COMPLESSITÀ ASTRATTA

Quando parliamo di complessità astratta non ci soffermiamo a discutere di cosa stiamo parlando, cioè quali tipo di oggetti stiamo valutando (un programma, un'immagine, un suono, un segnale digitale ecc.). Quando abbiamo oggetti molto diversi da esaminare ci aspettiamo intuitivamente che le misure di complessità di questi oggetti siano molto diversi e incompatibili fra di loro. Le considerazioni che facciamo a proposito di una misura non dovrebbero potersi trasferire nello studio di ogni altra misura. Questa intuizione è sbagliata infatti il **teorema del collegamento ricorsivo** dimostra che tutte le misure di complessità sono collegate fra loro.
Un altro concetto intuitivo che siamo portati a pensare è che la complessità dipende dalle limitazioni sulle risorse di calcolo utilizzabili, cioè se avessimo macchine più potenti potremmo fare più cose. In realtà non è così, infatti il **teorema della lacuna** dimostra che se la funzione che limita il calcolo ha certe caratteristiche, pur aumentando questi limiti non riusciamo a fare più cose.

DEFINIZIONE: Una funzione parziale binaria C su N si chiama **misura di complessità** se soddisfa i due assiomi seguenti, detti **assiomi di Blum**:

1. $C(x, i)$ è definita se e solo se $\Phi_i(x)$ è definita
2. Il predicato $C(x, i) \leq y$ è ricorsivo

La complessità C di un programma dipende dal programma stesso e dall'input che diamo a quel programma. Il primo assioma dice che C converge se e solo se la macchina universale che calcola il programma con numero di Gödel i e input x converge, cioè ogni volta che il programma termina allora si deve poter dare un misura della sua complessità. Il secondo assioma dice che C deve avere un limite superiore y e questo limite è calcolabile.

Esempio I: Sia $C(x, i) = C_i(x)$ = numero di passi in un calcolo effettuato dal programma i su input x.
Assioma 1: Se un programma termina dandomi un valore allora STP termina dandomi il numero dei passi che ha fatto.
Assioma 2: Se prendo un arbitrario valore di y posso dire se il programma con input x si ferma entro y.

Esempio II: $C_i(x)$ = massimo valore assunto da ogni variabile del programma numero i con input x, se $\Phi_i(x)$ è definita; altrimenti $C_i(x)$ non è definita.
Assioma 1: è soddisfatto per definizione.
Assioma 2: indichiamo il massimo valore assunto dalle variabili con $M_i(x)$. Per un programma dato sono possibili un numero finito di descrizioni istantanee nelle quali tutte le variabili assumono valori inferiori o uguali ad un certo y. Possiamo quindi controllare se, dati i, x, y, la condizione $M_i(x) \leq y$ è verificata o meno. Si possono verificare tre casi:

- Il calcolo termina con tutti i valori delle variabili che hanno valore ≤ y. Diamo il valore VERO.
- Raggiungiamo una configurazione nella quale almeno una variabile ha il valore > y. Diamo il valore FALSO.
- Si passa due volte per la stessa configurazione, cioè si è entrati in un ciclo infinito. Diamo il valore FALSO.

L'aver impostato una misura di complessità in questo modo non falsifica il teorema della fermata: se ci troviamo in un ciclo infinito che per esempio incrementa una variabile e ritorna all'istruzione precedente, a un certo punto la variabile diventa più grande di y e quindi il predicato si ferma e restituisce FALSO. Esso non si ferma perché capisce che il programma è in un ciclo infinito: questa è la differenza con il predicato HALT.

Esempio III: La funzione stessa non può essere presa come misura di complessità. Infatti mentre il primo assioma è banalmente soddisfatto, il secondo non lo è:
Sia $\Phi_i(x) = 0$ se $x \in S$, non definita altrimenti, dove S è un insieme ricorsivamente enumerabile, ma non ricorsivo. La condizione $\Phi_i(x) \leq 0$ è equivalente a $x \in S$ e quindi non è ricorsiva, contrariamente a quanto richiesto dall'assioma 2. Quindi non possiamo usare la funzione stessa come misura di complessità perché non abbiamo nessuna garanzia che la funzione converga sempre.

Definiamo **fattore di scala** una funzione ricorsiva $r(x)$ se
- r è crescente (cioè se $x \leq y$ allora $r(x) \leq r(y)$)
- r assume valori arbitrariamente grandi

NOTA: una funzione di scala è una funzione che prende un'altra funzione e la ingrandisce o la riduce di un determinato fattore.

Osserviamo il secondo assioma di Blum: $C(x, i) \leq y$ è ricorsivo. Riscriviamolo in modo separato:
- Il predicato $C(x, i) = y$ è ricorsivo
- Il predicato $C(x, i) < y$ è ricorsivo

Scriviamo ciascuno dei predicati in modo opportuno usando solo operazioni ricorsive:
- $C(x, i) = y \Leftrightarrow (C(x, i) \leq y \,\&\, \sim(C(x, i) \leq y \dot{-} 1)) \vee (y = 0 \,\&\, C(x, i) \leq y)$
- $C(x, i) < y \Leftrightarrow (\exists z)_{<y} (C(x, i) = z)$
- $C(x, i) \leq y \Leftrightarrow C(x, i) < y+1$

PROPOSIZIONE: Sia $C_i(x)$ una misura di complessità e sia $r(x)$ un fattore di scala.
Poniamo $D_i(x) = r(C_i(x))$. $D_i(x)$ è una misura di complessità.
DIM: Per dimostrare che $D_i(x)$ è una misura di complessità dobbiamo dimostrare che i due assiomi di Blum sono ancora validi:
Il primo assioma è soddisfatto perché la composizione di r e C_i non modifica il dominio di definizione di C_i. Ora dobbiamo verificare che $D_i(x) \leq y$ è ricorsivo:
Abbiamo che $D_i(x) = r(C_i(x)) \leq y$; sia t il numero tale che $r(0) \leq r(1) \leq \ldots \leq r(t) \leq y < r(t+1)$
Mostriamo adesso che $D_i(x) \leq y \Leftrightarrow C_i(x) \leq t$:
- se $C_i(x) \leq t$ è vero allora $D_i(x) = r(C_i(x)) \leq r(t) \leq y$
- se $C_i(x) \leq t$ è falso allora $C_i(x) \geq t+1$ e quindi $D_i(x) = r(C_i(x)) \geq r(t+1) > y$

TEOREMA (del collegamento ricorsivo tra misure di complessità):
Siano C e D due misure di complessità arbitrarie. Allora esiste una funzione ricorsiva $r(x, y)$ tale che $r(x, y) < r(x, y+1)$ e, per ogni i, $C_i(x) \leq r(x, D_i(x))$ quasi ovunque e $D_i(x) \leq r(x, C_i(x))$ quasi ovunque.
In altri termini: se abbiamo due misure di complessità arbitrarie C e D, esiste una relazione che lega tra loro i limiti di queste due misure di complessità.

Sia C una misura di complessità fissata e sia $t(x)$ un limite alla complessità di calcolo (limite imposto dall'esterno, per esempio la potenza della macchina). Tra i valori di x tali che $\Phi_i(x)$ è definita, abbiamo la possibilità di calcolare i valori assunti da Φ_i solo per quegli x tali che $C_i(x) \leq t(x)$.
Supponiamo ora che tale limite venga innalzato mediante una funzione ricorsiva g che ci fa passare quindi da $t(x)$ a $g(t(x))$. Intuitivamente ci aspettiamo di poter effettuare più calcoli dal momento che abbiamo a disposizione una macchina più potente. Il teorema della lacuna ci dice che per certe funzioni ricorsive t quest'affermazione non vale sempre: otteniamo un miglioramento solo per un numero finito di x e non per ogni x.
TEOREMA (della lacuna): Sia $g(x, y)$ una qualsiasi funzione ricorsiva tale che $g(x, y) > y$. Allora esiste una funzione ricorsiva $t(x)$ tale che, per $x > i$, $C_i(x) < g(x, t(x))$ implica che $C_i(x) \leq t(x)$, cioè se $g(x, t(x))$ è un limite per la funzione di complessità $C_i(x)$, allora lo è anche $t(x)$.

Siano C e D due calcolatori con C molto più lento di D.
Siano $C_i(x)$ il tempo di calcolo di C e $D_i(x)$ quello di D, quando gira il programma numero i su input x.
Applichiamo a C_i e D_i il teorema del collegamento ricorsivo:
Esiste una funzione ricorsiva r tale che $C_i(x) \leq r(x, D_i(x))$ quasi ovunque.
Definiamo $g(x, y) = r(x, y)+y+1$
$g(x, y) > y$ e $g(x, y+1) > g(x, y)$ quindi
$C_i(x) \leq g(x, D_i(x))$ quasi ovunque.
Sia $t(x)$ una funzione che soddisfa il teorema della lacuna per C e rispetto a g.
Sia P un programma numero i tale che $D_i(x) \leq t(x)$ quasi ovunque. Avremo che:
$C_i(x) \leq g(x, D_i(x)) \leq g(x, t(x))$ quasi ovunque
e per il teorema della lacuna $C_i(x) \leq t(x)$ quasi ovunque.