

비다양체 모델을 수용하는 CAD 시스템 커널을 위한 불리안 조작의 개발

김성환*, 이건우**, 김영진***

Development of Boolean Operations for CAD System Kernel Supporting Non-manifold Models

Sung-Hwan Kim*, Kunwoo Lee**, Young-Jin Kim***

ABSTRACT

The boundary evaluation technique for Boolean operation on non-manifold models which is regarded as the most popular and powerful method to create and modify 3-D CAD models has been developed. This technique adopted the concept of Merge and Selection in which the CSG tree for Boolean operation can be edited quickly and easily. In this method, the merged set which contains complete information about primitive models involved is created by merging primitives one by one, then the alive entities are selected following the given CSG tree.

This technique can support the hybrid representation of B-rep(Boundary Representation) and CSG(Constructive Solid Geometry) tree in a unified non-manifold model data structure, and expected to be used as a basic method for many modeling problems such as data representation of form features, and the interference between them, and data representation of conceptual models in design process, etc.

Key words : Non-manifold model, Boolean operation, Euler operator, Merge and selection, History record

1. 연구 배경

설계 분야에서 CAD 시스템의 유용성은 이미 광범위하게 인식되어 제품 개발 과정의 효율을 많이 향상시켜 왔다. 그러나 이러한 기존의 CAD 시스템들은 그 활용에 있어 약점을 가지고 있는데, 하나는 표현할 수 있는 형상모델의 제약성이고 다른 하나는 통합 시스템으로서의 폐쇄성이다. 여기서 형상모델의 제약성이라고 하는 것은 기존의 시스템들이 다루는 대상이 다양체(2-manifold)로 한정되어 있어 설계의 최종결과만을 저장할 수 있을 뿐 설계과정 중의 불완전

한 물체, 해석을 위한 축약해석모델, 설계의도등을 표현할 수 없다는 것이다. 또 통합시스템으로서의 폐쇄성이란 터키시스템의 형태를 띄는 기존의 범용 모델링 시스템은 일반 모델링 기능으로부터 각자의 적용 분야에 적합한 기능으로의 재구성, 보다 편리한 사용자 인터페이스로의 대처등 사용자 요구 환경으로 시스템을 재구축할 수 없다는 것을 일컫는다.

이러한 제약성들을 극복하기 위해서 CAD 시스템은 우선 다양체뿐만 아니라 비다양체(Non-manifold) 모델⁽¹⁾을 지원함으로써 표현 모델의 영역을 넓혀야 하며, 개방형 커널 시스템의 형태를 가져서 사용자가 함수 형태로 제공되는 모델링 기능들을 사용한 프로그램을 개발함으로써 자신의 요구에 맞는 고유한 모델링 시스템을 구축할 수 있도록 하여야 한다. 이러한 요구에 부응하여 서울대학교 CAD 연구실에

*정회원, 대우공업전문대학 기계과

**중신회원, 서울대학교 기계설계학과

***정회원, Dept. of General Engineering, Univ. of Illinois at Urbana-Champaign

서는 비다양체 기반의 모델링 커널 시스템을 개발하고 있으며⁽²⁾, 본 논문은 이의 일환으로 개발된 모델링 기능인 불리안 조작의 개발 내용을 기술한다.

2. 관련 연구 및 불리안 작업의 개요

비다양체(Non-manifold)에 관한 초기의 연구는 주로 비다양체를 표현하기 위한 자료구조의 제시와 이에 대한 충분성 및 효율성을 입증하는 것에 초점이 맞추어져 왔다.^{(1), (3-5)} 한편 일단 비다양체 모델을 표현할 수 있게 되면서 이의 모델링 방법이 필수적으로 필요하게 되는데 가장 기본적이고, 사용자에게 친숙하면서도 강력한 모델링 방법은 과거 다양체(2-manifold) 모델링 시스템의 경우처럼 역시 불리안 작업이 손꼽히고 있으며, 이러한 관점에서 비다양체에 대한 불리안 기능의 개발에 관한 연구가 이루어지고 있다.⁽⁶⁻⁸⁾ 이들 기존의 연구들이 가지는 공통점은 병합과 선택(Merge & Selection)의 개념을 사용한다는 것인데 Fig. 1은 이를 도식적으로 보여 주고 있다.

Fig. 1에 표시된 바와 같이 병합과 선택의 과정은 적용할 불리안 연산자의 종류에 관계없이 사용될 기본 물체(primitive)가 모두 모여 있는 병합체(merged set)를 구하는 것으로 시작된다. 병합체란 구체적으로, 사용된 기본 물체의 위상정보, 또 그들 간의 교차결과로 생긴 선이나 점들, 그리고 이들 위상요소가 어느 기본물체의 영향으로 생겨났는지 그리고 각 기본물체에 대한 포함 관계가 어떠한지등을 나타내어 주는 정보를 완벽하게 포함하고 있는 또 하나의 비다양체 모델이다.

병합체 모델이 구해지면 이제 불리안 연산자가 적용되어 병합체 모델의 위상 요소들 중 남아야 할 것들과 없어야 할 것들을 선별해 내는 선택(selection)과정이 진행된다. 선택과정은 병합체의 각 위상 요소들에 자기 자신이 어떠한 과정을 거쳐 생겨났는지를 표시하는 역력표식(history record)을 참조하여 수행된다. 여기서 선택과정의 결과로 얻어지는 것은

병합체의 각 위상요소들에 대해 그것이 남아야 할지 혹은 아닌지 하는 플래그(flag)이며 병합체 모델이 즉각적으로 결과 모델로 변화되는 것은 아니다. 이러한 병합체는 다음과 같은 성질을 갖는다.

- 병합체는 기본물체의 적용 순서에 관계없이 위상적으로 동일하다. 즉 A, B, C의 기본물체로 이루어진 병합체의 경우 $A \rightarrow B \rightarrow C$ 의 순서로 병합시키든 $C \rightarrow B \rightarrow A$ 등의 다른 순서로 병합시키는 위상적으로 서로 다르지 않다.
- 병합체에서 특정 기본물체의 추출(extraction)이 가능하다. 즉 A, B, C의 기본물체로 이루어진 병합체에서 추후에 기본물체 C를 추출해 낼 경우 이는 처음부터 A, B의 기본물체만으로 만들어진 병합체와 위상적으로 동일하다.
- 병합체를 구성하는 기본물체중 하나만 변화된 경우 이를 선택적으로 반영할 수 있다. 즉 A, B, C의 기본물체로 이루어진 병합체에서 기본물체 B가 B'로 변화된다면 병합체를 처음부터 새로 구할 필요 없이 B를 추출해 낸 후 여기에 B'를 다시 병합시키면 된다.

이러한 사실들은 병합과 선택을 모두 거친 후에라도 혹시 불리안 연산자가 수정되었을 때, 병합체에는 최소한의 수정을 가한 후, 수정된 불리안 연산자에 의한 선택과정만 다시 수행함으로써 원하는 결과를 얻을 수 있다는 것을 의미한다. 여기서 불리안 연산자의 수정이란 기본물체에 대한 적용 불리안 연산자의 종류의 변경이나 순서의 변경, 적용 기본물체의 이동이나 변형등을 지칭하는데, 적용 불리안 연산자의 종류의 변경이나 순서의 변경의 경우에는 병합체에는 아무런 수정을 가할 필요 없이 선택과정만 다시 수행하면 되며, 적용 기본물체의 이동이나 변형의 경우에는 병합체로부터 변경 전의 기본물체를 추출한 후 변경된 기본물체를 다시 병합시켜 선택과정을 수행하면 된다. 이때 병합과정은 경계계산(boundary evaluation)을 위해 위상요소들의 교차계산 등 복잡한 계산으로 많은 시간을 소비하고, 선택과정은 플래그만 검사하여 거의 즉각적으로 수행되는 것을 감안하면 불리안 연산자의 수정은 매우 간편하고 빠르게 이루어질 수 있다는 것을 알 수 있다.

기존의 다양체 모델링에서 불리안 기능을 사용하는 경우 모델의 형상이 복잡해지면 불리안 작업 시킬 기본물체와 불리안 연산자의 순서를 결정하는 작업이 빈번히 틀리게 되며, 이런 경우 수행시켰던 불리안 작업을 단계적으로 취소시키고 새롭게 다시 작업하여야 한다. 이러한 과정은 많은 계산을 요하는

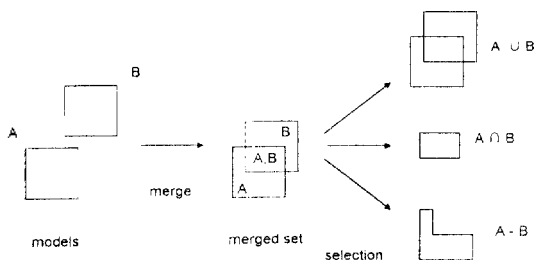


Fig. 1. Merge and selection process.

작업들을 배번 다시 해야 하기 때문에 모델링 효율을 저하시키는 주요 요인이 되었는데 위의 병합과 선택의 개념에 따른 불리안 연산은 이를 크게 개선할 수 있다. 비다양체의 불리안 조작에서 이러한 병합과 선택이 가능해진 이유는 보통 다양한 비다양체적인 특성을 가지게 되는 병합체를 그대로 수용할 수 있게 되었기 때문이다.

이러한 불리안 기법은 최근 대부분의 모델링 시스템이 지향하고 있는 경계표현(boundary representation)과 CSG(Constructive Solid Geometry) 트리의 복합표현(hybrid representation)을 데이터 구조 그 자체에 포함시킬 수 있으며, 기본물체의 적용순서에 관계없는 취소(undo) 기능을 제공할 수 있는 외에도, 비다양체가 가지는 표현 영역의 확장(솔리드는 물론 곡면모델 및 와이어프레임모델)이라는 장점과 맞물려, 설계모델이나 해석모델 또는 설계의도의 단일 자료구조에 의한 표현, 특징형상의 모델링에서 특징형상의 자료표현이나 특징형상간의 간섭문제 처리 등, 많은 분야에서의 활용이 기대되고 있다.

본 연구에서 개발한 불리안 기능은 이와 같은 병합과 선택의 개념에 바탕을 두고 있으며, 모델링 커널의 기반 기능 구축⁽¹⁾의 결과로 얻어진 간결한 자료구조를 사용하였고, 이 자료구조에 적합하도록 개발된 오일러 연산자⁽²⁾를 사용하여 불리안 연산자의 모듈화를 기하였다. Table 1은 본 연구에서 사용된 오일러 연산자의 사양을 보여 준다.

3. 병합(Merge) 알고리즘

3.1 병합 알고리즘의 개요

몇 개의 기본물체(primitive)로 이루어지는 병합체(merged set)를 만들어 가는 과정은 최초의 기본물체를 복사하여 이것을 하나의 기본물체로만 이루어진

Table 1. Euler operators

	Name	Description
Basic Euler Operators	MEV(KEV)	make(kill) edge, vertex
	MEC(KEC)	make(kill) edge, cycle
	MFKC(KFMC)	make(kill) face, kill(make) cycle
	MFR(KFR)	make(kill) face, region
	MVS(KVS)	make(kill) vertex, shell
	MVL(KVL)	make(kill) vertex, loop
Additional Euler Operators	SEMV(JEKV)	split(join) edge, make(kill) vertex
	MEF(KEF)	make(kill) edge, face
	KEML(MEKL)	kill(make) edge, make(kill) loop
	KEMS(MEKS)	kill(make) edge, make(kill) shell
Additional Topological Operators	MMR(KMR)	make(kill) model, region

병합체로 삼은 뒤, 두 번째 기본물체를 여기에 병합시키고 또 그 다음 기본물체를 병합시키고 마찬가지로 마지막 기본물체까지를 병합함으로써 이루어진다. 따라서 병합체가 구성되는 과정은 크게 두 가지 유형이 있는데, 그 첫번째는 하나의 기본물체로 최초의 병합체를 만들어 내는 작업이고 두번째는 하나 또는 그 이상의 기본물체로 구성되어 있는 병합체에 새로운 기본물체를 추가 병합시키는 경우이다. 전자의 경우는 비교적 간단히 수행될 수 있으며 후자의 경우가 일반적인 병합과정이다.

한편 일반적인 병합과정은 기존의 불리안 조작에서의 경계계산(boundary evaluation)과 유사한 과정을 거쳐 위상요소들의 변경 혹은 추가가 이루어지는데, 본 연구에서는 모든 위상 데이터의 접근 또는 수정을 오일러 연산자를 통해서만 가능하게 하였다. 이는 어떤 위상요소를 어떻게 바꿀 것인가 하는 것과 그것이 조작되어지는 과정을 명확히 설명할 수 있을 뿐만 아니라 매 조작 단위마다 항상 정당한 위상 형태를 유지할 수 있으며, 프로그램 개발자가 비다양체의 자료구조 그 자체로부터 분리되어 알고리즘이 단순화되고 모듈화되는 장점이 있다.

일단 병합체가 얻어지면 앞서 설명한 바와 같이 병합체 내의 위상요소들은 선택을 위한 정보로 연혁표식(history record)을 가지게 된다. 예를 들면 병합

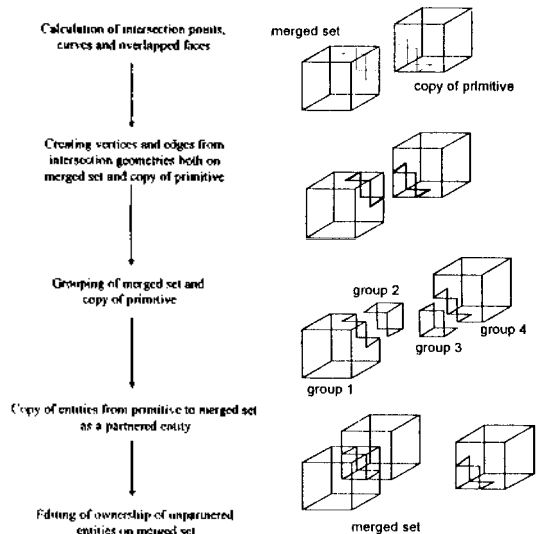


Fig. 2. Merge process.

체의 어느 영역은 기본물체의 어느 영역으로부터 생겨났다면, 병합체의 모서리는 기본물체의 어느 모

서리의 일부라든가 하는 것등이다.

Fig. 2는 본 연구에서 개발한 병합 알고리즘의 개요를 보여 주고 있다. 이 과정을 간단히 설명하면 다음과 같다.

① 병합체와 기본물체간의 교점, 교선 및 접친 면을 구함. (Intersection)

기본물체를 복사하여 원본은 놔두고 병합체와 복사체간의 교차계산을 수행한다. 복사체를 사용하는 이유는 병합과정 중 기본물체가 수정되어야 하는데 연혁표식이 의미를 가지려면 기본물체는 원형 그대로 보존되어 있어야 하기 때문이다. 따라서 병합과정의 진행 중에는 복사체에 대해 작업을 하며 모든 과정이 완료되면 변형된 복사체는 메모리에서 삭제된다.

② 교점 및 교선을 위상요소로 생성시키기

교차계산으로 구해진 교점과 교선을 병합체와 기본물체의 복사체에 각각 꼭지점과 모서리로 생성시켜 준다. 이때 생성된 위상요소들은 병합체와 복사체에 쌍으로 생기는데 이의 대응관계(partnership)도 저장하여 둔다. 이는 뒤의 ④단계에서 복사를 수행할 때 복사된 것이 어떤 것이고 아직 복사되지 않은 것이 어떤 것인지를 판단하는데 사용된다.

③ 병합체와 복사체의 위상요소들을 교점, 교선, 접친 면을 경계로 하여 그룹화

그룹화의 목적은 뒤의 ④단계의 복사과정에서 복사할 요소가 병합체의 어느 영역에 속해 있는지를 알아낼 때 계산 양을 줄여 주기 위함이다. 어떤 점이 어느 영역에 속해 있는가를 알아내는 소속영역 결정작업(region classification)은 교차계산을 포함한 상당량의 계산을 요하는데 이를 모든 위상요소에 대해 수행하는 것보다는 하나의 영역에 속할 그룹을 묶어 그룹에 대해 한번만 수행하면 매우 효율적이 된다.

④ 기본물체의 복사체로부터 병합체로의 위상요소 복사

병합체는 교차의 결과로 생긴 위상요소를 단계 ②의 과정으로 가지고 있으므로 여기서는 기본물체의 위상요소들중 병합체에는 아직 없는 위상요소들을 복사한다. 이 작업은 위에서 얻어진 그룹별로 수행하는데 기본물체쪽에 있는 그룹을 하나씩 선택하여 그 안의 꼭지점(vertex), 모서리(edge), 면(face)의 순서로 병합체에 복사 생성한다. 만들어질 때마다 대응관계를 계속 추가시키면서, 기본물체쪽의 위상요소중 대응관계가 없는 것을

계속 복사하면 빠짐없이 복사를 수행할 수 있다.

⑤ 병합체의 요소중 대응관계가 없는 위상요소들의 연혁표식(history record) 수정

위의 ①에서 ④까지의 단계를 거치면서 병합체쪽에 새로이 생성되거나 수정된 위상요소들은 그때마다 연혁표식이 적절히 수정되었다. 그러나 병합체쪽의 위상요소중 현재 병합되고 있는 기본물체에 의하여 변형이 되지 않는 것들(병합체의 위상요소중 대응관계가 없는 것들)은 연혁표식에 새로이 병합되고 있는 기본물체에 대한 내용이 더 첨가되어야 한다. 여기서는 이들이 기본물체의 어느 영역에 속하는지를 조사하여 이를 수행한다.

3.2 연혁표식(history record)을 저장하기 위한 자료구조

선택 작업을 위해서는 병합체의 위상요소가 연혁표식을 가지고 있어야 하는데, 연혁표식을 가져야 하는 위상요소의 종류는 영역(region), 면(face), 모서리(edge) 및 꼭지점(vertex)의 4가지이다. 이 4가지는 간결한 자료구조¹²⁾에서 모델에 속하는 공간과 그 경계에 해당하는 점 집합을 표현하는 셀 요소(cell entity)들로, 셀(shell), 루프(loop), 부분면(partial face), 부분모서리(partial edge), 부분꼭지점(partial vertex) 등과 같이 데이터 구조 내에서 셀요소들의 연결관계를 나타내 주는 존재와는 구별되며 불리언 연산자의 적용에 따른 직접적 선택의 대상이다.

한편 병합체에 속하는 위상요소가 생겨난 근원은 크게 두 가지로 나누어 볼 수 있다. 첫째로는 어느 기본물체(primitive)의 위상요소의 전부 혹은 잘라워진 일부를 그대로 상속 받는 경우(part_of)이고, 두 번째는 원래는 어느 기본물체에도 없었다가 교차의 결과로 생겨난 경우(intersection_from)이다. 연혁표식은 이 두 가지의 경우를 모두 저장하여야 한다. Fig. 3은 연혁표식의 내용을 저장하는 Ownership 자료구조이다. 여기서 part_of는 병합체의 위상요소가 각 기본물체에 대해 어느 부분에 속하는지를 나타내는 필드이다. 따라서 part_of 필드에 딸린 링크 필드의 수는 기본물체의 수와 같으며 각 기본물체에 대하여 병합체의 위상요소를 포함하는 기본물체의 꼭지점, 모서리, 면, 영역중 가장 차원이 낮은 위상요소를 기록하여야 한다. (꼭지점은 0차원, 모서리는 1차원, 면은 2차원 그리고 영역은 3차원이다.) 또 intersection_from은 교차결과로 생긴 위상요소의 경우에만 사용되는 필드이다. 즉 교차생성 요소를 만드는 원인이 되는 교차쌍들을 기록해 두는 곳이다.

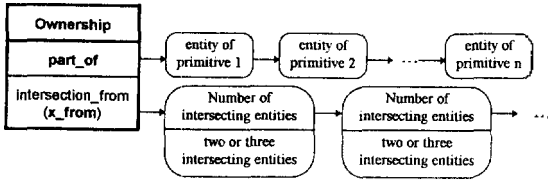


Fig. 3. Ownership data structure.

한편 교차생성의 경우에도 part_of 필드는 상속 생성의 경우와 동일한 방식으로 기록해 두는데 이렇게 하면 후에 선택(selection)과정에서 part_of 필드만 참조하면 되어 상속 생성의 경우와 알고리즘 상의 일관성이라는 이점이 있다.

3.3 교차계산(Intersection)

앞서 설명한 바와 같이 우선 기본물체를 복사한 복사체를 만드는 작업이 선행되어야 하는데, 만들어진 복사체의 모든 셀요소(cell entity)에는 연혁표식을 달아 둔다. 그 내용은 원래 대응하는 기본물체의 위상요소로부터의 상속(part_of)이다. 한편 하나의 기본물체로 최초의 병합체를 만드는 작업은 교차계산 등 더 이상의 조작 없이 복사체를 병합체로 삼으면 완료된다. 이제 교차계산을 수행하는데 결과로 얻어야 하는 것은 다음의 세 가지 정보이다.

- 교차점(intersection point)의 좌표 및 이때 교차시킨 위상요소의 쌍
- 교차선(intersection curve)의 방정식 및 이때 교차시킨 위상요소의 쌍
- 오버랩되는 면의 쌍(faces pair)들

그런데 여기서의 교차계산은 기존의 다양체의 교차계산과는 두 가지 점에서 차이가 있다. 첫 번째는 교차계산에 참여 시키는 위상요소로 면뿐만 아니라 셀 모서리(와이어 모서리), 셀 꼭지점(독립 꼭지점)도 포함시켜야 한다는 것이다. 그 이유는 다양체의 경우는 모델을 이루는 모든 경계위상요소가 면 및 면의 경계에 포함되어 있으므로, 면끼리의 교차계산이 모든 위상요소 사이의 계산 결과이지만, 비다양체의 경우 셀 모서리나 셀 꼭지점은 다른 위상요소의 경계가 아니므로 독립적으로 취급해야 하기 때문이다. 따라서 비다양체의 교차계산은 병합체와 복사체에서 면, 셀 모서리, 셀 꼭지점의 모든 가능한 조합에 대해 수행된다.

두 번째의 차이점은 교차계산의 결과를 다양체의 경우보다 풍부하게 찾아내어야 한다는 것이다. 예를 들면 면끼리의 교차나 면과 모서리의 교차결과가 한 점이 되어도 이를 찾아내어야 하며, 두개의 면이 부

분적으로 오버랩(overlap)되는 경우에는 오버랩되는 면의 쌍(faces pair)을 기록해 두어야 할 뿐만 아니라 오버랩 부분을 둘러 싸는 경계선들은 교차선으로 등록해 두어야 한다는 등이다. 또 모서리끼리 일부분이 오버랩 될 때는 오버랩 되는 부분은 교차선으로, 오버랩 되는 부분의 양 끝점은 교차점으로 등록해 놓는다.

한편 교차점이나 교차선과 함께 이들이 무엇의 교차결과로 나왔는지도 저장하는데, 이는 후에 어디에 속하는 위상요소로 만들어야 하는지를 알기 위함이다.

3.4 교차결과(intersection geometry)로부터 위상요소(topological entity) 생성하기

(1) 교차위상요소 생성의 개요

교차점과 교차선으로 꼭지점(vertex) 및 모서리(edge)를 병합체와 복사체에 각각 생성시키는 과정은 다음과 같다.

PROCEDURE Create_topology_from_Xgeom :

step 1 교차결과로부터 교차점을 하나씩 취해 다음을 수행한다. 더 이상 교차점이 없으면 step 2로 간다.

- ① 병합체와 기본물체의 복사체에 교차점을 꼭지점으로 생성한다. 이를 위해 Add_point_on_entity를 각각 한번씩 호출하는데, 이때 생성된 꼭지점과 base_entity가 반환된다. base_entity란 꼭지점이 실제로 놓여지는 꼭지점, 모서리 또는 면중 차원이 가장 작은 위상요소이다.
- ② 꼭지점의 base_entity가 양쪽 모두 꼭지점이고 새로운 꼭지점 쌍이 대응관계(partnership)에 등록되어 있으면 이 점은 이미 생성된 것이므로 다음번 교차점으로 넘어간다. (교차계산은 그 성질상 동일한 교차선이나 교차점을 중복하여 얻어내게 된다. 이 과정은 중복된 교차점 선택의 경우를 판단해 준다.)
- ③ 병합체쪽에 새로 생긴 꼭지점의 연혁표식인 ownership을 결정하여 저장한다. 이를 위해 Set_ownership_of_Xpoint_entity가 호출되는데, 이때 양쪽의 base_entity 정보를 필요로 한다.
- ④ 새로 생긴 꼭지점의 쌍을 대응관계(partnership)에 저장한다.

step 2 교차결과로부터 교차선을 하나씩 취해 다음을 수행한다. 더 이상 교차선이 없으면 step 3으로 간다.

- ① 병합체와 기본물체의 복사체에 교차선을 모서리로 생성한다. 이를 위해 Add_curve_on_entity를

각각 한번씩 호출하는데, 이때 만들어진 모서리와 base_entity가 반환된다. 만일 필요하다면 먼저 양 끝점을 꼭지점으로 생성한다. 모서리의 base_entity는 모서리 또는 면이 된다.

- ② 양끝 꼭지점에 대해 step 1에서와 동일하게 중복 생성 여부를 검사하고 새로 생긴 경우라면 ownership을 결정하고 partnership을 저장한다.
- ③ 모서리의 base_entity가 양쪽 모두 모서리이고 새로운 모서리 쌍이 대응관계(partnership)에 등록되어 있으면 이 교차선은 이미 생성된 것이므로 다음 번 교차선으로 넘어간다.
- ④ Set_ownership_of_Xcurve_entity를 호출하여 병합체쪽에 생성된 모서리의 ownership을 결정하여 저장한다. 이때 역시 양쪽의 base_entity 정보를 필요로 한다. 새로 생긴 모서리 쌍을 대응관계(partnership)에 저장한다.

step 3 오버랩 된 면의 쌍(overlap faces pair)의 정보를 수정한다. 또 병합체쪽 실제 오버랩 면의 ownership을 수정하고 실제 오버랩 면 쌍을 대응관계(partnership)에 저장한다.

여기서 step 3의 작업내용을 먼저 설명하면 다음과 같다. Fig. 4는 부분적으로 오버랩 되는 면을 보여준다. 처음에 교차체산에 의해서 F_1 과 F_2 가 오버랩 되고 있음을 알게 되면 오버랩 면의 정보는 (d)와 같이 저장된다. 그런데 경계선들이 모서리로 만들어지면 면들은 각각 분할되어 (c)와 같이 되고 이렇게 분할된 면들 중 실제로 겹치는 것이 무엇인지 알려면 다시 조사를 해 보아야 한다. 일단 면이 분할될 때에는 일단 후보가 되는 모든 면들을 (e)처럼 모두 저장해 놓았다가 교차선에 의한 모서리 생성이 모두 끝나면 실제의 오버랩 쌍을 (f)와 같이 찾는다. 이때 실제의 오버랩 쌍을 찾는 작업은 경계모서리들의 대응관계를 보면 쉽게 수행된다.

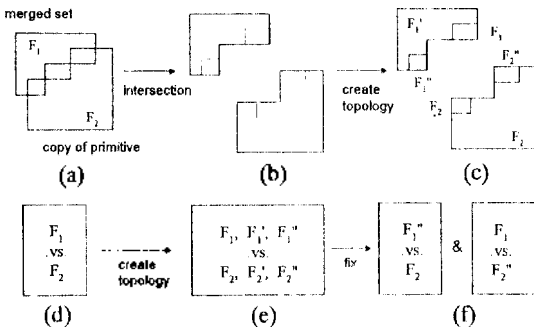


Fig. 4. Overlapped faces.

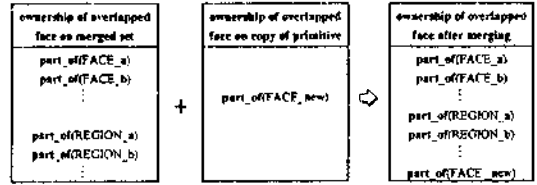


Fig. 5. Ownership of overlapped faces pair.

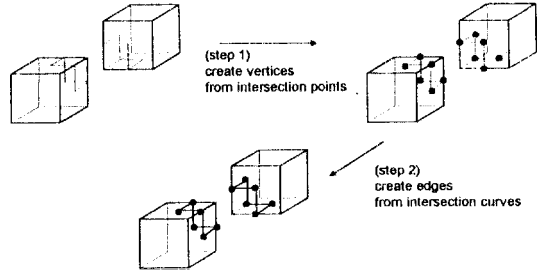


Fig. 6. Creation of topological entities from intersection

그리고 병합체쪽의 분할된 면 중 실제로 겹쳐지는 F_1 과 F_1'' 는 새로운 기본물체의 한 면(face)에 속하므로 그 내용이 ownership에 추가되어야 한다. 이 경우 F_1 과 F_1'' 는 병합체쪽의 면이므로 일반적으로 몇 개의 part_off(face of primitive) 및 part_off(region of primitive)를 연혁표식으로 가지고 있었을 것이고 F_2 는 하나의 part_off(face of new primitive)를 연혁표식으로 가지고 있으므로, F_1 과 F_1'' 의 수정된 ownership은 원래의 ownership에 F_2 의 ownership을 더해 주어 얻을 수 있다. 이 과정이 Fig. 5에 표시되어 있다.

이제 Create_topology_from_Xgeom 함수가 수행되고 나면 Fig. 2의 물체의 예는 Fig. 6과 같이 변한다.

(2) 교차점 및 교차선의 위상요소 생성방법

이제 Create_topology_from_Xgeom 안에 사용되었던 함수 중, Add_point_on_entity의 내용을 소개한다. 이는 입력으로 주어지는 교차 시킨 위상요소로부터 교차점이 실제로 놓여지는 위상요소인 base_entity를 알아내어 적절한 오일러 작업을 한다. Fig. 7은 모두 면을 교차 시켜 교차점이 얻어진 경우이지만 base_entity는 (a)의 경우 꼭지점이고, (b)의 경우는 모서리이며, (c)의 경우에는 그 면이 된다. Fig. 8은 교차점으로 꼭지점을 생성시키는 여러 경우와 그때 필요한 오일러 연산자를 보여 주는데 일고려준 내에 각 경우가 표시되어 있다.

PROCEDURE Add_point_on_entity :

input : 교차점의 좌표, 교차시킨 위상요소(꼭지점, 모서리 혹은 면)

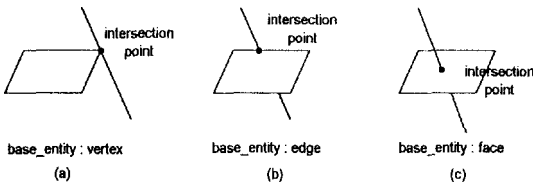


Fig. 7. Base entity of intersection point from intersecting a plane.

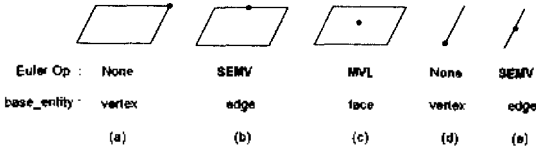


Fig. 8. Creation of a vertex from intersection point.

output : 생성된 꼭지점(new_vertex), 꼭지점이 놓여진 base_entity

IF 교차시킨 위상요소가 면이면

IF 교차점이 면의 경계 꼭지점 중 하나와 좌표 값이 일치하면 new_vertex ← 경계꼭지점, base_entity ← 경계꼭지점 (a)

ELSE IF 교차점이 면의 경계 모서리 위에 있으면 오일러 연산자 SEMV를 사용하여 new_vertex 생성, base_entity ← 경계 모서리 (b)

분할되어 새로 생긴 모서리에 ownership을 상속
ELSE IF 교차점이 면의 내부에 있으면 오일러 연산자 MVL을 사용하여 new_vertex 생성, base_entity ← 면 (c)

END IF

ELSE IF 교차시킨 위상요소가 모서리이면

IF 교차점이 모서리의 끝점 중 하나와 좌표 값이 일치하면 new_vertex ← 끝점, base_entity ← 끝점 (d)

ELSE IF 교차점이 모서리 위에 있으면 오일러 연산자 SEMV를 사용하여 new_vertex 생성, base_entity ← 모서리
 분할되어 새로 생긴 모서리에 ownership을 상속 (e)

END IF

ELSE IF 교차 시킨 위상요소가 꼭지점이면

new_vertex ← 꼭지점, base_entity ← 꼭지점

END IF

한편 위에서 (b)와 (d)와 같이 기존의 모서리가 분할(split)될 때에는 고려하여야 할 것이 한가지 더 있

다. 그것은 아직 처리되지 않은 교차선이나 교차점 중 그것을 얻기 위해 교차시킨 위상요소가 여기서 분할된 모서리인 경우이다. 이 경우에는 분할된 2개의 모서리 중 어느 하나만이 실제로 교차선이나 교차점을 포함하게 되는데, 둘 중에 어느 것인지 교차선의 중간점 혹은 교차점의 좌표 값의 포함 여부를 계산해 보아 재 조정해 주어야 한다. 그리고 *Create_topology_from_Xgeom* 안에 사용되었던 함수 중, *Add_curve_on_entity*는 *Add_point_on_entity*와 매우 유사하게 교차선으로부터 모서리를 생성한다.

(3) 교차 위상요소의 연혁표식의 결정

함수 *Create_topology_from_Xgeom*은 *Add_point_on_entity*를 사용하여 꼭지점을 생성하거나, *Add_curve_on_entity*를 사용하여 모서리를 만들 때마다 이의 연혁표식(ownership)을 결정하기 위해 *Set_ownership_of_Xpoint_entity*나 *Set_ownership_of_Xcurve_entity*를 각각 호출한다. 이들은 각기 병합체쪽의 base_entity와 이의 ownership, 그리고 복사체쪽의 base_entity와 이의 ownership을 참조하여 병합체쪽에 새로 만들어진 꼭지점이나 모서리의 ownership을 결정한다. 예를 들어 병합체쪽의 base_entity가 모서리이고, 복사체쪽의 base_entity가 면인 경우에 새로 만들어지는 꼭지점의 연혁표식을 정하는 방법이 Fig. 9에 표시되어 있다. 그림에서 병합체쪽의 base_entity의 ownership은 가장 일반적인 경우를 표시한 것이고, 복사체쪽의 base_entity의 ownership은 항상 지금 병합시키고 있는 기본물체에 대한 하나의 part_of 필드만을 가지게 되므로 이를 표시한 것이다. 또 VERTEX_a, EDGE_a, FACE_a, REG-

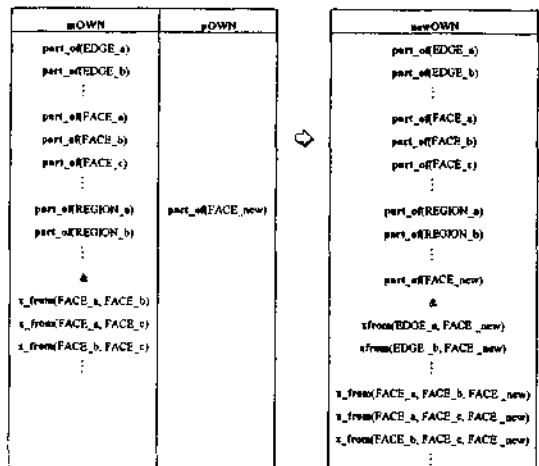


Fig. 9. Ownership of a vertex created by intersecting between edge of merged set and face of primitive.

ION_a 등은 기존의 병합체를 이루는 기본물체중 어느 하나의 꼭지점, 모서리, 면, 영역 등을 뜻하며, VERTEX_new, EDGE_new, FACE_new 등은 지금 병합되고 있는 기본물체의 어느 꼭지점, 모서리, 면 등을 지칭한다. 다른 경우로 생겨난 꼭지점이나 모서리의 경우도 유사하게 연혁표식을 정할 수 있다.

3.5 그룹화 작업(Grouping)

병합체와 기본물체의 교차로 인한 꼭지점, 모서리 등을 생성시키고 나면, 다음에 해야 할 작업은 그룹화이다. 그룹화란 병합체나 복사체의 위상요소들을 교점, 교선 또는 접친 면을 넘어서는 안될 경계로 삼았을 때, 서로 연결되어 있는 위상요소의 덩어리들을 찾아내는 작업이다. 이때 연결되어 있는 하나의 덩어리가 하나의 그룹이 된다. Fig. 2에 교차선을 경계로 하여 각각 2개의 그룹으로 분리된 병합체와 복사체의 예가 있다. 물론 그룹화란 이렇게 연결된 위상요소들을 알아내는 것이지, 실제로 위상 연결 관계를 자료구조에서 분리시키는 것은 아니다.

기본물체의 복사체에 대한 그룹화의 목적은 다음 단계에서 복사체로부터 병합체로의 위상요소를 복사할 때 계산의 효율성을 얻기 위함이다. 복사체 위상요소의 복사가 수행될 때에는 복사되는 위상요소가 병합체의 어떤 영역에 속하게 되는지 알아야 하는데(이는 오일러 연산자의 인수를 적절히 정하기 위해 필요하다), 이를 알기 위해서는 어떤 점이 병합체의 영역 중 어디에 속하는지를 알아내야 하는 점의 소속영역 결정작업(region classification)을 수행해야 한다. 이는 면(face)과 직선 사이의 교차계산(intersection)을 영역의 경계 면의 수만큼 실행해야 하므로 상당한 계산을 필요로 한다. 따라서 이러한 소속영역 결정작업을 복사되는 모든 위상요소에 대해 수행하는 것은 병합 알고리즘의 전체적인 효율에 영향을 주게 되는데, 그룹화를 통해 얻어진 각각의 그룹은 병합체의 동일한 영역으로 복사되는 성질을 가지므로, 그룹 내의 한 점만 검사하면 속하게 될 영역을 한꺼번에 결정할 수 있게 된다. Fig. 2에서는 그룹 3은 병합체의 내부 영역으로, 그룹 4는 병합체의 외부 영역으로 서로 다른 영역에 복사됨을 볼 수 있다.

또 병합체에 대한 그룹화의 목적은 후에 병합체의 대응요소가 없는 각 위상요소의 연혁표식을 수정할 때 효율성을 얻기 위함이다. 병합체의 위상요소 중 대응요소가 없는 것들은 병합의 마지막 단계에서 그 연혁표식(ownership)이 수정되어야 하는데 이때 그것이 원래 기본물체의 영역 중 어디로 속하는지를

알아야만 한다. 따라서 그룹별로 묶어 놓으면 역사 계산상의 이득을 볼 수 있다. Fig. 2에서는 그룹 1은 기본물체의 외부에, 그룹 2는 기본물체의 내부에 속함을 알 수 있다.

그룹화에서 그룹에 저장되어야 하는 위상요소는 면(face), 셸 모서리(shell edge) 및 셸 꼭지점(shell vertex)이다. 이러한 그룹화는 자료구조에 저장되어 있는 위상요소간의 연결 관계들을 이용하여, 교점 및 교선을 넘지 못하는 일종의 그래프 순회(traverse)를 통하여 수행될 수 있다.

3.6 기본물체의 복사체로부터 병합체로의 위상요소 복사

병합의 결과로 얻고자 하는 것은 모든 적용 기본물체의 정보를 포함하는 병합체이므로 기본물체의 요소 중 아직 병합체쪽에 없는 것들은 그 대응요소를 병합체에 복사해야 한다. 그런데 현 단계에서, 병합체 내에 새로운 기본물체의 영향으로 생긴 위상요소는 교차에 의해 생긴 모서리와 꼭지점, 그리고 접친 면이므로, 이외의 모든 위상요소들이 복사의 대상이 된다. 이들은 병합체쪽의 대응관계(partnership)가 없는 위상요소들로 이미 그룹화되어 있으므로, 복사는 이 그룹별로 이루어지게 된다.

복사의 진행과정은 개괄적으로 다음과 같다. 우선 처리되지 않은 그룹을 하나 선택한 후 그 그룹에 속하는 병합체쪽의 영역을 찾는다. 이제 그룹의 요소 중 셸 꼭지점들, 셸 모서리의 양끝 경계 점들, 그리고 면의 경계 루프에 속하는 꼭지점들 중 대응관계(partnership)가 없는 것들을 모두 모아 복사한다. 다음으로 셸 모서리들과 면의 경계 모서리들 중 대응관계(partnership)가 없는 것들을 모두 복사한다. 마지막으로 그룹 내의 면들을 복사한다. 이렇게 저 차원의 위상요소로부터 고차원의 위상요소로 계층적 순서를 거치는 이유는 만들려는 위상요소의 경계를 미리 만들어 놓음으로써 적용할 오일러 연산자의 경우의 수가 줄여지기 때문이다. 이는 다양체의 불리안에서는 취할 수 없었던 장점이기도 하다.

한편 모서리 또는 면을 생성시킬 때는 병합체쪽에 그 경계가 되는 꼭지점 또는 모서리들을 알아야 하는데 이는 복사체쪽의 경계요소를 찾아 이의 병합체쪽의 대응요소를 찾으면 된다. 또 위상요소가 하나씩 복사되어 생성될 때마다 그것의 연혁표식(ownership)이 정해져서 저장된다.

이러한 복사의 과정은 복사체의 그룹들 중 하나를 취해 이 그룹이 병합체쪽의 어느 영역으로 복사될

것인지를 계산한 뒤, 모든 꼭지점을 복사하는 *move_unpartnered_vertices*, 모든 모서리를 복사하는 *move_unpartnered_edges*, 그리고 모든 면을 복사하는 *move_unpartnered_faces*를 차례로 호출함으로써 이루어지고, 더 이상 취할 그룹이 없으면 끝나게 된다. 이 중 *move_unpartnered_edges* 함수의 내용을 소개하면 다음과 같다.

PROCEDURE *move_unpartnered_edges* :

input: 복사될 그룹, 그룹이 속할 병합체의 영역

step 1 그룹 내의 모든 셸 모서리를 찾아 리스트에 넣는다.

step 2 그룹 내의 면의 경계가 되는 모서리들 중 대응관계가 없는 것을 리스트에 넣는다.

step 3 리스트의 각 모서리에 대해 다음을 수행. 더 이상의 모서리가 없으면 종료.

- ① 병합체에서 모서리의 양끝이 될 꼭지점이 같은 셸(shell)에 속해 있으면, 오일러 연산자 MEC, 다른 셸(shell)에 속해 있으면, 오일러 연산자 MEKS를 사용하여 모서리(edge)를 만든다.
- ② 새로 생긴 모서리에 ownership을 정하여 저장한다.
- ③ 새로 생긴 모서리와의 대응관계를 만들어 저장한다.

한편 위의 과정에서 새로 생긴 모서리의 연혁표식을 정하는 방법은 Fig. 10에 표시된 바와 같다. 즉 모서리가 속한 병합체쪽 영역의 연혁표식은 일반적으로 병합체를 이루고 있는 기본물체들의 영역에 대한 몇 개의 part_of로 구성되어 있을 것인데, 여기에 지금 병합되고 있는 새로운 기본물체의 영역에 대한 것이 있으면 빼고, 복사되는 셸꼭지점의 연혁표식을 보태면 된다.

한편 Fig. 2에서 보였던 물체의 예에서 꼭지점과 모서리를 복사하고 나면 Fig. 11과 같이 된다. 다음으로 *move_unpartnered_faces*의 내용을 소개하면 다음과 같다.

PROCEDURE *move_unpartnered_faces* :

input: 복사될 그룹, 그룹이 속할 병합체의 영역

step 1 입력으로 들어온 그룹이 속한 병합체의 영역

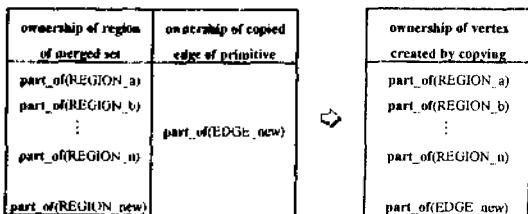


Fig. 10. Ownership of new edge copied to merged set.

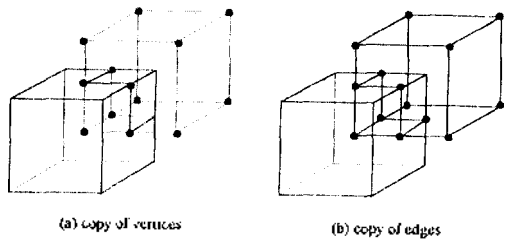


Fig. 11. Copy of vertices and edges.

을 후보 영역 리스트에 넣는다.

step 2 그룹에 저장되어 있는 각 면에 대해 다음을 수행. 더 이상의 면이 없으면 종료

- ① 후보 영역 리스트에 저장된 영역 중 실제로 면이 속하는 영역을 찾는다.
- ② 복사하려는 면이 하나 이상의 홀 루프(hole loop)를 가지면, 홀 루프 개수만큼의 브릿지 모서리(bridge edge)를 임의로 만든다.
- ③ 함수 *Shell_search_from_pseudo_face*를 호출하여 지금 만들 면에 의해 새로운 영역이 생기는 지를 검사한다.
- ④ 면을 만든다. 이때 적용할 오일러 연산자는 새로운 영역이 생기면 MFR, 그렇지 않으면 MFKC이다.
- ⑤ MFR이 사용되어 새로운 영역이 생겼으면, 이 영역을 후보 영역 리스트에 추가한다.
- ⑥ MFR이 사용되어 새로운 영역이 생겼으면, 분할된 두개의 영역의 ownership을 수정한다.
- ⑦ 새로 생긴 면에 ownership을 정하여 저장한다.
- ⑧ 새로 생긴 면과의 대응관계를 만들어 저장한다.
- ⑨ 임시 브릿지 모서리가 만들어졌다면 이를 지운다.

위와 같이 면을 복사하는 과정은 꼭지점이나 모서리의 경우에 비하여 다소 복잡하다. 우선 고려해야 될 것은 사용할 오일러 연산자의 선택이다. 이때 후보가 되는 연산자는 MFR과 MFKC인데 어느 것을 사용할 것인지를 구분 짓는 조건은, 지금의 면이 생성됨으로써 그것으로 둘러싸이는 폐쇄된 볼륨이 생기느냐, 그렇지 않느냐이다.

Fig. 12의 (a)는 두경 면을 덮음으로써 새로운 영역이 생기는 경우이고, (b)는 상자의 우측이 뚫려 있어 두경 면을 덮어도 영역의 생성이 없는 경우이다. 위의 알고리즘에서 *Shell_search_from_pseudo_face* 함수가 이것을 판단해 주는 기능을 한다. 이는 Fig. 12의 (c)에 표시된 바와 같이 만들어질 면의 경계가 될 모서리를 출발선으로 삼아, 한번은 만들 면의 앞쪽의 연결된 부분면(partial face)들로부터, 또 한번은

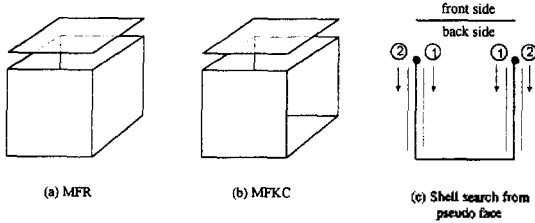


Fig. 12. Euler operators for face creation.

만들 면의 뒷쪽과 연결될 부분면 (partial face)들로부터 셸 탐색(shell search)을 수행하여, 그 결과의 동일 여부를 검사한다. 이때 물론 만들 면의 경계 모서리들은 타고 넘지 않도록 해야 한다. 그림의 (a)와 같은 경우는 양쪽에서 출발한 탐색이 서로 만날 수 없으므로 MFR을 적용해야 함을 알 수 있게 되고, (b)와 같은 경우는 양쪽에서의 탐색이 뚫린 부분으로 타고 넘어 그 결과가 같아져 MFKC를 적용해야 함을 알 수 있게 된다. 그리고 만일 MFR이 적용되어 영역의 분할로 새로운 영역이 생기면, 새로 생긴 영역은 분할되는 영역의 연혁표식을 그대로 상속 받게 된다.

그런데 만일 복사하려는 면이 홀 루프(hole loop)를 가지고 있는 경우에는 (step 2)에서와 같이 브릿지 모서리를 임시로 사용하게 된다. 그 이유는 이 면을 브릿지 모서리 없이 단번에 만드는 일은 상당히 복잡한 작업이므로 단 하나의 오일러 연산자로 지원할 수 없기 때문이다. 즉 면의 외곽 루프가 될 모서리들을 홀 루프가 될 모서리들과 서로 연결하는 브릿지 모서리를 일단 만들어 생성될 면이 외곽 루프만을 가지게 하면, MFR이나 MFKC로 면을 생성할 수 있게 된다. 브릿지 모서리를 만드는 오일러 연산자는 양끝 부분이 서로 연결되어 있지 않을 때는 MEKS, 연결되어 있을 때는 MEC이다. 이러한 브릿지 모서리의 위치는 시스템이 임의로 정하며, 병합체 모델에 있어서는 안되는 임시 위상요소이므로 해당 면의 복사가 끝날 때 오일러 연산자 KEML을 이용하여 지워 주게 된다.

위의 과정에서 새로 생기는 면의 연혁표식은 꼭지점이나 모서리를 복사할 때와 유사하게 정해져 할당되게 된다.

3.7 병합체의 요소중 대응관계가 없는 위상요소들의 연혁표식 수정

여기까지 작업이 이루어지면, 병합체는 비다양체 모델로 손색이 없는 완벽한 위상 연결 관계를 가지게 된다. 다만 추가로 조정되어야 하는 것은 병합체의 요소들 중 대응관계가 없는 것들의 연혁표식이다.

Fig. 2의 예를 다시 보면 원래 병합체쪽에 속해서 그룹 1과 그룹2에 저장된 위상요소들은 지금까지의 과정을 거치는 동안 위상조작의 대상이 된 적이 없기 때문에 연혁표식이 수정된 적이 없다. 그러나 병합이 이루어지면 그룹 1에 속한 위상요소들은 그 연혁표식에 자신이 새로운 기본물체에 대하여는 외부 영역에 속한다는 정보가 더해져야 하고, 그룹 2에 속한 위상요소들의 연혁표식에는 새로운 기본물체의 내부 영역에 포함된다는 정보가 추가되어야 한다. 따라서 병합체의 위상요소중 대응관계가 없는 위상요소들의 연혁표식 수정 방법은 다음과 같이 표시할 수 있다.

$$\text{새로운 ownership} \leftarrow \text{기존의 ownership} + \text{part_of} (\text{자신이 속한 기본물체의 영역})$$

한편 그룹이 기본물체의 어느 영역에 속하는가를 한번만 계산하면, 그 안에 저장된 각각의 위상요소들은 모두 같은 영역에 속하는 것을 알 수 있게 되므로, 그룹화에 의해 알고리즘의 효율성을 얻을 수 있는 것은 앞서 그룹화 과정에서 설명한 바와 같다.

4. 선택(Selection) 알고리즘

병합 과정을 거쳐 몇 개의 기본물체로 구성되어 있는 병합체가 얻어지면 이제 원하는 불리안 연산자를 정하여, 이에 따라 남아야 하는 위상요소와 그렇지 않은 위상요소를 골라내는 선택과정이 이어진다. 병합체의 위상요소 중 선택의 대상이 되는 위상요소의 종류는 셀 요소(cell entity), 즉 꼭지점, 모서리, 면 및 영역이다.

불리안 조작에서 사용하는 불리안 연산자는 합집합(+), 교집합(*) 및 차집합(-)의 새가지로 사용자로부터 주어지는 불리안 연산표현은 이들을 적당히 조합한 식이 된다. 예를 들면 다음과 같다.

$$((M1 + M2) * ((M3 - M4)) - M5) \tag{1}$$

불리안 연산표현이 주어지면 시스템은 주어진 불리안 식을 읽어서 이를 불리안 트리로 만든다. 이 과정은 식 (1)과 같이 중위표기(in-fix)로 주어지는 표현을 후위표기(post-fix)로 바꾼 뒤 이를 Fig. 13과 같은 불리안 트리로 바꾸는 방식으로 진행된다. 중위표기를 후위표기로 바꾸는 것은 임의의 수식을 인터프리트하는 기초과정으로, 자료구조론에 일반적인 알고리즘으로 소개되어 있고, 일단 후위표기 표현이 얻어지면 불리안 트리는 쉽게 얻어진다.

이렇게 얻어진 불리안 트리는, 단말노드(leaf node)

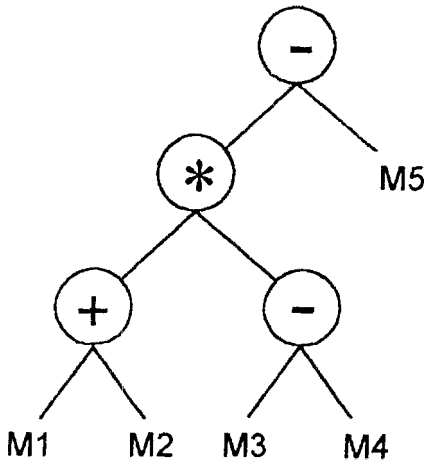


Fig. 13. Boolean tree.

의 내용은 항상 기본물체(primitive)이고, 이외의 중간 노드(non-leaf node)들은 불리안 연산자인 이진트리(binary tree)라는 성질을 가진다. 불리안 트리에 의한 선택작업은, 병합체의 모든 꼭지점, 모서리, 면 및 영역을 일단 **dead**로 표시한 뒤 불리안 트리의 루트 노드(root node)에 대해 함수 *Get_alive_entities_of_node*를 호출하여, 여기서 일어난 위상요소들에 **alive**로 표시함으로써 이루어진다.

이때 함수 *Get_alive_entities_of_node*의 기능은 불리안 트리의 임의의 노드가 주어졌을 때, 그에 따라 남아야 하는(**alive**로 되는) 위상요소들을 찾는 것이다. 이를 위해서는 주어진 노드로부터 단말노드(leaf node)까지의 노드들을 검색하고, 불리안 계산을 해야 하는데, *Get_alive_entities_of_node* 함수는 자기 자신을 재귀 호출하여 이를 수행한다. 그 내용은 다음과 같다.

PROCEDURE *Get_alive_entities_of_node* :

input : 병합체, 불리안 트리의 한 노드
output : **alive**로 찾아진 위상요소들의 리스트

IF 주어진 노드의 내용이 기본물체이면
 병합체의 모든 셀 요소에 대해 연혁표식으로 가지고 있는 **part_of**의 내용이 기본물체의 위상요소를 가리킨 것이 있으면 찾아진 위상요소 리스트에 넣는다.

ELSE(주어진 노드의 내용이 불리안 연산자이면)

① 좌측 자식노드(son node)에 대해 *Get_alive_entities_of_node*를 재귀 호출하여 결과 **left_alives**를 얻는다.

- ② 우측 자식노드(son node)에 대해 *Get_alive_entities_of_node*를 재귀 호출하여 결과 **right_alives**를 얻는다.
- ③ 불리안 연산자의 종류에 따라 **left_alives** 리스트와 **right_alives**간의 불리안 연산을 수행한다.
- ④ 불리안 연산자의 종류가 차집합(-)인 경우에는 **Closure**연산을 수행한다.

END IF

위의 ③에서 리스트간의 불리안 연산이 있는데, 두개의 리스트간의 합집합은 적어도 한쪽에 있는 요소를 찾는 것이고, 교집합은 양쪽에 모두 있는 요소를 찾는 것이며, 차집합은 앞의 것에는 있고 뒤의 것에는 없는 요소를 찾는 것을 말한다.

한편 차집합을 구한 경우에는 Fig. 14에서처럼 선택된 위상요소의 경계를 이루는 요소가 선택에서 제외될 수 있다. 그런데 이러한 경우는 모델에 속하는 어떤 위상요소의 경계를 이루는 위상요소들도 모델에 속해야 한다는 비다양체 모델의 정의에 위배된다. 따라서 위의 과정 중 ④에서는 **Closure**연산을 수행한다. **Closure**연산이란 현재 리스트에 선택되어 있는 위상요소의 경계가 되는 요소 중 리스트에서 빠진 것이 있으면 보충해 넣는 작업을 말한다.

5. 적용 예

본 연구에서 개발한 모델링 커널은 Silicon Graphics의 Indigo2 XZ workstation에서 C++언어로 프로그램되었다. 그래픽 환경은 X-Window에 OpenGL을 이용하여 구성되었으며, 곡선 및 곡면은 B-spline으로 표현된다. 또 개발환경의 안정성을 위해 곡면간의 교차계산만은 상용 모델링 커널인 ACIS의 Spline Interface의 기능을 사용하였으나 향후 독자 개발 프로그램으로 대체될 예정이다.

Fig. 15의 (a)는 8개의 기본물체로 이루어진 병합체를 보여 준다. 이 모델을 만들려면 우선 비어있는

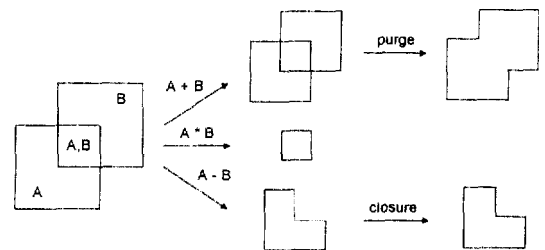


Fig. 14. Selection process.

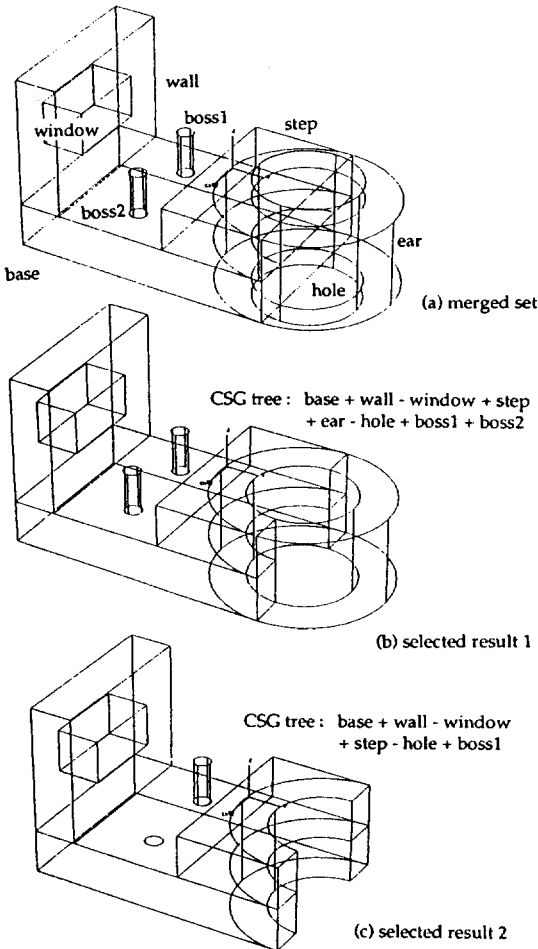


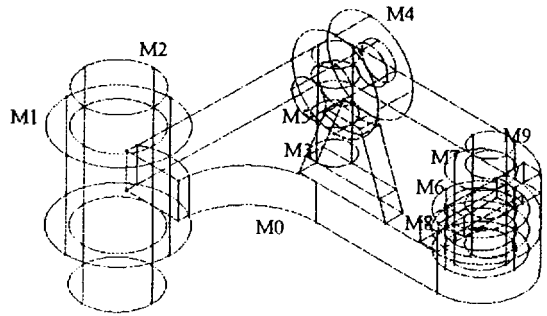
Fig. 15. Case study 1.

새로운 병합체를 생성시킨 뒤, 각 기본물체에 대해 병합명령을 내린다. 여기에 사용된 기본물체는 모두 육면체 또는 원기둥인데 이들은 이전에 만들어져서 적절히 위치되어 있어야 한다. 선택을 위해 주어진 불리안 연산자는 (b)와 (c)에 각각 나타나 있는데 이는 사용자가 키보드를 통해 입력한다.

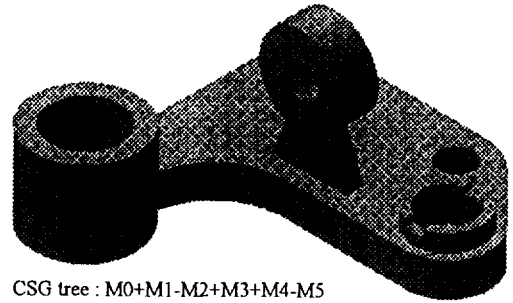
이 모델은 기본물체 하나하나가 특징형상(form feature)를 표현하고 있는데, 불리안 연산식을 바꿔가며 선택과정을 거쳐 각 특징형상을 선택적으로 가감할 수 있음을 보여주고 있다. Fig. 16은 이와 유사한 또 다른 병합체의 예를 보여 주고 있다.

6. 결 론

본 연구에서는 간결한 비다양체 자료구조를 바탕으로 한 불리안 기능을 성공적으로 개발하였다. 이로써



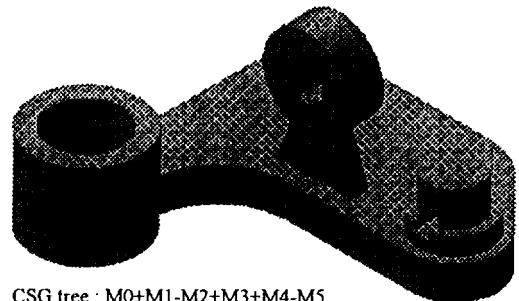
(a) merged set



CSG tree : M0+M1-M2+M3+M4-M5

+M6-M7-M8-M9-M10

(b) selected result 1



CSG tree : M0+M1-M2+M3+M4-M5

+M6+M7-M8

(c) selected result 2

Fig. 16. Case study 2.

비다양체 기반의 모델링 커널에 사용자에게 편리하고도 강력한 모델링 기능을 제공하게 되었고 앞서 개발된 비다양체 모델 자료구조의 정당성을 실증적으로 입증하게 되었다. 또한 이 불리안 기법은 병합과 선택의 개념을 사용하여 기본물체의 취소 또는 편집, 불리안 연산식의 수정등의 경우에 보다 편리한 사용 환경을 제공할 수 있을 뿐만 아니라, 경계표현(B-rep)과 CSG 트리표현의 복합표현을 단일 자료구조 안에 포함시키게 되었다. 이와 같은 병합체의 개념은 동일한 물체를

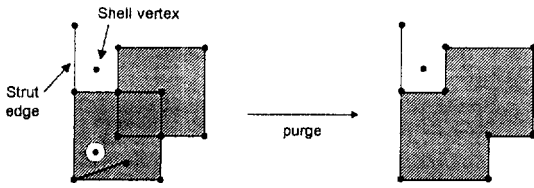


Fig. 17. Purge operator.

관점에 따라 다르게 표현하는 모델들의 통합(예를 들면 설계 모델과 이의 해석 모델의 통합 모델), 특징형상 기반의 모델링 시스템에서 모델의 표현과 조작등 다양한 적용 분야에서 활용이 크게 기대된다.

한편 물리안 조작이 사용자 수준으로 제공되려면 병합 연산자와 선택 연산자 외에도 다음과 같은 연산자들이 더 필요하며, 추후 개발될 예정이다.

- 추출(extraction) 연산자: 병합체를 구성하고 있는 기본물체중 하나를 빼내는 기능이다. 해당 기본물체의 영향으로 생겨났던 위상요소들을 없애므로 병합체는 애당초 이를 병합시키지 않은 것과 동일하게 된다.
- 플러쉬(flush) 연산자: 선택을 통해 얻어진 속성이 dead인 병합체의 위상요소들을 자료구조에서 제거하여 최종 결과 모델을 얻어내는 기능이다. 이 연산자의 작용은 비가역적이며 따라서 모델링과정의 마지막에 모델링이 올바르게 되었음이 검증되면 사용된다.
- 퍼지(purge) 연산자: 플러쉬 연산자를 적용하여 얻어진 모델도 일반적으로는 여전히 비다양체 모델이다. 즉 볼륨 타입의 영역(region)을 가로막는 칸막이 면(screen face)이나, 볼륨 타입의 영역 내에 존재하는 라미나 면(lamina face) 또는 셸꼭지점(shell vertex), 면의 내부에 빠져져 나와 있는 스트러트 보서리(strut edge) 등을 포함할 수 있다. 모델이라고 하는 것이 궁극적으로 공간상의 특정 점 집합을 표현하고자 하는 것이라 할 때, 이들 위상요소들은 모델 내의 일단의 점 집합들을 중복하여 표현하고 있다. 퍼지(purge) 연산자는 이와 같은 위상요소들을 제거해 주는 것으로 Fig. 17은 이를 도식적으로 보여 준다.

참고문헌

1. Weiler, K.J., "Topological Structures for Geometric Modeling," PhD Thesis, RPI, 1986.
2. 한영현, 이진우, 이상현, 김성환, 김영진, 배석훈, 안

재홍, 이경진, "비다양체를 지원하는 CAD 시스템 커널 개발을 위한 기반 구축," 산업공학, 제 8권, 제 3호, pp. 141-153, 1995.

3. Choi, Y., "Vertex-Based Boundary Representation of Non-Manifold Geometric models," PhD Thesis, Carnegie Mellon University, 1989.
4. Masuda, H., Shimada, K., Numao, M. and Kawabe, S., "A Mathematical Theory and Applications of Non-Manifold Geometric Modeling," International GI-IFIP Symposium 89, 1989.
5. Luo, Y. and Gabor, L., "Generalized Euler Operators for Non-Manifold Boundary Solid Modeling," *Geometric Modelling Studies*, 1990.
6. Gursoz, E. L., Choi, Y. and Prinz, F. B., "Boolean set operations on non-manifold boundary representation objects," *Computer Aided Design*, Vol. 23, No. 1, 1991.
7. Crocker, G. A. and Reinke, W. F., "An Editable Non-manifold Boundary Representation," *IEEE Computer Graphics & Applications*, March, 1991.
8. Masuda, H., "Form-feature representation based on non-manifold geometric modeling," *Proceedings of MICAD'92*, 1992.
9. 이상현, "사출 성형 제품의 실체를 및 해석의 통합 환경을 제공하기 위한 특징형상 기반 비다양체 모델링 시스템의 개발," 박사학위 논문, 서울대학교 대학원, 1993년 8월



김 성 환

1986년 서울대 기계설계학과 학사
 1988년 서울대 기계설계학과 석사
 1994년 서울대 기계설계학과 박사
 현 재 대우공업연구원 교수
 관심분야: Solid modeling, Assembly modeling

이 진 우

제 1 권 제 1 호 참조



김 영 진

1986년 서울대 기계설계학과 학사
 1988년 서울대 기계설계학과 석사
 1995년 서울대 기계설계학과 박사
 현 재 Post-Doc. Univ. of Illinois at Urbana-Champaign
 관심분야: Solid modeling, Computer aided inspection