare le risposte.*

Un linguaggio é deterministico quando viene riconosciuto da un automa deterministico, i linguaggi regolare sono deterministici perché riconoscibili con l'automa a stati finiti (di cui esiste sempre la versione deterministica) e lo sono anche i linguaggi finiti dal momento che sono regolari.

*16) I linguaggi finiti sono sempre regolari? Motivare la risposta*

*17) Dimostrare che i linguaggi generati dalle grammatiche lineari destre sono regolari.v* I linguaggi lineari destri sono regolari perché attraverso una serie di passaggi é possibile ricavare l'espressione regolare che lo denota: si converte la grammatica in una serie di unioni, si raccogono i non terminali a fattori comune, si toglie la ricorsione utilizzando l'equivalenza S→ uS U v = u\*.v e infine si sostituiscono i non terminali con quello che generato (si riparte dal punto 2 se vi sono ancora ricorsione.)

*18) Dire quando una frase di un linguaggio è ambigua e fare un esempio*

Una frase del linguaggio é ambigua quando possiede più alberi di derivazione.

Esempio: E→ E+E|E\*E|e

Per generare e+e\*e posso generare prima l'addizione e poi la moltiplicazione oppure il contrario.

*19) Descrivere le regole per la costruzione di una grammatica lineare destra equivalente ad un automa finito M.*

L'insieme di stati Q equivale ai non terminali, tutte le transizioni diventano regole nella forma A-->aB, i non terminali che corrispondono a uno stato finale devono poter generare epsilon, lo stato symbol equivale a qo.