

Ron. 7/20-176

UNIVERSIDAD DE GRANADA



Departamento de Lenguajes y Sistemas Informáticos

**MODELADO DE SOLIDOS Y
ALGEBRA DE OBJETOS GRAFICOS**

TESIS DOCTORAL

Francisco Feito Higuieruela

Granada, 1995

FEITO HIGUERUELA, FCO RAMON

CURSO 94.95



Biblioteca Universitaria de Granada



01535007

UNIVERSIDAD DE GRANADA

22 DIC. 1994

COMISION DE DOCTORADO

T
15
148

UNIVERSIDAD DE GRANADA
Facultad de Ciencias
Fecha 13-1-95
ENTRADA NUM. 79

MODELADO DE SOLIDOS Y ALGEBRA DE OBJETOS GRAFICOS

Tesis Doctoral:

Dirigida por el

Dr. D. Juan Carlos Torres Cantero

presentada por

Francisco Feito Higuera

BIBLIOTECA UNIVERSITARIA
GRANADA
Nº Documento 619685701
Nº Copia 12123663X



Agradecimientos.

Quiero expresar mi agradecimiento a todas las personas que de un modo u otro me han ayudado y orientado durante la realización de este trabajo.

En particular a Buenaventura Clares y Miguel Vega por su ayuda y orientaciones.

A Ildefonso Castro y Antonio Garrido por sus continuos alientos.

Y muy especialmente al profesor Juan Carlos Torres, por su apoyo y ayuda constantes así como por su paciencia y entusiasmo, y al profesor Robert Juan por sus valiosas sugerencias.

A mi familia

CONTENIDOS

Introducción	1
Capítulo I. Modelado de sólidos. Conceptos básicos	5
1.1. Modelos	5
1.2. Modelado de sólidos	11
1.3. Sistemas de representación	16
1.3.1 Descomposición celular; octrees	16
1.3.2 B-rep: modelo de representación de fronteras	20
1.3.3 CSG: Geometría constructiva de sólidos	23
1.3.4 Barrido (sweeping)	26
1.4. Conversión entre diferentes representaciones	28
Capítulo II. Sistemas de generadores en un álgebra gráfica	35
2.1 Tipos de sólidos	36
2.2 Antecedentes	37
2.3 Operaciones regularizadas	45
2.4 Sistemas de generadores	52
2.5 Expresiones canónicas	64

Capítulo III. Algoritmos de inclusión en sólidos	77
3.1 Algoritmo de inclusión para sólidos bidimensionales	78
3.2 Algoritmo de inclusión en 3 dimensiones	86
Capítulo IV. Modelado de sólidos por recubrimientos simpliciales	95
4.1 Recubrimientos simpliciales en 2 dimensiones	96
4.2 Recubrimientos en tres dimensiones	103
4.3 Recubrimientos simpliciales canónicos. Introducción	114
4.3.1 Descomposiciones minimales	114
4.3.1.1 Descomposición canónica en dos dimensiones	115
4.3.1.2 Descomposición canónica en tres dimensiones	122
4.4 Operaciones entre sólidos	129
4.4.1 Complementación de sólidos	130
4.4.2 Diferencia, unión e intersección de sólidos	131
4.5 Subordinación en dos dimensiones	147
4.6 Subordinación en tres dimensiones	150
4.7 Conversión de B-rep a CSG	156
4.8 Conclusiones	158
Capítulo V. Conclusiones y principales aportaciones	161
Bibliografía	165

Figuras y tablas

Capítulo I

Figura 1.1	Objeto ambiguo	14
Figura 1.2	Representación B-rep	21
Figura 1.3	Representación CSG	24
Figura 1.4	Representación de barrido	27
Figura 1.5	Conversión de B-rep a CSG	32

Capítulo II

Figura 2.1	Operaciones entre objetos gráficos	43
Figura 2.2	Objetos a regularizar	48
Figura 2.3	Tetraedro orientado	55
Figura 2.4	Triángulos originales	56
Figura 2.5	Pirámides originales	57
Figura 2.6	Descomposición de un objeto en símlices originales	59
Figura 2.7	Conversión formal de B-rep a CSG	64

Capítulo III

Figura 3.1	Demostración teorema 3.1	79
Figura 3.2	Pertenencia de un punto a un triángulo original	82
Figura 3.3	Algoritmo de inclusión en dos dimensiones	85
Tabla I	Estudio de tiempos de algoritmos en dos dimensiones	85
Figura 3.4	Pertenencia de un punto a un tetraedro original	89
Figura 3.5	Algoritmo de inclusión en tres dimensiones	90
Figura 3.6	Sólidos tridimensionales	91
Tabla II	Estudio de tiempos de algoritmos en tres dimensiones	92
Tabla III	Implementación del algoritmo de inclusión en tres dimensiones	93

Capítulo IV

Figura 4.1	Triángulos subordinados	97
Figura 4.2	Descomposición de polígonos	99
Figura 4.3	Demostración Teorema 4.1	100
Figura 4.4	Demostración lema 4.1	105
Figura 4.5	Pirámides subordinadas	106
Figura 4.6	Descomposición de poliedros	107
Figura 4.7	Demostración Teorema 4.2	108
Figura 4.8	Descomposición de un poliedro	110
Figura 4.9	Descomposición de una pirámide original	113
Figura 4.10	Demostración lema 4.2	116
Figura 4.11	Sólido no separado	118
Figura 4.12	Obtención de un sólido por capas	121
Figura 4.13	Descomposición de sólidos con agujeros	128
Figura 4.14	Complementación de sólidos	131
Figura 4.15	Demostración lema 4.10	135
Figura 4.16	Intersección de sólidos	136
Figura 4.17	Intersección de sólidos	137
Figura 4.18	Posiciones relativas de tetraedros originales	152
Figura 4.19	Cálculo de subordinaciones	155

Introducción:

Palabras clave: Sistemas formales en informática gráfica. Modelado de sólidos y álgebra gráfica. Representaciones B-rep y CSG. Conversión de B-rep a CSG

Teoría de objetos gráficos de Fiume. Generalización de Torres. Sistema B-rep en el algebra de gráficos. Sistemas de generadores. Símplices. Representaciones canónicas. Operaciones entre representaciones canónicas. Algoritmos. Modelado de sólidos por recubrimientos simpliciales. Primera aproximación. Sólidos separados. Índice y capa. Operaciones entre sólidos representados por recubrimientos simpliciales.

El modelado de sólidos es un campo muy joven de investigación científica en el que se han producido importantes avances en los últimos años. Tanto por el interés teórico de sus contenidos como por la gran cantidad de aplicaciones se puede decir que es uno de los campos de la ciencia informática en los que más se ha avanzado en los últimos años. Cabe destacar los trabajos que fundamentaron muchos de los principios actuales, realizados por los miembros del Production Automated Project (PAP) de la Universidad de Rochester. Sus informes técnicos son los más referenciados en los trabajos de modelado de sólidos; una historia muy bien documentada puede encontrarse en el trabajo de Voelcker y Requicha ([VorRe93]).

Al igual que en otros campos de la Informática Gráfica se está realizando un gran esfuerzo para lograr una formalización de los conocimientos que permita un tratamiento sistemático global de los problemas a la vez que facilite su posterior implementación y uso. Cabe, en este sentido, destacar los trabajos de Duce ([Duce_89] y [Duce_91], Fiume ([Fiume89]) y Torres ([Torre92], [TorC193] y [TorC194]).

El Dr. Torres, en su teoría de objetos gráficos, generaliza el concepto de objeto gráfico introducido por Fiume, a la vez que logra dotar al conjunto de objetos gráficos de una estructura algebraica que permite una mejor especificación y por tanto facilita su posterior implementación. Así mismo, y dentro de la misma teoría, se formalizan

fácilmente algunos sistemas de representación como los octrees, barrido y Constructive Solid Geometry (CSG).

La representación de fronteras (B-rep) es uno de los sistemas de representación más utilizados y junto con el CSG, los más importantes y en los que se basan la mayoría de los sistemas comerciales actuales. En aquellos casos que es necesario, partimos de la representación "Star-edge" propuesta por Karasick ([Karas88]).

El objetivo de esta tesis es la formalización, en el marco de la teoría de objetos gráficos del Dr. Torres, de los sistemas de modelado de sólidos de poliedros de caras planas basados en la representación de su frontera, así como la regularización de las operaciones del álgebra y la obtención de sistemas de generadores que faciliten las operaciones con objetos gráficos.

Junto a la consecución de los objetivos anteriores y como fruto de las metas alcanzadas se presentan además algoritmos de inclusión en 2 y 3 dimensiones más eficientes y robustos que los existentes anteriormente así como la resolución del problema de la conversión de B-rep a CSG para poliedros de caras planas sin necesidad de añadir nuevos elementos (tal y como hacen Shapiro y Vossler [ShaVo92] en sus trabajos) y aportando una solución sencilla y eficiente, a la vez que se aumenta las aplicaciones prácticas respecto de otros métodos experimentales (como los presentados en [Dey__91], [PauRS93] y [ShaPe91]).

La tesis se estructura del siguiente modo: en el primer capítulo se hace una descripción básica de los métodos de modelado de sólidos así como de la conversión entre ellos.

En el capítulo dos se presenta un breve resumen de la teoría de objetos gráficos desarrollada por el Dr. Torres. En el marco de dicha teoría se definen las operaciones regularizadas entre objetos gráficos y se demuestra que el conjunto de objetos gráficos con las operaciones regularizadas mantienen las mismas propiedades que con respecto a las operaciones sin regularizar. Se definen a continuación un conjunto de elementos singulares (d -símplices originales), demostrando que forman un sistema de generadores para todos los elementos del álgebra. Puesto que dichos elementos se definen a partir de la frontera de un objeto gráfico, la demostración anterior equivale a la formalización

dentro del álgebra de cualquier objeto definido por su frontera. Posteriormente se definen la expresión canónica de un objeto gráfico y se demuestran un importante conjunto de propiedades para las operaciones del álgebra en relación a las expresiones canónicas. Como una consecuencia evidente se demuestra formalmente la equivalencia dentro del álgebra entre los dos métodos de representación más importantes, B-rep y CSG.

En el capítulo tres y como una consecuencia práctica de la formalización realizada, se enuncian y demuestran unos algoritmos nuevos de inclusión en 2 y 3 dimensiones. Así mismo se presentan los estudios de implementaciones realizadas que confirman la aseveración de mejorar en robustez y tiempo de respuesta en relación a otros ya existentes anteriormente.

En el capítulo cuatro se plantea una nueva expresión CSG válida tanto para sólidos variedad y no-variedad, basada en el desarrollo formal realizado en el capítulo dos: todo sólido se puede expresar como un recubrimiento jerárquico y minimal de símplexes. Primeramente se presenta un sistema sencillo aunque no minimal, que permite representar fácilmente cualquier sólido a partir de conjuntos muy sencillos (2-símplexes originales en 2 dimensiones y pirámides ó 3-símplexes originales en 3 dimensiones). La solución es un árbol binario que aporta para cada sólido una estructura jerárquica. Al objeto de conseguir sistemas minimales se define una relación de orden para sólidos separados (previamente se demuestra que de cualquier sólido se puede obtener un sólido separado) que da lugar a los conceptos de índice y capa. Estos conceptos permiten definir expresiones CSG minimales de modo que cualquier sólido no variedad se puede expresar mediante un árbol en el que cada primitiva aparece una sola vez. Se obtienen, posteriormente, el resultado de las operaciones booleanas aplicadas a sólidos representados por medio de recubrimientos simpliciales. Finalmente se aplican los resultados obtenidos para aportar un método de conversión de sólidos expresados mediante una representación B-rep a un modelo CSG.

Tanto el capítulo tres como el cuatro demuestran las ventajas que reporta la búsqueda y consecución de sistemas formales en el ámbito de la Informática, a la vez que resuelven teórica y prácticamente diversos problemas planteados en el ámbito del modelado de sólidos. Dichos métodos son aplicables a poliedros de caras planas y ,por medio de aproximaciones, a sólidos de forma libre.

CAPITULO I

Modelado de sólidos: Conceptos básicos

El uso de técnicas informáticas en el diseño y fabricación ha aumentado considerablemente en los últimos años. A esto han contribuido tanto el aumento de la capacidad del hardware como el desarrollo de algoritmos más eficientes o con mejor funcionalidad.

El modelado geométrico juega un papel importante tanto en el diseño y fabricación asistido por computadora como en otras aplicaciones gráficas avanzadas.

A lo largo de este capítulo se hace una breve introducción del concepto de modelo y de las características básicas del modelado geométrico, se introduce brevemente el modelado de superficies y se estudian los diversos métodos de modelado de sólidos, describiendo sus propiedades, características y fundamentos. Por último, se describen algunos algoritmos de conversión entre los principales métodos de modelado.

1.1 Modelos

Un modelo se puede definir como la representación conceptual de un fenómeno, susceptible de ser tratado matemáticamente. Ese fenómeno estará ligado a algunos aspectos de una entidad. El objetivo de los modelos es facilitar, y en algunos casos permitir, el estudio de esos aspectos, así como de sus modificaciones.

Es evidente que en los últimos años han aumentado el número de modelos susceptibles de tratamiento computacional. Nos vamos a restringir en este estudio a los modelos basados en computadora, y en concreto, a aquellos que son susceptibles de una

interpretación gráfica. La informática gráfica puede aquí servirnos tanto para la creación y edición de modelos, como para la representación y visualización de propiedades del modelo. Es clara la distinción entre el modelo y los medios gráficos usados para crearlo ó visualizarlo. Podríamos distinguir tres grandes tipos de modelos, dentro de nuestro campo de interés [Foley90]:

- Modelos organizativos: Corresponden a algún tipo de jerarquía o relación: diagramas de flujo, organigramas de empresa, clasificaciones, etc.
- Modelos cuantitativos: Vienen habitualmente representados por ecuaciones o colecciones de datos correspondientes a sistemas econométricos y financieros; sociológicos; de fenómenos físicos y químicos.
- Modelos geométricos: Implican la existencia de una colección de componentes con una geometría bien definida y a menudo con interacción entre ellas: estructuras de ingeniería y arquitectura; modelos moleculares; estructuras geográficas, etc.

El modelo asistido por computadora permite, de un modo más económico y muchas veces más eficaz, el estudio de determinados fenómenos. Incluso en bastantes casos no sería posible dicho estudio si no es por medio del modelo de la computadora. La bondad de los datos obtenidos dependerá siempre de la buena elección y diseño del modelo elegido.

Los modelos geométricos, en general, describen componentes con propiedades geométricas inherentes, por lo que es más evidente y necesario el uso de una representación gráfica. Varios son los elementos que deben estar presentes en un modelo geométrico:

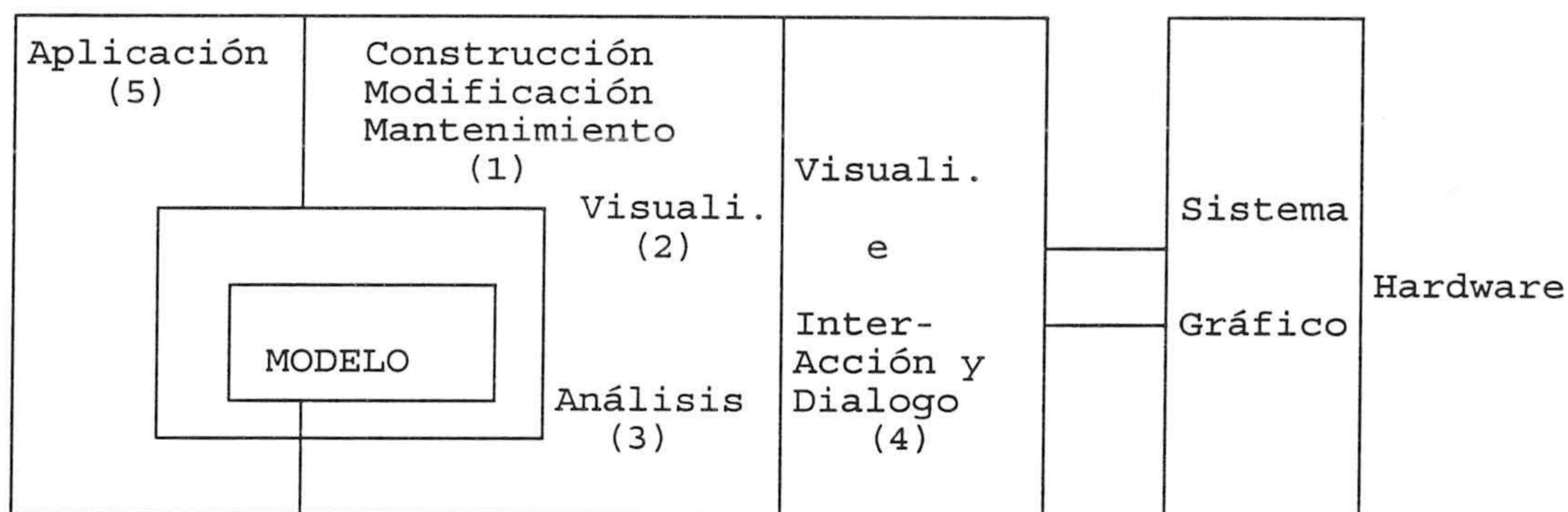
- Representación espacial y forma de los componentes y demás atributos que corresponden a la presencia de los componentes (color, textura, ..)

- Conectividad de los componentes (estructura topológica) que ha de indicarse si no viene determinada intrínsecamente por su geometría.
- Datos de la aplicación específica y propiedades asociadas a los componentes.

Asociados con el modelo se deben presentar los diversos algoritmos de proceso.

Habrà un constante movimiento entre los datos guardados explícitamente en el modelo y los que se obtienen para un determinado análisis o proceso (visualización, por ejemplo).

Un posible esquema de los elementos que intervienen en un modelo sería el siguiente:



Aunque hay muchos tipos de sistemas de modelado, en un nivel conceptual podemos distinguir ciertas funciones específicas del modelado geométrico que son comunes a todos ellos: representaciones, interfase de usuario (formada por la entrada del modelo y por la visualización del modelo) e interfase de aplicación ([BronPo88]).

-Representaciones. La base del sistema de modelado geométrico es la estructura de datos en la cual el modelo reside. A veces hay una sola, aunque lo normal es que haya varias representaciones, la primaria ó principal y las auxiliares. Existen procesos de mantenimiento de la consistencia entre ambas. La finalidad de representaciones múltiples es realizar cada operación con la representación más adecuada. Así se podrá visualizar a partir de una representación, y calcular el volumen a partir de otra.

Las representaciones pueden ser permanentes ó temporales. Las primeras permanecen almacenadas en la memoria auxiliar, pueden actualizarse en cualquier momento y a partir de ella puede recuperarse el modelo si es necesario. Las temporales desaparecen al finalizar una sesión de trabajo; pueden ser necesarias para algunas de las tareas del sistema.

Se pueden distinguir representaciones exactas y aproximadas.

-Entrada y edición de modelos. Dos son las técnicas habituales para la entrada de modelos: lenguajes de órdenes y técnicas gráficas interactivas. Un lenguaje de órdenes puede ser considerado como una representación para la entrada que posteriormente es convertido por el sistema de modelado en la representación principal. Puede ser guardado permanentemente y puede ser usado tanto en modo interactivo como en modo diferido.

El modelo de gráficos interactivos se basa en la utilización de un dispositivo apuntador y técnicas de posicionamiento y selección para construir el modelo editándolo sobre una visualización del mismo. Algunas técnicas, tales como selección de menús, se usan a menudo junto con la manipulación directa de objetos visualizados.

Junto a la especificación del modelo inicial deben existir herramientas de edición de modo que se permita al usuario la modificación y extensión interactiva de modelos existentes. A veces se conocen estas herramientas como operaciones de forma. Posteriormente las veremos con más detalle.

-Generación de visualizaciones. Es una parte integral del sistema de modelado debido a su importancia en un proceso de modelado interactivo, y la gran conexión existente con la función principal del sistema de modelado.

La visualización puede ser generada con diferentes niveles de calidad, desde el simple modelo de alambres hasta las complicadas imágenes sombreadas. Es evidente que el tiempo necesario para visualizar un modelo es directamente proporcional a la calidad. En la mayoría de los sistemas de modelado interactivos, el usuario puede elegir entre dos ó más niveles de calidad.

-Interfase de aplicación. La justificación de los sistemas de modelado geométrico radica en su potencia para soportar una gran variedad de aplicaciones. Así, el modelo debe ser accesible para la consulta y conversión ofreciendo un interfase de aplicación bien definido. Los programas de aplicación necesitan acceder a información contenida en el modelo; esto implica a veces que la aplicación se deba hacer a medida según la estructura de datos particular de un modelo determinado.

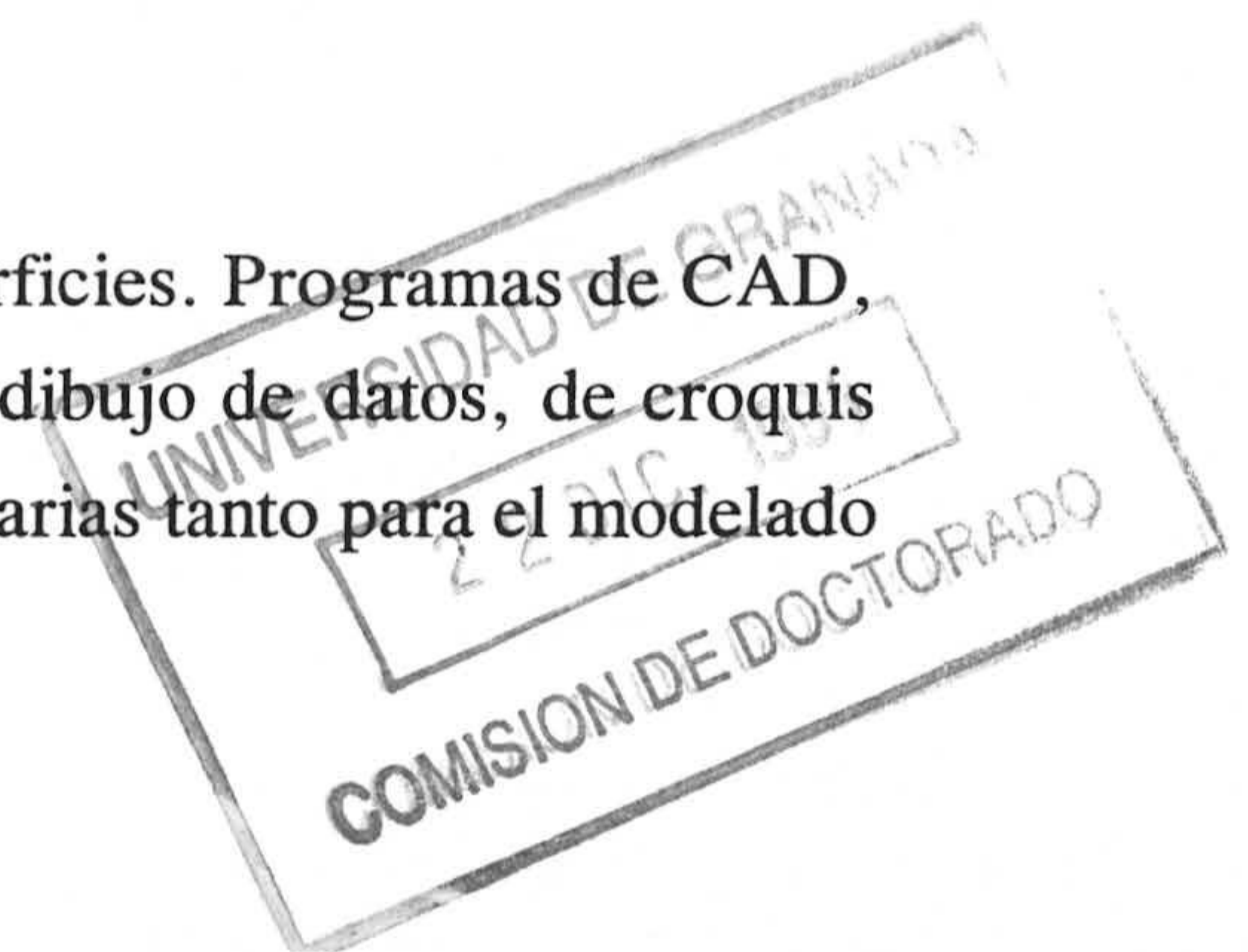
Para facilitar el encapsulamiento de datos-operaciones sería conveniente que existiese una representación auxiliar estandarizada que pudiera ser generada para este fin, lo que sería muy conveniente para gran número de aplicaciones. Se han desarrollado algunos estandares como IGES o XBF , aunque no son universalmente aceptados ni soportan todas las representaciones posibles para el modelado de sólidos y los métodos necesarios para representar superficies de forma libre. De ahí que todavía se deban continuar usando aplicaciones hechas a medida para un sistema particular. Este trabajo pretende avanzar en la búsqueda de representaciones estándar proponiendo un modelo formal en base al cual se pueden representar sólidos poliédricos mediante expresiones booleanas de símlices originales.

Dentro del modelado geométrico se pueden distinguir:

- Modelado de sólidos: El modelo corresponde a algún objeto rígido tridimensional real.
- Modelado de superficies: El modelo corresponde a una superficie compleja, denominada a veces superficie de forma libre o esculpida.

Es evidente que el primero hace uso del segundo, ya que en muchos casos un sólido rígido tendrá por frontera varias superficies esculpidas. A pesar de esto, es usual hacer la distinción antes indicada

Muchos de los objetos reales se aproximan por superficies. Programas de CAD, de generación de fuentes de caracteres de alta calidad, de dibujo de datos, de croquis artísticos, etc. contienen también curvas y superficies, necesarias tanto para el modelado



de objetos existentes como para el modelado de bocetos de objetos en los que no existe una representación previa.

Tres son las técnicas más generales de representación de superficies tridimensionales: redes de polígonos, superficies paramétricas y cuádricas. Puede encontrarse más información de estos temas en [Farin87], [Farin89] y [Ross91].

1.2 Modelado de sólidos

Se puede establecer un conjunto de requisitos que debe cumplir un subconjunto de \mathbb{R}^3 para poder servir de modelo de un objeto físico ([Requi80]). Estos requisitos son:

- Rigidez: La configuración y forma del sólido debe ser independiente de su localización y orientación.
- Homogeneidad tridimensional: Los sólidos deben tener un interior y su frontera no puede tener partes aisladas del resto del sólido.
- Finitud: Un sólido debe ocupar una porción finita del espacio.
- Cerrado bajo movimientos rígidos y bajo ciertas operaciones booleanas: al someter el sólido a un movimiento rígido (traslación, rotación) y al añadir o quitar material del sólido, se debe obtener otro sólido representable en el mismo modelo.
- Descripción finita: debe ser posible representarlo en la computadora.
- Frontera bien determinada: La frontera de un sólido debe determinar de forma no ambigua cual es el interior y por tanto que es lo que comprende el sólido.

Estos requisitos se resumen indicando que los modelos válidos de sólidos son subconjuntos de E^3 que son limitados, cerrados, regulares y semianalíticos:

- limitados porque ocupan una porción finita de espacio

-cerrados en el sentido topológico del término

-regulares en cuanto que su frontera no presenta partes aisladas

-semianalíticos en cuanto que su frontera está perfectamente determinada (no ocurriría así si, por ejemplo, parte de su frontera fuese la función $\text{sen}(\pi/x)$).

Los sólidos no tienen porque ser conexos y además pueden tener agujeros. Es claro que estos conjuntos no son cerrados bajo las operaciones booleanas de unión, intersección y complementación aunque sí bajo las denominadas operaciones regularizadas: una operación regularizada (unión, intersección ó complementación) se define como la clausura del interior del resultado de la operación usual. Se puede demostrar que bajo estas operaciones, el conjunto de sólidos regulares constituye un álgebra de Boole ([Requi78]).

Es evidente la existencia de muchos tipos posibles de representaciones a la hora de trabajar con sólidos. Vamos a estudiar a continuación las propiedades que estas deberían tener. Posteriormente enunciaremos diferentes tipos de representaciones existentes y señalaremos algunos algoritmos para pasar de unas a otras.

Una representación será sintácticamente correcta cuando esté formada con los símbolos de un alfabeto y de acuerdo con unas reglas sintácticas determinadas. La colección de todas las representaciones se denominará un espacio de representación R. Este puede ser visto como un lenguaje generado por alguna gramática. Sin embargo hay que observar que se admiten además de cadenas otro tipo de representaciones (grafos por ejemplo).

Se define la semántica de un espacio de representación, asociando entidades geométricas a cada representación posible. Así, se postula un espacio matemático M cuyos elementos son sólidos abstractos (conjuntos regulares) y se establece una correspondencia entre los elementos de M y los elementos de R.

Un sistema de representación se define formalmente como una relación [Requi80]

$$s: M \longrightarrow R$$

Denotamos el dominio de s por D y la imagen de D bajo s por V . Una representación será válida si es sintáctica y semánticamente correcta: es decir, pertenece a V . No se indica que toda representación sintácticamente correcta sea válida ni que todo sólido pueda tener su representación.

Una representación r en V será no ambigua o completa si corresponde a un único objeto. Será única si su antiimagen tiene un único elemento.

Un esquema de representación s es no ambiguo o completo si todas las representaciones válidas son no ambiguas; un sistema de representación es único si todas sus representaciones válidas son únicas. Si cumplen ambas propiedades, s equivale a una correspondencia biyectiva entre el dominio y la imagen.

Podríamos considerar las siguientes propiedades formales de un sistema de representación:

Dominio: El dominio es el conjunto de elementos representables en un sistema. El dominio de una sistema de representación nos da la potencia de descripción de dicho sistema.

Validez: El conjunto de todas las representaciones válidas será el rango del sistema. No es deseable dejar al usuario la tarea de comprobar la validez de una representación concreta. El sistema ha de ser capaz de comprobar la validez o no de cada representación.

No ambigüedad: Un sistema será no ambiguo o completo si cada representación define un único objeto.

Unicidad: Hay unicidad cuando cada objeto tiene una representación única. En la práctica no es fácil que se de esta propiedad por dos motivos: Por poder permutarse los elementos de la representación o por corresponder distintas representaciones a objetos iguales, colocados en diferente lugar.

Junto a las propiedades anteriores, sería deseable que un buen sistema de representación contase con otras propiedades denominadas "informales" ([Requi80]):

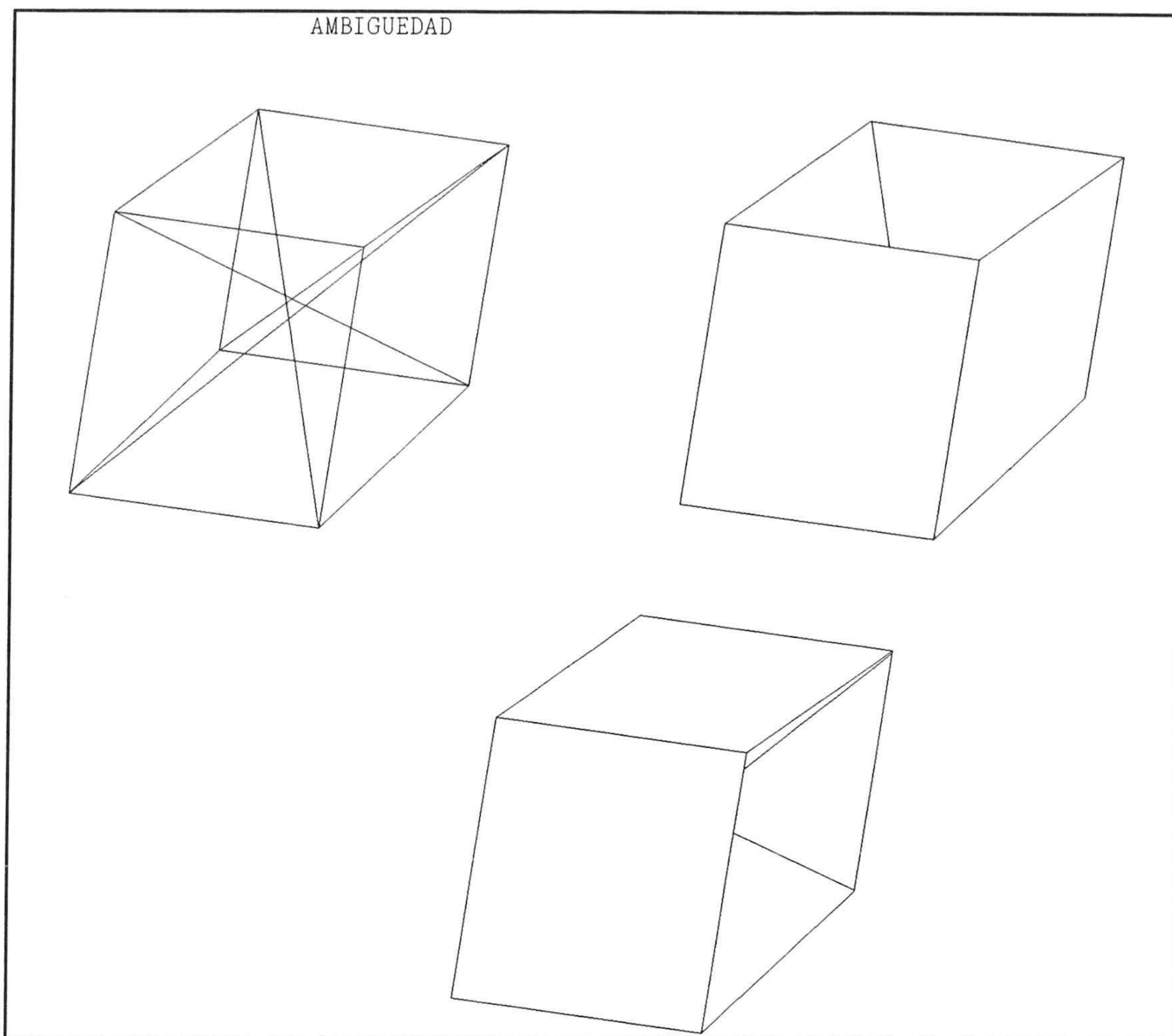


Figura I.1 Objeto ambiguo

Conciso: El tamaño de una representación ha de ser el menor posible, conteniendo pocos datos redundantes. A pesar de esto, una redundancia selectiva y controlada puede tener grandes ventajas en la práctica.

Fácil: En la mayoría de los casos el usuario será el hombre, y es evidente el deseo de que sea fácil crear representaciones. Si el sistema se basa en representaciones detalladas debe de tener un potente subsistema de entrada que facilite y ayude al usuario en la creación de representaciones.

Eficaz: Cada esquema es frecuente que sea usado en un campo concreto. Es deseable que en el campo concreto, el sistema sea eficaz. El diseño del esquema de representación debe permitir el uso de

buenos algoritmos para la evaluación de funciones útiles. Estos algoritmos deben ser correctos, extensibles, eficientes y robustos ante la posibilidad de errores numéricos.

Que un algoritmo sea correcto sólo puede asegurarse si se dispone de una especificación precisa del mismo. En el terreno de la geometría, una especificación de un algoritmo consiste en la definición de una función matemática (incluyendo su dominio y su rango o imagen) a ser evaluada por el algoritmo, junto con la definición de sistemas de representación para el dominio y el rango. La adecuada especificación de algoritmos geométricos no ha recibido la atención que merece (sí algo más a partir de 1985 con los estudios de geometría computacional, entre los que destaca la obra de Preparata y Shamos [PreSh85]). También ha sido facilitada con el desarrollo de la especificación formal de sistemas.

Los métodos clásicos de análisis y comparación de algoritmos ignoran los problemas de robusted y extensibilidad, y se aplican principalmente a algoritmos que operan para un dominio fijo y en un rango determinado de espacios de representación. El problema de comparación alternativa según el algoritmo y sistema de representación usado no es bien conocido, en parte por no existir un relación clara entre los diversos sistemas de representación.

Resumiendo, poco se conoce en sentido abstracto sobre la eficacia de un sistema de representación. Necesitamos no sólo significación formal para la caracterización de clases de funciones que pueden ser evaluadas eficientemente por algoritmos operando sobre representaciones en un esquema particular, sino también que sirvan para penetrar dentro de la interrelación entre esquemas y algoritmos.

No se puede decir que un sistema sea el mejor cuando se debe aplicar a muchos campos distintos, y en los que la redundancia de datos a menudo juega un papel importante en cuanto a la eficiencia.

La existencia de representaciones múltiples introduce problemas de redundancia en un sistema de modelado y provoca problemas de consistencia. Hay que asegurarse de que varias estructuras de símbolos que pretendan representar el mismo objeto u objetos en diferentes sistemas no comporten información contradictoria.

Dos representaciones r en R y r' en R' son consistentes si existe al menos un objeto m de M que tiene por representación r y r' . Varias pueden ser las representaciones para m en R y en R' (cuando los sistemas no poseen unicidad), y otros elementos de M pueden corresponder a r y r' (si el esquema es ambiguo).

La consistencia no es en general una relación de equivalencia entre las representaciones válidas de objetos. Podemos definir una relación de equivalencia del siguiente modo: dos representaciones r y r' son equivalentes si representan el mismo objeto. Esta relación puede ser extendida a los sistemas de representación del siguiente modo: dos sistemas de representación s y s' son equivalentes si cada representación r de s posee un equivalente r' de s' , y viceversa. Se sigue de la definición que esquemas equivalentes tienen que tener el mismo dominio y que dos sistemas no ambiguos de representación son equivalentes si y solo si tienen el mismo dominio.

La consistencia, (y no la equivalencia), es el concepto importante cuando se trabaja con representaciones ambiguas. Por ejemplo, es importante exigir que las representaciones de las vistas de alzado y planta de un objeto sean consistentes (ambas son representaciones ambiguas de sólidos). No sería razonable exigir que las vistas sean equivalentes: que todos los sólidos que presente un mismo alzado presenten también una misma planta.

La necesidad de forzar la consistencia tiene importantes implicaciones en el diseño de múltiples algoritmos de los sistemas de modelado. En [BroPo87], [Hoffm89], [JanBP90], [Kalay88], [Manty88], [Morte85] y [Requi92] puede profundizarse en los temas aquí expuestos.

1.3 Sistemas de representación

Entre los sistemas de representación más usados cabe destacar: descomposición celular, representación de fronteras (B-rep), geometría constructiva de sólidos (CSG) y barrido (sweeping). A continuación vamos a estudiar las características de estos sistemas de representación.

No hablaremos del sistema de alambres (Wire-frame) por no ser considerado este un modelador de sólidos. Dicho sistema se ha usado, y aún se usa, como esquema

de visualización incluso en tres dimensiones debido a la sencillez de los algoritmos. A pesar de ello, la ambigüedad de las representaciones hace que la geometría del objeto no se describa correctamente. Algunos otros métodos, como el de instanciación de primitivas y la parametrización, se describirá en la edición de modelos, y otros se consideran más operaciones de forma que representaciones completas propiamente dichas.

1.3.1 Descomposición celular; octrees.

A pesar de sus posibles variantes, lo que caracteriza a este sistema es la representación de un objeto mediante la enumeración de una colección de celdas que lo determinan.

En su variante más sencilla (enumeración espacial) el espacio en el que reside el objeto es dividido en un gran número de cubos de igual tamaño que se sitúan formando una malla tridimensional. Para cada cubo, denominado voxel, se determina y guarda su posición respecto al objeto: dentro o fuera; para cada voxel que parcialmente está fuera y parcialmente está dentro se decide si incluirlo fuera o dentro en función de algún criterio preestablecido (si hay más de la mitad del voxel fuera ó si el centro del voxel está situado dentro, etc.).

La estructura de datos usada en la enumeración espacial es muy simple. Consiste en una matriz tridimensional, con un elemento para cada voxel, en el que se indica si está dentro o fuera. Las ventajas de la enumeración espacial son:

- Simplicidad
- Facilidad para algunas aplicaciones, como por ejemplo, cálculo del volumen y de la masa de objetos

Las desventajas:

- la mayoría de los objetos, especialmente si tienen caras curvas, solo pueden ser aproximados por este método; esto nos da una representación con saltos y no exacta.
- Se puede aumentar la precisión disminuyendo el tamaño de los voxels, aunque esto supone un aumento en la cantidad de memoria usada.

Otro método dentro de los modelos de descomposición son los bintrees o árboles binarios. Se parte de un cubo inicial que se va dividiendo alternativamente en cada uno de los tres ejes, rotando cíclicamente en la elección de este eje, en cada subdivisión. Se van obteniendo paralelepípedos hasta un tamaño determinado. Una de las ventajas de este método es que es independiente de la dimensión del espacio, pudiéndose utilizar en el espacio de 4 dimensiones (espacio-tiempo). Es un modelo no ambiguo, único, válido y fácil de usar para el cálculo de volúmenes y para operaciones booleanas. Sin embargo, en caso de traslaciones y giros, el modelo debe recalcularse completamente.

Similar al método de bintrees es el método Binary Space Partition (BSP): partición binaria del espacio (BSP). La diferencia fundamental con los árboles binarios es que en estos se divide por la mitad al encontrar una celda heterogénea: en BSP se usa uno de los planos que contienen las caras del sólido. Las conclusiones de esto son:

- Los poliedros pueden ser modelados exactamente ya que las celdas no son exclusivamente paralelepípedos, sino que están asociadas al modelo. Por ello, también las transformaciones geométricas pueden ser fácilmente soportadas.
- Por ser las celdas cualquier tipo de poliedro convexo, los algoritmos asociados se complican respecto a otros tipos de modelos de descomposición.
- El modelo es no ambiguo, pero no único, ya que depende de las decisiones que se tomen en la generación del BSP.
- Las operaciones booleanas se ejecutan mediante la inserción de los planos del segundo BSP en el primero.

Una cuarta posibilidad es el método de los octrees. En este caso, los voxels no son de igual tamaño. Se parte de una caja que contenga todo el objeto - voxel inicial ó universal- y se divide en ocho cubos iguales (de aquí el término octree) y se etiquetan con tres posibilidades: blanco, si está totalmente fuera; negro si está totalmente dentro y gris si hay parte fuera y parte dentro. En el paso siguiente sólo los cubos grises se subdividen; con el resultado, se repite el proceso, y así sucesivamente hasta que se llega a un tamaño de celda predeterminado por el sistema. Con las últimas celdas grises se

decide si incluirlos dentro o fuera con algún criterio similar al indicado en la enumeración espacial.

La estructura de datos utilizada es un árbol; en cada nodo hay un registro que guarda el estado de la celda (blanco, negro ó gris) y si es gris hay punteros a los ocho hijos en que se subdivide. Una estructura de datos alternativa es el uso de octrees lineales. Aquí solo se guarda información de los voxels que están dentro del objeto: para todos estos cubos se considera un código en octal. La longitud del código depende del tamaño del cubo: cada subdivisión añade un número extra; este número depende de la posición del cubo: cada uno de los ocho cubos tiene un número fijo. El valor del código octal de un cubo determina unívocamente la posición del cubo en el espacio y su tamaño. El octree lineal consiste así en una secuencia de códigos octales para todos los cubos dentro del objeto. Esto requiere menos memoria que la estructura de datos árbol.

Las ventajas de los octrees son casi las mismas que las de la enumeración espacial, a pesar de que la estructura de datos es algo más complicada. Lo mismo ocurre con los inconvenientes (aunque en el caso de los octrees se necesita menos memoria).

Las operaciones booleanas son fáciles de realizar entre octrees: Para calcular la unión o intersección de dos octrees se recorren ambos árboles en paralelo, comparando cada par de nodos correspondientes. Por ejemplo en el caso de la unión, se comparan: si uno de los dos es negro, se añade un nodo negro al árbol resultado. Si uno de los nodos del par es blanco, se crea en el nuevo árbol el nodo correspondiente con el valor del otro nodo del par. Si ambos nodos son grises, se añade un nodo gris y se aplica el algoritmo a los pares de hijos. En este último caso hay que revisar los hijos antes de aplicar el algoritmo: si todos son negros, se borran y su padre se cambia de gris a negro. La intersección es similar, intercambiando los papeles del blanco y del negro.

Es fácil aplicar transformaciones simples: rotación alrededor de un eje de un ángulo múltiplo de 90° y escalado de potencias de 2. Las traslaciones y el resto de transformaciones son bastantes más complejas.

Uno de los problemas que presentan los octrees es que la representación del

objeto es siempre aproximada. Los octrees extendidos eliminan esto en parte, al menos en el caso de poliedros ([AyBRN85]). En ellos se considera la existencia de hasta cinco tipos de celdas: negra, blanca, nodo vértice (que contiene un vértice simple y sus caras y aristas conectadas), nodo arista (que contiene parte de una arista simple y sus caras) y nodo cara, que está cortada por un trozo de una cara. Simultáneamente, y de forma independiente, se propusieron los politrees ([CarCV85]). Las diferencias entre estos dos modelos son exclusivamente el modo de representar los nodos cara, arista y vértice. En los politrees se computa la geometría exacta de la superficie del objeto dentro del cubo. En los octrees se guarda solo la configuración y un conjunto de punteros a una única lista de semiespacios. La configuración es la información auxiliar necesaria para clasificar un punto del nodo como interior ó exterior al sólido.

Se puede demostrar que los octrees extendidos proporcionan un modelo exacto para la representación de poliedros ([NavAB86]): en este sentido, permiten el recálculo de la representación de fronteras del sólido.

El modelo de octrees extendidos puede ser usado como una representación secundaria en un Sistema de modelado de sólidos, junto con una representación primaria de fronteras (ver la explicaciones posteriores de este modelo de representación), en orden a reducir la complejidad de las operaciones booleanas.

El rendimiento de los octrees se puede resumir en :

- Son representaciones que contienen una descomposición jerárquica del volumen interno del objeto, que se suelen usar como representaciones secundarias.
- Son muy adecuados para las operaciones booleanas y para el cálculo de propiedades del volumen.
- El principal problema es la complejidad de las transformaciones geométricas.

Cabe destacar los trabajos [AyBRN85], [Ayala88], [CarCV85], [Meagh82], [BruAy87], [BruNa92], [Garga82] y [NavAB86] en todo lo referente a octrees, octrees extendidos y politrees. Así mismo los trabajos de [Samet90a] y [Samet90b].

1.3.2 B-rep: modelo de representación de fronteras

Este modelo se puede considerar una generalización del modelo de alambres (destinado fundamentalmente a los modelos de dibujo o las diversas técnicas de visualización).

Se basa en el Teorema de la curva de Jordan ([Kurat66]) y su generalización para superficies en el espacio. Una curva simple y cerrada en el plano descompone a este en dos zonas (una interior a la curva y otra exterior) que tienen a dicha curva por frontera común.

En el caso de poliedros de caras planas, la frontera estará formada por un número de caras, limitadas a su vez por un número de aristas, que a su vez están limitadas por vértices.

En una primera aproximación, la estructura de datos puede estar formada por registros, que contienen tanto la información geométrica como topológica. La información topológica se refiere a la relación entre las caras, aristas y vértices del objeto. Esta relación suele venir dada por punteros entre los diversos tipos de registros. Así una cara apuntaría a las diversas aristas que forman su frontera. Una arista apuntaría a los dos vértices que la determinan.

La información geométrica significa información relativa a las diversas ecuaciones de curvas y superficies, por una lado, y coordenadas por otro, que determinan las caras, aristas y vértices.

Si se permiten sólo ecuaciones de planos y cuádricas para las superficies, la mayoría de los objetos podrán ser representados exactamente por medio del modelo de fronteras. Sin embargo la mayoría de los sistemas de modelado permiten solo caras planas por lo que las curvas y superficies se aproximan por redes de polígonos. El motivo es que así la representación es más simple, y que todas las operaciones implicadas también lo son. Por ejemplo, la mayoría de los algoritmos para ocultación de superficies trabajan con caras planas. La desventaja de usar sólo aproximaciones es evidente ya que la aproximación nunca será exacta. También es necesaria más cantidad de memoria al aumentar el número de caras, aristas y vértices que componen la figura aproximada.

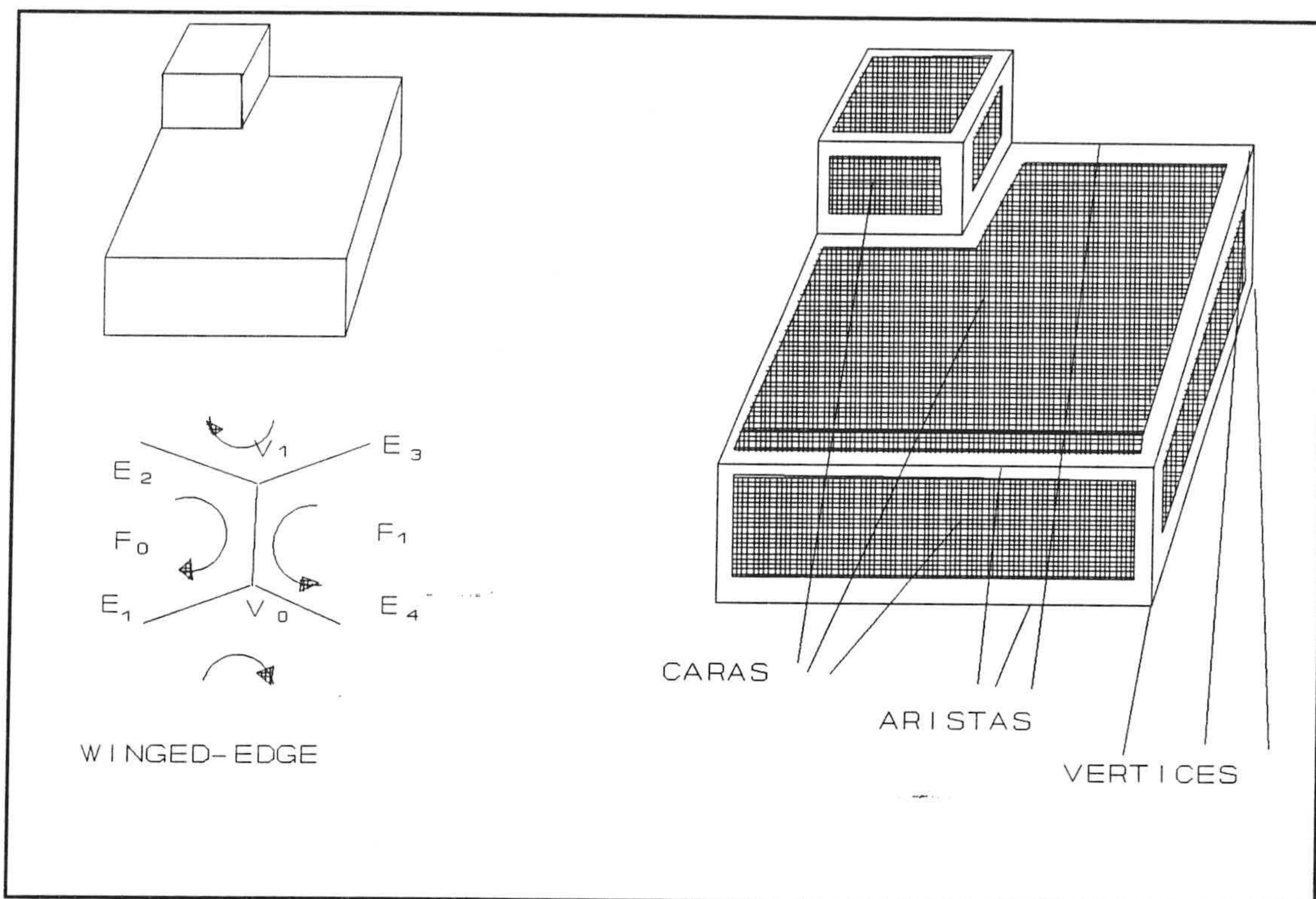


Figura I.2 Modelo B-rep

Existen muchas variantes del modelo de fronteras. Unas guardan más información que otras. Por ejemplo, si sólo se permiten caras planas, basta guardar sólo las coordenadas de los vértices, y determinar las ecuaciones de caras y aristas a partir de ellas y de la información topológica. Se puede, sin embargo, guardar en memoria las ecuaciones de aristas y caras, lo que ahorra tiempo de computación en muchas aplicaciones aunque requiera más memoria.

Hay también variación en cuanto a la información topológica que se almacena. La relación puede ser la indicada al principio, o contener mucha más información. El motivo es que en el primer caso si, por ejemplo, queremos hallar las caras que intersectan una arista concreta, habría que recorrer todas las caras.

En [Baumg75], [Kalay88], [Karas88], [Manty88] y [Vanec89] se describen diversos esquemas de desarrollo de B-rep.

Una variante a la representación anterior es denominada "winged-edge" la cual es un ejemplo de estructura de datos basada en aristas, que guarda mucha más información topológica. Consiste en una lista de todas las caras, de todas las aristas y

de todos los vértices.

Cada arista tiene asignada una dirección y de ella hay punteros a:

- Los dos vértices que la limitan
- Las dos caras que la limitan, denominada Cderecha y Cizquierda, donde derecha e izquierda son relativas a la arista vista desde fuera del objeto, y en el orden prefijado.
- Las cuatro aristas que la limitan

Para cada cara se guarda sólo una de las aristas que conectan con ella, e igual para cada vértice

Esta información topológica nos da acceso directo a alguna información, como las dos caras que intersectan una arista. Hay aún información que debe ser obtenida indirectamente, pero sin búsquedas exhaustivas. En general, se puede decir que la estructura de datos de la representación "winged-edge" es adecuada para el tipo de preguntas que se hacen a la representación de fronteras.

Posteriormente se han definido otro tipo de modelos de fronteras, como las introducidas por Karasick [Karas88], Mantyla [Manty88], Vanecek [Vanec89], Weiler [Weile88] y Yamaguchi [YamTo84].

Las ventajas de la representación de fronteras son:

- la información acerca de caras, aristas y vértices está explícitamente presente en la representación, lo cual es útil en muchos casos tales como en los métodos de visualización
- los modelos pueden ser construidos paso a paso y son fáciles las modificaciones locales.

Las desventajas son :

- se requiere gran cantidad de memoria
- la estructura de datos es compleja, y el usuario, o mejor el sistema de modelado, ha de tener gran cuidado para que la representación no cree objetos inválidos.

- Es difícil la entrada de representaciones a mano.

En lo que hace referencia al modelo B-rep en los capítulos siguientes, en los que se supone que contamos con una representación B-rep de un sólido utilizaremos la representación de Karasick, que es aplicable a sólidos variedad y no variedad.

La representación de fronteras Star-Edge, de Karasick, se basa en el concepto de arista orientada: una arista orientada es un par orientado cuyas componentes son una arista y una cara que son coincidentes. Cada arista orientada tiene tres componentes: una arista dada por sus vértices V_1 y V_2 , una cara que contiene a esa arista y un bit de orientación que nos indica si al recorrer la arista de V_1 a V_2 , la cara queda a la derecha ó a la izquierda de la arista. Si la arista tiene orientación derecha, V_1 se denomina primer vértice y V_2 segundo; si tiene orientación izquierda, al revés.

A partir de esos tres datos se puede calcular otra información importante como es el vector tangente a la arista (que es el dirigido del primer al segundo vértice), así como el vector dirección de la cara en la arista orientada que se define como el vector unitario que es el producto vectorial del vector tangente a la arista con el vector normal a la cara.

Es evidente que cada arista pueden determinar varias aristas orientadas.

1.3.3 CSG: Geometría constructiva de sólidos

Con la geometría constructiva de sólidos (CSG) el usuario tiene a su disposición un número de primitivas de objetos como cubos, esferas, cilindros, etc. Estos pueden ser posicionados en el espacio, a diversas escalas, y combinados por medio de las operaciones booleanas de unión, intersección y diferencia, para obtener objetos más complejos.

Los objetos simples pueden ser representados por la intersección de un número pequeño de semiespacios. Por ejemplo, un cubo unidad puede ser representado por la intersección de los siguiente semiespacios:

$$x \geq -1, x \leq 1, y \geq -1, y \leq 1, z \geq -1, z \leq 1.$$

El semiespacio $x^2 + y^2 + z^2 \leq 1$, representa una esfera unidad.

Los modeladores basados en CSG no ofrecen semiespacios a los usuarios, sino que ofrecen un conjunto restringido de objetos simples, predefinidos internamente con semiespacios. La ventaja de esto es que no se pueden construir inadvertidamente objetos no limitados. En algunos modeladores, las primitivas se restringen a cajas y cilindros, aunque la mayoría admiten el uso de esferas, conos, toros y otras.

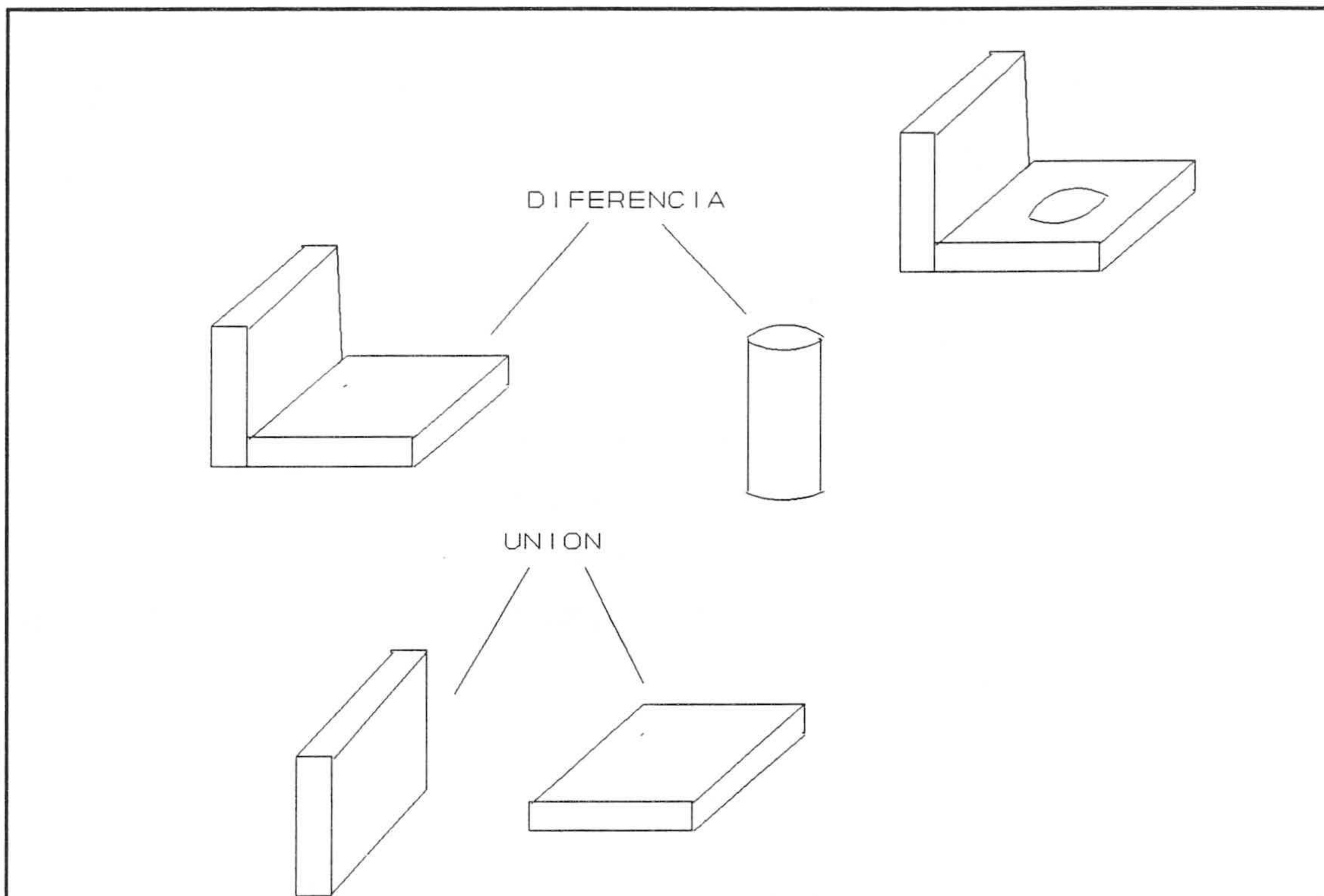


Figura I.3. Modelo CSG

Para combinar primitivas y obtener así objetos más complejos se dan tres pasos:

1. El usuario selecciona la primitiva, lo cual es posible asignándole un código a cada una de las primitivas. Además, cada primitiva tiene un número de parámetros que deben ser consignados. Por ejemplo, en un bloque, los parámetros son el largo, ancho y alto; para una esfera, el radio; etc.
2. Todas las primitivas son inicialmente situadas en una posición estándar y con una orientación estándar, Para mover la primitiva a su posición

correcta y orientarla correctamente, se dispone de una serie de transformaciones:

- transformaciones para moverlo en las direcciones X, Y y Z.
- rotaciones para girarlo alrededor de los ejes X, Y y Z.

Además, suelen existir:

- escalado en las direcciones X, Y y Z (equivale a cambiar los parámetros de la primitiva)
- cizallados para cambiar los ángulos con respecto a los ejes X, Y y Z.

3. Para combinar las primitivas y obtener objetos más complejos, tenemos los operadores unión, intersección y diferencia. La unión de dos objetos es el objeto formado por todas las partes pertenecientes tanto a A como a B; la intersección son las partes pertenecientes a la vez a ambos, A y B; la diferencia A-B son las partes de A que no pertenecen a B (es claro que la diferencia no es conmutativa: A-B no es lo mismo que B-A). Las operaciones a ser aplicadas a las primitivas se especifican por expresiones CSG, que consisten en las primitivas como operandos, y las operaciones como operadores

Junto al método anterior, basado en una forma de interacción vía menú, está también muy extendido el método basado en un lenguaje de órdenes: por medio de un fichero de texto se introducen las órdenes que se desean ejecutar. Por ejemplo:

```

1   cilindro 10 30
2   esfera  10
3   esfera  10
    mover z 30
4   cilindro 8 30
5   esfera  8
6   esfera  8
    mover z 30
operar (1+2+3)-(4+5+6)

```

En [Requi80], [Morte85] y [Laidl86] se puede profundizar en este tema.

La estructura de datos usada en la representación CSG consiste en un árbol binario, que es denominado árbol CSG. Todos los nodos del árbol son registros; en las hojas hay información sobre las primitivas; en los nodos internos, hay información sobre el tipo de operación a ser aplicada a los objetos definidos por los hijos, y punteros a esos hijos.

Las ventajas de CSG son

- todos los objetos que pueden ser modelados son válidos
- la representación es compacta.
- es fácil definir modelos; se ve que con solo cinco primitivas se pueden construir objetos muy complejos, en concreto la mayoría de las piezas mecánicas.

Las desventajas son:

- No hay información explícita sobre la aristas y vértices del objeto y por tanto no hay representación de la frontera; esto significa que no es muy adecuado para aplicaciones en las que se necesiten los elementos de la frontera.

1.3.4 Barrido (sweeping)

Con el sweeping (barrido) un objeto es definido moviendo un contorno cerrado bidimensional a lo largo de una trayectoria tridimensional. El contorno define la sección transversal del objeto. El objeto puede ser definido por medio del contorno y la trayectoria.

Se pueden distinguir tres tipos distintos de barrido:

- barrido translacional, prismático o radial (beam): el contorno es arbitrario, pero la trayectoria es una línea recta.

- barrido rotacional o superficie de revolución: el contorno es arbitrario; la trayectoria es un arco de circunferencia, es decir, el contorno se rota sobre un eje.
- barrido general o cilindros generalizados: tanto el contorno como la trayectoria son arbitrarios.

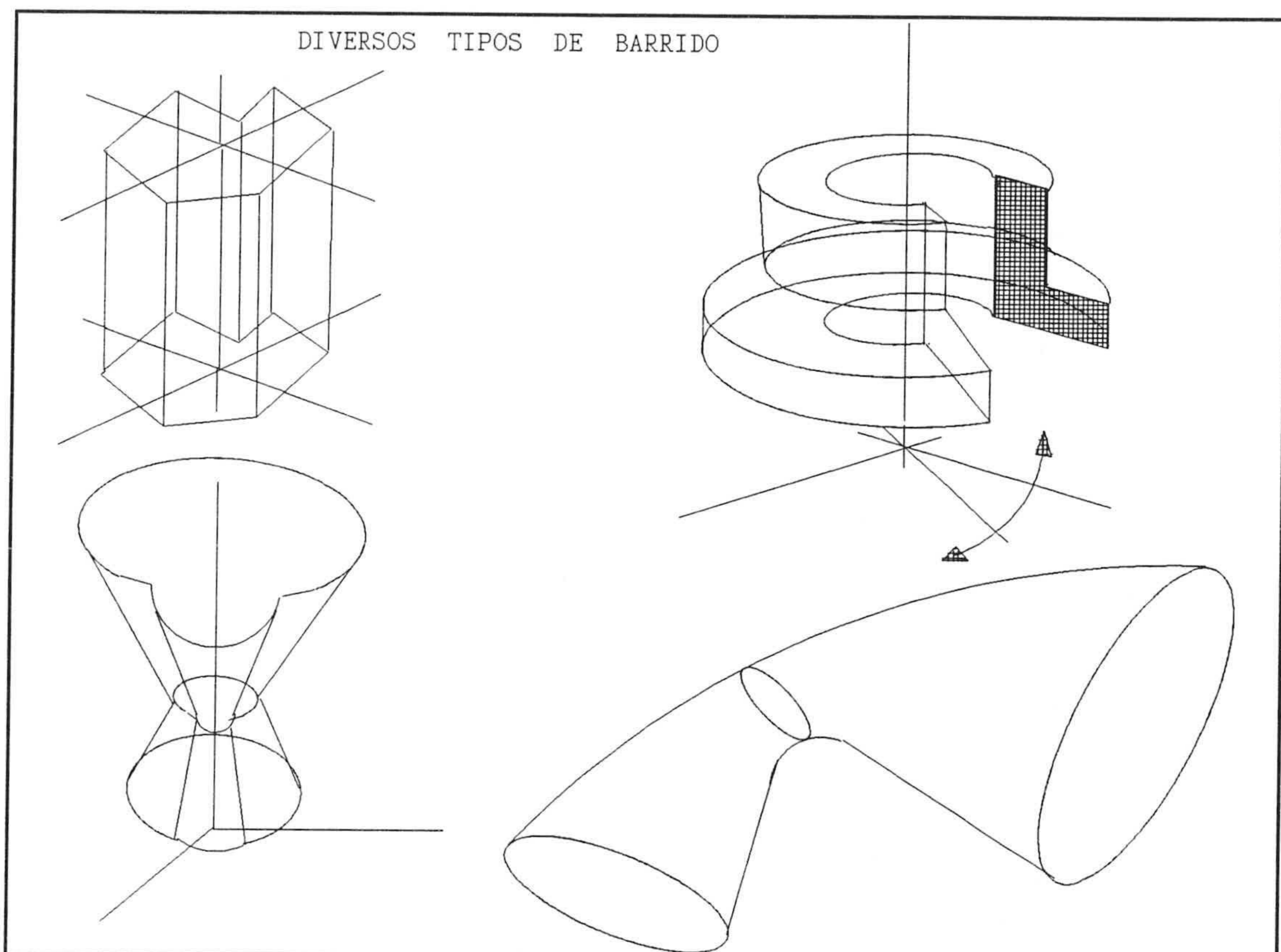


Figura I.4 Modelo de barrido

Los contornos y trayectorias arbitrarias consisten en una secuencia continua de curvas simples (líneas rectas, arcos de círculo y otras), pero también pueden ser curvas generalizadas (por ejemplo curvas de Bezier y B-splines). La estructura de datos para estas curvas consiste en la secuencia de las coordenadas de los puntos de control que determinan la curva.

En [Brons90] se puede encontrar una descripción de los diversos sistemas de barrido.

Una extensión útil de las formas básicas de barrido es permitir que el tamaño y la orientación del contorno varíe al movernos sobre la trayectoria. De este modo se pueden obtener objetos con estrechamientos y torceduras (twist). La especificación puede ser hecha dando factores de escalado y rotación en un número de posiciones a lo largo de la trayectoria.

Las ventajas del barrido son:

- es muy adecuado para la entrada de modelos: es relativamente simple especificar un contorno y una trayectoria.
- la representación es compacta.

Las desventajas son:

- El dominio es restringido, ya que sólo algunos objetos pueden ser modelados por barrido; por ejemplo, con barrido rotacional sólo los objetos que poseen simetría rotacional pueden ser modelados.

En la práctica, el método de barrido no se considera un esquema de representación, sino un método para la generación de primitivas ó para la introducción de modelos.

1.4 Conversión entre diferentes representaciones

Hemos visto que cada sistema de representación tiene ventajas e inconvenientes. Por ejemplo, con CSG es fácil introducir modelos pero más complicado su visualización. Por otro lado, la representación de fronteras es muy adecuada en orden a la visualización, pero requiere más cantidad de memoria. En general se puede decir que hay diferentes requerimientos para los esquemas de representación según: entrada, visualización, archivo permanente y aplicaciones.

Muchos sistemas de modelado proporcionan varios sistemas de representación. Por ejemplo, la entrada se hace vía CSG, la cual es guardada en una base de datos. Si es requerido un dibujo la representación se convierte a fronteras. La conversión puede

ser exacta (produce una representación exacta del objeto) ó aproximada (lo que hay es una aproximación del objeto inicial), por ejemplo usando solo celdas cúbicas.

No todas las conversiones entre representaciones son posibles. En principio es imposible pasar de CSG a barrido, ó de una representación aproximada a una exacta. En otros casos está abierto el problema de obtener los algoritmos adecuados para una conversión que en principio es posible.

Vamos a indicar brevemente métodos en los que se basan algunos algoritmos de conversión.

-De CSG a enumeración espacial ([BroPo88]):

Los métodos de conversión se basan en determinar para cada voxel de la rejilla si la intersección con el objeto es no nula. Para un objeto compuesto se aplica recursivamente el operador al árbol, entre los nodos correspondientes; por ejemplo si un voxel está dentro de uno de los hijos y el operador es la unión, estará en el nodo.

Para una primitiva, se hace por sustitución del centro del voxel en las ecuaciones de los semiespacios que definen la primitiva

-De barrido translacional (barrido simple) a B-rep ([BroPo88]):

Se generan caras perpendicular a la trayectoria limitada por el contorno en ambos extremos de la trayectoria, y se generan caras paralelas a la trayectoria limitada por partes del contorno y aristas paralelas a la trayectoria.

-De descomposición celular a octrees ([Samet90]):

Se recorren en postorden la matriz 3D y los nodos terminales se generan cuando todos los voxel correspondientes son de la misma clase. Al ascender en el árbol, si todos los hermanos son idénticos se eliminan y al nodo padre se le asigna su valor (blanco ó negro). Los nodos grises se generan cuando los nodos hijo no son idénticos.

-De CSG a octrees extendidos ([NavBF87]):

En general, la conversión de CSG a otras representaciones puede ser hecha de modo incremental ó no incremental. En el primer caso, se trabaja de cada hoja hacia la raíz, convirtiendo cada primitiva a la representación adecuada - por ejemplo, octrees- y aplicando la operación booleana indicada en cada caso; los métodos no incrementales obtienen la representación final directamente del árbol CSG. En este último caso, se trabaja restringiendo el árbol CSG a cada cubo obtenido en el proceso de subdivisión y se recorta el árbol considerando semiespacios ó primitivas con valor constante en el cubo. Cuando el árbol es aceptable como nodo octree terminal, se escribe en el octree de salida -computando previamente el tipo de nodo-; si no es aceptable se subdivide y se repite el proceso de modo recursivo.

-De B-rep a octrees extendidos:

Un posible algoritmo se basa en la división recursiva del espacio y la inclusión de la información de intersección obtenida dentro de los cubos. Para los octrees extendidos, el procedimiento se basa en un algoritmo recursivo: para el nodo inicial, se construye una lista de caras que lo atraviesan ó que están dentro. Se calcula la configuración de todos los vértices de la representación de fronteras y se genera una segunda lista con los vértices contenidos en el nodo. En general, la lista contendrá todas las caras y vértices del objeto. Al dividirse el nodo en ocho hijos, se generan las listas con las caras y vértices contenidos en cada octante. Se continua la división hasta que se obtienen los nodos terminales. Para ver si un nodo es terminal basta comprobar si su lista asociada está vacía.

-De octrees extendidos a B-rep ([BruNa85]):

Al objeto de obtener la representación b-rep, es necesario calcular la información geométrica y topológica. La información geométrica es inmediata ya que o bien está contenida en el octree o se puede obtener inmediatamente: las ecuaciones de los planos de las caras del objeto se obtienen directamente de los nodos extendidos del octree; las coordenadas de los vértices se obtienen por intersección de los planos asociados con los nodos tipo vértice.

Para la información topológica es necesario asociar los vértices con los polígonos asociados a las caras y puede obtenerse del siguiente modo: Se recorre el octree teniendo sólo en cuenta los no vértices. Para cada cara del poliedro se genera una lista: cada vértice se añade a las listas de las caras que convergen en él; estos elementos contienen información sobre las dos caras contiguas de acuerdo al orden cíclico alrededor del vértice. La información contenida en cada lista se ordena, conectando punteros a las caras del entorno, en orden a detectar polígonos cerrados.

-De octrees extendidos a CSG ([Juan_89]):

No es propiamente una conversión sino una simplificación. Se trata de expresar el árbol octree en función de un árbol CSG en la que los elementos van a ser los semiespacios que definen las caras de los octrees extendidos.

Cada hoja del octree se puede expresar como intersección de los semiespacios que definen cada cara ó de su complementario. Posteriormente se simplifica lo más posible la expresión obtenida.

-De B-rep a CSG:

Es un problema que permanece abierto en el caso general, en 3 dimensiones. En el caso de poliedros en 2 dimensiones se asigna a cada arista un semiespacio. En el caso de concavidades se añaden más operaciones.

En 3D se han aportado algoritmos para algunas clases de sólidos, en particular para los sólidos RH descomponibles ([Juan_88]). Un sólido rectangular (caras paralelas a los planos de coordenadas) es RH descomponible si todas sus caras pertenecen al cierre convexo rectangular (mínimo paralelepípedo rectángulo que lo contiene) ó puede dividirse en dos ó más conjuntos RH descomponibles. En este caso, para cada sólido RH descomponible se puede expresar en función de los subespacios que determinan sus caras, por lo que el sólido global también se puede expresar como un árbol CSG.

Así mismo se ha desarrollado una teoría en torno a la separación de objetos por medio de subespacios que ha permitido implementar de forma experimental la

conversión de objetos definidos mediante cuádricas en forma B-rep a representaciones eficientes en CSG ([ShaVo92]).

De CSG a b-rep

Vamos a estudiarlo con más detenimiento debido a su gran importancia y por ser estos dos modelos los que en la práctica están más extendidos. Este algoritmo se denomina Evaluación de fronteras.

La mayoría de los algoritmos trabajan sólo con primitivas de caras planas, por lo que las primitivas con caras curvas se han de aproximar. Los algoritmos que trabajan con superficies curvas son muy difíciles de implementar, debido a que es necesario determinar la intersección de superficies, lo cual es muy costoso para superficies complejas.

Uno de los algoritmos más elegantes para dos primitivas es el siguiente (si hay más de dos primitivas, la evaluación se hace paso a paso: primero dos y el resultado con la siguiente). La entrada del algoritmo consiste de un número de primitivas (primitivas B-rep) y el árbol CSG que indica como son combinadas. La salida es la representación en fronteras del objeto combinado.

El algoritmo se divide en dos pasos:

- determinar las líneas de intersección de las primitivas de entrada.
- determinar el objeto resultante combinando las partes relevantes de las primitivas de entrada.

La determinación de las líneas de intersección de las primitivas puede hacerse como sigue:

- determinar los puntos de intersección de todas las aristas de una primitiva con todas las caras de la otra y viceversa.
- determinar para cada cara de las primitivas, comenzando desde los puntos de intersección, cadenas de líneas de intersección.

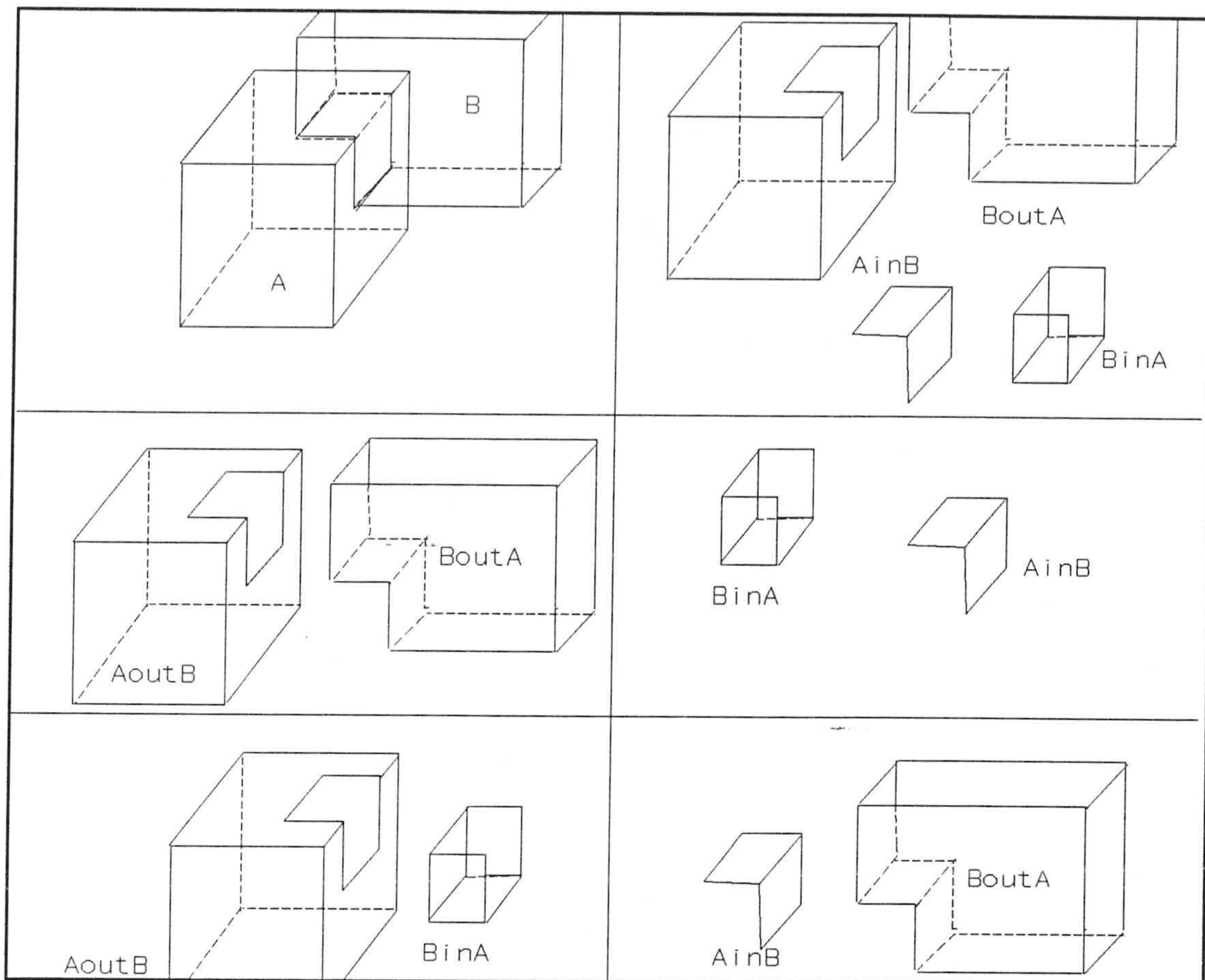


Figura I.5 Conversión de CSG a B-rep

La determinación del objeto resultante se divide a su vez en dos pasos:

- subdividir las primitivas en dos partes
- seleccionar las partes relevantes, dependiendo del operador y combinar esas partes.

Las caras de las primitivas se subdividen en la cadena de las líneas de intersección. Las partes resultantes de las caras de una primitiva (A) se clasifican como dentro o fuera de la otra primitiva (B). A es subdividido entre $A \text{ in } B$, que consiste en la partes de A dentro de B y $A \text{ out } B$, que consiste en las partes de A fuera de B.

- Para determinar el objeto resultante, las partes relevantes de A y B se seleccionan dependiendo del operador :

- $A \cup B$: $A_{out}B$ y $B_{out}A$

- $A \cap B$: $A_{in}B$ y $B_{in}A$

- $A - B$: $A_{out}B$ y $B_{in}A$

- $B - A$: $B_{out}A$ y $A_{in}B$

Las partes seleccionadas se combinan para construir la frontera del objeto resultante.

Todas las operaciones mencionadas principalmente consisten en accesos a las estructuras de datos de las primitivas y del objeto resultante. Por ejemplo, en la determinación de las líneas de intersección, se añaden nuevos vértices a las estructuras de datos de las primitivas.

Sólo se ha indicado un breve esbozo de un algoritmo de evaluación de fronteras. Un algoritmo implementado en la práctica es mucho más complejo. Por ejemplo, la determinación de la intersección de las líneas debe hacerse tan eficientemente como sea posible, teniendo en cuenta que pueden existir caras paralelas y que se deben evitar las inestabilidades numéricas en la determinación de las líneas de intersección.

Se han diseñado varios algoritmos para la aceleración de estas operaciones. Se siguen dos métodos: la definición de "zonas activas" (en el cálculo de la frontera sólo interviene una parte de la primitiva que se está evaluando) y de las "fronteras S" (cada primitiva se encierra en una caja sobre la que se hacen operaciones previas).

En este campo cabe destacar los trabajos de Requicha ([ReqVo95]), Hoffmann ([HofHK89]), Flaquer([FlaCM87]), Rossignac ([RosVo89]), Tilove ([Tilov80]), Cameron ([CamRo89]) y Vanecek ([Vanec89]), al objeto de acelerar y mejorar los algoritmos de evaluación de fronteras.

CAPITULO II

Sistemas de generadores en un álgebra gráfica

Una vez revisados los conceptos fundamentales del modelado de sólidos vamos a establecer una formalización que nos permita trabajar con modelos B-rep dentro de un álgebra gráfica. La resolución se hará definiendo unos elementos singulares que serán sistemas de generadores dentro del álgebra y que quedan perfectamente definidos a partir de la representación B-rep.

En primer lugar realizaremos una breve descripción de los objetos con los que trabajaremos.

A continuación repasaremos brevemente las dos formalizaciones más completas realizadas hasta ahora, detallando más la de Torres, que sirve de fundamento a este trabajo.

Posteriormente daremos un conjunto de definiciones necesarias para todo el desarrollo posterior. Por último enunciaremos un teorema de descomposición de objetos en elementos generadores que supone la parte central de este capítulo, y se darán una serie de consecuencias prácticas.

2.1 Tipos de sólidos

Ya se ha indicado anteriormente, en el apartado 1.2, que los sólidos a representar habían de ser rígidos, homogéneos, finitos, cerrados bajo movimientos rígidos y bajo ciertas operaciones booleanas, con descripción finita y con frontera bien determinada.

Un conjunto es regular si puede expresarse como la unión de su interior y su frontera, siendo esta cerrada y orientable. En principio, la frontera podría ser cualquier superficie, aunque sólo nos interesan aquellas que pueden ser descritas de un modo finito, y en particular solo trabajaremos con superficies que puedan ser aproximadas por conjuntos finitos de elementos superficiales planos.

En muchos sistemas de representación se exige que la superficie que representa la frontera de un sólido sea una variedad inmersa en el espacio euclídeo tridimensional. Una superficie es una variedad si para todo punto existe un entorno, en la topología inducida por el sólido, que es homeomorfo a un disco. Así mismo debe ser orientable, es decir, tener definido claramente el exterior y el interior de la superficie.

Trabajar con sólidos variedad simplifica las relaciones topológicas pero las operaciones regularizadas no son cerradas en el conjunto de las variedades. Por este motivo se están revisando los tipos de sólidos manejados por los sistemas de representación.

Tres son las soluciones posibles ([Hoffm89]):

- trabajar sólo con variedades y considerar como un error las operaciones que dan como resultado un sólido no-variedad.
- trabajar con sólidos variedades pero permitir coincidencias de estructuras topológicas separadas.
- permitir sólidos no-variedades tanto en la entrada como en el resultado de las operaciones.

La tercera solución es la que presenta más ventajas siempre que se elija correctamente la estructura de datos que los represente y que en los algoritmos se tenga en cuenta la existencia de vértices o aristas no-variedades.

Supondremos en todo el desarrollo posterior que al hablar de sólidos nos referimos a sólidos variedad y no variedad en dos y tres dimensiones, en las que cada componente es un polígono determinado por aristas rectas, en el caso de dos dimensiones, ó un poliedro determinado por caras planas en el caso de tres dimensiones.

Podemos, por tanto, tener en cuenta el siguiente concepto de sólido [Karas88]:

Un **politopo** convexo regular en \mathbb{R}^n es un conjunto limitado de puntos formado como la intersección regularizada de un número finito de semiespacios cerrados.

Un **sólido** es el resultado de combinar un número limitado de politopos convexos regulares usando operaciones regularizadas.

2.2 Antecedentes

Tal y como hemos indicado anteriormente, los modelos computacionales proceden de una abstracción matemática previa. Una vez que se han obtenido resultados prácticos en un campo determinado se busca un modelo matemático capaz de caracterizar formalmente los diversos elementos de trabajo, que sirva a la vez para unificar el tratamiento dado al problema.

Son varios los trabajos realizados hasta ahora en torno a la formalización en Informática Gráfica (ver [Duce_89] y [Duce_91]). Dos son los que, con nuestro enfoque resultan más interesantes: el modelo de Fiume [Fiume89], orientado especialmente a la formalización del proceso de visualización en dispositivos raster y el de Torres [Torres92], que propone una extensión del modelo de Fiume que contempla operaciones de modelado.

La teoría de Fiume sobre gráficos raster tiene por objeto el estudio del proceso de visualización en dispositivos raster. Su formalismo se basa en el concepto de objeto gráfico. Un objeto estático se define como un par:

$$(Z_0, I_0) \quad \text{donde } Z_0 \subseteq \mathbb{R}^3, I_0: Z_0 \rightarrow C$$

y C es un espacio de color. El espacio de color se puede definir como un subconjunto \mathbb{R}^c donde $c \geq 1$. Un objeto puede ser definido como una primitiva dando su conjunto Z_0 y su función I_0 , ó bien como una combinación de otros objetos gráficos. Los objetos pueden ser unidos, intersectados ó restados, aunque en estos casos las funciones de visualización no mantienen una estructura matemática (en el caso de la unión se promedian, en la intersección se realiza la operación "or", etc), por lo que dicha formalización no puede ser usada como un entorno de trabajo completo para el modelado.

El formalismo de Torres ([Torre92]) es una extensión de la Teoría de Fiume. Generaliza el concepto de objeto gráfico de Fiume, dotando al conjunto de objetos de operaciones que permiten interpretar muchas de las operaciones usuales de la informática gráfica. Los conceptos de partida de dicha teoría son los siguientes:

-Un *transformación geométrica*, $\tau: \mathbb{R}^n \rightarrow \mathbb{R}^n$, es una biyección que cumple la siguiente propiedad

$$\forall X, Y \in \mathbb{R}^n \quad \forall \beta \in [0, 1] \subset \mathbb{R} \Rightarrow \tau(\beta \cdot X + (1-\beta) \cdot Y) = \beta \cdot \tau(X) + (1-\beta) \cdot \tau(Y)$$

Dicha definición implica la transformación de segmentos de recta en segmentos de recta, aunque no implica que una transformación geométrica tenga que ser aplicación lineal.

- Una ξ -*estructura* sobre un cuerpo K es un conjunto Z en el cual está definidas las siguientes operaciones:

- Una operación binaria interna, $+$, denominada suma con la cual Z es un grupo conmutativo.

- Una operación externa sobre el cuerpo \mathbb{K} , $*$, denominada producto escalar con la cual $(Z, +, *)$ es espacio vectorial.
- Dos operaciones binarias internas, \cup y \cap denominadas unión e intersección y una operación unaria interna, \sim , denominada complemento, con las cuales Z es un álgebra de Boole. El elemento vacío coincide con el elemento neutro del grupo conmutativo.
- Una operación binaria interna, \times , denominada producto que satisface las propiedades: asociatividad, conmutatividad, existencia de elemento identidad y el elemento neutro del grupo conmutativo actúa como elemento adsorbente.

Es fácil comprobar que existen estructuras algebraicas que cumplen estas propiedades. Veamos un ejemplo.

Sea Z el conjunto de números enteros comprendidos entre 0 y 2^p-1 , con p entero. Se sabe que definiendo la suma y el producto por escalar como las operaciones usuales módulo 2^p , respecto de ellas Z es un espacio vectorial.

Si consideramos el producto de enteros restringido a Z , por ser Z un intervalo, dicha operación verifica las propiedades asociativa, conmutativa, elemento identidad, y el 0 actúa como elemento adsorbente.

Para dotar a Z de estructura de álgebra de Boole, consideramos la representación binaria de cada uno de los enteros de Z , y las operaciones unión e intersección realizadas bit a bit entre dichas representaciones. El 0 es el elemento vacío de esta álgebra de Boole.

Por lo anterior, Z dotado de las operaciones indicadas es una ζ -estructura.

Otro ejemplo similar en el que existen elementos positivos y negativos es el intervalo $Z' = [-2^{r-1} + 1, 2^{r-1}]$, donde r es entero. Consideramos la suma y el producto por escalar como las operaciones usuales entre enteros restringidas a Z' . Con dichas operaciones Z' es un espacio vectorial.

Si consideramos el producto de enteros restringido a Z' , por ser Z un intervalo, dicha operación verifica las propiedades asociativa, conmutativa, elemento identidad, y el 0 actúa como elemento adsorbente.

Para dotar a Z' de estructura de álgebra de Boole, consideramos la representación en complemento a dos de cada uno de los opuestos de los enteros de Z' , y las operaciones unión e intersección realizadas bit a bit entre dichas representaciones. El 0 es el elemento vacío de esta álgebra de Boole.

El motivo de considerar en Z' el complemento a dos del opuesto de cada entero de Z' , es para que el complementario del 0 sea el 1, lo que en muchos casos será más adecuado para el significado que queremos dar a la ζ -estructura. Cuando se verifique esta propiedad diremos que la ζ -estructura es normal.

La definición de un objeto gráfico contempla dos factores: presencia y aspecto. El *aspecto* de un objeto gráfico contiene información de su apariencia visual y atributos no geométricos (color, transparencia, coeficientes de reflexión y refracción, densidad, tipo de material, etc.)

La *presencia* de un objeto gráfico describe la ocupación del espacio, es decir, el grado de multiplicidad con que un objeto ocupa un espacio determinado. Un objeto gráfico tiene presencia en un punto si hay material del objeto en ese punto. La presencia nos permite contar la existencia ó no del objeto, por lo que podemos construir objetos por superposición ó substracción de otros objetos.

Para manejar la presencia y el aspecto de un objeto gráfico definimos las estructuras algebraicas de los dominios de presencia y aspecto. Un *dominio de aspecto* ó *espacio de aspecto*, δ , es un conjunto de valores de aspecto que es una ζ -estructura. En el mismo sentido, un *dominio de presencia* ó *espacio de presencia*, π , es un conjunto de valores de presencia que es una ζ -estructura. Supondremos también que los dominios de aspecto y presencia están dotados de un orden total.

Formalmente, un *volumen*, V , es un subconjunto de \mathbb{R}^n . El *volumen* de un objeto gráfico es la porción de espacio donde el objeto está, es decir donde tenemos alguna información sobre el objeto.

Un *universo* de objetos gráficos, \mathcal{U} , es una terna $u=(\pi,\delta,n)$ donde π y δ son dominios de presencia y aspecto respectivamente, y n la dimensión del espacio euclídeo.

Un *objeto gráfico*, en el universo $\mathcal{U}=(\pi,\delta,n)$, es un par (μ,α) , donde:

μ es una función, denominada *función de presencia*, definida como:

$$\mu: \mathbb{R}^n \rightarrow \pi$$

α es una función, denominada *función de aspecto*, definida como:

$$\alpha: \mathbb{R}^n \rightarrow \delta$$

En el caso de que tanto μ como α sean constantes, es decir, tomen el mismo valor en todo el volumen del objeto, diremos que el objeto es *homogéneo*.

Por ejemplo, un triángulo podría definirse como:

Triángulo($\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$) = (μ, α) con:

$$\mu(\mathbf{P}) = \begin{cases} 1 & \text{si } \exists \beta, \gamma, \delta \in [0,1], \beta + \gamma + \delta = 1 \mid \mathbf{P} = \beta \mathbf{P}_1 + \gamma \mathbf{P}_2 + \delta \mathbf{P}_3 \\ 0 & \text{en otro caso} \end{cases}$$

$$\alpha(\mathbf{P}) = k \cdot \mu(\mathbf{P})$$

donde k es el valor de aspecto para el triángulo.

Cada objeto gráfico tiene asociado un volumen, que puede ser definido como la zona del espacio en el que hay información del objeto. Este volumen no debe confundirse con el volumen material. El *volumen* V asociado a un objeto gráfico $O=(\mu,\alpha)$ es el conjunto de puntos que verifican:

$$\text{Vol}(O) = \{\mathbf{P} \in \mathbb{R}^n \mid \alpha(\mathbf{P}) \neq 0 \vee \mu(\mathbf{P}) \neq 0\}$$

Supondremos que tanto α como μ son semianalíticas.

En el conjunto de objetos gráficos tenemos definidas las siguientes operaciones: suma, producto por escalar, unión, intersección, complementación, producto de objetos y producto circular, además de las transformaciones geométricas. Para dos objetos, $O_1=(\mu_1,\alpha_1)$ y $O_2=(\mu_2,\alpha_2)$, y cualquier valor $k \in \mathbf{K}$, estas operaciones se definen como sigue [Torres92]:

Transformación geométrica de un objeto:

$$T(O_1)=(\mu_1 \circ T^{-1}, \alpha_1 \circ T^{-1})$$

Suma: $O_1 + O_2 = (\mu_1 + \mu_2, \alpha_1 + \alpha_2)$

Producto escalar: $k*O_1 = (k*\mu_1, k*\alpha_1)$

Unión: $O_1 \cup O_2 = (\mu_1 \cup \mu_2, \alpha_1 \cup \alpha_2)$

Intersección: $O_1 \cap O_2 = (\mu_1 \cap \mu_2, \alpha_1 \cap \alpha_2)$

Complementación: $\sim O_1 = (\bar{\mu}_1, \bar{\alpha}_1)$

Producto: $O_1 \times O_2 = (\sum_{\delta(O_1,O_2,P)} \mu_1(\mathbf{P}_1) \times \mu_2(\mathbf{P}_2), \sum_{\delta(O_1,O_2,P)} \alpha_1(\mathbf{P}_1) \times \alpha_2(\mathbf{P}_2))$

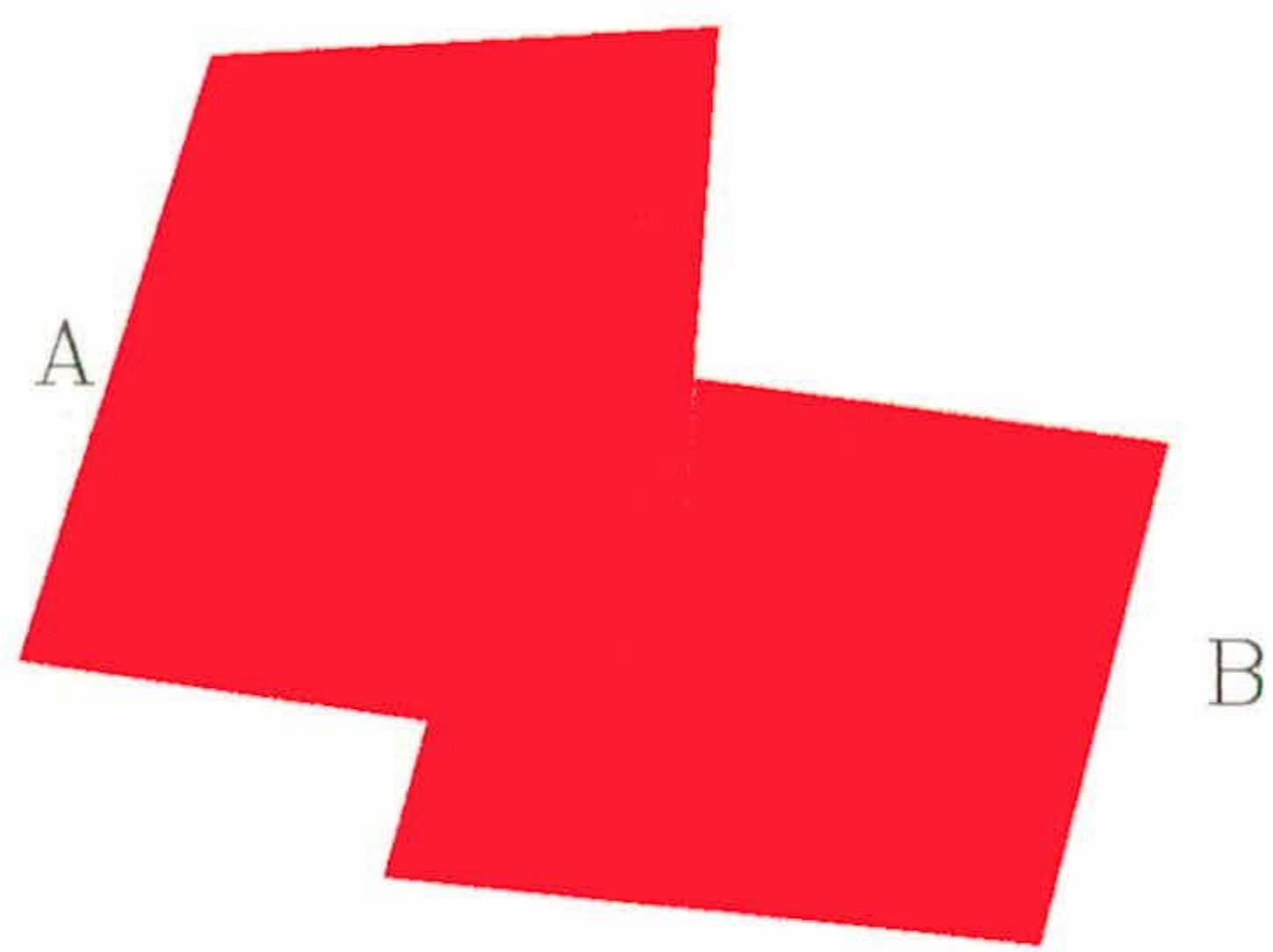
$$\delta(O_1,O_2,P) = \{ (\mathbf{P}_1,\mathbf{P}_2) \mid \mathbf{P}_1 \in \text{Vol}(O_1), \mathbf{P}_2 \in \text{Vol}(O_2) \text{ and } \mathbf{P} = \mathbf{T}_t^{[P_1]}(\mathbf{P}_2) \}$$

Producto circular: $O_1 \otimes O_2 = (\sum_{\delta_c(O_1,O_2,P)} \mu_1(\mathbf{P}_1) \times \mu_2(\mathbf{P}_2), \sum_{\delta_c(O_1,O_2,P)} \alpha_1(\mathbf{P}_1) \times \alpha_2(\mathbf{P}_2))$

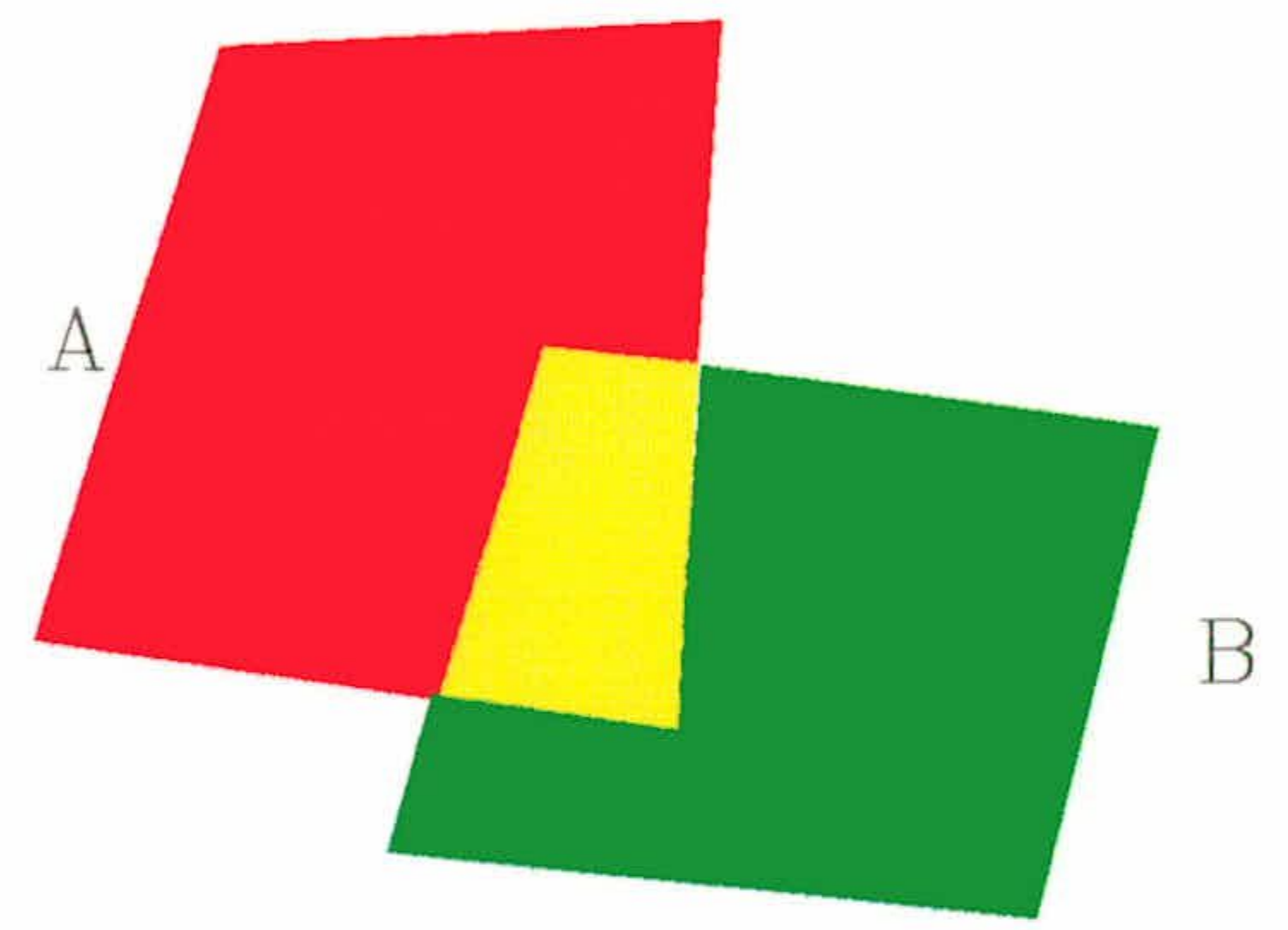
$$\delta_c(O_1,O_2,P) = \{ (\mathbf{P}_1,\mathbf{P}_2) \mid \mathbf{P}_1 \in \text{Vol}(O_1), \mathbf{P}_2 \in \text{Vol}(O_2) \text{ and } \mathbf{P} = \mathbf{T}_c^{[P_1]}(\mathbf{P}_2) \}$$

El conjunto de objetos gráficos en un universo es una ζ -estructura [TorC193].

Las operaciones con objetos gráficos nos permiten modelar objetos complejos a partir de otros más simples. La Figura 2.1 nos muestra una composición de dos objetos bidimensionales mediante las operaciones de unión, intersección, suma y diferencia. Los objetos de la columna izquierda tienen el mismo valor de presencia y aspecto. En la columna derecha se han calculado las mismas operaciones con dos objetos que tienen diferentes valores de aspecto y presencia.



a) UNION



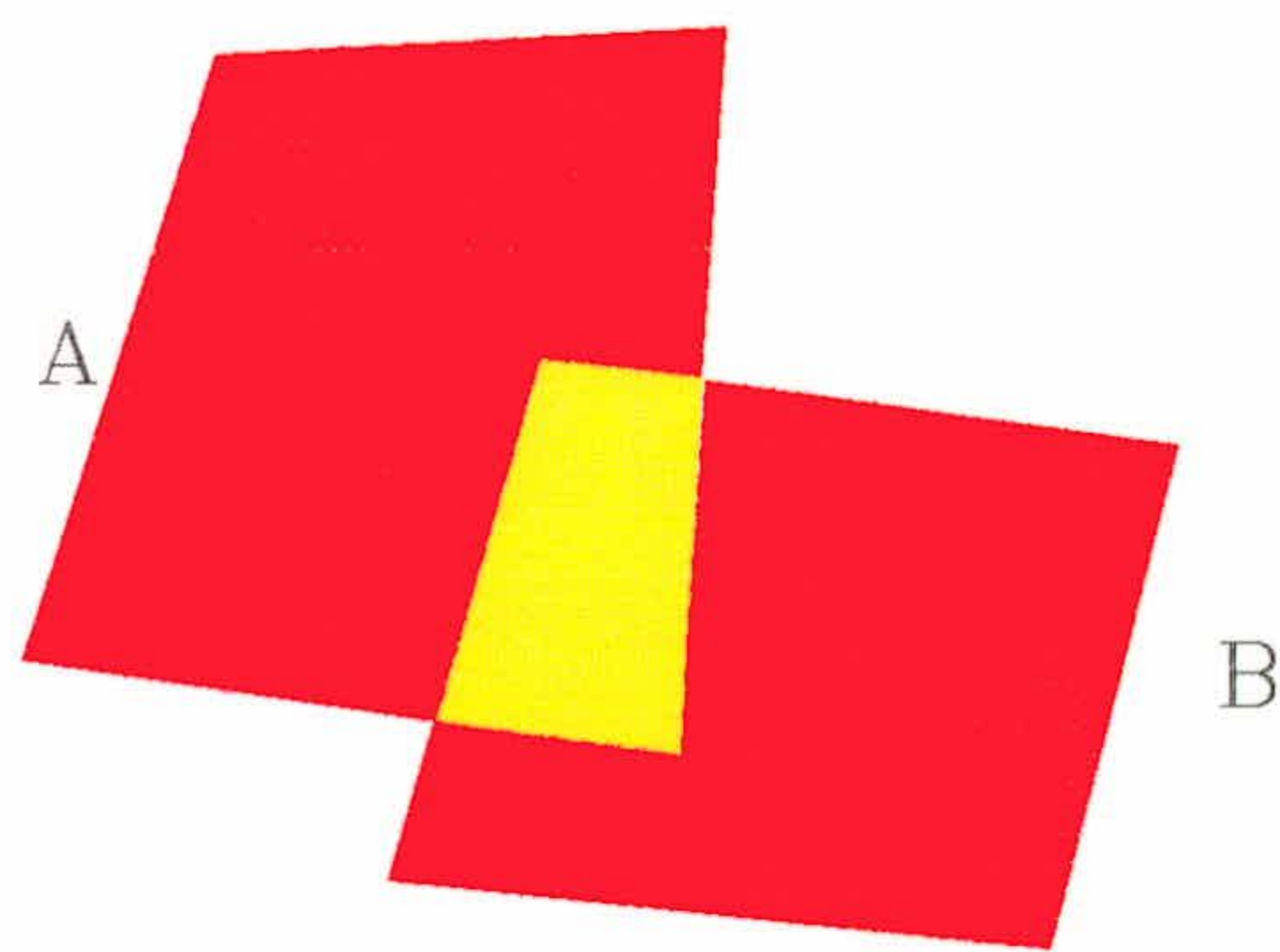
b) UNION



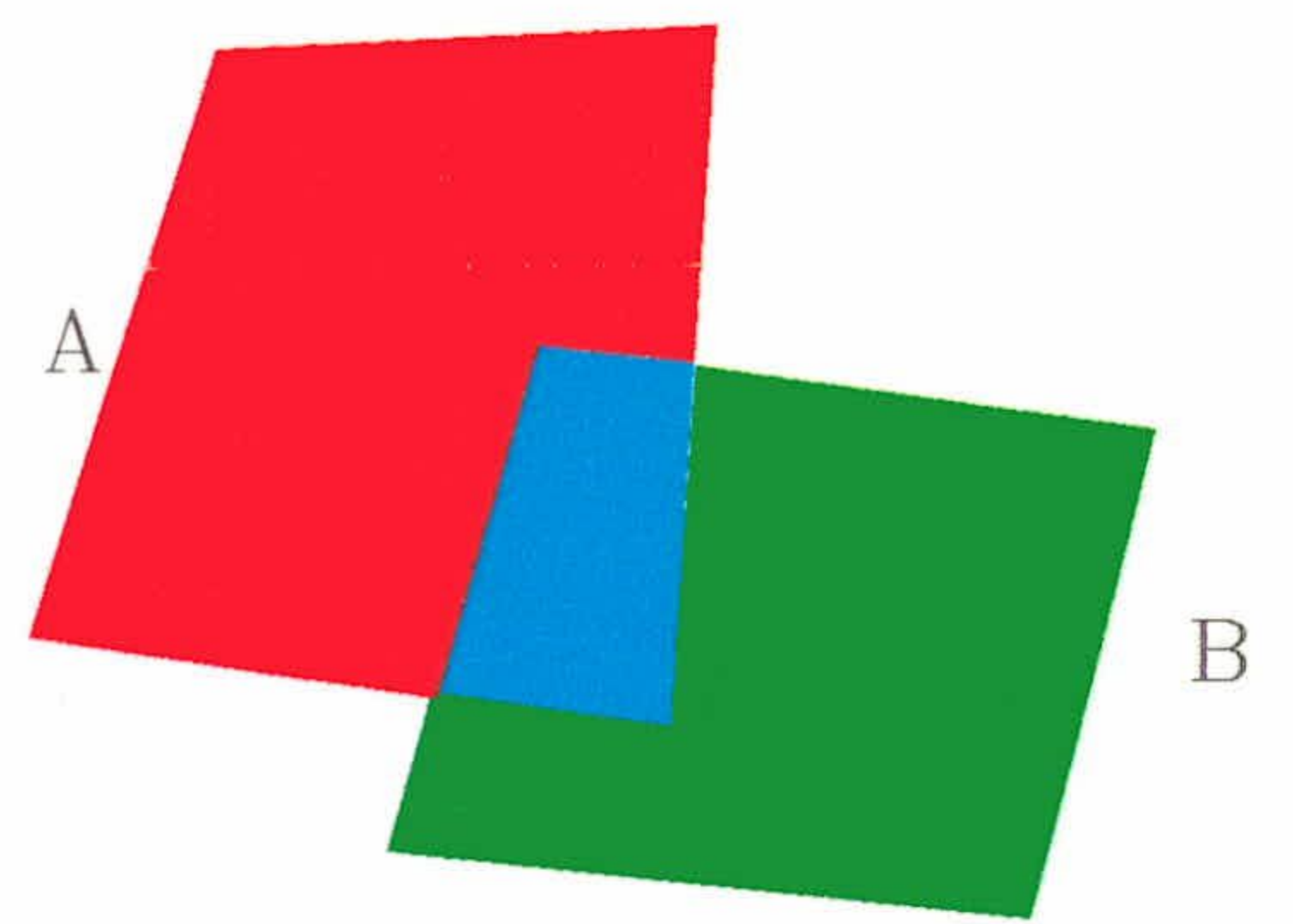
c) INTERSECCION



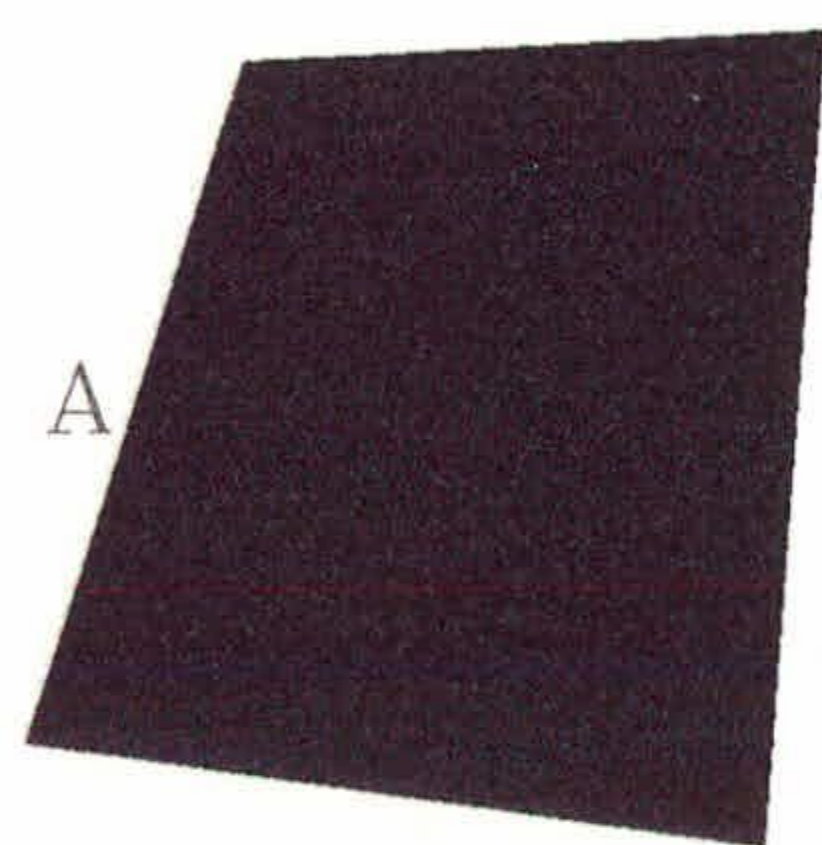
d) INTERSECCION



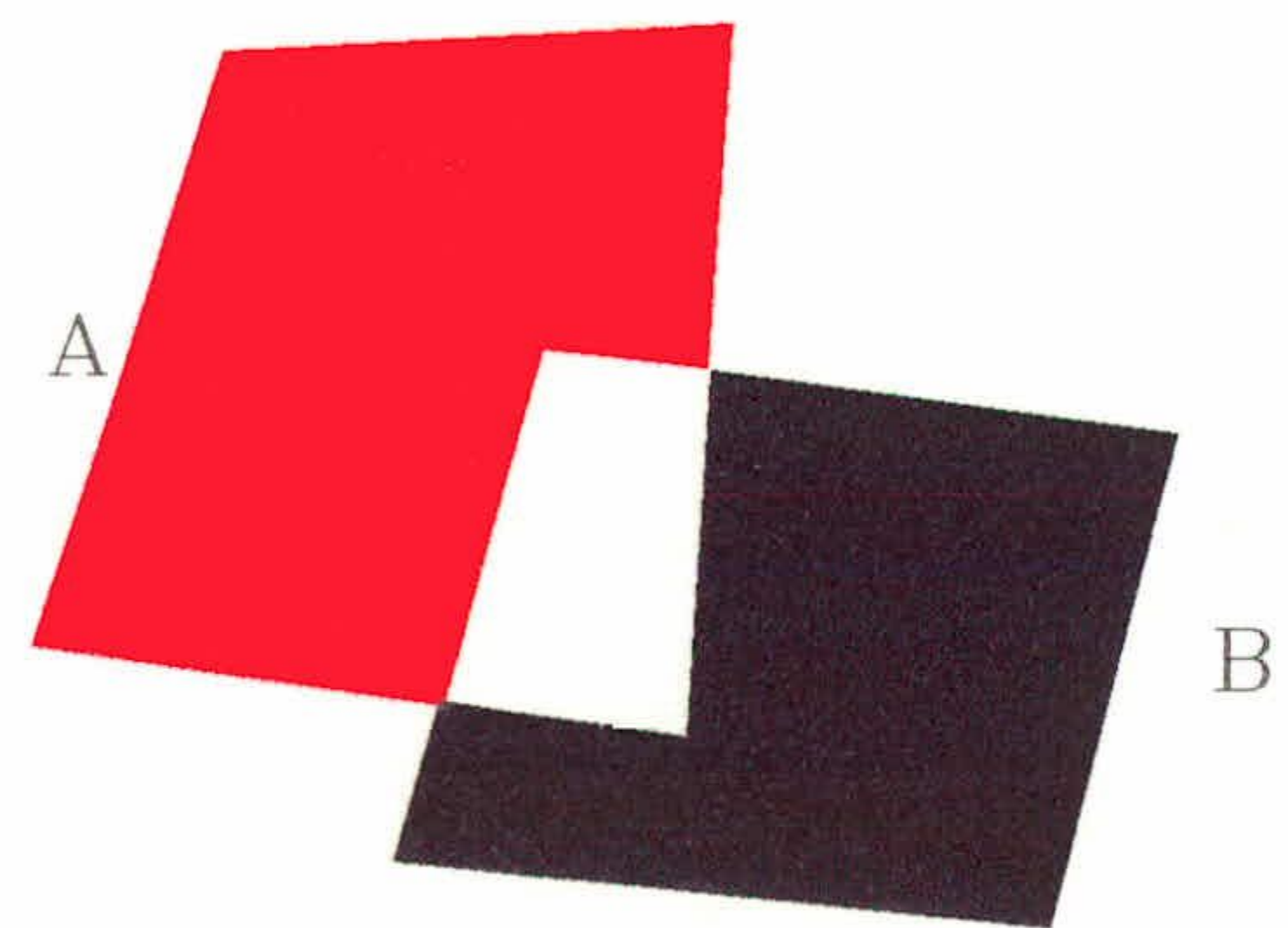
e) SUMA



f) SUMA



g) DIFERENCIA



h) DIFERENCIA

Figura 2.1 Operaciones de objetos

Por medio de las operaciones anteriores se pueden generar mediante expresiones de objetos gráficos los sólidos formados por expresiones booleanas (modelado CSG) así como los obtenidos por medio de enumeración espacial. Mediante las operaciones de producto y de producto circular se generaliza la representación de sólidos por barrido (ver [TorC193]).

Uno de los puntos que queda abierto en este trabajo es el modelado de sólidos a partir de la frontera, y por tanto la posibilidad de un tratamiento global de las representaciones B-rep y CSG. Estas representaciones son las más importantes tanto desde el punto de vista teórico como práctico.

Junto a lo anterior, al objeto de aplicar las operaciones booleanas en métodos de modelado se hace necesario definir las operaciones regularizadas. A continuación vamos a definir y estudiar el comportamiento del álgebra gráfica en relación a esas operaciones regularizadas.

2.2 Operaciones regularizadas

Suponemos que trabajamos en \mathbb{R}^n , al que consideramos, por un lado, como un espacio topológico, con la topología inducida por la métrica usual, y por otro lado como espacio afín, en el que están definidas las operaciones usuales sobre puntos, y por extensión sobre conjuntos de puntos. Así mismo suponemos establecida la formalización de Torres.

Haremos uso de las siguientes definiciones topológicas [Kurat66]

Definición 2.1 Diremos que \mathbf{P} es un *punto interior* a un conjunto A , si existe un entorno del punto, $\varepsilon(\mathbf{P})$, totalmente contenido en A . $\text{Int}(A)$ denotará el conjunto de punto interiores de un conjunto A :

$$\text{Int}(A) = \{ \mathbf{P} \mid \exists \varepsilon(\mathbf{P}), \varepsilon(\mathbf{P}) \subset A \}$$

Definición 2.2 Diremos que \mathbf{P} es un *punto frontera* de A si en todo entorno de \mathbf{P} , $\varepsilon(\mathbf{P})$, existen puntos de A y de su complementario A^c . $\text{Fr}(A)$ denotará la frontera de un conjunto A :

$$\text{Fr}(A) = \{ \mathbf{P} \mid \forall \varepsilon(\mathbf{P}), \varepsilon(\mathbf{P}) \cap A \neq \emptyset \wedge \varepsilon(\mathbf{P}) \cap A^c \neq \emptyset \}$$

Definición 2.3 Diremos que \mathbf{P} es un *punto adherente* de A si en todo entorno de \mathbf{P} , $\varepsilon(\mathbf{P})$, existen puntos de A . $\text{Cl}(A)$ denotará el conjunto de puntos adherente de un conjunto A , y lo denominaremos clausura de A :

$$\text{Cl}(A) = \{ \mathbf{P} \mid \forall \varepsilon(\mathbf{P}), \varepsilon(\mathbf{P}) \cap A \neq \emptyset \}$$

Puede demostrarse fácilmente la siguiente igualdad [Kurat66]:

$$\text{Cl}(A) = (\text{Int}(A^c))^c$$

El objetivo de definir la operaciones regularizadas en los sistemas de modelado es evitar que alguna operación dé como resultado sólidos de dimensionalidad no uniforme. Así, si dos sólidos de \mathbb{R}^3 tienen en común una sola arista, esta sería la intersección común, que no es de dimensión 3. En [Requi78] se definen las operaciones regularizadas y se demuestra que respecto a ellas los sólidos forman un Algebra de Boole. Necesitamos dar un significado a esas definiciones dentro del álgebra gráfica.

Antes de introducir el concepto de operación regularizada es necesario que introduzcamos el concepto de *objeto gráfico regular*. El problema radica en que en un objeto gráfico cuyo volumen no sea un conjunto regular, al calcular su clausura existen puntos en los que es necesario definir las funciones de aspecto y presencia. Por ejemplo, cuando calculamos la clausura de un objeto gráfico cuyo volumen es un conjunto abierto, es necesario decidir que valores de aspecto y presencia asignaremos a los puntos de la frontera. Así mismo podría ocurrir que existieran puntos de un objeto gráfico con unos valores determinados de aspecto y presencia, y que no pertenecieran a la clausura del volumen. Este caso se daría con aquellos puntos adherentes al volumen de un objeto y que no estén en la clausura de dicho volumen. Comenzaremos caracterizando los puntos indicados anteriormente.

Definición 2.4 Dado un objeto gráfico, $O=(\mu, \alpha)$, y un punto \mathbf{P} de R_n . Diremos que \mathbf{P} es *un punto irregular* de O si para todo entorno de \mathbf{P} arbitrariamente pequeño existe al menos otro punto \mathbf{P}_1 tal que $\mu(\mathbf{P}) \neq \mu(\mathbf{P}_1)$ o $\alpha(\mathbf{P}) \neq \alpha(\mathbf{P}_1)$.

Lo puntos no irregulares se denominarán *puntos regulares*.

En la figura 2.2 aparecen algunos ejemplos en los que es necesario regularizar el objeto gráfico:

En el caso a), se tiene un rectángulo con valores de aspecto y presencia 1, en el que no se incluye la frontera. Los puntos de la frontera son puntos irregulares.

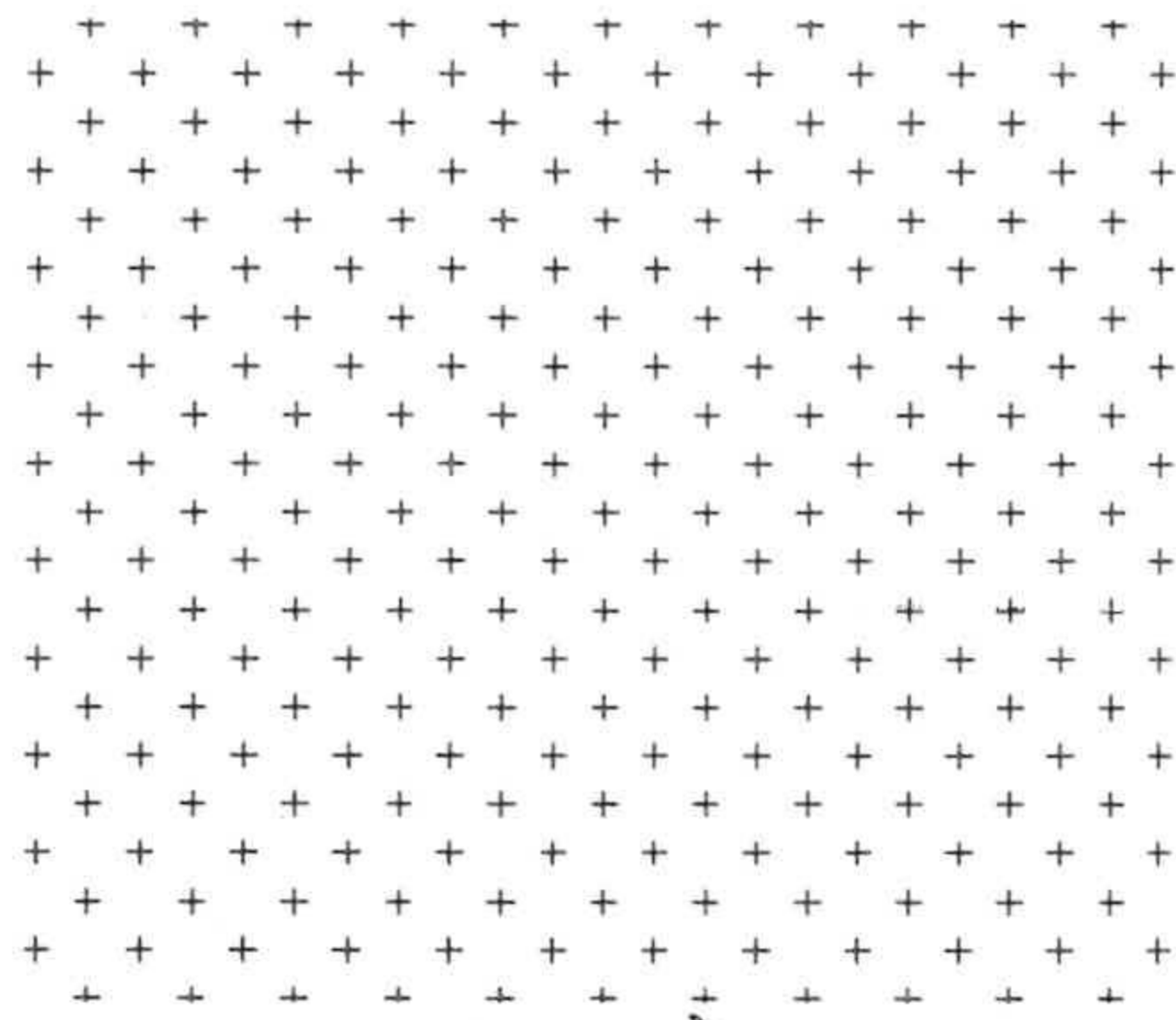
En el caso b), se tiene un rectángulo con valores de aspecto y presencia 1, y un segmento exterior al rectángulo, con valores de aspecto y presencia 1. Los puntos de la frontera (formada por la frontera del rectángulo y el segmento exterior) son puntos irregulares.

En el caso c), se restan dos rectángulos de aspecto y presencia 1, y que tienen una arista en común. Los puntos de la frontera de ambos rectángulos son puntos irregulares.

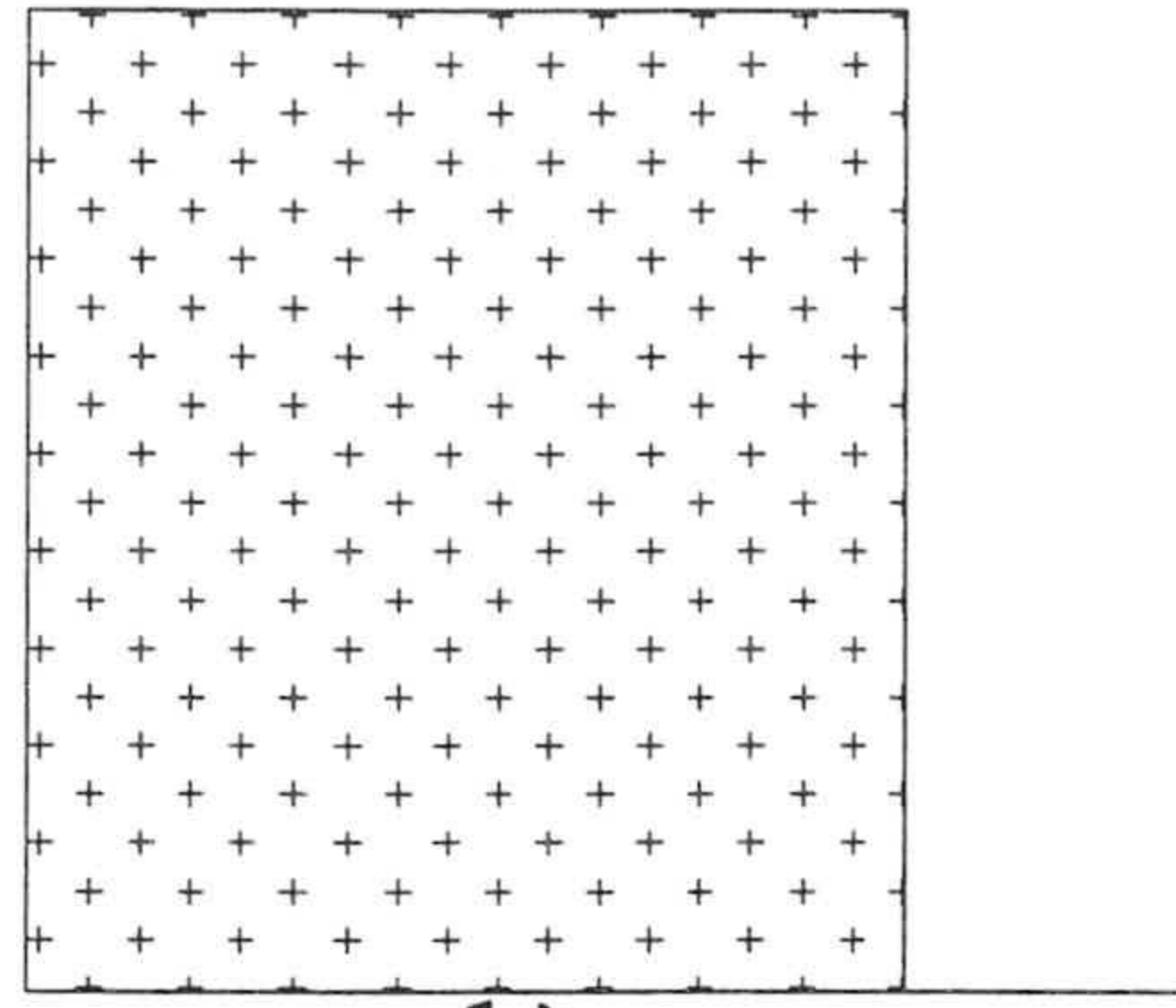
En el caso d), se tiene un rectángulo de aspecto y presencia -1 en el que no se incluye la frontera. Los puntos de la frontera son puntos irregulares.

En el caso e), se tiene un rectángulo, incluida la frontera pero excluido el centro. Los puntos de la frontera y el centro son puntos irregulares.

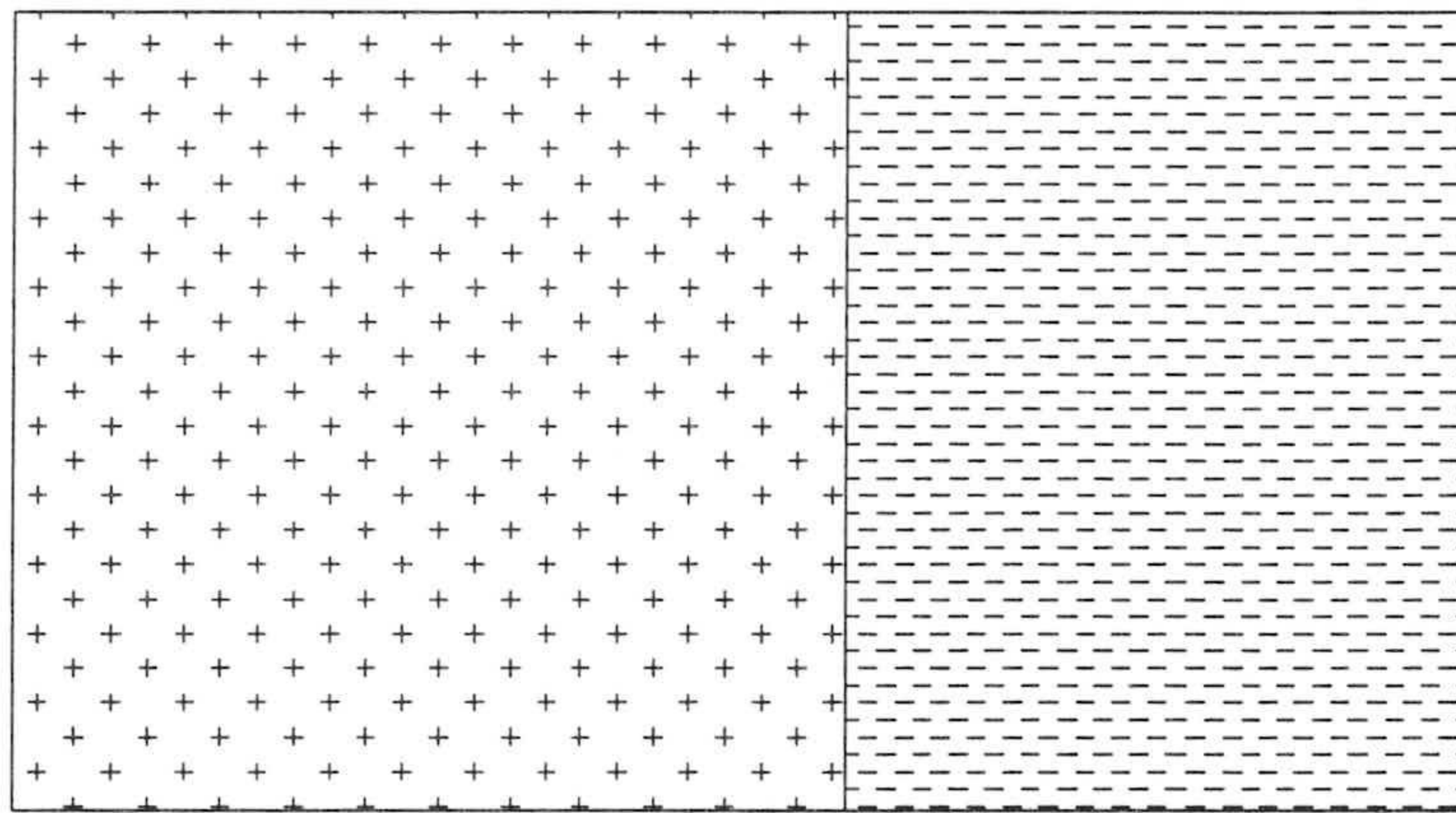
En orden a regularizar un objeto será necesario asignar valores de presencia y/o aspecto a los puntos irregulares, aunque en algunos casos se mantengan los valores de aspecto y presencia.



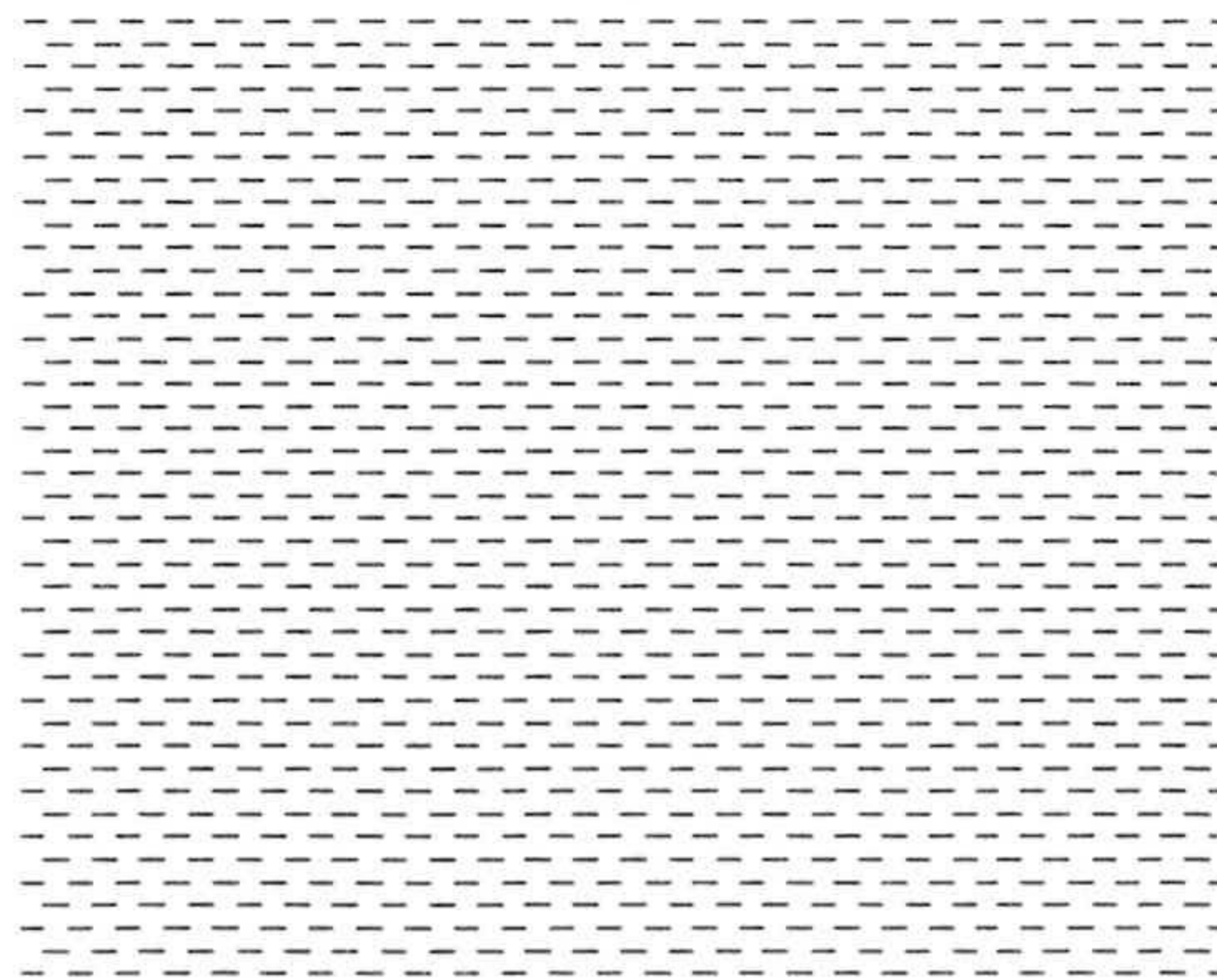
a)



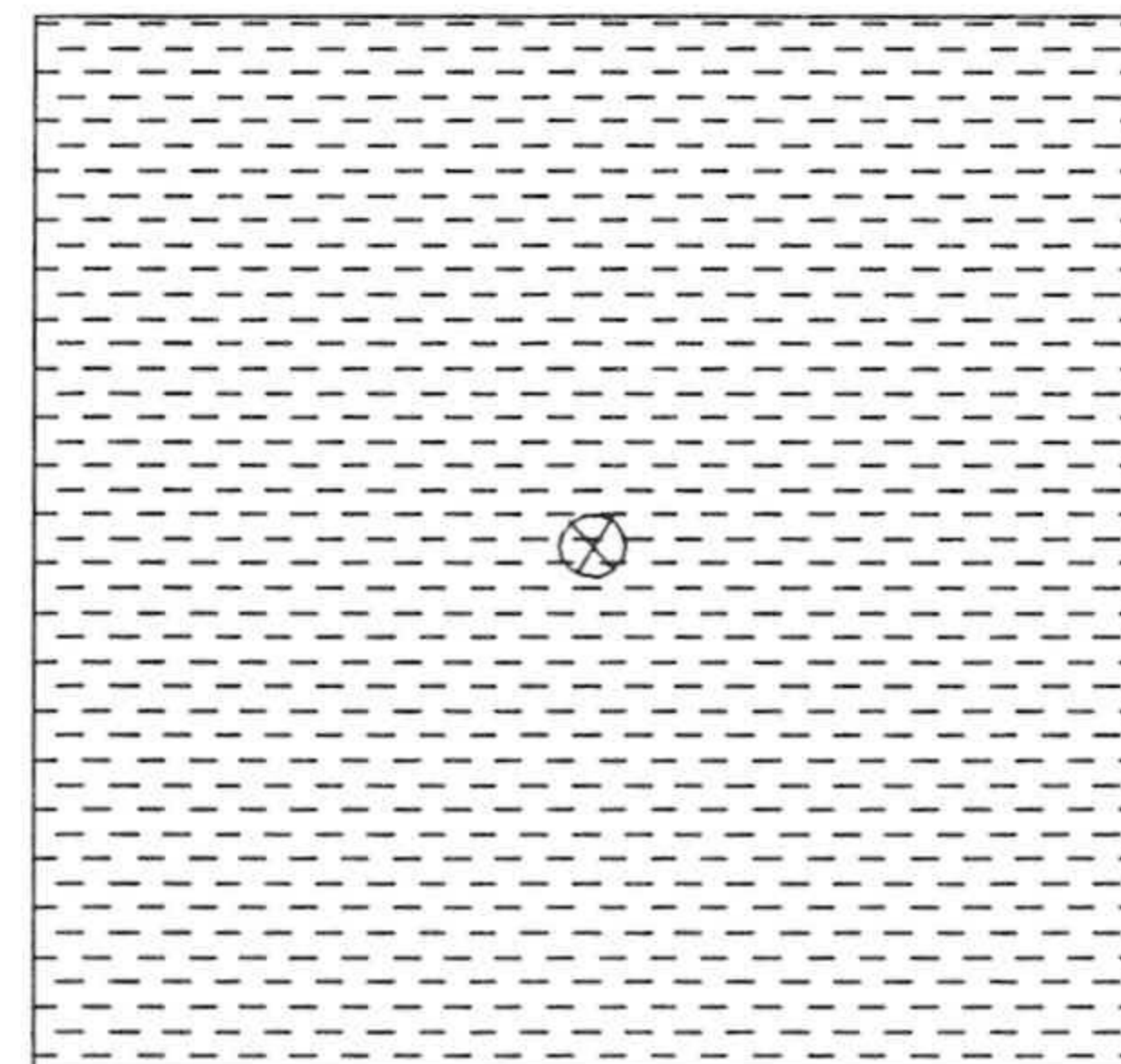
b)



c)



d)



e)

Figura 2.2 Objetos a regularizar

Definición 2.5 Dado un objeto gráfico, $\mathbf{O}=(\mu, \alpha)$, definimos la *clausura gráfica* $\text{Clg}(\mathbf{O})$, como el objeto que resulta de asignar valores de aspecto y presencia a cada punto irregular del siguiente modo:

Si \mathbf{P} está en $\text{vol}(\mathbf{O})$ y no en su clausura se asignan valores de aspecto y presencia 0.

Si \mathbf{P} está en $\text{cl}(\text{vol}(\mathbf{O}))$ se le asigna como aspecto (respectivamente presencia) el máximo de los valores positivos de aspecto (respectivamente presencia) de los puntos regulares existentes, si los hay, en un entorno tan pequeño como se quiera, centrado en \mathbf{P} . Si no existen valores positivos de aspecto (respectivamente presencia), se asigna el mínimo valor negativo.

Si un objeto coincide con su clausura gráfica diremos que es un objeto *gráfico regular*.

El hecho de elegir en unos casos el máximo y en otros el mínimo, dependiendo de si existen o no valores con aspecto o presencia positiva, se justifica con la figura 2.2,d): si eligiésemos para la frontera el máximo, este sería 0, y el volumen del objeto no sería un conjunto regular.

Es claro que algunos puntos irregulares mantienen los mismos valores de aspecto y presencia antes y después de la regularización: esto ocurre con los puntos frontera de objetos regulares.

Los puntos irregulares determinan una partición de todo objeto en subconjuntos abiertos (denominadas *componentes regulares*) cuyas fronteras son precisamente los puntos irregulares. El número de componentes regulares será finito si el objeto ha sido construido por medio de un número finito de operaciones del álgebra realizada entre objetos homogéneos.

Podemos ahora definir todas las operaciones regularizadas:

Unión:	$O_1 \cup^* O_2 = \text{Clg}(O_1 \cup O_2)$
Intersección:	$O_1 \cap^* O_2 = \text{Clg}(O_1 \cap O_2)$
Complemento:	$\sim^* O = \text{Clg}(\sim O)$
Suma:	$O_1 +^* O_2 = \text{Clg}(O_1 + O_2)$
Producto escalar:	$k^* O_2 = \text{Clg}(k^* O_2)$

Es necesario que el conjunto de objetos gráficos con las operaciones regularizadas definidas, mantenga las mismas propiedades que con las operaciones sin regularizar. Es decir, necesitamos que el conjunto de objetos gráficos tenga estructura de espacio vectorial y de álgebra booleana, en relaciones a las operaciones regularizadas. Sería posible una demostración exhaustiva de todas y cada una de las propiedades, aunque el siguiente teorema no evita realizar esa demostración individualizada. Vamos a demostrar que toda propiedad que se cumple con una operación determinada también se cumple considerando la misma operación regularizada.

Previamente enunciaremos un lema que nos permitirá simplificar la demostración del teorema posterior.

Lema 2.1 Para cualquier par de objetos gráficos regulares $O_1 = (\mu_1, \alpha_1)$ y $O_2 = (\mu_2, \alpha_2)$ y cualquier operación del álgebra, los puntos regulares de cada operando que no es irregular en el otro, son puntos regulares del resultado.

Demostración: Si \mathbf{P} es regular de un operando y no irregular del otro:

$$\exists \varepsilon(\mathbf{P}) \mid \forall \mathbf{P}' \in \varepsilon(\mathbf{P}) \text{ se verifica}$$

$$\mu_1(\mathbf{P}') = \mu_1(\mathbf{P}), \mu_2(\mathbf{P}') = \mu_2(\mathbf{P}), \alpha_1(\mathbf{P}') = \alpha_1(\mathbf{P}) \text{ y } \alpha_2(\mathbf{P}') = \alpha_2(\mathbf{P})$$

Por lo tanto, la expresión se evaluará como una constante en $\varepsilon(\mathbf{P})$ con lo que \mathbf{P} es regular del objeto resultante. ■

Podemos ahora enunciar el teorema que nos permitirá establecer la misma estructura algebraica para los objetos con las operaciones regularizadas.

Teorema 2.1 El conjunto de objetos gráficos con las operaciones regularizadas tiene las mismas propiedades que con las operaciones sin regularizar.

Demostración:

Supongamos que una cierta propiedad **Pr** se verifica para la operación **op** en el conjunto de objetos gráficos **O**; necesitamos demostrar que esta propiedad también se cumple para la operación regularizada **op***. La propiedad puede expresarse como una igualdad $E^*(O_1, \dots, O_p) = E'^*(O_1, \dots, O_p)$, donde E^* y E'^* son dos expresiones con operaciones regularizadas (por ejemplo $O_1 +^* O_2 = O_2 +^* O_1$, para la propiedad conmutativa). Supongamos que existen dos objetos, O_1 y O_2 , para los cuales la propiedad no se cumple. Existirá, por tanto, al menos un punto **Q**, tal que $E^*(O_1, \dots, O_p) \neq E'^*(O_1, \dots, O_p)$ en **Q**.

Es evidente que **Q** tiene que ser un punto irregular en alguno de los operandos ya que en caso contrario la operación regularizada da el mismo resultado que la operación sin regularizar, y por tanto en ese punto la propiedad debería cumplirse. Por lo anterior, **Q** es irregular en alguno de los operandos O_i ; las expresiones E^* y E'^* se calculan como a partir de los valores en los puntos regulares de un entorno de **Q** tan pequeño como se quiera; pero ya que dicho valor se calcula usando solo los puntos del entorno que son regulares, las expresiones E y E' toman el mismo valor, por lo que también deben mantenerse los valores 0, el máximo ó el mínimo, y se concluye que la propiedad debe también cumplirse en **Q**. ■

El teorema anterior nos permite concluir que el conjunto de objetos gráficos en un universo, con las operaciones regularizadas, es una ζ -estructura.

Una representación CSG de un sólido consiste en una expresión booleana de objetos gráficos, denominados primitivas. Es inmediata la formalización de dicho sistema de representación en el álgebra gráfica usando operaciones regularizadas.

Así mismo se pueden añadir las operaciones de suma y producto por escalar a las expresiones CSG, manteniendo la uniformidad de la representación. Estas operaciones permiten variar los valores de presencia y aspecto, manteniendo el mismo

volumen, mientras que las operaciones booleanas permiten a variar la forma manteniendo la homogeneidad de las componentes ya existentes.

En lo sucesivo supondremos que estamos trabajando con objetos regulares ó que han sido obtenidos a partir de ellos por medio de un número finito de operaciones regularizadas del álgebra.

2.4 Sistemas de generadores

Una vez establecida la estructura del conjunto de objetos gráficos con las operaciones regularizadas parece conveniente el estudio de elementos singulares de dicho conjunto en relación a los demás. En particular nos interesa la posible existencia de sistemas de generadores, es decir, de conjuntos de elementos singulares a partir de los cuales puedan obtenerse todos los demás por medio de las operaciones del álgebra.

Ya hemos visto anteriormente que además de las operaciones definidas existen otras (producto de objetos y producto circular de objetos) que permiten variar la dimensión (como ocurre con las operaciones de barrido a las que modelizan). Ya que nuestro campo de estudio se centra en las dos representaciones CSG y B-rep, no tendremos en cuenta estas operaciones.

En CSG se considera como sistema de generadores el conjunto de primitivas que ofrecen la representación (usualmente paralelepípedos, cilindros, esferas, conos, toros, etc.). La disparidad de estos elementos implica que sea difícil hacer un tratamiento más o menos uniforme de todos los objetos. Un avance importante realizado en este campo es el trabajo de Shapiro [Shapi91] en su tesis doctoral, en la que propone la construcción de sólidos que pueden definirse por medio de semiespacios definidos por funciones algebraicas.

En B-rep, al definirse un objeto a partir de su frontera, no aparece claro el sentido de sistemas de generadores. Las operaciones de EULER, al ser unarias, no permiten una definición adecuada de sistemas de generadores. Un estudio muy completo en la generación de modelos B-rep es el realizado en la Universidad de la Sapienza (

[PauRS93]), trabajando con triangulaciones y tetraedrizaciones así como los trabajos de Dey ([Dey__91]) y Shaphidis ([ShaPe91]).

Es evidente la importancia de los símlices como elementos singulares dentro del modelado de sólidos. Las triangulaciones y tetraedrizaciones plantean problemas por la falta de un modelo uniforme para todo tipo de sólidos, así como en la implementación práctica en el caso de fronteras con agujeros y concavidades.

Vamos a trabajar, dentro del álgebra, con recubrimientos del objeto, lo que permite generarlo a partir de símlices, a la vez que se logra un tratamiento uniforme de todos los problemas de modelado. Lo aplicaremos a objetos gráficos cuyo volumen determine un sólido de acuerdo con la definición dada en el apartado 2.1. Así mismo supondremos que los objetos son uniformes, es decir, que tienen presencia y aspecto constante, y que el espacio de presencia es normal.

Así mismo supondremos que trabajamos con los sistemas de referencia usuales (XY para el plano y XYZ para el espacio).

En el plano, supondremos que los polígonos simples que determinan un sólido bidimensional se referencian por medio de sus aristas recorridas en el sentido contrario del reloj (*orientación positiva*). En el espacio, supondremos que los poliedros de caras planas simples que determinan un sólido tridimensional se referencian mediante una relación de las aristas de cada cara de modo que el vector normal de cada cara se dirija hacia fuera del poliedro. En ambos casos, diremos que el sólido tiene una *orientación conforme o positiva*

Haremos uso de las siguientes definiciones básicas [Kurat66]:

Definición 2.6 Un *d-simplex* es la clausura convexa de $d+1$ puntos linealmente independientes.

Un *d-simplex original* es un *d-simplex*, uno de cuyos puntos es el origen.

La clausura convexa de $d+1$ puntos, $P_0, P_1, P_2, \dots, P_d \in \mathbb{R}^n$ es el conjunto de todas sus combinaciones afines, es decir:

$$S = \{ P \mid P = \lambda_0 P_0 + \lambda_1 P_1 + \lambda_2 P_2 + \dots + \lambda_d P_d \text{ con } \lambda_0 + \lambda_1 + \lambda_2 + \dots + \lambda_d = 1, \lambda_i \geq 0 \}$$

Un conjunto de $d+1$ puntos $P_0, P_1, P_2, \dots, P_d \in \mathbb{R}^n$ son linealmente independientes si ninguno de ellos está contenidos en la clausura convexa de los d restantes.

Un 0-simplex es un punto, un 1-simplex es un segmento de recta, un 2-simplex es un triángulo y un 3-simplex es un tetraedro.

La frontera de un d -simplex es el conjunto de los $(d-j)$ -simplex, con $0 < j < d$, obtenidos como un subconjunto de los $d+1$ puntos que generan el d -simplex. Llamaremos k -cara de un d -simplex a cualquier k -simplex de su frontera. Para cualquier simplex, una 0-cara es un vértice y una 1-cara es una arista.

El orden en el cual aparecen los puntos no cambia la definición del simplex, pero puede cambiar el resultado de cálculos que trabajen con la secuencia de puntos, por lo que es imprescindible fijar un orden a la secuencia de puntos que definen el d -simplex.

Definición 2.7 Un *d -simplex orientado* es un d -simplex para el cual los puntos están dados en una sucesión específica ([Kurat66]).

Diremos que dos d -simplex orientados tienen la misma orientación si la secuencia de puntos de uno de ellos puede obtenerse de la del otro por medio de una permutación de orden par (es decir, una permutación con un número par de inversiones).

Las dos orientaciones de un 1-simplex orientado son AB y BA . Para un 2-simplex son ABC y ACB . Podemos expresar las dos orientaciones posibles de un 3-simplex fijando un punto; para un 3-simplex original, podemos fijar el origen y obtener las orientaciones $OABC$ y $OACB$.

Daremos las siguientes definiciones que nos permitirá calcular la orientación de símlices.

Definición 2.8 (función signo) Si x es un número real, la función signo se define como:

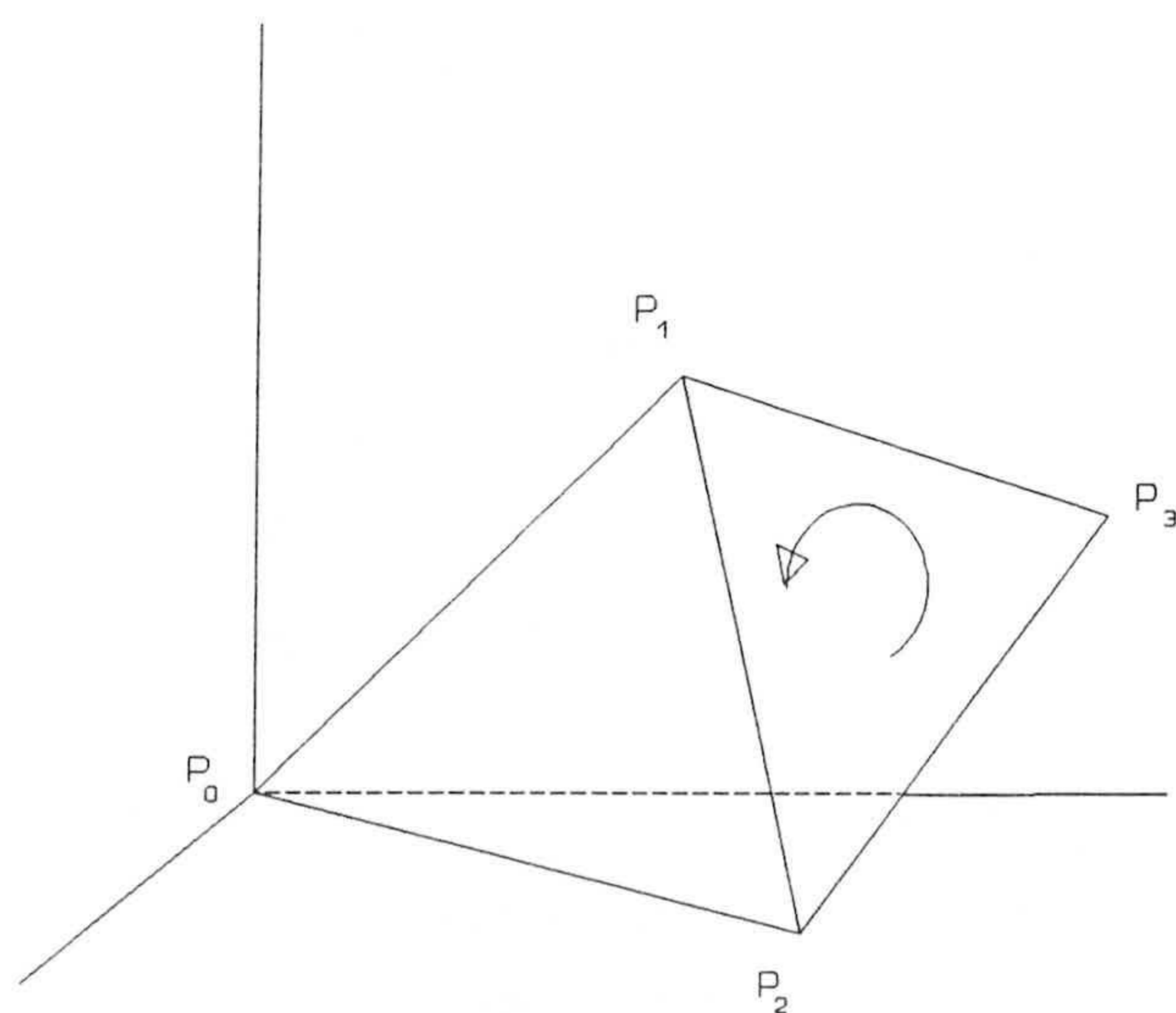
$$\text{sign}(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x = 0 \\ -1 & \text{si } x < 0 \end{cases}$$

Definición 2.9 (Area con signo de un triángulo) El área con signo de un triángulo definido por una triada de puntos ordenados $P_1P_2P_3$, (donde $P_i = (x_i, y_i)$, $i = 1, 2, 3$), viene dada por

$$[P_1P_2P_3] = \frac{1}{2} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

Ver [Berge87].

Definición 2.10 (volumen con signo de un tetraedro) El volumen con signo de un tetraedro definido por una cuaterna de puntos ordenados $P_0P_1P_2P_3$ (donde $P_i = (x_i, y_i, z_i)$) viene dado por la expresión



$$[P_0P_1P_2P_3] = \frac{1}{6} \begin{vmatrix} x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\ x_2 - x_0 & y_2 - y_0 & z_2 - z_0 \\ x_3 - x_0 & y_3 - y_0 & z_3 - z_0 \end{vmatrix}$$

Si $P_1P_2P_3$ se ve en sentido contrario del reloj desde el lado opuesto a P_0 , el signo será positivo, y negativo en caso contrario.

Ver [O'Rou94].

Figura 2.3 Tetraedro orientado

Definición 2.11(volumen con signo de una pirámide) El volumen con signo de una pirámide R , con cúspide V y base S , se denota por $[R]$ y coincide como el volumen de P si la pirámide tiene una orientación positiva y como menos el volumen de R si la orientación es negativa.

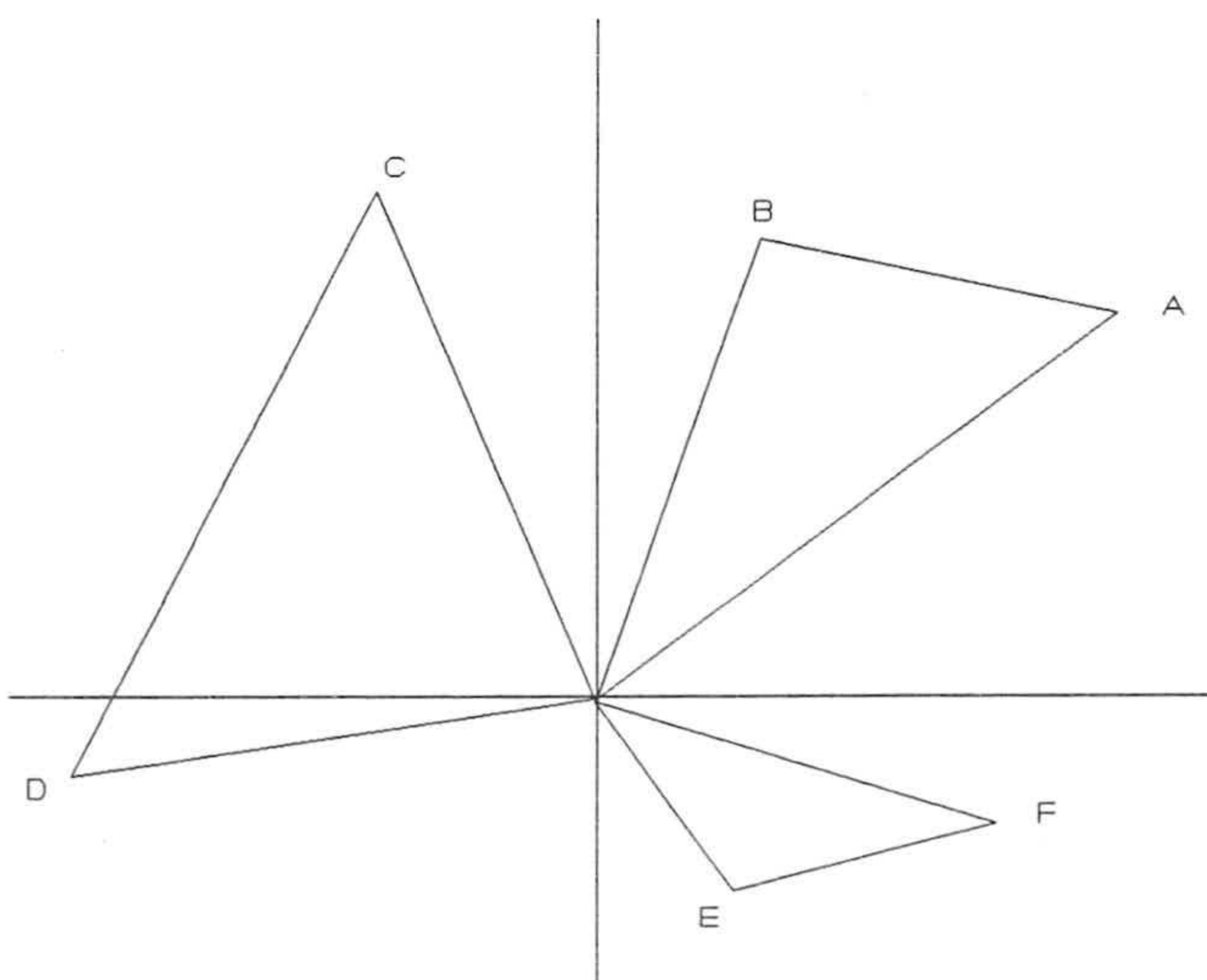
Si la base se recorre en sentido contrario del reloj desde el lado contrario del vértice, el volumen será positivo.

En una primera aproximación, es evidente que los objetos de presencia y aspecto constante e igual a 1 forman un sistemas de generadores de los objetos homogéneos, ya que por medio del producto escalar se pueden obtener el resto de los valores de presencia y aspecto. Este sistema de generadores es muy amplio y poco efectivo para el objetivo que buscamos. A pesar de esto, ese hecho nos permite trabajar con objetos de presencia y aspecto 1.

Vamos a enunciar a continuación los teoremas que nos permiten construir cualquier objeto gráfico a partir de elementos singulares del álgebra. Previamente definiremos los objetos gráficos que intervendrán:

Definición 2.12 (2-simplex) Dados tres puntos arbitrarios definimos el objeto

gráfico 2-simplex como aquel cuyo volumen es el 2-simplex determinado por los tres puntos arbitrarios y cuyos valores de aspecto y de presencia son 1 para todos los puntos del volumen.



En el caso de que alguno de los tres puntos coincida con el origen, el objeto gráfico se denominará 2-simplex ó triángulo original.

Las aristas a las que pertenece el origen las denominaremos originales y a la arista restante, no original.

Figura 2.4 Triángulos originales

En la figura 2.4 pueden verse algunos ejemplos de triángulos originales.

Definición 2.13(pirámide) Dado un punto arbitrario en el espacio \mathbf{R} y un

polígono definimos el objeto gráfico pirámide de base C y cúspide \mathbf{R} como aquel cuyo volumen es la pirámide y cuyos valores de aspecto y presencia son 1 para todos los puntos del volumen.

En el caso de que \mathbf{R} coincida con el origen, el objeto gráfico se denominará pirámide original. Denominaremos aristas y caras originales a aquellas que tienen un vértice en el origen, y no originales al resto.

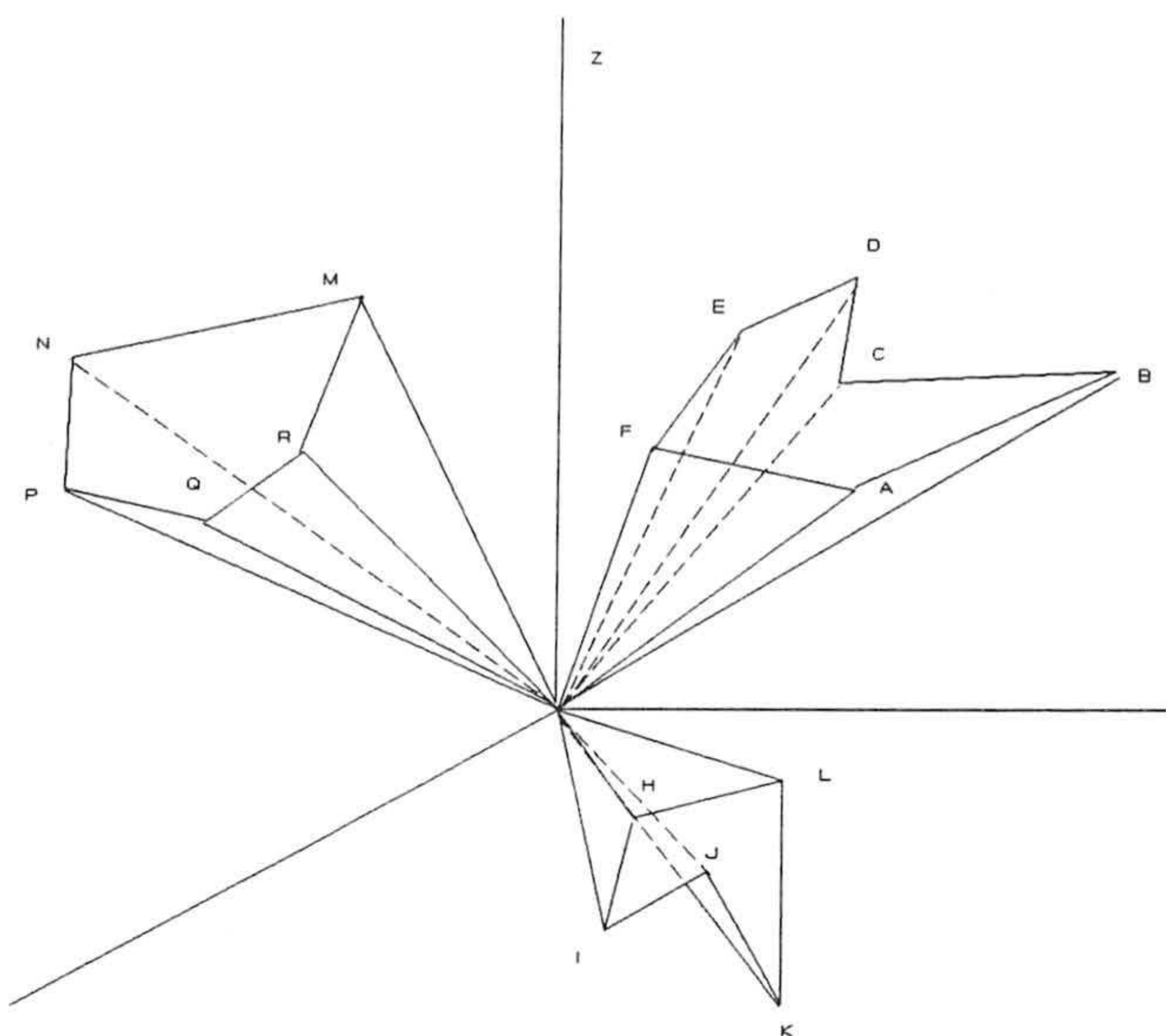


Figura 2.5 Pirámides originales

En la figura 2.5 pueden verse algunos ejemplos de pirámides originales.

Definición 2.14 (tetraedro) Dados cuatro puntos arbitrarios en el espacio definimos el objeto gráfico 3-simplex como aquel cuyo volumen es el 3-simplex determinado por los cuatro puntos y cuyos valores de aspecto y presencia son 1 para todos los puntos del volumen.

En el caso de que alguno de los cuatro puntos coincida con el origen el objeto gráfico se denominará 3-simplex ó tetraedro original.

Denominaremos aristas y caras originales a aquellas que tienen un vértice en el origen, y no originales al resto.

Enunciaremos a continuación los teoremas fundamentales de esta sección: vamos a demostrar que todo objeto se puede obtener a partir de símlices originales (triángulos originales en el caso de dos dimensiones y pirámides o tetraedros originales en el caso de tres dimensiones).

Teorema 2.2 Dado un punto fijo y arbitrario, R , todo objeto gráfico bidimensional O , de aspecto y presencia 1, definido por la secuencia de aristas $\{T_i, i=1..k\}$ (en una orientación conforme), con $T_i=(Q_i, Q_{i+1})$, puede ser representado como una expresión de objetos gráficos del modo siguiente:

$$O = \sum_{i=1}^k \text{sig}([S_i]) * S_i$$

donde S_i es el objeto gráfico 2-simplex (triángulo) determinado por los puntos RQ_iQ_{i+1} .

Demostración:

Al objeto de demostrar el teorema podemos considerar tres tipos de puntos

a) Puntos A que están en el interior del sólido y en ninguna frontera de los S_i . Habrá que demostrar que estos puntos tienen presencia y aspecto 1 en el objeto determinado por el sumatorio indicado en el teorema.

Consideremos una semirrecta que parte de R y pasa por A . Ya que A no está en la frontera de ningún S_i , la semirrecta no puede pasar por ningún vértice del sólido. Por ser el punto A interior al sólido, el número de intersecciones de la semirrecta entre A e infinito con las aristas del sólido es un número impar ([PreSh85]). Cada intersección implica que A pertenece al objeto gráfico 2-simplex determinado por R y la arista de intersección. Cada intersección supone una transición del interior del sólido al exterior ó viceversa. Por ser el punto interior, la primera intersección implica una transición de dentro hacia afuera del sólido, y el resto de intersecciones se van alternando. Si n es el número de intersecciones que implican una transición del interior al exterior del sólido, habrá $n-1$ intersecciones que impliquen una transición del exterior al interior del sólido.

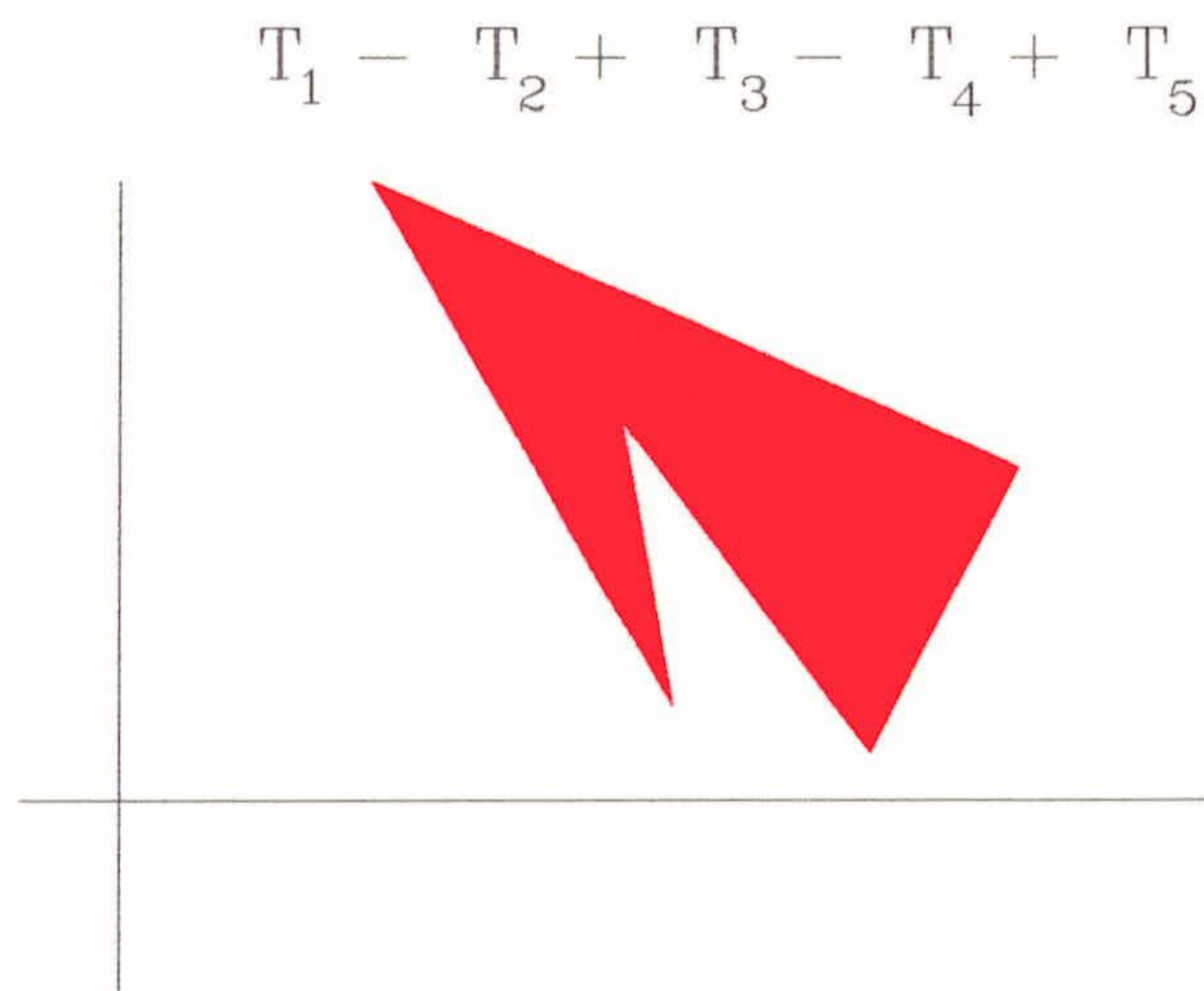
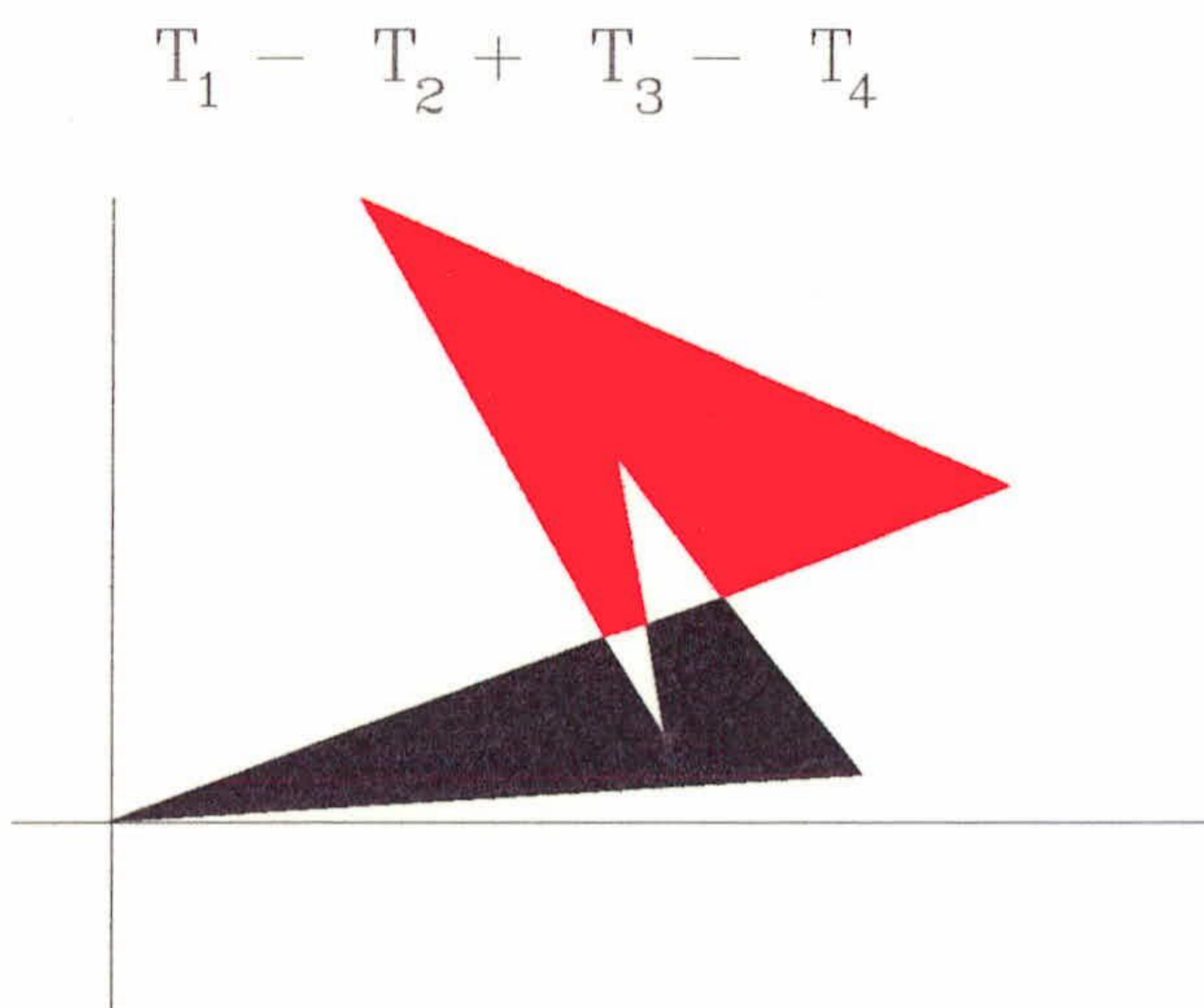
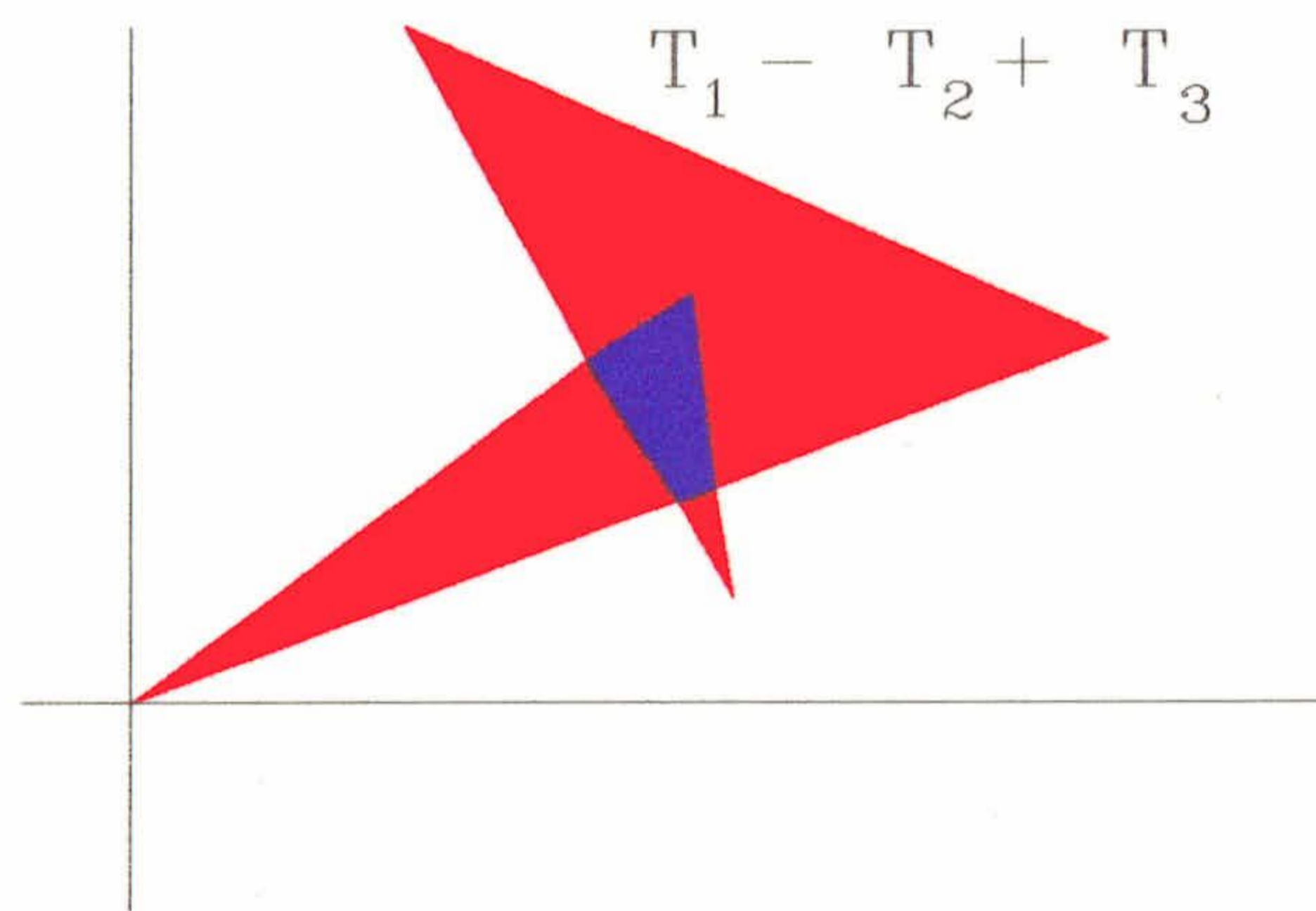
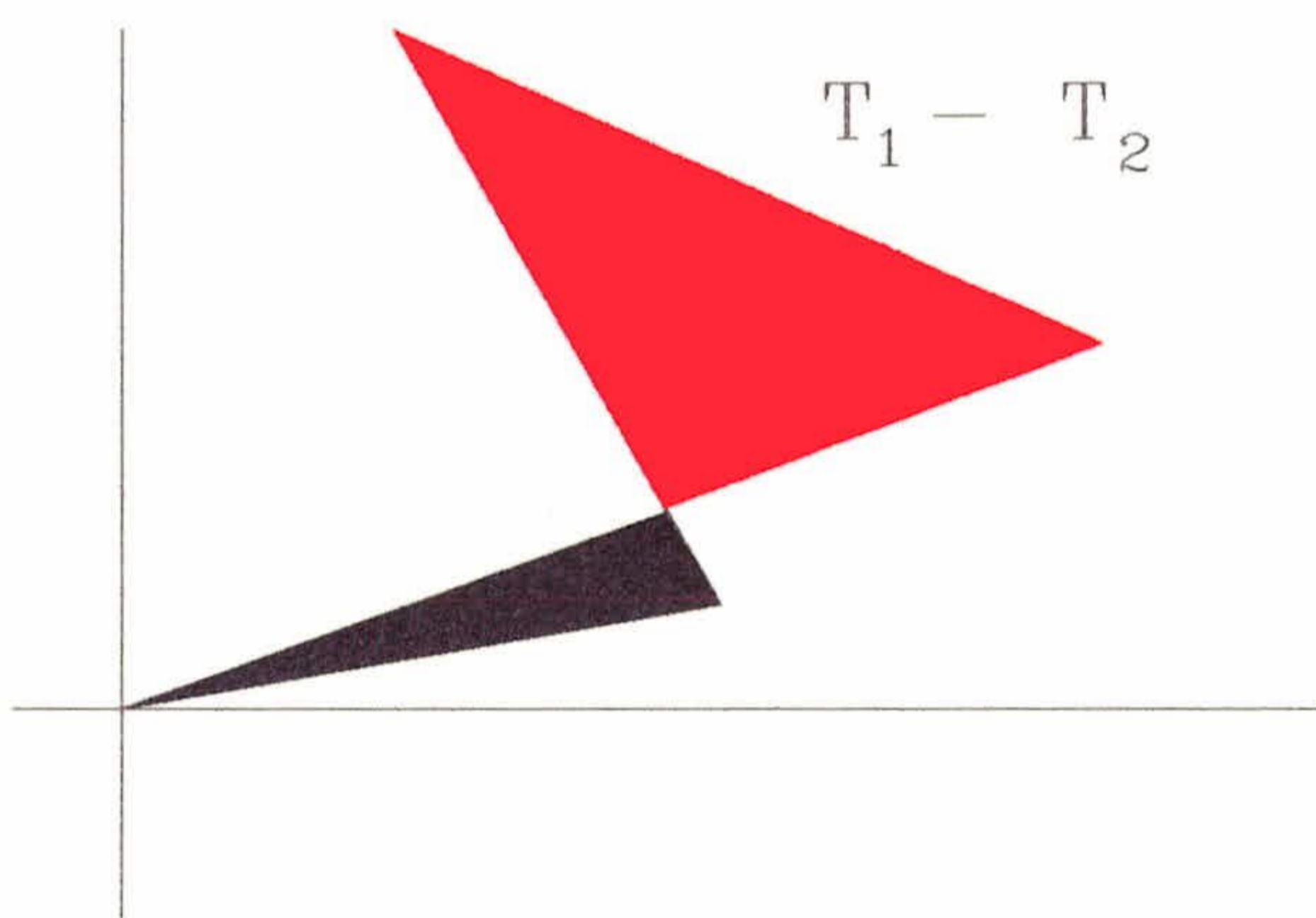
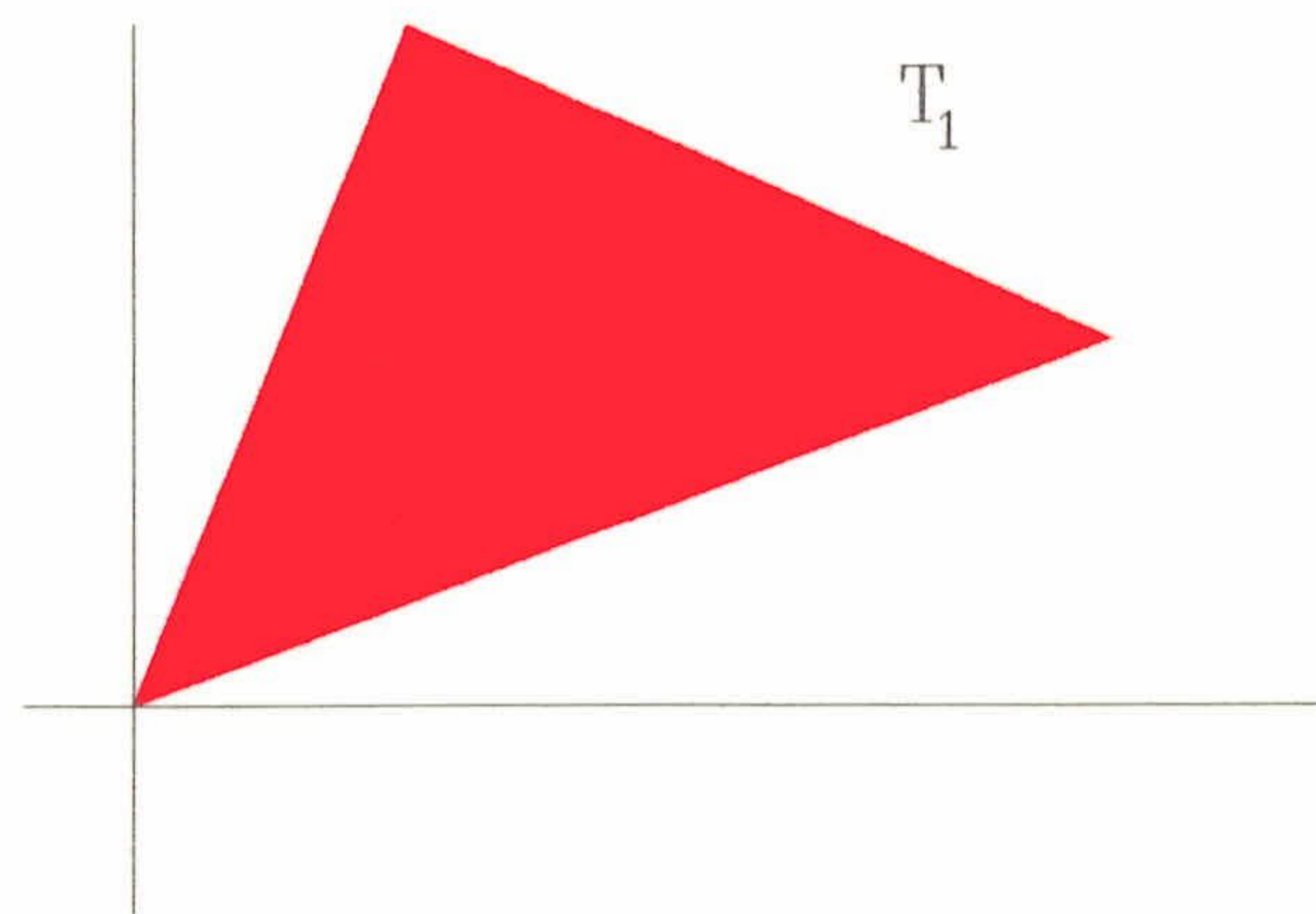


Figura 2.6 Descomposición de un objeto en símplices originales
Las operaciones son regularizadas



Cuando en un punto de intersección entre la semirrecta y la arista T_i existe una transición del interior hacia el exterior quiere decir que el punto R está situado a la izquierda de la arista T_i , recorrida esta de Q_i a Q_{i+1} , lo que implica que el triángulo RQ_iQ_{i+1} está orientado positivamente y que su signo es positivo. El punto A pertenecerá, por tanto, a n 2-símplices de signo positivo y a $n-1$ 2-símplices de signo negativo, por lo que su presencia y aspecto será: $n+n-1=1$.

b) Puntos B que están en el exterior del sólido y en ninguna frontera de los S_i . Habrá que demostrar que estos puntos tienen presencia y aspecto 0 en el objeto determinado por el sumatorio indicado en el teorema.

Sea R la semirrecta que parte de R , pasa por B y se dirige al infinito. Dicha semirrecta tendrá a partir de B un número par de intersecciones con las aristas del sólido O . Por un razonamiento similar al realizado en el apartado a), el punto B pertenecerá a n 2-símplices de signo positivo y a n 2-símplices de signo negativo, por lo que su presencia y aspecto será: $n-n=0$

c) Puntos que están en la frontera de al menos un S_i

Todos estos puntos C serán irregulares. Por ser regularizadas las operaciones que se está usando, a estos puntos se les asignará un valor de presencia y aspecto en función de los valores de aspecto y presencia en un entorno de C tan pequeño como se quiera. Si C es interior ó frontera al objeto, se le asignará presencia y aspecto 1. Si C es exterior se le asignará presencia y aspecto 0. ■

Podemos ver un ejemplo de la aplicación de este teorema en la figura 2.6

El siguiente teorema generaliza el resultado anterior a tres dimensiones, considerando las pirámides originales como sistema de generadores.

Teorema 2.3 Dado un punto fijo y arbitrario, \mathbf{R} , todo objeto gráfico tridimensional O , de aspecto y presencia 1, limitado por el conjunto de caras $\{C_i, i=1..k\}$, dadas en una orientación conforme, puede ser expresado como

$$O = \sum_{i=1}^k \text{sig}([P_i]) * P_i$$

donde P_i es el objeto gráfico pirámide con base C_i y vértice en el punto de referencia \mathbf{R} con valor de presencia y aspecto 1.

Demostración:

Se puede realizar siguiendo el mismo esquema usado para el teorema 2.2. ■

El siguiente teorema nos permite usar tetraedros como elementos generadores, por lo que las expresiones obtenidas son más uniformes.

Teorema 2.4 Dado un punto fijo y arbitrario \mathbf{R} , todo objeto gráfico tridimensional de aspecto y presencia 1, limitado por el conjunto de caras $\{C_i, i=1..k\}$, con aristas $C_i = \{T_{j,i}, j=1..l\}$, dadas en una orientación conforme, por $T_{j,i} = (\mathbf{Q}_{j,1,i}, \mathbf{Q}_{j,2,i})$, puede ser expresado como

$$A = \sum_{i=1}^k \left(\sum_{j=2}^{l-1} \text{sig}(S_{ij}) * S_{ij} \right)$$

donde S_{ij} es el objeto gráfico 3-simplex (tetraedro) $\mathbf{RQ}_{1,1,i}\mathbf{Q}_{j,1,i}\mathbf{Q}_{j,2,i}$, definido con valor de presencia y aspecto 1.

Demostración

El teorema se demuestra usando el mismo esquema utilizado en el teorema 2.2. ■

Los teoremas anteriores establecen una representación, basada en el álgebra gráfica, para todo sólido bi o tridimensional, usando elementos muy sencillos. En la figura 2.7 puede observarse un ejemplo en dos dimensiones.

El hecho de que la representación dependa de un punto de referencia, no es ninguna restricción, ya que es similar al problema que ocurre con toda representación en coordenadas. Habitualmente podremos trabajar con símlices originales, aunque en otros casos será más conveniente elegir otro punto.

En el caso de 3 dimensiones, al hacer la descomposición en tetraedros, hemos elegido como punto de referencia de cada cara el primer vértice (en el orden en el que aparecen al describir su frontera). La elección de este punto es arbitraria, al igual que la del punto \mathbf{R} en dos dimensiones, siempre que esté contenido en el mismo plano que la cara, ya que lo que se está haciendo es aplicar el teorema 2.2 a cada una de las caras.

Trabajar con pirámides como la base de los objetos gráficos, en vez de elegir los símlices, reduce en gran manera el número de componentes de un sólido, aunque se pierde la uniformidad y sencillez de las componentes.

Hemos indicado anteriormente que en el ambiente del álgebra gráfica podemos considerar como expresión CSG es toda aquella expresión formada por operaciones entre objetos gráficos de la misma dimensión.

Hemos indicado en el capítulo I que una representación se podría entender como una aplicación entre el conjunto de sólidos y el conjunto de elementos susceptibles de ser representados dentro de un esquema concreto. Si consideramos como sólidos todos los poliedros de caras planas es conocido que se puede representar adecuadamente por medio de su frontera.

En el contexto del álgebra gráfica, hemos demostrado que todo sólido (según la definición introducida en el apartado 2.1) se puede expresar a partir de un conjunto de primitivas por medio de operaciones del álgebra (lo que correspondería a una representación tipo CSG). Por otro lado, existe una aplicación uno a uno entre los elementos generadores y las aristas del poliedro (que representan exactamente las fronteras del sólido).

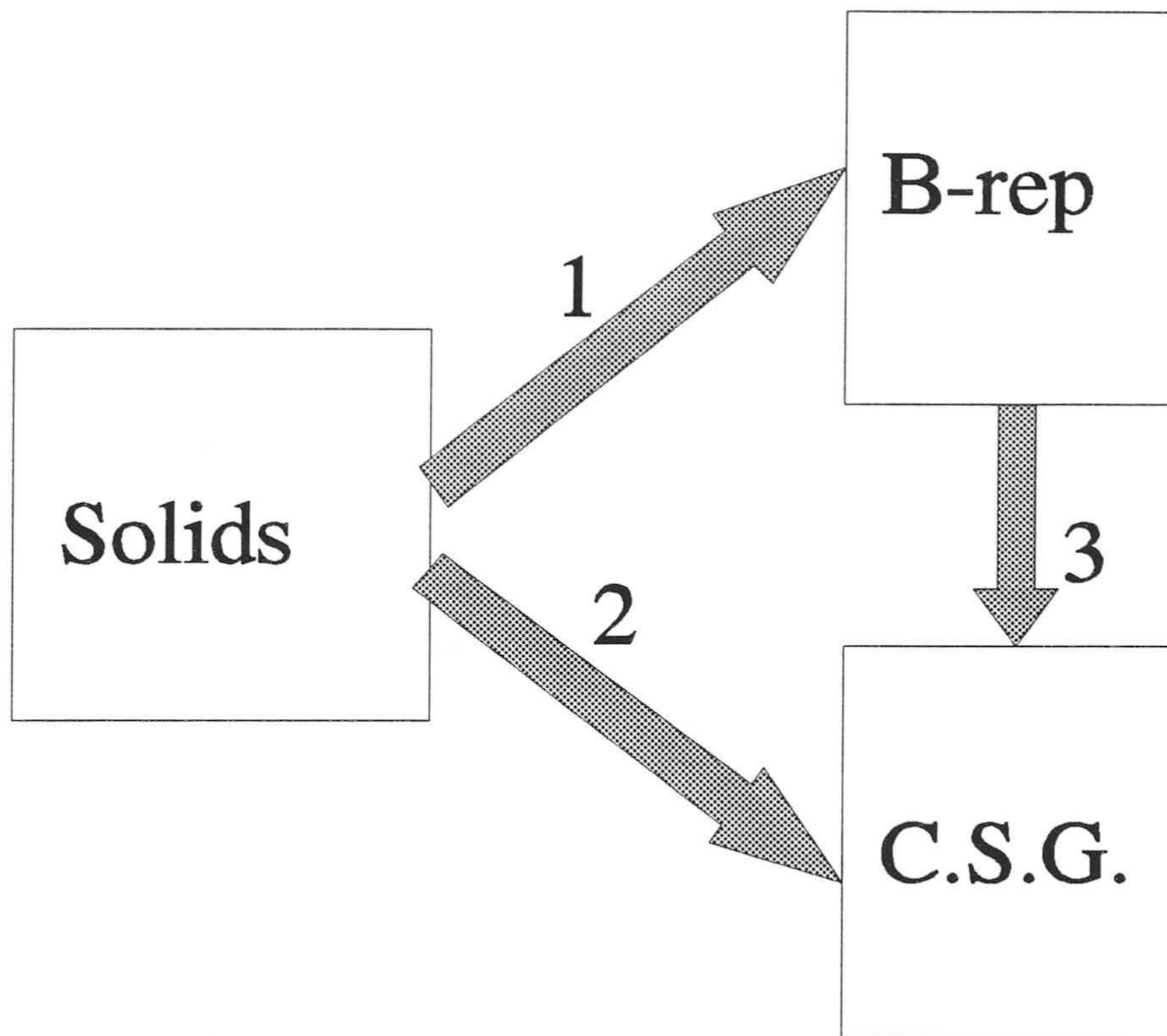


Figura 2.7 Conversión formal de B-rep a CSG

Los teoremas anteriores ofrecen una demostración formal, en el contexto del álgebra gráfica, de la conversión de representaciones B-rep a CSG, en un modo más sencillo que el indicado por Shapiro [Shavo92] (aunque el método de Shapiro es válido tanto para sólidos modelados por caras planas como para algunos tipos de sólidos modelados por caras no planas). Quedaría abierto el problema de extender la representación a sólidos modelados directamente con caras no planas (aunque una solución válida sería la aproximación por medio de caras planas).

2.5 Expresiones canónicas

Una vez obtenida la expresión de cualquier sólido en función de elementos singulares del álgebra gráfica parece lógico plantearse cómo se comportan las operaciones regularizadas en relación a esta representación.

Así mismo será necesario estudiar la relación entre las propiedades que determinan la estructura de espacio vectorial y aquellas que determinan la estructura de álgebra de Boole, es decir, la relación entre la suma y la unión e intersección.

Vamos a estudiar en este apartado cómo se comportan las descomposiciones efectuadas en relación con las operaciones definidas anteriormente.

Ya que el objetivo de nuestro trabajo no es directamente la visualización, prescindiremos de todo lo que hace referencia al aspecto de los sólidos (por similitud con el espacio de presencia, se podrían generalizar muchos de los resultados aquí obtenidos).

En lo que sigue supondremos que hemos elegido como punto de referencia el origen. En el caso de tres dimensiones elegimos como referencia de cada cara el primer punto que define la misma tal y como hemos hecho en el teorema 2.4). Esta presunción se mantendrá a lo largo de todo este apartado.

En primer lugar, caracterizaremos mediante una definición la representación de sólidos a partir de las descomposiciones realizadas mediante los teoremas 2.2 y 2.4.

Definición 2.15 Dado un objeto O , se denomina *representación canónica* del objeto a la expresión de O en función de sus símlices originales asociados.

Si el objeto no es de presencia constante, se podrá expresar como suma de componentes regulares, y cada componente regular se podrá expresar por medio de sus representación canónica.

Si el objeto es de presencia constante es fácil comprobar que la expresión canónica es la expresión con un menor número de elementos, dentro del sistema de generadores con el que estamos trabajando (símlices originales).

Lema 2.2 Sea O un objeto de presencia constante, y E_0 su expresión canónica. Entonces toda expresión E' de O en función de símlices originales tiene más elementos que E_0 .

Demostración:

Supongamos que E' tiene menos elementos E_0 . Entonces existirá al menos un simplex original S de E_0 que no está en E' . Existirá, por tanto, un vértice V que no es vértice de ninguno de los símlices de E' . Dicho vértice tiene que estar dentro de E' (ya que E' es una expresión de O). V estará dentro de un simplex S' de E' y por tanto alguno de los vértices de S' estará fuera de O , lo que contradice al hecho de que E' es una expresión de O . ■

La suma regularizada de objetos y el producto por escalares son operaciones inmediatas a partir de sus expresiones canónicas debido a la estructura de espacio vectorial. Sólo cabría estudiar las simplificaciones de expresiones dadas. No son inmediatas las operaciones del álgebra de Boole entre objetos, por lo que es necesario hacer un estudio más detallado.

Lema 2.3 Sea O_1 y O_2 dos objetos de presencia constante p_1 y p_2 . Entonces:

$$O_1 \cap^* O_2 = (p_1 \cap p_2)^{**} (O'_1 \cap^* O'_2)$$

donde O' indica el mismo volumen del objeto O , pero con presencia 1.

Demostración

Es obvio que $O'_1 \cap^* O'_2$ da como resultado un objeto con volumen igual a

$$\text{vol}(O_1) \cap^* \text{vol}(O_2)$$

y con presencia 1, por tratarse de dos objetos homogéneos con presencia igual a 1. Al multiplicar este objeto por $(p_1 \cap p_2)$ su presencia será $p_1 \cap p_2$ en todo su volumen, y por tanto se tiene la igualdad. ■

Bastará demostrarlo para todo punto que pertenece al interior de la intersección conjuntista del volumen de 1 y del volumen de 2, ya que el resto de puntos son irregulares y al regularizar la operación su presencia vendrá dada en función de los valores en un entorno suficientemente pequeño.

A partir del lema anterior veamos como realizar la intersección entre objetos a partir de los elementos de las representaciones canónicas:

Teorema 2.5 Sea O_1 y O_2 dos objetos de presencia 1 y $\{T_i\}$ y $\{T'_j\}$ sus componentes canónicas. Entonces:

$$O_1 \cap^* O_2 = \sum_i^* (\sum_j^* (sig([T_i]) * sig([T'_j])) * (T_i \cap^* T'_j))$$

Demostración

Demostraremos la doble inclusión.

Bastará demostrarlo para todo punto regular del interior de la intersección conjuntista del volumen del objeto 1 y del objeto 2, ya que el resto de puntos son irregulares y al regularizar la operación su presencia vendrá dada en función de los valores de presencia de los puntos regulares de un entorno lo suficientemente pequeño.

Probaremos que todo punto regular del conjunto determinado por el primer miembro tiene la misma presencia que en el conjunto determinado por el segundo miembro.

Sea P un punto regular de $O_1 \cap^* O_2$ con presencia 1. El punto debe estar en O_1 y en O_2 . Si q es el número de símlices negativos asociados a O_1 que contienen a P , habrá $q+1$ símlices positivos de O_1 que contienen a P . Del mismo modo si hay m símlices negativos de la descomposición de O_2 que

contienen a P habrá $m+1$ símlices positivos de O_2 que lo contienen. Al operar la expresión de la derecha y calcular la presencia en todas las intersecciones posibles de pares de símlices de O_1 y O_2 tendremos cuatro posibilidades, que son:

- T_i y T'_j son ambos positivos. Habrá $(q+1)*(m+1)$ pares para este caso.
- T_i es positivo y T'_j es negativo. Habrá $(q+1)*m$ pares
- T_i es negativo y T'_j es positivo. Habrá $q*(m+1)$ pares
- T_i y T'_j son ambos negativos. Habrá $q*m$ pares

El sumatorio total para P da:

$$q*m + q + m + 1 - q*m - m - q*m - q + q*m = 1$$

por lo que el valor de la presencia de P es la misma en los dos miembros.

Demostremos la otra inclusión por reducción al absurdo demostrando que los puntos regulares de los símlices asociados a O_1 y a O_2 , exteriores a $O_1 \cap O_2$ son también exteriores al sumatorio, procediendo para puntos que están sólo en O_1 o sólo en O_2 ó fuera de los dos.

Si P es un punto regular de los símlices asociados a O_1 y a O_2 , que está en O_1 pero no en O_2 , habrá $q+1$ símlices positivos de O_1 en los que estará P y q símlices negativos en los que estará. Así mismo habrá m símlices positivos de O_2 en los que estará y m símlices negativos en los que estará. Al operar la expresión de la derecha y calcular la presencia en todas las intersecciones posibles de pares de símlices de O_1 y O_2 tendremos cuatro posibilidades, que son:

- T_i y T'_j son ambos positivos. Habrá $(q+1)*m$ pares para este caso.
- T_i es positivo y T'_j es negativo. Habrá $(q+1)*m$ pares
- T_i es negativo y T'_j es positivo. Habrá $q*m$ pares
- T_i y T'_j son ambos negativos. Habrá $q*m$ pares

El sumatorio total para P da

$$q*m + m - q*m - m - q*m + q*m = 0$$

por lo que el valor de la presencia de P es la misma en los dos miembros.

Del mismo modo se ve que si P es un punto regular de los símlices asociados a O_1 y a O_2 , que está en O_2 pero no en O_1 , el valor de la presencia de P es la misma en los dos miembros.

Si P es un punto regular de los símlices asociados a O_1 y a O_2 , que no está ni en O_1 ni en O_2 , habrá q símlices positivos de O_1 en los que estará P y q símlices negativos en los que estará. Así mismo habrá m símlices positivos de O_2 en los que estará y m símlices negativos en los que estará. Al operar la expresión de la derecha y calcular la presencia en todas las intersecciones posibles de pares de símlices de O_1 y O_2 tendremos cuatro posibilidades, que son:

- T_i y T'_j son ambos positivos. Habrá $q \cdot m$ pares para este caso.
- T_i es positivo y T'_j es negativo. Habrá $q \cdot m$ pares
- T_i es negativo y T'_j es positivo. Habrá $q \cdot m$ pares
- T_i y T'_j son ambos negativos. Habrá $q \cdot m$ pares

El sumatorio total para P da

$$q \cdot m - q \cdot m - q \cdot m + q \cdot m = 0$$

por lo que el valor de la presencia de P es la misma en los dos miembros. ■

Ya que en la unión de dos objetos hay puntos que corresponden sólo al volumen de ellos, no es posible un lema similar al 2.3 aplicable a la unión. Sí se cumple una propiedad similar a la del teorema 2.5 cuando los dos objetos coinciden en sus valores de presencia.

Lema 2.4 Sean O_1 y O_2 dos objetos de presencia constante p . Entonces:

$$O_1 \cup^* O_2 = p \cdot (O'_1 \cup^* O'_2)$$

donde O' indica el mismo volumen del objeto O , pero con presencia 1 en cada punto.

Demostración:

Es obvio que $O'_1 \cup^* O'_2$ da como resultado un objeto con volumen igual a

$$\text{vol}(O_1) \cup^* \text{vol}(O_2)$$

y con presencia 1, por tratarse de dos objetos homogéneos con presencia igual a 1. Al multiplicar este objeto por p su presencia será p en todo su volumen, y por tanto se tiene la igualdad. ■

Finalmente enunciaremos un lema correspondiente a la complementación

Lema 2.6 Sea O_1 un objeto de presencia constante p . Entonces:

$$\sim O_1 = (\sim p)^*(\sim O'_1)$$

donde O' indica el mismo volumen del objeto O_1 , pero con presencia 1 en cada punto.

Demostración:

Demostraremos la doble inclusión, solo para puntos regulares tal y como hemos hecho anteriormente.

Si P es un punto interior del complementario de O_1 tendrá por presencia $\sim p$, la misma que tiene en el segundo miembro.

Demostraremos la otra inclusión por reducción al absurdo:

Si P es exterior al complementario, ya que O' tiene el mismo volumen de O_1 también será exterior a O' y por tanto tendrá presencia 0 también en el segundo miembro ($\sim p * 0 = 0$), por lo que no pertenecerá al conjunto determinado por el segundo miembro. ■

Mediante los últimos teoremas podemos realizar casi todas las operaciones booleanas con objetos por medio de sus representantes canónicos. Debido a la sencillez de los elementos que forman las representaciones canónicas, es relativamente fácil la realización de casi todas las operaciones booleanas. Dos son los aspectos que quedan pendientes: la realización de la unión de objetos y la simplificación de expresiones canónicas. Teniendo en cuenta que el conjunto \mathbb{R}^n no es propiamente dicho un sólido, no lo podemos representar a partir de símplices, por lo que no es posible representar totalmente los complementarios por medio de símplices asociados. Una posible solución es la que adoptaremos en el apartado 4.1, en el que definiremos unos conjuntos universales (la caja máxima bidimensional y el cubo máximo tridimensional) que a efectos prácticos se comportarán como el conjunto total \mathbb{R}^n .

Para expresar la unión de objetos de presencia 1 en función de las representaciones canónicas de los objetos enunciaremos y demostraremos un teorema nos permitirá relacionar las operaciones del espacio vectorial con las operaciones del álgebra booleana, siempre que dicha relación esté previamente establecida en los espacios de presencia y aspecto (tal y como ocurre en los ejemplos indicados en la sección inicial). Para ello daremos la siguiente definición.

Definición 2.16 Un espacio de aspecto, δ , ó un espacio de presencia, π , se dice que es *uniforme* si para todos sus valores se verifica que:

$$x \cup y = x + y - (x \cap y)$$

Los ejemplos que aparecen en la sección 2.2 satisfacen esta condición.

Lema 2.7 La ζ -estructura definida en la sección 2.2 es uniforme

Demostración

Habría que demostrar que para todo x, y de Z , se verifica que:

$$x \cup y = x + y - (x \cap y)$$

Supongamos que $x = a_n a_{n-1} \dots a_1 a_0$ e $y = b_n b_{n-1} \dots b_1 b_0$ son la representación binaria de x e y .

Habría que demostrar que :

$$2^{n-1}(a_{n-1} \cup b_{n-1}) + 2^{n-2}(a_{n-2} \cup b_{n-2}) + \dots + 2(a_1 \cup b_1) + (a_0 \cup b_0) =$$

$$2^{n-1}(a_{n-1} + b_{n-1}) + 2^{n-2}(a_{n-2} + b_{n-2}) + \dots + 2(a_1 + b_1) + (a_0 + b_0) -$$

$$- (2^{n-1}(a_{n-1} \cap b_{n-1}) + 2^{n-2}(a_{n-2} \cap b_{n-2}) + \dots + 2(a_1 \cap b_1) + (a_0 \cap b_0))$$

Ello equivale a demostrar que para todo i :

$$2^i(a_i \cup b_i) = 2^i(a_i + b_i) - 2^i(a_i \cap b_i)$$

Se pueden dar tres casos:

Si $a_i = b_i = 0$, se cumple el lema ya que $0 \cup 0 = 0 + 0 - (0 \cap 0)$

Si $a_i \neq b_i$, se cumple el lema ya que $1 \cup 0 = 1 + 0 - (0 \cap 1)$ y también

$$0 \cup 1 = 0 + 1 - (1 \cap 0)$$

Si $a_i = b_i = 1$, se cumple el lema ya que $1 \cup 1 = 1 + 1 - (1 \cap 1)$. ■

El lema anterior nos asegura la existencia de al menos una ζ -estructura que es uniforme por lo que es posible plantear el siguiente teorema.

Teorema 2.6 Sean $O_1=(\mu_1,\alpha_1)$ y $O_2=(\mu_2,\alpha_2)$ dos objetos gráficos de presencia y aspecto constantes. Entonces si los espacios de presencia y aspecto sobre los que están definidos son uniformes:

$$O_1 \cup^* O_2 = O_1 +^* O_2 -^* (O_1 \cap^* O_2)$$

Demostración

Demostraremos la doble inclusión.

Por el mismo argumento utilizado en la demostración del teorema 2.5, bastará demostrar la doble inclusión sólo para puntos regulares.

Sea P es un punto regular de $O_1 \cup O_2$. Si el punto pertenece sólo a uno de los dos objetos, dicho punto no estará en la intersección y por tanto el aspecto y la presencia serán sólo las de un objeto.

Si el punto está en los dos objetos O_1 y O_2 , estará también en la intersección. Por ser los espacios de presencia y aspecto uniformes, para cualquier punto P se verifica:

$$\mu(P) = \mu_1(P) \cup \mu_2(P) = \mu_1(P) + \mu_2(P) - (\mu_1(P) \cap \mu_2(P))$$

y por tanto la función de presencia de P en el segundo miembro es la misma.

Para el espacio de aspecto tendríamos la misma igualdad, es decir:

$$\alpha(P) = \alpha_1(P) \cup \alpha_2(P) = \alpha_1(P) + \alpha_2(P) - (\alpha_1(P) \cap \alpha_2(P))$$

y por tanto la función aspecto en P es igual a la del segundo miembro.

Demostraremos la inclusión del segundo miembro en el primero por reducción al absurdo.

Supongamos que P es un punto regular del objeto determinado por el segundo miembro que no está en el objeto determinado por el primer miembro. Por tanto P no estará ni en O_1 ni en O_2 . Ello implica que tampoco está en la intersección y por tanto la presencia y el aspecto de P en el segundo miembro serían:

$$\mu_1(P) + \mu_2(P) - (\mu_1(P) \cap \mu_2(P)) = 0 + 0 - 0$$

$$\alpha_1(P) + \alpha_2(P) - (\alpha_1(P) \cap \alpha_2(P)) = 0 + 0 - 0$$

y ello contradice a que P es un punto regular del objeto definido en el segundo miembro del teorema. ■

Un corolario interesante, consecuencia del teorema que acabamos de demostrar es que si dos objetos tienen intersección vacía su unión coincide con su suma.

Corolario 2.1 Sean O_1 y O_2 dos objetos gráficos con presencia y aspecto constantes. Entonces si su intersección es vacía:

$$O_1 \cup^* O_2 = O_1 +^* O_2$$

Demostración:

Basta considerar en la igualdad del teorema 2.6 que la intersección es vacía y por tanto

$$O_1 \cup^* O_2 = O_1 +^* O_2 -^* (O_1 \cap^* O_2) = O_1 +^* O_2. \quad \blacksquare$$

El último teorema y su corolario nos relacionan la unión con la suma e intersección, lo que nos permite realizar operaciones de objetos por medio de sus componentes canónicas.

A lo largo de este capítulo hemos definido las operaciones regularizadas en el álgebra gráfica, demostrando que se mantiene la misma estructura que para las operaciones sin regularizar.

Hemos demostrado que todo objeto gráfico puede expresarse por medio de símlices originales. Estos símlices, para cada objeto, vienen determinados unívocamente por su frontera, lo que permite el tratamiento de las representaciones B-rep en el álgebra gráfica.

Finalmente hemos demostrado que las operaciones entre objetos se pueden realizar a partir de las representaciones canónicas de cada uno de ellos.

CAPITULO III

Algoritmos de inclusión en sólidos

En este capítulo se desarrolla una de las posibles aplicaciones del formalismo expuesto en el capítulo anterior: la obtención de algoritmos de inclusión en sólidos.

Los algoritmos de inclusión son esenciales en cualquier modelador de sólidos, ya que son imprescindibles para verificar la consistencia de los modelos así como para la realización de operaciones entre sólidos.

Se propondrán algoritmos de inclusión en dos y tres dimensiones. Los algoritmos se basan en el hecho de que todo sólido se puede expresar como una combinación de símplexes y , por tanto, para decidir si un punto está ó no en un sólido bastará comprobar si está o no en cada uno de los símplexes que lo generan y realizar la suma con el signo adecuado. Las funciones necesarias para comprobar si un punto está o no en un simplex son muy sencillas de implementar y evitan la resolución de sistemas de ecuaciones que implican la realización de cocientes entre números reales, cuyo uso produce problemas de estabilidad en los algoritmos.

Los algoritmos serán aplicables a sólidos en el sentido de la definición dada en el apartado 2.1.

Veremos en primer lugar el algoritmo para dos dimensiones y posteriormente el de tres dimensiones.

3.1 Algoritmo de inclusión para sólidos bidimensionales

El algoritmo de inclusión en dos dimensiones ha sido ampliamente estudiado [PreSh85], siendo varias las soluciones aportadas para su resolución.

En nuestro caso podemos considerar un sólido bidimensional como un conjunto de presencia unidad, que puede ser expresado por medio de símlices originales de dimensión 2 (triángulos originales), cada uno de ellos definido por los dos vértices de cada una de las aristas y el origen de coordenadas.

Tal y como hemos visto en el capítulo anterior es necesario realizar la suma regularizada entre símlices originales. Dicha suma difiere de la suma normal en los puntos irregulares; esos puntos son los puntos frontera de cada símlice.

Al objeto de poder implementar la suma regularizada introducimos la siguiente definición.

Definición 3.1 (triángulo original incompleto) Dados dos puntos en \mathbb{R}^2 , P_1 y P_2 (de modo que O , P_1 y P_2 está orientado en sentido contrario a las agujas del reloj), definimos el triángulo original incompleto como el conjunto de puntos del triángulo OP_1P_2 , incluyendo la frontera y excluyendo la arista P_2O . Si los tres puntos son colineales el triángulo original incompleto degenerado es la arista P_1P_2 , excluyendo el punto P_2 .

$$T = \text{int}(OP_1P_2) \cup (P_1P_2 - \{P_2\}) \cup OP_1$$

Los elementos anteriores (triángulos originales no completos) van a ser los que nos van a generar, sin necesidad de operaciones regularizadas, el sólido bidimensional, y a partir de ellos determinaremos la inclusión del punto objeto de estudio al polígono, estudiando si el punto está o no en triángulos originales incompletos en los que se descompondrá.

Vamos a ver la descomposición de un polígono en triángulos originales no completos. Supondremos que el sólido está determinado por n aristas y orientado

positivamente (en el caso de los agujeros la orientación positiva implica que las aristas de su frontera están orientadas en sentido de las agujas del reloj) y que la presencia del sólido es 1.

Teorema 3.1 Sea P un sólido bidimensional, A_i , $i=1..n$, sus aristas, T_i el triángulo original incompleto formado por el origen y A_i . Entonces

$$Int(P) = O' + Int \left(\sum_{i=1}^n sign([OA_i]) * T_i \right)$$

O' es un conjunto que coincide con el origen si este punto pertenece al interior P , y que es vacío en caso contrario.

Demostración:

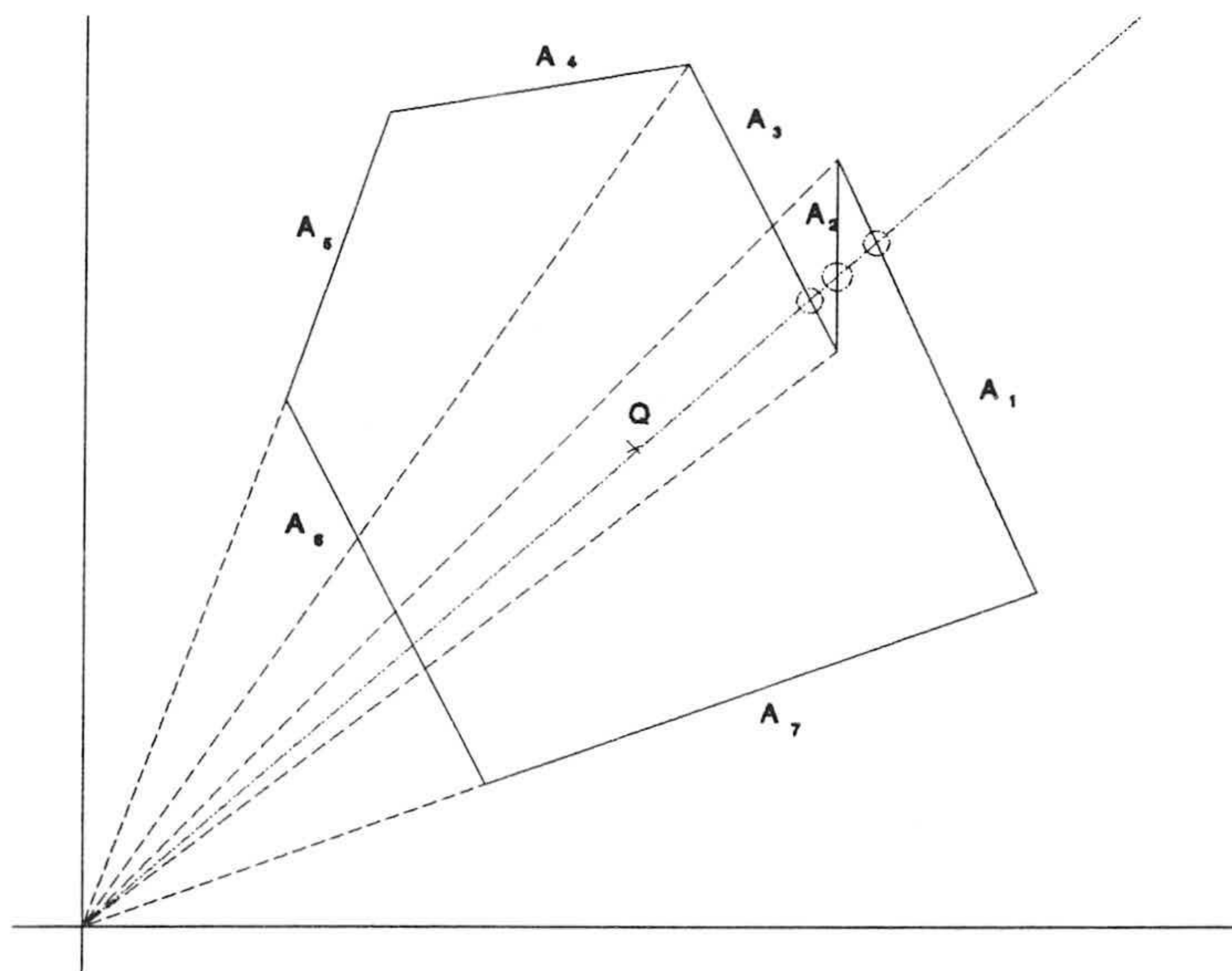


Figura 1. Teorema 2.1

Demostraremos la doble inclusión. Comenzaremos viendo la inclusión del primer miembro en el segundo.

Sea Q un punto interior a P distinto del origen. Si $Q=O$ la inclusión del origen en el segundo miembro sería evidente. La semirrecta que parte del origen, pasa por Q y se dirige al infinito, contiene, a partir de Q , un número impar de intersecciones con la frontera del sólido.

Para cada triángulo original incompleto positivo que contiene a Q , habrá un punto de intersección entre la semirrecta y la arista no original de dicho triángulo. Esto necesariamente implica que la semirrecta en ese punto de intersección va de dentro del sólido hacia fuera del sólido. Lo contrario ocurre

con los triángulos negativos: la semirrecta avanza del exterior al interior. Por tanto, contar el número de intersecciones es equivalente a contar los signos de los triángulos, con el único caso especial de que la semirrecta atravesase algún vértice.

Si la semirrecta pasa por un vértice tenemos dos posibilidades:

- a) el vértice es común a dos triángulos positivos ó a dos negativos (debido a la estructura de los triángulos originales no completos el punto Q pertenecería solamente a uno de ellos y sólo se contaría un signo, que es lo adecuado)
- b) el vértice es común a un triángulo positivo y a uno negativo: en este caso, en el punto de intersección no hay cambio de orientación de dentro hacia afuera o viceversa y no se debe contar como punto de intersección; esto ocurre así ya que la suma de las presencias de esos dos triángulos sería 0.

Si la semirrecta contiene una arista, se consideran sólo como puntos de corte los vértices de dicha arista y se reduciría a uno de los casos anteriores.

Por ser Q interior existirá un entorno de Q totalmente contenido en P . Para todos los puntos del entorno se puede repetir el mismo argumento anterior por lo que dichos puntos estarán en el conjunto determinado por el segundo miembro, y por tanto Q será un punto interior de ese conjunto.

Para demostrar la inclusión del segundo miembro en el primero lo haremos por reducción al absurdo. Supongamos que existe un punto Q' perteneciente al segundo miembro y exterior a P . La semirrecta que parte del origen, pasa por Q' y se dirige al infinito, contiene, a partir de Q' , un número par de intersecciones con la frontera del sólido.

Para cada triángulo original incompleto positivo que contiene a Q' , habrá un punto de intersección entre la semirrecta y la arista no original de dicho triángulo. Esto necesariamente implica que la semirrecta en ese punto de intersección va de dentro del sólido hacia fuera del sólido. Lo contrario ocurre con los triángulos negativos: la semirrecta avanza del exterior al interior. Por

tanto, contar el número de intersecciones es equivalente a contar los signos de los triángulos, con el único caso especial de que la semirrecta atravesase algún vértice.

Si la semirrecta pasa por un vértice tenemos, al igual que antes, dos posibilidades:

- a) el vértice es común a dos triángulos positivos ó a dos negativos (debido a la estructura de los triángulos originales no completos el punto Q' pertenecería solamente a uno de ellos y sólo se contaría un signo, que es lo adecuado)
- b) el vértice es común a un triángulo positivo y a uno negativo: en este caso, en el punto de intersección no hay cambio de orientación de dentro hacia afuera o viceversa y no se debe contar como punto de intersección; esto ocurre así ya que la suma de las presencias de esos dos triángulos sería 0.

Si la semirrecta contiene una arista, se consideran sólo como puntos de corte los vértices de dicha arista y se reduciría a uno de los casos anteriores. ■

Para usar en la práctica este algoritmo de inclusión es necesario determinar si un punto pertenece o no al interior de un triángulo original no completo, o si un punto está contenido o no en una arista. Los dos lemas siguientes nos permiten determinar estas cuestiones.

Lema 3.1 Dados tres puntos distintos P_1, P_2 y P_3 , donde (x_i, y_i) son las coordenadas de P_i ($i=1,2,3$), entonces P_3 pertenece a la arista P_1P_2 si, y solo si,:

$$\text{sign}([P_1P_2P_3])=0 \text{ y } (x_2 - x_3)*(y_2 - y_3) + (x_1 - x_3)*(y_1 - y_3) < 0$$

Demostración

La primera condición (área del triángulo igual a cero) es equivalente a que los tres puntos son colineales. La segunda condición, supuesta ya la primera, determina que los vectores P_3P_2 y P_3P_1 forman un ángulo de 180, ya que el signo viene determinado por el coseno del ángulo, y los ángulos posibles sería 0 y 180. ■

Lema 3.2 Sea OAB un triángulo original incompleto con orientación positiva y Q un punto arbitrario. Entonces Q pertenece a OAB si: $\text{sign}([QOA]) \geq 0$, $\text{sign}([QAB]) \geq 0$ y $\text{sign}([QBO]) > 0$.

Demostración

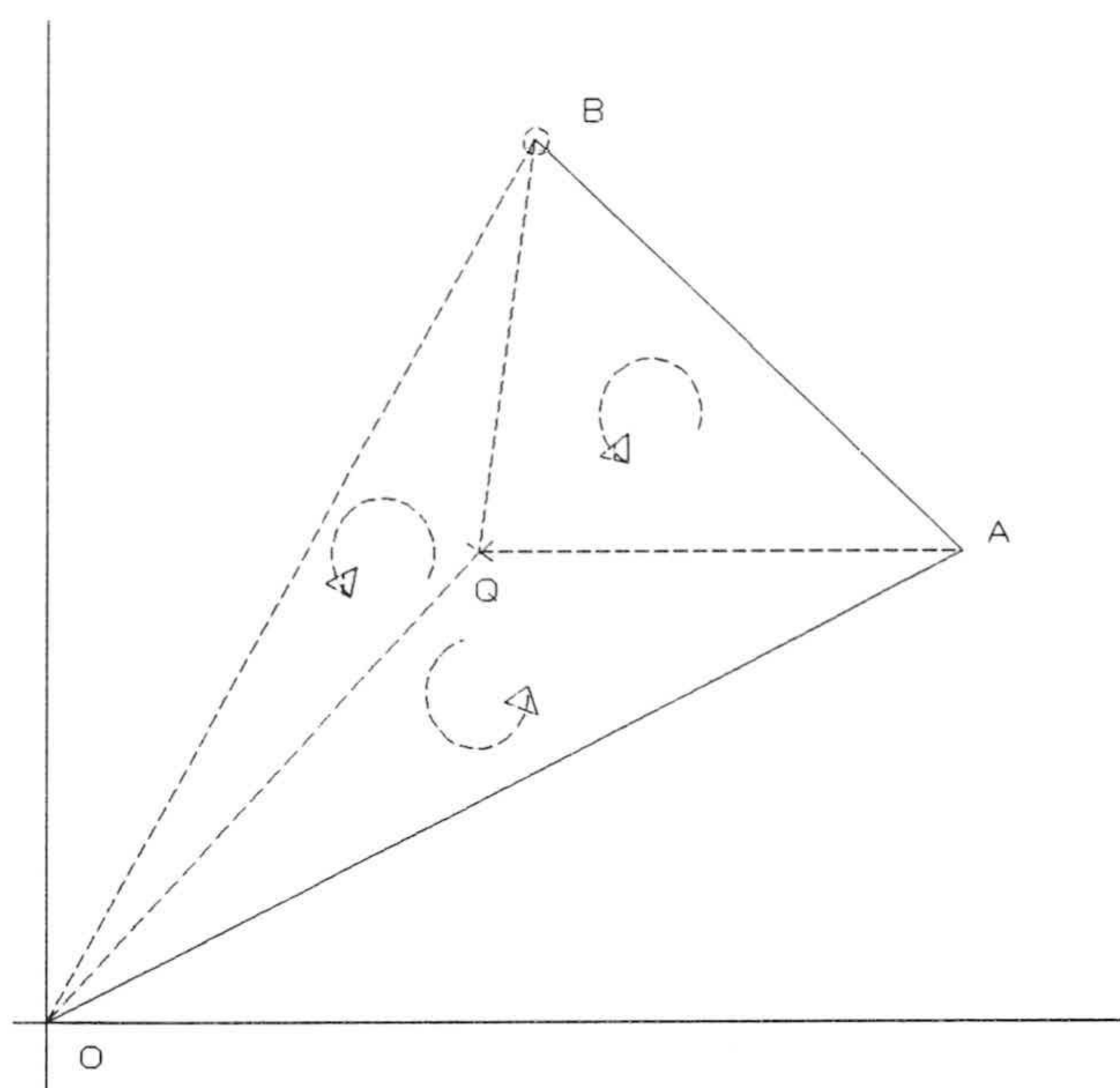


Figura 3.2 Pertenencia de Q a OAB

Q será un punto interior, si y solo si está a la izquierda de los segmentos AB , BO and OA , lo que implica que las áreas signadas $[QAB]$, $[QBO]$ y $[QOA]$ son positivas. Si además una de las áreas es cero, entonces el punto está en la arista correspondiente.

Si el triángulo OAB tuviese orientación negativa, bastaría estudiar el triángulo OBA . ■

Podría pensarse que la necesidad de realizar el estudio del conjunto O' hace el algoritmo ineficaz en el caso de tener que determinar la inclusión o no del origen de coordenadas. Esto no es así, ya que si queremos estudiar si el origen de coordenadas es ó no interior a un sólido bidimensional bastaría hacer una traslación de ejes, tomando como nuevo origen uno de los vértices del sólido (claramente la traslación de los ejes de coordenadas no cambia la posición relativa de un punto respecto de un sólido bidimensional).

Cabe destacar que en la implementación del algoritmo, el estudio de igualdades numéricas del tipo $E=0$ debe adaptarse al trabajo con aritmética real. Por tanto, se debería definir un umbral positivo E' de modo que $E=0$ se calcule como $|E| < E'$.

Pasaremos ahora a exponer el algoritmo de inclusión que se deriva de estos lemas. Para ello supondremos que tenemos un sólido bidimensional, pudiendo estar formado por varios elementos, cada uno de ellos determinados por ciclos de aristas que determina su frontera, y que poseen una orientación coherente; si esto no fuese así sería

necesario primero ordenar coherentemente las aristas que determinan su frontera. Supondremos también que el origen no es un punto posible de estudio.

Con las consideraciones anteriores, el algoritmo en pseudocódigo se podría expresar del siguiente modo:

$P = A_{a_2} \dots A_n$ un sólido bidimensional, siendo A_i las aristas y T_i los triángulos originales no completos determinados por cada una de las aristas y el origen, $T_i = [OA_i]$:

Función para determinar si el punto Q está dentro del sólido P

PRECONDICIONES: $Q \neq O$
 P con orientación positiva

POSTCONDICIONES: DENTRO = $Q \in P$

```

INCLUSION=0
  For i=1 To n
    If  $Q \in A_i$  Then DENTRO=SI. Exit
    If sign[ $OA_i$ ] >0 and  $Q \in T_i$  then
      INCLUSION=INCLUSION+1
    If sign[ $OA_i$ ] <0 and  $Q \in T_i$  then
      INCLUSION=INCLUSION-1
  Endfor
  DENTRO=(INCLUSION == 1)
End
  
```

Una arista con signo cero (colineal con el origen) no afecta a la presencia final. En cualquier caso, se comprueba si el punto Q está o no en una arista. Lo mismo pasaría con ciclos formados por un solo punto ó por dos ó mas puntos colineales.

A continuación presentamos un ejemplo de la aplicación del algoritmo en varios casos, mostrados en la figura 3.3.

Para el punto Q_1 tendríamos:

Q_1 está en T_1 e $INCLUSION = \text{sig}([OA_1]) = 1$

Q_1 está en T_2 e $INCLUSION = INCLUSION + \text{sig}([OA_2]) = 1 - 1 = 0$

Q_1 está en T_3 e $INCLUSION = INCLUSION + \text{sig}([OA_3]) = 0 + 1 = 1$

Q_1 no está en el resto de triángulos y el valor final es $INCLUSION = 1$

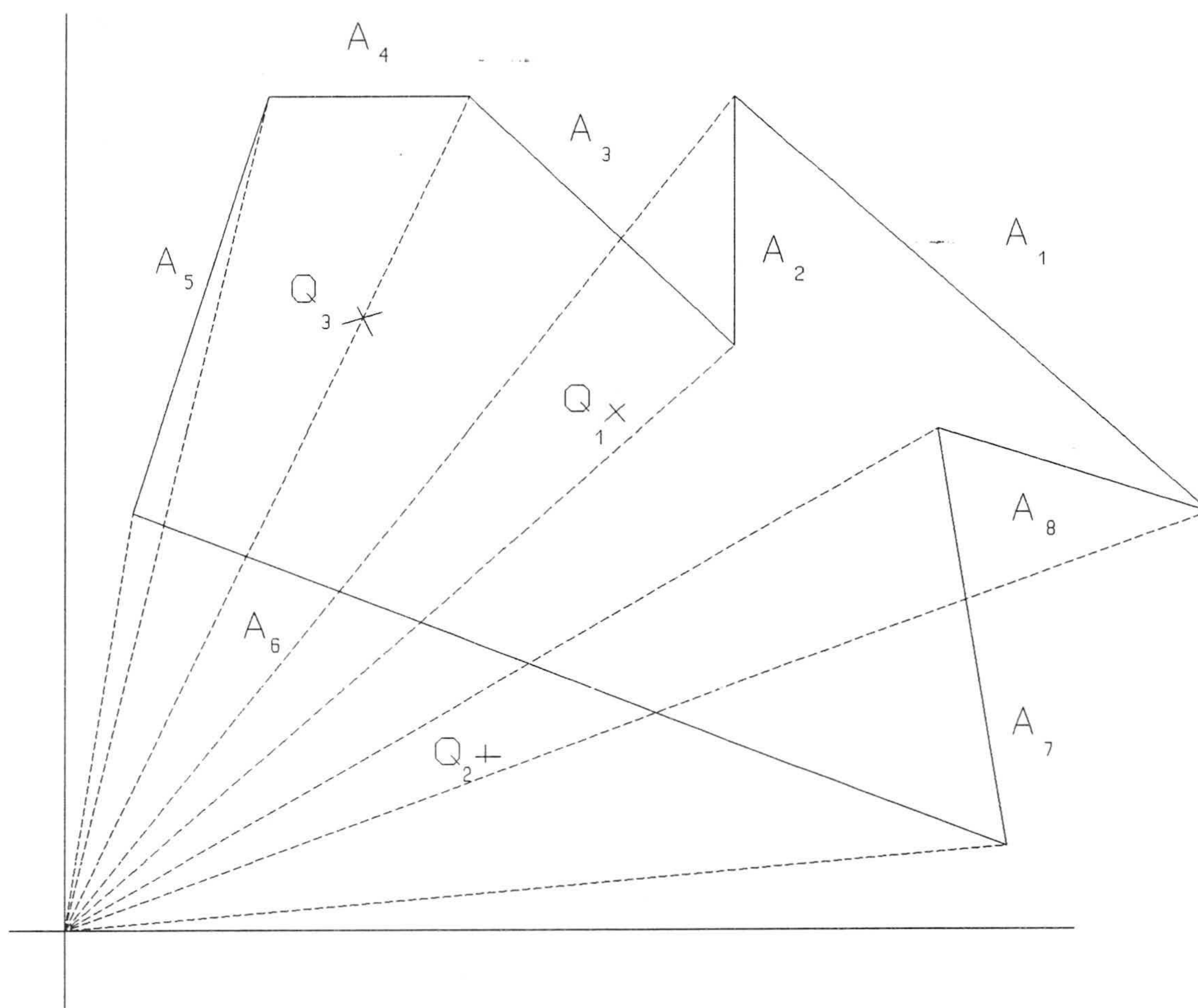


Figura 3.3 Algoritmo de inclusión en 2D

Para el punto Q_2 , los sucesivos valores de $Q \in T_i$ serán

cierto, falso, falso, falso, falso, cierto, cierto, falso

y por tanto los de la variable $INCLUSION$ serían: 1,1,1,1,1,0,1,0

Para Q_3 , los valores sucesivos de $Q \in T_i$ serán

falso, falso, falso, cierto, falso, falso, falso, falso

de la variable INCLUSION serían: 0,0,0,1,1,1,1,1

Ya que hay que evaluar la presencia del punto Q en cada triángulo original, la complejidad de este algoritmo es $O(n)$, donde n es el número de aristas (por cada arista existe un triángulo original). Los algoritmos presentados previamente ([PreSh85]) poseen la misma complejidad. La diferencia del algoritmo que proponemos esta, además de en su sencillez, en su robustez y eficiencia, ya que el algoritmo no necesita evaluar ni cocientes ni funciones trigonométricas.

En una implementación realizada en una CPU Intel 386, con coprocesador matemático, al objeto de comparar este algoritmo (A) con el algoritmo de inclusión por medio del numero de giro ("winding number") ([Kalay89]) se obtuvo la siguiente tabla de tiempos de ejecución (en cada uno de los casos se realizo el proceso con 1000 polígonos simples distintos generados previamente y guardados en un fichero) ([FeiTU92]).

	Número de Aristas							
	7	8	9	10	11	12	13	14
A	1971	2295	2626	2966	3280	3608	3945	4269
B	3764	4257	4993	5581	6180	6663	7148	8065
A/B en %	52,36	53,91	52,59	53,14	53,07	54,15	55,19	52,93

Tabla I

Sin embargo pensamos que la importancia del algoritmo presentado no es sólo la reducción de tiempo y la sencillez de las operaciones a implementar, sino que se fundamenta en conceptos que pueden generalizarse fácilmente a dimensiones superiores, tal y como vamos a ver a continuación.

3.2 Algoritmo de inclusión en tres dimensiones

Junto con el algoritmo general basado en el cálculo del número de intersecciones, Kalay presenta otros dos algoritmos que se basan en la reducción del problema a dos dimensiones [Kalay82]. Estos dos algoritmos son complejos de implementar y además presentan gran cantidad de casos singulares.

Nuestro algoritmo se basa, al igual que el de dos dimensiones, en la descomposición de un sólido (tridimensional en este caso) realizada en el capítulo 2. Ya que estamos en el espacio los elementos de la descomposición serán 3-símplices (tetraedros en tres dimensiones).

Al igual que en dos dimensiones daremos un conjunto de lemas que justificarán la correcta definición e implementación del algoritmo.

En el caso de dos dimensiones era muy sencillo implementar las operaciones regularizadas necesarias por medio de triángulos originales incompletos. Ya que en tres dimensiones es muy complicado trabajar correctamente con símlices contiguos, es más fácil implementar en el propio algoritmo los condicionamientos a que dan lugar las operaciones regularizadas.

Si el punto de estudio es el origen, previamente se haría, como en dos dimensiones, una traslación de ejes considerando como nuevo origen uno de los vértices del sólido.

Los lemas 3.1 y 3.2 dados anteriormente nos permiten determinar si un punto coplanario con una de las caras de un tetraedro está dentro ó fuera de la cara, aplicando el lema 3.2 a una cualquiera de las tres proyecciones sobre los ejes de la cara correspondiente). Además, podemos conocer si un punto colineal con una arista está dentro de ella o no aplicando el lema 3.1 a una de las proyecciones ó generalizando el lema a tres dimensiones.

Los 3-símplices (tetraedros) que aparecen para cada cara, estarán formados por el origen de coordenadas, un vértice fijo de la cara (el primero por ejemplo) y cada una de las aristas de cada cara no incidentes con el vértice fijo elegido.

Para determinar el algoritmo se tendrá en cuenta que se pueden dar los siguientes cuatro casos posibles:

- a) El punto está dentro de una cara: es evidente que el punto pertenece al sólido. Para determinar si el punto está dentro o fuera de una cara estudiamos si el punto es coplanario con la cara (la distancia de dicho punto al plano determinado por la cara sería cero) y además el punto está contenido en una de las tres proyecciones sobre los ejes de la cara (se puede elegir cualquiera de ellas excepto en el caso de que la cara sea paralela a uno de los planos de coordenadas). Para determinar si el punto de proyección está ó no en la proyección de la cara se usaría el algoritmo en dos dimensiones.
- b) El punto está en el interior de un tetraedro. Según el teorema presentado en el capítulo dos, la inclusión del punto en el sólido vendrá determinada por la suma total de las presencias del punto en todos los tetraedros que forman el sólido (Teorema 2.4).
- c) El punto es interior a una de las caras del tetraedro (triángulos) que son comunes a dos tetraedros. Para realizar correctamente la operación regularizada bastará asignar a estos puntos una presencia de $1/2$ (o $-1/2$ si la orientación es negativa) para que la presencia entre los dos tetraedros sea 1, 0 ó -1 según el caso.
- d) El punto pertenece a una de las aristas del tetraedro. La presencia regularizada del punto en el sólido debe ser $+1$ o -1 (dependiendo del signo de la cara objeto de estudio), aunque la arista sea común a varios tetraedros. Es necesario guardar los vértices que verifiquen esta propiedad al objeto de considerar la presencia una sola vez. Ya que puede haber caras positivas y negativas será necesario, para el punto de estudio, definir dos conjuntos: conjunto de vértices de caras positivas que definen una arista original que contiene al punto de estudio y conjunto de vértices de caras negativas que definen una arista original que contienen al punto de estudio.

El esquema del algoritmo, en pseudocódigo, es el siguiente (una forma más completa aparece en la tabla II, al final del capítulo):

```

DENTRO=NO;INCLUSION=0;
sets: vértices_POSITIVOS=empty, vértices_NEGATIVOS=empty
For i=1 hasta Número_caras
  If Q en cara(i) DENTRO=SI.EXIT
  If Q en la primera ó última arista original de pirámide(i)
    y punto no en sets
      INCLUSION=INCLUSION + signo(cara(I))
      actualizar sets
  Else
    For j=1 hasta Numero_tetraedros_cara(i)
      estudiar casos:

      si punto en cara del tetraedro
        INCLUSION=INCLUSION + 1/2*signo(tetraedro)
      si punto en arista original y punto no en sets
        INCLUSION=INCLUSION + signo(cara(i))
        actualizar sets
      si punto en tetraedro
        INCLUSION=INCLUSION + signo(tetraedro)
    Endfor
  Endif
Endfor
DENTRO=INCLUSION == 1
  
```

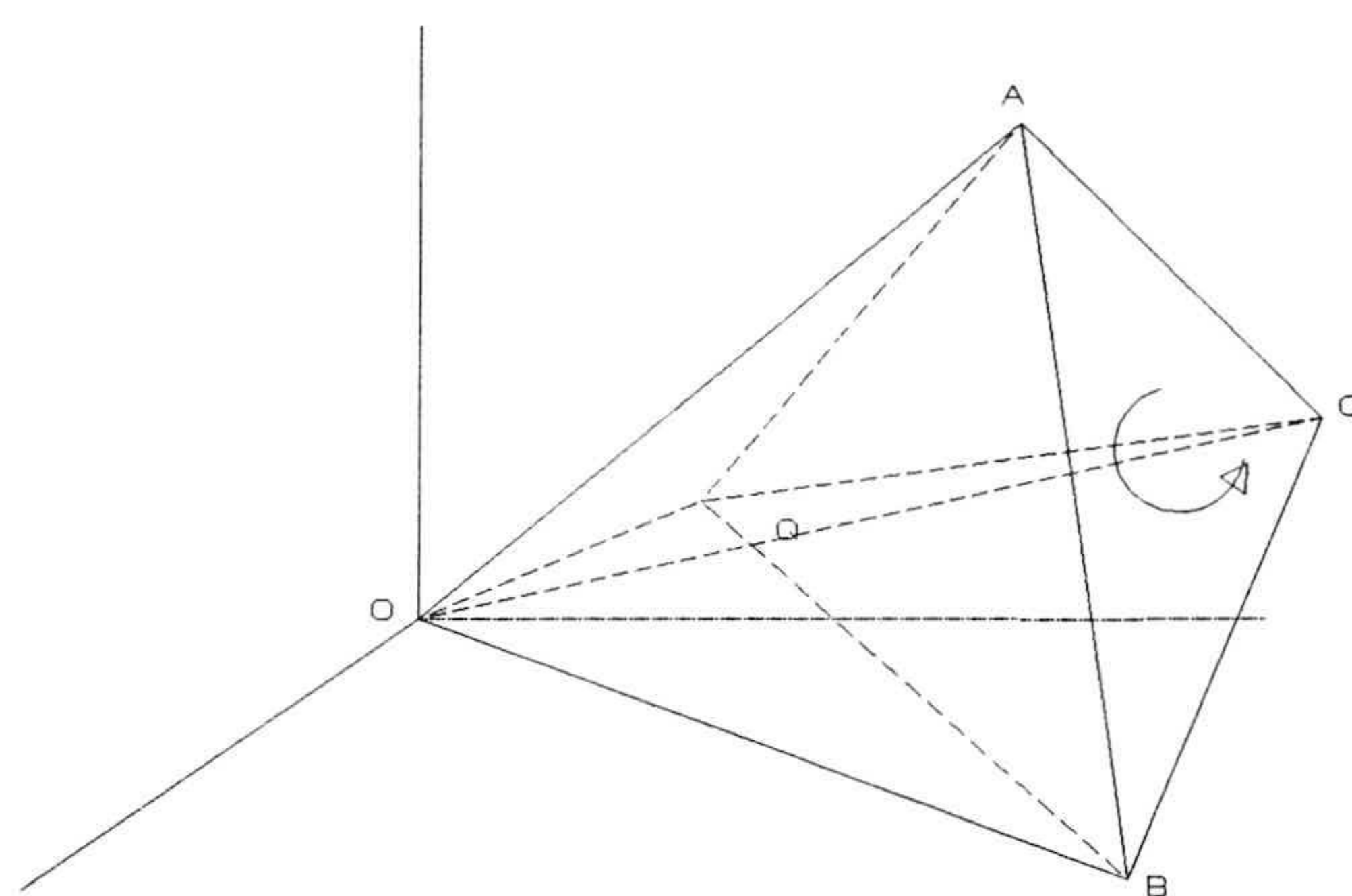
Suponemos que las caras y aristas que determinan el poliedro son accesibles por medio de una estructura de datos adecuada, y que el poliedro tiene una orientación conforme.

Al objeto de implementar el algoritmo sería necesario disponer de un test de inclusión para un punto en un tetraedro (ya que el resto de los casos se estudiaría por

medio de los lemas dados anteriormente). Esta inclusión se resuelve por medio del siguiente lema.

Lema 3.3. Sea $OABC$ un tetraedro con orientación positiva y Q un punto del espacio. Q será interior al tetraedro si, y solo si

$$\begin{aligned} &\text{sign}([OABQ]) > 0 \quad \text{y} \\ &\text{sign}([QABC]) > 0 \quad \text{y} \\ &\text{sign}([OQBC]) > 0 \quad \text{y} \\ &\text{sign}([OACQ]) > 0. \end{aligned}$$



Demostración

Q será un punto interior si y solo si ve los tres puntos de cada una de las caras en sentido de las agujas del reloj, lo que implica que los volúmenes signados $[OABQ]$, $[QABC]$, $[OQBC]$ y $[OACQ]$ han de ser positivos (ver figura3.4). ■

Figura 3.4 Lema 3.3

En la figura 3.5 podemos ver un ejemplo de la aplicación del algoritmo.

Para el punto Q_1 , su situación relativa respecto de cada tetraedro de cada una de las caras determina que los sucesivos valores de la variable INCLUSION son:

$$0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,0,0,0.$$

Para el punto Q_2 , su situación relativa respecto de cada tetraedro de cada una de las caras determina que los valores sucesivos de la variable INCLUSION son:

$$0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1.$$

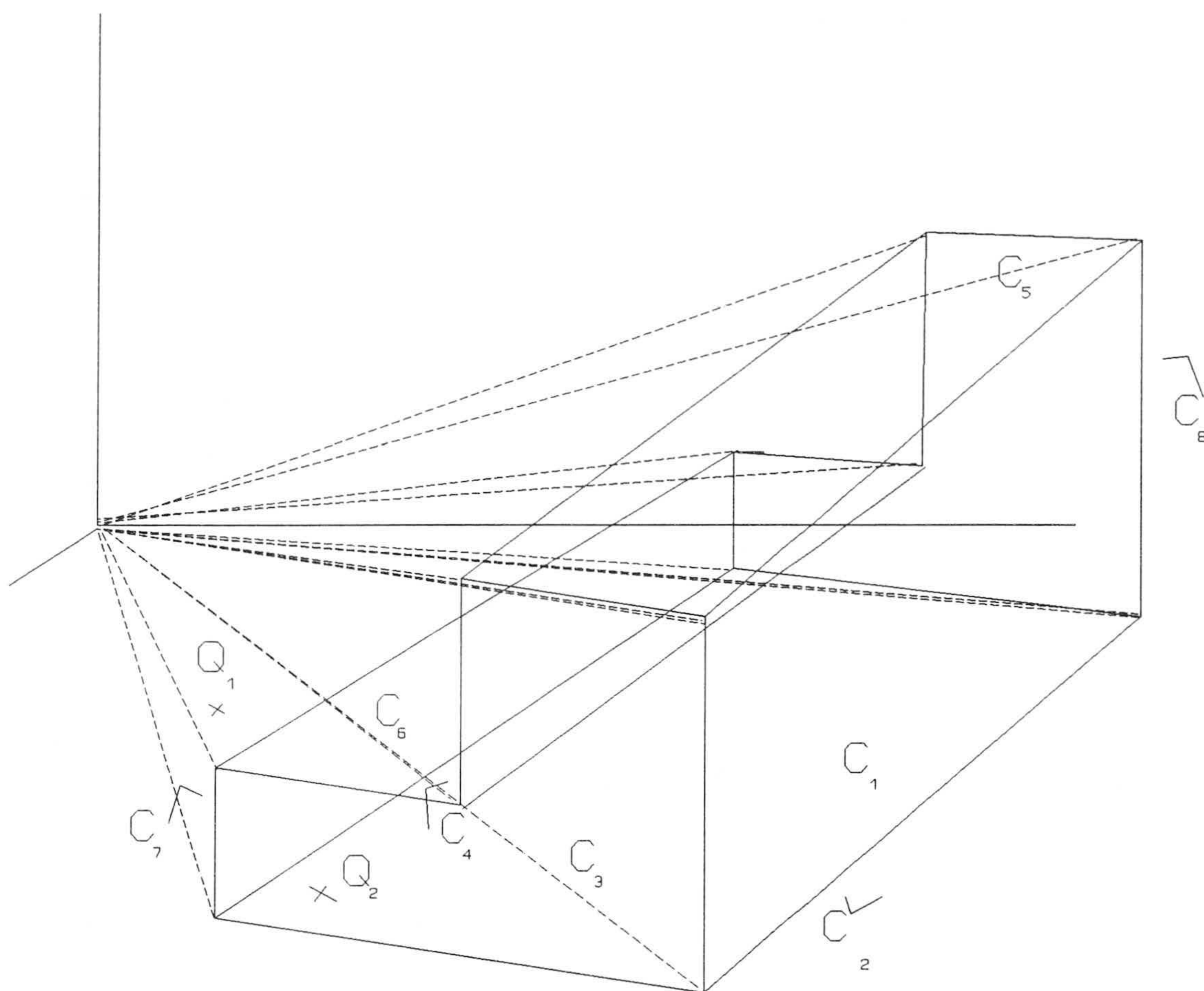


Figura 3.5 Algoritmo de inclusión en 3D

Los casos singulares de estudio son mayores en el algoritmo de paridad ya que hay que evaluar el tipo de punto de intersección, y esta evaluación es más compleja que en la implementación hecha de las operaciones regularizadas.

Hemos comparado nuestro algoritmo (A) con el algoritmo de paridad de intersecciones (B), aplicándolo a 5 figuras (ver figura 3.6). Las figuras tienen todas, aproximadamente, la misma caja mínima envolvente.

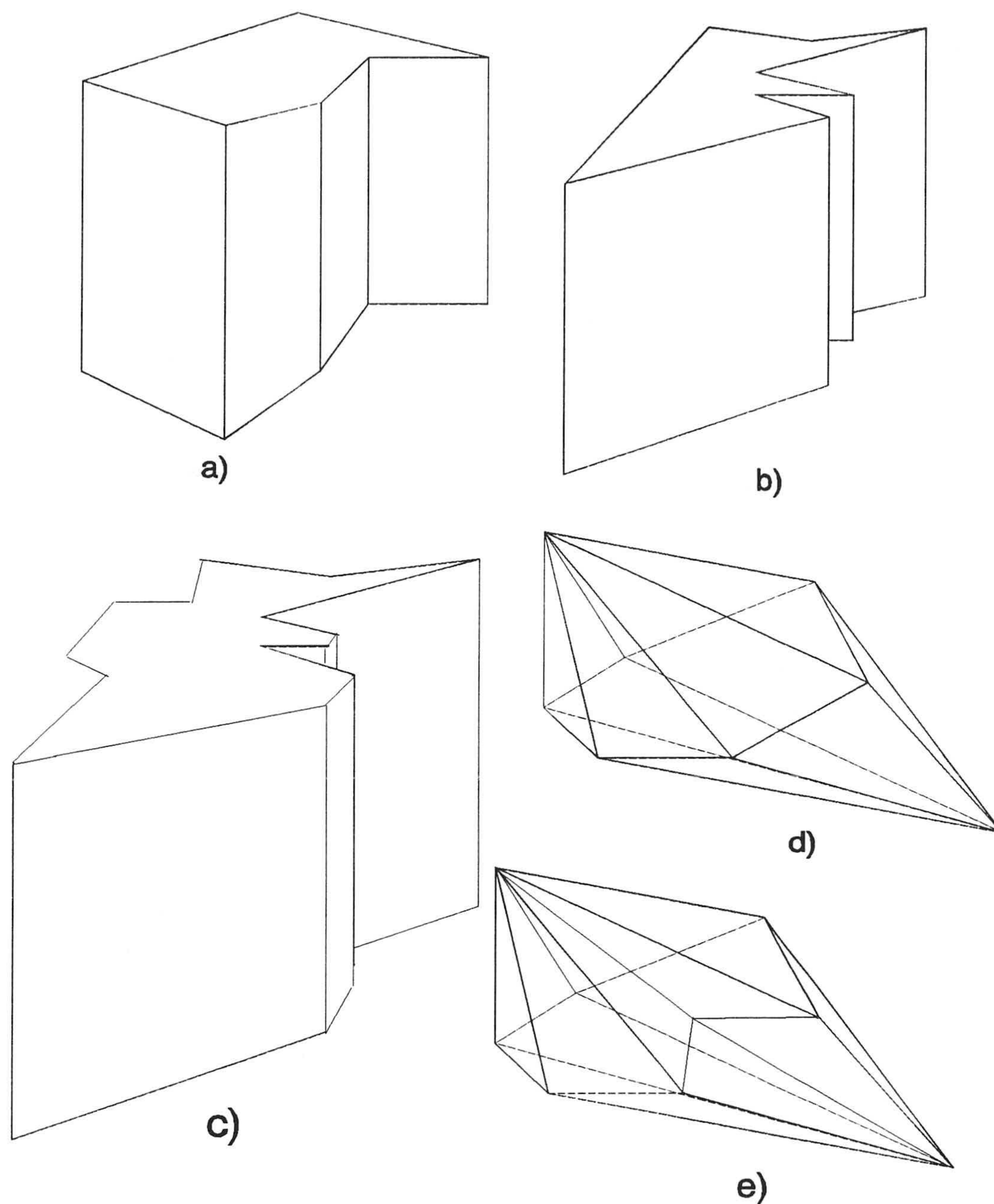


Figura 3.6 Sólidos tridimensionales

Se ha aplicado el algoritmo 1000 puntos generados aleatoriamente, y cumpliendo que estaban dentro de la caja mínima envolvente (si el punto está fuera de la caja mínima evidentemente está fuera del sólido).

Sólidos	Algor. A	Algor.B	A/B en %
Fig a	5.317	6.509	81,69
Fig b	7.404	8.800	84,14
Fig c	7.919	10.219	77,49
Fig d	4.073	5.162	78,90
Fig e	4.861	6.196	78,45

Tabla II

Es este capítulo hemos presentado dos algoritmos de inclusión para dos y tres dimensiones, que se basan en el desarrollo formal realizado en el capítulo anterior. La implementación es sencilla de realizar. El tiempo de ejecución se mejora respecto del algoritmo del número de giro (en el caso de 2 dimensiones) y del de paridad de intersecciones (en el caso de tres dimensiones).

```

INCLUSION=0
vertex_POSITIVE=vacio
vertex_NEGATIVE=vacio
for i=1 to n {
    if (Q ∈ Ci) { INSIDE=true; EXIT }
    if ((Q ∈ OV1,1,i) and ((sign(OCi)>0 and not V1,1,i ∈ vertex_POSITIVE) or
    (sign(OCi)<0 and not V1,1,i ∈ vertex_NEGATIVE))) {
        INCLUSIONS=INCLUSION+sign(OCi)
        if (sign(OCi)<0)
            vertex_NEGATIVE=vertex_NEGATIVE + V1,1,i
        else vertex_POSITIVE=vertex_POSITIVE + V1,1,i }
    else if ((Q in OVki,1,i) and ((sign(OCi) >0 and not Vki,1,i ∈ vertex_POSITIVE) or
    (sign(OCi)<0 and not Vki,1,i ∈ vertex_NEGATIVE)))) {
        INCLUSIONS=INCLUSION+sign(OCi)
        if (sign(OCi) <0) vertex_NEGATIVE=vertex_NEGATIVE + Vki,1,i
        else vertex_POSITIVE=vertex_POSITIVE + Vki,1,i }
    else for j=2 to Ki-1 {
        if ((Q ∈ interior of OV1,1,iVj,1,i) or (Q ∈ interior of OV1,1,iVj,2,i)
        are (Q ∈ interior of OVj,1,iVj,2,i)) {
            INCLUSION=INCLUSION + 1/2*sign([OV1,1,iVj,1,iVj,2,i])
        else if ((Q in OVj,1,i) and (( sign([OV1,1,iVj,1,iVj,2,i])>0
        and not Vj,1,i ∈ vertex_POSITIVE) or (sign([OV1,1,iVj,1,iVj,2,i])<0
        and not Vj,1,i in vertex_NEGATIVE))) {
            INCLUSION=INCLUSION+sign([OV1,1,iVj,1,iVj,2,i])
            if sign([OV1,1,iVj,1,iVj,2,i])<0
                vertex_NEGATIVE=vertex_NEGATIVE + Vj,1,i
            else vertex_POSITIVE=vertex_POSITIVE + Vj,1,i }
        else if (Q in OV1,1,iVj,1,iVj,2,i)
            INCLUSION=INCLUSION+sign([OV1,1,iVj,1,iVj,2,i])
        }
    }
}

```

Sea $P=C_1, C_2, \dots, C_n$ un poliedro donde C_i , ($i=1..n$) son las caras, $A_{j,i}$ ($j=1..K_i$) las aristas de la cara i , y $V_{m,1,i}, V_{m,2,i}$ son los dos vértices de la arista m . $\text{Sign}(OC_i)$ es el signo de la pirámide OC_i . El algoritmo decide si el punto Q está dentro del poliedro; el resultado se obtiene en la variable INCLUSION, la cual será 1 solo si $Q \in P$.

Tabla III

CAPITULO IV

Modelado de sólidos por recubrimientos simpliciales

Los resultados obtenidos en el capítulo 2 permiten la posibilidad de representar un objeto gráfico como una expresión de símlices originales.

En este capítulo planteamos la posibilidad de utilizar una representación basada en símlices originales para objetos poliédricos.

Ya que en los métodos de modelado actuales no están implementadas todas las operaciones posibles del álgebra desarrollada anteriormente nos tendremos que restringir a aquellas que sí son habituales en cualquier sistema de modelado. Es decir, no usaremos las operaciones del espacio vectorial sino exclusivamente las operaciones del álgebra de Boole: unión, intersección, complementación y diferencia conjuntista.

El problema radica en que los teoremas demostrados en el capítulo 2 se han formulado para la operación suma y su opuesta. El problema radica en formular este mismo desarrollo no a nivel abstracto, sino aplicándolo a las operaciones booleanas regularizadas.

En primer lugar, se realiza una aproximación del método, no óptima, pero aplicable en muchos casos debido a su sencillez. Se estudia tanto en dos como en tres dimensiones, y en este caso tanto para recubrimientos por pirámides como por tetraedros.

Posteriormente, y con el objeto de lograr recubrimientos óptimos se introduce una sencilla clasificación que establece en cada sólido una relación de orden definida para sus símlices asociados.

4.1 Recubrimientos simpliciales en 2 dimensiones

En este apartado trabajaremos con sólidos en dos dimensiones; es decir polígonos limitados por ciclos de aristas que supondremos están orientados en sentido contrario del reloj (orientación positiva). Podrían existir intersecciones de dimensión 0 entre varias partes completas. Existirán agujeros, y en este caso su orientación será la contraria de los ciclos que forman las fronteras exteriores. Ya se ha indicado en el capítulo dos que a la hora de determinar los símlices de una descomposición, se podría elegir cualquier punto de referencia. Por sencillez, y sin que suponga pérdida de generalidad, supondremos que elegimos como punto de referencia el origen de coordenadas.

Hemos de suponer que tenemos accesibles los diferentes elementos básicos que conforman adecuadamente el sólido: los ciclos de aristas ordenadas adecuadamente y para cada arista los vértices que la definen orientados correctamente.

Usaremos los conceptos de triángulo original (2-simplex original), pirámide y de función signo introducidos en el capítulo dos. A partir de estos conceptos, el signo de un triángulo será 1, -1 ó 0 según que, vista desde el origen, se recorra la arista no original en sentido contrario de las agujas del reloj (orientación positiva) o en sentido de las agujas del reloj (orientación negativa). Si los tres vértices del triángulo son colineales, su signo es cero.

Al objeto de establecer una jerarquía entre los triángulos originales asociados a un sólido bidimensional introducimos la siguiente definición:

Definición 4.1 (Triángulos subordinados): Dados dos triángulos originales $T=OP_1P_2$ y $T'=OQ_1Q_2$, diremos que T' está subordinado a T , $T \triangleleft T'$ (ó que T es dominante de T' , $T' \triangleright T$) si $T=T'$ o si existe algún punto S perteneciente al interior (unidimensional) del segmento Q_1Q_2 que también pertenece al interior del triángulo T . Esto es:

$$\exists \lambda \in (0,1) \mid \lambda \cdot Q_1 + (1-\lambda) \cdot Q_2 \in \text{int}(T)$$

Si el segmento Q_1Q_2 está totalmente contenido en T diremos que T' está totalmente subordinado a T .

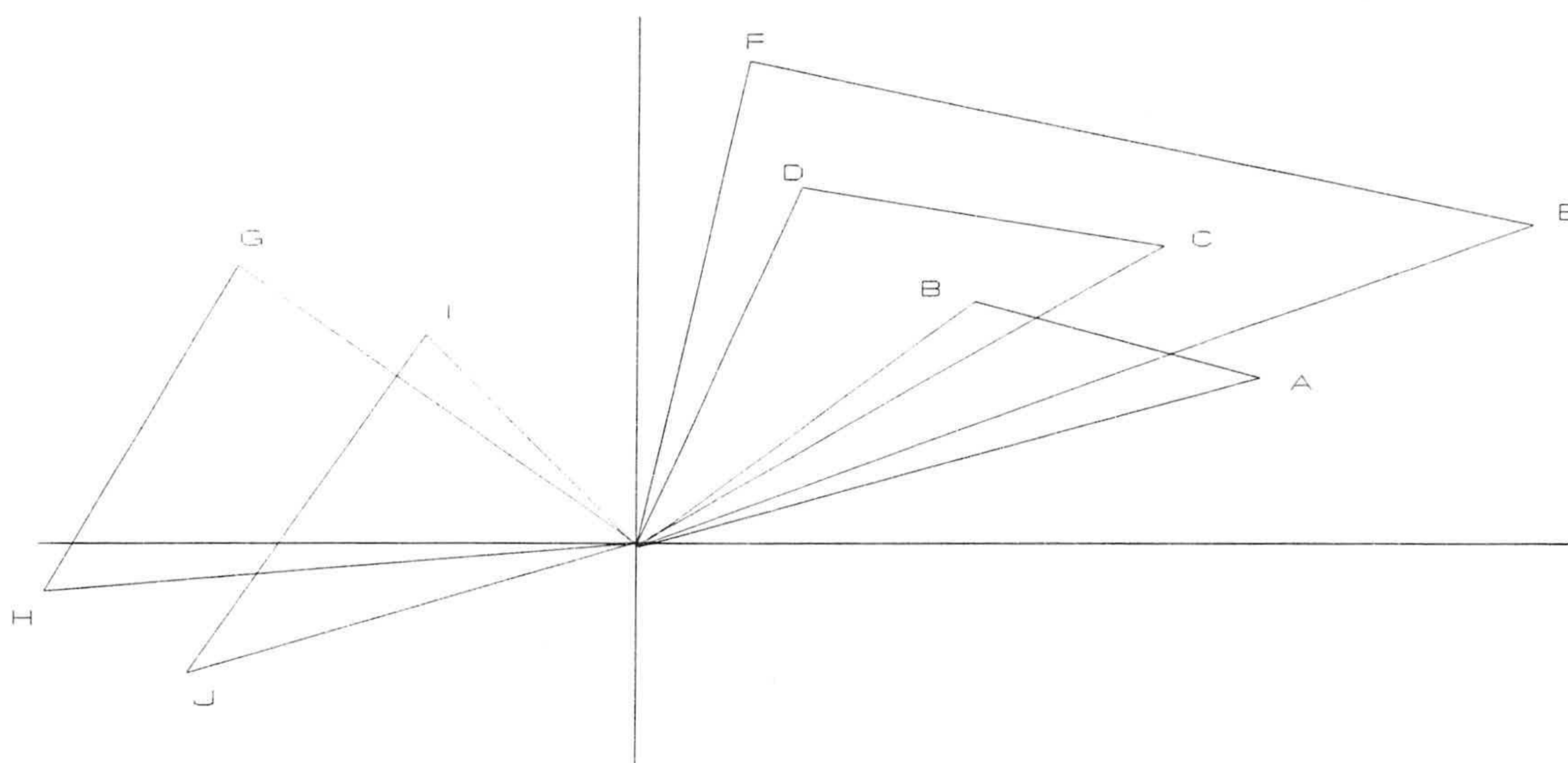


Figura 4.1 Triángulos subordinados

La Figura 4.1 muestra algunos triángulos subordinados. El triángulo $OAB \triangleleft OEF$; $OAB \triangleleft OEF$; $OIJ \triangleleft OGH$.

Nuestro objetivo es obtener la descomposición de un sólido bidimensional en triángulos originales. Los triángulos implicados en esta descomposición serán aquellos definidos por las aristas del sólido y el origen de coordenadas, por lo que sólo estamos interesados en el estudio de la relación de subordinación entre triángulos originales en los que las aristas no originales solo se pueden cortar en los vértices (por pertenecer a un sólido bidimensional correctamente definido, en el sentido indicado en el capítulo 1: entre los diversos elementos completos sólo se admiten intersecciones de dimensión 0, es decir, puntos).

Podemos estudiar la subordinación de T y T' evaluando la situación relativa de los vértices Q_1 y Q_2 de T' respecto de T . Caben tres posibilidades:

- Q_1 y Q_2 están ambos dentro de T . En este caso hay subordinación total de T' a T .
- solo uno entre Q_1 y Q_2 está en T . En este caso hay subordinación.
- Q_1 y Q_2 están ambos fuera de T . En este caso hay subordinación si el segmento Q_1Q_2 corta a las aristas originales de T .

Usando los tests anteriores es muy sencillo determinar si un triángulo está subordinado a otro, como veremos posteriormente.

Es obvio que en muchos casos el sólido bidimensional puede expresarse como la unión de los triángulos originales de signo positivo menos la unión de los triángulos originales de signo negativo. El polígono a) de la figura 4.2. podría expresarse como:

$$P = (OA_1 \cup OA_2 \cup OA_3) - OA_4$$

Sin embargo, como observamos en b) de la figura 4.2, cuando el polígono es cóncavo esto no es siempre posible.

Nuestro método divide al sólido en varios trozos, uno por cada triángulo original positivo. Este trozo se calcula como dicho triángulo menos la unión de todos sus triángulos subordinados.

En orden a obtener una definición formal de esta descomposición introduciremos las siguientes definiciones:

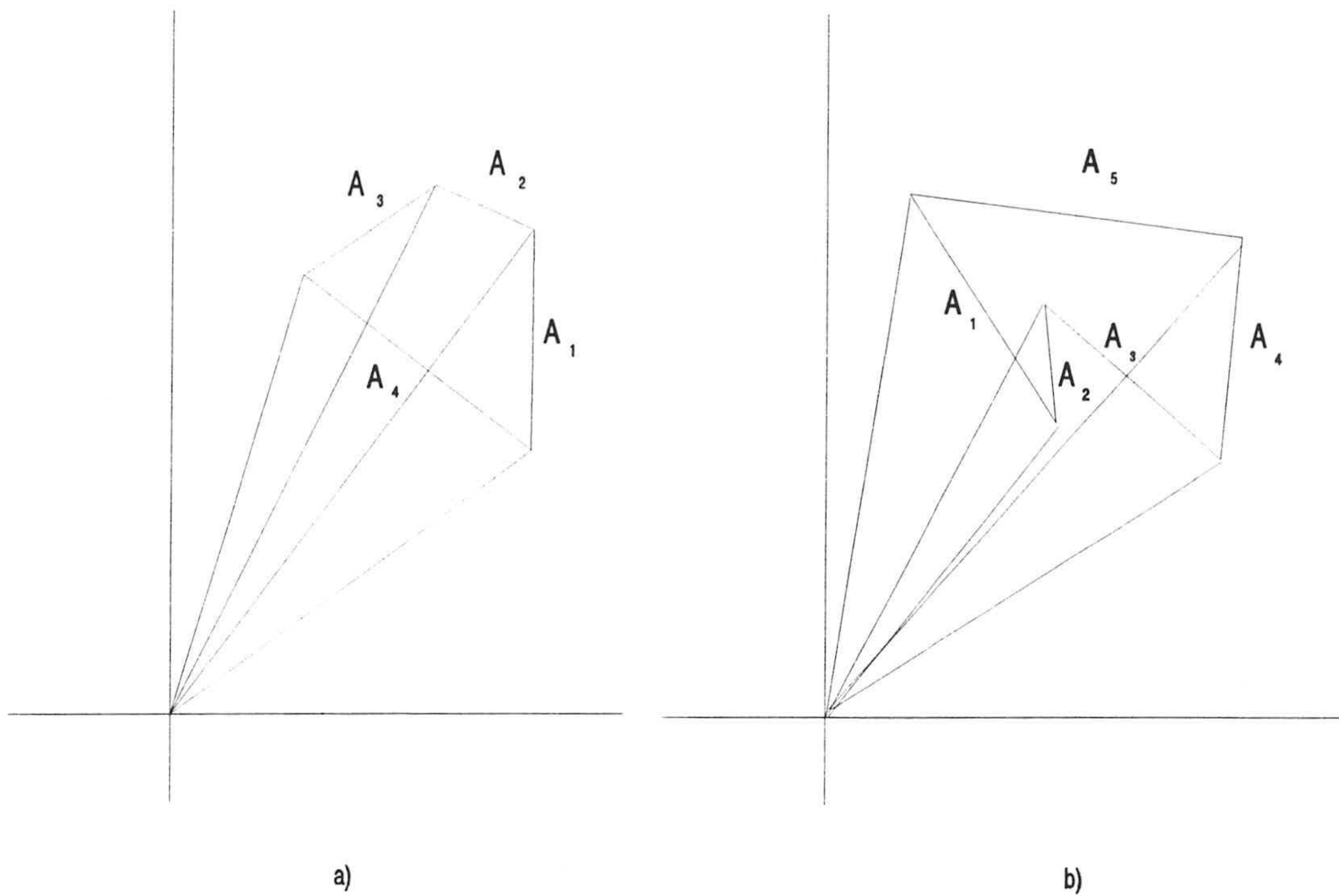


Figura 4.2 Descomposición de polígonos

Definición 4.2 (I^+ , I^- , y I_i): Dado un sólido bidimensional de n aristas $P=A_1A_2A_3\dots A_n$ definimos su conjunto de subíndices positivos I^+ , como

$$I^+ = \{ i \mid \text{sign}(OA_i) = 1 \}$$

Su conjunto de los subíndices negativos I^- , es

$$I^- = \{ i \mid \text{sign}(OA_i) = -1 \}$$

Para cada $i \in I^+$, el conjunto de índices subordinados negativos es

$$I_i = \{ j \in I^- \mid OA_j \triangleleft OA_i \}$$

En base a estos conceptos podemos expresar cualquier sólido bidimensional, definido adecuadamente por su frontera, como una expresión booleana de triángulos originales.

Teorema 4.1. Todo un sólido bidimensional de n aristas $P=A_1A_2A_3..A_n$, orientado positivamente, puede ser expresado como

$$P = \bigcup_{i \in I^*} (T_i -^* (\bigcup_{j \in I_i} T_j))$$

Donde $T_i = OA_i$

El símbolo "*" indica que estamos usando operaciones regularizadas. Esto es importante en orden a evitar situaciones extrañas en los puntos de la frontera de los triángulos originales.

Demostración

Comenzaremos demostrando la inclusión del primer término en el segundo. Probaremos que cualquier punto Q que pertenezca al interior de P está también contenido en el conjunto del segundo miembro (ya que los dos conjuntos son cerrados y acotados, con esto demostraremos la inclusión de P en el segundo miembro).

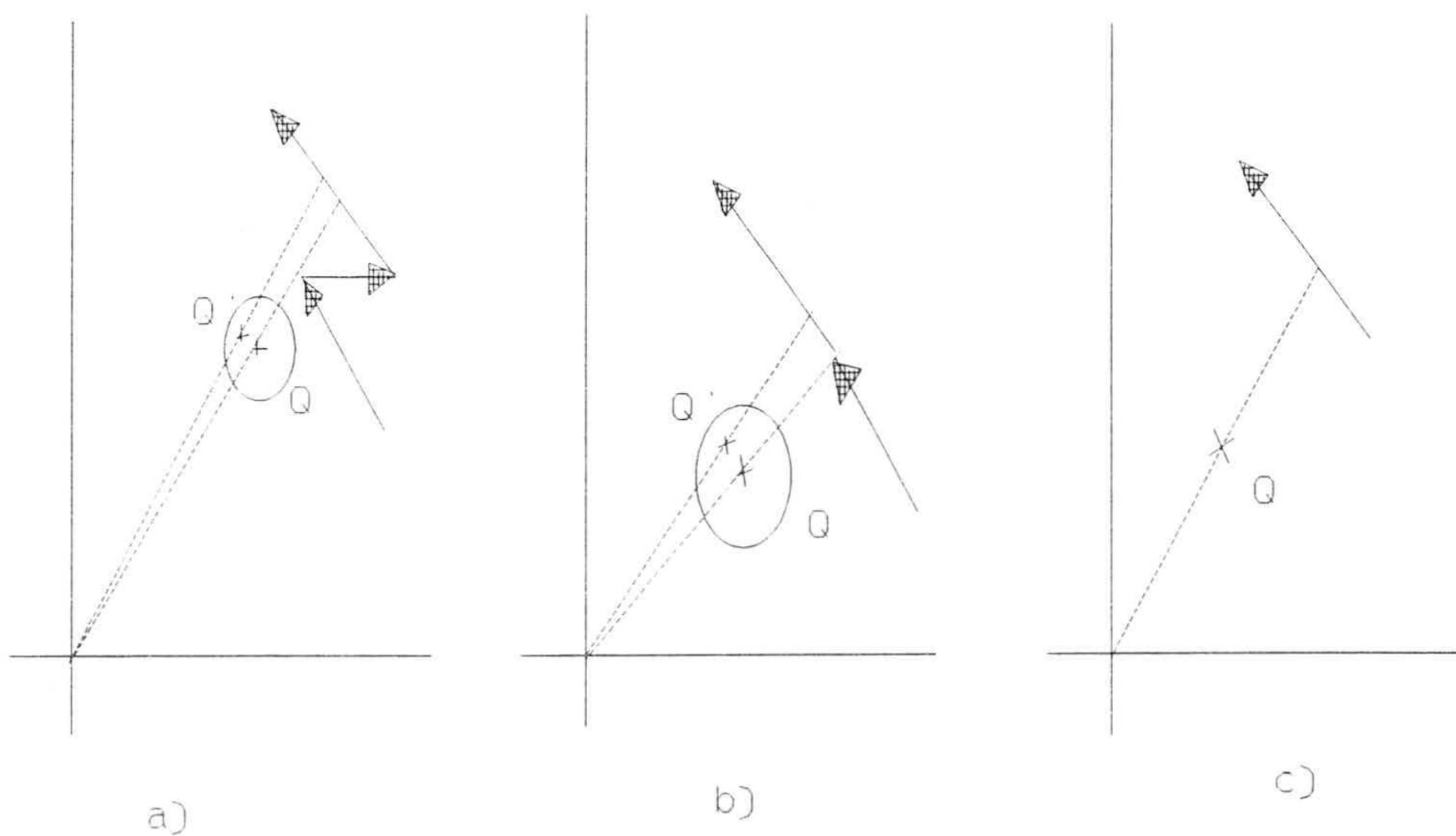


Figura 4.3 Demostración teorema 4.1

Una semirrecta que parta del origen de coordenadas \mathbf{O} y pase a través de \mathbf{Q} intersectará una primera arista entre \mathbf{Q} e infinito, que necesariamente formará un triángulo original positivo T_i . Pueden darse tres casos según se ve en la figura 4.3.

- a) La línea intersecta en un vértice común a un triángulo negativo y a uno positivo.
- b) La línea intersecta un vértice común a dos triángulos positivos.
- c) La línea no intersecta ningún vértice.

En el caso c) se tiene:

$$Q \in T_i -^* (\bigcup_{j \in I_i} T_j) \text{ y por tanto } Q \in \bigcup_{i \in I^+} (T_i -^* (\bigcup_{j \in I_i} T_j))$$

ya que $Q \in T_i$ y no pertenece a ningún triángulo subordinado a T_i .

En los casos a) y b), ya que \mathbf{Q} pertenece al interior de P , en todo entorno de \mathbf{Q} existirá un punto \mathbf{Q}' que verifica el caso c) por lo que \mathbf{Q}' pertenecerá al segundo miembro. \mathbf{Q} será de acumulación del segundo miembro y por ser este cerrado y acotado, \mathbf{Q} estará contenido en el segundo miembro.

Probaremos ahora la inclusión del segundo término en el primero. Para todo punto \mathbf{Q} contenido en el conjunto definido por el segundo miembro existirá al menos un "i", tal que:

$$Q \in T_i -^* (\bigcup_{j \in I_i} T_j)$$

Así $Q \in T_i$, y por lo tanto pertenece a la frontera ó al interior de este triángulo original; y \mathbf{Q} no pertenece a ningún triángulo subordinado a T_i . Un semirrecta que parta del origen, pase por \mathbf{Q} y se dirija al infinito, no corta a ninguna arista entre \mathbf{Q} y la arista no original de T_i ; ya que este triángulo es positivo dicho corte implica una transición de dentro hacia fuera. Concluimos por tanto que $Q \in P$. ■

La similitud entre la descomposición que se está realizando y los teoremas demostrados en el capítulo dos es evidente. Lo mismo puede decirse de las demostraciones.

Por medio de la descomposición anterior obtendremos una partición de P, lo que es importante para el estudio de determinados algoritmos ó propiedades. En algunos casos puede ser más interesante intentar simplificar esta descomposición, tal y como veremos posteriormente.

Veamos el proceso anterior con algunos ejemplos. El primer paso es calcular para el sólido bidimensional P los conjuntos I^+ , I^- y para cada $i \in I^+$, el conjunto de subíndices subordinados I_i . Estos conjuntos son, para la figura 4.2 a), los siguientes:

$$I^+ = \{ 1, 2, 3 \}$$

$$I^- = \{ 4 \}$$

$$I_1 = \{ 4 \}$$

$$I_2 = \{ 4 \}$$

$$I_3 = \{ 4 \}$$

Por lo tanto, P puede ser expresado como

$$P = (T_1 \text{ -* } T_4) \cup (T_2 \text{ -* } T_4) \cup (T_3 \text{ -* } T_4)$$

Esta expresión puede simplificarse, dando la siguiente:

$$P = (T_1 \cup T_2 \cup T_3) \text{ -* } T_4$$

Siguiendo el mismo proceso para la figura 4.2 b) obtenemos:

$$I^+ = \{ 2, 4, 5 \}$$

$$I^- = \{ 1, 3 \}$$

$$I_2 = \{ 1 \}$$

$$I_4 = \{ 3 \}$$

$$I_5 = \{ 1, 3 \}$$

y P se puede expresar como:

$$P = (T_2 \text{ -}^* T_1) U^* (T_4 \text{ -}^* T_3) U^* (T_5 \text{ -}^* (T_1 U^* T_3))$$

En este último caso no es tan sencilla la búsqueda de una simplificación.

Hemos demostrado un método que nos permite expresar cualquier sólido bidimensional en función de elementos singulares muy sencillos. La descomposición (dependiente del punto de referencia elegido) no es minimal, ya que algunos elementos aparecen repetidos. Más adelante intentaremos resolver esta cuestión.

4.2 Recubrimientos en tres dimensiones

Dentro de esta sección supondremos que estamos trabajando con sólidos tridimensionales formados por caras planas y que tiene una orientación conforme. Ello implica que sus caras son polígonos simples, que pueden tener agujeros; también que el vector normal a cada cara se dirige hacia fuera del poliedro, y por tanto que los vértices de cada cara están dados en sentido positivo, vistos desde fuera del poliedro.

En primer lugar vamos a usar como elementos recubridores pirámides originales, que son elementos sencillos y fáciles de usar. Tienen el inconveniente de no ser uniformes, y de poder tener bases definidas por polígonos no convexos, lo que complica algunas operaciones. A pesar de esto pueden ser usadas en una primera fase de muchos problemas de modelado. En caso necesario, se pueden usar tetraedros, tal y como veremos posteriormente.

Un sólido tridimensional puede ser descrito por una lista de caras, $P = C_1 C_2 \dots C_n$. Así mismo, una pirámide puede ser descrita por medio de su cúspide y de las aristas que determinan su base.

Usando tres vértices consecutivos de una cara podríamos calcular la ecuación del plano que la contiene. Ya que esto produce problemas con la orientación del vector normal al plano, así como problemas de fiabilidad al usar aritmética real, es mejor usar el método de Newell [Newel93].

$$A = 1/2 \sum_{i=1}^n (y_i + y_{i \oplus 1})(z_{i \oplus 1} - z_i)$$

$$B = -1/2 \sum_{i=1}^n (x_i + x_{i \oplus 1})(z_{i \oplus 1} - z_i)$$

$$C = 1/2 \sum_{i=1}^n (x_{i \oplus 1} - x_i)(y_i + y_{i \oplus 1})$$

donde $P_i = (x_i, y_i, z_i)$ son los vértices de la cara.

El valor de D se calcula sustituyendo uno de los puntos en la ecuación obtenida.

Al igual que en el caso de dos dimensiones vamos a dar una un lema y una serie de definiciones que nos permitirán, más adelante, descomponer un sólido en pirámides originales.

En primer lugar daremos un lema que nos permite calcular fácilmente el signo de una pirámide.

Lema 4.1 Supongamos que $P = OC_1C_2..C_n$, es una pirámide original y $Ax + By + Cz + D = 0$ la ecuación del plano que contiene a la base; entonces el signo de la pirámide, $\text{sign}(P)$, es $+1$ si D es negativo, -1 si D es positivo y 0 si D es cero.

Demostración:

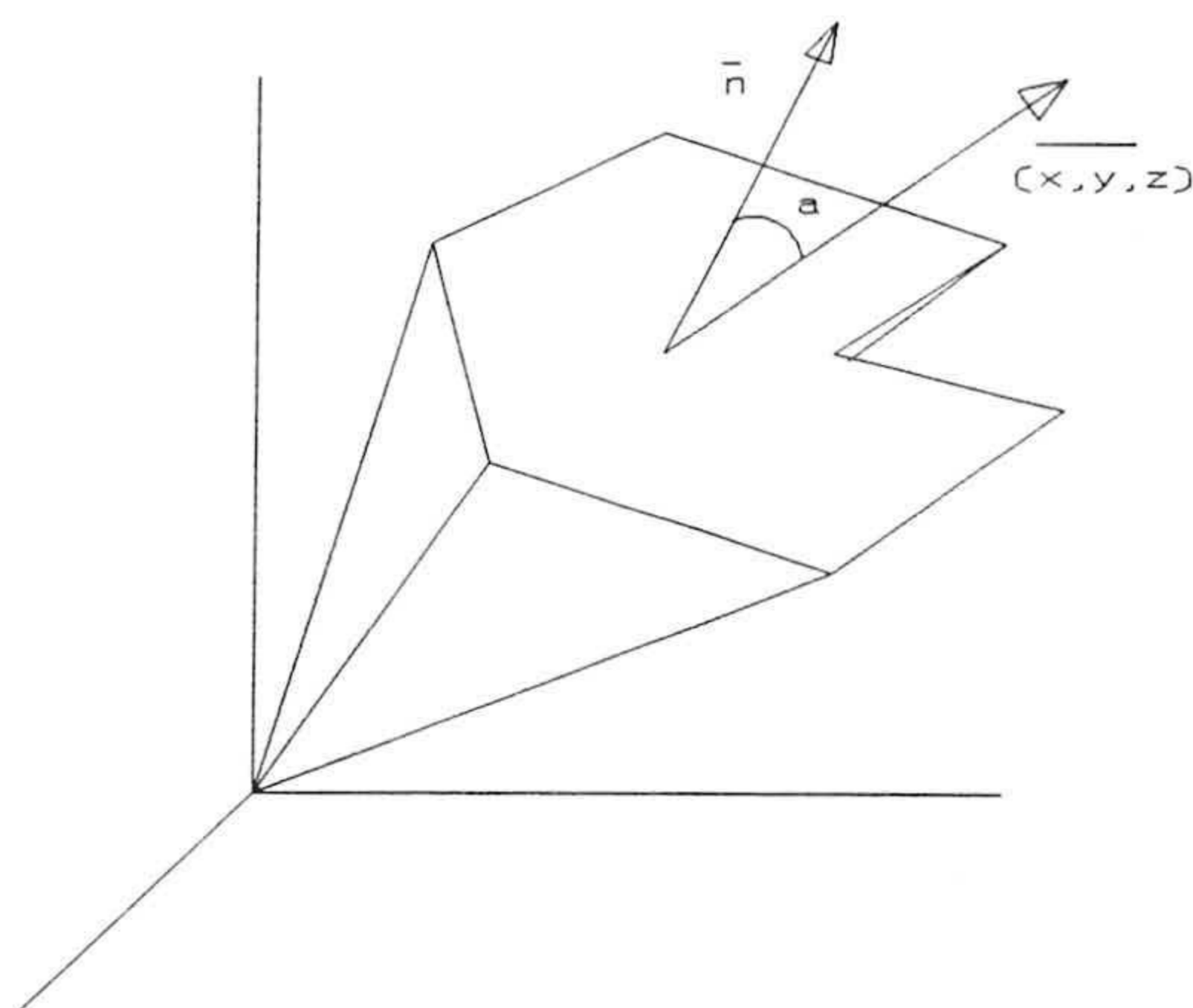


Figura 4.4 Demostración lema 4.1

Sea $Q=(x,y,z)$ un punto del plano que contiene a la base. El vector OQ (donde O es el origen) es (x,y,z) . El signo del coseno del ángulo que forma este vector y el vector normal al plano (A,B,C) , puede ser calculado por medio del signo del producto escalar:

$$(A,B,C) \cdot (x,y,z) = Ax + By + Cz$$

y esta expresión es igual a

$-D$ (ya que Q está en el plano). Si el ángulo es mayor de 90° , el signo del coseno es negativo y por tanto D es positivo. Si el ángulo es menor de 90° , el signo del coseno es positivo y D es negativo.

Si el plano pasa a través del origen el valor de D es cero y se tiene un pirámide degenerada. ■

Podemos definir ahora una relación entre pirámides originales que nos permitirá organizarlas adecuadamente:

Definición 4.3 (Pirámides subordinadas): Dadas dos pirámides originales $P=OC_1C_2..C_n$ y $P'=OC'_1C'_2..C'_n$, diremos que P' está subordinada a P , $P' \triangleleft P$ (o que P es dominante de P' , $P \triangleright P'$) si el interior (bidimensional) de la superficie $C'_1C'_2..C'_n$ intersecta el interior de la pirámide P .

Si la pirámide P' está contenida en P diremos que P' está totalmente subordinada a P . Denominaremos aristas originales y caras originales a aquellas que tienen un vértice en el origen, y no originales al resto.

De hecho, sólo estamos interesados en el estudio de pirámides originales cuyas bases se cortan a lo sumo en un vértice ó en una arista. De este modo, es relativamente fácil comprobar si dos pirámides originales cuyas bases no se cortan están subordinadas o no, aunque hay algunos casos que requieren un tratamiento especial. Posteriormente veremos algunos algoritmos que facilitan este cálculo.

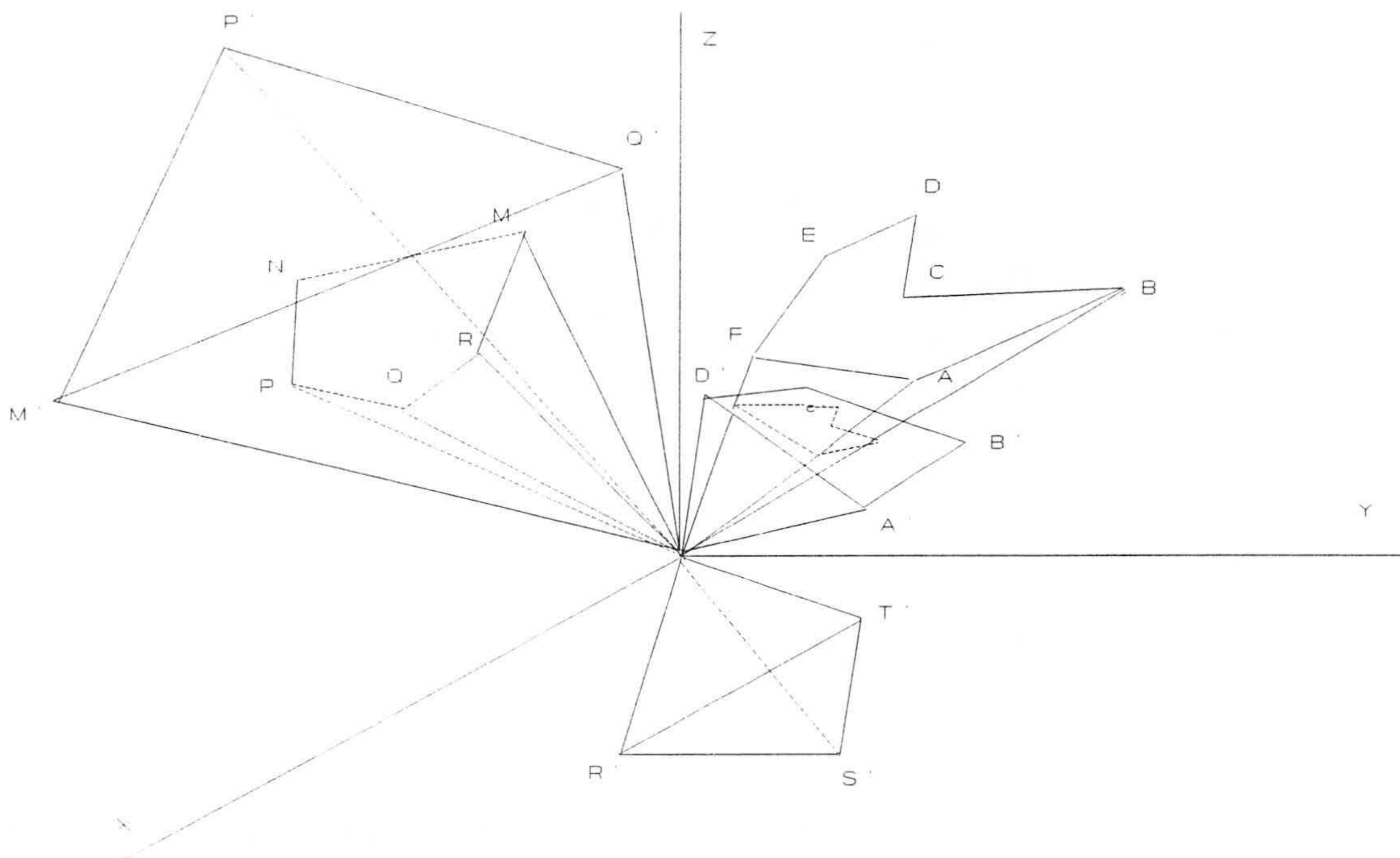


Figura 4.5 Pirámides subordinadas

En la figura 4.5 se muestran varias pirámides subordinadas. Por ejemplo, $OA'B'C'D' \triangleleft OABCDEF$; $OMNPQR \triangleleft OP'M'Q'$.

En algunos casos, el poliedro puede definirse como la unión de las pirámides originales de signo positivo menos la unión de las de signo negativo.

Sin embargo, como se ve en la figura 4.6 si existen concavidades, esto no es siempre posible.

El método para la descomposición de sólidos tridimensionales es una generalización del método estudiado previamente para dos dimensiones. Básicamente se trata de construir un trozo por cada una de las pirámides originales positivas. Este trozo se define como dicha pirámide positiva menos la unión de todas sus pirámides subordinadas.

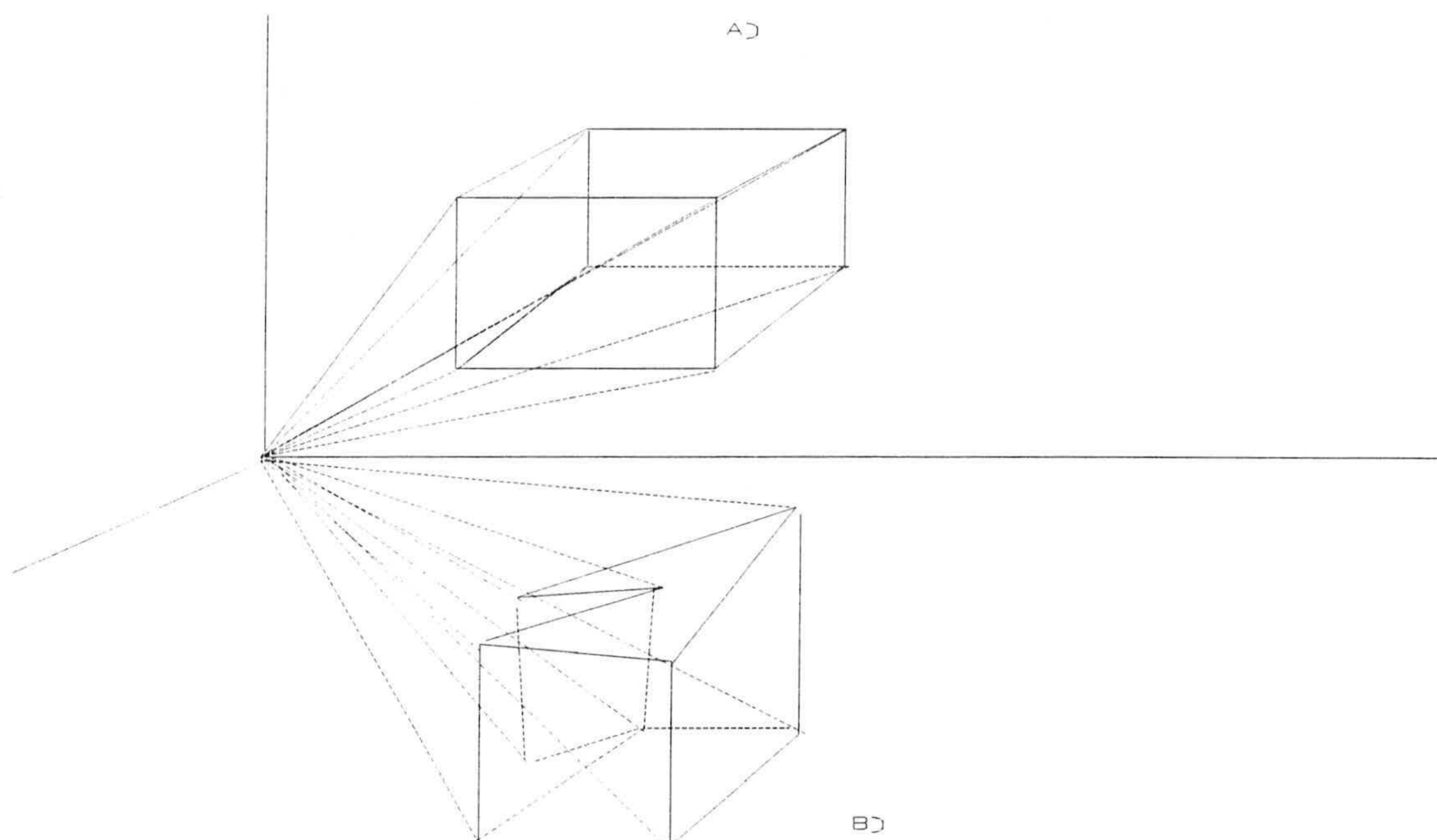


Figura 4.6 Descomposición de poliedros

En orden a dar una definición formal de esta descomposición introducimos las siguientes definiciones:

Definición 4.4 (I^+ , I^- , I_i) Dado un sólido de n caras $C_1, C_2, C_3, \dots, C_n$ definimos el conjunto de subíndices positivos I^+ , como:

$$I^+ = \{i \mid \text{sign}(OC_i) = 1 \}$$

donde OC_i es una pirámide original.

El conjunto de subíndices negativos I^- , se define como:

$$I^- = \{i \mid \text{sign}(OC_i) = -1 \}$$

Para cada $i \in I^+$, su conjunto de índices negativos subordinados se define como

$$I_i = \{j \in I^- \mid OC_j \triangleleft OC_i\}$$

Los conjuntos anteriores son, en general, fáciles de calcular. En algunos casos, aumentan las complicaciones y puede ocurrir que no sea nada sencillo determinar si una pirámide es o no subordinada a otra. Como ya se ha indicado, al final de este capítulo veremos algoritmos para realizar este estudio.

El siguiente teorema demuestra una forma de descomposición, permitiendo expresar un sólido tridimensional representado por su frontera por medio de una expresión booleana de pirámides originales.

Teorema 4.2 Dado un sólido de n caras $P = C_1 C_2 C_3 \dots C_n$, con una orientación conforme, se tiene que:

$$P = \bigcup_{i \in I^+}^* (P_i -^* (\bigcup_{j \in I_i}^* P_j))$$

Donde P_i representa la pirámide original OC_i .

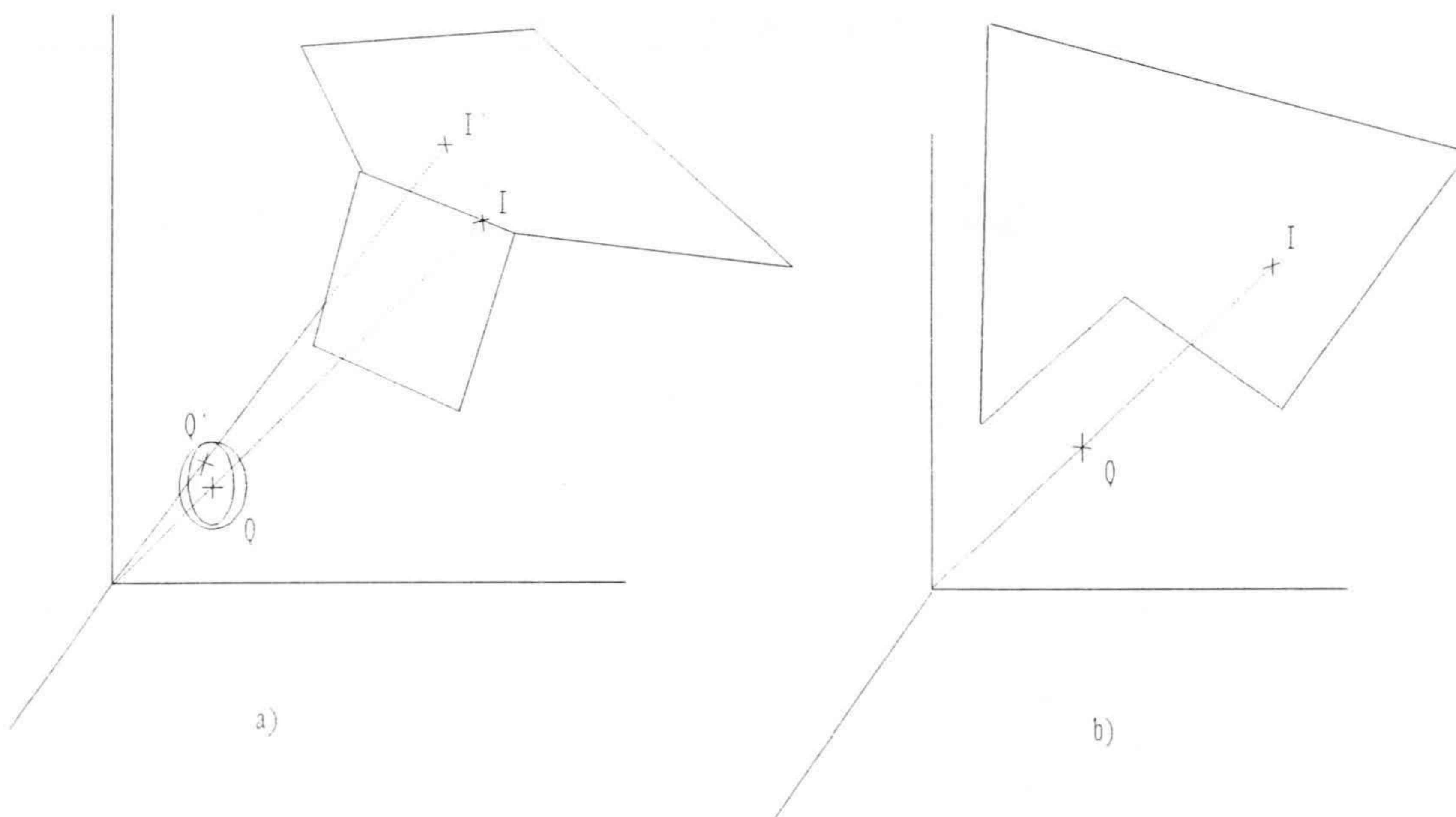


Figura 4.7 Demostración teorema 4.2

Demostración:

Seguiremos un proceso similar a la demostración del teorema 4.1. Primeramente, probaremos la inclusión del conjunto indicado por el primer miembro en el segundo. Demostraremos que cualquier punto Q perteneciente al interior de P pertenece al conjunto indicado por el segundo miembro de la igualdad (ya que los dos conjuntos son cerrados y acotados, esto determinará la inclusión de P en el segundo término).

Una semirrecta que parta del origen de coordenadas y pase por Q intersectará, después de pasar por Q , una primera cara en un punto que necesariamente formará un pirámide positiva P_i . Pueden ocurrir tres casos, como se ve en la figura 4.7:

- a) La línea intersecta una arista común al menos a dos pirámides,
- b) La línea no intersecta ninguna arista

En el caso b), evidentemente

$$Q \in P_{i-}^* \left(\bigcup_{j \in I_i}^* P_j \right) \text{ y por tanto } Q \in \bigcup_{i \in I^+}^* \left(P_{i-}^* \left(\bigcup_{j \in I_i}^* P_j \right) \right)$$

En el caso a), al menos una de las pirámides debe ser positiva, y ya que Q pertenece al interior de P , en todo entorno de Q existirá un punto Q' que verifique el caso b). Por lo tanto Q será punto de acumulación del conjunto indicado por el segundo miembro, y por ser este conjunto cerrado y acotado, Q estará contenido en él.

Demostraremos ahora la inclusión inversa. Para cualquier punto Q contenido en el conjunto determinado por el segundo miembro, existirá al menos un "i" tal que :

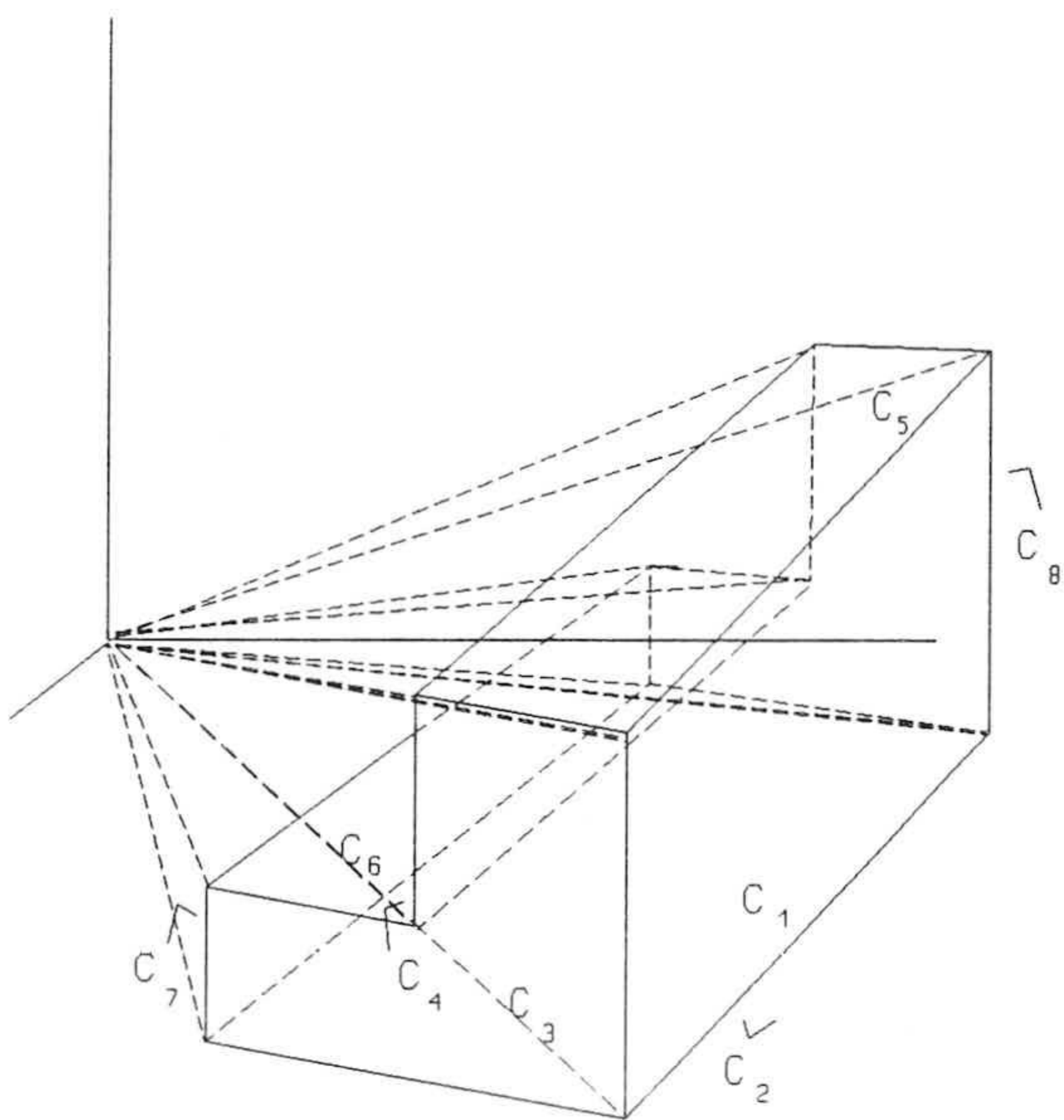
$$Q \in P_{i-}^* \left(\bigcup_{j \in I_i}^* P_j \right)$$

Así, Q pertenece a P_i (a su interior ó a su frontera) y no pertenece a

ninguna pirámide subordinada a P_i . Si trazamos una semirrecta que parta del origen, pase por Q y se dirija al infinito, entre Q y la cara no original de P_i no se cortará ninguna otra cara y ya que la transición anterior es de dentro hacia fuera, concluimos que Q pertenece a P . ■

Esta descomposición genera una partición de P . En muchos casos puede ser más interesante simplificar la descomposición obtenida.

Veamos con algunos ejemplos el proceso de descomposición. El primer paso es calcular los conjuntos I^+ , I^- , y para cada $i \in I^+$, sus pirámides subordinadas P_j . Estos conjuntos, para el poliedro mostrado en la figura 4.8 son:



$$I^+ = \{ 1, 3, 5, 6, 8 \}$$

$$I^- = \{ 4, 7 \}$$

$$I_1 = \{ 4, 7 \}$$

$$I_3 = \{ 4, 7 \}$$

$$I_5 = \{ 4 \}$$

$$I_6 = \{ 7 \}$$

$$I_8 = \{ 4, 7 \}$$

Figura 4.8 Descomposición de un poliedro

Por tanto el poliedro puede ser expresado como

$$P = (P_1 -^*(P_4 \cup P_7)) \cup (P_3 -^*(P_4 \cup P_7)) \cup (P_5 -^* P_4) \cup (P_6 -^* P_7) \cup (P_8 -^*(P_4 \cup P_7))$$

En orden a expresar un poliedro como una expresión booleana de primitivas (similar por tanto a un árbol CSG) puede ser más interesante usar tetraedros como primitivas. Esto puede hacerse descomponiendo el poliedro en pirámides y expresando las pirámides como unión de varios tetraedros, ó bien descomponiendo directamente el poliedro en tetraedros. Una vez realizadas las demostraciones anteriores es más sencillo, y consecuente, demostrar que toda pirámide original puede ser expresada en función de tetraedros originales. Por otro lado, no es fácil determinar un algoritmo que tetraedrice un sólido de modo que pueda ser aplicado de modo general.

Hay dos posibilidades para obtener la descomposición de una pirámide original en tetraedros originales:

a) Se realiza una triangulación de la base de la pirámide original. Así podemos obtener tetraedros cuya intersección regularizada sea vacía. El problema aquí es encontrar el mejor método para realizar la triangulación ya que en muchos casos esta triangulación puede ser muy complicada. Esto también implicaría que no se tratarían de modo uniforme todos los sólidos. El uso de triangulaciones y tetraedrizaciones ha sido ampliamente estudiado ([PauRs93], [Dey__91] y [Shape91]).

b) Se elige un punto arbitrario del plano que contiene a la base de la pirámide y se descompone la base en triángulos usando el método presentado en la sección 1 de este capítulo, considerando como origen el punto elegido.

Vemos más adecuado usar la opción b) ya que se corresponde con el desarrollo formal realizado en el capítulo II.

La elección del punto arbitrario puede hacerse de muchos modos, ya que la validez del método no depende del punto elegido. Una solución sencilla es elegir uno de los vértices de la base de la pirámide (por ejemplo el primero referenciado) ó bien elegir otro punto singular del plano que contiene a la base (por ejemplo, el centro de gravedad de los vértices de la base de la pirámide). En lo que sigue se asume que se toma el primer vértice referenciado de cada cara.

Es necesario adaptar las definiciones anteriores, ya que en el caso de tetraedros pertenecientes a una misma cara, sus caras no originales pueden intersectarse. Daremos, por tanto, las siguientes definiciones

Definición 4.5 (Tetraedros subordinados): Dados dos tetraedros originales $T=OP_1P_2P_3$ y $T'=OQ_1Q_2Q_3$, diremos que T' está subordinado a T , $T' \triangleleft T$ (o que T es dominante de T' , $T \triangleright T'$) si $T' \triangleleft T$ considerados como pirámides originales o en el caso de que ambos tetraedros pertenezcan a la misma pirámide (tienen un vértice común y todos los vértices no originales son coplanarios), si la proyección sobre cualquiera de los planos de coordenadas del triángulo $Q_1Q_2Q_3$ está subordinada a la proyección sobre el mismo plano de $P_1P_2P_3$, considerando como origen a P_1 (que coincide con Q_1).

Definiremos, seguidamente, los conjuntos de subíndices asociados a los tetraedros originales.

Definición 4.6 (I^+ , I^- , I_i) Dada una pirámide original cuya base está formada por las aristas $A_1, A_2, A_3 \dots A_n$, siendo $V_{i,1}$ y $V_{i,2}$ los vértices de la arista A_i , definimos el conjunto de subíndices positivos I^+ , como:

$$I^+ = \{i \mid \text{sign}(OV_{1,1}V_{i,1}V_{i,2}) = 1 \}$$

donde $OV_{1,1}V_{i,1}V_{i,2}$ es un tetraedro original.

El conjunto de subíndices negativos I^- , se define como:

$$I^- = \{i \mid \text{sign}(OV_{1,1}V_{i,1}V_{i,2}) = -1 \}$$

Para cada $i \in I^+$, su conjunto de índices negativos subordinados se define como

$$I_i = \{j \in I^- \mid OV_{1,1}V_{j,1}V_{j,2} \triangleleft OV_{1,1}V_{i,1}V_{i,2}\}$$

A partir de las definiciones anteriores, podemos enunciar el siguiente teorema:

Teorema 4.3 Dada una pirámide original cualquiera de base $P = A_1 A_2 A_3 \dots A_n$, siendo A_i las aristas y $V_{i,1}, V_{i,2}$ los vértices de A_i , y con una orientación conforme se cumple que:

$$P = \bigcup_{i \in I^+}^* (T_i -^* (\bigcup_{j \in I_i}^* T_j))$$

Donde T_i representa el tetraedro original $OA_{1,1}A_{i,1}A_{i,2}$.

Demostración:

Evidente, aplicando el teorema 4.1 a la base de la pirámide, tomando como origen del plano que contiene a la base el punto $V_{1,1}$. ■

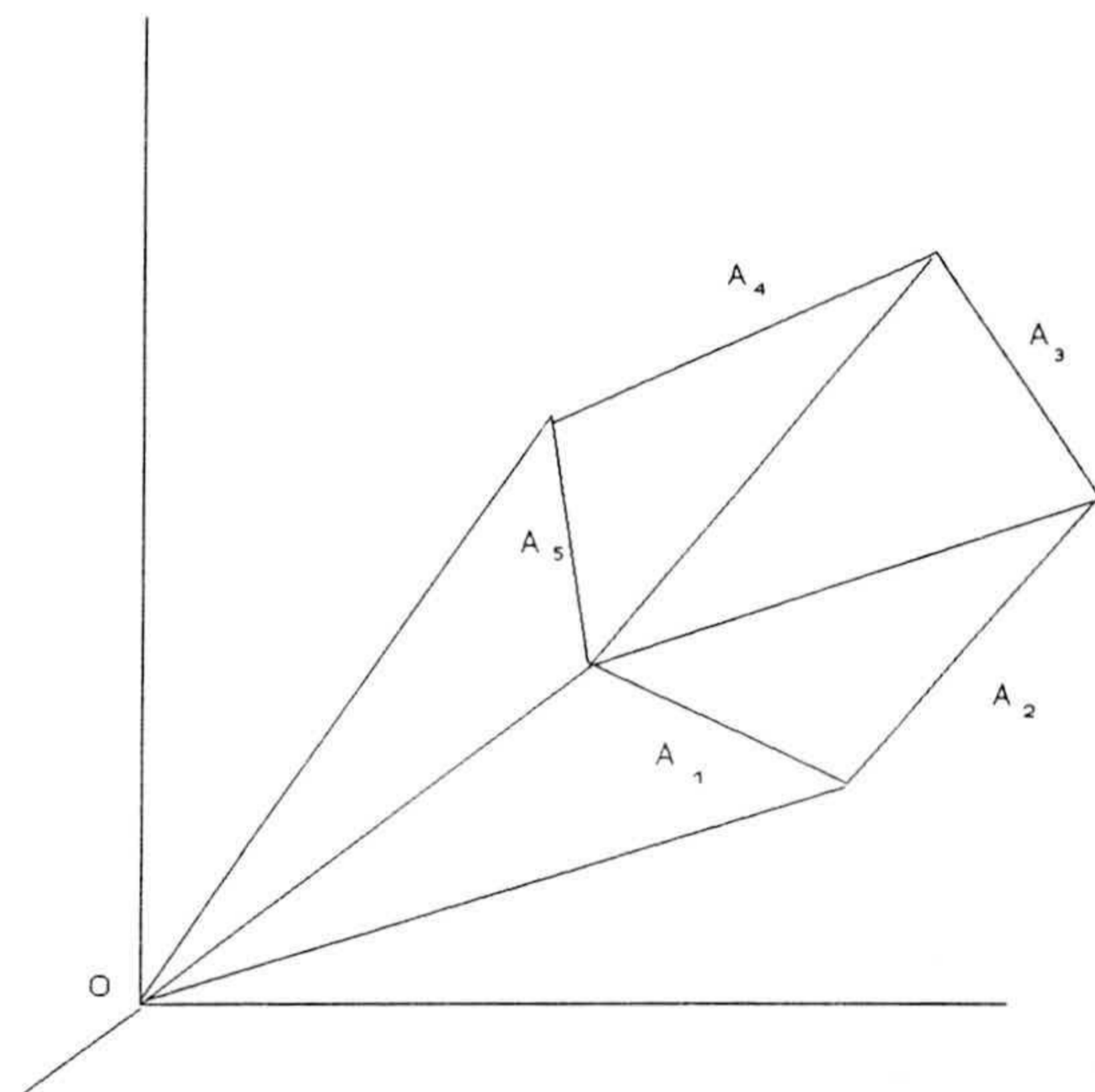


Figura 4.9 Descomposición de una pirámide original

de simplificar la expresión obtenida.

En la figura 4.9 puede observarse un ejemplo de descomposición de una pirámide original en tetraedros originales.

Mediante la aplicación sucesiva de los teoremas 4.2 y 4.3, obtendríamos la descomposición de cualquier sólido en función de tetraedros originales.

La aplicación sucesiva se completaría con un estudio de subordinación de tetraedros originales pertenecientes a pirámides originales subordinadas al objeto

Si hubiésemos partido de los tetraedros las fórmulas que se habrían obtenido no serían idénticas, aunque sí equivalentes.

Al igual que ocurre en el caso bidimensional es posible que las primitivas (pirámides o tetraedros originales) sean redundantes. En el siguiente apartado se estudiará la obtención de expresiones canónicas.

4.3 Recubrimientos simpliciales canónicos. Introducción

Una vez establecido en los puntos anteriores que todo sólido (tanto bidimensional como tridimensional) puede expresarse a partir de su frontera como un árbol CSG cuyas primitivas son tetraedros, considerando símlices como primitivas, se plantean dos preguntas evidentes: primero, la posibilidad de obtener expresiones CSG lo más simplificadas posible y a ser posible "canónicas" (es decir, independientes del modo de haber obtenido el árbol), y, segundo, la posibilidad de realizar las operaciones booleanas sobre esta representación directamente.

La primera pregunta tiene una respuesta clara en relación a los métodos usualmente utilizados en la conversión de B-rep a CSG. Describiremos a continuación el modo de obtener un árbol CSG, único salvo en el orden, para cada objeto bi ó tridimensional. Posteriormente describiremos y demostraremos la obtención de expresiones CSG a partir de una ó varias operaciones de objetos.

4.3.1 Descomposiciones minimales

En general, en casi todos los trabajos que buscan la conversión de B-rep a CSG, se toman como primitivas los semiespacios determinados por las caras, ordenadas las aristas de estas según un criterio prefijado (habitualmente se ordenan de modo que el vector normal determinado por el orden de los vértices se dirija hacia fuera del sólido). En este ambiente de trabajo, se denomina fórmula estilo Peterson a aquella que es monótona (no aparece la complementación) y en la que cada primitiva aparece una sola vez. Así mismo, y hasta ahora, se había demostrado que no existe una fórmula estilo Peterson aplicable a poliedros generales en 3 dimensiones, considerando como primitivas, los semiespacios [DoGHS88]. Sí ha sido resuelto el problema usando más primitivas de las que definen el sólido (semiespacios separadores de Shapiro [ShaVo93]).

Manteniendo el sentido anterior, lo que buscamos en nuestra representación es una simplificación que nos de una fórmula estilo Peterson usando los triángulos, pirámides o tetraedros como primitivas y que pueda ser aplicada en tres dimensiones.

Al objeto de facilitar mejor la comprensión haremos por separado el estudio en 2 y 3 dimensiones, aunque como se podrá comprobar, la fórmula y métodos seguidos son esencialmente los mismos: el caso de tres dimensiones es una generalización natural del caso en dos dimensiones.

4.3.1.1 Descomposición canónica en dos dimensiones

En la fórmula obtenida en el teorema 4.1 puede observarse claramente que algunos triángulos originales pueden aparecer más de una vez. Esto ocurre en concreto con los triángulos originales negativos que sean a la vez subordinados de más de un triángulo positivo. En algunos casos, podría eliminarse uno de los triángulos, ó podría descomponerse alguno de los triángulos en dos triángulos contiguos. De cualquier modo, estos métodos no serían siempre aplicables, ó provocarían una mayor complicación del modelo.

Al objeto de simplificar al máximo la fórmula indicada y conseguir a la vez obtener más fácilmente la representación del polígono por medio de un árbol binario, introducimos una cierta clasificación de todos los triángulo según su "mayor" ó "menor" nivel de profundidad. Esta idea intuitiva de nivel de profundidad se formaliza a partir de la definición de triángulo subordinado y dominante dada anteriormente.

Esta relación nos permitirá dotar a la descomposición de un orden, lo que facilitará trabajos posteriores.

Antes de proseguir en esta línea veamos una caracterización de la subordinación de triángulos originales:



Lema 4.2 Sean T y T' dos triángulos originales cuyas aristas no originales se cortan como máximo en un vértice. Si $T \triangleleft T'$, toda recta que pasa por el origen y corta a las aristas no originales de los dos triángulos, distinta de las que contienen las aristas originales de los triángulos, corta primero a la de T y luego a la de T' .

Demostración:

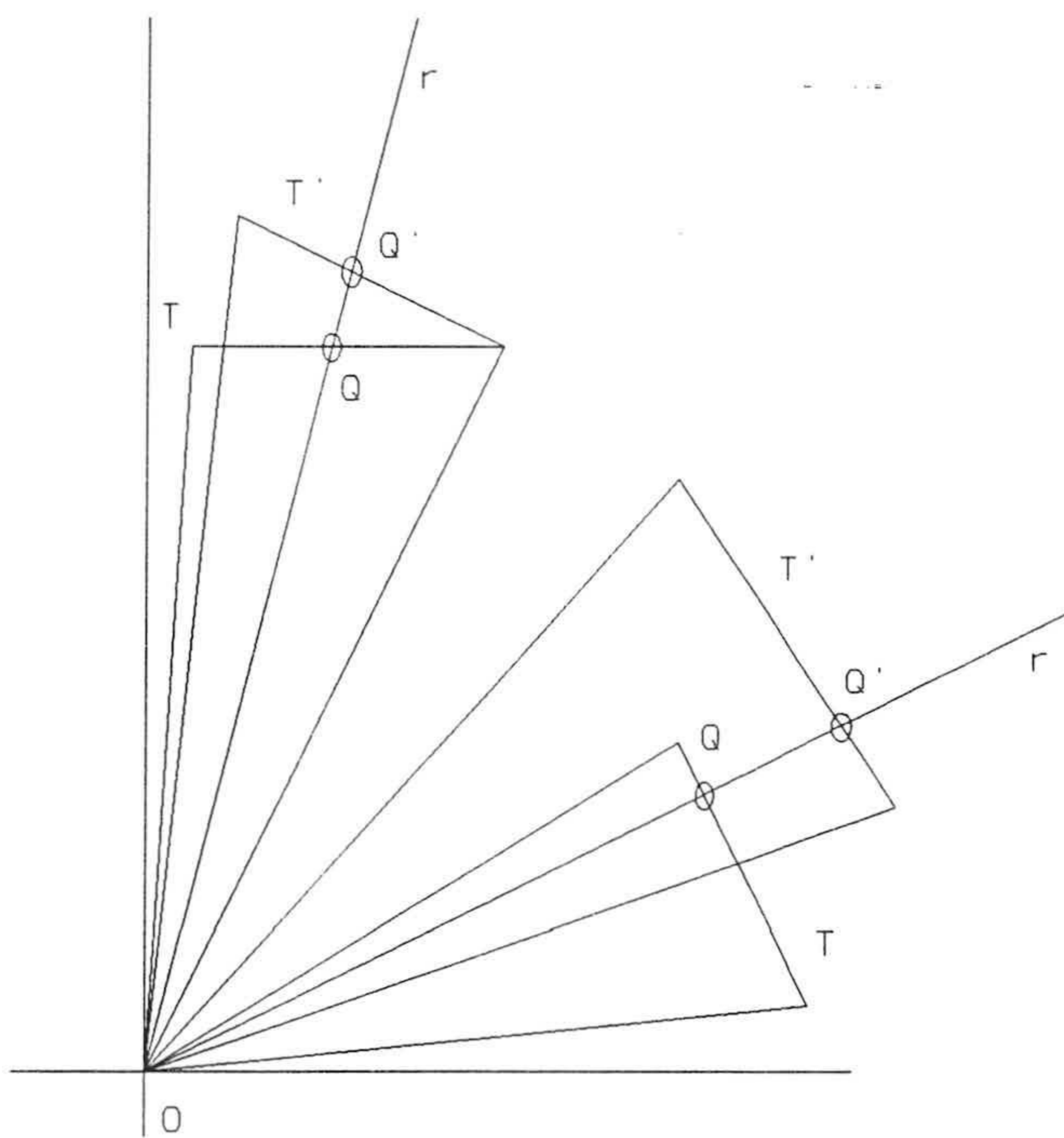


Figura 4.10 Lema 4.2

Sea r una recta cualquiera y Q y Q' los puntos de corte. Si $T \triangleleft T'$, el punto Q , de corte de r con T será necesariamente interior a T' y por tanto Q estará más cerca del origen que Q' . ■

La relación de subordinación definida en 4.1 es reflexiva (por definición) y antisimétrica (consecuencia del lema anterior), aunque no es transitiva como fácilmente puede comprobarse. Sí es transitiva la relación de subordinación completa, aunque esta relación puede que no se de entre los triángulos originales asociados a un sólido bidimensional.

Podemos generalizar la definición e intentar obtener una relación que sí tenga la propiedad transitiva. Una forma posible sería a partir de la siguiente definición:

Definición 4.7 Dado un conjunto $P = \{T_i \mid i=1, \dots, n\}$ de triángulos originales, tales que sus aristas no originales se cortan como máximo en un vértice, diremos que T_j es menor ó igual que T_k ($T_j \leq T_k$) si existe una cadena de triángulos de P , $P' = \{T_{j,0}, T_{j,1}, \dots, T_{j,m}\}$, con $T_{j,0} = T_j$, $T_{j,m} = T_k$, y tal que $T_{j,l} \triangleleft T_{j,l+1}$.

Esta relación cumple las propiedades reflexiva y transitiva como vamos a demostrar:

Lema 4.3 La relación definida en 4.7, cumple las propiedades reflexiva y transitiva:

Demostración:

Comenzaremos viendo que es reflexiva: Sea T_j un triángulo de P . Ya que todo triángulo está subordinado a sí mismo, la cadena $P' = \{T_j\}$ verifica la definición 4.7 por lo que $T_j \leq T_j$, y la relación es reflexiva.

Demostremos ahora la propiedad transitiva. Supongamos que $T_i \leq T_j$ y que $T_j \leq T_k$. Por la primera relación existe una cadena $P' = \{T_i, \dots, T_j\}$; por la segunda relación existe una cadena $P'' = \{T_j, \dots, T_k\}$. La cadena $P''' = \{T_i, \dots, T_j, \dots, T_k\}$, claramente verifica las condiciones exigidas en la definición, por lo que $T_i \leq T_k$. ■

Para que fuese una relación de orden sería necesario que cumpliese también la antisimétrica. Esto nos permitiría asociar a cada triángulo original un cierto valor de "profundidad" y obtendríamos una descomposición idónea.

La figura 4.9 muestra un ejemplo en que no se verifica la propiedad antisimétrica. Cada triángulo está subordinado al siguiente, formando una relación cíclica.

Otra posible solución es introducir un vértice más, por ejemplo V en la figura 4.11, con lo que también se resuelve el problema. Utilizaremos esta segunda posibilidad, y resolveremos el problema con la siguiente definición:

Definición 4.8 Dado un conjunto $P = \{T_i\}$, $i=1, \dots, n$, de triángulos originales, tales que sus aristas no originales se cortan como máximo en un vértice, diremos que P es no separado si existen al menos dos triángulos T_i y T_k tal que

$$T_i \leq T_k \text{ y } T_k \leq T_i$$

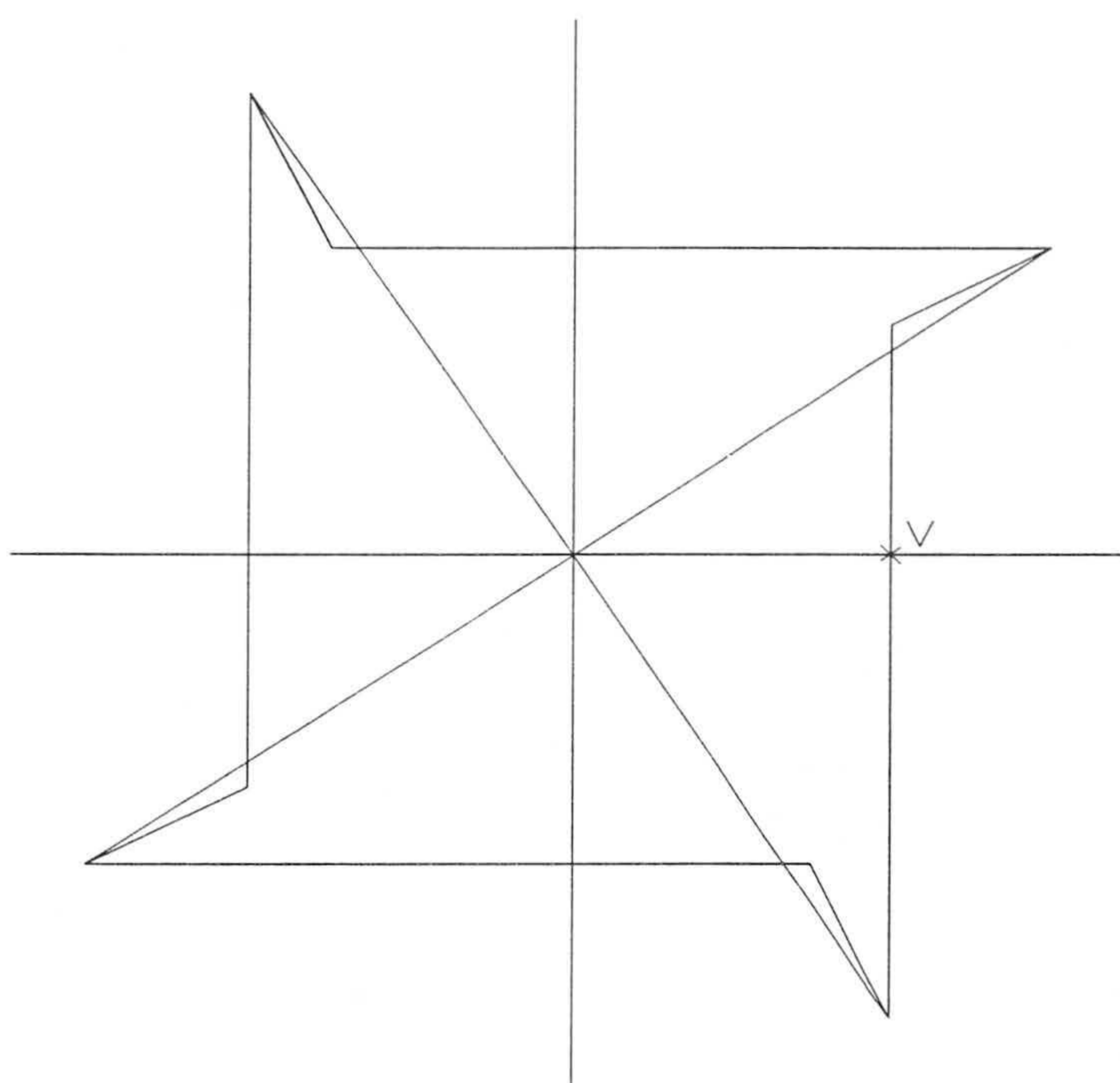


Figura 4.9 Sólido no separado

En caso contrario diremos que P es separado.

La figura 4.11 es un ejemplo de que existen conjuntos de triángulos no separados.

Vamos a demostrar a continuación que a partir de todo conjunto no separado se puede definir un conjunto separado.

Lema 4.4 Sea $P = \{T_i\}$, $i = 1, \dots, n$, un conjunto de triángulos originales no separado y T_i y T_k dos triángulos distintos tales que $T_i \leq T_k$ y $T_k \leq T_i$. Entonces existe al menos un punto en la arista no original de T_i ó de T_k tal que al partir el triángulo por ese punto se obtiene un nuevo conjunto de triángulos originales que es separado.

Demostración:

Si $T_i \leq T_k$ y $T_k \leq T_i$ existen al menos dos cadenas de triángulos tales que algunas de las relaciones de subordinación no es total.

Si fuesen todas las subordinaciones totales, por ser este tipo de subordinación antisimétrica, T_i estaría subordinado a T_k y T_k estaría subordinado a T_i , y por lo tanto $T_i = T_k$. Supongamos que T'_j está subordinado a T_i y que ésta no es completa. Por tanto existe un punto Q tal que T'_j no está subordinado a T'_i . Sustituyendo T_i por T'_i y T''_i ó bien se obtiene un conjunto separado ó se repetiría el proceso. Ya que el número de triángulos es finito, el proceso es finito y se obtendría al final un conjunto de triángulos originales separado. ■

Una vez resueltos los casos especiales, podemos demostrar que la relación definida anteriormente es una relación de orden.

Teorema 4.4 La relación definida en 4.7 es una relación de orden para conjuntos separados.

Demostración:

Evidente a partir de los lemas anteriores. ■

El teorema anterior, por tanto, nos permite introducir una relación de orden entre triángulos originales cuyas aristas no originales se cortan como máximo en un vértice. Ya que los triángulos originales asociados a un sólido bidimensional verifican que sus aristas no originales se cortan a lo sumo en un vértice, se sigue que hemos introducido una relación de orden entre los triángulos asociados a un sólido, siempre y cuando este sea separado. El orden no es total, ya que puede haber triángulos que no estén subordinados a ninguno, ni que sean dominantes de ningún otro. Esto ocurriría, por ejemplo, en un cuadrado centrado en el origen. Esos triángulos serán siempre positivos, por lo que no representan un problema a la hora de la descomposición. Por otro lado, la relación de orden nos permite representar cada sólido como una estructura jerárquica de la cual se puede extraer muy fácilmente un árbol CSG.

Así mismo, el hecho de que la relación introducida sea una relación de orden nos permite dotar a cada sólido de una estructura de retículo ([Kurat66])

Introducimos las siguientes definiciones:

Definición 4.9 (índice) Dado un sólido separado P y siendo $\{T_i \mid i=1, \dots, n\}$ el conjunto de sus triángulos originales asociados, se denomina índice de T_i ($\text{ind}(T_i)$) al nivel de profundidad en el retículo determinado por la cadena ordenada de triángulos, expresados estos de mayor a menor, y supuesto que a los elementos maximales se les asigna el valor 1 (si un elemento es minimal y maximal a la vez, como ocurre con los elementos aislados, estos se consideran de índice 1).

Definición 4.10 (capa) Dado un sólido separado P y siendo $\{T_i \mid i=1, \dots, n\}$ el conjunto de sus triángulos originales asociados, se denomina capa de índice k al conjunto de triángulos originales de T_i de índice k .

Vamos a demostrar que según las definiciones anteriores, todos los triángulos de índice 1 ó impar son positivos y que los de índice par son negativos.

Lema 4.5 Sea P un sólido y $\{T_i \mid i=1, \dots, n\}$ el conjunto de sus triángulos originales asociados. Entonces si el índice de T_i es impar, T_i es positivo y si T_i es par, es negativo.

Demostración:

Haremos la demostración por inducción.

Sea T_i un triángulo de índice 1. Quiere ello decir que T_i no está subordinado a ningún otro triángulo original por lo que toda semirrecta que parta del origen y corte a la arista no original de T_i no cortará posteriormente a ninguna otra arista de P . Ello implica que la intersección de la semirrecta indicada con la arista no original de T_i implica una transición de dentro hacia fuera, por lo que T_i es positiva.

Supongamos que T_j es de índice k ($k > 1$). Existirá por tanto un triángulo original T_i tal que $T_j \triangleleft T_i$ con índice $k-1$, de modo que toda semirrecta que parta del origen y corte a las aristas no originales de T_j y de T_i , corte primero a la arista no original de T_j y luego a la de T_i , sin que entre medias de ambas intersecciones se corte a otra arista no original de P . Quiere ello decir que si T_i es positivo, la intersección de la semirrecta con T_i implicará una transición de dentro hacia fuera; por tanto la intersección de la semirrecta con T_j indicará una transición de fuera hacia dentro y por tanto T_j es negativo.

Similarmente, si T_i fuese negativo, T_j sería positivo. ■

El siguiente teorema nos permite establecer el cálculo de árboles CSG minimales en los que cada primitiva aparece una sola vez. Para ello se va construyendo el sólido a partir de sus capas, de fuera hacia adentro.

Teorema 4.5 Sea P un sólido de aristas $\{A_1, A_2, \dots, A_n\}$ y $\{T_i \mid i=1, \dots, n\}$ el conjunto de sus triángulos originales asociados que supondremos separado. Sea C_i la capa de índice i . Entonces el sólido P se puede expresar como:

$$P = (((C_1 \text{ - } C_2) \cup C_3) \text{ - } C_4) \dots \cup C_{2N-1} \text{ - } C_{2N}$$

$$\text{siendo } C_i = \cup \{T_j \mid \text{ind}(T_j) = i\}$$

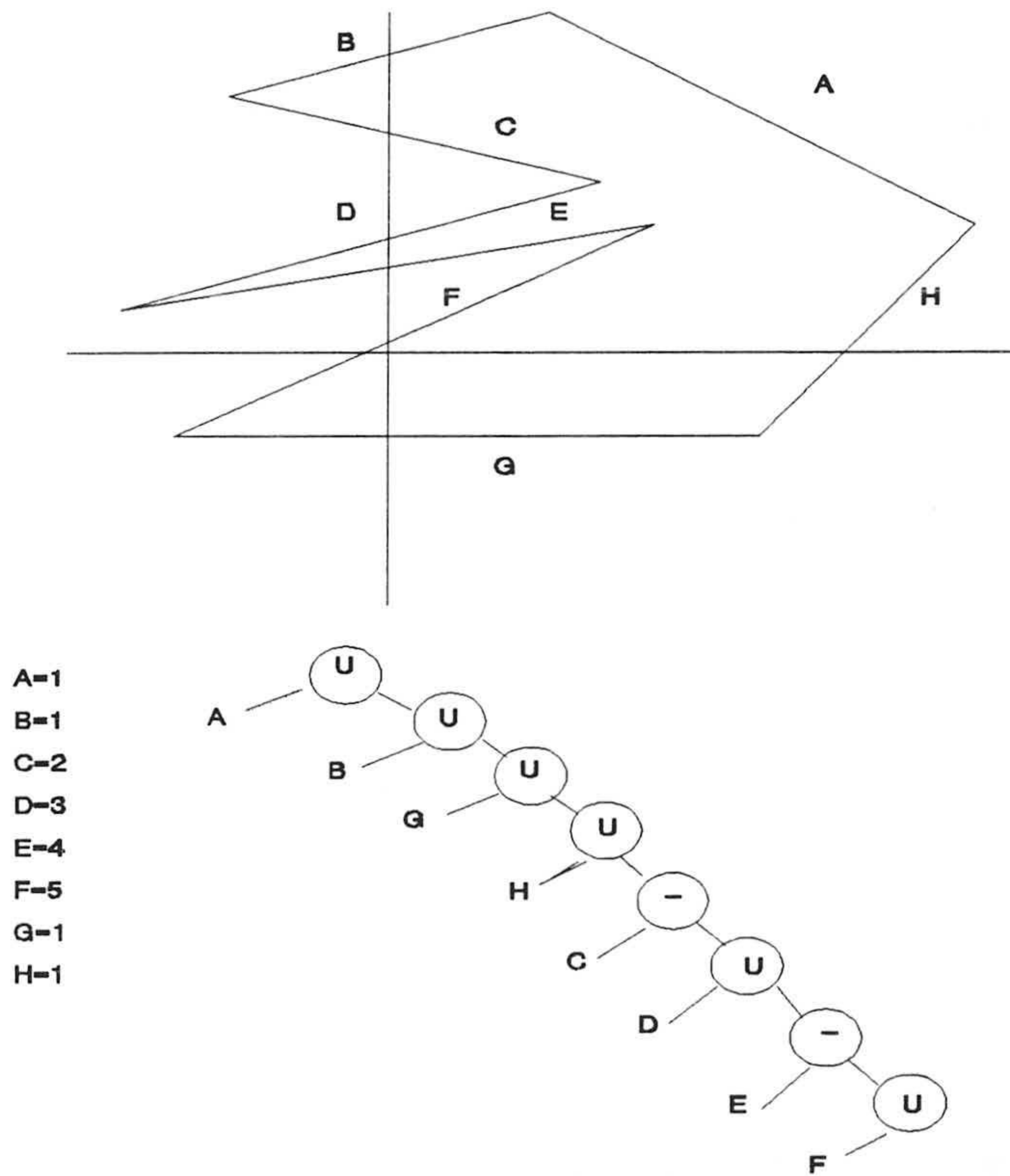


Figura 4.12 Obtención de un sólido por capas

Demostración:

Demostraremos la doble inclusión. Por ser las operaciones regularizadas, bastará hacer la demostración para los puntos del interior de cada uno de los conjuntos.

Sea Q un punto del interior de P . Si trazamos una semirrecta que parta del origen, pase por Q y se dirija al infinito, existirá un primer triángulo T_j al cual cortará después de pasar por Q . Por ser Q interior a P , T_j será positivo y de índice impar k . Por tanto el punto Q estará en el triángulo positivo T_j y no estará en ninguno de sus subordinados, por lo que estará en la capa C_k , con k impar, y no estará en ninguna capa de índice par superior por lo que estará en el conjunto determinado por el segundo miembro.

Si Q está en la expresión quiere decir que existe un valor impar k tal que Q está en la capa k , C_k , y Q no está en ninguna capa de índice superior. Por tanto existirá un triángulo positivo T_i de índice k tal que Q está en él y Q no pertenece a ningún triángulo negativo de índice superior. Por lo tanto Q está en P . ■

Para generar la fórmula, y por tanto un árbol CSG, basta aplicar el teorema.

En la figura 12 se observa una descomposición obtenida por la aplicación del teorema.

El proceso de cálculo de subordinados es sencillo, como veremos posteriormente, por lo que también lo es el cálculo del árbol CSG asociado.

Es evidente, por el método seguido para obtener la fórmula, que cada triángulo positivo aparece una sola vez. Ya que el número de triángulos positivos es finito, el proceso es finito y se obtiene una fórmula en tiempo finito.

4.3.1.2 Descomposición canónica en tres dimensiones

Ya se ha indicado antes que el método seguido en dos dimensiones es válido para tres dimensiones. Veamos los pasos.

Para simplificar al máximo la fórmula obtenida en el teorema 4.2 y obtener más fácilmente la representación del poliedro por medio de un árbol binario basado en pirámides originales podemos introducir el mismo concepto de nivel de profundidad definido entre triángulos. Dicho nivel de profundidad se formaliza a partir de las definiciones de pirámides subordinadas y dominantes dadas anteriormente.

Veamos en primer lugar una caracterización de la subordinación de pirámides originales:

Lema 4.6 Sean P y P' dos pirámides originales cuyas caras no originales se cortan a los sumo en un vértice o arista. Si $P \triangleleft P'$, toda semirrecta, distinta de las que están contenidas en las caras originales de las pirámides, que parte del origen y corta a las caras no originales de las dos pirámides, corta primero a la de P y luego a la de P' .

Demostración:

Sea r una semirrecta cualquiera y Q y Q' los puntos de corte con P y P' . Si $P \triangleleft P'$, el punto Q' será necesariamente interior a P' y por tanto Q estará más cerca del origen que Q' . ■

Al igual que en dos dimensiones, la relación de subordinación definida anteriormente se puede considerar reflexiva y antisimétrica, aunque no transitiva.

Vamos a generalizar la definición de subordinación, para intentar obtener una relación de orden entre las pirámides subordinadas:

Definición 4.11 Dado un conjunto $P = \{P_i \mid i=1, \dots, n\}$ de pirámides originales, tales que caras no originales se cortan como máximo en una arista ó vértice, diremos que P_j es menor ó igual que P_k ($P_j \leq P_k$) si existe una cadena de pirámides de P , $P' = \{P_{j,0}, P_{j,1}, \dots, P_{j,m}\}$, con $P_{j,0} = P_j$, $P_{j,m} = P_k$, y tal que $P_{j,l} \triangleleft P_{j,l+1}$.

Esta relación cumple las propiedades reflexiva y transitiva como vamos a demostrar.

Lema 4.7 La relación definida en 4.11, cumple las propiedades reflexiva y transitiva:

Demostración:

Comenzaremos viendo que es reflexiva: Sea P_j una pirámide. Ya que toda pirámide está subordinada a sí misma, la cadena $P' = \{P_j\}$ verifica la definición 4.11 por lo que $P_j \leq P_j$, y la relación es reflexiva.

Demostremos ahora la propiedad transitiva. Supongamos que $P_i \leq P_j$ y que $P_j \leq P_k$. Por la primera relación existe una cadena $P' = \{P_i, \dots, P_j\}$; por la segunda relación existe una cadena $P'' = \{P_j, \dots, P_k\}$. La cadena $P''' = \{P_i, \dots, P_j, \dots, P_k\}$, claramente verifica las condiciones exigidas en la definición, por lo que $P_i \leq P_k$. ■

Al igual que hemos hecho en dos dimensiones, es necesario introducir una nueva definición para conseguir que la relación anterior verifique la propiedad antisimétrica y sea una relación de orden.

Definición 4.12 Dado un conjunto $P = \{P_i\}$, $i=1, \dots, n$, de pirámides originales, tales que sus caras no originales se cortan como máximo en una arista ó vértice, diremos que P es no separado si existen al menos dos pirámides P_i y P_k tal que

$$P_i \leq P_k \text{ y } P_k \leq P_i$$

En caso contrario diremos que P es separado.

Vamos a demostrar que a partir de todo conjunto no separado se puede definir un conjunto separado.

Lema 4.8 Sea $P = \{P_i\}$, $i = 1, \dots, n$, un conjunto de pirámides originales no separado y P_i y P_k dos pirámides tales que $P_i \leq P_k$ y $P_k \leq P_i$. Entonces existe al menos una subdivisión de la cara no original de P_i ó de P_k tal que al partir la pirámides de ese modo se obtiene un nuevo conjunto de pirámides originales que es separado.

Demostración: Si $P_i \leq P_k$ y $P_k \leq P_i$ existen al menos dos cadenas de pirámides tales que algunas de las relaciones de subordinación no es total.

Si fuesen todas las subordinaciones totales, por ser estas transitivas, $P_i \triangleleft P_k$ y $P_k \triangleleft P_i$, lo que en virtud del lema 4.6 es imposible. Supongamos que $P'_j \triangleleft P_i$ y que ésta subordinación es completa. Por tanto existe una subdivisión de la cara no original de P_j (que coincide con la proyección de la cara no original de P_i) tal que P'_j no está subordinada a P'_i . Sustituyendo P_i por P'_i y P''_i ó bien se obtiene un conjunto separado ó se repetiría el proceso. Ya que el número de pirámides es finito, el proceso es finito y se obtendrá al final un conjunto de triángulos originales separado. ■

Una vez resueltos los casos especiales, podemos demostrar que la relación definida anteriormente es una relación de orden.

Teorema 4.6 La relación definida en 4.9 es una relación de orden para conjuntos separados.

Demostración:

Evidente a partir de los lemas anteriores. ■

El teorema anterior, por tanto, nos permite introducir una relación de orden entre pirámides originales. Ya que las pirámides originales asociadas a un sólido verifican que sus caras no originales se cortan como máximo en un segmento, se sigue que hemos introducido una relación de orden entre las pirámides asociadas a un sólido, siempre y cuando este sea separado. El orden no es total, ya que puede haber pirámides

que no estén subordinadas a ninguna, ni que sean dominantes de ninguna otra. Esas pirámides serán siempre positivas, por lo que no representan un problema a la hora de la descomposición. Por otro lado, la relación de orden nos permite representar cada poliedro como un retículo del cual se puede extraer muy fácilmente el árbol CSG.

Así mismo, el hecho de que la relación introducida sea una relación de orden nos permite dotar a cada sólido de una estructura de retículo ([Kurat66]).

Mantienen su validez las definiciones 4.9 y 4.10 de índice y capa, aunque aplicadas a pirámides.

Vamos a demostrar que según las definiciones anteriores, todas las pirámides de índice 1 ó impar son positivas y que las de índice par son negativas.

Lema 4.9 Sea P un sólido y $\{P_i \mid i=1,\dots,n\}$ el conjunto de sus pirámides originales asociadas. Entonces si el índice de P_i es impar, P_i es positiva y si P_i es par, es negativa.

Demostración:

Puede realizarse siguiendo el mismo proceso que en el lema 4.5. ■

El teorema que nos permite obtener una fórmula en la que cada pirámide original aparece una sola vez es el siguiente:

Teorema 4.7 Sea P un sólido de caras $\{C_1, C_2, \dots, C_n\}$ y $\{P_i \mid i=1,\dots,n\}$ el conjunto de sus pirámides originales asociadas que supondremos separados. Sea X_i la capa de índice i . Entonces el sólido P se puede expresar como:

$$P = (((X_1 \cdot X_2) \cup X_3) \cdot X_4) \dots \cup X_{2N-1} \cdot X_{2N}$$

$$\text{siendo } X_i = \cup \{P_j \mid \text{ind}(P_j) = i\}$$

Demostración:

Demostraremos la doble inclusión. Por ser las operaciones regularizadas, bastará hacer la demostración para los puntos del interior de cada uno de los conjuntos.

Sea Q un punto del interior de P . Si trazamos una semirrecta que parte del origen, pasa por Q y se dirige al infinito, existirá una primera pirámide P_j a la cual cortará después de pasar por Q . Por ser Q interior a P_j , esta será positiva y de índice impar k . Por tanto el punto Q estará en la pirámide positiva P_j y no estará en ninguna de sus subordinadas, por lo que estará en la capa C_k , con k impar, y no estará en ninguna capa de índice superior, por lo que estará en el conjunto determinado por el segundo miembro.

Si Q está en la expresión del segundo miembro, quiere decir que existe un valor par k tal que Q está en la capa k , X_k y Q no está en ninguna capa de índice superior. Por tanto existirá una pirámide original positiva P_i de índice k tal que Q está en ella y Q no pertenece a ninguna pirámide negativa de índice superior. Por lo tanto Q está en P , y se tiene el contenido. ■

Para generar la fórmula, y por tanto un árbol CSG, basta aplicar lo indicado en el teorema.

Es evidente, por el método seguido para obtener la fórmula, que cada pirámide positiva aparece una sola vez. Ya que el número de pirámides positivas es finito, el proceso es finito y se obtiene una fórmula en tiempo finito.

Si deseamos expresar la pirámide en función de sus tetraedros relacionados bastaría unir los resultados de los teoremas 4.5 y 4.7. Partimos de los tetraedros relacionados obtenidos a partir de cada arista de la pirámide original y de las que resultan de unirlos con un punto determinado (por ejemplo, el primer vértice referenciado de cada cara).

Según el teorema 4.5 para cada polígono (base de la pirámide en este caso) existirá un árbol CSG único en el que cada triángulo aparece una sola vez. Sustituyendo en la fórmula obtenida en el teorema 4.7 cada pirámide por su árbol CSG relacionado

se obtiene la expresión de cualquier poliedro en función de tetraedros originales (el número total de tetraedros coincide con el número total de aristas orientadas de Karasick). Se podría hacer un recálculo de las subordinaciones entre los tetraedros ya que el hecho de que una pirámide esté subordinada a otra no quiere decir que todos sus tetraedros asociados estén subordinados a los tetraedros de la otra. Esto equivale a partir inicialmente de todos los tetraedros y estudiar la relación de subordinación entre ellos.

Lo anterior nos indica que podemos trabajar a dos niveles: tomando como primitivas las pirámides ó tomando como primitivas los tetraedros. Las ventajas de esta última descomposición es que se usa conjuntos uniformes muy sencillos (tetraedros originales), aunque aumenta el número total de primitivas.

Todo el desarrollo realizado sirve también para polígonos y poliedros con agujeros aunque no se haya incidido en ello especialmente. Estos basta definirlos con una orientación contraria a la frontera exterior. Se estudiaría la subordinación tanto de los triángulos originales correspondientes a la frontera exterior como a la del agujero y se cotendría una expresión booleana que integra cada simplex en su orden, tal y como puede observarse en la figura 4.13

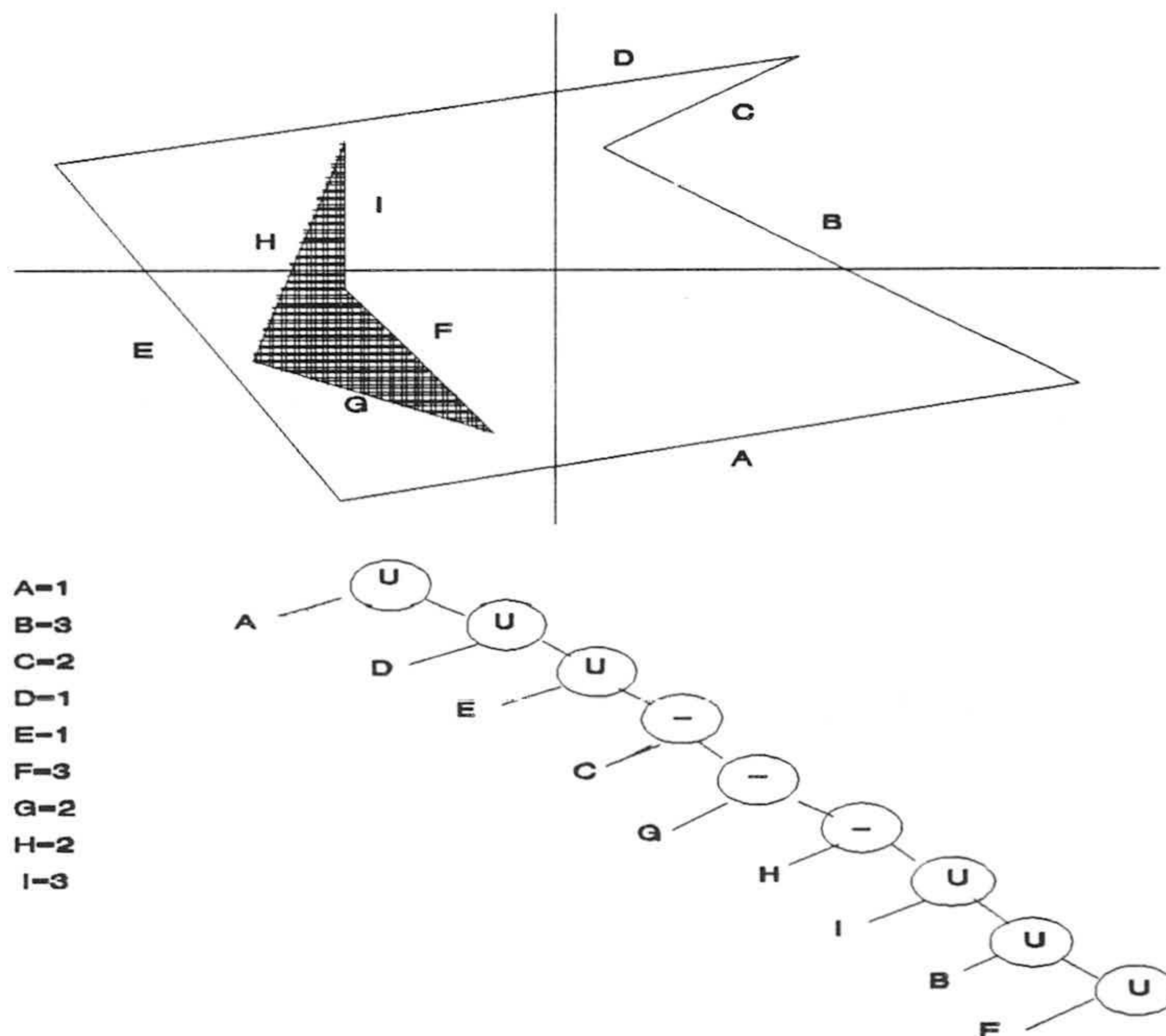


Figura 4.13 Descomposición de sólidos con agujeros

Vamos a continuación a estudiar las operaciones booleanas entre sólidos que están definidos a partir de un recubrimiento simplicial.

4.4 Operaciones entre sólidos

En este apartado vamos a estudiar el modo de realizar operaciones booleanas con sólidos representados por recubrimientos simpliciales.

Ya que \mathbb{R}^2 ni \mathbb{R}^3 son sólidos en el sentido de la definición dada en el apartado 2.1, no es posible obtener una representación de esos conjuntos en función de símlices originales. Para realizar las operaciones de complementación definiremos un conjunto universal. Dicho conjunto será la caja bidimensional (respectivamente tridimensional) determinada por M , donde M indica un punto de \mathbb{R}^2 (respectivamente de \mathbb{R}^3) con magnitud lo suficientemente grande para que la caja encierre a los objetos con los que se va a trabajar.

En el caso de tres dimensiones por ser una caja convexa su descomposición en símlices sería simplemente la unión de todas las pirámides originales P_i asociadas a las caras C_i . Todas estas pirámides tienen índice 1. Podemos también considerar los tetraedros originales obtenidos a partir de cada pirámide tomando una de las diagonales de cada cara.

En la mayoría de los casos, al realizar operaciones entre sólidos, el origen de coordenadas será común por lo que no será necesario el recálculo de las subordinaciones.

Sí sería necesario el recálculo de subordinaciones si consideramos la descomposición asociada al sólido y no al origen de coordenadas. Esto ocurre cuando se toma como punto de referencia de cada sólido su centro de gravedad, que es una elección interesante ya que es invariante para muchas de las transformaciones que se aplican a un sólido.

En el estudio que vamos a hacer a continuación de operaciones entre sólidos suponemos que el punto de referencia elegido es común para los diversos sólidos que intervienen en las operaciones.

Comenzaremos viendo la operación más sencilla que es la complementación

4.4.1 Complementación de sólidos

Ya se ha indicado anteriormente que todo el desarrollo realizado es válido para polígonos y poliedros con agujeros. Obtener el complemento de un sólido equivale a considerar el conjunto universal (la caja bidimensional o tridimensional definida anteriormente) y el sólido como un agujero. Para ello se cambia el signo de cada una de las primitivas originales del sólido. En relación a los índices, equivale a sumar una unidad al índice de cada una de las primitivas originales que determinan el sólido. La demostración es sencilla y se basa en el hecho de que hemos rodeado el sólido con una serie de primitivas de índice 1 (las que corresponden a la caja considerada) y además hemos cambiado el signo de todas las primitivas originales.

Las pirámides originales se pueden sustituir por los árboles asociados de tetraedros. Si partimos directamente del árbol de tetraedros originales, basta considerar estas como primitivas originales y sumar una unidad a cada uno de los índices.

Veamos un teorema que nos permite obtener el complemento de un sólido a partir de sus capas.

Teorema 4.8 Sea O Un sólido definido por capas como

$$O = (((X_1 - X_2) \cup *X_3) \dots \cup *X_{2N-1}) - X_{2N}$$

Entonces

$$O^c = (((W - X'_2) \cup *X'_3) \dots \cup *X'_{2N-1}) - X'_{2N} \cup *X'_{2N+1}), \text{ con } X'_i = X_{i-1}, \text{ y siendo } W \text{ el conjunto universal.}$$

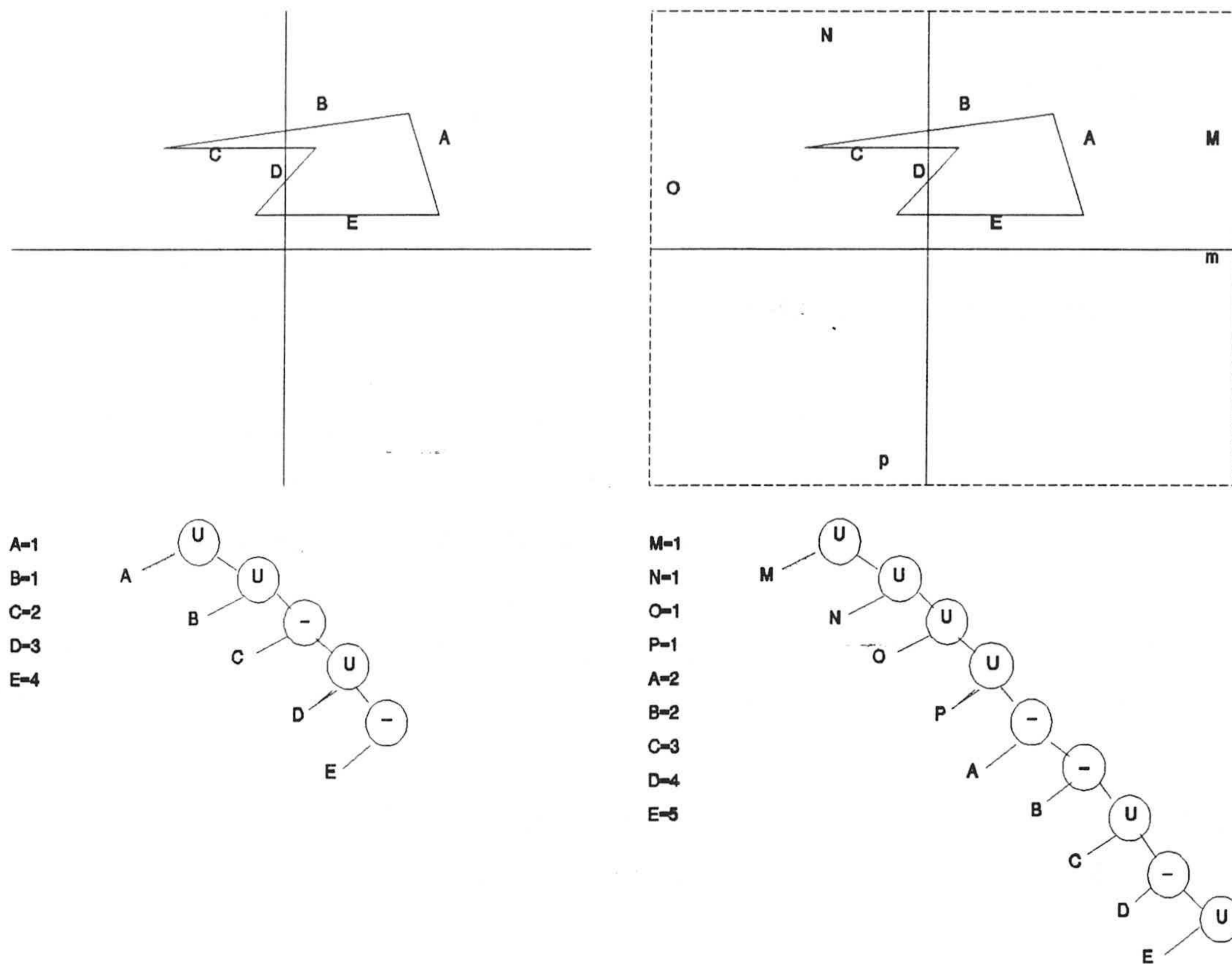


Figura 4.14 Complementación de un sólido

Demostración:

Ya que $O^c = W - O$ toda primitiva está subordinada a W (hemos considerado que W es el conjunto universal), por lo que por la propia definición de complementación se tiene la igualdad. ■

4.4.2 Diferencia, unión e intersección de entre sólidos

A la hora de considerar el resto de las operaciones (diferencia, unión e intersección entre sólidos) hay que tener en cuenta que para la mayoría de las aplicaciones no será necesario evaluar el árbol obtenido, sino que bastará expresar el

árbol como diferencia ó unión ó intersección de los árboles sobre los que se va a trabajar, es decir, de los árboles que representan cada uno de los sólidos.

Por otra parte las relaciones entre las operaciones booleanas nos permiten obtener algunas de forma indirecta. Concretamente se hará uso de que la diferencia de dos conjuntos $A-B$ es igual a la intersección de A con el complementario de B y de que la unión de dos conjuntos es igual al complementario de la intersección de sus complementarios.

Intersección:

Ya se ha comentado que las operaciones básicas se reducirán todas a la intersección. Va a ser esta operación la que veamos en más detalle, aunque no se hará un estudio pormenorizado de todos los casos posibles. Queda abierto para futuros desarrollos el estudio completo del algoritmo de intersección para su posterior implementación.

En el modelo en el que trabajamos partimos de la siguiente información: árbol no minimal obtenido a partir de la matriz de subordinación de cada sólido en la que se guardan las relaciones de subordinación entre los símlices originales de un sólido. A partir de dicha matriz se obtienen los índices de cada símlice y el árbol CSG minimal asociado.

Para obtener la intersección de dos sólidos podemos considerar simplemente el árbol no evaluado determinado por la intersección de los árboles que determinan cada sólido, que será suficiente para determinados problemas; por ejemplo para la visualización básica realizada por técnicas de trazado de rayos que se lanzan contra el árbol, evaluando cada una de las posibles intersecciones ([Brons90]).

El problema surge cuando es necesario evaluar el árbol, es decir, realizar la evaluación de fronteras. Podríamos aplicar las técnicas usuales, aunque es necesario aprovechar la información que ya se tiene y la simplicidad de los elementos que intervienen en las operaciones, ya que sino se estarían aportando algoritmos que

complican el cálculo de la intersección más que simplificarlo. Sabemos que la frontera del sólido intersección estará formada por elementos que proceden de la frontera de alguno de los sólidos, por lo que en muchos casos sólo será necesario realizar intersecciones entre caras no originales de cada uno de los simples originales. Por otro lado y debido a la información ya existente se podría intentar reducir el número de intersecciones a realizar.

Parece natural basar el cálculo de intersecciones en los dos aspectos indicados: usar la información sobre la frontera ya existente (es decir, utilizar sólo caras no originales para realizar la intersección) así como intentar reducir el cálculo del número de intersecciones utilizando los datos de que disponemos. El resultado del algoritmo debería ser la frontera del sólido intersección y el árbol minimal asociado obtenido a partir de los índices de cada simplex original.

El algoritmo indicado no aporta especiales resultados a las operaciones entre sólidos, aunque dentro de la teoría que estamos desarrollando facilita la obtención de árboles asociados al sólido resultante.

En todo el resto del apartado y mientras no se indique lo contrario presupondremos que las operaciones que se indiquen son regularizadas.

En 2 dimensiones, un esquema del método a seguir sería el siguiente:

- intersectar cada arista no original de los triángulos originales asociados a A con cada arista de B, obteniendo una descomposición en triángulos originales cuyas aristas no originales se intersectan a lo sumo en un vértice o son coincidentes.
- en caso de que haya habido intersección, se estudia la subordinación entre los nuevos triángulos originales de A y de B, y se aplican los siguientes criterios:

- . Si los dos son positivos, se escogen sólo los subordinados, despreciando los otros.
- . Si los dos son negativos se eligen los dominantes.
- . Si uno es positivo y otro negativo se elige el negativo subordinado a positivo y el positivo dominante del negativo, si existiesen.
- si no hay intersección, se comprueba si la arista no original de los triángulos de A está contenida en B o si la de B está en A.
- si los triángulos originales coinciden incluso en el signo, se elige; si tienen signos distintos se desprecian los dos.
- el estudio de la subordinación se hace por índices, de modo que simultáneamente se puede hacer la asignación de los nuevos índices: primero los de índice 1 de A con todos los de B; los de índice 2 de A con todos los de B; y así sucesivamente.

Para evitar tener que estudiar todas las intersecciones, puede estudiarse previamente la intersección de triángulos originales positivos: si dos triángulos positivos no se cortan, tampoco lo hacen su negativos subordinados en cuanto subordinados a ellos: es decir, no tendría que estudiarse la intersección con los subordinados que lo son exclusivamente de ellos (lo que hemos denominado totalmente subordinados). Este estudio se comenzaría con los de índice 1. Para ello es necesario determinar si dos triángulo originales se cortan ó no. Esto nos lo indica el siguiente lema:

Lema 4.10 Sean T_1 y T_2 dos triángulos originales. La condición necesaria y suficiente para que se corten es que alguno de los vértices no originales de T_1 esté contenido en la zona angular determinada por T_2 o viceversa.

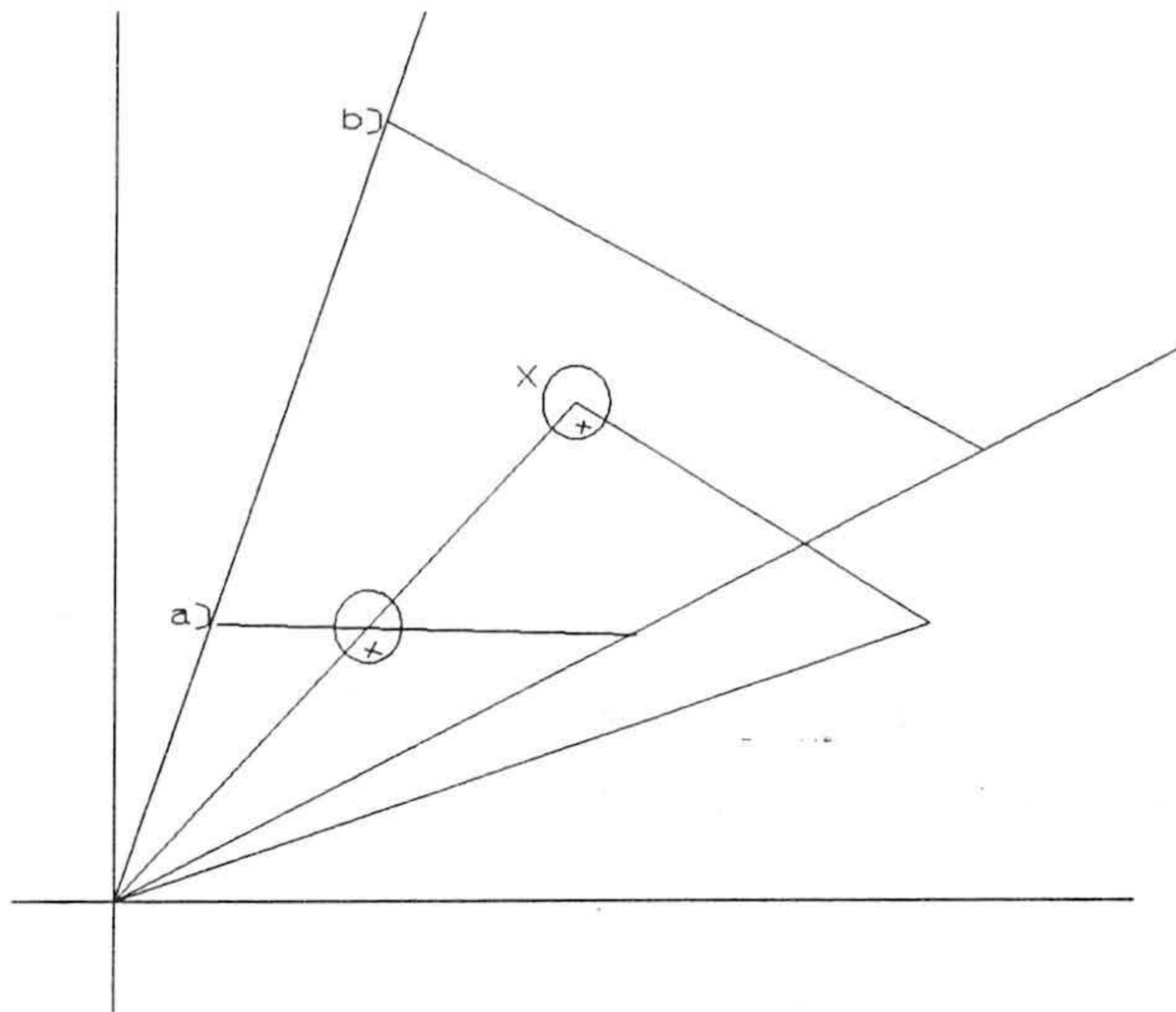


Figura 15. Lema 4.10

Demostración:

La condición es equivalente a que la semirrecta que parte del origen y pasa por el vértice existente (X en la figura 13) corta al segmento no original del otro triángulo (bien entre X y el origen ó entre X e infinito), y esto equivale a que los triángulos tienen algún punto en común. ■

La determinación de si un punto está o no en una zona angular, de vértice el origen es muy sencilla de calcular en función del signo: basta

que desde dicho punto se vean cada uno de los puntos y el origen con el mismo signo: es decir $\text{sig}(Q, P_1, O) = \text{sig}(Q, O, P_2)$.

Tal y como hemos indicado anteriormente, mediante este cálculo previo se pueden evitar el cálculo de determinadas intersecciones por lo que es de esperar que la complejidad en el caso medio se reduciría de n^2 (donde n representa el número de primitivas que se están considerando).

Además de lo anterior hay que tener en cuenta que la subordinación se va a estudiar en muchos casos entre triángulos originales que tiene un vértice en común (el que coincide con el punto de intersección), por lo que el estudio de la subordinación se reduce en estos casos a estudiar una posible intersección.

Veamos dos ejemplos que nos ayuden a entender mejor los pasos a dar.

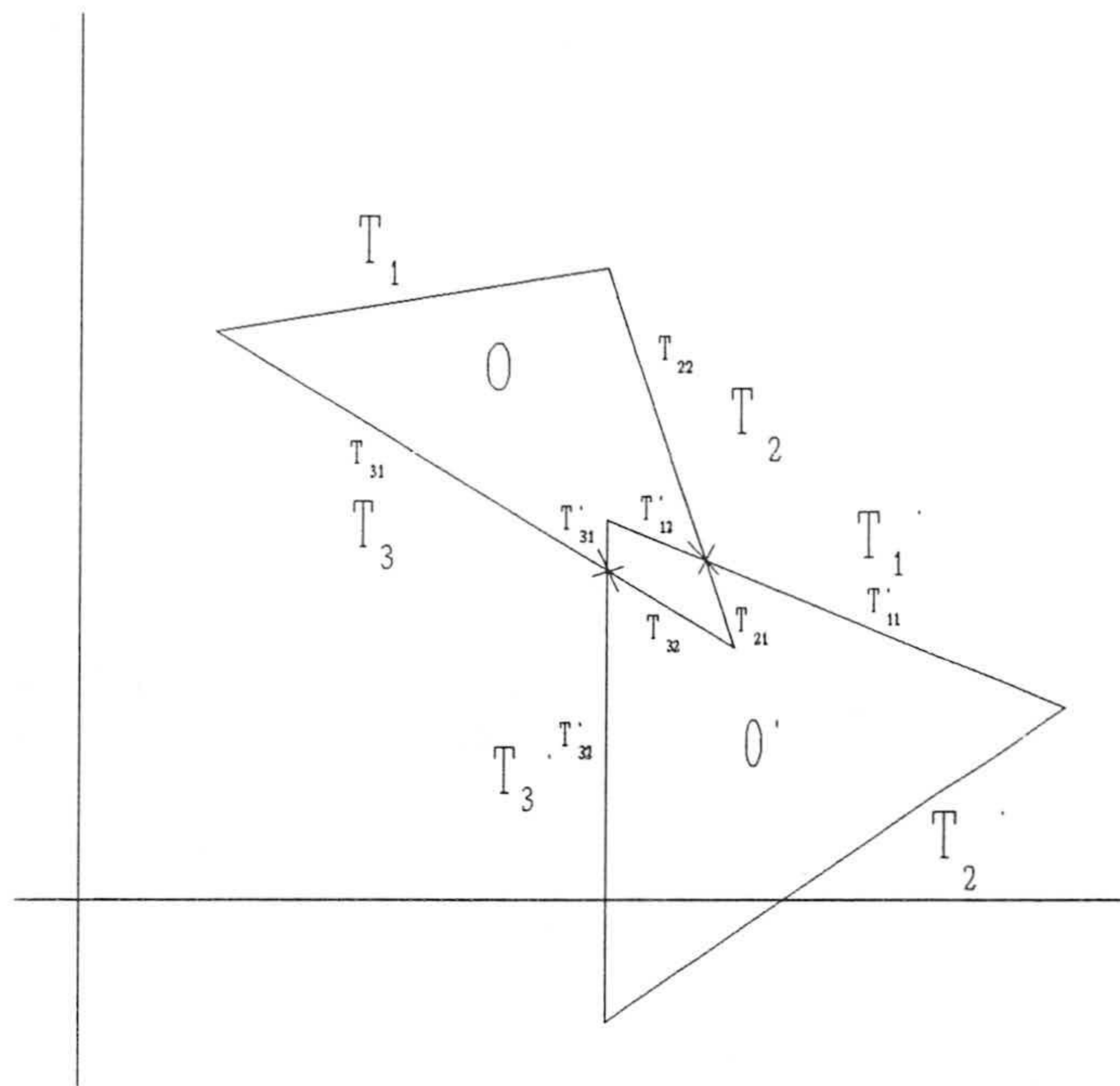


Figura 4.16 Intersección de sólidos

Los triángulos de la figura 4.16 pueden representarse como:

$$O_1 = T_1 \cup^* T_2 -^* T_3$$

$$O' = T'_{11} \cup^* T'_{12} -^* T'_{21} -^* T'_{31} -^* T'_{32}$$

Al estudiar la intersección de aristas se obtiene:

$$O = T_1 \cup^* T_{12} \cup^* T_{22} -^* T_{31} -^* T_{32}$$

$$O' = T'_{11} \cup^* T'_{12} \cup^* T'_{22} -^* T'_{21} -^* T'_{31} -^* T'_{32}$$

Realizando las operaciones para cada índice, tenemos

Índice 1: T_1 no intersecciona ni está contenido en O' . Se elimina

T_{21} domina a T'_{22} : se elimina y se elige T'_{22}

T_{22} está subordinado a T'_{21} : se elige y se elimina T'_{21}

Índice 2: T_{31} está subordinado a T'_{31} : se elimina y se elige T'_{31} .

T_{32} domina a T'_{32} : se elige y T'_{31} se elimina.

El resultado final es :

$$O \cap^* O' = T'_{22} \cup^* T_{22} -^* T'_{31} -^* T_{32}$$

Al no existir triángulos negativos exclusivamente subordinadas a uno positivo no es necesario realizar el estudio de intersecciones entre triángulos originales.

Es claro que si existen más índices se complica el cálculo.

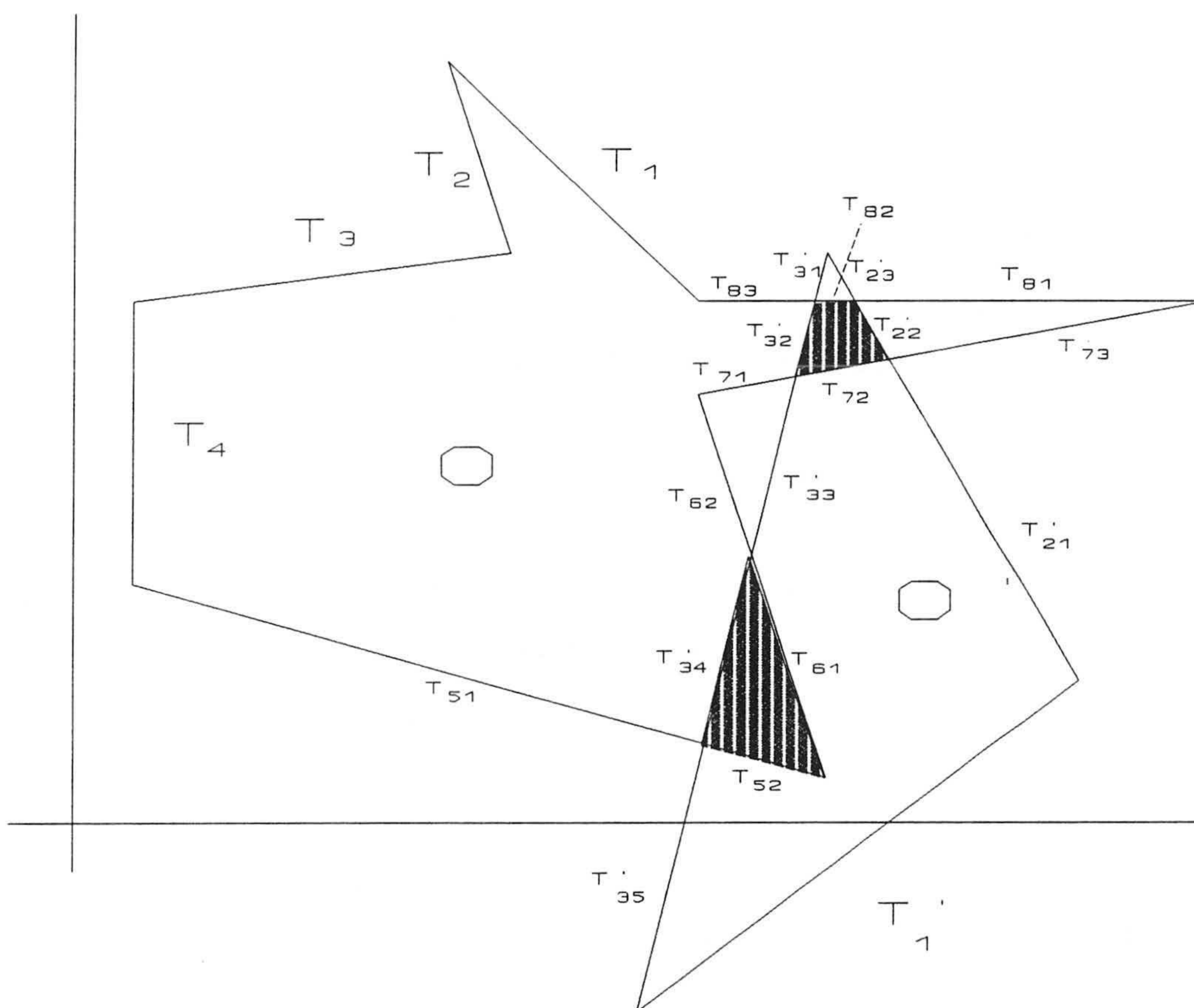


Figura 17 Intersección de polígonos

Veamos el ejemplo de la figura 17.

Los polígonos pueden expresarse como :

$$O = (T_1^- * T_2^- * T_5) \cup^* (T_3^- * T_4^- * T_5) \cup^* (T_6^- * T_5) \cup^* (T_8^- * T_7^- * T_5)$$

$$= (((T_1 \cup^* T_8)^- * T_2^- * T_7) \cup^* T_3 \cup^* T_6)^- * T_4^- * T_5$$

$$O' = T'_1 \cup^* T'_2^- * T'_3$$

Ya que en O hay triángulos negativos que están subordinados a un sólo positivo, es necesario estudiar la relación entre triángulos negativos:

- T_1 y T'_1 no se cortan, por lo que no es necesario hacer el estudio con T_2
- T_1 y T'_2 no se cortan, por lo que no es necesario hacer el estudio con T_2
- T_8 se descompone en T_{81} , T_{82} y T_{83}
- T_7 se descompone en T_{71} , T_{72} y T_{73}
- T_3 y T'_1 no se cortan, por lo que no es necesario hacer el estudio con T_4
- T_3 y T'_2 no se cortan, por lo que no es necesario hacer el estudio con T_4
- T_6 se descomponen en T_{61} y T_{62}
- T_5 se descompone en T_{51} y T_{52}
- T'_1 no está dentro de O y no se ha cortado con ningún triángulo original de O por lo que se elimina
- T'_2 se descompone en T'_{21} , T'_{22} y T'_{23}
- T'_3 se descompone en T'_{31} , T'_{32} , T'_{33} , T'_{34} y T'_{35}

La descomposición de los triángulos T'_1 , T'_2 y T'_3 viene determinada por las intersecciones de estos triángulos con los de O que se han ido obteniendo. En cada punto de intersección de las aristas no originales se va determinando que triángulos son los que se cortan.

Al objeto de determinar los triángulos definitivos se realizan las siguientes comprobaciones:

T_{81} domina a T'_{22} : se elimina T_{81} y se deja T'_{22} con índice 1

T_{82} está subordinado T'_{23} : se deja T_{82} y se elimina T'_{23}

T_{82} domina a T'_{32} : se deja T'_{32} con índice 1

T_{83} está subordinado a T'_{31} : se eliminan los dos

T_{71} está subordinado a T'_{32} : se elimina T_{71} .

T_{72} está subordinado a T'_{22} : se deja T_{72} con índice 2

T_{72} domina a T'_{33} se elimina este último.

T_{73} domina a T'_{21} : se eliminan los dos.

T_{61} domina a T'_{34} se dejan los dos con índices 1 y 2 respectivamente

T_{51} está subordinado a T'_{34} : se elimina T_{51}

T_{52} domina a T'_{35} : se deja con índice 4 y T'_{35} se elimina.

El resultado final sería:

$$O \cap O' = T'_{21} \cup T_{82} \cup T_{61} \cup T'_{32} \cup T_{72} \cup T'_{34} \cup T_{52}$$

La justificación de los ejemplos anteriores se basa en los algoritmos usuales para determinar la intersección de dos sólidos, a los que se les añaden las técnicas indicadas a lo largo de este trabajo para el cálculo de subordinaciones.

Aunque el fundamento de la operación de intersección es el mismo en 2 y en 3 dimensiones (descomponer en símlices originales cuyas caras no originales se intersecten a lo sumo en un punto ó arista), es claro que la complejidad y el número de casos especiales aumenta al pasar de 2 a 3 dimensiones.

Suponemos, al igual que en el caso de 2 dimensiones, que disponemos de la información de las relaciones de subordinación entre las pirámides originales asociadas a cada sólido, así como del árbol CSG minimal que define el sólido. Así mismo están disponibles para cada sólido la caja envolvente minimal y la pirámide envolvente minimal de cada una de las pirámides originales, lo que permite en algunos casos poder predeterminar si las pirámides o sólidos se intersectan o no.

El algoritmo requiere los siguientes pasos, supuestos los sólidos tridimensionales A y B:

- Intersectar cada pirámide original de A con cada pirámide original de B, de modo que se obtiene para cada sólido una descomposición en pirámides originales cuyas caras no originales se intersectan a lo sumo en puntos o aristas comunes, o coinciden.
- Una vez realizada la descomposición anterior se van estudiando las relaciones de subordinación (gran parte de la información necesaria se obtiene en el proceso de intersección) y se aplican los mismo criterios que para el caso bidimensional:
 - . Si las dos son positivas, se escogen sólo las subordinadas, despreciando las otras.
 - . Si las dos son negativas se eligen las dominantes.

- . Si una es positiva y otra negativa se elige la negativa subordinada a positiva y la positiva dominante de la negativa, si existiesen.

- si no hay intersección, se comprueba si la base no original de las pirámides de A está contenida en B o si la de las pirámides de B está contenida en A.

- si las pirámides originales coinciden incluso en el signo, se eligen ambas; si tienen signos distintos se desprecian las dos.

- el estudio de la subordinación se hace por índices, de modo que simultáneamente se puede hacer la asignación de los nuevos índices: primero las de índice 1 de A con todas las de B; las de índice 2 de A con todas las de B; y así sucesivamente.

Al igual que en el caso de dos dimensiones, para evitar tener que estudiar todas las intersecciones, puede estudiarse previamente la intersección de pirámides originales positivas: si dos pirámides positivas no se cortan, tampoco lo hacen su negativas subordinadas en cuanto subordinadas a ellas.

También podemos utilizar la propiedad de que si todos los vértices no originales de una de las pirámides tiene el mismo signo respecto del plano que determina la base de la otra pirámide no hay intersección entre sus bases que es un resultado conocido de la geometría analítica. La inversa no tiene porque ser cierta.

El siguiente lema facilita la realización de esta comprobación.

Lema 4.11 Sean OC_1 y OC_2 dos pirámides originales. Una condición suficiente para que se corten es que alguno de los vértices de C_1 esté contenido en la zona angular determinada por OC_2 o viceversa.

Demostración:

La condición es equivalente a que la semirrecta que parte del origen y pase por el vértice de una pirámide contenido en la zona angular de la otra corta a la base de la otra pirámide y esto quiere decir que las pirámides tienen intersección no vacía. ■

Para determinar si un punto está en la zona angular determinada por una pirámide basta comprobar que signo final tiene respecto al total de zonas angulares determinadas por el recubrimiento de tetraedros que definen la pirámide.

Al igual que en el caso bidimensional, es muy sencillo determinar si un punto Q está en la zona angular determinada por el tetraedro $OP_1P_2P_3$. Basta que se cumpla:

$$\text{sign}(Q,P_1,P_2,O) = \text{sign}(Q,P_2,P_3,O) = \text{sign}(Q,P_3,P_1,O)$$

Ya hemos indicado anteriormente que no era el objetivo de esta tesis un estudio detallado de las operaciones entre sólidos, sino mostrar como a partir de la información contenida en la matriz de subordinación y en el árbol CSG minimal se pueden facilitar las operaciones.

En el esquema anteriormente hecho, la mayor complicación es la división de cada una de las pirámides originales, de modo que cada cara no original esté totalmente fuera del otro sólido ó totalmente dentro, ó que sea coincidente con otra cara. Un estudio detallado de algoritmos que obtienen esta división puede encontrarse en la tesis doctoral de Vanecek ([Vanec89]).

En el esquema del algoritmo indicado anteriormente no hemos utilizado los resultados del capítulo II: lo que hemos hecho es simplemente utilizar técnicas ya estudiadas por otros autores para la evaluación de fronteras que permiten realizar operaciones booleanas y hemos añadido un estudio de la subordinación entre simplices originales. Podría ser utilizado en algunos desarrollos prácticos pero parece más conveniente intentar obtener algún algoritmo de intersección basado en la fundamentación teórica que proporcionan los sistemas de generadores formados por objetos originales.

La diferencia entre las operaciones definidas en el espacio vectorial de los objetos gráficos y las operaciones que constituyen a dicho conjunto en un álgebra de Boole es que la suma y resta son acumulativas en cuanto al valor de la presencia de un punto mientras que la unión y la diferencia conjuntista no lo son. Ya hemos visto en el corolario 1 del capítulo 2 que si dos conjuntos tienen intersección vacía, entonces la unión coincide con la suma.

En el caso de la diferencia conjuntista de dos objetos $O_1 - O_2$ con presencia 1, basta no considerar los puntos de O_2 que están en el complementario de O_1 para obtener similares resultados a la resta vectorial.

Por otro lado, la unión conjuntista con respecto a la resta conjuntista no son conmutativas ni asociativas, propiedades que sí que cumpliría la suma de objetos gráficos si partimos de una ζ -estructura.

Realizaremos la intersección de los objetos capa a capa.

Podemos suponer, sin pérdida de generalidad, que la última capa tiene signo negativo (lo que equivale a que el origen no está en el interior del objeto). De no ser así, la última capa sería el conjunto vacío.

Como ya se ha dicho antes, hay que tener en cuenta que las operaciones unión y diferencia no son asociativas entre ellas y que las operaciones se ejecutan de izquierda a derecha.

El siguiente teorema nos permite expresar la intersección de dos objetos a partir de las capas que los componen:

Teorema 4.9: Sean O_1 y O_2 dos objetos gráficos representados por

$$\begin{aligned} O_1 &= C_1(O_1) \cdot C_2(O_1) \cup C_3(O_1) \cdot \dots \cup C_{2^{*n-1}}(O_1) \cdot C_{2^{*n}}(O_1) \\ O_2 &= C_1(O_2) \cdot C_2(O_2) \cup C_3(O_2) \cdot \dots \cup C_{2^{*m-1}}(O_2) \cdot C_{2^{*m}}(O_2) \end{aligned}$$

Entonces:

$$O_1 \cap^* O_2 = \bigcup_{i=1}^n \bigcup_{j=1}^m ((C_{2*i-1}(O_1) \cap^* C_{2*j-1}(O_2)) -^* (C_{2*i}(O_1) \cup^* C_{2*j}(O_2)))$$

Demostración:

La demostración del teorema la realizaremos en dos pasos: en primer lugar supondremos los objetos formados por su capas asociadas. Posteriormente realizaremos la operación entre cada una de las capas.

Teniendo en cuenta la propiedad distributiva de la unión respecto de la intersección y considerando cada capa positiva y su siguiente negativa como un sólo elemento sería:

$$\Gamma_{2*i-1}(O) = C_{2*i-1}(O) -^* C_{2*i}(O)$$

$$O_1 \cap^* O_2 = (\Gamma_1(O_1) \cup^* \Gamma_3(O_1) \cup^* \dots \cup^* \Gamma_{2*n-1}(O_1)) \cap^* (\Gamma_1(O_2) \cup^* \Gamma_3(O_2) \cup^* \dots \cup^* \Gamma_{2*m-1}(O_2)) =$$

$$\bigcup_{i=1}^n \bigcup_{j=1}^m (\Gamma_{2*i-1}(O_1) \cap^* \Gamma_{2*j-1}(O_2))$$

Pero:

$$\begin{aligned} \Gamma_p(O_1) \cap^* \Gamma_q(O_2) &= (C_p(O_1) -^* C_{p+1}(O_1)) \cap^* (C_q(O_2) -^* C_{q+1}(O_2)) = \\ &= (C_p(O_1) \cap^* C_{p+1}(O_1)^c) \cap^* (C_q(O_2) \cap^* C_{q+1}(O_2)^c) = \\ &= (C_p(O_1) \cap^* C_q(O_2)) \cap^* (C_{p+1}(O_1)^c \cap^* C_{q+1}(O_2)^c) = \\ &= (C_p(O_1) \cap^* C_q(O_2)) \cap^* (C_{p+1}(O_1) \cup^* C_{q+1}(O_2))^c = \\ &= (C_p(O_1) \cap^* C_q(O_2)) -^* (C_{p+1}(O_1) \cup^* C_{q+1}(O_2)) \end{aligned}$$

Y sustituyendo esta última igualdad en el sumatorio anterior se obtiene el teorema. ▀

Hay una cierta similitud entre este teorema y los demostrados en el capítulo II, aunque debido a las diferencias indicadas entre suma y resta por una lado, y unión y diferencia conjuntista por otro, no se obtienen formulas idénticas.

Es claro que la fórmula obtenida no tiene porque ser minimal, y se necesitarían simplificaciones posteriores. Tal y como indica Paoluzzi en [PauRS93], para unas operaciones booleanas en cierto modo similares a las aquí presentadas, la simplificación es la fase más compleja de las operaciones booleanas.

Cuando deseamos que en una descomposición en la que interviene una diferencia de conjuntos aparezcan los menos elementos posibles, basta calcular previamente la intersección que tienen dichos conjuntos. Es decir, por las propiedades de las operaciones con conjuntos sabemos que $A-B = A-(A \cap B)$. De este modo se evita que aparezcan los elementos de B que no están en A, que no contribuyen al cálculo de la diferencia; además, al expresarlo así, como se ha indicado anteriormente, la diferencia conjuntista coincide con la resta de los objetos.

Basándonos en la propiedad anterior, podemos enunciar una lema que nos facilita una mejor simplificación:

Lema 4.12 Sean O_1 y O_2 dos objetos y $\Gamma_p(O_1)$ y $\Gamma_q(O_2)$ los elementos definidos por:

$$\Gamma_p(O) = C_p(O) - *C_{p+1}(O)$$

$$\Gamma_q(O) = C_q(O) - *C_{q+1}(O)$$

Entonces :

$$\Gamma_p(O_1) \cap * \Gamma_q(O_2) = (C_p(O_1) \cap *C_q(O_2)) - ((C_{p+1}(O_1) \cap *C_q(O_2)) \cup *(C_p(O_1) \cap *C_{q+1}(O_2)))$$

Demostración:

$$\begin{aligned}
 \Gamma_p(O_1) \cap^* \Gamma_q(O_2) &= (C_p(O_1) -^* C_{p+1}(O_1)) \cap^* (C_q(O_2) -^* C_{q+1}(O_2)) \\
 &= (C_p(O_1) -^* (C_{p+1}(O_1) \cap^* C_p(O_1))) \cap^* (C_q(O_2) -^* (C_{q+1}(O_2) \cap^* C_q(O_2))) \\
 &= C_p(O_1) \cap^* (C_{p+1}(O_1) \cap^* C_p(O_1))^c \cap^* C_q(O_2) \cap^* (C_{q+1}(O_2) \cap^* C_q(O_2))^c \\
 &= C_p(O_1) \cap^* C_q(O_2) \cap^* (C_{p+1}(O_1) \cap^* C_p(O_1))^c \cap^* (C_{q+1}(O_2) \cap^* C_q(O_2))^c \\
 &= (C_p(O_1) \cap^* C_q(O_2)) \cap^* ((C_{p+1}(O_1) \cap^* C_p(O_1)) \cup^* (C_{q+1}(O_2) \cap^* C_q(O_2)))^c \\
 &= (C_p(O_1) \cap^* C_q(O_2)) -^* ((C_{p+1}(O_1) \cap^* C_p(O_1)) \cup^* (C_{q+1}(O_2) \cap^* C_q(O_2))). \blacksquare
 \end{aligned}$$

Observando la última fórmula del lema 4.12 queda más patente su similitud con los teoremas del capítulo II. Tal y como se ha indicado anteriormente no es necesario "sumar" la intersección común de los negativos ya que la unión no es aditiva en cuanto a la presencia de puntos.

Para obtener el resultado final de la intersección bastará sustituir cada una de las capas por los respectivos símlices, y aplicar las oportunas propiedades de las operaciones conjuntistas. Bastaría aplicar en la fórmula del teorema anterior el siguiente lema:

Lema 4.13 Sean O_1 y O_2 dos objetos y $C_i(O_1)$ y $C_j(O_2)$ dos capas arbitrarias. Si

$$C_i(O_1) = T_{i,1}(O_1) \cup^* T_{i,2}(O_1) \cup^* \dots \cup^* T_{i,l_i}(O_1)$$

$$\text{y } C_j(O_2) = T_{j,1}(O_2) \cup^* T_{j,2}(O_2) \cup^* \dots \cup^* T_{j,k_j}(O_2)$$

son la descomposición de cada capa en símlices originales, entonces:

$$C_i(O_1) \cap^* C_j(O_2) = \bigcup_{p=1}^{l_i} \bigcup_{q=1}^{k_j} (T_{i,p}(O_1) \cap^* T_{j,q}(O_2))$$

Demostración:

Es evidente, aplicando la distributividad de las operaciones unión e intersección. ■

Podría argumentarse que al realizar las operaciones anteriores se están realizando intersecciones que no tendría porque ser necesarias. Esto no es un inconveniente ya que los elementos que intervienen son sencillos (símplices originales de dimensiones 2 y 3), y en muchos casos al objeto de facilitar la simplificación no será necesario realizar la intersección, sino dejarla indicada para una simplificación posterior. Esto lo haremos, en concreto, cuando las caras no originales no se intersecten.

Al realizar las intersecciones de símplices conviene tener en cuenta que una capa está formada por conjuntos de símplices que tienen intersección vacía ó una cara lateral común.

Vamos a continuación a presentar una serie de algoritmos que nos permiten calcular fácilmente si dos símplices son o no subordinados. Debido a que esta definición es clave en todo el desarrollo que se ha realizado, es conveniente que se den métodos adecuados para aplicarla en la práctica. El estudio en dos y tres dimensiones se hará por separado.

4.5 Subordinación en dos dimensiones

Ya se indicó que la definición dada en 4.1, al principio del capítulo, se iba a aplicar a triángulos originales cuyas aristas no originales o coinciden o a lo sumo tienen un vértice común, ya que forman parte de la frontera de un sólido bidimensional.

De lo anterior se deduce que sólo son posibles los siguientes casos para que dados dos símplices T y T' , el segundo esté subordinado al primero:

- T' está contenido en T
- Un sólo vértice de T' es interior a T.
- Los dos vértices de T' son exteriores a T y la arista no original de T' corta a las aristas originales de T (esto equivale a que los vértices no originales de T son exteriores a T' pero contenidos en la zona angular determinada por el origen y las semirrectas que contienen a las aristas originales de T').

En la práctica sólo es necesario estudiar la relación de subordinación entre símlices positivos y negativos ya que dos símlices del mismo signo o no se cortan o tienen entre medias un simplex de signo contrario.

Supondremos un valor ε positivo que nos permitirá sustituir igualdades del tipo $x = 0$.

La primera función que presentamos nos permite determinar la orientación de tres puntos. Si la orientación es contraria al sentido de las agujas del reloj, la orientación es positiva y su signo es +1; si la orientación es según las agujas del reloj, la orientación es negativa y el signo -1; si el triángulo que forman es degenerado, el signo es cero. Se supone que cada punto viene determinado por dos coordenadas.

```

signo2d( point a, point b, point c)
{
sig=a.x*b.y + b.x*c.y + a.y*c.x - b.y*c.x - c.y*a.x - a.y*b.x;
if (sig > ε) return(1);
if (sig < -ε) return(-1);
return(0);
}

```

La misma función anterior nos permite determinar la orientación de un triángulo original cualquiera (y por tanto la orientación de una arista cualquiera de un sólido bidimensional). Basta para ello incluir como primer punto el origen de coordenadas, y como los otros dos, los dos vértices que determinan la arista no original.

```

signos[Ti]=signo2d(ORIGEN, vértice[i], vértice[i+1]);

```

Si suponemos que una arista está orientada positivamente, para determinar si un punto está incluido ó no en un triángulo original se aplicaría la siguiente función:

```
punto_en_T(point a, point b, point c)
{
  if (signo2d( a, ORIGEN, b) >=0 && signo2d( a,b,c) >=0
      && signo2d( a,c,ORIGEN) >=0) return(1);
  return(0);
}
```

Las dos funciones anteriores nos permiten determinar además si el punto objeto de estudio está o no sobre una de las aristas. Si uno de los signos es igual a cero, quiere decir que el punto "a" estaría en la arista formada por los otros dos.

También las funciones anteriores nos permiten determinar si dos segmentos determinados por los puntos a,b por un lado y b y c por otro se cortan. Para ello es necesario que :

$$\text{signo2d}(a,c,d) = !\text{signo2d}(b,c,d) \ \&\& \ \text{signo2d}(c,a,b) = !(d,a,b)$$

La función que nos permite determinar si un triángulo original está subordinado a otro sería que se muestra a continuación. Se supone que el triángulo determinado por a y b tiene orientación positiva y el determinado por c y d negativa. La función determina si el triángulo ORIGEN,c,d está subordinado a ORIGEN,a,b

```
compara(point a,point b, point c, point d)
{ if (signo2d(c,ORIGEN,b) >=0 && signo2d(b,c,d) > 0 && signo2d(ORIGEN,d,c) > 0
    && signo2d(d,b,ORIGEN) > 0)
  return(1);
  if (signo2d(d,a,ORIGEN) > 0 && signo2d(a,b,d) > 0 && signo2d(ORIGEN,d,c) > 0
    && signo2d(c,ORIGEN,a) > 0)
  return(1);
  if (punto_en(d,a,b) && punto_en(c,a,b))
  return(1);
  return(0);}
```

Cabe destacar en las funciones anteriores que no es necesario acudir a la resolución de sistemas de ecuaciones o a otras operaciones que fácilmente producen inconsistencias. Ya que habitualmente se trabaja con números reales bastará tomar un valor adecuado de ε a la hora de determinar si un valor es cero ó no.

4.6 Subordinación en tres dimensiones

La determinación de subordinaciones en tres dimensiones es bastante más complicada que en dos dimensiones, aunque se pueden generalizar algunos de los resultados obtenidos en dos dimensiones a tres dimensiones.

Estudiaremos primero la subordinación entre tetraedros, que es más sencilla que entre pirámides que será la que estudiaremos al final.

En primer lugar, y al igual que en dos dimensiones introducimos la función signo de cuatro puntos:

```
signo3d(punto3d a, punto3d b, punto3d c, punto3d d)
{
/* positivo: "a" hacia el vector normal del plano b, c y d*/
sig=0;
sig=sig + a.x*(b.y*c.z + b.z*d.y + c.y*d.z - c.z*d.y - d.z*b.y - b.z*c.y);
sig=sig - b.x*(a.y*c.z + a.z*d.y + c.y*d.z - c.z*d.y - a.z*c.y - d.z*a.y);
sig=sig + c.x*(a.y*b.z + b.y*d.z + a.z*d.y - b.z*d.y - d.z*a.y - a.z*b.y);
sig=sig - d.x*(a.y*b.z + b.y*c.z + a.z*c.y - b.z*c.y - c.z*a.y - a.z*b.y);
if (sig > ε) return(1);
if (sig < -ε) return(-1);
return(0);
}
```

El signo será positivo si desde un punto "a" se ven los otros tres puntos en el sentido contrario de las agujas del reloj. En el caso de tetraedros originales, a, b y c serían los puntos vistos desde fuera del sólido y d sería el origen.

A partir de la función anterior se puede determinar también si un punto está incluido ó no en un tetraedro original. Para ello el punto debe ver con la misma orientación cada una de las caras del tetraedro.

```
dentro_tetraedro(punto3d p0, punto3d p1, punto3d p2, punto3d p3)
{
if (signo3d(p1,p2,p3,p0) <=0 && signo3d(p0,ORIGEN3d,p1,p2) >=0 &&
    signo3d(p0,ORIGEN3d,p2,p3) >=0 && signo3d(p0,ORIGEN3d,p3,p1) >=0)
    return(1);
return(0);
}
```

Se supone que el tetraedro original $p_1p_2p_3ORIGEN3d$ está orientado positivamente.

Si uno de los signos es 0 quiere decir que el punto p_0 está sobre esa cara.

En el caso de los tetraedros para ver que no están relacionados (subordinados ó dominantes) basta determinar que ninguno de ellos tiene alguno de sus puntos contenido en la zona angular determinada por el otro. Los casos posibles se indican en la figura 16, en la que se representan la proyección sobre un mismo plano de las caras no originales de los tetraedros.

El sentido de la zona angular es que para que T' este subordinado a T es condición suficiente que alguno de los vértices de T' esté en el interior de T ó que alguno de los vértice de T esté en el interior de la zona angular determinada por T' . Vamos a demostrar formalmente esta propiedad.

Lema 4.14 Sea T y T' dos tetraedros originales. Si alguno de los vértices de T' está en el interior de T o alguno de los vértices de T está en la zona angular determinada por T' entonces T' está subordinado a T .

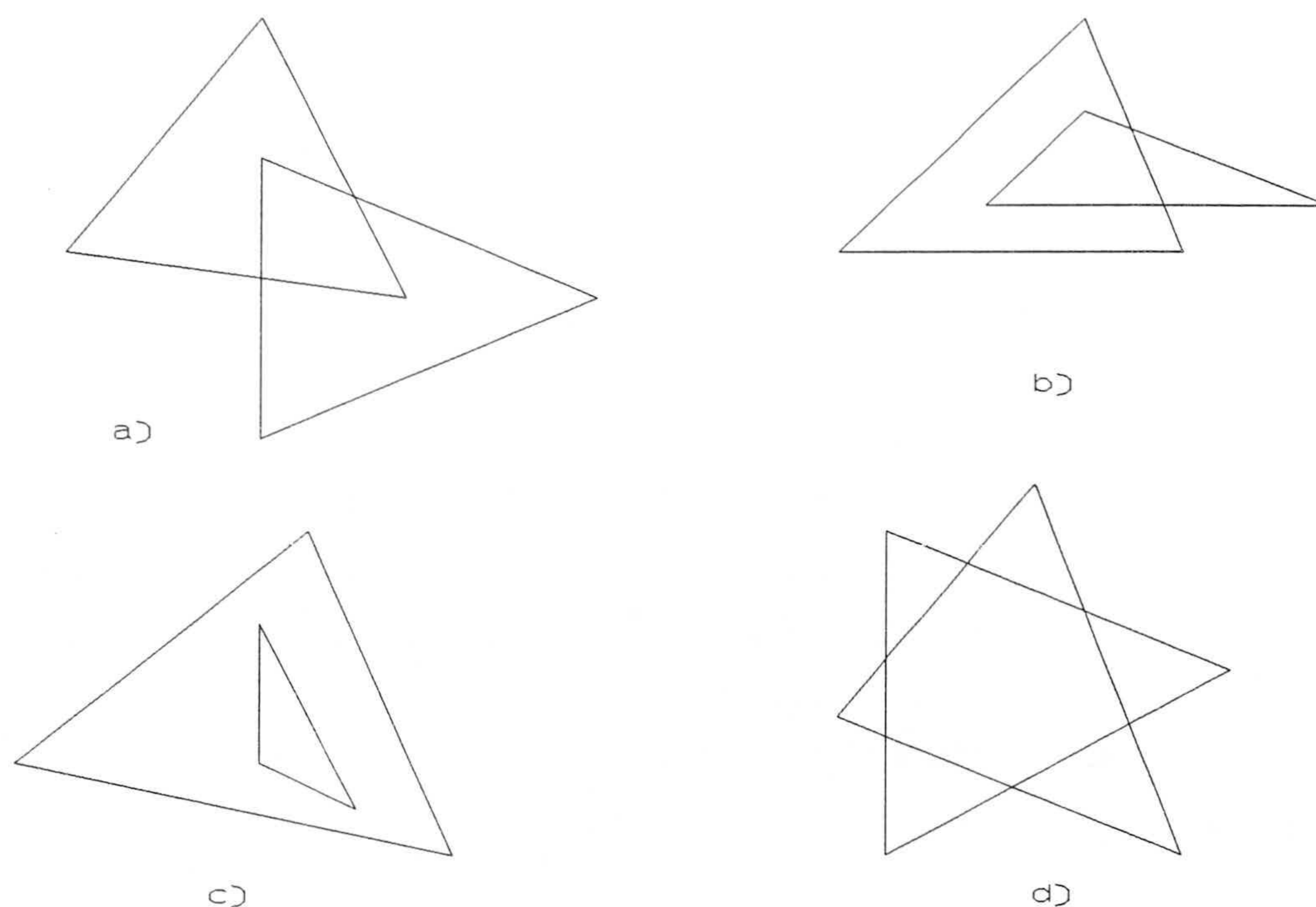


Figura 18 Posiciones relativas de tetraedros originales. Proyecciones

Demostración:

Si alguno de los puntos de T' es interior a T entonces se verifica la condición de subordinación. Si no se da esta condición y uno de los vértices de T está en la zona angular determinada por T' , entonces en la arista de T determinada por ese punto habrá otro que también esté en la base no original de T' , por lo que también se dará la subordinación. ■

La condición del lema 4.14 no es necesaria ya que pueden darse tetraedros subordinados a otros que tengan todos los vértices externos (figura 18 c) de la figura 16). En estos casos es necesario que al menos dos vértices tengan distinto signo respecto de al menos dos caras, tal y como se muestra en la figura 18 d).

La relación de subordinación entre pirámides es similar al caso de los tetraedros aunque existen más casos particulares. Sigue siendo válido el método de estudiar la zona angular determinada por una pirámide, aunque si todos los puntos están fuera puede que haya o no subordinación.

Necesitaremos determinar una función que nos indique si un punto está o no en una pirámide original.

Para la resolución del problema anterior basta aplicar el algoritmo indicado en el capítulo II (tabla II) al caso particular de una pirámide.

Las zonas angulares determinadas por una pirámide facilitan el cálculo de pirámides subordinadas del modo siguiente:

Lema 4.15 Sea P y P' dos pirámides originales. Si alguno de los vértices de P' está en el interior de P ó alguno de los vértices de P está en la zona angular determinada por P' entonces P' está subordinado a P .

Demostración:

Si alguno de los puntos de P' es interior a P entonces se verifica la condición de subordinación. Si uno de los vértices de P está dentro de la zona angular determinada por P' , en esa arista habrá un punto que también esté en la base de P' , por lo que también se dará la subordinación. ■

Sigue existiendo el problema de determinar si hay subordinación o no cuando no se dan las relaciones de inclusión indicadas en el lema anterior.

Al objeto de facilitar el cálculo en algunos de estos casos pueden definirse pirámides envolventes del siguiente modo:

Definición 4.13 Dada una pirámide original $P=OC_1C_2..C_n$, se denomina pirámide cuadrangular envolvente a aquella pirámide original cuya base es un cuadrado coplanario con la base de P , circunscrito a la circunferencia de centro el centro de gravedad de la base de P y radio el máximo de los módulos de los vectores que unen el centro de gravedad de la base de P con cada uno de los vértices, siendo el vértice que corresponde a dicho máximo el punto de tangencia entre el cuadrado y la circunferencia.

La idea de las pirámides cuadrangulares envolventes es facilitar la resolución de algunos algoritmos (tal y como se hace con las cajas envolventes minimales).

Para calcular las pirámides cuadrangulares basta calcular los vectores que unen el centro de gravedad y cada uno de los vértices, calcular el de mayor módulo, su opuesto y el resultado de multiplicar cada uno de los dos por el vector normal unitario al plano.

La relación entre pirámides cuadrangulares envolventes es muy sencillo de estudiar por lo que se puede, en la mayoría de los casos, estudiar si dos pirámides son o no subordinadas a partir de sus pirámides envolventes.

Las pirámides cuadrangulares envolventes facilitan determinar rápidamente muchos de los casos en que dos pirámides originales no tendrán intersección y por tanto no habrá relación de subordinación.

El estudio de relación entre cajas equivale a estudiar la relación de puntos respecto a planos.

Pueden darse casos en que las pirámides cuadrangulares envolventes se intersecten, no se den las condiciones del lema 4.15 y no se sepa decidir si existe ó no relación de subordinación. La figura 19 nos muestra dos casos:

En esos casos es necesario evaluar si una de las aristas de una de las pirámides (P) atraviesa a una de las caras de la otra (P') (atraviesa ya que si el otro fuese interior ya se habría evaluado). Para ello bastaría ver que el vértice objeto de discusión (arista de corte) y los dos vértices contiguos de la base de P tienen signo distinto respecto del plano determinado por el segmento de interés de P' y el origen (dirección de corte de fuera a dentro).

El siguiente lema permite el estudio de esos casos particulares (figura 19 a.):

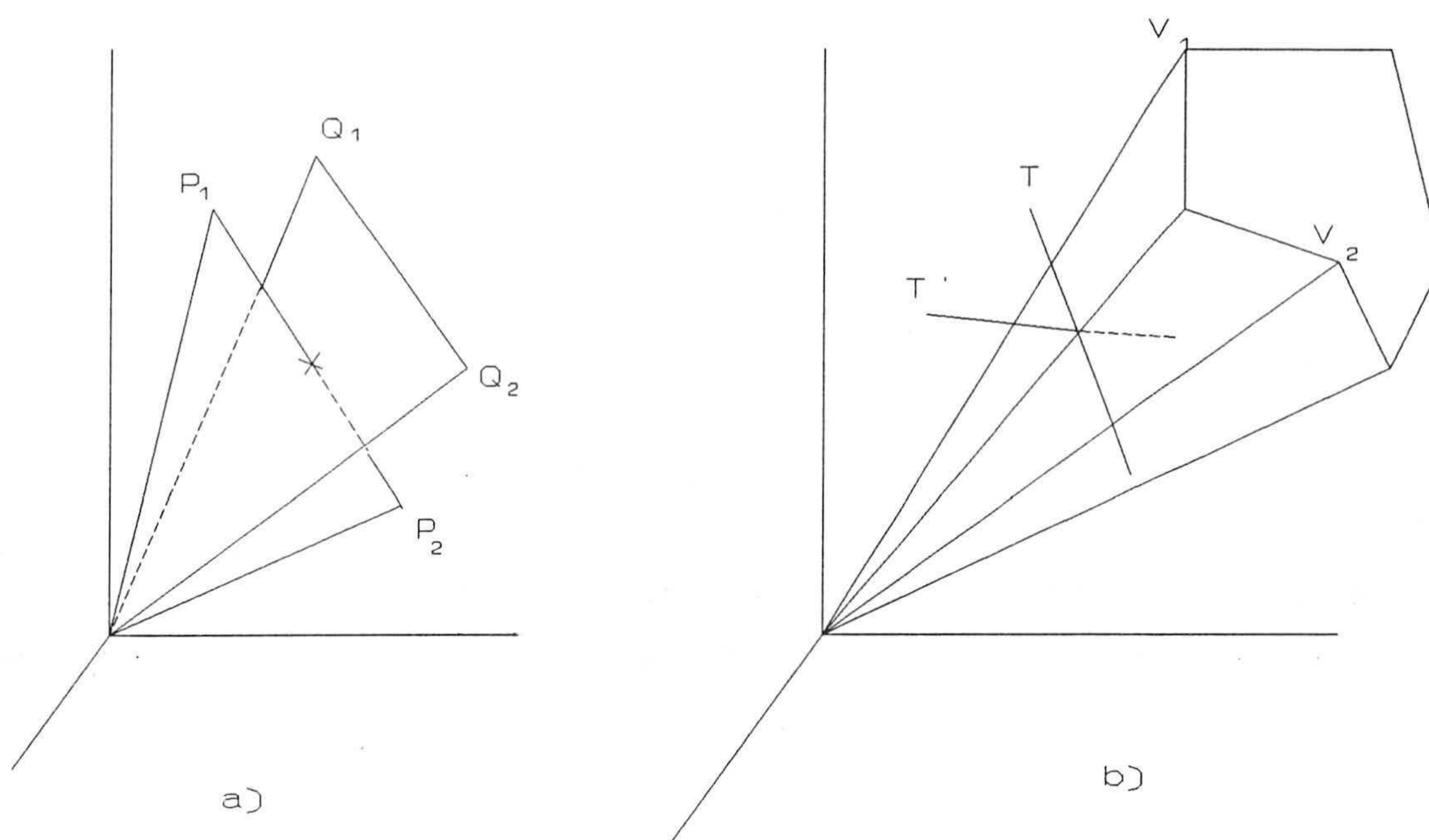


Figura 19. Casos especiales de subordinación

Lema 4.16 Sea P_1P_2 y Q_1Q_2 dos segmentos del espacio. Entonces es condición necesaria y suficiente para que los triángulos originales determinados por dichas aristas se corten es que

$$\text{sign}(OP_1P_2Q_1) < > \text{sign}(OP_1P_2Q_2) \text{ y } \text{sign}(OQ_1Q_2P_1) < > \text{sign}(OQ_1Q_2P_2).$$

Además, las aristas se cortan sí y solo sí los cuatro puntos son coplanarios.

Demostración:

Si hay un punto de corte, esto equivale a que cada punto está a un lado del plano determinado por el origen y el otro segmento y por tanto es equivalente a que los signos sean distintos. Si además los cuatro puntos de corte son coplanarios, equivale a que el punto de corte debe estar en el mismo plano y por tanto los segmentos se cortan.

Si uno de los signos es 0, quiere decir que el punto de corte está en una de las aristas originales de uno de los triángulos.

Un punto está en un segmento si coincide con uno de sus extremos, ó bien si los vectores determinados por dicho punto y los extremos tiene producto vectorial cero y producto escalar negativo. ■

El lema anterior nos permite estudiar la relación entre una arista de una pirámide original y las aristas de la otra.

El único caso que queda es poder determinar si un segmento penetra en una pirámide cortando a una aristas ó no. Por un lado tiene que tener dos puntos de corte, cortar a una arista común a dos triángulos originales, laterales de la pirámide original que se está tratando.

Para que una arista no penetre en una pirámide, aunque tenga un punto de corte coincidente con una de las aristas originales de la pirámide basta que el vértice de dicha arista original y sus dos vértices contiguos tenga el mismo signo respecto del plano determinado por el origen y la arista objeto de estudio (Figura 19 b).).

Es importante destacar que en ninguno de los algoritmos anteriores es necesario calcular el punto de intersección por lo que se evitan muchos problemas de robustez en la implementación.

Vamos, por último a aplicar los resultados obtenidos en los capítulos precedentes a la obtención de una método de conversión de B-rep a CSG, para poliedros planos.

4.7 Conversión de B-rep a CSG

En el capítulo dos hemos dado un método formal que permite, para cualquier sólido determinado por caras planas y expresado a partir de su frontera, obtener una representación CSG equivalente, en la que se toman como primitivas 2-símplices ó 3-símplices y como operaciones, todas las del álgebra.

Una vez obtenido, a lo largo de este capítulo, un método de modelado basado en recubrimientos simpliciales aplicable a cualquier sólido de caras planas, podemos obtener no solo un método formal basado en el álgebra de objetos gráficos sino también un método aplicable a los sistemas de representación actuales.

Dobkin et al[DoGHS88], después de resolver eficientemente el problema de encontrar una representación CSG de polígonos simples, demuestran que una generalización del método a tres dimensiones no siempre es posible, planteando como cuestión abierta la posibilidad de representar el interior de un poliedro de n caras, usando como máximo n símbolos. Es cierto que en ese trabajo, al igual que en otros que tratan la conversión de B-rep a CSG, se parte de primitivas que son semiespacios.

El objetivo de una representación CSG es obtener un sólido por medio de un árbol en el que sólo se usen un número determinado de primitivas para generar el sólido. Pensamos, por ello, que el tipo de primitivas no tiene que estar restringido a semiplanos, sino que se pueden considerar cualquier primitiva siempre que esta sea sencilla de definir y tratar.

A lo largo de este capítulo, hemos demostrado que todo sólido puede ser obtenido a partir de un árbol de símlices de dimensión n : todo sólido bidimensional a partir de triángulos (2-símlices) y todo sólido tridimensional a partir de tetraedros (3-símlices). Puede, por ello, darse el siguiente resultado final:

Teorema 4.10 Sea O un sólido del cual se supone conocida su representación de fronteras. Entonces O puede ser representado mediante un árbol CSG cuyas primitivas son todas tetraedros, y verificándose que $n = e_d$, donde n indica el número total de tetraedros es igual al número total de aristas orientadas.

Demostración:

Supongamos que O tiene v vértices, c caras y a aristas, y que no existen aristas ni vértices aislados (es decir, todo vértice es adyacente, al menos, a tres aristas y a dos caras).

Dada una representación B-rep de un sólido (que supondremos separado, según la definición 4.8), podemos obtener su representación en función de aristas dirigidas e_d . Dado el conjunto de vértices v consideramos como punto de referencia \mathbf{R} el centro de gravedad del sólido, y para cada cara, el punto \mathbf{R}_c , centro de gravedad de los vértices que corresponden a esa cara. Para cada arista orientada, equivalente a un par (a,c) , definimos el tetraedro determinado por \mathbf{R} , \mathbf{R}_c y la arista a . Por el teorema 4.7, el sólido se puede obtener a partir del árbol definido por las pirámides que definen el punto \mathbf{R} y cada una de las caras, y cada pirámide puede obtenerse a partir de los tetraedros definidos por el punto \mathbf{R} , el punto \mathbf{R}_c de cada cara y cada una de las aristas de la cara, con lo que se obtiene el teorema.

Ya que definimos un tetraedro por cada arista orientada, es evidente que se obtiene la igualdad entre el número de tetraedros y el número de aristas orientadas. ■

4.8 Conclusiones

Hemos definido en este capítulo un nuevo sistema de representación aplicable a sólidos cuya frontera está determinada por caras planas.

En primer lugar, y basándonos en el concepto de subordinación entre símlices, hemos obtenido una representación que no es minimal ya que pueden repetirse las primitivas que entran en la representación.

Posteriormente hemos definido una relación de orden entre los simples asociados a cada sólido. Dicha definición permite obtener descomposiciones en las que cada primitiva interviene una sola vez.

Se han estudiado las operaciones entre sólidos representados por recubrimientos simpliciales, especialmente la operación de intersección.

Se han introducido algunas funciones que facilitan el cálculo de subordinaciones en dos y tres dimensiones.

Finalmente se da un teorema para la conversión de la representación de fronteras de un sólido definido por caras planas a una representación CSG cuyas primitivas son simples.

CAPITULO V

Conclusiones y principales aportaciones

A lo largo de esta tesis hemos aportado un método de modelado de sólidos poliédricos, basado en recubrimientos simpliciales.

Partiendo de la formalización presentada por el Dr. Torres, que es una generalización de la propuesta por Fiume, se han definido las operaciones regularizadas dentro del álgebra de objetos gráficos y se ha demostrado que el conjunto de objetos gráficos con esas operaciones regularizadas mantienen las mismas propiedades que las operaciones usuales.

Posteriormente se ha generalizado el álgebra gráfica para poder trabajar con representaciones B-rep, lo que en cierto modo completa el alcance del sistema formal en relación al modelado de sólidos: la representación CSG, de octrees y de barrido ya había sido presentada, por lo que al abarcar también las representaciones B-rep se puede decir que a nivel formal se resuelve el enlace entre los diversos métodos de representación.

La formalización de la representación B-rep se ha realizado obteniendo a partir de la frontera un conjunto de elementos a partir de los cuales se puede obtener todo el sólido: de hecho, se ha obtenido un sistema de generadores dentro del espacio vectorial

de los objetos gráficos. Dicho sistema de generadores está formado por elementos muy sencillos y fáciles de tratar con los que se pueden obtener representaciones canónicas de cualquier objeto. Se ha estudiado la relación entre las descomposiciones canónicas obtenidas y el resto de las operaciones del álgebra, especialmente la booleanas: unión, intersección, complementación y diferencia conjuntista. Se han estudiado así mismo las diversas posiciones que pueden tener entre sí los símlices tanto bidimensionales como tridimensionales, lo que facilitará la implementación de dichas operaciones.

Como una aplicación de estos conceptos se ha presentado un algoritmo de inclusión. El algoritmo se basa en la descomposición del sólido en símlices, por lo que para determinar si un punto está, o no, en el sólido inicial bastará estudiar si está incluido en cada uno de los elementos generadores y posteriormente operar correctamente entre dichas inclusiones. Se ha demostrado que es correcto tanto en el caso de dos como de tres dimensiones y se ha presentado que reduce el tiempo de ejecución en relación con otros algoritmos y aumenta la estabilidad debido a no necesitar la resolución de sistemas de ecuaciones.

Los resultados obtenidos se han aplicado para obtener un sistema de modelado basado en recubrimientos simpliciales. Inicialmente se obtienen unos resultados que permiten expresar cualquier sólido en función de símlices asociados a sus elementos singulares (aristas ó caras). La representación obtenida no es minimal ya que algunos elementos aparecen más de una vez en la fórmula. Para obtener fórmulas minimales se ha introducido una relación de orden para objetos separados (se demuestra que todo objeto se puede separar), lo que permite definir el concepto de orden y capa. A partir de estos elementos sí que es posible obtener fórmulas en los que cada elemento aparece una sola vez. De este modo, para cada sólido se obtiene una representación basada en un árbol CSG obtenido a partir de elementos muy sencillos: triángulos originales en dos dimensiones y pirámides o tetraedros originales en tres dimensiones. Se ha elegido el origen de coordenadas por comodidad, aunque se podría hacer el mismo desarrollo para otro punto de referencia.

Se han estudiado las diversas operaciones entre sólidos. La complementación se hace en función de un conjunto universal que coincide con una caja bi ó tridimensional, lo suficientemente grande para que, a efectos prácticos, sea equivalente al espacio total.

La unión y la diferencia conjuntista se obtienen a partir de la intersección, por lo que es esta operación la que se estudia con más detalle.

La intersección se describe como una operación obtenida a partir de la evaluación de fronteras de la expresión CSG obtenida. La ventaja en este caso es que los órdenes se pueden ir obteniendo casi simultáneamente a los diversos elementos.

Aunque es probable que en muchos casos no sea necesario evaluar el árbol CSG obtenido, se describe el algoritmo de intersección como una operación a obtener a partir de las operaciones entre los símlices asociados y su posterior reducción.

Finalmente se estudia la posibilidad de obtener un árbol CSG basado en tetraedros, en número igual al de aristas orientadas, a partir de su representación B-rep, usando como aplicación los resultados obtenidos a lo largo de esta tesis.

Son diversos los campos posibles de desarrollo y de futuros trabajos:

- . en primer lugar, la completa implementación del sistema de modelado presentado y de las operaciones a realizar entre sólidos.
- . la aplicación de las técnicas de ray casting para la visualización de objetos representados por el método expuesto en el capítulo cuarto de esta tesis.
- . la extensión del modelo para la representación de sistemas e indexaciones espaciales, que parece posible debido a la estructura jerárquica del modelo obtenido.

Otro campo de futuro desarrollo supone la extensión del modelo a sólidos definidos por caras no planas.

Bibliografía

Junto a la bibliografía directamente referenciada a lo largo de esta tesis se ha incluido también otra que aunque no directamente referenciadas puede ser de gran interés en relación al trabajo de esta tesis.

- Ayala88 Ayala, D. "Boolean Operations between solids and surfaces by octrees; models and algorithms", *Computer Aided Design*, vol. 20,8, pp. 452-465 (1988).
- AyBRN85 Ayala D.; Brunet, P.; Juan, R.; Navazo, I.: "Object representation by means of nonminimal division quadtrees and octrees", *ACM TOG*, 4,(1), January 1985.
- BaeEH79 Baer, A.; Eastman, C.; Henrion, M. "Geometric modelling: a survey", *CAD*, vol 11 (1979), n° 5 pp 253-272
- BajDe92 Bajaj, C.; Dey, T.; "Convex decomposition of polyhedra and robustness", *SIAM Journal Computing*, vol.21, No. 2, pp. 339-364, april 1992
- Baung75 Baungart, B.G., "A polyhedron Representation for Computer Vision", *NCC*, 75, 589-596
- Berge87a Berger, M. *Geometry I*, Springer-Verlag, 1987
- Berge87b Berger, M. *Geometry II*, Springer-Verlag, 1987
- BroOw93 Brodlie, K; Owen, G.S.; eds. *Eurographics Workshop on Graphics and Visualization Education*. Eurographics, 1993
- BroPo88 Bronsvoort, W.F. y Post, F.H., "Geometric Modelling" en *Advance in Computer Graphics III*, Ed. Ruiter MM.
- Brons90 Bronsvoort, W.F. *Direct display algorithms for solid modelling*. Ph.D.thesis. Delft University Press, 1990

- BruAy87 Brunet, P., and Ayala, D. "Extended octree representation of free form surfaces". *Computer Aided Geometric Design* vol.4 (1987), 141-154
- BruNa85 Brunet, P.; Navazo, I.; "Geometric Modelling of Volumes", *Proc. Eurographics 85*, North Holland, pp. 159-169, 1985
- BruNa92 Brunet, P.; Navazo, I.; "Geometric Modelling of Volumes", *Eurographics 92*, Tutorial Notes. Eurographics, 1992
- CamRo89 Cameron, S.A.; Rossignac, J.R. "Relationship between S-bound and Active Zones in Constructive Solid Geometry", en W. Straser, ed. *Theory and Practice of Solid Modelling*, Springer Verlag, 1989
- CarCV85 Carlbom, I.; Chakravarty, I.; Vanderschel, D. "A Hierarchical Data Structure for Representing the Spatial Decomposition of 3-D Objects". *IEEE Computer Graphics and Applications*, vol 5 (1985) n° 4, pp. 24-31
- ChaPa90 Chazelle, B.; Palios, L.; "Triangulating a Nonconvex Polytope", *Discrete Computational Geometry* vol 5 (1990) pp. 505-526
- Chaze91 Chazelle, B. "Triangulating a Simple Polygon in Linear Time", *Discrete Computational Geometry* vol 6 (1991) pp. 485-524
- Chiyo88 Chiyokura, H. *Solid Modelling with Designbase*, Addison Wesley, 1988
- DeyBS91 Dey, T.; Bajaj, C.; Sugihara, K. "On Good Triangulations in Three Dimensions", in *Proceedings on Solid Modeling Foundations and CAD/CAM Applications*, Austin, TEXAS, ACM. Junio 5-7, 1991
- Dey__91 Dey, T.K. "Triangulation and CSG representation of Polyhedra with Arbitrary Genus", *Seventh ACM Symposium on Computational Geometry*, ACM Press pp 364-372, 1991
- DoGHS88 Dobkin D.; Guibas L.; Hershberger, J; Snoeyink, J. "An efficient Algorithm for Finding the CSG Representation of a Simple Polygon", *Computer Graphics*, Vol. 22, N.4, August 1988

-
- Du_Hw92 Du, Ding-Zhu; Hwang, Frank; eds. *Computing in Euclidean Geometry*, World Scientific 1992
- Duce_89 Duce, D.A. "Formal Methods in Computer Graphics", Eurographics'89, State of the Art Seminar, Eurographics 1989.
- Duce_91 Duce, D.A. "Report on the EUROGRAPHICS Workshop on Formal Methods in Computer Graphics", *Computer Graphics Forum*, vol 10, nº 4 pp 311-327 1991.
- Farin87 Farin, G.E., ed. *Geometric Modeling: Algorithms and new trends*, SIAM 1987.
- Farin89 Farin, G.E. *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, 2ª edi. 1989.
- Feito93 Feito, F.; "Fundamentos de modelado de sólidos", Report Interno, 1993, Departamento de Lenguajes y Sistemas Informáticos. Universidad de Granada, 1993.
- FeiTU92 Feito, F.; Torres, J.C.; Ureña, A. "Simplicidad, orientación y test de inclusión de polígonos planos", Universidad de Granada, Dpt. Lenguajes y Sistemas Informáticos. Internal Report 92.4.
- Fiume89 Fiume, E.L. *The mathematical Structure of Raster Graphics*, Academic Press, Boston 1989
- FlaCM87 Flaquer, J.; Carbajal, A.; Mendez, M.A. "Edge-edge relationships in geometric modelling" *CAD*, vol 19, nº 5 (1987) pp 237-244
- FoVFH90 Foley, J., Van Dam, A., van Dam, A., Feiner, S., Hughes, J. *Computer Graphics: Principles and Practice*, 2ª Edición, Addison Wesley, 1990.
- Garga82 Gargantini I., "Linear octrees for fast processing of the tridimensional objects". *Computer Graphics & Image Processing* 20(4):365-375.

- Ghosh93 Ghosh, P.K. "A Solution of Polygon Containment, Spatial Planning and Other Related Problems Using Minkowski Operations" *Computer Vision, Graphics and Image Processing* vol 49 (1990) pp. 1-35
- GurCP90 Gursoz, E.L.; Choi, Y; Prinz, F. "Boolean set operations on non-manifold boundary representation objects", *CAD*, vol 23, n° 1, january-february 1991
- GurCP90 Gursoz, E.L.; Choi, Y; Prinz, F. "Vertex-based representation of non-manifold boundaries" in Wozny, M; Turner, J, and Preiss, K(eds) *Geometric Modeling for Product Engineering* North Holland (1990), pp 107-130
- HagRo91 Hagen, H.; Roller, D; eds. *Geometric Modeling*, Springer-Verlag, 1991
- Hilbe30 Hilbert, D. *Grünlagen der Geometrie*, 7ª edición, Gotinga (traducción al español: *Fundamentos de la geometría*, Colección Textos Universitarios del CSIC, n° 5, Madrid 1991)
- Hoffm89 Hoffmann, *Geometric and Solid Modeling*, Morgan Kaufmann, 1989
- HofHK89 Hoffmann, C.M., Hopcroft, J.E. and Karasick, M.S. "Robust set operation on polyhedral solids", *IEEE Computer Graphics and Applications*. vol 9 (1989), pp 50-59
- JanBP90 Jansen, E.; Bronsvoort, W; Post, F. "Design and Display of Solid Models", *Eurographics 90*, Tutorial Notes. Eurographics, 1990
- Janse87 Jansen, E. *Solid modelling with faceted primitives*. Ph.D.thesis. Delft University Press, 1987
- Juan_88 Juan, R. "Boundary to Constructive Solid Geometry: a step towards 3D Conversion", *Eurographics 88*, Duce, D.A.; Jancene, P. edit. , Elsevier Science, 1988
- Juan_89 Juan, R. "On boundary to CSG and extended octrees to CSG conversions", en W. Straser, ed. *Theory and Practice of Solid Modelling*, Springer Verlag, 1989

-
- Kalay82 Klay, Y.E. "Determining de Spatial Containment of a Point in General Polyhedra", *Computer Graphics and Image Processing*, vol 19(1982) n° 1, pp 203-334
- Kalay88 Kalay, Y.E. *Modeling Objets and Environments*, Wyley Interscience, 1989
- Karas88 Karasick, M.S. *On the Representation and Manipulations of Rigid Solid*. Ph.D.thesis, Departament of Computer Science, McGill University, montreal, Canada, 1988
- Karas89 Karasick, M.S. "The Same-Object Problem for Polyhedral Solids", *Computer Vision, Graphics and Image Processing*, vol 46 (1989), pp 22-36
- KunSY85 Kunii, T.L.; Satoh, T.; Yamaguchi, K. "Generation of Topological Boundary Representations from Octree Encoding", *IEEE Computer Graphics and Applications* vol 5 (1985), n° 3, pp. 29-38
- Kurat66 Kuratowski, K. *Introducción a la Teoría de Conjuntos y a la Topología*, Ed. Vicens-Vives, 1966
- LaiTH86 Laidlaw, D.H., Trumbore, W.B., Hughes. "Constructive solid geometry for polyhedral objects. *Proc ACM SIGGRAPH'86, Computer Graphics* 20(4): 122-134
- Masse72 Massey, W.S. *Introducción a la Topología Algebraica*, Ed. Reverte 1972
- Mänty88 Mäntylä, M., *An Introduction to Solid Modeling*, Computer Science Press, 1988
- Meagh82 Meagher D., "Geometric Modelling using octree encoding", *Computer Graphics and Image Processing* 19(2):129-147
- Mille89 Miller, J.R. "Architectural Issues in Solid Modelers", *IEEE Computer Graphics and Applications* vol.9 No.5, pp. 72-87, September, 1989
- Mille93 Miller, J.R. "Incremental Boundary Evalutaion Using Inference of Edge Classifications" *IEEE Computer Graphics and Applications* vol 13 (1993) n° 1 , pp. 71-78

- Morte85 Mortenson, M.E. *Geometric Modeling*, John Wiley&Sons, 1985
- NavAB86 Navazo, I., Ayala, D. y Brunet, P., "A Geometric modeler based on the exact octree representation of polyhedra". *Computer Graphics Forum* 5 (2), 91-104.
- NavBF87 Navzo, I.; Brunet, P.; Fontdecaba, J. "Extended Octrees, Between CSG Trees and Boundary Representation. *Proc. Eurographics '87*, North Holland, pp. 239-247, 1987
- O'Rou94 O'Rourke, J. *Computational Geometry in C*, Ed. Cambridge University Press, 1994.
- PauRS89 Paoluzzi, A.; Ramella, M.; Santarelli, A. "Boolean algebra over linear polyhedra", *CAD*, vol 21 , pp. 474-484, 1989
- PauRS93 Paoluzzi, A.; Ramella, M.; Santarelli, A. "Dimension-independent Modeling with Simplicial Complexes", *ACM Transaction on Graphics*, vol 12 , n° 1, pp 56-102, 1993
- Peter86 Peterson, D.P. "*Boundary to constructive solid geometry mappings: a focus on 2D issues*", C.A.D., vol.18, n°1, January 1986.
- PosK186] Post, FH., Klok, F., "Deformation of sweep objects in solid modelling". In Requicha AAG(ed) *Proc. Eurographics '86*. Elsevier Science Publishers BV (North-Holland), Amsterdam , p.103
- PreSh85 Preparata, FP. Shamos, M.I., *Computational Geometry: An introduction*, Springer-Berlag, New York 1985
- PutSu86 Putnan, L.K.; Subrahmanyam, P.A., " Booleana Operations on n-dimensional objects", *IEEE Computer Graphics and Applications*, vol n° 3 pp 43-51 1986
- Requi77 Requicha, A.A.G., *Mathematical Models of Rigid Solids. TM-28, Production Automation Project*. University of Rochester. 1977
- Requi78 Requicha, A.A.G., *Mathematical Foundations of Constructive Solid Geometry: General Topology of Closed regular Sets. TM-27a, Production Automation Project*. University of Rochester. 1978

- Requi80 Requicha, A.A.G. "Representations for Rigid Solids: Theory, Methods, and Systems", *Computing Survey*, Vol 12, No. 4, December 1980
- Requi92 Requicha, A.A.G.; Rossignac, J.; "Solid Modeling and Beyond", *IEEE Computer Graphics and Applications*, vol 12, n° 5, pgs. 31-44
- ReqVo82 Requicha, A.A.G.; Voelcker, H. "Solid modeling: a historical summary and contemporary assessment", *IEEE Computer Graphics and Applications*, vol.2, n° 2, 1982
- ReqVo85 Requicha, A.A.G.; Voelcker, H.: "Boolean operation in solid modelling: boolean operation and merging algorithms" *Proc. IEEE*, Vol. 73, N.1, January pp. 30-44, 1985.
- ReqVol83 Requicha, A.A.G.; Voelcker, H. "Solid modeling: current status and research directions", *IEEE Computer Graphics and Applications*, vol.3, Octubre 1983
- Rossi93 Rossignac, J. "Representations, Design and Visualization of Solids and Geometric Structures", *Eurographics 93*, Tutorial Notes. Eurographics, 1993
- RosOc90 Rossignac, J; O'Connor, M.A. "SGC: a dimension-independent model for pointset with internal structures and incomplete boundaries" en Wozny,M; Turner, J, and Preiss, K(eds) *Geometric Modeling for Product Engineerin* North Holland (1990),pp 107-130
- Rossi91a Rossignac, J. edit. "Special Issue: Beyond Solid Modeling", *CAD*, Vol 23, No. 1, January/February 1991
- RosTu91b Rossignac, J.; Turner, J.; edi. "*Symposium on Solid Modeling and CAD/CAM Applications*", ACM PRESS, 1991
- RosVo89 Rossignac, J.; Voelcker, H. "Active Zones in CSG for Accelerating Boundary Evaluation, Interference Detection, and Shading Algorithms", *ACM Transactions on Graphics*, Vol. 8 (1989), N0. 1
- SamWe88 Samet, H.; Webber, R. "Hierarchical Data Structures and Algorithms for Computer Graphics. Part I. Fundamentals", *IEEE Computer Graphics and Applications* vol 8 (1988) n° 3, pp. 48-68

-
- SamWe88 Samet, H.; Webber, R. "Hierarchical Data Structures and Algorithms for Computer Graphics. Part II. Applications", *IEEE Computer Graphics and Applications* vol 8 (1988) n° 4, pp. 59-75
- SamTa85 Samet, H; Tamminen, M. "Bintrees, CSG Trees and Time", *ACM Computer Graphics* 19 (1984), n° 3 pp. 121-130
- Samet90a Samet, H., *The Design and Analysis of Spatial data Structures*, Addison Wesley, 1990
- Samet90b Samet, H. *Applications os Spatial Data Structures: Computer Graphics, Image Processing and GIS*, Addison Wesley, 1990
- SegSe88 Segal, M. and Sequin, C.H. "Partitioning polyhedral objects into nonintersecting part" *IEEE Computer Graphics and Applications* vol 8 (1988), pp 53-67
- ShaPe91 Shapidis, N.S.; Perucchio, R. " Domain Delanuay Tetraedrization of Arbitrarily Shaped Curved Polyhedra Defined in a Solid Modeling System", *Proceeding of the Symposium of Solid Modeling Foundations and CAD/CAM Applications*, ACM Press 1991.
- Shapi91 Shapiro, V. *Representation of semi-algebraic sets in finite algebras generated by space decompositions*, Pd Thesis. Cornell University, 1991
- ShaVo92 Shapiro, V.; Vossler, D.L. "Separation for Boundary to CSG Conversion", *ACM Transaction on Graphics*, Vol.12,No.1, January 1993, Pages 35-55.
- Silva81 Silva, C.E. Alternative Definitions of faces in bopoundary representations of solid objects. TM-36, *Production Automation Project*. University of Rochester. 1981
- Snyde92 Snyder, J.M. *Generative Modeling for Computer Graphics and CAD*, Academic Press 1992
- StrSe89 Straser, W; Seidel, H.; edits. *Theory and Practice of Geometric Modeling*, Springer-Verlag, 1989

- TarVa88 Tarjan, R.E.; Van Wyk, C.J.: "An $O(n \log \log n)$ -Time algorithm for triangulating a simple polygon". *SIAM Journal Computing*. 17(1). February 1988.
- ThiNa87 Thibault, W.C. and Naylor, B.F. "Set operations on polyhedra using binary space partitioning trees", *Proc ACM SIGGRAPH'87, Computer Graphics* vol 21 (1987) pp 153-162
- Tilov80 Tilove, R.B. "Set membership classification: a unified approach to geometric intersection problems", *IEEE Transaction on Computer* vol c-29 n° 10 (1980) pp 874-883
- TilRe80 Tilove, R.B. ; Requicha, A.A.G. "Closure of Boolean operations on geometric entities", *CAD*, vol.12, n° 5, September 1980
- Torre92 Torres, Juan C. *Representación abstracta de sistemas gráficos. Teoría de objetos gráficos*, Tesis Doctoral. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada, Granada 1992
- TorCl93 Torres, J.C.; Clares, B. "Graphic objects: A mathematical abstract model for computer graphics", *Computer Graphics Forum*, vol 12 (1993), n° 5, pp 311-327
- TorCl94 Torres, J.C.; Clares, B. "A Formal Approach to the Specification of Graphic Object Functions" *Computer Graphics Forum*, vol 13 (1994), n° 3, pp 371-380
- Turne88 Turner, J. "Accurate Solid Modeling Using Polyhedral Approximations", *IEEE Computer Graphics and Applications* vol 8 (1988), n°3, pp. 14-28
- Vanec89 Vanecek, G. Set Operation on Volumes Using Decomposition Methods. PhD thesis, University of Maryland, College Park, 1989
- VoeRe93 Voelcker, H.B.; Requicha, A.A.G. "Research in Solid Modeling at the University of Rochester:1972-87" en Pielg L. ed. *Fundamental Developments of Computer-Aided Geometric Modeling*, Ed. Academic Press, 1993

-
- Voss185 Vossler, D.L. "Sweep to CASG conversion using pattern recognition techniques", *IEEE Computer Graphics and Applications* vol 5 (1985), n° 4, pp 61-68, 1985
- Weile85 Weiler, K.J. "Edge-Based Data Structures for Solid Modeling in Curved-Surface Environments", *IEEE Computer Graphics and Applications* vol 5 (1985), n°1 pp 21-40
- Weile88 Weiler, K.J. "The radial Edge Structure: A topological Representation for Non-Manifold Geometric Modeling" en Wozny, M; McLaughlin H and Encarnacao, J.(eds) *Geometric Modeling for CAD Applications*, North Holland (1988), pp 3-36.
- Wijk_86 Wijk, JJ van, "SML: a solid modelling language", *CAD* 18(8): 443-449
- Woo__85 Woo, T.C.; "A combinatorial Analysis of Boundary Data Structure Schemata", *IEEE Computer Graphics and Applications* vol 5 (1985), n° 2, pp 19-27
- YamTo84 Yamaguchi, F. and Tokieda, T. "A unified algorithm for Boolean shape operations" *IEEE Computer Graphics and Applications* vol 4 (1984), pp 24-37
- Yamag85 Yamaguchi, F. "A unified approach to interference problems using a triangular processor" Proc ACM SIGGRAPH'85 *Computer Graphics* vol 19, n° 3 (1985) pp 141-149.