Traduzione: traslitterazioni traduzioni regolari e sintattiche

Prof. A. Morzenti

aa 2008-2009

TRADUZIONE SEMANTICA E ANALISI STATICA

TRADUZIONI SINTATTICHE:

1. Definizione astratta della *corrispondenza tra due linguaggi formali*.

2. Traduzioni basate su trasformazioni locali del testo sorgente prodotte dalla sostituzione di un carattere con un altro (o con una stringa) in accordo con una tabella di *traslitterazione tra i due alfabeti*.

3. *Traduzioni definite mediante espressioni regolari* e *trasduttori finiti* (automi finiti arricchiti della capacità di emettere una stringa).

4. Schemi sintattici o grammatiche di traduzione che usano (invece del linguaggio regolare del modello precedente) una *grammatica libera* per definire il linguaggio sorgente e pozzo. La macchina astratta che le calcola è il *trasduttore a pila* che sfrutta gli algoritmi di parsificazione visti in precedenza.

TRADUZIONI SEMANTICHE BASATE SU SINTASSI:

5. *Grammatiche ad attributi e traduzioni semantiche basate sulla sintassi*:
   un approccio semiformale che si appoggia sui concetti precedenti e consente la progettazione dei compilatori. La grammatica con attributi specifica regola per regola le funzioni da applicare per calcolare la traduzione in maniera compositiva.

L'aggiunta di attributi e funzioni semantiche ai traduttori finiti trova illustrazione nell'analisi lessicale, nel controllo dei tipi, nella traduzione delle istruzioni condizionali, nell'analisi sintattica guidata dalla semantica.

ANALOGIE tra TEORIA DEI LINGUAGGI e TEORIA DELLE TRADUZIONI

Sul piano della definizione insiemistica: l'insieme delle frasi del linguaggio corrisponde all'insieme delle coppie (stringa sorgente, stringa pozzo) costituenti la relazione di traduzione.
Sul piano della definizione generativa: la grammatica del linguaggio diventa quella della traduzione, che genera coppie di frasi sorgente e pozzo.
Sul piano della definizione operativa: l'automa riconoscitore a stati finiti o a pila diventa l'automa trasduttore o l'analizzatore sintattico che calcola la traduzione.

1. RELAZIONE E FUNZIONE DI TRADUZIONE

Dati i due alfabeti Σ (sorgente) e Δ (pozzo), una TRADUZIONE è una corrispondenza tra le stringhe sorgente e pozzo formalizzabile come relazione matematica tra i due linguaggi universali Σ^* e Δ^*. E' un sottoinsieme del prodotto cartesiano $\Sigma^* \times \Delta^*$.

LA RELAZIONE DI TRADUZIONE ρ
è un insieme di coppie (x, y)
con $x \in \Sigma^*$ e $y \in \Delta^*$
L_1 proiezione di ρ sulla prima comp.
L_2 proiezione di ρ sulla seconda comp.

$$\rho = \{(x, y), ...\} \subseteq \Sigma^* \times \Delta^*$$
$$L_1 = \{x \in \Sigma^* \mid \text{per qualche } y : (x, y) \in \rho\}$$
$$L_2 = \{y \in \Delta^* \mid \text{per qualche x} : (x, y) \in \rho\}$$

oppure possiamo considerare che:

UNA TRADUZIONE È UNA FUNZIONE τ
(in genere a più valori)

La funzione di traduzione τ può essere resa totale aggiungendo un valore "errore" per le stringhe che non ha senso tradurre.

$$\tau : \Sigma^* \rightarrow \text{parti finite di } \Delta^*;$$
$$\tau(x) = \{y \in \Delta^* \mid (x, y) \in \rho\}$$
$$L_2 = \tau(L_1) = \{y \in \Delta^* \mid \exists x \in \Sigma^* : y \in \tau(x)\}$$
$$\tau^{-1}(y) = \{x \in \Sigma^* \mid y \in \tau(x)\}$$

A seconda delle proprietà matematiche della funzione si hanno i seguenti casi di traduzione:

a) totale: ogni stringa di alfabeto sorgente ha un'immagine
b) a un solo valore (univoca): nessuna stringa sorgente ha due diverse immagini
c) a più valori: una stringa sorgente ammette più traduzioni
d) iniettiva: stringhe sorgente diverse hanno immagini diverse (a ogni stringe di alfabeto pozzo corrisponde al più una stringa di alfabeto sorgente, la traduzione inversa è dunque univoca)
e) suriettiva: una funzione è suriettiva quando l'immagine coincide con il codominio (ogni stringa di alfabeto pozzo è immagine di almeno una stringa del codominio)
f) biiettiva (biunivoca): vi è una corrispondenza uno a uno tra le stringhe sorgente e pozzo

Nozione di "decompilatore": interessante ma solo in senso teorico

2. TRASLITTERAZIONI

Si definisce la traduzione attraverso la ripetizione, carattere per carattere, di trasformazioni puntuali.

La trasformazione più semplice è la TRASLITTERAZIONE (omomorfismo). Ogni carattere sorgente è trascodificato in un corrispondente carattere pozzo (o più in generale in una stringa).

La traduzione risulta univoca ma non sempre è univoca la inversa. Se l'omomorfismo ha cancallature, l'inversa non è mai univoca (e.g., "quadratini" per lettere greche nelle stampanti…).

Se la funzione inversa è univoca, la corrispondenza tra la stringa sorgente e la stringa pozzo è biunivoca, quindi è possibile risalire dalla seconda alla prima.

La traslitterazione è un processo che non considera il contesto della lettera via via tradotta quindi è un metodo insufficiente per la compilazione dei linguaggi tecnici.

3. TRADUZIONI REGOLARI

Per definire una relazione di traduzione si può sfruttare l'approccio delle espressioni regolari: gli operatori di unione, concatenamento e stella si applicano a coppie di stringhe (sorgente e pozzo).

ESEMPIO – Traslitterazione coerente di un operatore

Il testo sorgente è una lista di numeri unari separati dal segno di divisione.
La traduzione traslittera il segno di divisione nel segno : oppure $\div$ ma sempre nello stesso modo all'interno di una stessa stringa.
I numeri sono scritti in notazione unaria.

$$\Sigma = \{1, /\} \quad \Delta = \{1, :, \div\}$$

$$(3/5/2\ ,\ 3:5:2)\ ,\ (3/5/2\ ,\ 3 \div 5 \div 2)$$

$$(1,1)^{+}((/,:)(1,1)^{+})^{*} \cup (1,1)^{+}((/,\div)(1,1)^{+})^{*}$$

$$\left(\frac{1}{1}\right)^{+}\left(\frac{/}{:}\left(\frac{1}{1}\right)^{+}\right)^{*} \cup \left(\frac{1}{1}\right)^{+}\left(\frac{/}{\div}\left(\frac{1}{1}\right)^{+}\right)^{*}$$

$$\frac{1\ /\ 1\,1}{1 \div 1\,1} \qquad (1/11,\ 1 \div 11)$$

espr.del linguaggio regolare

coppia ottenuta per proiezione

DEFINIZIONE – Espressione regolare o razionale di traduzione (e.r.t.)

Un'espressione regolare o razionale di traduzione (e.r.t.) *r* è un'espressione regolare con gli operatori di unione, concatenamento e stella (e croce) i cui termini sono coppie (u, v), di stringhe anche vuote, di alfabeto rispettivo sorgente e pozzo.
Si dice regolare o razionale la relazione di traduzione:

$$\tau = \{(x, y), ...\} \subseteq \textit{ parti finite di } (\Sigma^* \times \Delta^*)$$

definita nel seguente modo:

$Z \subset \Sigma^* \times \Delta^*$ è l'insieme delle coppie (u, v) che compaiono nell'espressione.

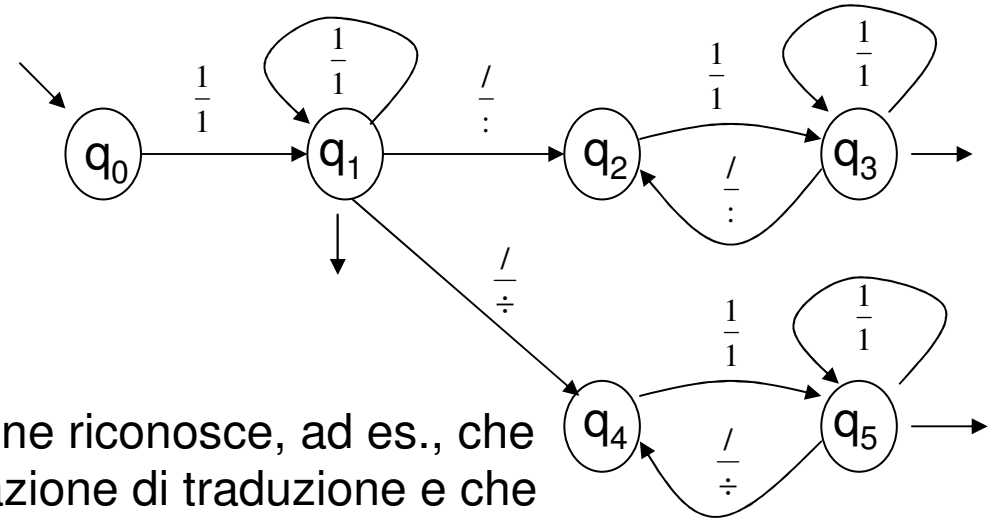
Le coppie (x,y) di stringhe sorgente e pozzo corrispondenti sono tali che:
- esiste una stringa $z \in Z^*$ appartenente all'insieme regolare definito da *r;*
- x e y sono rispettivamente la proiezione di z sulla prima componente e sulla seconda.

L'insieme delle stringhe sorgente definite da una e.r.t. è un linguaggio regolare. Così è per l'insieme della stringhe pozzo.

Non è detto invece che una relazione di traduzione avente linguaggio sorgente e linguaggio pozzo regolari sia definibile mediante una e.r.t. (esempio banale: $\tau(x)=x^R$, $x \in \Sigma^*$, dominio e codominio sono regolari ma la traduzione non è regolare)

AUTOMA RICONOSCITORE A DUE INGRESSI

Se si pensa l'insieme Z delle coppie usate dalla e.r.t. come alfabeto terminale, è semplice associare alla traduzione una automa finito.



ESEMPIO precedente (continua)
Il riconoscitore regolare di traduzione riconosce, ad es., che (11/1, 11:1) appartengono alla relazione di traduzione e che (11/1,1:1) non appartengono alla relazione di traduzione

Il concetto di e.r.t. e di riconoscitore a due nastri ha valore come tecnica di specifica della traduzione stessa e come metodo per la costruzione rigorosa di funzioni di traduzione. Permette inoltre di trattare in modo uniforme le traduzioni calcolate dagli analizzatori lessicali e da quelli sintattici.

FORME DELL'AUTOMA A DUE INGRESSI

Quando in un 2I-automa si proiettano le etichette degli archi del grafo sulla prima componente, si ottiene un automa (con un solo ingresso) di alfabeto Σ, che è detto l'**automa d'ingresso soggiacente** alla traduzione e che riconosce il linguaggio sorgente

Nel descrivere il riconoscitore a due ingressi si può supporre, senza perdita di generalità, che ogni sua mossa legga uno, e un solo, carattere del nastro sorgente.

DEFINIZIONE di 2I-automa

Si suppone, senza perdita di generalità, che mossa del 2I-automa legga un carattere dal nastro sorgente e zero o più caratteri (una stringa) dal nastro pozzo

Un automa finito a due ingressi (2I-automa) possiede, come un automa finito a un solo ingresso, un insieme di stati *Q*, lo stato iniziale q_0, un insieme $F \subseteq Q$ di stati finali. La funzione di transizione è $\delta : (Q \times \Sigma \times \Delta^*) \rightarrow \text{parti finite di } Q$

Se $q' \in \delta(q,a,u)$ l'automa, leggendo *a* dal primo nastro e *u* dal secondo, può andare nello stato prossimo q'. La condizione di riconoscimento è analoga a quella degli automi finiti non deterministici.

Facile mostrare che questo modello è riconducibile, mediante aggiunta di opportuni stati, a modello in forma normale, in cui ogni mossa può leggere un carattere da uno solo dei due nastri, e I due tipi di mosse si alternano. Le etichette degli archi sono dei seguenti tipi:

$$\frac{a}{\varepsilon} con\ a \in \Sigma \quad oppure \quad \frac{\varepsilon}{a} con\ a \in \Delta$$

Come per automi a stati finiti, per maggiore concisione, senza aumentare potenza espressiva del modello, si possono etichettare le transizioni con espressioni regolari

Esempio: una sequenza di caratteri *a* seguita da *b* si traduce in *d*, seguita da *c* si traduce in *e*

$$\boxed{\frac{(a)^* b}{d} \mid \frac{(a)^* c}{e}}$$

EQUIVALENZA TRA I MODELLI DELLA TRADUZIONE

Analoga a nota equivalenza tra espressioni regolari e automi finiti

PROPRIETÀ – La famiglia delle relazioni regolari di traduzione e quella delle relazioni riconosciute da un 2l-automa finito (in generale non deterministico) coincidono.

Una e.r.t. definisce un linguaggio regolare R avente come alfabeto l'insieme delle coppie di stringhe

$$(u,v) \equiv \frac{u}{v}, \text{dove } u \in \Sigma^*, v \in \Delta^*$$

Separando i terminali sorgente e pozzo,
si può riformulare rapporto tra linguaggi e traduzioni regolari

TEOREMA DI NIVAT - Le seguenti condizioni sono equivalenti:

1. La relazione di traduzione ρ è regolare con $\rho \subseteq \Sigma^* \times \Delta^*$

2. Esistono un linguaggio regolare R e un alfabeto C e due omomorfismi alfabetici
   $h_1: C \rightarrow \Sigma \cup \{\varepsilon\}$ *e*
   $h_2: C \rightarrow \Delta \cup \{\varepsilon\}$ tali che:
   $$\rho = \{(h_1(z), h_2(z)) \mid z \in R\}$$

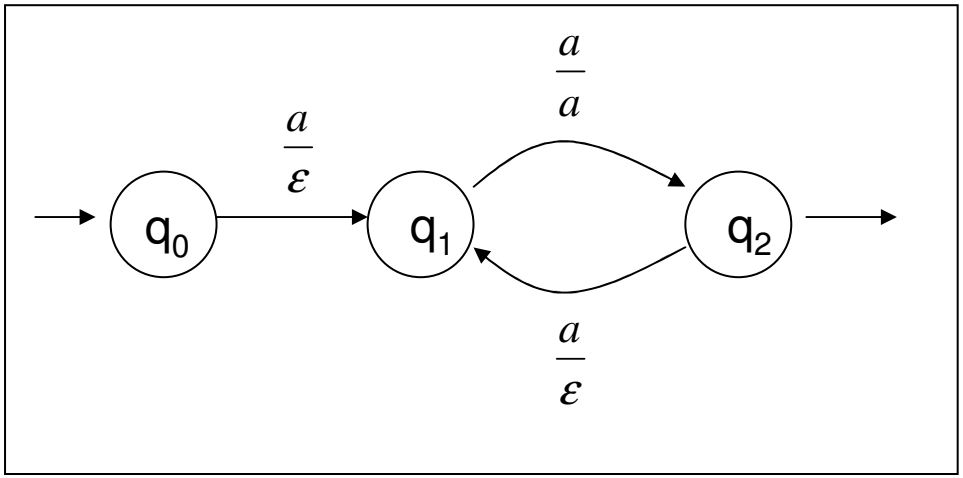
3. Se i due ***alfabeti*** sono ***disgiunti***, esiste un ling. reg. R di alfabeto $\Sigma \cup \Delta$ tale che:
   $$\rho = \{(h_\Sigma(z), h_\Delta(z)) \mid z \in R\}$$
   dove h_Σ *e* h_Δ sono rispettivamente le ***proiezioni*** dell'alfabeto $\Sigma \cup \Delta$ sugli alfabeti sorgente e pozzo.

ESEMPIO: DIVISORE PER DUE La stringa immagine è quella sorgente dimezzata.

Th.Nivat, ramo I dell'enunciato

Relazione di traduzione: $\{(a^{2n}, a^n) \mid n \geq 1\}$ e.r.t. $\left(\frac{aa}{a}\right)^+$

Un 2I-automa equivalente:



Teorema di Nivat (ramo 2 dell'enunciato):

Gli omomorfismi alfabetici producono la traduzione voluta. Ad esempio preso $z = cdcd \in R$ si ha $h_1(z) = aaaa$, $h_2(z) = aa$

$\left(\frac{a}{\varepsilon}\frac{a}{a}\right)^+ \quad \frac{a}{\varepsilon} = c \quad \frac{a}{a} = d \quad C = \{c, d\}$

la e.r.t diventa $R = (cd)^+$ omomorfismi:

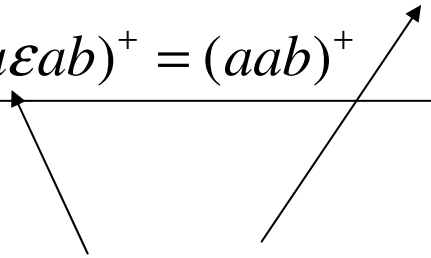
	h_1	h_2
c	a	ε
d	a	a

Teorema di Nivat
(ramo 3 dell'enunciato)

Alfabeto pozzo cambiato per renderlo disgiunto da quello sorgente. La proiezione sugli alfabeti sorgente e pozzo definisce le stringhe che si corrispondono nella traduzione.

$$\Sigma = \{a\}\ \Delta = \{b\} \quad \{(a^{2n}, b^n) \mid n \geq 1\} \quad \left(\frac{a}{\varepsilon}\frac{a}{b}\right)^+$$

$$R \subseteq (\Sigma \bigcup \Delta)^* \quad R = (a\varepsilon ab)^+ = (aab)^+$$

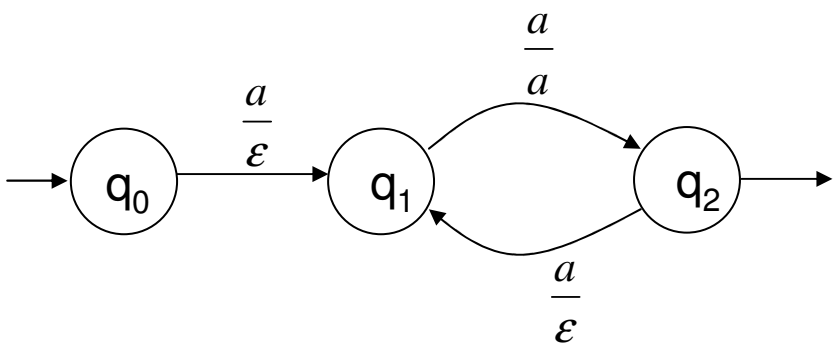


Si cancellano le linee di frazione e si concatenano le stringhe

Es.: h_Σ(aabaab)=aaaa

h_Δ(aabaab)=bb

La corrispondenza tra il modello degli automi finiti e il modello delle grammatiche lineari a destra, permette di scrivere una **grammatica di traduzione** al posto del 2l-automa o delle e.r.t.



$$S \to \frac{a}{\varepsilon} Q_1$$

$$Q_1 \to \frac{a}{a} Q_2$$

$$Q_2 \to \frac{a}{\varepsilon} Q_1 \mid \frac{\varepsilon}{\varepsilon}$$

La definizione mediante grammatica può essere preferita nelle traduzioni sintattiche più generali che hanno come supporto una grammatica libera.

FUNZIONE DI TRADUZIONE E AUTOMA TRASDUTTORE

Studiamo un automa come algoritmo che calcola una funzione di traduzione.
Nuovo modello: **Trasduttore** o **IO-automa** che legge la stringa sorgente dall'ingresso e scrive nell'uscita la stringa immagine.

Tratteremo più approfonditamente le funzioni di traduzione a un solo valore e (tra queste) quelle calcolabili da un automa deterministico.

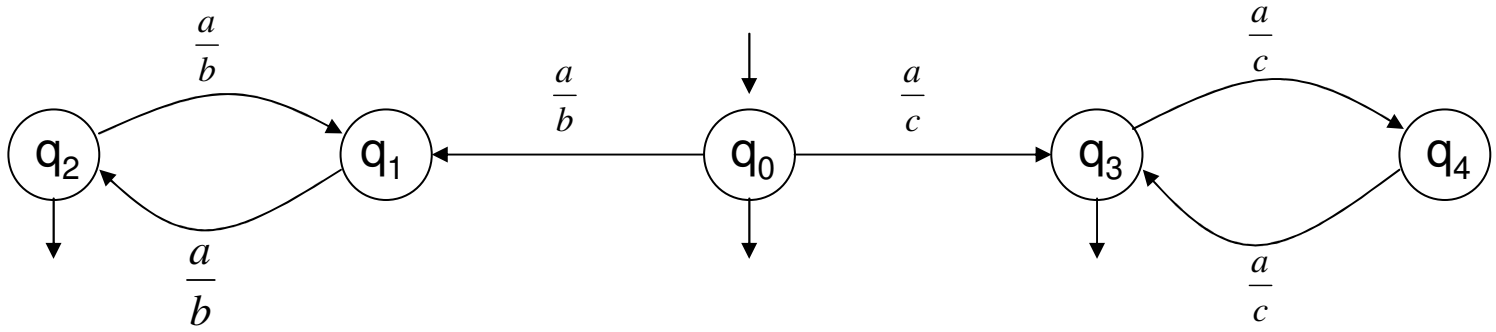
ESEMPIO – Trasduzione non deterministica

mostra una traduzione a un solo valore che non può essere calcolata deterministicamente da un IO-automa finito.

NB: l'automa qui sotto può essere visto come
-2I-automa deterministico, oppure
-I/O automa indeterministico

$$\rho = \left\{(a^{2n}, b^{2n}) \mid n \geq 0\right\} \cup \left\{(a^{2n+1}, c^{2n+1}) \mid n \geq 0\right\}$$

$$\tau(a^n) = \begin{cases} b^n, n \text{ pari}, \\ c^n, n \text{ dispari}. \end{cases} \quad \text{e.r.t.} \quad \left(\frac{a^2}{b^2}\right)^* \cup \frac{a}{c}\left(\frac{a^2}{c^2}\right)^*$$



2I-automa deterministico,
IO-automa non deterministico.

$$\tau(aa) = bb$$

$$\tau(aaa) = ccc$$

A fronte della stringa di ingresso *aa* sono possibili due calcoli $q_0 \rightarrow q_1 \rightarrow q_2$ oppure $q_0 \rightarrow q_3 \rightarrow q_4$ ma solo il primo è valido perché raggiunge uno stato finale.

Il calcolo della funzione di traduzione svolto in modo efficiente scandisce l'ingresso e costruisce la stringa immagine. Al termine della lettura, alla lettura della marca di fine testo, l'algoritmo scrive la stringa che dipende dallo stato finale in cui si trova.

DEFINIZIONE – Modello deterministico di traduttore sequenziale. Un traduttore finito IO-automa sequenziale T è una macchina deterministica così definita. Esso ha un insieme Q di stati, un alfabeto sorgente Σ, uno stato iniziale q_0, un insieme $F \subseteq Q$ di stati finali e tre funzioni, tutte a un solo valore:

1. la funzione di transizione δ, calcola lo stato prossimo
2. la funzione di uscita, η, calcola la stringa da scrivere in concomitanza di una mossa;
3. la funzione finale φ calcola l'ultimo suffisso da concatenare (eventualmente) alla traduzione al termine del calcolo.

I domini della funzione sono: $\delta : Q \times \Sigma \to Q, \quad \eta : Q \times \Sigma \to \Delta^*, \quad \varphi : Q \times \{\text{-|}\} \to \Delta^*$

$$\begin{cases} \delta(q,a) = q' \\ \eta(q,a) = u \end{cases} \qquad q \xrightarrow[u]{a} q' \qquad \varphi(r, \text{-|}) = v$$

τ(x) realizzata da T è il concatenamento di due stringhe prodotte dalla funzione di uscita e da quella finale

$$\left\{ yz \in \Delta^* \mid \exists \text{ un calcolo etichettato } \frac{x}{y} \text{ terminante in } r \in F \wedge z = \varphi(r, \text{-l}) \right\}$$

L'IO-automa sequenziale è deterministico: infatti l'automa d'entrata (Q,Σ,δ,q_0,F) soggiacente a T è deterministico e inoltre la funzione d'uscita e finale sono a un solo valore.
L'IO-automa invece non è deterministico, anche a fronte di un automa d'entrata deterministico, se tra due stati di T vi è un arco del tipo

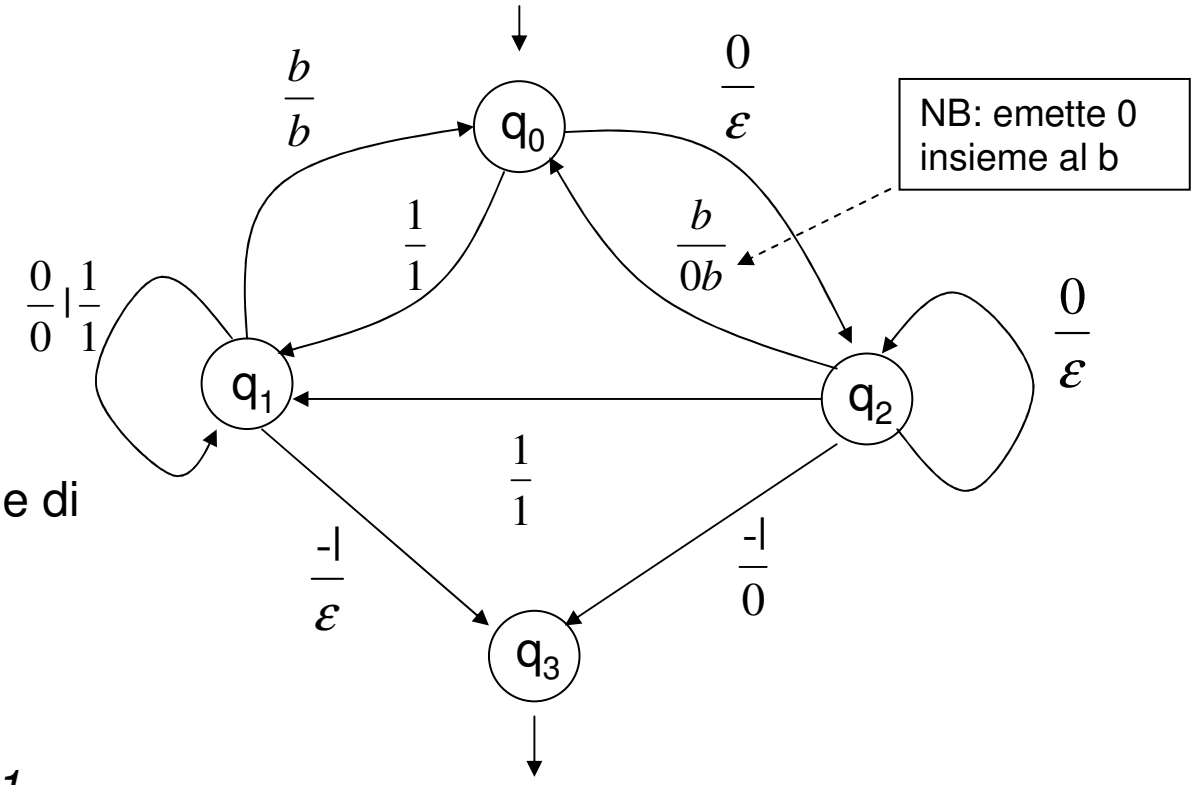
$$\frac{a}{\{b\} \cup \{c\}}$$

Una funzione calcolabile mediante un trasduttore sequenziale di dice **funzione sequenziale**

ESEMPIO – Eliminazione zeri non significativi (NB: non sfrutta emissione stringa finale)

Un testo è una lista di numeri binari interi, separati da uno spazio bianco (*b*). La traduzione (univoca) sopprime gli zeri non significativi

$$\left(\frac{0^{+}b}{0b} \mid \left(\frac{0}{\varepsilon}\right)^{*}\frac{1}{1}\left(\frac{0}{0} \mid \frac{1}{1}\right)^{*}\frac{b}{b}\right)^{*}\left(\frac{0^{+}\text{-}|}{0} \mid \left(\frac{0}{\varepsilon}\right)^{*}\frac{1}{1}\left(\frac{0}{0} \mid \frac{1}{1}\right)^{*}\frac{\text{-}|}{\varepsilon}\right)$$



Il calcolo della traduzione di

00b01-|

attraversa gli stati

$q_0q_2q_0q_2q_1q_3$

e scrive

ε.ε.0b.ε.1.ε=0b1

ESEMPIO

$$\tau(a^n) = \begin{cases} p, \ n \text{ pari}, \\ d, \ n \text{ dispari} \end{cases}$$

L'esempio sfrutta il concetto di funzione finale offerto dal modello di trasduttore sequenziale.

Il trasduttore sequenziale ha due stati, entrambi finali, corrispondenti alla classe di parità della stringa sorgente. Esso non scrive niente nell'effettuare le mosse, poi a secondo dello stato finale raggiunto al termine della lettura, emette *p* oppure *d.*

PROPRIETA`
La composizione di due funzioni sequenziali è ancora una funzione sequenziale.

TRASDUZIONI SEQUENZIALI A DUE PASSATE RIFLESSE – Data una funzione di traduzione, specificata attraverso una espressione regolare o un 2I-automa, non sempre esiste un trasduttore sequenziale che la calcola direttamente. La funzione può sempre essere calcolata operando in due passate: un trasduttore sequenziale produce una stringa intermedia e un secondo trasduttore sequenziale produce la stringa immagine desiderata processando la stringa intermedia da destra a sinistra. Si può trattare così la traduzione di a^n in b^n se n è pari e in c^n se n è dispari.

$$\tau_1 = \left(\frac{a}{a'} \frac{a}{a''} \right)^* \left[\frac{a}{a'} \right]$$

$$\tau_2 = \left(\frac{a''}{b} \frac{a'}{b} \right)^* \bigcup \left(\frac{a'}{c} \frac{a''}{c} \right)^* \frac{a'}{c}$$

$$\tau_2((\tau_1(aa))^R) = \tau_2((a'a'')^R) = \tau_2(a''a') = bb$$

NB: puo` emettere subito b perche` vede subito a”

Fino ad ora abbiamo visto traduzioni regolari, svolte da un algoritmo deterministico con memoria finita che esamina la stringa sorgente in una o due passate. Ma la finitezza della memoria è un limite per molte situazioni che si presentano in informatica.

ESEMPIO - La relazione non è regolare: è necessario immagazzinare tutta la stringa in memoria illimitata prima di emettere la sua immagine.

$$\left\{(x, x^R) \mid x \in (a \mid b)^*\right\}$$

Ispirandosi a Nivat: concatenando tra loro la stringa sorgente x e la sua immagine $y = (x')^R$, che conviene traslitterare nell'alfabeto disgiunto {a', b'}, l'insieme delle stringhe così ottenute non è regolare ma libero.

TRADUZIONI SINTATTICHE PURE

Quando un linguaggio sorgente è definito da una grammatica è naturale considerare delle traduzioni in cui le strutture sintattiche siano tradotte individualmente. L'immagine dell'intera frase può poi essere prodotta componendo le singole traduzioni.

Per questo è necessario mettere in corrispondenza le regole della grammatica sorgente con quelle della grammatica pozzo.

DEFINIZIONE - TRADUZIONE LIBERA DA CONTESTO (o ALGEBRICA)

Una grammatica di traduzione G è una grammatica libera con alfabeto terminale C che è l'insieme di coppie (u,v) di stringhe sorgente e pozzo.

$$G = (V, \Sigma, \Delta, P, S)$$

$$C \subseteq \Sigma^* \times \Delta^* \quad \frac{u}{v}$$

Lo *schema di traduzione sintattica* associato alla grammatica è un insieme di coppie di regole sorgente e pozzo, ottenute eliminando dalle regole della grammatica G rispettivamente i caratteri dell'alfabeto pozzo e dell'alfabeto sorgente

La *relazione di traduzione* definita dalla grammatica è:

$$\tau(G) = \{(x, y) \mid \exists z \in L(G) \wedge x = h_\Sigma(z) \wedge y = h_\Delta(z)\}$$

$$\text{con } h_\Sigma : C \to \Sigma \text{ e } h_\Delta : C \to \Delta$$

Lo schema di traduzione sintattica e la relazione di traduzione sono varianti notazionali che definiscono la stessa relazione di traduzione.

ESEMPIO – Grammatica di traduzione per la riflessione

Grammatica di traduzione:

$$S \rightarrow \frac{a}{\varepsilon} S \frac{\varepsilon}{a} \mid \frac{b}{\varepsilon} S \frac{\varepsilon}{b} \mid \frac{\varepsilon}{\varepsilon}$$

Schema di traduzione associato:

Gramm. sorgente G_1	Gramm. pozzo G_2
$S \rightarrow aS$	$S \rightarrow Sa$
$S \rightarrow bS$	$S \rightarrow Sb$
$S \rightarrow \varepsilon$	$S \rightarrow \varepsilon$

Per ottenere una coppia di stringhe della relazione di traduzione:

a) si costruisce una derivazione

$$S \Rightarrow \frac{a}{\varepsilon} S \frac{\varepsilon}{a} \Rightarrow \frac{a}{\varepsilon}\frac{a}{\varepsilon} S \frac{\varepsilon}{a}\frac{\varepsilon}{a} \Rightarrow$$
$$\Rightarrow \frac{a}{\varepsilon}\frac{a}{\varepsilon}\frac{b}{\varepsilon} S \frac{\varepsilon}{b}\frac{\varepsilon}{a}\frac{\varepsilon}{a} \Rightarrow \frac{a}{\varepsilon}\frac{a}{\varepsilon}\frac{b}{\varepsilon}\frac{\varepsilon}{\varepsilon}\frac{\varepsilon}{b}\frac{\varepsilon}{a}\frac{\varepsilon}{a} = z$$

b) si proietta la frase z sui due alfabeti

$$h_{\Sigma}(z) = aab \qquad h_{\Delta}(z) = baa$$

ESEMPIO – Altro modo di descrivere traduzione $\tau(u)=u^R$

Si sfrutta ***linguaggio libero*** di alfabeto O={a, b, a'. b'}, simile a quello dei palindromi

L={ $u(u^R)'$ | $u \in (a|b)^*$ }={ε, …, abbb'b'a', …} NB (v)' indica v con lettere '-ate

Il linguaggio è libero: produzioni della sua grammatica { S →aSa' | bSb' | ε }

Usando omomorfismi alfabetici h_1 e h_2

Stringhe $z \in L$ traslitterate in coppie di stringhe

della relazione di traduzione ρ

$z \in L \Rightarrow (h_1(z)\ h_2(z)) \in \rho$

	h_1	h_2
a	a	ε
b	b	ε
a'	ε	a
b'	ε	b

e.g., z=abb'a' ==> $(h_1(abb'a')\ h_2(abb'a'))=(ab,ba) \in \rho$

Ciò porta alla seguente proprietà (che sta alle traduzioni libere come il teorema di Nivat sta alle regolari)

PROPRIETÀ – Linguaggio libero e traduzione libera.

Le seguenti condizioni sono equivalenti.

1. La relazione di traduzione è definita da una grammatica di traduzione G
2. Esiste un linguaggio libero L di alf. C e due omormorfismi alf. h_1 e h_2 tali che:
3. Se Σ e Δ sono disgiunti esiste un linguaggio libero L di alf. Σ U Δ tale che:

$$\tau \subseteq \Sigma^* \times \Delta^*$$

$$\tau = \{(h_1(z), h_2(z)) \mid z \in L\}$$

$$\tau = \{(h_\Sigma(z), h_\Delta(z)) \mid z \in L\}$$

ESEMPIO – Traduzione della stringa $x \in (a|b)^*$ nella sua riflessa $y = x^R$

La traduzione si esprime mediante il linguaggio libero dei palindromi

$$L = \left\{ uu'^{R} \mid u \in (a \mid b)^* \wedge u' \in (a' \mid b')^* \right\} = \left\{ \varepsilon, aa', ..., abbb'b'a', ... \right\}$$

omomorfismi alfabetici:

	h_1	h_2
a	a	ε
b	b	ε
a'	ε	a
b'	ε	b

La stringa *abb'a'* $\in L$ si proietta nella coppia di stringhe (ab,ba)

SCRITTURE INFISSE E POLACCHE

Applicazione delle traduzioni libere: conversione di espressioni aritmetiche, logiche, … tra le diverse rappresentazioni utilizzate che differiscono per la posizione degli operatori e per la presenza o meno di parentesi e altri deliminatori.

Grado di un operatore: numero dei suoi argomenti (operatorti unari, binari, …)
Operatore variadico: operatore di grado non fisso
Operatore prefisso / postfisso / infisso / mistofisso (mixfix)

$o_o\ arg_1\ o_2\ arg_2 \ldots o_{n-1}\ arg_n\ o_n$

if arg_1 then arg_2 [else arg_3]

jump_if_false $arg_1\ arg_2$

Forma polacca: priva di parentesi e gli operatori sono tutti prefissi o tutti postfissi.
E.g., istruzioni di codice macchina tutte prefisse (iniziano con codice operativo)

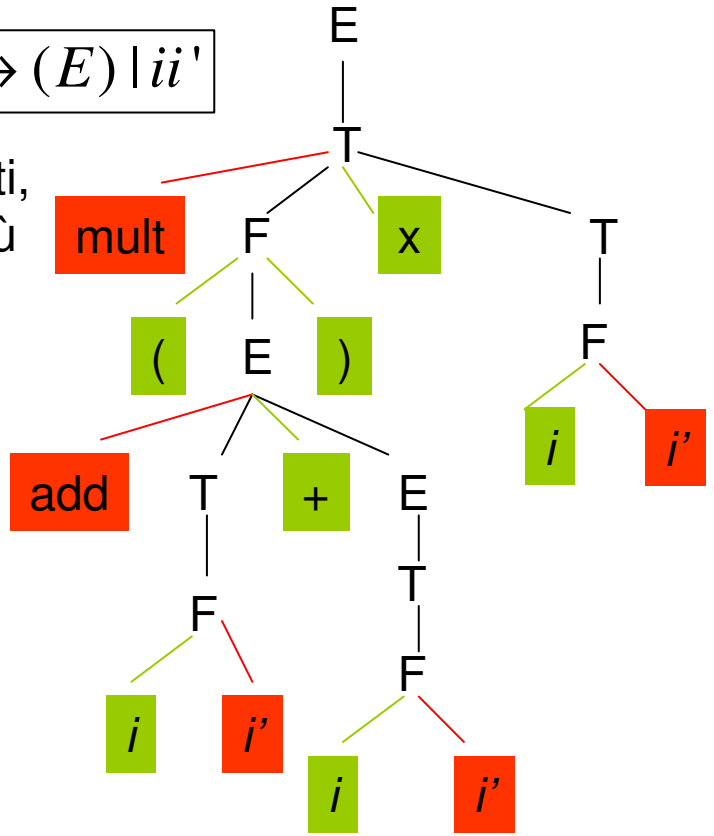
ESEMPIO – Conversione di una espressione aritmetica dalla notazione infissa a quella polacca (traduzione usata spesso nei compilatori per per eliminare le parentesi). Grammatica di traduzione:

$$E \to \text{add } T + E \mid T \quad T \to \text{mult } F \times T \mid F \quad F \to (E) \mid ii'$$

Nota: con alfabeti sorgente e pozzo non sovrapposti, la scrittura della grammatica di traduzione risulta più semplice perché I terminali nelle regole possono essere scritti senza la linea di frazione

$$E \to \text{add } T + E \quad \ldots \quad E \to \frac{\varepsilon}{\text{add}} \; T \frac{+}{\varepsilon} E$$

Gramm sorgente G_1	Gramm pozzo G_2
$E \to T + E$	$E \to \text{add } T\ E$
$E \to T$	$E \to T$
$T \to F \times T$	$T \to \text{mult } F\ T$
$T \to F$	$T \to F$
$F \to (E)$	$F \to E$
$F \to i$	$F \to i'$



(i + i) x i

mult add i’ i’ i’