Capitolo 1

Teoria degli errori

1.1 Rappresentazione dei numeri

Scelto un qualunque numero intero $\beta > 1$, ogni numero non nullo $x \in \mathbb{R}$ ammette una *rappresentazione in base* β

$$x = sign(x)\, \beta^b \sum_{i=1}^{\infty} \alpha_i \beta^{-i} \,,$$

dove $b \in \mathbb{Z}$ e ogni α_i è un intero tale che $0 \le \alpha_i \le \beta - 1$, $i = 1, 2, \ldots$, ($sign(x)$ è definita come una funzione che assume valore 1 se $x > 0$ e valore -1 se $x < 0$). Il numero β dicesi appunto la *base*, b l'*esponente* e i numeri α_i si dicono le *cifre* della rappresentazione.

Vale il seguente teorema.

Teorema 1.1.1 (di rappresentazione) *Data una base intera $\beta > 1$ e un qualunque numero reale x diverso da zero, esiste un'unica rappresentazione in base β tale che:*

1. *sia $\alpha_1 \neq 0$,*

2. *non vi sia un intero k per cui si abbia $\alpha_j = \beta - 1$, $\forall j > k$.*

La rappresentazione definita dal teorema precedente dicesi *rappresentazione in virgola mobile normalizzata* del numero reale x.

Fissato β, sono quindi univocamente determinati i numeri b e α_i, $i = 1, 2, \ldots$, della rappresentazione normalizzata e la serie $\sum_{i=1}^{\infty} \alpha_i \beta^{-i}$ è detta *mantissa* del numero x; è immediato verificare che

$$\frac{1}{\beta} \leq \sum_{i=1}^{\infty} \alpha_i \beta^{-i} < 1.$$

All'interno di un calcolatore si possono rappresentare solo un certo numero m di cifre della mantissa di x. La rappresentazione sul calcolatore corrisponde perciò al numero $y = tr(x)$ oppure a $y = rd(x)$ dove $tr(x)$ ed $rd(x)$ sono approssimazioni di x, definite dai seguenti due criteri

$$y = tr(x) = sign(x)\, \beta^b \sum_{i=1}^{m} \alpha_i \beta^{-i}, \tag{1.1}$$

$$y = rd(x) = \begin{cases} tr(x) & \text{se } 0 \leq \alpha_{m+1} < \frac{\beta}{2} \\ sign(x)\, \beta^b \left[\sum_{i=1}^{m} \alpha_i \beta^{-i} + \beta^{-m}\right] & \text{se } \frac{\beta}{2} \leq \alpha_{m+1} < \beta \end{cases} . \tag{1.2}$$

Nel caso particolare in cui risulti $\alpha_1 = \alpha_2 = \cdots = \alpha_m = \beta - 1$ e $\alpha_{m+1} \geq \beta/2$, nella seconda delle (1.2) la normalizzazione porta ad aumentare di una unità l'esponente b. Per esempio, con $\beta = 10$ e $m = 3$, se $x = 0.9997 \times 10^5$ si ha $rd(x) = 0.100 \times 10^6$.

La (1.1) si chiama rappresentazione per *troncamento* del numero reale x mentre la (1.2) fornisce la rappresentazione per *arrotondamento*. Si può dimostrare che $\mid tr(x) - x \mid < \beta^{b-m}$ e $\mid rd(x) - x \mid \leq \frac{1}{2}\beta^{b-m}$, per cui, tra le due, la rappresentazione per arrotondamento è, in generale, una migliore approssimazione del numero reale x.

Si indichi con M l'insieme dei numeri z rappresentabili all'interno di un calcolatore, comunemente chiamati *numeri di macchina*.

M è un insieme finito; infatti, fissati β ed m e supposto $L \leq b \leq U$ $(L, U \in \mathbb{Z})$, la cardinalità di M risulta $2(\beta^m - \beta^{m-1})(U - L + 1) + 1$. Spesso l'insieme M viene indicato con il simbolo $F(\beta, m, L, U)$ per meglio evidenziare le caratteristiche della macchina.

Dato un qualunque numero reale $x \neq 0$, non è assicurata l'esistenza di $rd(x)$ fra i numeri di macchina.

Sia, per esempio, $F(10, 3, -99, 99)$ e $x = 0.9998 \times 10^{99}$; si ha $rd(x) = 0.1 \times 10^{100}$ che non rientra nell'insieme dei numeri di macchina considerato. In questo caso si ha una situazione di *overflow* e cioè il numero da rappresentare è "troppo grande" e non appartiene a M (tutti i calcolatori segnalano il presentarsi di un overflow, alcuni arrestano l'esecuzione del programma, altri proseguono con $rd(x) = sign(x) \max_{y \in M} \mid y \mid$).

Per contro, si abbia il numero $x = 0.01 \times 10^{-99}$; risulta $rd(x) = 0.1 \times 10^{-100}$ che non è un numero di macchina. In questo caso si ha una situazione di *underflow*, cioè il numero da rappresentare è "troppo piccolo" e non appartiene a M (non tutti i calcolatori segnalano questa situazione e nel caso in cui proseguano l'esecuzione del programma pongono $rd(x) = 0$).

Si dimostra che $rd(x)$ soddisfa la relazione

$$\mid rd(x) - x \mid \leq \mid z - x \mid, \quad \forall z \in M,$$

per cui, se $rd(x) \in M$, esso, in valore assoluto, differisce da x meno di qualunque altro numero di macchina.

Si definisce *errore assoluto* della rappresentazione del numero reale x il valore

$$\delta_x = rd(x) - x$$

ed *errore relativo* il valore

$$\epsilon_x = \frac{rd(x) - x}{x} = \frac{\delta_x}{x}.$$

Dalle precedenti considerazioni si ricavano le limitazioni

$$\mid \delta_x \mid \leq \frac{1}{2}\beta^{b-m},$$

$$\mid \epsilon_x \mid < \frac{1}{2}\beta^{1-m}.$$

Il numero $u = \frac{1}{2}\beta^{1-m}$ si dice *precisione di macchina*. In generale quando l'errore relativo di una approssimazione non supera $\frac{1}{2}\beta^{1-k}$ si dice che l'approssimazione è corretta almeno fino alla k-esima cifra significativa. Da quanto sopra segue quindi che $rd(x)$ approssima x almeno fino alla m-esima cifra significativa.

Se si assume la base $\beta = 2$ ogni cifra della rappresentazione di un numero ha valore 0 o 1 e si dice *bit* [1] (binary digit), mentre si dice *parola* di lunghezza t l'insieme dei t bit rappresentanti un numero. Per ragioni tecniche si usa spesso una base $\beta = 2^n$ $(n > 1)$ in cui ciascuna cifra, tradotta in rappresentazione binaria, richiede n bit.

Si distinguono la rappresentazione dei numeri interi da quella dei reali e l'aritmetica che opera solo con numeri interi da quella che opera indifferentemente con numeri interi e reali.

In 1.1.1, 1.1.2, 1.1.3, si farà riferimento al caso $t = 32$.

[1] Il termine bit è usato nel seguito in forma invariata anche al plurale.

1.1.1 Numeri interi

Per rappresentare un numero intero i 32 bit della parola vengono così utilizzati: un bit fornisce il segno del numero (per esempio 0 se il numero è positivo e 1 se è negativo), i restanti 31 bit servono a rappresentare in base 2 le cifre del numero dato. E' evidente che il massimo intero rappresentabile è, in valore assoluto, $2^{31} - 1 = 2147483647$.

1.1.2 Numeri reali (precisione semplice)

Nella rappresentazione dei numeri reali una situazione caratteristica è la seguente: la base è 16 (sistema esadecimale), un bit è riservato al segno del numero, sette bit sono riservati alla rappresentazione dell'esponente b, i restanti 24 bit sono utilizzati per le cifre α_i della mantissa che, in base 16, sono 6 in quanto occorrono 4 bit per rappresentare in binario una cifra esadecimale compresa tra 0 e 15. L'esponente b non è rappresentato in segno ma in traslazione rispetto a 64 e cioè viene rappresentato il numero $b^* = b+64$; ne segue che b è compreso tra -64 e 63.

Di conseguenza, indicando con F la cifra esadecimale che corrisponde al numero 15[2], il massimo numero rappresentabile (in precisione semplice) è $0.FFFFFF \times 16^{63} \simeq 7.23 \times 10^{75}$, il minimo numero non negativo rappresentabile risulta $0.1 \times 16^{-64} \simeq 5.39 \times 10^{-79}$ e la precisione di macchina è $u_s = \frac{1}{2}16^{-5} \simeq 4.75 \times 10^{-7}$.

1.1.3 Numeri reali (doppia precisione)

I numeri reali si dicono in doppia precisione quando sono rappresentati con una doppia parola e quindi con 64 bit. Rispetto alla precisione semplice cambia solo il numero delle cifre esadecimali della mantissa che da 6 divengono 14 in quanto essa viene ad includere i 32 bit aggiunti. L'insieme dei numeri rappresentabili non cambia molto mentre cambia la precisione di macchina che diviene $u_d = \frac{1}{2}16^{-13} \simeq 1.11 \times 10^{-16}$.

È possibile anche l'uso di precisioni multiple in cui si incrementa ulteriormente il numero delle cifre della mantissa, migliorando di conseguenza la precisione di macchina.

[2]Le cifre della rappresentazione in base 16 sono: $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$.

1.2 Le operazioni di macchina

Nell'insieme M non tutte le proprietà delle quattro operazioni elementari risultano verificate, in quanto il risultato di una operazione deve essere ricondotto ad un numero di macchina: perciò le operazioni elementari all'interno di una macchina sono diverse dalle corrispondenti operazioni ordinarie. Si indicano nel seguente modo le quattro *operazioni di macchina*:

$\oplus$ addizione
$\ominus$ sottrazione
$\otimes$ moltiplicazione
$\oslash$ divisione.

Un esempio in cui l'addizione ($\oplus$) non gode della proprietà associativa è il seguente.

Sia $F(10, 3, -99, 99)$ e sia $x = 0.135 \times 10^{-4}$, $y = 0.258 \times 10^{-2}$, $z = -0.251 \times 10^{-2}$; si ha

$$\begin{aligned} x \oplus (y \oplus z) &= 0.135 \times 10^{-4} \oplus \left(0.258 \times 10^{-2} \oplus -0.251 \times 10^{-2}\right) \\ &= 0.135 \times 10^{-4} \oplus 0.700 \times 10^{-4} \\ &= 0.835 \times 10^{-4}, \end{aligned}$$

mentre

$$\begin{aligned} (x \oplus y) \oplus z &= \left(0.135 \times 10^{-4} \oplus 0.258 \times 10^{-2}\right) \oplus -0.251 \times 10^{-2} \\ &= 0.259 \times 10^{-2} \oplus -0.251 \times 10^{-2} \\ &= 0.800 \times 10^{-4}. \end{aligned}$$

In modo analogo si possono dare esempi in cui l'operazione $\otimes$ non gode della proprietà distributiva rispetto alla addizione.

Quando si sottraggono due numeri di macchina dello stesso segno che hanno lo stesso esponente b e con le mantisse che differiscono di poco, si incorre in una perdita di cifre significative nel risultato. Tale fenomeno, detto *cancellazione*, produce, come si vedrà più avanti, una notevole amplificazione degli errori relativi.

1.3 Errore nel calcolo di una funzione

Una funzione non razionale φ viene sempre sostituita, all'interno di un calcolatore, da una funzione razionale f, il cui uso comporta un errore $f-\varphi$ che si dice *errore di troncamento.* Tale errore, all'occorrenza è facilmente maggiorabile.

Tuttavia anche nel calcolo di una funzione razionale $f(x_1, x_2, \ldots, x_n)$ in un punto assegnato $P_0 = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})$, in generale, non si ottiene il valore $f(P_0)$ cercato, a causa delle approssimazioni che si introducono.

Tali approssimazioni producono due tipi di errore.

Un primo errore nasce dal fatto che le operazioni aritmetiche che compaiono nella $f(P)$ devono essere sostituite con le corrispondenti operazioni di macchina e organizzate in un certo algoritmo e ciò equivale in definitiva alla sostituzione di $f(P)$ con un'altra funzione $f_a(P)$ che la approssimi.

Un secondo tipo di errore si presenta quando non è possibile rappresentare esattamente le coordinate del punto P_0 e quindi si devono approssimare tali coordinate con numeri di macchina (basti pensare, per esempio, al punto $P_0 = (\sqrt{2}, \pi)$).

1.3.1 Errore assoluto

Assegnato il punto $P_0 \in \mathbb{R}^n$ di coordinate $x_i^{(0)}$, $i = 1, 2, \ldots, n$, nel quale si vuole calcolare la funzione $f(P)$, si consideri l'insieme

$$D = \{P \in \mathbb{R}^n \mid a_i \leq x_i \leq b_i,\ i = 1, 2, \ldots, n\}$$

dove $a_i, b_i \in \mathbb{R}$ e si supponga che sia $a_i \leq x_i^{(0)} \leq b_i$, $i = 1, 2, \ldots, n$; D si dice *insieme di indeterminazione* del punto P_0.

In effetti il valore cercato $f(P_0)$ viene sostituito dal valore calcolato $f_a(P_1)$ dove P_1, di coordinate $x_i^{(1)}$, $i = 1, 2, \ldots, n$, appartiene a D e quindi l'errore commesso risulta $f_a(P_1) - f(P_0)$; di questo errore, detto *errore totale*, si può dare una stima.

Si pone

$$f_a(P_1) - f(P_0) = f_a(P_1) - f(P_1) + f(P_1) - f(P_0)$$

dove la differenza $f_a(P_1) - f(P_1)$ è detta *errore algoritmico* mentre la differenza $f(P_1) - f(P_0)$ è detta *errore trasmesso dai dati.*

L'errore algoritmico, una volta fissato l'algoritmo che fornisce $f_a(P)$, risulta definito e stimabile.

All'errore trasmesso, nell'ipotesi che $f(P) \in C^1(D)$, si può dare una rappresentazione generale: dalla formula di Taylor arrestata al primo termine e con punto iniziale P_0 si ottiene

$$f(P_1) - f(P_0) = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(x_i^{(1)} - x_i^{(0)}) \tag{1.3}$$

dove le derivate parziali della funzione $f(P)$ sono calcolate in un punto opportuno. Indicando con

$$\begin{array}{ll} \delta_f = f_a(P_1) - f(P_0) & \text{l'errore totale,} \\ \delta_a = f_a(P_1) - f(P_1) & \text{l'errore algoritmico,} \\ \delta_d = f(P_1) - f(P_0) & \text{l'errore trasmesso dai dati,} \end{array}$$

risulta

$$\delta_f = \delta_a + \delta_d.$$

Ponendo poi

$$\delta_{x_i} = x_i^{(1)} - x_i^{(0)}, \quad \rho_i = \frac{\partial f}{\partial x_i},$$

la (1.3) diventa

$$\delta_d = \sum_{i=1}^{n} \rho_i \delta_{x_i}.$$

I valori ρ_i sono detti *coefficienti di amplificazione* degli gli errori δ_{x_i}.

Una limitazione per il modulo dell'errore assoluto $f_a(P_1) - f(P_0)$ è

$$| f_a(P_1) - f(P_0) | \leq E_a + E_d$$

dove si è posto

$$\begin{array}{lll} E_a & = & \text{massimo modulo dell'errore assoluto dovuto} \\ & & \text{al particolare algoritmo usato,} \end{array}$$

$$E_d = \sum_{i=1}^{n} A_{x_i} \mid \delta_{x_i} \mid, \tag{1.4}$$

con

$$A_{x_i} \geq \sup_{x \in D} \left| \frac{\partial f}{\partial x_i} \right|, \quad i = 1, 2, \ldots, n. \tag{1.5}$$

Se si conosce una stima dell'errore di troncamento e degli errori δ_{x_i} nonché le A_{x_i}, si può stabilire a posteriori un confine superiore per l'errore assoluto con cui si è calcolata la funzione nel punto desiderato; questo problema è detto *problema diretto*.

Il *problema inverso* consiste nel richiedere a priori che il valore $f_a(P_1)$ sia tale che l'errore assoluto $\mid f_a(P_1) - f(P_0) \mid$ risulti minore di un valore prefissato, per cui si deve cercare sia un algoritmo $f_a(P)$ sia un opportuno punto P_1 che soddisfino la richiesta.

1.3.2 Errore relativo

L'*errore relativo* che si commette nel calcolo di una funzione $f(P)$ in un assegnato punto P_0 è definito da

$$\epsilon_f = \frac{f_a(P_1) - f(P_0)}{f(P_0)}.$$

Si verifica facilmente che

$$\epsilon_f = \frac{f(P_1) - f(P_0)}{f(P_0)} + \frac{f_a(P_1) - f(P_1)}{f(P_1)} \left(1 + \frac{f(P_1) - f(P_0)}{f(P_0)}\right)$$

per cui, indicando con

$$\epsilon_a = \frac{f_a(P_1) - f(P_1)}{f(P_1)}$$

l'*errore relativo algoritmico* e con

$$\epsilon_d = \frac{f(P_1) - f(P_0)}{f(P_0)}$$

l'*errore relativo trasmesso dai dati*, si ottiene

$$\epsilon_f = \epsilon_a + \epsilon_d + \epsilon_a \epsilon_d. \tag{1.6}$$

Nella (1.6) si trascura il termine $\epsilon_a \epsilon_d$ in quanto di ordine superiore.

Quanto all'errore relativo trasmesso dai dati, dalla relazione (1.3) si ricava

$$\epsilon_d = \frac{f(P_1) - f(P_0)}{f(P_0)} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \frac{(x_i^{(1)} - x_i^{(0)})}{f(P_0)}$$

ed ancora, definendo $\epsilon_{x_i} = \frac{x_i^{(1)} - x_i^{(0)}}{x_i^{(0)}}$,

$$\epsilon_d = \sum_{i=1}^{n} \frac{x_i^{(0)}}{f(P_0)} \frac{\partial f}{\partial x_i} \epsilon_{x_i}.$$

Infine, ponendo $\gamma_i = \frac{x_i^{(0)}}{f(P_0)} \frac{\partial f}{\partial x_i}$, si ha

$$\epsilon_d = \sum_{i=1}^{n} \gamma_i \epsilon_{x_i} \ ,$$

dove i valori γ_i sono detti *coefficienti di amplificazione* degli errori relativi.

Se i coefficienti γ_i sono tali che l'errore ϵ_d risulta dello stesso ordine degli errori ϵ_{x_i} il problema del calcolo della funzione si dice *ben condizionato*; si dice invece *mal condizionato* se a piccoli errori relativi ϵ_{x_i} corrisponde un errore ϵ_d rilevante.

Dalla (1.6) si vede che all'errore ϵ_f concorre anche l'errore algoritmico ϵ_a: se l'algoritmo è tale da produrre errori accettabilmente limitati nella sua applicazione, si dice *stabile*, mentre è detto *instabile* nel caso contrario.

1.4 Gli errori nelle quattro operazioni

Analizzando le quattro operazioni fondamentali per quanto riguarda gli errori trasmessi dai dati si ottiene la seguente tabella:

operazione	δ_d	ϵ_d
$x \oplus y$	$\delta_x + \delta_y$	$\frac{x}{x+y}\epsilon_x + \frac{y}{x+y}\epsilon_y$
$x \ominus y$	$\delta_x - \delta_y$	$\frac{x}{x-y}\epsilon_x - \frac{y}{x-y}\epsilon_y$
$x \otimes y$	$y\delta_x + x\delta_y$	$\epsilon_x + \epsilon_y$
$x \oslash y$	$\frac{1}{y}\delta_x - \frac{x}{y^2}\delta_y$	$\epsilon_x - \epsilon_y$

Si deduce che le operazioni di addizione e sottrazione non danno problemi per quanto riguarda l'errore assoluto, mentre possono rendere grande l'errore relativo nel caso in cui i due termini dell'operazione siano molto vicini in valore assoluto, in quanto può accadere che i denominatori che compaiono nei coefficienti di amplificazione dell'errore relativo siano molto piccoli in valore assoluto (fenomeno della cancellazione già accennato in 1.2). La moltiplicazione non amplifica l'errore relativo e comporta un errore assoluto che dipende dall'ordine di grandezza dei fattori; anche la divisione non produce amplificazione per quanto riguarda l'errore relativo, mentre l'errore assoluto diminuisce se aumenta (in valore assoluto) il divisore.

1.5 Complementi ed esempi

Si riportano due esempi sul calcolo degli errori assoluti e relativi.

Esempio 1.5.1 Si vuole calcolare la funzione $f(x_1, x_2) = x_1/x_2$ nel punto assegnato $P_0 = (\sqrt{5}, \pi)$.

Si assuma $D = \{(x_1, x_2) \in \mathbb{R}^2 | 2.23 < x_1 < 2.24,\ 3.14 < x_2 < 3.15\}$ come insieme di indeterminazione del punto P_0 (dalle (1.4) e (1.5) si ricava che più si riduce l'insieme D e più stringente è la maggiorazione dell'errore assoluto).

Problema diretto

Si calcola il rapporto x_1/x_2 arrotondando il risultato alla seconda cifra decimale, ottenendo quindi un errore $E_a \leq \frac{1}{2}10^{-2}$.
Assumendo $P_1 = (2.235, 3.145)$ si ha sicuramente $| \delta_{x_1} |, | \delta_{x_2} | \leq \frac{1}{2}10^{-2}$.

Dalla funzione e dall'insieme D si traggono le maggiorazioni

$$\sup_{x \in D} \left| \frac{\partial f}{\partial x_1} \right| \leq A_{x_1} = 0.32,$$

$$\sup_{x \in D} \left| \frac{\partial f}{\partial x_2} \right| \leq A_{x_2} = 0.23;$$

da cui si ha il massimo errore assoluto

$$E_a + E_d \leq \frac{1}{2}10^{-2} + (0.32 + 0.23)\frac{1}{2}10^{-2} = 0.775 \times 10^{-2}.$$

Problema inverso

Si vuole avere un valore che differisca dal valore esatto $f(P_0)$ meno di un prefissato errore $E = 10^{-2}$.

Si può imporre che E_a ed E_d siano, ciascuno, non superiore a metà dell'errore totale richiesto e quindi che sia $E_a \leq \frac{1}{2}10^{-2}$ e $E_d \leq \frac{1}{2}10^{-2}$.

Per l'errore di troncamento è sufficiente arrotondare il risultato della divisione alla seconda cifra decimale.

Poiché l'errore trasmesso dai dati è formato dalla somma di due addendi, si suddivide ancora il contributo a tale errore in due parti uguali per cui si ha

$$A_{x_1} \mid \delta_{x_1} \mid \leq \frac{1}{4}10^{-2},$$
$$A_{x_2} \mid \delta_{x_2} \mid \leq \frac{1}{4}10^{-2};$$

assumendo per A_{x_1} e A_{x_2} i valori calcolati in precedenza si ottiene

$$\mid \delta_{x_1} \mid \leq \frac{1}{4}10^{-2}\frac{1}{0.32} \simeq 0.0078,$$
$$\mid \delta_{x_2} \mid \leq \frac{1}{4}10^{-2}\frac{1}{0.23} \simeq 0.0109,$$

per cui si può porre $P_1 = (2.24, 3.14)$.

Si ha quindi $f_a(P_1) = 0.71$ che differisce da $f(P_0)$ meno di E. $\square$

Esempio 1.5.2 Si vuole stimare l'errore relativo commesso nel calcolo della funzione $f(x, y, z, w) = x(y/z - w)$.

Si può ricorrere all'uso dei *grafi*; si calcola la funzione eseguendo una operazione dopo l'altra e stabilendo per ciascuna di esse l'entità degli errori relativi.

In questo esempio la sequenza delle operazioni è

$$r_1 = \frac{y}{z}, \qquad r_2 = r_1 - w, \qquad r_3 = xr_2 \; .$$

Il grafo in Fig. 1.1 evidenzia per ogni operazione i coefficienti di amplificazione lungo i cammini orientati. Gli errori relativi dei dati sono $\epsilon_x, \epsilon_y, \epsilon_z, \epsilon_w$, mentre ϵ_{r_i} va inteso come l'errore relativo per la funzione r_i e si calcola dalla (1.6) senza l'ultimo termine.

Per stimare l'errore relativo totale si procede a ritroso seguendo il grafo. Indicando con ϵ_i l'errore algoritmico della i-esima operazione si ha

$$\begin{aligned}
\epsilon_f &= \epsilon_{r_3} \\
&= \epsilon_3 + \epsilon_x + \epsilon_{r_2} \\
&= \epsilon_3 + \epsilon_x + \epsilon_2 + \frac{y}{y - zw}\epsilon_{r_1} - \frac{zw}{y - zw}\epsilon_w \\
&= \epsilon_3 + \epsilon_x + \epsilon_2 + \frac{y}{y - zw}(\epsilon_1 + \epsilon_y - \epsilon_z) - \frac{zw}{y - zw}\epsilon_w \\
&= \epsilon_3 + \epsilon_2 + \frac{y}{y - zw}\epsilon_1 + \epsilon_x + \frac{y}{y - zw}(\epsilon_y - \epsilon_z) - \frac{zw}{y - zw}\epsilon_w \; .
\end{aligned}$$

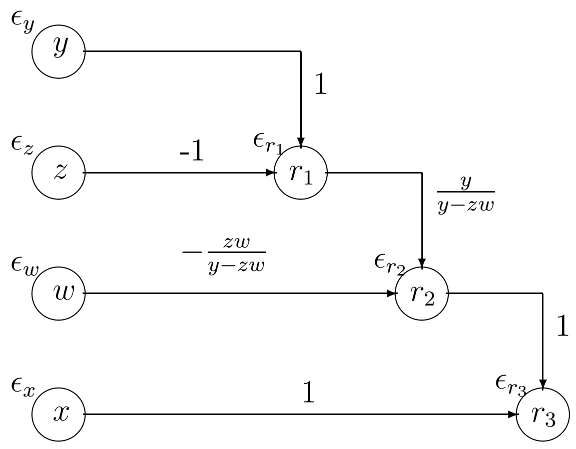

Figura 1.1: Esempio di grafo.

Se ne conclude che l'errore algoritmico è

$$\epsilon_a = \epsilon_3 + \epsilon_2 + \frac{y}{y - zw}\epsilon_1$$

e l'errore trasmesso è

$$\epsilon_d = \epsilon_x + \frac{y}{y - zw}(\epsilon_y - \epsilon_z) - \frac{zw}{y - zw}\epsilon_w \ .$$

$\square$

Osservazione 1.5.1 L'errore relativo calcolato nell'Esempio 1.5.2 dipende dall'algoritmo seguito per il calcolo della funzione $f(x, y, z, w)$; seguendo un altro algoritmo si trova, in generale, un errore relativo diverso.

Bibliografia: [1], [5], [28], [29].