Capitolo 1

Introduzione alla modellazione solida

1.1 Introduzione

Testo: M. Mantyla, *An Introduction to Solid Modeling*, Computer Science Press. Collocazione: `MAT 68-1988-24IN`, `MAT 68-1988-48IN`.

La **Modellazione Solida** è un sottoinsieme della **Modellazione geometrica**.

Per "modellazione solida" si intende lo sviluppo di *oggetti solidi* adatti a rispondere a "*query*" geometriche arbitrarie *in modo algoritmico.*

Nella modellazione si opera su tre livelli:

Livello 1. Spazio reale: oggetti fisici che costituiscono il nostro mondo tridimensionale;

Livello 2. Spazio di modellazione: oggetti matematici (idealizzazione degli oggetti fisici);

Livello 3. Spazio delle rappresentazioni.

Il compito della modellizzazione è assegnare ad un oggetto matematico una rappresentazione adatta per essere manipolabile con un elaboratore.

1.2 Richiami di topologia

Sia X un insieme.

Definizione 1 *Una* topologia *per X è una famiglia T di sottoinsiemi di X che soddisfano le seguenti tre proprietà:*

- *X e l'insieme vuoto appartengono a T;*
- *l'unione di una collezione qualunque di elementi di T appartiene a T;*
- *l'intersezione di una collezione finita qualunque di elementi di T appartiene a T.*

Definizione 2 *Gli elementi di T sono detti* insiemi aperti.

Definizione 3 *Uno* spazio topologico *è una coppia (X, T), con T come dalla definizione 1.*

Definizione 4 *Un sottoinsieme C di X è detto* chiuso $\Leftrightarrow X - C$ $(\equiv \mathcal{C}_X C)$ *è aperto.*

Definizione 5 *Un* intorno di un punto $x \in X$ *è un qualunque aperto contenente x.*

Definizione 6 *Dato $x \in X$. x è detto* punto di accumulazione *di $A \subset X \Leftrightarrow$ ogni intorno di x contiene un punto di A diverso da x.*

Definizione 7 *La* chiusura di un insieme $A \subset X$ *(indicata con $\overline{A}$ oppure* $\mathrm{ch}(A)$*) è definita come*

$$\mathrm{ch}(A) = A \cup \{\text{insieme dei punti di accumulazione di } A\}$$

Definizione 8 *Il* contorno *(o* frontiera*) di A (indicato con* $\mathrm{b}(A) = \dot{A}$*) è definito come l'intersezione della chiusura di A con la chiusura del suo complementare:*

$$\mathrm{b}(A) = \overline{A} \cap \overline{\mathcal{C}_X A}$$

Definizione 9 *L'*interno *di A (indicato con* $\mathrm{i}(A) = \mathring{A}$*) è il più grande aperto contenuto in A.*

Nota.

- $\mathrm{b}(A) = \mathrm{ch(A)} - \mathrm{i}(A)$
- $\mathrm{i}(A) = \mathrm{ch(A)} - \mathrm{b}(A)$

1.3 Modelli matematici di oggetti tridimensionali

Come modelli matematici di *oggetti solidi* consideriamo gli *r-set*.

Definizione 10 *Un* r-set *è un sottoinsieme connesso di $\mathbf{R}^3$ che gode delle seguenti proprietà:*

1. *limitato;*
2. *chiuso;*
3. *rigido;*
4. *regolare;*
5. *dotato di contorno descrivibile mediante superfici algebriche.*

Definizione 11 *Per* trasformazioni rigide *si intendono traslazioni e rotazioni.*

Definizione 12 *Un* solido rigido *è una classe di equivalenza di punti in $\mathbf{R}^3$ descritta dalla relazione "$\sim$": siano A, B due sottoinsiemi di $\mathbf{R}^3$, $A \sim B \Leftrightarrow A$ può essere trasformato in B mediante trasformazioni rigide.*

Definizione 13 *Sia $A \subset \mathbf{R}^3$. La* regolarizzazione *di A, $r(A)$, è definita come la chiusura del suo interno:*

$$\mathrm{r}(A) = \mathrm{ch}(\mathrm{i}(A))$$

Definizione 14 *Un insieme A è detto* regolare *se e solo se* $\mathrm{r}(A) \equiv A$ *(cioè A coincide con la sua regolarizzazione).*

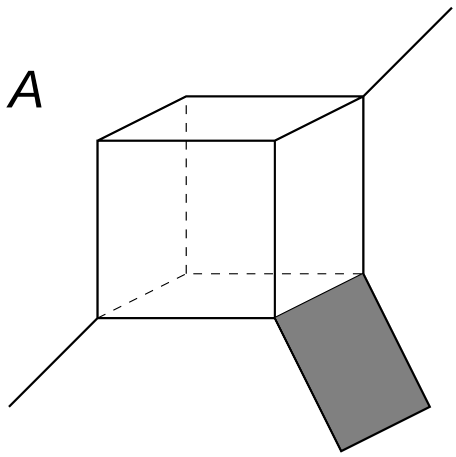


Figura 1.1: *Esempio di insieme non regolare: ci sono uno spigolo ed una faccia "penzolanti" (rispettivamente* dangling edge *e* dangling face*).*

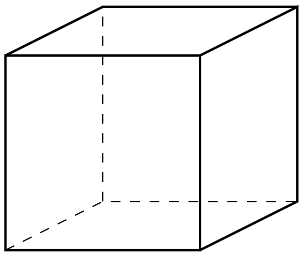


Figura 1.2: ch(i(A)) *(vedere figura 1.1).*

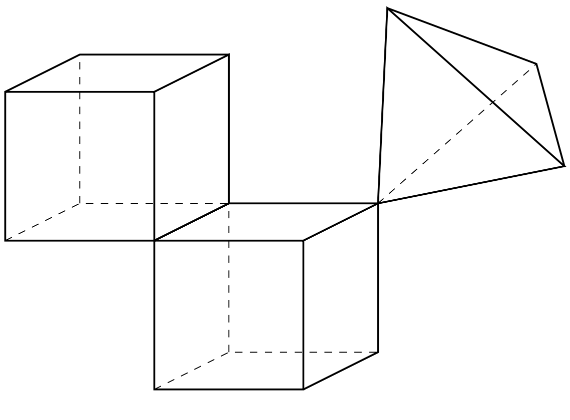


Figura 1.3: *Esempio di r-set.*

1.3.1 Superfici

Il contorno dei solidi può essere descritto da *superfici algebriche* (rappresentabili da funzioni delle variabili x, y, z).

Un solido di questo tipo non può essere modellato mediante funzioni matematiche "semplici" (ad es. polinomi).

1.3.2 Varietà bidimensionali (2-manifold)

Una superficie è un sottoinsieme di $\mathbf{R}^3$ bidimensionale: ogni suo punto è circondato da una regione bidimensionale di punti appartenenti alla superficie stessa. È necessario definire la nozione di "superficie" in modo astratto e, quindi, costruire per essa un semplice modello bidimensionale in termini di una speciale topologia definita nello spazio euclideo bidimensionale $\mathbf{R}^2$.

Definizione 15 *Una* varietà bidimensionale (o 2-manifold) *è uno spazio topologico in cui ogni punto ha un intorno topologicamente equivalente ad un disco aperto di* $\mathbf{R}^2$ *(vedere nota [1]).*

Sia A un r-set. Se un 2-manifold M ed il contorno di A, indicato con b(A), sono topologicamente equivalenti, allora A è una *realizzazione* di M in $\mathbf{R}^3$.

Nota. Non tutti gli r-set sono realizzazioni di qualche 2-manifold.

Ogni punto di un r-set che non ha un intorno topologicamente equivalente ad un 2-manifold è detto *punto non-manifold.*
Un modello matematico più restrittivo è costituito dagli r-set con contorno 2-manifold (tale modello sarà per esempio usato negli schemi di rappresentazione boundary). Nel seguito, ove non precisato, chiameremo "solido" indifferentemente un r-set o un set con contorno 2-manifold.

In figura 1.4 sono visibili alcuni esempi di r-set che non sono manifold bidimensionali, in quanto dotati di punti che non hanno un intorno topologicamente equivalente ad un disco aperto:

- il punto P;

- i punti del segmento P_1P_2

[1] Disco aperto di $\mathbf{R}^2 = \{(x, y) \in \mathbf{R}^2 | x^2 + y^2 < r^2, \text{con } r \in \mathbf{R}^+\}$

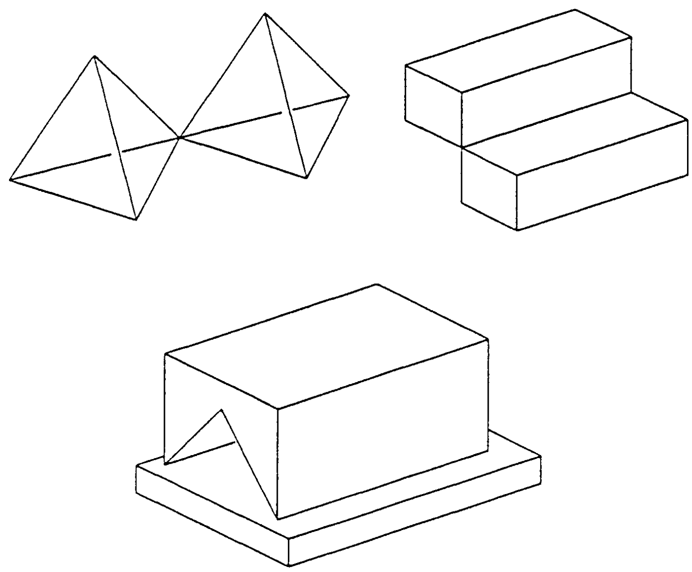


Figura 1.4: *Esempi di r-set che non sono manifold bidimensionali.*

- i punti dei due segmenti in cui il cuneo poggia sul parallelepipedo.

1.4 Schemi di rappresentazione

Siano

$$\mathcal{M} = \text{lo spazio dei modelli,}$$
$$\mathcal{R} = \text{lo spazio delle rappresentazioni.}$$

Una rappresentazione $r \in \mathcal{R}$ è una collezione finita di simboli appartenenti ad un alfabeto finito che codifica un solido $\in \mathcal{M}$.

Definizione 16 *Uno* schema di rappresentazione *è un'applicazione*

$$\sigma : \mathcal{M} \to \mathcal{R}$$

che associa ad un solido $s \in \mathcal{M}$ una rappresentazione $r \in \mathcal{R}$

Indichiamo con:

$$\mathcal{D} = \text{il dominio di } \sigma,$$
$$\mathcal{V} = \sigma(\mathcal{D}) \text{ il codominio di } \sigma.$$

Definizione 17 *Una rappresentazione r è detta* valida *$\Leftrightarrow r \in \mathcal{V}$*

Pertanto, ogni rappresentazione valida è l'immagine di almeno una entità di $\mathcal{M}$.

Esempio. Sia $\mathcal{M}$ l'insieme dei solidi con contorno 2-manifold e $\mathcal{D}$ l'insieme dei solidi con contorno 2-manifold a facce piane.

* * *

La figura 1.5 mostra graficamente uno schema di rappresentazione.

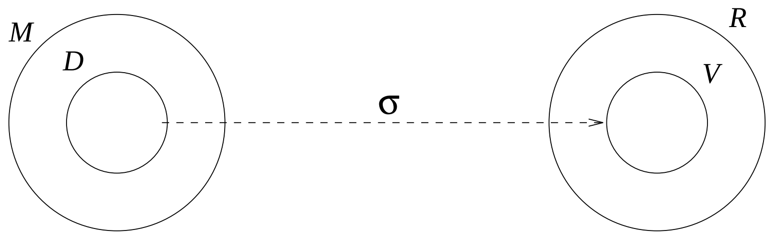


Figura 1.5: *Definizione di uno schema di rappresentazione.*

1.4.1 Proprietà

Enunciamo ora alcune proprietà che uno schema di rappresentazione *può* avere.

Proprietà formali

1. *Non ambiguità o completezza;* uno schema σ è detto *non ambiguo* o *completo* se ogni rappresentazione valida modella esattamente un solido (ovvero se σ^{-1} è una funzione); ne consegue che una rappresentazione valida individua un solo solido; viceversa uno stesso solido può avere più rappresentazioni valide.

2. *Unicità;* uno schema σ è detto *unico* se tutti i solidi hanno esattamente una rappresentazione (σ è una funzione); si noti che l'unicità richiede che un solido abbia una rappresentazione unica *indipendentemente* dalla sua posizione e dal suo orientamento nello spazio.

3. *Unicità debole;* uno schema σ è detto *unico debolmente* se ogni istanza orientata di un qualunque solido $s \in \mathcal{D}$ ha una sola rappresentazione.

Osservazioni:

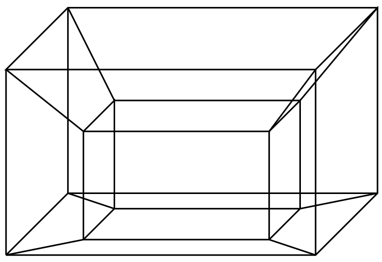
- Schemi non ambigui ed unici $\Rightarrow$ corrispondenza biunivoca fra solido e sua rappresentazione (σ è una funziona iniettiva);

- Situazione ammessa se σ è *completo* (ma non ammessa se σ è unico): σ^{-1} funzione;

- Situazione non ammessa se σ è *unico* (ma non ammessa se σ è completo): σ funzione.

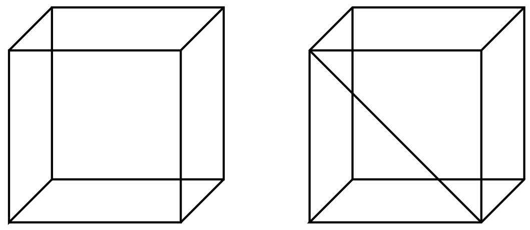
Esempi:

- Rappresentazione ambigua. Il *wireframe*, un solido a facce piane descritto dai suoi vertici e dai suoi spigoli:



Quali sono le facce?

- Rappresentazione non unica:



Proprietà non formali

1. *concisione:* dimensione delle strutture dati ridotta;
2. *chiusura delle operazioni;*
3. *facilità computazionale:* efficienza delle operazioni fondamentali e semplicità di scrittura degli algoritmi corrispondenti;
4. *applicabilità:* esistono svariati campi di applicabilità dello schema.

1.4.2 Classificazione

Esistono tre categorie di schemi di rappresentazione:

Schemi boundary: (o secondo il contorno) descrivono un solido in termini delle superfici che lo contornano (capitolo 6 del Mantyla).

Schemi scompositivi: descrivono un solido come una collezione di solidi primitivi molto semplici combinati mediante una operazione di unione disgiunta (capitolo 4 del Mantyla).

Schemi costruttivi: descrivono un solido come combinazione booleana (mediante operazioni di intersezione, unione e differenza) di solidi predefiniti o di semispazi (capitolo 5 del Mantyla).

Nota. Gli schemi costruttivi e scompositivi sono schemi *volumetrici* in quanto descrivono lo spazio tridimensionale occupato dal solido.

1.5 Schemi scompositivi

Schemi basati sull'oggetto (scomposizioni cellulari)
unione quasi-disgiunta di celle di ricoprono il volume del solido.

Schemi basati sullo spazio occupato (enumerazioni spaziali)
Solido descritto come unione di celle (cubiche) derivanti da una suddivisione dell'universo in cui il solido è immerso.

1.5.1 Scomposizioni cellulari (cell decompositions)

Una scomposizione cellulare è la rappresentazione di un solido per mezzo di celle quasi-disgiunte (che hanno in comune al più il contorno) di forma semplice. Un esempio è costituito dalla *tetraedralizzazione*, ovvero la scomposizione del solido in tetraedri aventi in comune al più una faccia, spigoli o vertici. Oppure, in due dimensioni, la *triangolazione* di un poligono (vedere figura 1.6). Una scomposizione

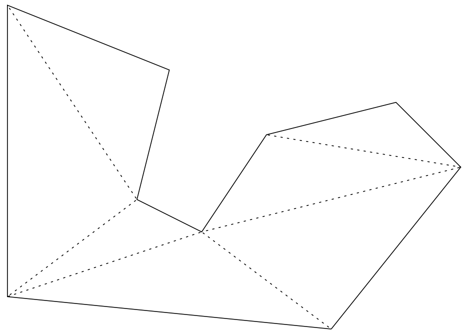


Figura 1.6: *Scomposizione cellulare di un poligono (triangolarizzazione).*

cellulare, essendo una rappresentazione che dipende dal solido, è invariante per trasformazioni rigide. Esaminiamo le proprietà di questa rappresentazione:

1. *validità:* difficile da stabilire (è necessario effettuare un test di intersezione per tutte le coppie di celle);
2. *non ambiguità:* una scomposizione cellulare valida descrive completamente un solido;
3. *unicità:* lo schema non è unico, in quanto possono esistere più triangolazioni di uno stesso poligono, o più tetraedralizzazioni di uno stesso solido;
4. necessita di algoritmi automatici per la costruzione della scomposizione cellulare (problema aperto nell'ambito della geometria computazionale).

1.5.2 Scomposizioni per enumerazione spaziale

Queste scomposizioni generalizzano gli schemi di rappresentazione di regioni piane basati sulla scomposizione di un universo quadrato in pixel. Si distinguono due schemi di enumerazione: di tipo classico e di tipo adattivo (schemi gerarchici).

Schemi enumerativi classici

Per schema enumerativo si intende una suddivisione di un universo cubico $\mathcal{U}$, contenente il solido, in cubi di dimensione ed orientazione uniforme (detti *voxel*), per mezzo di una suddivisione regolare $\mathcal{S}_R$ dello spazio.

Si rappresenta un solido S come una collezione di voxel $\in \mathcal{S}_R$ aventi intersezione non nulla con S. È una rappresentazione volumetrica regolare del solido (vedere figura 1.7 per un esempio nel piano).

Si possono definire (in modo analogo al caso bidimensionale) i concetti di:

Definizione 18 *6-adiacenza: due voxel v_1 e v_2 si dicono* 6-adiacenti *se e solo se hanno in comune una faccia.*

Definizione 19 *18-adiacenza: due voxel* v_1 *e* v_2 *si dicono* 18-adiacenti *se e solo se hanno in comune una faccia o uno spigolo.*

Definizione 20 *26-adiacenza: due voxel* v_1 *e* v_2 *si dicono* 26-adiacenti *se e solo se hanno in comune una faccia, uno spigolo o un vertice.*

Basate su queste definizioni, è possibile definire i concetti di *6-connessione*, *18-connessione* e *26-connessione* (si veda il capitolo 1 delle dispense di Modellazione Geometrica). Un solido (r-set) è rappresentato da un sottoinsieme massimale di voxel pieni 26-adiacenti (purché il solido goda della proprietà di connessione).

Esercizio. Come viene definita la rappresentazione a voxel per un solido connesso con contorno 2-manifold? (si veda il capitolo 1 delle dispense di Modellazione Geometrica).

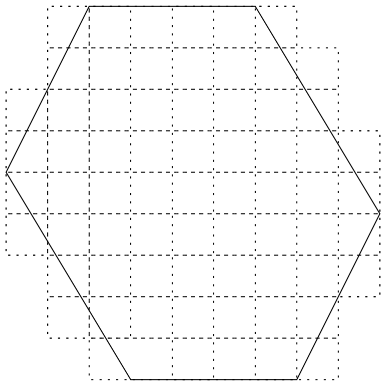


Figura 1.7: *Enumerazione di un poligono.*

La strutttura dati utilizzata per rappresentare un'enumerazione è una matrice C, $n \times n \times n$, in cui ciascun elemento c_{ijk} descrive un voxel nell'universo:

$$c_{ijk} = \begin{cases} 1 & \text{pieno} \\ 0 & \text{vuoto} \end{cases}$$

Questa struttura dati richiede n^3 bit di memoria, dove n^3 è il numero di voxel in cui è suddiviso l'universo.

Svantaggi:

1. è una rappresentazione approssimata del solido (con un errore di approssimazione proporzionale alla dimensione del voxel);
2. schema non conciso: richiede una notevole quantità di memoria;
3. è una rappresentazione dipendente dalla posizione dell'oggetto all'interno dell'universo.

Vantaggio: rende semplice il calcolo di intersezioni, unioni e differenze (che si riducono ad operazioni su matrici di bit).

Proprietà

Validità: verificabile mediante algoritmi di connessione, che verifichino la 26-connessione e le massimalità del sottoinsieme di voxel pieni.

Schemi enumerativi di tipo adattivo (Octree)

Gli Octree sono estensioni, per solidi, delle rappresentazioni gerarchiche basate su pixel per regioni bidimensionali, i Quadtree. L'idea di base è fondere voxel adiacenti dello stesso tipo (pieni o vuoti) in voxel di dimensione maggiore. La rappresentazione con Octree opera mediante una suddivisione ricorsiva dello spazio di interesse in otto *ottanti*, detti blocchi, organizzati in un albero in cui ogni nodo interno (corrispondente ad un blocco) ha esattamente *otto* figli.

I nodi interni corrispondono a blocchi che vengono ulteriormente suddivisi e sono detti "nodi parziali".

I nodi foglia possono essere di due tipi: *pieni*, se hanno intersezione non vuota con il solido, oppure *vuoti*, se hanno intersezione nulla con il solido.

Struttura dati. Si possono utilizzare due strutture dati per descrivere un Octree: un albero con puntatori oppure una struttura lineare. Nel primo caso, ogni nodo consiste di una variabile *status* e di otto puntatori agli otto figli, numerati da 0 a 7. Lo *status* può assumere tre valori: '*pieno*' se il nodo è pieno, '*vuoto*' se il nodo è vuoto e '*interno*' se il nodo è interno; essendo un alfabeto di tre caratteri, sono necessari 2 bit per definire lo *status* di ogni nodo. Un esempio di struttura lineare (valida anche per Quadtree, ed in generale per alberi) è la cosiddetta *espressione-DF (Depth-First)*, ottenuta visitando l'Octree in ordine pregresso (preorder).

Algoritmi sviluppati per gli Octree.

Operazioni binarie: considerano due Octree (definiti nello stesso spazio) e restituiscono un nuovo Octree come risultato delle operazioni booleane di unione, intersezione o differenza dei due argomenti;

Operazioni di analisi: calcolo del volume, della massa e dei momenti del solido rappresentato;

Operazioni geometriche: per effettuare rotazioni, traslazioni, etc. etc.

Vantaggi di un Octree rispetto ad uno schema di enumerazione classico.

1. minore occupazione di memoria (si può dimostrare che il numero di nodi necessari alla rappresentazione di un solido è proporzionale all'area della sua superficie, nell'Octree è invece indipendente dal solido);

2. maggiore precisione raggiungibile, grazie alla compattezza della rappresentazione.

Svantaggi di un Octree (comuni agli schemi enumerativi classici).

1. precisione limitata (alla dimensione dei voxel);

2. è un metodo dipendente dalla posizione del solido nell'universo di riferimento.

Proprietà di un Octree.

1. *validità*: per verificare la validità è necessario applicare anche in questo caso algoritmi per la determinazione di congruenti k-connessioni (ad esempio $k = 26$) che operano sulla struttura gerarchica;

2. *non ambiguità*: nei limiti imposti dalla risoluzione, tutte le rappresentazioni sono non ambigue;

3. *unicità*: non si ha una unicità posizionale in quanto la rappresentazione dipende dalla posizione e dell'orientamento del solido.

1.6 Schemi di rappresentazione di tipo costruttivo

Uno schema di rappresentazione di tipo costruttivo descrive un solido come combinazione booleana di solidi primitivi o di insiemi di punti.

1.6.1 Schemi basati sui semi-spazi

Un insieme $A \subset \mathbf{R}^3$ può essere descritto da una *funzione caratteristica* $g_A(X) : X \to \{0, 1\}$ tale che

$$g_A(X) = \begin{cases} 1 \text{ se } X \in A \\ 0 \text{ se } X \notin A \end{cases}$$

Per molti insiemi di punti la funzione caratteristica può essere rappresentata in termini di una funzione analitica $f(x, y, z)$ definita ovunque in $\mathbf{R}^3$. Il requisito di analiticità elimina alcuni oggetti "patologici" (vedere capitolo 3 del testo). Tutti i punti per cui $f(X) \geq 0$ sono considerati appartenenti all'insieme, mentre tutti i punti per cui $f(X) < 0$ appartenenti al suo complementare.

Poiché la superficie $f(X) = 0$ divide lo spazio in due sottoinsiemi, gli insiemi definiti da $f(X) \geq 0$ e $f(X) \leq 0$ rappresentano due *semispazi*. Ad esempio

$$ax + by + cz + d \geq 0$$

$$x^2 + y^2 - r^2 \leq 0$$

definiscono rispettivamente un *semispazio piano* ed un *semispazio cilindrico*.

Un solido è rappresentato come una combinazione booleana di semispazi.

Primitive per la modellazione basata su semispazi

La modellazione basata su semispazi consiste nell'applicazione di operazioni booleane (unione, intersezione e differenza) ai semispazi stessi (le primitive). Un cilindro di raggio r ed altezza finita h, ad esempio, può essere rappresentato come intersezione di un semispazio cilindrico con due semispazi planari:

$$\begin{aligned} H_1 :&\quad x^2 + y^2 - r^2 \leq 0 \\ H_2 :&\quad z \geq 0 \\ H_3 :&\quad z \leq h \\ \\ C =&\quad H_1 \cap H_2 \cap H_3 \end{aligned}$$

Rappresentazione interna

La rappresentazione più comune di un semispazio fa uso dei sottospazi quadratici, che possono essere rappresentati nella forma:

$$f(x, y, z) = (x\ y\ z\ 1) \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Un'espressione booleana di semispazi può essere realizzata come una somma di prodotti di semispazi,

$$S = \cup_i \cap_j H_{ij}$$

dove H_{ij} indica uno dei sottospazi di S.

Proprietà

1. *potere espressivo:* dipende dal tipo di semispazi utilizzati;
2. *validità:* i semispazi sono insiemi di punti infiniti. Una combinazione di semispazi può essere infinita ("solido" non limitato) e quindi non è detto che qualunque combinazione sia valida;
3. *non ambiguità:* ogni combinazione valida di semispazi determina un solido;
4. *unicità:* tali rappresentazioni sono non uniche;
5. *concisione:* rappresentazioni in genere concise (poche centinaia di semispazi consentono di modellare parti in modo adeguato);
6. *chiusura delle operazioni:* le operazioni booleane sono chiuse nel dominio dei semispazi.

1.6.2 Schemi basati su primitive limitate

È più facile trattare primitive limitate piuttosto che semispazi non limitati. Gli schemi basati su primitive limitate (o Geometria Solida Costruttiva, CSG) usano unicamente insiemi di punti limitati come primitive.

Rappresentazione di modelli CSG

Una rappresentazione CSG è descritta da un albero, detto *albero CSG*, in cui:

- le foglie descrivono le primitive solide;
- i nodi interni descrivono le operazioni tra le primitive (operazioni booleane, trasformazioni rigide).

Poiché le primitive sono sempre solidi limitati, una rappresentazione CSG è sempre un solido limitato.

Alcune operazioni tra primitive CSG possono non soddisfare i requisiti di "solidità" descritti in precedenza; ovvero, partendo da due insiemi regolari ed applicando ad essi una operazione booleana non è detto che il risultato sia un insieme regolare (vedere figura 1.8). Per questo motivo si sono introdotti i concetti di operatori booleani regolariazzati, indicati con $\cup^*, \cap^*, -^*$, che vengono definiti come

$$A \cup^* B = ch(i(A \cup B))$$
$$A \cap^* B = ch(i(A \cap B))$$
$$A \backslash^* B = ch(i(A \backslash B))$$

si può dimostrare che se le primitive CSG usate sono insiemi regolari limitati, le operazioni booleane regolarizzate sono algebricamente chiuse nella classe degli insiemi regolari e limitati.

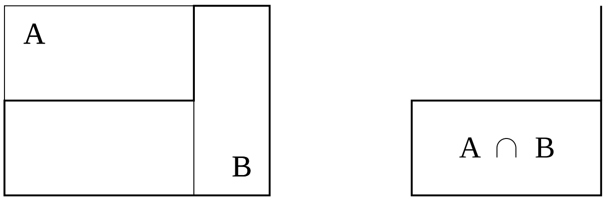


Figura 1.8: *Esempio di operazione tra set regolari che dà origine ad un set non regolare.*

Algoritmi che operano su modelli CSG

La struttura di una rappresentazione CSG suggerisce di utilizzare algoritmi di tipo "divide et impera" per effettuare operazioni su alberi CSG. Il particolare paradigma utilizzato per effettuare operazioni su alberi CSG è detto *set membership classification* (classificazione di appartenenza ad un insieme). Esso viene usato per:

- valutazione del contorno di un solido rappresentato da un albero CSG (ricostruzione della sua rappresentazione boundary);
- calcolo di proprietà integrali di un solido (volume, massa, momenti, etc.);
- generazione di immagini di un solido;
- query geometriche (ad esempio, localizzazione di un punto, intersezione con un segmento, intersezione con un piano).

Set membership classification

In generale un algoritmo di set membership classification lavora su due insiemi di punti: l'*insieme candidato* C (descritto dall'albero CSG) e l'*insieme di riferimento* R. L'algoritmo *classifica* C su R generando tre insiemi:

1. $CinR$: parte di C interna ad R;
2. $ConR$: parte di C sul contorno di R;
3. $CoutR$: parte di C esterna ad R.

La specifica del significato e delle rappresentazioni degli insiemi (C, R, $CinR$, $ConR$, $CoutR$) definisce un particolare algoritmo di "set membership classification".

Esempio. Classificazione di un segmento e nei confronti di un solido s descritto da un albero CSG. In questo caso:

- C: spigolo descritto da una coppia ordinata di terne di coordinate;
- R: insieme descritto da un albero CSG s;
- $CinR$, $ConR$, $CoutR$: $cinr$, $conr$, $coutr$, sottosegmenti di c tali che $cinr \cup conr \cup coutr = r$

Algoritmo:

```
CLASSIFY(s, e)
begin
    if s.node = primitive then
        return PRIMITIVE(s.node, e)
    else
        return COMBINE(CLASSIFY(s.left, e), CLASSIFY(s.right, e), s.node)
    endif
end
```