프로그래밍과 수학

특정 도메인의 문제를 해결해 나가는 것이 필요하다. 하나씩 해결해 나간다.

# Full 3D Collision Detection and Navigation

구현의 관점이 아닌 알고리즘의 개발로 접근한다.

# WPF 프로그래밍

예쁜 UI를 WPF로. CEF?

# 명제 프로그래밍

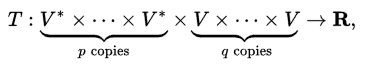
수학은 명제로 구성된다. 프로그래밍은 상태 변화로 이루어진다. 프로그램도 명제로 구성할 수는 없는가? Assertion을 먼저 쓰고 이를 채우는 방식의 프로그래밍이 있다고 들었다.

# 일반적인 사고 과정

여러 가지 방향에서 한번에 진행된다. 잘 정의된 개념과 성질을 따라 가기도 하고 갑작스럽게 한번에 나타나기도 한다.

개념과 개념의 실체화 도구로서의 정의가 있다. 그래프는 연결된 어떤 것들이란 관점에서 Set을 도구로 Vertex와 Edge 집합으로 나뉘고 Edge는 Vertex의 쌍 (순서쌍)으로 정의된다. 개념이 실체로서 타잎을 갖게 되고 각 타잎들은 적절한 연산을 갖게 된다.

텐서는 멀티리니어 함수로 vector space 와 vector space의 dual space에서 정의 된다.



수학에서는 주로 집합, 카테고리를 도구로 여러 가지 타잎과 연산들이 있다. 이들 타잎과 연산을 구체적으로 이해하는 것도 중요하지만 개념으로, 말로, 아이디어로, 직관으로 체득해야 사용이 쉬워지고 즐거워진다.

이는 프로그래밍에서도 마찬가지이며 각 영역별로 분화가 심하기 때문에 해당 영역에서 중요 도구와 타잎, 연산들을 설계와 코드로 확보해 두어야 한다.

# 구현 과정

개념, 정의, 자료 구조.

자료 구조. 정적 / 동적.

Adjacency Matrix

N(v) \* N(v) matrix

항상 trade off가 발생. 메모리 보다는 연산의 동작 특성 때문에 발생.

Density =~ |E|/|V|