Aiello, Albano, Attardi, Montanari

Teoria della Computabilità,
Logica, teoria
dei linguaggi formali

Materiali didattici ETS, 1979

Pagine selezionate su Espressioni
Regolari come soluzioni di sistemi di
equazioni tra linguaggi

b) per ogni livello di precedenza deve esistere un'opportuna direzione di associatività per valutare in modo non ambiguo espressioni che coinvolgono sequenze di operatori di uguale precedenza. Ad esempio, l'espressione $a \uparrow b \uparrow c$ ha due valori diversi a seconda che venga interpretata come $((a \uparrow b) \uparrow c)$ oppure $(a \uparrow (b \uparrow c))$. Noi assumeremo per $\uparrow$ e $+$ rispettivamente l'associatività a destra e a sinistra.

Mostriamo come la grammatica data imponga una struttura tale sulle stringhe del linguaggio che la valutazione abbia queste proprietà.

Nel nostro esempio alla radice dell'albero sintattico è associato il valore: $((a + b) + (a \uparrow (c \uparrow b))) + a$.

Entrambe le condizioni a, b sono soddisfatte, infatti:

– L'espressione $(a \uparrow c \uparrow b)$ viene interpretata come $(a \uparrow (c \uparrow b))$. Questo fatto è dovuto alla *ricorsione a destra* presente nella produzione $T \to F \uparrow T$: T che può generare altri elevamenti a potenza è alla destra di $\uparrow$ e quindi viene valutato prima. F invece è alla sinistra di $\uparrow$ e non può generare altri operatori.

– L'espressione $(a + b + a \uparrow c \uparrow b)$ viene interpretata come $((a + b) + a \uparrow c \uparrow b)$ per effetto della *ricorsione a sinistra* presente nella produzione $E \to E + T$. Se ci sono più somme adiacenti, per effetto della ricorsione a sinistra, essi vengono valutati da sinistra a destra.

– L'espressione $(a \uparrow c \uparrow b + a)$ viene interpretata come $((a \uparrow c \uparrow b) + a)$ rispecchiando così la precedenza di $\uparrow$ su $+$. Questa precedenza è espressa dalla produzione $E \to E + T$: prima di applicare $+$ bisogna valutare T che può generare $\uparrow$. In generale se esiste una produzione del tipo $X \to X \rho Y$ e Y può generare l'operatore τ allora τ ha precedenza sull'operatore ρ.

12.9. Sistemi di equazioni tra linguaggi

Nel seguito vedremo come i linguaggi possano essere descritti anche come soluzioni di sistemi di equazioni. A questo scopo presentiamo qui alcune definizioni e osservazioni.

Un sistema di equazioni fra linguaggi su un alfabeto A è detto in forma normale quando è del tipo:

$$(1) \qquad \begin{aligned} X_1 &= F_1(X_1, X_2, \ldots, X_n) \\ &\vdots \\ X_n &= F_n(X_1, X_2, \ldots, X_n) \end{aligned}$$

dove le F_i sono funzioni di $U^n \to U$ con $U = 2^{A^*}$ (l'insieme dei linguaggi su A) e X_i variabili su U.

Chiameremo soluzione del sistema (1) una n-upla di linguaggi $(L_1, \ldots, L_n)$ che soddisfi le equazioni del sistema, cioè:

$$\begin{aligned} L_1 &= F_1(L_1, \ldots, L_n) \\ &\vdots \\ L_n &= F_n(L_1, \ldots, L_n) \end{aligned}$$

In generale possono esistere una, più di una o nessuna soluzione di un sistema. Ad esempio per l'equazione:

$$X = XX + \{aa\}$$

i linguaggi

$$L_k = \{a^{2n} \mid n \geq 0\} + \{a^{2n+k} \mid n \geq 0\}$$

sono tutti soluzioni per ogni valore di k.

Una soluzione $(\bar{L}_1, \bar{L}_2, \ldots, \bar{L}_n)$ si dice minima se per ogni altra soluzione $(L_1, L_2, \ldots, L_n)$ si ha che

$$\bar{L}_1 \subseteq L_1, \ \bar{L}_2 \subseteq L_2, \ldots, \bar{L}_n \subseteq L_n$$

Naturalmente la soluzione minima, se esiste, è unica. Nell'esempio precedente la soluzione minima è

$$\bar{L} = \{a^{2n} \mid n \geq 1\}$$

Per semplicità consideriamo ora il caso di una sola equazione in forma normale: tutto quanto diremo può essere esteso al caso generale considerando un sistema come un'unica equazione in forma vettoriale.

Una soluzione dell'equazione

$$X = F(X)$$

è dunque un punto fisso di $F : U \to U$ in quanto deve essere $L = F(L)$.

Nel caso in cui F sia una funzione continua, cioè per ogni sequenza di linguaggi

$$L_1 \subseteq L_2 \subseteq L_3 \subseteq \ldots$$

si abbia che

$$F\left(\sum_{i=1}^{\infty} L_i\right) = \sum_{i=1}^{\infty} F(L_i)$$

allora vale il teorema del punto fisso:

Teorema 12.8.1.

Se F è una funzione continua, allora il suo minimo punto fisso $\bar{L}$ esiste ed è dato da

$$\bar{L} = \sum_{i=1}^{\infty} F^i(\phi)$$

dove

$$F^i(\phi) = \underbrace{F(F(\ldots(F(\phi)\ldots))}_{\text{i volte}}$$

Sono ad esempio funzioni continue quelle ottenute per composizione delle operazioni tra linguaggi definite precedentemente è cioè somma, concatenazione e iterazione.

Esempio 12.13. Le seguenti funzioni sono continue:

$$\begin{aligned} F_1(X) &= \{a\} + X \\ F_2(X) &= XX + \{aa\} \\ F_3(X) &= \{a\}X + X^*(\{b\} + X\{a\}X) \end{aligned}$$

Il minimo punto fisso di F_1, cioè la soluzione minima di $X = F_1(X)$ è dato da:

$$\begin{aligned} \bar{L}_1 &= \sum_{i=1}^{\infty} F_1^i(\phi) \\ &= F_1(\phi) + F_1(F_1(\phi)) + F_1(F_1(F_1(\phi))) + \ldots \\ &= \{a\} + (\{a\} + \{a\}) + (\{a\} + (\{a\} + \{a\})) + \ldots \\ &= \{a\} \end{aligned}$$

Notiamo che anche $\{a, b\}$, ad esempio, è punto fisso di F_1 in quanto:

$$F_1(\{a, b\}) = \{a\} + \{a, b\} = \{a, b\}$$

e che in effetti $\overline{L}_1 = \{a\} \subseteq \{a, b\}$

12.10. Grammatiche come equazioni fra linguaggi

Riprendendo le osservazioni fatte alla fine del paragrafo 12.5 notiamo come i simboli non terminali di una grammatica rappresentino quelle che avevamo chiamato "categorie sintattiche", cioè insiemi di stringhe che hanno una caratteristica comune rispetto alla sintassi del linguaggio. Si può allora pensare che con un non terminale si indichi un linguaggio. Ogni produzione poi esprime una relazione tra tali linguaggi che ne precisa la struttura in termini di concatenazione di altri linguaggi e simboli terminali. Nell'Esempio 12.12 il non terminale E rappresenta l'insieme delle espressioni, T l'insieme dei termini e la prima produzione $E \to E + T$ indica che le espressioni possono essere formate da una espressione seguita dal simbolo +, seguita da un termine. Se consideriamo il simbolo + come rappresentazione del linguaggio $\{+\}$ potremo dire che ogni produzione determina una equazione tra due linguaggi, ciascuno ottenuto per concatenazione di altri linguaggi: $E = E + T$. Se più produzioni hanno la stessa parte sinistra, allora si formerà un'unica equazione che ha sulla destra l'unione delle parti destre, ad esempio alle produzioni

$$C \to aCA \mid bCB$$

corrisponderà l'equazione

$$C = aCA + bCB$$

Ad una grammatica (libera) completa corrisponde quindi un sistema di equazioni tra linguaggi sull'alfabeto dei simboli terminali, in cui le variabili sono i simboli non terminali della grammatica.

A questo punto è naturale chiedersi che relazione vi sia tra il linguaggio generato da una grammatica e le soluzioni del corrispondente sistema di equazioni.

Consideriamo pertanto una grammatica libera $G = (V, T, S_0, P)$ e sia $V - T = \{S_0, S_1, \ldots, S_n\}$ l'insieme dei simboli non terminali. Associamo a ciascuno di questi il linguaggio $\overline{S}_i$ da essi generato:

$$\overline{S}_i = \{w \mid w \in T^*, s_i \xrightarrow[G]{*} w\}$$

che costituiscono l'insieme dei linguaggi definito da G.

Tra questi, il linguaggio L(G) è naturalmente $\overline{S}_0$. La risposta al nostro quesito è data dal seguente teorema:

Teorema 12.9.1.

L'insieme dei linguaggi generato da una grammatica libera è la soluzione minima del corrispondente sistema di equazioni.

Dimostrazione. Che i linguaggi $\overline{S}_i$ costituiscano una soluzione risulta direttamente dalla loro definizione. Per far vedere che è la minima osserviamo che per ogni produzione $S_i \to \beta$ di G deve esistere nel sistema una equazione del tipo $S_i = \beta + t$ per qualche termine t. Da tale equazione si ricava quindi la disequazione $S_i \supseteq \beta$. Consideriamo ora una parola w tale che $S_i \to \beta \to \ldots\ldots \to w$. Si vede subito (per induzione) che effettuando sostituzione di disequazioni corrispondentemente all'applicazione di produzioni, si ottiene che $S_i \supseteq \beta \supseteq \ldots\ldots \supseteq w$ e quindi che ogni linguaggio soluzione per S_i, visto che deve soddisfare tutte le disequazioni, deve contenere w. Questo deve valere per ogni w tale che $S_i \xrightarrow[G]{*} w$, e quindi $S_i \supseteq \overline{S}_i$. ∎

Problemi

1) Definire le grammatiche regolari che generano:
a) identificatori di lunghezza arbitraria che iniziano con una lettera (come in ALGOL)
b) identificatori lunghi al più sei caratteri e che iniziano per I, J, K, L, M o N (come le variabili intere in FORTRAN)
c) le costanti reali come in PL/1 o FORTRAN, ad esempio −10.8, 3.14, 3., 5.5 E−10.

2) Sia G una grammatica con produzioni del tipo $A \to wB$ oppure $A \to w$, con $A, B \in (V - t)$ e $w \in T^+$. Dimostrare che L(G) è un linguaggio tipo 3.

3) Descrivere il linguaggio generato dalle produzioni:
a) $S \to aAS/a$
   $A \to SbA/SS/ba$
b) $S \to bSS/a$

4) Per ognuna delle seguenti grammatiche dare l'albero di derivazione, se esiste, per le parole: aab, abcd, aabb

G1: $S \rightarrow aS \mid Sb \mid a \mid c \mid cd$

G2: $S \rightarrow AB$
$A \rightarrow aa \mid aA$
$aB \rightarrow a$
$B \rightarrow bcd$
$AB \rightarrow aB$

5) Definire le grammatiche libere che generano:
a) tutte le possibili sequenze di parentesi bilanciate
b) le istruzioni di assegnamento in FORTRAN
c) $\{a^m b^n c^m \mid m, n > 0\}$
d) tutti i numeri naturali divisibili per 4 e rappresentati nella notazione binaria
e) come caso d) ma i numeri rappresentati in notazione decimale.

6) Dimostrare che il linguaggio generato dalla grammatica dell'Esempio 12.6 è $\{a^n b^n c^n \mid n \geq 1\}$.

7) Definire le grammatiche tipo 1 che generano:
a) $\{ww \mid w \in \{a, b\}^+\}$
b) $\{w \mid w \in \{a, b, c\}^+$ e w contiene lo stesso numero di a, b e c$\}$

8) Dimostrare che ogni linguaggio tipo 1 è ricorsivo.

9) Descrivere una rappresentazione di una grammatica libera su alfabeti V e T con una stringa di caratteri. E' possibile dare una grammatica libera che generi tutte e sole le stringhe che rappresentano grammatiche libere?

13 I LINGUAGGI REGOLARI

Abbiamo visto come i linguaggi regolari, su un alfabeto A, sono i sottoinsiemi di A* generati da grammatiche regolari. In questo capitolo discuteremo altri modi per rappresentare questi linguaggi attraverso il formalismo delle espressioni regolari, gli automi a stati finiti deterministici e quelli non deterministici. Infine mostreremo la completa equivalenza di queste rappresentazioni dimostrando la seguente catena di implicazioni:

gram. reg. → espr. reg. → automa non det. → automa det. → gram.reg.

13.1. Le espressioni regolari

Le espressioni regolari su un alfabeto $A = \{a_1, a_2, \ldots, a_n\}$ costituiscono un formalismo per la rappresentazione di linguaggi allo stesso modo in cui le MdT e il formalismo di Mc Carthy sono formalismi per rappresentare funzioni.

Per descrivere la sintassi di tale formalismo, possiamo servirci sta-

volta di una grammatica, che, come si vede, è di tipo 2:

$$G = (\{E\} \cup T, T, E, P) \quad \text{dove}$$
$$T = \{\lambda, \phi, +, *, \cdot,), (\} \cup A$$

e P è costituito dalle produzioni:

$$E \to (E + E) \mid (E \cdot E) \mid (E)^*$$
$$E \to \lambda \mid \phi \mid a_1 \mid a_2 \mid ..... \mid a_n$$

Il linguaggio generato da questa grammatica costituisce l'insieme delle espressioni regolari sull'alfabeto A.

Esempio 13.1. Sono espressioni regolari sull'alfabeto $A = \{a, b\}$:

1) $(a \cdot b)$
2) $((a \cdot a) + b)$
3) $(\phi + ((a)^* \cdot b))$

Per semplificare la scrittura adottiamo la convenzione di dare precedenza nell'ordine agli operatori $*$, $\cdot$, $+$ in modo di evitare qualche parentesi, di sottintendere il $\cdot$ e di indicare $\underbrace{e \ldots\ldots e}_{n \text{ volte}}$ con e^n. Scriveremo quindi:

1′) ab
2′) $a^2 + b$
3′) $\phi + a^*b$

La semantica di una espressione regolare e è il linguaggio L(e) ottenuto in base alle seguenti regole.

$$L[\phi] = \phi$$
$$L[\lambda] = \{\lambda\}$$
$$L[a_i] = \{a_i\}$$
$$L[e_1 + e_2] = L[e_1] \cup L[e_2]$$
$$L[e_1 \cdot e_2] = L[e_1] \cdot L[e_2]$$
$$L[e^*] = L[e]^*$$

Esempio 13.2. Per le espressioni dell'esempio precedente:

1) $L[ab] = L[a] \,.\, L[b] = \{a\} \,.\, \{b\} = \{ab\}$
2) $L[a^2 + b] = L[a^2] + L[b] = \{aa\} + \{b\} = \{aa, b\}$
3) $L[\phi + a^* b] = \phi + \{a\}^* \{b\} = \{b, ab, a^2 b, a^3 b, ...\}$

Sulle espressioni regolari si possono dimostrare alcune proprietà di tipo algebrico che qui riportiamo:

(a)	$e_1 (e_2 + e_3) = e_1 e_2 + e_1 e_3$	(g)	$(e_1 + e_2) e_3 = e_1 e_3 + e_2 e_3$
(b)	$e_1 + (e_2 + e_3) = (e_1 + e_2) + e_3$	(h)	$\phi^* = \lambda$
(c)	$e_1 (e_2 e_3) = (e_1 e_2) e_3$	(i)	$e^* + e = e^*$
(d)	$e\lambda = \lambda e = e$	(j)	$(e^*)^* = e^*$
(e)	$\phi e = e\phi = \phi$	(k)	$(e_1 + e_2)^* = (e_1^* e_2^*)^*$
(f)	$e + e = e$	(l)	$e + \phi = e$

Si osservi però che queste identità non riguardano le espressioni ma i linguaggi che esse rappresentano, analogamente a quanto avviene nel formalismo delle MdT dove si possono avere MdT diverse (sintatticamente) che calcolano la stessa funzione (e quindi "semanticamente" uguali).

13.2. Relazione tra espressioni regolari e grammatiche regolari

Data la particolare forma delle produzioni di una grammatica lineare destra, il sistema di equazioni ad esso associato risulta di tipo lineare, cioè ha la seguente forma:

$$(1) \qquad \begin{aligned} X_1 &= A_{10} + A_{11} X_1 + A_{12} X_2 + ... + A_{1n} X_n \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ X_n &= A_{n0} + A_{n1} X_1 + A_{n2} X_2 + ... + A_{nn} X_n \end{aligned}$$

Ora ci occuperemo quindi di sistemi di questo genere cercando dapprima un metodo per risolvere l'equazione lineare:

$$(2) \qquad X = AX + B$$

dove A e B sono linguaggi noti.

La risposta ci viene data dal seguente

Teorema 13.2.1.

La soluzione minima dell'equazione (2) è data da

$$X = A^*B$$

Dimostrazione. La soluzione minima dell'equazione è il minimo punto fisso della funzione

$$F(X) = AX + B$$

e quindi è data, in base al teorema 12.8.1, da:

$$\bar{L} = \sum_{i=1}^{\infty} F^i(\phi)$$

Vediamo che:

$$F^1(\phi) = B$$
$$F^2(\phi) = F(F(\phi)) = B + AB = (\lambda + A)B$$
$$F^3(\phi) = B + A(B + AB) = B + AB + A^2B = (\lambda + A + A^2)B$$
. . . . . .
$$F^i(\phi) = B + A(B + AB + \ldots.. + A^{i-1}B) = (\lambda + A + \ldots.. + A^i)B$$
$$\sum_{i=1}^{\infty} F^i(\phi) = (\lambda + A + A^2 + \ldots..)\, B = A^*B$$ ■

Teorema 13.2.2.

Se $\lambda \notin A$ la soluzione dell'equazione (2) è unica.

Dimostrazione. Se $\lambda \notin A$, per ogni altra soluzione L si ha che $L \subseteq A^*B = \bar{L}$ e quindi, poichè quest'ultima è quella minima, si avrà che $L = A^*B$. Infatti, se $w \in L$ e $|w| = k$; siccome abbiamo che

$$L = AL + B \qquad \text{si avrà che}$$
$$L = A(AL + B) + B = A^2L + AB + B$$

e ripetendo k volte queste sostituzioni otteniamo

$$L = A^{k+1}L + A^kB + A^{k-1}B + \ldots.. + AB + B$$

Se $\lambda \notin A$, allora tutte le stringhe di $A^{k+1}X$ hanno almeno lunghezza $k + 1$.

Perciò, se $w \in L$ deve essere che $w \in A^iB$ per qualche i

$k \geqslant i \geqslant 0$, e quindi $w \in A^*$ dato che $A^iB \subseteq A^*B$. ■

Se la condizione $\lambda \notin A$ non è soddisfatta, A^*B è comunque la soluzione minima (teorema 13.2.1) ma anche $L = A^*(B + C)$ è una soluzione, qualunque sia C, e quindi si hanno infinite soluzioni.

Vediamo ora come si risolve il sistema (1).

Ricaviamo X_1 dalla prima equazione sfruttando il teorema 13.2.1:

$$X_1 = A_{11}^*(A_{10} + A_{12}X_2 + \ldots.. + A_{1n}X_n)$$

e sostituiamo nelle successive. Quindi semplifichiamo:

$$X_2 = A_{20} + A_{21}A_{11}^*A_{10} + (A_{22} + A_{21}A_{11}^*A_{12})X_2 + \ldots..+(A_{2n} + A_{21}A_{11}^*A_{1n})X_n$$
. . . . . .
$$X_{11} = A_{n0} + A_{n1}A_{11}^*A_{10} + (A_{n2} + A_{n1}A_{11}^*A_{12})X_2 + \ldots..+(A_{nn} + A_{n1}A_{11}^*A_{1n})X_n$$

In tal modo otteniamo un sistema di $n - 1$ equazioni in $n - 1$ variabili.

Possiamo perciò iterare il procedimento fino ad ottenere una espressione senza incognite per il linguaggio X_n. Le espressioni per $X_1, \ldots, X_{n-1}$ si ottengono poi per sostituzione(*).

I linguaggi soluzione minima di un sistema lineare sono quindi ottenuti dai linguaggi coefficienti A_{ij} per mezzo di operazioni di $+$, . e $*$. Possiamo perciò affermare che

Teorema 13.2.3.

I linguaggi generati da grammatiche regolari sono rappresentabili mediante espressioni regolari.

Dimostrazione. Da quanto detto, se i linguaggi coefficienti di un sistema lineare possono essere rappresentati mediante espressioni regolari, ciò sarà vero anche per i linguaggi soluzione minima. Nel caso di un sistema associato ad una grammatica regolare questi sono o simboli terminali o ϕ, quindi tale condizione è soddisfatta. Pertanto il linguaggio

(*) Si può dimostrare (vedi Appendice) che la soluzione così ottenuta è proprio quella minima.

generato dalla grammatica, che è il linguaggio corrispondente al simbolo iniziale della grammatica nella soluzione minima del sistema a questa associato, è rappresentabile mediante una espressione regolare. ■

Osserviamo infine che se $\lambda \notin A_{ij}$, allora la soluzione del sistema lineare (1) è unica in quanto ad ogni passo del procedimento ci ritroviamo con sistemi lineari in cui i coefficienti delle variabili non contengono λ e quindi vale la condizione di unicità del teorema. Questa condizione è chiaramente soddisfatta nel caso il sistema sia quello associato ad una grammatica lineare.

Esempio 13.3. Dalla grammatica $G = (\{X, Y, a, b\}, \{a, b\}, X, P)$ dove P sono le produzioni:

$$X \to aX \mid bY \mid a$$
$$Y \to bX \mid bY \mid b$$

si ottiene il sistema:

$$X = aX + bX + a$$
$$Y = bX + bY + b$$

Ricaviamo Y dalla seconda equazione e sostituiamo nella prima:

$$Y = b^*(bX + b)$$
$$X = (a + bb^*b)X + a + bb^*b$$

Quindi abbiamo che

$$X = (a + bb^*b)^*(a + bb^*b)$$

è l'espressione che rappresenta il linguaggio generato da G.

13.3 Procedura di riconoscimento

Proponiamoci di risolvere il seguente problema: trovare una procedura che, per ogni stringa, o parola, su un assegnato alfabeto, stabilisca se essa appartiene o no ad un certo linguaggio. Come si è già accennato, è possibile risolvere questo problema ogni qualvolta il linguaggio assegnato sia generato da una grammatica dipendente dal contesto o da un'altra meno potente. Ora, comunque, vogliamo vedere più in generale che caratteristiche deve avere una tale procedura iniziando con una descrizione molto semplice.

Con un discorso molto approssimativo possiamo dire che una tale procedura deve in un qualche modo esaminare uno dopo l'altro i caratteri che formano la parola e ad ogni carattere letto modificare un "qualcosa", che chiameremo "stato", in modo che giunti alla fine della parola lo stato risultante indichi se la parola appartiene o non al linguaggio.

Per precisare un po' meglio queste cose diamo una descrizione modellistica di questa procedura di riconoscimento che chiameremo "automa". Il modello consiste di un dispositivo di controllo, con un numero finito di stati, una testina di lettura e un nastro infinito diviso in celle Fig. 13.1. La stringa di ingresso è scritta sul nastro con un carattere per ogni cella, con il carattere iniziale a sinistra, e tutte le altre celle a destra della stringa contengono il carattere bianco. All'inizio del riconoscimento, la testina è posizionata sul carattere iniziale e il controllo è nello stato iniziale.

Supponiamo che ad ogni carattere che si legge corrisponde un passo del funzionamento della procedura. La cosa più semplice che viene in mente è di prescrivere che il passo successivo ad uno considerato consista nella lettura del carattere successivo (il nastro viene esaminato da sinistra a destra) e nel passaggio ad un altro stato, determinato da una *funzione di transizione* esplicitamente assegnata e, naturalmente, definita per ogni coppia (stato, carattere). Esaminata tutta la stringa in ingresso l'automa si arresta e a seconda del suo stato decideremo se la parola appartiene o no al linguaggio.

Questa descrizione fa subito pensare alle Macchine di Turing, ma ci sono delle differenze che, come vedremo, rendono questo tipo di automa meno potente. La differenza fondamentale sta nel fatto che un automa può solo *leggere* dal nastro di ingresso e non può scriverci utilizzando cioè come una memoria esterna illimitata.

Esempio 13.4. Consideriamo un automa con quattro stati che accetta in ingresso parole sull'alfabeto $\{a, b\}$. Lo stato iniziale e finale è q_0 e la funzione di transizione è data dalla tabella in Fig. 13.2. Lo stato iniziale e quello finale sono contrassegnati rispettivamente con una freccia e con un cerchio.

APPENDICE 2

Diamo qui la dimostrazione del fatto che un sistema di equazioni può essere risolto col metodo della eliminazione di variabili.

Teorema A2.1. Consideriamo il sistema di equazioni del tipo:

$$(1) \qquad \begin{cases} x_1 = F_1(x_1, x_2) \\ x_2 = F_2(x_1, x_2) \end{cases}$$

con F_1 e F_2 continue. Indichiamo con $H(x_2)$ il m.p.f. di $F_1(x_1, x_2)$ rispetto ad x_1. Il sistema 1) ha la stessa soluzione minima del sistema:

$$(2) \qquad \begin{cases} x_1 = H(x_2) \\ x_2 = F_2(H(x_2), x_2) \end{cases}$$

Dimostrazione. $H(x_2)$ è continuo per la continuità dell'operazione di m.p.f. e per la continuità di F_1 rispetto al suo secondo argomento. Inoltre, per definizione, per ogni valore a di x_2, $H(a)$ è il m.p.f di $F_1(x_1, a)$ e quindi $H(a) = F_1(H(a), a)$.

Sia $\bar{x}_2$ il m.p.f. di $F_2(H(x_2), x_2)$ e $\bar{x}_1 = H(\bar{x}_2)$. Sempre per definizione di H si ha che

$$F_1(\bar{x}_1, \bar{x}_2) = F_1(H(\bar{x}_2), \bar{x}_2) = H(\bar{x}_2) = \bar{x}_1$$

e per la definizione di $\bar{x}_2$

$$F_2(\bar{x}_1, \bar{x}_2) = F_2(H(\bar{x}_2), \bar{x}_2) = \bar{x}_2$$

Quindi $(\bar{x}_1, \bar{x}_2)$ è anche soluzione del sistema 1).

Sia ora (x'_1, x'_2) un'altra soluzione di 1). Mostriamo come $(\bar{x}_1, \bar{x}_2) \sqsubseteq (x'_1, x'_2)$.

Per definizione $H(x'_2)$ è m.p.f. di $F_1(x_1, x'_2)$ quindi

$$(*) \qquad H(x'_2) \sqsubseteq x'_1$$

dato che $x'_1 = F_1(x'_1, x'_2)$ ed è perciò punto fisso di $F_1(x_1, x'_2)$.

Poniamo $G(x_2) = F_2(H(x_2), x_2)$. Quindi $\bar{x}_2 = \bigsqcup_{i=1}^{\infty} G^i(\phi)$.

Mostriamo per induzione su i come

$$(**) \qquad G^i(\phi) \sqsubseteq x'_2$$

Per $i = 0$ è ovviamente $\phi \sqsubseteq x'_2$. Supponiamo che $G^i(\phi) \sqsubseteq x'_2$ per ualche i. Essendo G continua e quindi monotona si ha:

$$G^{i+1}(\phi) \sqsubseteq G(x'_2) = F_2(H(x'_2), x'_2) \sqsubseteq F_2(x'_1, x'_2) = x'_2$$

Quindi la (**) è vera e questo implica che $\bar{x}_2 \sqsubseteq x'_2$. Infine per la monotonicità di H e per la (*)

$$\bar{x}_1 = H(\bar{x}_2) \sqsubseteq H(x'_2) \sqsubseteq x'_1 \qquad \blacksquare$$

Teorema A2.2. Il sistema 1) ha la stessa soluzione minima del sistema:

$$\begin{cases} x_1 = F_1(x_2, F_2(x_1, x_2)) \\ x_2 = F_2(x_1, x_2) \end{cases}$$

La dimostrazione è ovvia.