Sistemi logici programmabili

Note sulla rappresentazione dei numeri e sull'aritmetica

THE question “What is a number?” is one which has been often asked, but has only been correctly answered in our own time. The answer was given by Frege in 1884, in his “Grundlagen der Arithmetik. Although this book is quite short, not difficult, and of the very highest importance, it attracted almost no attention, and the the definition of number which it contains remained practically unknown until it was rediscovered by the present author in 1901.

In seeking a definition of number, the first thing to be clear about is what we may call the grammar of our inquiry. Many philosophers, when attemting to define number, are really setting to work to define plurality, which is quite a different thing. Number is what is characteristic of numbers, as man is what is characteristic of men. A plurality is not an instance of number, but of some particular number. A trio of men, for example, is an instance of the number 3, and the number 3 is and instance of number; but the trio is not an instance of number. This point of view seem elementary and scarsely worth mentioning; yet it has proved too subtle for the philosophers, with few exceptions.

A particular number is not identical with any collection of terms having that number: the number 3 is not identical with the trio consisting of Brown, Jones, and Robinson. The number 3 is something which all trios have in common, and which distinguishes them form other collections. A number is something that characterizes certain collections, namely, those that have that number”.

Bertrand Russel (1872-1970)

1. Rappresentazione dei numeri naturali

I numeri naturali sono generalmente rappresenti ricorrendo a una “rappresentazione in cifre”.

Si noti che nel seguito, quando sarà necessario esprimere il valore di un particolare numero naturale, si farà riferimento alla consueta rappresentazione in decimale.

In una rappresentazione in cifre, il numero naturale x è rappresentato da un vettore (ovvero da ua lista ordinata) di simboli, dette cifre. Indichiamo la rappresentazione X del numero naturale x con la scrittura:

$$X = (X_{n-1}, X_{n-2}, \dots, X_1, X_0) \qquad (1.1)$$

Il sistema che consente di far corrispondere a numero x una rappresentazione X è detto sistema di numerazione ed è definito dai seguenti elementi:

A. il numero n delle cifre usate per la rappresentazione;
B. l’insieme dei valori numerici D_i che possono essere assunti da ciascuna cifra X_i;
C. una regola di interpretazione che fa corrispondere a un vettore (di cifre uno ed un solo numero naturale.

Si osservi che:

- quando si lavora con carta e penna, il numero di cifre n può essere non pre-definito nel senso che siamo abituati ad aumentare il numero di cifre usate nel corso dei calcoli. Nel contesto in cui stiamo lavorando, il numero di cifre n deve essere definito.
- Non è detto che le singole cifre siano ottenute a partire dallo stesso numero e dallo stesso insieme di simboli. Indichiamo in generale con $|D_i|$ la cardinalità dell’insieme dei valori (ovvero il numero di valori possibili) della cifra X_i.
- Mentre è necessario che ad una rappresentazione corrisponda uno ed un solo numero, non vale in contrario. Lo stesso numero può avere più rappresentazioni distinte e questo può essere utile nella implementazione di algoritmi di calcolo.
  Se accade che ad una rappresentazione corrisponde uno ed un solo numero, la rappresentazione si dice non ridondante.
- Il numero delle rappresentazioni possibili è finito e pari a:

$$k = \prod_{i=0}^{n-1} |D_i| \qquad (1.2)$$

  Se il sistema di numerazione è non ridondante allora k è anche il numero dei numeri distinti che possono essere rappresentati.

I sistemi di numerazione più comunemente usati sono sistemi di numerazione “pesati”, ovvero sistemi per i quali la legge di interpretazione della rappresentazione è la seguente:

$$x = \sum_{i=0}^{n-1} [X_i] w_i \qquad (1.3)$$

dove $[X_i]$ è il valore della cifra nella i-esima posizione è w_i è un intero che costituisce il “peso” per la cifra nella i-esima posizione. I pesi sono in generale diversi per ciascuna posizione i.

L’insieme dei pesi si chiama vettore dei pesi W e si indica con:

$$W = (w_{n-1}, w_{n-2}, \dots, w_1, w_0) \qquad (1.4)$$

I sistemi pesati possono essere del tipo “a radice” se si ha:

$$w_0 = 1 \text{ e } w_i = w_{i-1} r_{i-1} \qquad (1.5)$$

Si noti che nell’equazione precedente abbiamo usato l’espressione “pari all’unità” per evitare di dover usare una rappresentazione particolare per esprimere, appunto l’unità.

Si definisce il vettore delle radici R come:

$$R = (r_{n-1}, r_{n-2}, \dots, r_1, r_0) \qquad (1.6)$$

Si noti che la definizione della radice r_{n-1} non è necessaria per la definizione dei pesi nella (1.5) e pertanto può essere omessa.

Si ha, ovviamente:

$$w_0 = 1;\ w_i = \prod_{j=0}^{i-1} r_j \qquad (1.7)$$

Un sistema pesato a radice è a radice fissa se r_i=r per ogni i. In questo caso si ha:

$$\begin{gathered} W = (r^{n-1}, r^{n-2}, \dots, r, 1) \\ x = \sum_{i=0}^{n-1} [X_i] r^i \end{gathered} \qquad (1.8)$$

Un sistema pesato a radice si dice canonico se per ogni cifra $|D_i|=r_i$ (ovvero se il numero di valori previsti per la i-esima cifra è pari a r_i) e se questi valori sono tutti i valori da zero a r_i-1.

In generale (almeno nell’ambito di questo corso) ci limiteremo a sistemi di numerazione pesati a radice canonici e a radice fissa, altrimenti conosciuti come sistemi di numerazione convenzionali a radice “*r*”.

Il sistema decimale che abbiamo imparato sin dalle elementari e il sistema di numerazione “binario naturale” appartengono a questa categoria.

Ai sistemi di numerazione convenzionali a radice si possono facilmente estendere gli algoritmi per la somma, la differenza, la moltiplicazione e la divisione imparati alle elementari.

Si ricordi che gli algoritmi di calcolo non si applicano ai “numeri”, ma alle loro rappresentazioni. In particolare, quando di esegue una somma si opera algoritmicamente sulla rappresentazione dei due addendi per ottenere la rappresentazione del numero somma.

2. Rappresentazione dei numeri relativi

Uno dei metodi più diffusi per la rappresentazione di numeri relativi è quella di operare una prima trasformazione matematica sui numeri relativi (ristretti a un certo intervallo di valori) per trasformali in numeri naturali. Ad essere "rappresentati" sono quindi i numeri naturali così ottenuti e opportuni algoritmi che operano su queste rappresentazioni consentono, sotto opportune condizioni, di ottenere le rappresentazioni del numero che corrisponde ad operazioni matematiche sui numeri relativi di partenza.

Una delle rappresentazioni di questo tipo più utilizzate è la rappresentazione "in complemento alla radice". Per arrivare a discutere di tale rappresentazione, sono necessarie alcune premesse.

Quoziente e resto della divisione intera per un numero positivo.

Sia x un numero **intero relativo** e C una costante intera non negativa. Si definiscono quoziente e resto della divisione intera di x per C i numeri interi q e r per i quali si può scrivere:

$$x = q \times C + r \qquad (2.1)$$

rispettando la condizione:

$$0 \leq r < C \ (2.2)$$

Si può dimostrare che la coppia di interi che soddisfa le Eq. 2.1 e 2.2 è unica.

La condizione (2.2) è fondamentale perché q e r siano univocamente definiti. Se la condizione (2.2) viene rimossa, esistono infinite coppie di valori di q e di r che soddisfano la (2.1). Per esempio, sia x=23 e C=5. La (2.1), senza la condizione (2.2) è soddisfatta per le infinite coppie di numeri (q,r) :

…..... (-2,33), (-1,28), (0,23), (1,18), (2,13), (3,8), (4,3), (5,-2), (6,-7), (7,-13)........

Solo la coppia (4,3), però, soddisfa sia la (2.1) sia la (2.2).

Se x è un numero non negativo, l'algoritmo per il calcolo di q e di r è quello imparato alle scuole elementari, se x è un numero negativo, ci si può ricondurre all'algoritmo valido nel caso di x positivi. Sia x<0. Si ha allora che y=-x è un intero positivo e possiamo facilmente determinare con l'algoritmo della divisione elementare i numeri q_y e r_y che soddisfano la (2.1) e la (2.2). Si ha quindi:

$$y = q_y \times C + r_y 0 \leq r_y < C \qquad (2.3)$$

ora, poiché x=-y, si ha certamente:

$$x = -y = (-q_y) \times C - r_y \ (2.4)$$

Possiamo ora individuare due casi distinti:

- r_y=0. In questo caso, le coppie di valori (-q_y) e (-r_y =0) sono il quoziente e il resto della divisione intera di x per C in quanto verificano la (2.2).
- r_y>0. In questo caso, le coppie di valori (-q_y) e (-r_y) *non sono* il quoziente e il resto della divisione intera di x per C in quanto non verificano la (2.2). Se però sommiamo e sottraiamo C al secondo membro della (4) si ha (si ricordi che per ipotesi r_y>0):

$$x = -y = (-q_y) \times C - r_y + C - C = -(q_y + 1) \times C + (C - r_y) 0 < (C - r_y) < C \qquad (2.5)$$

Si ha pertanto che, se siamo nel secondo caso, il quoziente e il resto della divisione intera di *x* per *C* sono rispettivamente pari a -(q_y +1) e a (C-r_y).

Alcuni esempi:

Sia *x*=17, *C*=13; si ha *q*=1, *r*=4.

Sia *x*=-11, *C*=4; si ha *q*=-3, *r*=1.

Sia *x*=-12, *C*=4; si ha *q*=-3, *r*=0.

Operazione "modulo *C*".

Il resto della divisione intera di un numero relativo *x* per una costante positiva *C* si indica con la scrittura:

$$r = x \bmod C \qquad (2.6)$$

e si legge "*r* uguale a *x* modulo *C*".

Elenchiamo alcune proprietà della funzione modulo;

1. *r*=*x* mod *C* è tale che si ha in ogni caso: $0 \le r < C$;
2. se $0 \le x < C$ allora *x* mod *C*=*x*;
3. se *x*=*kC* con *k* numero intero relativo, *x* mod *kC*=0;
4. *x* mod *C*=(*x* mod *C*) mod *C* (ovvia dalla proprietà 1);
5. (*x*+*y*) mod *C*= (*x* mod *C* + *y* mod *C*) mod *C*

Dimostrazione:

Sappiamo che si può scrivere:

$$\begin{array}{ll} x = q_x \times C + r_x & 0 \le r_x < C \\ y = q_y \times C + r_y & 0 \le r_y < C \end{array} \qquad (2.7)$$

Si può allora certamente scrivere:

$$x + y = \left(q_x + q_y\right) \times C + \left(r_x + r_y\right) \qquad 0 \le (r_x + r_y) < 2\mathrm{C} \qquad (2.8)$$

non si può però concludere che (r_x +r_y) rappresenta (*x*+*y*) *mod C* visto che, in generale, non rispetta la condizione (2). D'altra parte esistono solo due possibilià:

- (r_x +r_y) è minore di *C,* per cui questa quantità rappresenta effettivamente (*x*+*y*) mod *C* e per la proprietà 2 rappresenta anche dalla (*x* mod *C* +*y* mod *C*) mod C .
- si ha che $C \le (r_x + r_y) < 2\mathrm{C}$ e allora si può scrivere

$$\left(r_x + r_y\right) = 1 \times C + r_{xy} \qquad 0 \le r_{xy} < C \quad (2.9)$$

Dalla relazione precedente si evince che r_{xy} è proprio (x mod C +y mod C) mod C. D'altra parte, in base alla (2.9), la (2.8) può essere riscritta come:

$$x + y = \left(q_x + q_y + 1\right) \times C + r_{xy} \qquad\qquad 0 \le r_{xy} < C \ (2.10)$$

e pertanto r_{xy} è anche uguale a $(x+y)$ *mod C.*

6. la funzione x mod C è periodica di periodo C. Sia $x_2=x_1+kC$ con k intero relativo. Dalle proprietà 4 e 5 si ha:

$$\begin{aligned} x_2 modC &= (x_1 + kC)\text{mod } C = (x_1 \text{mod } C + kC \text{ mod } C)\text{mod } C = \\ &= (x_1 \text{ mod } C) \text{ mod } C = x_1 modC \end{aligned} \qquad (2.11)$$

Se si restringe opportunamente il dominio della funzione, la funzione modulo è invertibile. Nella rappresentazione in notazione a complemento si sfrutta questa possibilità. Fra le infinite scelte possibili, si può operare quella di restringere il dominio a un intervallo di valori che comprenda valori di x sia positivi sia negativi. Nel caso in cui la costante C è dispari, si può sceglie l'intervallo simmetrico (si tenga sempre presente che x può assumere solo valori interi):

$$-\frac{C}{2} < x < \frac{C}{2} \qquad \text{per } C \text{ dispari} \qquad (2.12)$$

Si noti che gli estremi non sono compresi in quanto $C/2$, se C è dispari, non è un numero intero.
Se invece C è pari, allora si può includere uno degli estremi nella (2.12) fra i valori di x per i quali la funzione modulo risulta invertibile. La scelta cade normalmente sull'intervallo sinistro giacché, come vedremo in seguito, questa scelta semplifica alcuni algoritmi. Si ha pertanto:

$$-\frac{C}{2} \le x < \frac{C}{2} \qquad C \text{ pari} \qquad (2.13)$$

Se ci limitiamo alle restrizioni (2.12) o (2.13), valgono le seguenti relazioni:

$$x_R = x \text{ mod } C = \begin{cases} x & 0 \le x < \frac{C}{2} \\ C + x & -\frac{C}{2} \le x < 0 \end{cases} \qquad (2.14)$$

Per l'inversione della funzione modulo, sempre nell'ambito delle restrizioni fissate, si ha:

$$x = \begin{cases} x_R & 0 \le x_R < \frac{C}{2} \\ x_R - C & \frac{C}{2} \le x_R < C \end{cases} \qquad (2.15)$$

La funzione modulo, con le proprietà fino a ora elencate è alla base della rappresentazione in complemento dei numeri interi relativi. In particolare, la funzione modulo viene usata, con una opportuna scelta della costante C e nei limiti delle restrizioni (2.12) e (2.13) per trasformare un intervallo di numeri relativi x in un insieme di numeri non negativi x_R.

Rappresentazione in complemento

Supponiamo che ci venga chiesto di eseguire la somma di due numeri interi x_1 e x_2 dati. A prima vista il significato della richiesta che ci viene formulata è assolutamente chiaro, ma se ci fermiamo un momento a riflettere sul senso delle parole usate, scopriamo che non è così. In base alla nostra esperienza, quando parliamo di due "numeri dati", immaginiamo che qualcuno scriva su un pezzo di carta le cifre che *rappresentano* i due numeri secondo la convenzione che ci appare più naturale perché ci è stata insegnata sin dalle scuole elementari e che è quella della rappresentazione nel sistema decimale. E cosa vuol dire "eseguire la somma"? Vuol dire operare mediante un algoritmo sulle *rappresentazioni* dei due numeri per costruire la *rappresentazione* del numero somma. E' evidente che ciò è possibile solo se è assolutamente chiaro quale sia la *rappresentazione* alla quale si fa riferimento e quale sia, allo stesso tempo, la regola di interpretazione che ci fa risalire dalla rappresentazione ottenuta al numero che essa rappresenta. L'espressione "fare dei calcoli su dei numeri" può essere interpretata solo come "*svolgere operazioni o algoritmi sulle rappresentazioni dei numeri per ottenere la rappresentazione di altri numeri che costituiscono il risultato di una operazione matematica fra i numeri di partenza*".

La richiesta che ci viene formulata, allora, ci appare chiara solo perché diamo per scontato che i numeri ci saranno forniti mediante la loro rappresentazione decimale e che ci viene richiesto di produrre la rappresentazione decimale della loro somma.

Non è detto che la rappresentazione più adatta allo svolgimento degli algoritmi di calcolo mediante penna e carta sia necessariamente la più adatta allo svolgimento di calcoli mediante ausili meccanici, grafici o elettronici. Tutti coloro che hanno studiato la teoria delle linee di trasmissione hanno familiarità con la carta di Smith e con gli algoritmi che consentono, per esempio, di calcolare l'impedenza vista lungo una linea di trasmissione mediante l'uso del compasso. Alcuni probabilmente hanno sentito parlare o hanno visto usare il regolo calcolatore, mediante il quale si eseguono operazioni di moltiplicazione (leggi "si eseguono gli algoritmi che servono per trovare la rappresentazione del prodotto di due numeri a partire dalla loro rappresentazione") mediante lo scorrimento meccanico relativo di due aste e lo sfruttamento delle proprietà della funzione logaritmo. Nel caso dei sistemi di calcolo elettronico, risulta particolarmente semplice implementare l'algoritmo di somma fra numeri interi naturali (particolarmente quando si usa una rappresentazione in binario naturale), mentre algoritmi apparentemente altrettanto semplici come quelli che si usano nei conti "a mano" per il calcolo della differenza o della somma di interi relativi in rappresentazione in modulo e segno (per esempio "+3"+"-7" in decimale o "+010" + "-101") risultano più complessi. Fortunatamente esiste la possibilità di eseguire una trasformazione da un insieme finito di numeri interi relativi a un insieme finito di numeri naturali in modo tale che l'operazione di somma fra i numeri relativi di partenza possa essere ricondotta a una operazione di somma aritmetica fra i numeri naturali trasformati.

Poniamo da un punto di vista più generale il problema che intendiamo affrontare. Siano x e y due numeri interi relativi. Supponiamo che esista una funzione di trasformazione invertibile $f_R(x)$ mediante la quale i numeri vengano trasformati in numeri interi naturali x_R e y_R, ovvero:

$$\begin{aligned} x_R &= f_R(x) \\ y_R &= f_R(y) \end{aligned} \tag{2.16}$$

Siano ora X_R e Y_R le rappresentazioni in un opportuno sistema di numerazione dei numeri naturali x_R e y_R. Supponiamo che sia stato dimostrato che una certa operazione fra i numeri di partenza x e y si traduca in una operazione equivalente sui numeri x_R e y_R nel senso che se z è il risultato della operazione fra x e y, $z_R = f_R(z)$ è il risultato della operazione equivalente fra x_R e y_R. Supponiamo che l'operazione da compiere sui numeri x_R e y_R si traduca in un algoritmo da applicare alle loro rappresentazioni X_R e Y_R che consente di ricavare la rappresentazione Z_R del numero z_R.

In questa situazione, svolgere l'operazione che da x e y produce z si può tradurre nelle seguenti fasi:

1. sfruttando la funzione di trasformazione $f_R(x)$ otteniamo i numeri x_R e y_R;
2. a partire da x_R e y_R otteniamo le loro rappresentazioni in termini di vettori di cifre X_R e Y_R;
3. si esegue l'algoritmo opportuno che consente di ricavare le cifre della rappresentazione di Z_R di z_R;
4. Si usa la regola di interpretazione del sistema di numerazione per risalire al numero z_R a partire dalla sua rappresentazione Z_R.
5. Sfruttando la funzione inversa di $f_R()$, ovvero $f^{-1}_R()$, si ricava il numero z che costituisce il risultato dell'operazione fra i numeri x e y di partenza: $z = f^{-1}_R(z_R)$.

Evidentemente non tutte le possibili funzioni di trasformazione e non tutti i sistemi di numerazione sono ugualmente adatti a rendere più semplice l'esecuzione di una determinata operazione. Esiste inoltre una stretta relazione fra la funzione di trasformazione e gli algoritmi di elaborazione delle rappresentazioni dei numeri trasformati.
Per quanto riguarda il problema della somma fra numeri interi relativi, la soluzione che appare la più conveniente, nel caso in cui si debba realizzare un sistema di calcolo per uso generale, è quello cha fa riferimento al sistema di numerazione binario naturale e alla funzione di trasformazione in complemento. Nel seguito discuteremo intanto della natura e delle proprietà della funzione di trasformazione in complemento sia con riferimento al sistema di numerazione decimale (che è quello probabilmente più familiare), sia con riferimento al sistema binario naturale.

Sistemi di rappresentazione decimale e binario di numeri interi naturali.
Sia il sistema di rappresentazione decimale, sia il sistema di rappresentazione binario appartengono alla famiglia di rappresentazioni posizionali, non ridondanti, pesati a radice fissa, di tipo canonico.

In questo tipo di sistemi si ha che, fissato il numero di cifre *n* della rappresentazione, un numero naturale (intero non negativo) *x* viene rappresentato dal vettore di cifre *X*:

$$X_R = (X_{n-1}, X_{n-2}, \dots, X_{1,} X_0) \qquad (2.17)$$

Ciascuna cifra X_i rappresenta un numero intero compreso fra 0 e *r*-1, dove *r* è la radice che vale “dieci” nel sistema decimale e “due” nel sistema binario. Si noti che abbiamo usato volutamente le parole “dieci” e “due” per indicare i valori della radice prescindendo da una specifica rappresentazione. Ciò non è sempre facile o conveniente dal punto di vista della chiarezza della discussione, per cui si raccomanda il lettore di prestare attenzione al contesto e di mantenere sempre chiara la distinzione fra numero e sua rappresentazione.
Nel sistema decimale i simboli per le cifre sono 0,1,2,.....,8 e 9 per indicare rispettivamente i numeri “zero”,”uno”,”due”,.....,”otto” e “nove”.
Nel sistema binario i simboli per le cifre sono 0 e 1 per indicare rispettivamente i numeri “zero” e “uno”.
In entrambi i sistemi di numerazione, la regola di interpretazione per risalire al numero rappresentato è la seguente:

$$x = \sum_{i=0}^{n-1} [X_i] r^{i-1} \qquad (2.18)$$

Si noti che, nell'espressione precedente, $[X_i]$ è il numero che corrisponde al simbolo X_i . Non ha senso fare operazioni matematiche su un simbolo! Si rifletta bene su questo aspetto prima di proseguire.

E' evidente che noi non maneggiamo mai numeri astratti, ma sempre e comunque loro rappresentazioni. Siamo così assuefatti alla rappresentazione decimale che tendiamo a confondere la rappresentazione del numero con il numero stesso. E in effetti è difficile trovare altro modo di riferirsi a uno specifico numero senza ricorrere a una sua rappresentazione. Si può ben parlare in astratto del doppio del doppio del numero intero che si ottiene arrotondando il numero π, ma possiamo anche dire che stiamo parlando del numero la cui rappresentazione in notazione decimale su due cifre è "12". Nel seguito, per semplificare, molto spesso diremo "il numero 12", intendiamo in realtà dire "il numero la cui rappresentazione in notazione decimale su due cifre è ("1","2"). Così una affermazione del tipo "il numero 12 ha 12 come rappresentazione decimale su due cifre" può avere senso. Avrà anche senso dire che la rappresentazione del numero 12 in binario naturale su 4 cifre è 1100.

Avendo chiaro quanto appena discusso, tutte le volte che vorremo riferirci a un numero particolare lo faremo mediante la sua rappresentazione decimale senza che ciò venga ripetuto in tutte le circostanze.

Se abbiamo a che fare con due numeri naturali rappresentati in notazione decimale, possiamo ottenere la notazione decimale del numero somma usando un algoritmo che a partire dalle cifre della rappresentazione dei due addendi (attenzione, non "dalle cifre dei suoi addendi"!) produce la rappresentazione del numero somma.

Nel seguito assumeremo noto l'algoritmo che consente di effettuare la somma fra due numeri rappresentati in notazione decimale.

Rappresentazione di numeri interi relativi in complemento

Per rappresentazione di numeri interi relativi in complemento intendiamo in realtà riferirci alla rappresentazione del numero naturale x_R ottenuto come trasformazione di un numero intero relativo x secondo la funzione (2.14). Tratteremo separatamente il caso della notazione decimale e della notazione in binario naturale.

Rappresentazione in complemento (*sistema decimale*)

Questo tipo di rappresentazione non è usata nella implementazione di sistemi di calcolo numerico. Viene qui discussa allo scopo di mettere in evidenza le proprietà della rappresentazione in complemento operando nel sistema di rappresentazione dei numeri naturali che ci è più familiare. Con riferimento al sistema decimale, esistono due scelte particolarmente interessanti della costante di complementazione C. Se n è il numero di cifre disponibili per la rappresentazione (in tutti gli esempi che seguono assumeremo n=3), queste due scelte corrispondono a scegliere $C=r^n$ e $C=r^n-1$. Nel primo caso di parla di rappresentazione in "complemento alla radice" mentre nel secondo caso di parla di "complemento alla cifra".

Anche se sarebbe interessante e istruttivo occuparsi di entrambe le rappresentazioni, per semplicità di limiteremo al caso del complemento alla radice perché più direttamente legata al complemento a due, che sarà la notazione che useremo più di frequente in questo corso.

Con riferimento al caso n=3, operare nell'ambito del complemento alla radice corrisponde a scegliere C=1000. Come si vede, C è pari, portando così a un intervallo asimmetrico per la rappresentazione dei numeri positivi e negativi.

I numeri relativi rappresentabili vanno da $-r^n/2$ fino a $r^n/2$ -1. Con n=3, per esempio, sono rappresentabili tutti i numeri interi relativi da -500 a 499, estremi compresi. Le considerazioni che seguono verranno riferite al caso n=3, ma sono immediatamente estendibili al caso di n qualunque.

I numeri x da 0 a 499 vengono trasformati nei numeri x_r da 0 a 499 e sono quindi caratterizzati dal fatto che la loro rappresentazione comincia in ogni caso con una cifra da 0 a 4. I numeri da -500 a -1 vengono trasformati nei numeri x_r da 500 a 999, estremi compresi. I numeri negativi sono quindi caratterizzati dal fatto che la loro rappresentazione comincia in ogni caso con una cifra uguale o superiore a 5. La capacità di capire immediatamente della prima cifra della rappresentazione se questa si riferisce a un numero non negativo (cifra iniziale da 0 a 4) o a un numero negativo (cifra iniziale da 5 a 9) è molto utile, come risulterà chiaro più avanti. Si noti che questa possibilità è il risultato dell'aver scelto di includere il numero -500 e non il numero 500 nell'intervallo di rappresentazione.

Data la rappresentazione di due numeri interi x e y, la rappresentazione della loro somma, purché essa rientri nell'intervallo dei valori rappresentabili, può essere ottenuta operando sulle rappresentazioni dei numeri x_r e y_r.

Per capire come ciò sia possibile, osserviamo che l'operazione che dovrebbe essere svolta in lina di principio per ottenere la rappresentazione z_r del numero z=(x+y), purché z rientri nell'intervallo di rappresentazione, sarebbe:

$$z_r = (x + y) \bmod C \quad (2.19)$$

D'altra parte, sappiamo anche che:

$$x_r = x \bmod C \qquad y_r = y \bmod C \quad (2.20)$$

Per le proprietà della funzione modulo studiate in precedenza, l'operazione (2.19) è equivalente alla seguente:

$$z_r = (x \bmod C + y \bmod C) \bmod C = (x_r + y_r) \bmod \mathrm{C} \quad (2.21)$$

Dalla (2.21) è quindi evidente che si può ottenere z_r direttamente da x_r e y_r, senza bisogno di passare da x e y. Si ricordi tuttavia, che, perché si produca un risultato univocamente interpretabile, è necessario sapere (per ora è una conoscenza che poniamo "a priori") che il risultato (z) sia rappresentabile. Il vantaggio di operare con x_r e y_r è che noi sappiamo ottenere la rappresentazione della loro somma applicando l'algoritmo di somma (delle scuole elementari) alle loro rappresentazioni. Si noti infatti che, pensando al consueto algoritmo di somma fra numeri naturali, con C=1000 l'operazione modulo C nella (2.21) corrisponde semplicemente a ignorare l'eventuale riporto che darebbe origine a una rappresentazione su 4 cifre. Facciamo alcuni esempi:

Esempio 1:

$x = 3, x_r = 3, X_r = (\text{"0"}, \text{"0"}, \text{"3"}); y = 117, y_r = 117, Y_r = (\text{"1"}, \text{"1"}, \text{"7"})$

la somma z=x+y=120 è rappresentabile (rientra nell'intervallo per il quale la funzione modulo, con le nostre convenzioni, è invertibile). Il risultato dell'algoritmo di somma sulle rappresentazioni è la successione di cifre ("1", "2", "0"). L'operazione modulo 1000 consiste semplicemente nell'assumere Z_r=("1", "2", "0"). Si ha quindi

$Z_r = (1\text{"}, \text{"}2\text{"}, \text{"}0\text{"}) \rightarrow z_r = 120 \rightarrow z = 120$

Esempio 2:

$x = -12, x_r = 988, X_r = (\text{"9"}, \text{"8"}, \text{"8"}); y = 117, y_r = 117, Y_r = (\text{"1"}, \text{"1"}, \text{"7"})$

la somma z=x+y=105 è rappresentabile. Il risultato dell'algoritmo di somma sulle rappresentazioni, se la facessimo a mano su un foglio di carta, sarebbe la successione di cifre "1","1", "0", "5". L'operazione modulo 1000 consiste semplicemente nell'ignorare la prima della 4 cifre ottenendo Z_r=("1", "0", "5"):

$Z_r = (\text{"1"}, \text{"0"}, \text{"5"}) \rightarrow z_r = 105 \rightarrow z = 105$

Esempio 3:

$x = -12, x_r = 988, X_r = ("9","8","8"); y = -117, y_r = 883, Y_r = ("8","8","3")$

la somma $z=x+y=-129$ è rappresentabile. Il risultato dell'algoritmo di somma sulle rappresentazioni è la successione di cifre "1", "8", "7", "1". L'operazione modulo 1000 consiste nell'assumere Z_r=("8", "7", "1"), ignorando la prima cifra della successione). Si ha quindi

$Z_r = ("8","7","1") \rightarrow z_r = 871 \rightarrow z = -129$

Esempio 4:

$x = -333, x_r = 667, X_r = ("6","6","7"); y = -200, y_r = 800, Y_r = ("8","0","0")$

Questa volta la somma $z=x+y=-533$ *non è rappresentabile*. Eseguiamo comunque l'algoritmo di somma sulle rappresentazioni. Il risultato è la successione di cifre "1", "4", "6", "7". L'operazione modulo 1000 consiste nell'assumere Z_r=("4", "6", "7") Si ha quindi

$Z_r = ("4","6","7") \rightarrow z_r = 467 \rightarrow z = 467$

come si vede, in questo caso l'applicazione dell'algoritmo porta a un risultato errato.

Esempio 5:

$x = 333, x_r = 333, X_r = ("3","3","3"); y = 200, y_r = 200, Y_r = ("2","0","0")$

Anche questa volta la somma $z=x+y=533$ non è rappresentabile. Eseguiamo comunque l'algoritmo di somma sulle rappresentazioni. Il risultato è la successione di cifre "5", "3", "3". L'operazione modulo 1000 consiste nell'assumere Z_r=("5", "3", "3") . Si ha quindi

$Z_r = ("5","3","3") \rightarrow z_r = 533 \rightarrow z = -467$

come si vede, anche in questo caso l'applicazione dell'algoritmo porta a un risultato errato.

Perché l'algoritmo di somma descritto possa essere impiegato in un sistema di calcolo automatico, è quanto meno necessario avere la capacità di individuare il fatto che l'algoritmo ha portato a un risultato errato.

Fortunatamente, con la particolare rappresentazione scelta, l'individuazione di un risultato errato (situazione alla quale ci si riferisce con il termine "overflow") a partire dalle rappresentazioni degli addendi e del risultato dell'algoritmo è piuttosto semplice.

Per arrivare alla regola di riconoscimento di una situazione di overflow partiamo dalla considerazione che se i numeri x e y sono rappresentabili su n cifre, la loro somma è certamente rappresentabile su n+1 cifre. Questo è vero sia nel caso r=10, sia nel caso r=2. Supponiamo ora di voler ottenere la rappresentazione di x con riferimento a $C'=r^{n+1}$ a partire dalla rappresentazione con riferimento a $C=r^n$. Se x è un numero positivo o nullo, x_r (trasformazione rispetto a C) e x'_r (trasformazione rispetto a C') coincidono. Pertanto la rappresentazione X'_r si ottiene aggiungendo semplicemente la cifra 0 all'inizio della stringa di bit X_r. Nel caso in cui x è negativo, x_r e x'_r non coincidono. Si ha:

$$x'_r = C' + x = (r-1) \times r^n + C + x = (r-1) \times r^n + x_r \quad (2.22)$$

Con riferimento alla base 10 si ha che la rappresentazione X'_r si ottiene aggiungendo la cifra 9 alla sinistra delle cifre che costituiscono la rappresentazione X_r.

Poiché, come abbiamo già osservato, si può desumere dalla prima cifra della rappresentazione se questa è relativa a un numero negativo o non negativo, si può desumere una regola molto semplice per estendere la rappresentazione di un numero da n a n+1 cifre. La regola è la seguente: se la prima cifra di una rappresentazione è inferiore a 5, allora l'estensione a n+1 cifre si ottiene aggiungendo la cifra zero alla sinistra di tutte le altre. Se la prima cifra della rappresentazione è superiore o uguale a 5, l'estensione della rappresentazione si ottiene aggiungendo la cifra 9 alla sinistra di tutte le altre. Facciamo alcuni esempi relativi all'estensione della rappresentazione di numeri rappresentati su 3 cifre a 4 cifre:

$123 \rightarrow 0123;\ 002 \rightarrow 0002; 512 \rightarrow 9512;\ 712 \rightarrow 9712; 997 \rightarrow 9997$

Si osservi ora che un numero rappresentato su n+1 cifre potrebbe essere rappresentabile anche su sole n cifre. Sulla base delle considerazioni precedenti si ha che la rappresentazione di un numero su n+1 cifre può essere ridotta a n cifre se la cifra più a sinistra è 0 e quella immediatamente a destra è minore di 5 o se la cifra più a sinistra è 9 e la cifra immediatamente a destra è maggiore o uguale a 5. La riduzione si ottiene, nei casi in cui è possibile, semplicemente eliminando la cifra più a sinistra. Alcuni esempi:

$0231 \rightarrow 231$; 0733 non è riducibile; $9521 \rightarrow 521$; 9234 non è riducibile; $9993 \rightarrow 993$;

A questo punto siamo pronti a determinare la regola per individuare una situazione di overflow. In linea di principio i passaggi da fare sono:

I) si estende la rappresentazione di x e y su n+1 cifre.
II) Si esegue l'algoritmo di somma su n+1 cifre. Il risultato considerando le n+1 cifre più a sinistra è certamente corretto.
III) Se è possibile ridurre la rappresentazione del risultato da n+1 cifre a n cifre non c'è overflow e il risultato può essere correttamente rappresentato su n cifre; se il risultato non è riducibile a n cifre significa evidentemente che siamo in una condizione di overflow.

Fortunatamente non è necessario eseguire effettivamente i calcoli su n+1 cifre. Valgono infatti le seguenti osservazioni:

1. Se x e y sono di segno opposto, il risultato è certamente rappresentabile su n cifre.
2. Se x e y sono non negativi, è immediato verificare che l'estensione su n+1 cifre della rappresentazione (uno zero aggiunto alla sinistra per tutte due le rappresentazioni) porta a un risultato su n+1 in cui la cifra più a sinistra è certamente 0. Il risultato è riducibile su n cifre solo se l'ennesima cifra a partire da destra è inferiore a 5.
3. Se x e y sono negativi, è immediato verificare che l'estensione su n+1 cifre della rappresentazione (un 9 aggiunto alla sinistra per tutte due le rappresentazioni) porta a un risultato su n+1 cifre in cui la cifra più a sinistra è certamente 9 (si ricordi che l'algoritmo prevede che in ogni caso sole n+1 cifre più a sinistra devono essere prese in considerazione) pertanto il risultato è riducibile a n cifre solo se l'ennesima cifra a partire da sinistra è maggiore o uguale a 5.

Pertanto, senza svolgere effettivamente la somma su n+1 cifre si può formulare la seguente regola per stabilire la presenza di overflow:

Si ha overflow se e solo se:

- le cifre più a sinistra delle rappresentazioni dei due addendi su n cifre sono entrambe inferiori a 5 e la cifra più a sinistra dell'algoritmo di somma (modulo r^n) è maggiore o uguale a 5;
- le cifre più a sinistra delle rappresentazioni dei due addendi su n cifre sono entrambe maggiori o uguali a 5 e la cifra più a sinistra dell'algoritmo di somma (modulo r^n) è minore di 5;

Un altro modo di formulare la regola precedente è la seguente: si ha overflow se e solo se nella somma di due rappresentazioni su n cifre che si riferiscono a numeri dello stesso segno si ottiene una rappresentazione (su n cifre) che si riferisce numero di segno opposto rispetto a quello degli addendi. Ai soli fini dell'applicazione di questa regola si può considerare lo 0 come appartenente alla categoria di "numeri positivi".

Si verifichi questo criterio sui risultati degli esempi di somma riportati in precedenza.

Esercizio per lo studente: si ottenga l'algoritmo necessario per ottenere la rappresentazione dell'opposto di un numero, data la sua rappresentazione in complemento alla radice (r=10) su n cifre. L'operazione è sempre possibile? Se non è sempre possibile, come si fa ad individuare il fatto che l'operazione ha prodotto un risultato sbagliato?

Rappresentazione in complemento a 2

Nel seguito assumeremo che i numeri naturali x_R siano sempre rappresentati in "binario naturale su *n* cifre". Più esattamente, assumeremo in ogni caso che, fissato un numero di cifre *n*, rappresenteremo ogni intero in un sistema canonico binario.

Ricordiamo che:

- in un sistema canonico binario, ciascuna delle cifre $X_{n-1},\ldots.X_0$ che compongono la rappresentazione può assumere solo il simbolo "1" o il simbolo "0", con ["1"]=1 e ["0"]=0. Pertanto, sono possibili un massimo di 2^n rappresentazioni distinte.
- La regola di interpretazione per risalire al numero x_R rappresentato è la seguente:

$$x = \sum_{i=0}^{n-1} [X_i] 2^{i-1} \qquad (2.23)$$

Le operazioni aritmetiche su numeri naturali rappresentati in binario naturale si svolgono operando sulle cifre della rappresentazione secondo gli algoritmi di somma, differenza, prodotto e quoziente appresi alle scuole elementari. Supporremo acquisita la conoscenza di questi algoritmi.

Come in precedenza, il nostro problema sarà quello di determinare, se esistono, i metodi che consentono di ottenere una rappresentazione del risultato di una operazione fra numeri interi relativi mediante operazioni svolte esclusivamente su numeri interi naturali ottenuti dalla trasformazione in complemento.

Osservazione:

- Gli algoritmi necessari per ottenere il risultato di operazioni aritmetiche operando sulle cifre della rappresentazione binaria naturale dei numeri stessi sono particolarmente semplici e facilmente implementabili mediante circuiti logici. E' per questa ragione che si cerca di ricondurre l'esecuzione di operazioni aritmetiche su numeri relativi a operazioni elementari sui numeri naturali ottenuti mediante la trasformazione in complemento.
- Anche quando dobbiamo eseguire operazioni su numeri relativi con "carta e penna" operiamo in modo analogo. Infatti ci riconduciamo generalmente a operazioni sui moduli dei numeri e stabiliamo poi il segno del risultato. La notazione che impieghiamo nei nostri calcoli "a mano" è quella in modulo e segno. Tuttavia le regole di calcolo in modulo e segno sono relativamente complesse da implementare direttamente in termini di circuiti elettronici. E' per questa ragione che si ricorre alla rappresentazione in complemento che consente, al prezzo di una complicazione interpretativa, di implementare più semplicemente gli algoritmi di operazioni aritmetiche su numeri relativi che sono alla base di ogni altro calcolo numerico.

Nel caso di un sistema di rappresentazione in complemento alla radice con *r*=2 si parla più semplicemente di rappresentazione in complemento a 2. Se la rappresentazione usa *n* cifre, la costante di complementazione è $C=2^n$.

La scelta dell'intervallo di valori di *x* al quale restringere la rappresentazione è la seguente:

$$-\frac{C}{2} \leq x \leq \frac{C}{2} - 1 \quad ovvero \quad -2^{n-1} \leq x \leq 2^{n-1} - 1 \quad (2.24)$$

Con queste restrizioni per x si ottiene:

$$x_R = \begin{cases} x & se \quad x \geq 0 \\ C + x & se \quad x < 0 \end{cases} \quad (2.25)$$

Si noti che per gli x_r corrispondenti a numeri x positivi o nulli si ha:

$$0 \leq x_R \leq 2^{n-1} - 1 \quad (2.26)$$

Per tutti gli x_R corrispondenti a numeri x negativi si ha:

$$2^{n-1} \leq x_R \leq 2^n - 1 \quad (2.27)$$

Pertanto la rappresentazione in complemento a 2 gode della seguente notevole proprietà:

Se la cifra (bit) più significativa della rappresentazione di x_R in binario naturale vale 1, allora quel particolare x_R si riferisce a un numero x negativo; se invece la cifra più significativa della rappresentazione è 0, allora quel particolare valore di x_R si riferisce a un numero positivo o nullo.

Sia X la rappresentazione in binario naturale su n cifre di x_R. Si ha:

$$X : \{X_{n-1}; X_{n-2};...; X_1; X_0\} \quad (2.28)$$

Se se X_{n-1}=0 allora si ha che X è la rappresentazione di un numero x_R che corrisponde a un numero x positivo o nullo (un po' impropriamente si potrebbe dire che X è la rappresentazione in complemento a due di un numero x positivo o nullo) e si ha:

$$x = x_R = 0 \cdot 2^{n-1} + \sum_{i=0}^{n-2} [X_i] 2^i = \sum_{i=0}^{n-2} [X_i] 2^i \quad (2.29)$$

Se se X_{n-1} è "1" allora si ha che X è la rappresentazione di un numero x_R che corrisponde a un numero x negativo (un po' impropriamente si potrebbe dire che X è la rappresentazione in complemento a due di un numero x negativo) e si ha:

$$x = x_R - C = 1 \cdot 2^{n-1} + \sum_{i=0}^{n-2} [X_i] 2^i - 2^n = -2^{n-1} + \sum_{i=0}^{n-2} [X_i] 2^i \quad (2.30)$$

Facciamo alcuni esempi.

Sia n=8. Allora si ha:

$$C = 256; \; C/2 = 128; \; -128 \leq x \leq 127; \; 0 \leq x_R \leq 255$$

Trasformazione in complemento a due di alcuni numeri e rappresentazione della trasformazione in binario naturale (nella rappresentazione, per semplicità. non sono state usate le virgolette per evidenziare che si tratta di simboli e non di numeri: si faccia attenzione a tenerlo presente).

x=17	x_R=17	X:{0001 0001}
x=-13	x_R=C+x=256-13=243	X:{1111 0011}

x=-128	x_R=C+x=256-128=128	X:{1000 0000}
x=128	*Non trasformabile*	==========
x=-1	x_R=C+x=256-1=255	X:{1111 1111}

Per quanto riguarda l'interpretazione della rappresentazione di un numero x_R dato:

X	x_R	*Interpretazione in base alla rappresentazione*	*Interpretazione in base al valore di* x_R
1010 1010	170	X_{n-1}= "1"=> x=-128+32+8+2=-86	$x_R \geq C/2$=128=> x=-256+170=-86
0111 0000	112	X_{n-1}= "0"=> x=64+32+16=112	$x_R < C/2$=128=> x=112

Somma di numeri relativi in rappresentazione in complemento

Come in precedenza, ci occuperemo prima del problema di ottenere la rappresentazione della somma assumendo che la somma stessa sia rappresentabile nel sistema a complemento usato. Più avanti discuteremo di come sia possibile rilevare la condizione di somma non rappresentabile (condizione di overflow).

Ricordiamo che le rappresentazioni in complemento sono utilizzate in quanto la rappresentazione della somma di due numeri x e y in complemento si può ottenere a partire dalla somma delle rappresentazioni dei numeri naturali x_R e y_R ottenuti da x e y mediante le trasformazioni prima discusse.

Supporremo quindi di avere a disposizione un sistema elettronico capace di elaborare la rappresentazione in binario naturale della somma di due numeri naturali rappresentati in binario naturale. In generale, la rappresentazione corretta della somma di due numeri naturali rappresentati in binario naturale su n cifre, richiede n+1 cifre. Ricordiamo però che l'algoritmo per la somma di due numeri in complemento a 2 richiede che si esegua l'operazione "modulo 2^n", ovvero che si considerino solo n cifre.

Il blocco funzionale che chiameremo addizionatore in binario naturale e che, come è noto, è possibile realizzare agevolmente in termini di porte logiche elementari, è rappresentato in Fig.1. Con riferimento allo schema di addizionatore in binario naturale riportato in Fig.1, siano:

- X la rappresentazione in binario naturale del numero naturale x_R

- Y la rappresentazione in binario naturale del numero naturale y_R
- c_{in} una cifra che può assumere valore 1 o 0.

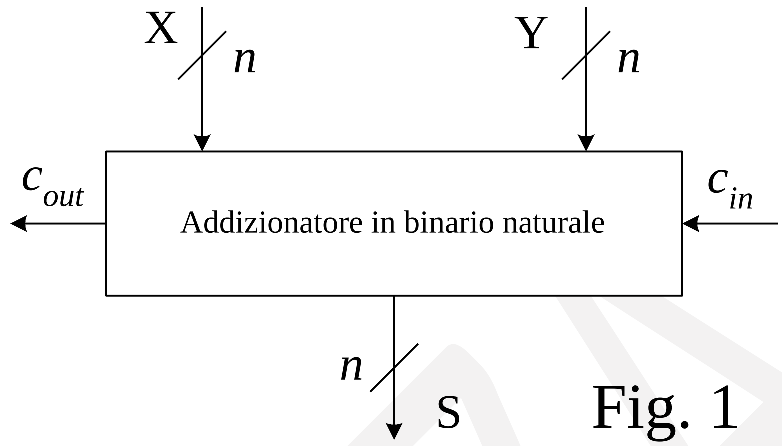

Fig. 1

Assumiamo allora come assunto di partenza per quanto segue che il blocco "Addizionatore" opera in modo che la stringa di bit:

$$\{c_{out}, S\} = \{c_{out}; S_{n-1}; S_{n-2}; \ldots; S_1; S_0\}$$

è la corretta rappresentazione in binario naturale della quantità *somma* data da:

$$somma = x_R + y_R + c_{in}$$

Si noti che la corretta rappresentazione della somma richiede in generale n+1 cifre. Solo nel caso in cui risulti c_{out}=0 si ha che la sola stringa di bit S rappresenta correttamente il risultato della somma.

Ora ci proponiamo di capire se e come, utilizzando un addizionatore in binario naturale, si può ottenere la corretta rappresentazione in complemento di due addendi rappresentati a loro volta in complemento. Prima di procedere è opportuno discutere il problema della riduzione/estensione del numero di cifre necessarie a rappresentare un numero in complemento.

Estensione-riduzione del numero di cifre nella rappresentazione in complemento

Abbiamo già discusso questo problema in termini generali. Qui ripetiamo quanto già fatto con specifico riferimento al complemento a 2.

Nella rappresentazione dei numeri naturali in decimale, è ovvio che le rappresentazioni 17, 017, 00017, 0000017 corrispondono allo stesso numero. Quando eseguiamo calcoli a mano abbiamo l'abitudine di non scrivere gli zeri "inutili" a sinistra dell'ultima cifra significativa. In un sistema di calcolo automatico, tutte le n cifre che rappresentano il numero devono essere espressamente indicate.

Supponiamo di avere numeri in binario naturale rappresentati su n cifre e di volerne ottenere la rappresentazione su n+1 cifre. Se $X^{(n)}$ è la rappresentazione di un numero naturale x_R su n cifre, si ricava immediatamente che la sua rappresentazione $X^{(n+1)}$ cifre si ottiene nel modo seguente:

$$X^{(n)} = \{X_{n-1}; X_{n-2};...; X_1; X_0\}$$
$$X^{(n+1)} = \{0, X^{(n)}\} = \{0; X_{n-1}; X_{n-2};...; X_1; X_0\}$$

D’altra parte non è evidentemente sempre possibile passare dalla rappresentazione a n+1 cifre di un numero x_R alla sua rappresentazione su n cifre. Questo è infatti possibile solo nel caso in cui $x_R<2^n$, ovvero se la prima cifra X_n della rappresentazione su n+1 cifre è zero. Se si verifica questa situazione, la rappresentazione su n cifre si ottiene semplicemente ignorando la presenza della cifra più a sinistra.

Supponiamo ora che $X^{(n)}$ sia la rappresentazione su n cifre del numero naturale x_R ottenuto come trasformazione in complemento del numero relativo x. E’ possibile ottenere la rappresentazione di x in complemento su n+1 cifre a partire dalla rappresentazione in complemento su n cifre?

Si noti che nella rappresentazione in complemento a 2 e in complemento a 1, modificare il numero di cifre della rappresentazione significa modificare anche la costante di complemento.

Osserviamo inoltre che la modifica della costante di complemento influisce sul valore dell’intero x_R nel caso in cui questo sia la trasformazione di un numero x negativo, mentre non si ha alcuna influenza su x_R quando questo è la trasformazione di un numero positivo.

Si ha pertanto che, nel caso in cui la rappresentazione $X^{(n)}$ di x_R si riferisca a un numero positivo (cosa cha avviane se X_{n-1}=0), la rappresentazione in complemento del numero x su n+1 cifre si ottiene semplicemente aggiungendo una cifra pari a zero a sinistra delle n cifre già presenti.

Se chiamiamo ora $x_R^{(n)}$ il numero intero che rappresenta un numero negativo x dopo una operazione di complemento con la costante $C^{(n)}$, si ha che $x_R^{(n+1)}$ è necessariamente diverso da $x_R^{(n)}$ visto che la costante di complementazione su n+1 cifre $C^{(n+1)}$ è divesa da $C^{(n)}$.

Si ha infatti che, se x<0, si ha allora, per il caso di complemento a 2:

$$x_R^{(n)} = 2^n + x \ ; \ x_R^{(n+1)} = 2^{n+1} + x \Rightarrow x_R^{(n+1)} = 2^n + x_R^{(n)}$$

Si conclude quindi che per estendere la rappresentazione di un numero negativo da n a n+1 cifre è sufficiente aggiungere alla sinistra della rappresentazione di $x_R^{(n)}$ la cifra “1”.

Se ora osserviamo che per estendere di una cifra la rappresentazione di un numero positivo (la cui rappresentazione X comincia con la cifra “0”) occorre aggiungere uno “0” a sinistra e che per estendere di una cifra la rappresentazione di un numero negativo (la cui rappresentazione X comincia con la cifra “1”) occorre aggiungere un “1” a sinistra, possiamo concludere che tutte le volte che si deve estendere la rappresentazione di un numero da n a n+1 è sufficiente aggiungere a sinistra la cifra X_n con X_n=X_{n-1}.

Dalle osservazioni precedenti si conclude anche che è possibile ridurre la rappresentazione di un numero da n+1 a n cifre eliminando la cifra più a sinistra della rappresentazione solo se risulta X_{n+1}=X_n.

Osservazione:

Si ha che in generale la rappresentazione della somma di due numeri rappresentabili in complemento su n cifre non è rappresentabile su n cifre. Si può invece dimostrare facilmente che la somma di due numeri rappresentabili in complemento su n cifre è certamente rappresentabile su n+1 cifre.

Dalla osservazione precedente possiamo dedurre la seguente regola per determinare se la somma di due numeri rappresentabili su *n* cifre è rappresentabile su *n* cifre. Infatti, la somma di due numeri rappresentabili su *n* cifre in complemento è rappresentabile su *n* cifre in complemento se la sua rappresentazione su *n*+1 cifre (che certamente esiste) è riducibile (ovviamente senza errori) alla rappresentazione su *n* cifre.

Questa osservazione ci sarà utile nel seguito.

Nel caso di rappresentazione in complemento a 2, la costante C vale 2^n e pertanto, l'operazione necessaria per ottenere la quantità s_R è esattamente quella svolta dal blocco addizionatore in binario naturale quando di consideri come risultato la sola quantità s.

Tuttavia, come abbiamo visto, il risultato è corretto solo se sappiamo a priori che la quantità $x+y$ è rappresentabile. Certamente, se la quantità somma non è rappresentabile su *n* cifre vogliamo quanto meno avere una segnalazione di risultato scorretto.

A tal fine supponiamo di agire come segue: a partire da un addizionatore aritmetico su *n* cifre ne otteniamo uno su *n*+1 cifre come indicato in Fig. 2 con l'aggiunta di un blocco full adder.

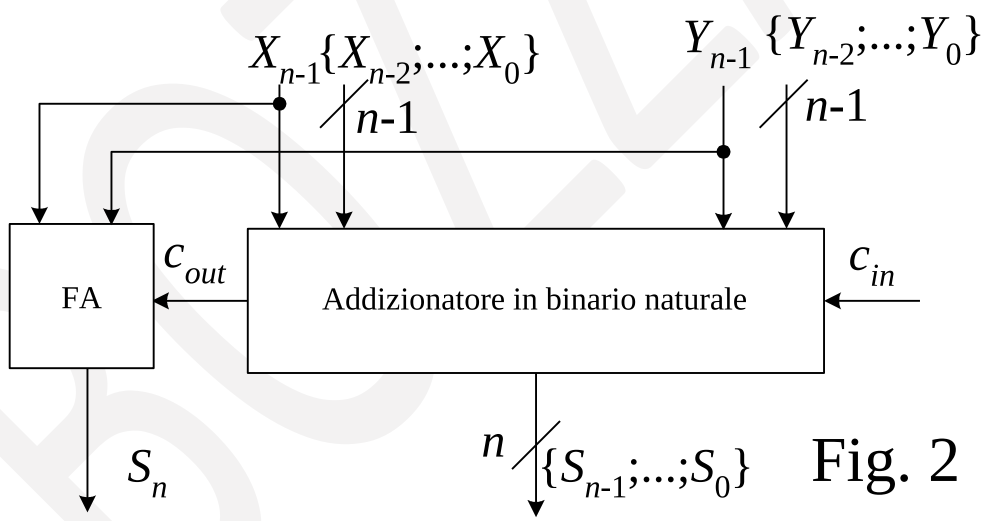

Fig. 2

Poiché la somma di due numeri rappresentabili su *n* cifre in complemento a 2 è certamente rappresentabile su *n*+1 cifre, siamo certi che se estendiamo la rappresentazione di x_R e y_R a *n*+1 cifre ed eseguiamo l'addizione con l'addizionatore a *n*+1 cifre, la somma (su *n*+1 bit) è in ogni caso la rappresentazione corretta della somma. A questo punto, se la somma è riducibile a *n* bit

(ovvero se S_n=S_{n-1}), allora possiamo ignorare la cifra S_n e ottenere la somma corretta su *n* cifre, altrimenti, se questa situazione non si verifica, sappiamo che la somma non è corretta e riconosciamo la presenza dell'errore.

Vogliamo ora dimostrare che la possibilità di riduzione della rappresentazione si può dedurre dal valore del bit c_{out} e dalla cifra S_{n-1} senza che sia necessario estendere la rappresentazione di x_R e y_R e aggiungere il blocco full adder per ottenere la somma corretta o la segnalazione di “non rappresentabilità su *n* cifre” comunemente detta condizione di overflow (OV).

Nella tabella seguente sono riassunti tutti i possibili casi con riferimento alla situazione ipotetica di impiegare un addizionatore a *n*+1 cifre.

Caso					
1	X_n=X_{n-1}= “1”	Y_n=Y_{n-1}= “1”	c_{out}= “1”	S_n= “1”	
2	X_n=X_{n-1}= “0”	Y_n=Y_{n-1}= “0”	c_{out}= “0”	S_n= “0”	
3	X_n=X_{n-1}= “1”	Y_n=Y_{n-1}= “0”	c_{out}	S_n=(c_{out})’	S_{n-1}=(c_{out})’
4	X_n=X_{n-1}= “0”	Y_n=Y_{n-1}= “1”	c_{out}	S_n=(c_{out})’	S_{n-1}=(c_{out})’

Esaminiamo preliminarmente i casi 3 e 4, che sono equivalenti. Questi casi corrispondono alla somma di un numero negativo con un numero non negativo. Il valore di c_{out} dipende dal valore del riporto in ingresso all’ultimo full adder nell’addizionatore a *n* cifre. Tuttavia, si nota che qualunque sia il valore di c_{out} si ha che S_n=S_{n-1} e pertanto la somma è sempre rappresentabile su *n* cifre.

Solo i casi 1 e 2 possono condurre alla condizione di overflow. Si ha in particolare:

$$OV = X_{n-1} Y_{n-1} \overline{S_{n-1}} + \overline{X_{n-1}}\;\overline{Y_{n-1}}\; S_{n-1}$$

Abbiamo quindi dimostrato come sia possibile ottenere la rappresentazione della somma di numeri in complemento a due mediante la somma delle rappresentazioni degli addendi in complemento a 2.