In questo modulo saranno descritte le modalità di rappresentazione dei numeri in virgola mobile e le rispettive proprietà e caratteristiche che gli identificano.

Aritmetica Floating - Point

Prof. Michele Tarantino

Tutti i diritti riservati.
Il presente testo può essere utilizzato liberamente per motivi di studio, didattica e attività di ricerca purché sia presente il riferimento bibliografico.

L’aritmetica floating-point o aritmetica in virgola mobile, identifica un metodo di rappresentazione dei numeri razionali e di approssimazione di numeri reali. Si può definire un’aritmetica come l’insieme delle proprietà elementari (addizione, sottrazione, moltiplicazione e divisione) sui numeri. In questo caso, le operazioni di base sono effettuate sui numeri floating-point.

NUMERI FLOATING-POINT

In tutti i moderni calcolatori la più piccola unità di informazione è rappresentata da bit (*Binary DIgit*) che può assumere valori 0 oppure 1 (sistema binario). In molti calcoli la quantità di cifre di un numero che deve essere utilizzata è troppo grande per essere utilizzata come stringa di bit direttamente dal calcolatore. Basta pensare a calcoli scientifici di importanza vitale, dove bisogna considerare anche ottanta o più cifre significative dopo la virgola. Si potrebbe utilizzare l’aritmetica a precisione multipla per ottenere cifre più significative. Però, quello che si vuole utilizzare è sostanzialmente un sistema per la rappresentazione dell’informazione (o più specificatamente di numeri) che sia indipendente dal numero delle cifre significative che si vuole esprimere.

Un modo per rappresentare tale sistema si basa sulla notazione scientifica di un numero: la notazione scientifica è un modo conciso per esprimere i numeri reali utilizzando le potenze intere di dieci, ed è usata per numeri molto grandi o molto piccoli senza includere lunghe file di zeri. Dato che tutti i calcolatori utilizzano un sistema di numerazione binario, si sostituisce la base decimale usualmente utilizzata in diversi campi scientifici, con la base binaria (o base 2). Quindi, ogni numero n può essere espresso come:

$$n = f * b^e$$

dove f viene definita come frazione o mantissa, e è un numero intero positivo o negativo denominato esponente e b è la base di rappresentazione del numero (nella versione informatica si ha b=2). La quantità di numeri esprimibile dipende dal numero di cifre dell’esponente e la precisione viene determinata dal numero di cifre della frazione.

In tutti i moderni calcolatori l'uso di operazioni aritmetiche in virgola mobile è ad oggi il metodo più diffuso per la gestione di numeri reali. La mantissa di un numero scritto con questo metodo si presenta quindi nella forma ±d.ddd...ddd (una quantità *p* di cifre *d* comprese tra 0 e b-1). Se la prima cifra della mantissa non è zero, il numero è definito normalizzato, altrimenti denormalizzato.

STANDARD FLOATING-POINT IEEE 754

Fino al 1980 circa, ogni produttore di microprocessori utilizzava un formato per la rappresentazione di numeri floating-point. L'obiettivo di standardizzazione rappresenta in generale una base di riferimento per la produzione di tecnologie fra loro compatibili. Lo standard IEEE per il calcolo in virgola mobile (IEEE 754) (ufficialmente definito come *IEEE Standard for Binary Floating-Point Arithmetic (ANSI/IEEE Std 754-1985) o anche IEC 60559:1989, Binary floating-point arithmetic for microprocessor systems*) fu definito solamente nel 1985 ed è attualmente lo standard più diffuso nel campo del calcolo automatico. Questo standard definisce il formato per la rappresentazione dei numeri in virgola mobile (compreso ±0), i numeri denormalizzati, gli infiniti e i *NaN* ("*Not a Number*" utilizzato per gestire numeri indefiniti, come la divisione per zero), ed un insieme di operazioni effettuabili su questi. Inoltre, specifica quattro metodi di arrotondamento e ne descrive cinque eccezioni. Con questo standard vengono definiti quattro formati per i numeri in virgola mobile:

- a precisione singola (32 bit);
- precisione doppia (64 bit);
- precisione singola estesa (≥ 43 bit) usato raramente;
- precisione doppia estesa (≥ 79 bit, supportata solitamente con 80 bit).

La precisione singola è il minimo richiesto dallo standard, gli altri sono opzionali.

Secondo lo standard un numero in virgola mobile è rappresentato su parole di 32, 64 o 80 bit divisi in tre parti ciascuna congiunta alla successiva nel seguente ordine:

- un bit di segno *s*;
- un campo di esponente *e*;

- un campo di mantissa *m*.

I bit di una parola composta da *n* bit sono indicizzati in modo decrescente con numeri interi da 0 a *n-1* e l'importanza del bit decresce con il suo indice.

Graficamente possiamo rappresentare i diversi formati dei numeri floating-point come in Figura 1:

Figura 1: Rappresentazione di un numero floating-point

Il valore del numero rappresentato è calcolabile come:

$$(-1)^s * 2^e * m$$

Il campo *s* specifica il segno del numero: 0 per i numeri positivi, 1 per i numeri negativi. Il campo *e* contiene l'esponente del numero in forma intera: se è costituito da *n* bit, si hanno 2^n possibili valori. Solitamente ad alcuni di questi valori sono riservate funzioni speciali mentre gli altri permettono di rappresentare tutti gli altri valori per i numeri in forma normale.

A seconda della precisione vengono assegnati per l'esponente e la mantissa un numero di bit prefissato. Nella Tabella 1 sono riportati, per le due rappresentazioni principali, il numero di bit utilizzati rispettivamente per l'esponente e la mantissa (il segno utilizza solo un bit per tutti i formati).

Formato	**Esponente**	**Mantissa**
Precisione singola	8	23
Precisione doppia	11	52

Tabella 1: Formati di rappresentazione principali

Nel caso venga utilizzata la precisione singola, dato che l'esponente è composto da 8 bit, tale formato permette di rappresentare 256 (2^8) valori. Ai valori 0 e 255 vengono riservate

funzioni speciali (descritte in Tabella 2) mentre gli altri numeri permettono di rappresentare 254 valori per i numeri in forma normale, compresi tra -126 e 127.

I numeri vengono distinti in prima istanza dall'esponente e in seconda istanza dalla mantissa, ed in base ai valori assunti si determina l'appartenenza ad una categoria descritta in Tabella 2.

Per i valori floating-point a doppia precisione cambiano sostanzialmente gli intervalli di rappresentazione che come definito precedentemente, dipendono dall'esponente.

Categoria	**Esponente**	**Mantissa**
Zeri	0	0
Numeri denormalizzati	0	Diverso da zero
Numeri normalizzati	1-254	Qualsiasi
Infiniti	255	0
NAN (Not A Number)	255	Diverso da zero

Tabella 2: Categorie dei valori floating-point a singola precisione

La mantissa, rappresentato dal campo *m*, è una stringa di bit che rappresenta la sequenza di cifre dopo la virgola. Tutte le mantisse sono normalizzate: questo significa che il numero prima della virgola deve essere uguale ad 1; per cui, per un dato valore *M* il valore matematico corrispondente è:

$$M=1,m$$

In pratica, la mantissa è costituita dal numero binario 1, seguito dalla virgola e dalla parte intera del numero rappresentato ovviamente in forma binaria. Quando un numero è normalizzato, il primo bit della mantissa (che è pari ad 1) viene omesso per convenienza: in questo modo si risparmia un bit e si può aumentare l'intervallo di numeri rappresentabili. Il primo bit ad 1 che viene omesso è definito come bit nascosto o bit implicito.

Con questo sistema di rappresentazione sorge un problema: tutti i numeri che rientrano nell'intervallo hanno sia un valore positivo (impostando il bit di segno a zero) sia negativo (impostando il bit di segno ad uno); quindi anche il numero 0 ha due rappresentazioni (+0 e −0) a seconda del valore del primo bit. Tale rappresentazione però è utile solamente in analisi matematica.

PROPRIETÀ DELL'ARITMETICA FLOATING-POINT

L'aritmetica floating-point presenta diverse proprietà dall'aritmetica reale: lavora su insiemi finiti, e non valgono alcune proprietà che valgono per l'aritmetica dei numeri reali. In particolare, non valgono le seguenti proprietà:

- associativa: l'ordine di valutazione è irrilevante se l'operazione appare più di una volta in un'espressione:

$$(x + y) + z \neq x + (y + z)$$

$$(x * y) * z \neq x * (y * z)$$

- distributiva (a sinistra) rispetto all'operazione di moltiplicazione:

$$x * (y + z) \neq (x * y) + (x * z)$$

L'invalidità di queste due proprietà deriva dal fatto che l'ordine in cui vengono eseguite più operazioni in virgola mobile può variarne il risultato. Per esempio, nella maggior parte delle applicazioni in virgola mobile, 1,0 + (10100 + -10100) fornisce come risultato 1,0, mentre (1,0 + 10100) + -10100 fornisce come risultato 0,0.

Inoltre, esiste l'elemento neutro della moltiplicazione e dell'addizione, ma questi non sono unici per quanto definito nel precedente paragrafo.

CONFRONTO TRA NUMERI REALI E NUMERI FLOATING-POINT

I numeri floating-point non sono in grado di rappresentare ogni singolo numero reale, in quanto i calcolatori lavorano su grandezze finite e quindi la gamma di valori possibili è delimitata da un estremo superiore ed un estremo inferiore, che dipendono come definito precedentemente dall'esponente e dalla precisione. Già di per sé, la definizione matematica non formale di numero reale, lo definisce come quel numero a cui è possibile attribuire uno sviluppo decimale infinito. Questo nei calcolatori non è possibile. Si possono però utilizzare i numeri floating-point per simulare il sistema di numeri reali della matematica anche se sono presenti alcune importanti differenze:

1- numeri negativi più piccoli dell'estremo inferiore (*underflow*);

2- numeri positivi più grandi dell'estremo superiore (*overflow*);

3- numeri positivi o negativi più piccoli della precisione ammissibile (assorbimento).

Un'altra importante differenza è che presi due numeri reali qualsiasi diversi tra loro a e b esiste sempre un insieme infinito di numeri reali compresi nell'intervallo $[a,b]$. Questo non è possibile con i numeri floating-point, in quanto sono finiti, e nello stesso intervallo $[a,b]$ ci possono essere zero, uno o più numeri floating-point. Ciò significa che l'aritmetica in virgola mobile provoca degli errori di arrotondamento. In particolare, l'errore relativo è variabile ma è sempre al di sotto del valore trovato il quale non dipende da p e quindi non dipende dal numero rappresentato, come invece fa l'errore assoluto che viene anche detto precisione macchina.

Un altro grosso problema che si verifica con i numeri floating-point e la rispettiva aritmetica è l'implementazione corretta in hardware. Il calcolo floating-point è difficile da implementare correttamente per il progettista hardware medio. Nei prossimi due capitoli sarà affrontato lo studio della verifica di correttezza di hardware floating-point mediamente diverse tecniche di analisi.

Resta connesso e informato sui prossimi eventi, corsi e seminari:

Web
www.profmicheletarantino.com

Email
profmicheletarantino@gmail.com

Telefono
349 83 54 521

Facebook
@micheletarantinodocente

Instagram
@profmicheletarantino

Hai bisogno di un modulo personalizzato? Non esitare a contattarmi!