Coq을 이용한 이분 그래프의 표현과 증명

배성경
KAIST

연구 목적

Coq을 이용하여…

- (이분)그래프를 표현하는 것
- 그래프 이론의 몇몇 정리들을 증명하는 것

동기

동기

- 수학 증명들을 자동으로 하게 만들 수 있겠다.

동기

- 수학 증명들을 자동으로 하게 만들 수 있겠다.
- 그런 프로그램 만들어 볼까?

동기

- 수학 증명들을 자동으로 하게 만들 수 있겠다.
- 그런 프로그램 만들어 볼까?
- Coq이라는 게 있네!

동기

- 수학 증명들을 자동으로 하게 만들 수 있겠다.
- 그런 프로그램 만들어 볼까?
- Coq이라는 게 있네!
- 공부해 볼래요.

동기

- 수학 증명들을 자동으로 하게 만들 수 있겠다.
- 그런 프로그램 만들어 볼까?
- Coq이라는 게 있네!
- 공부해 볼래요.
- 뭐 하나 직접 증명 해보면서 공부하는 게…

동기

- 수학 증명들을 자동으로 하게 만들 수 있겠다.
- 그런 프로그램 만들어 볼까?
- Coq이라는 게 있네!
- 공부해 볼래요.
- 뭐 하나 직접 증명 해보면서 공부하는 게…
- 해석학? 대수학? … 그래프!

동기

- 수학 증명들을 자동으로 하게 만들 수 있겠다.
- 그런 프로그램 만들어 볼까?
- Coq이라는 게 있네!
- 공부해 볼래요.
- 뭐 하나 직접 증명 해보면서 공부하는 게…
- 해석학? 대수학? … 그래프!
- 그래프 이론의 König's theorem을 증명해 봐야지.

왜 그래프 이론인가?

- 수학의 한 분야이지만, 전산학에서도 활발하게 사용되는 이론

왜 그래프 이론인가?

- 수학의 한 분야이지만, 전산학에서도 활발하게 사용되는 이론
- Curry-Howard's isomorphism에 의해 수학과 전산의 언어를 통일시킬 수 있다.

이전 연구 결과

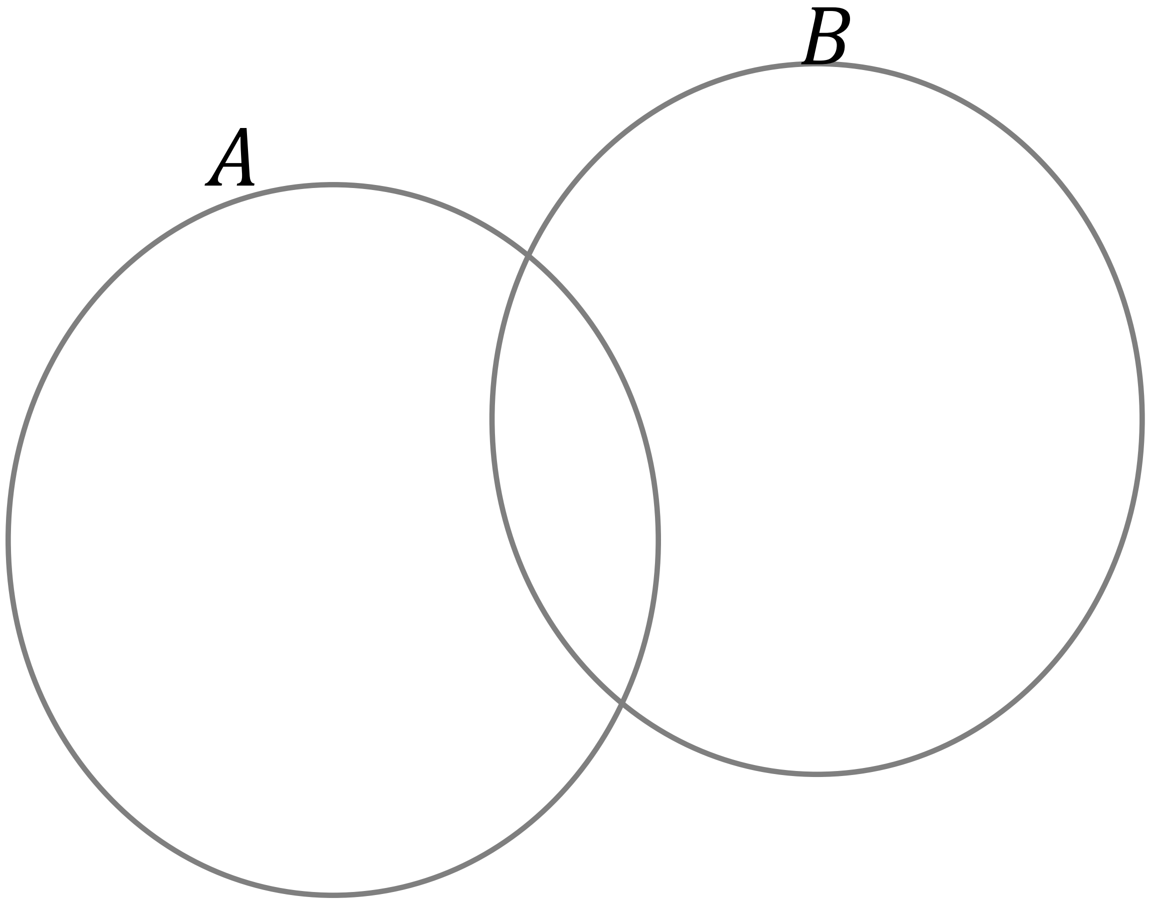
GraphBasics

- Jean Duprat, 2001
- “Standard library”가 아닌

“User’s contribution”

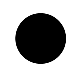
GraphBasics로 정의할 수 있는 것

- 집합



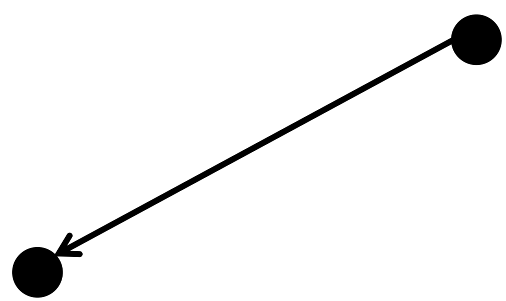
GraphBasics로 정의할 수 있는 것

- 정점



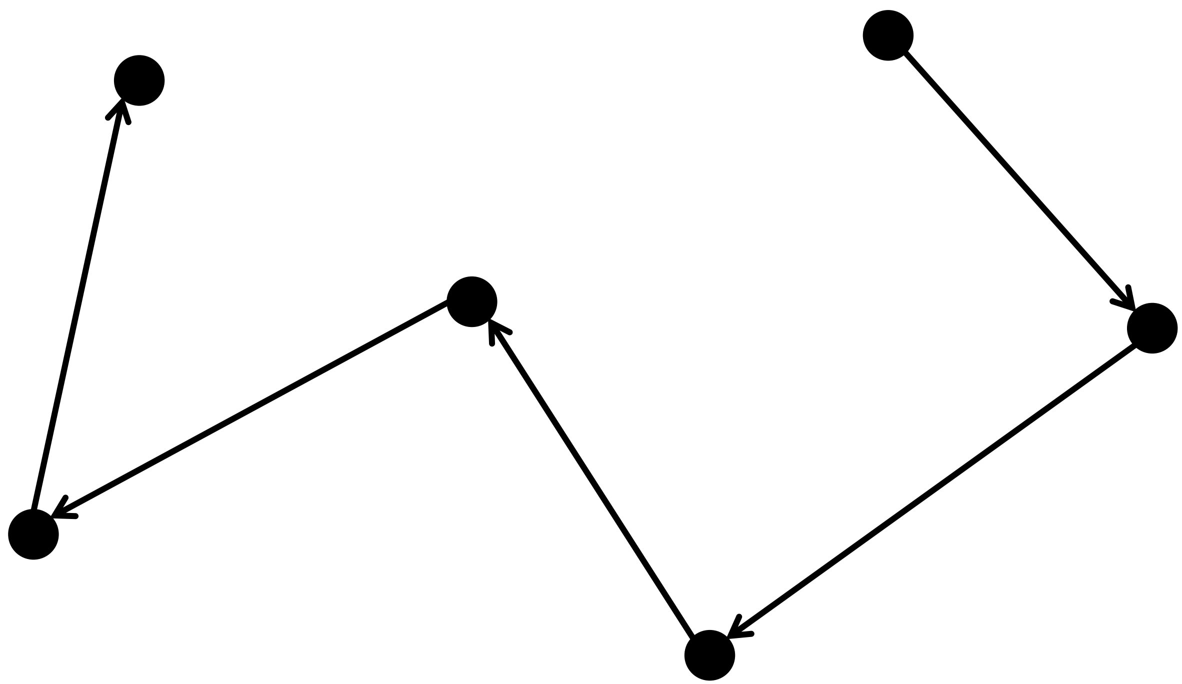
GraphBasics로 정의할 수 있는 것

- 간선



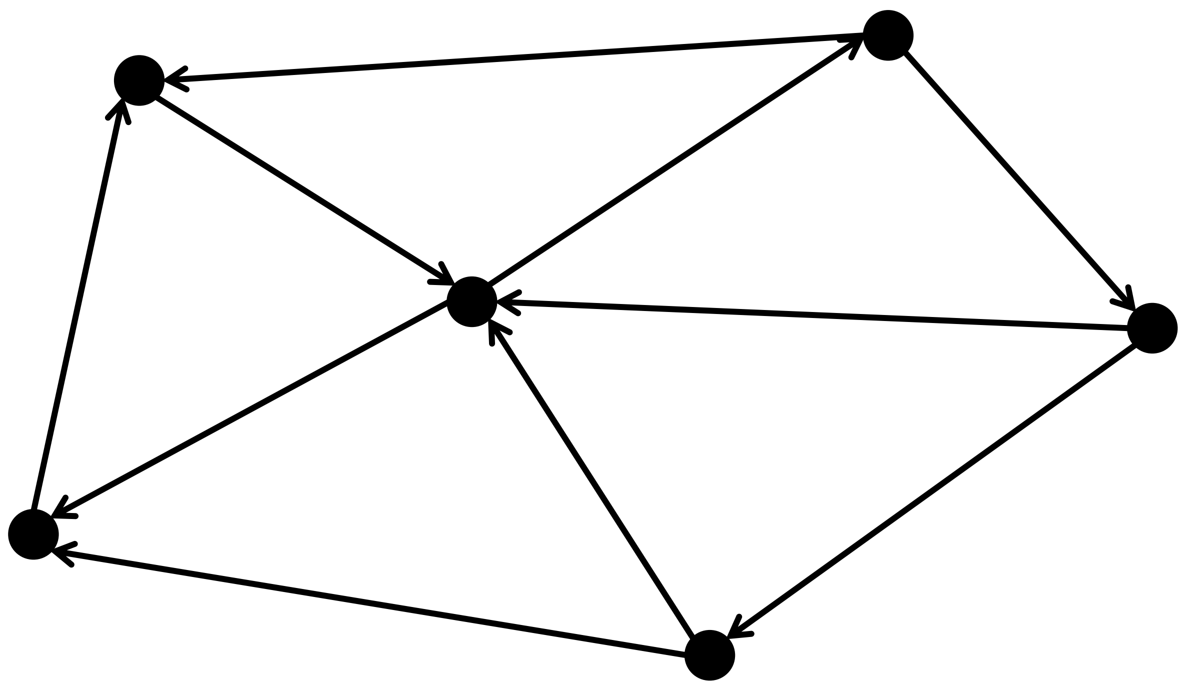
GraphBasics로 정의할 수 있는 것

- 경로



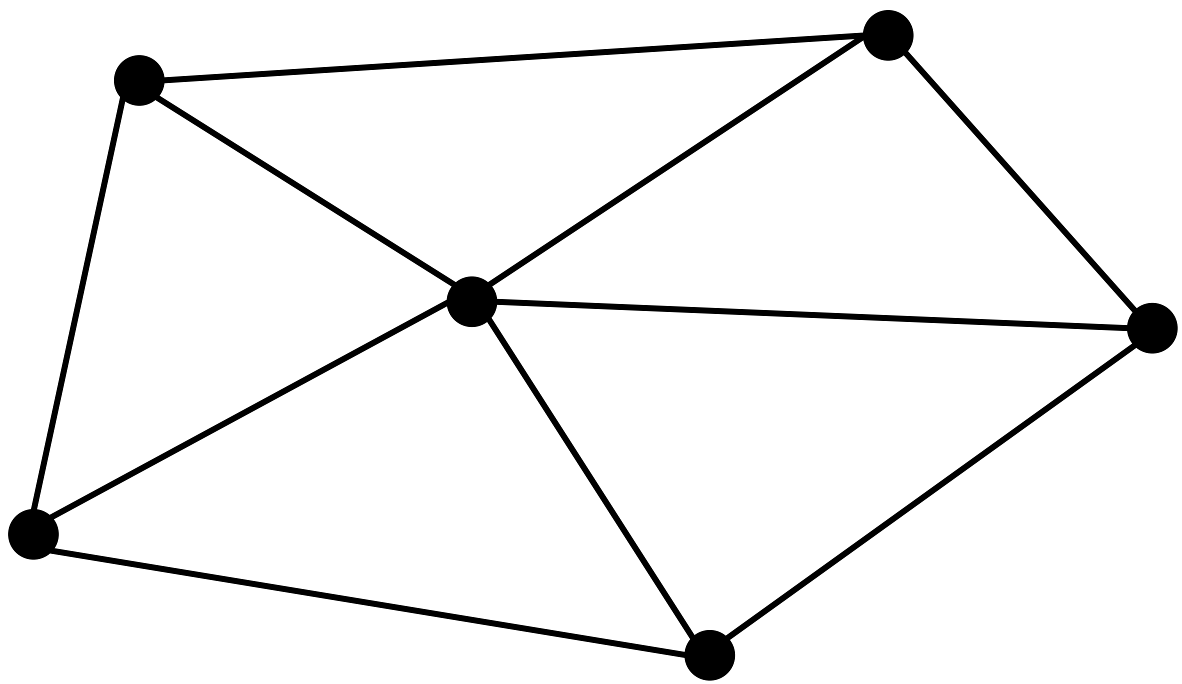
GraphBasics로 정의할 수 있는 것

- 방향그래프



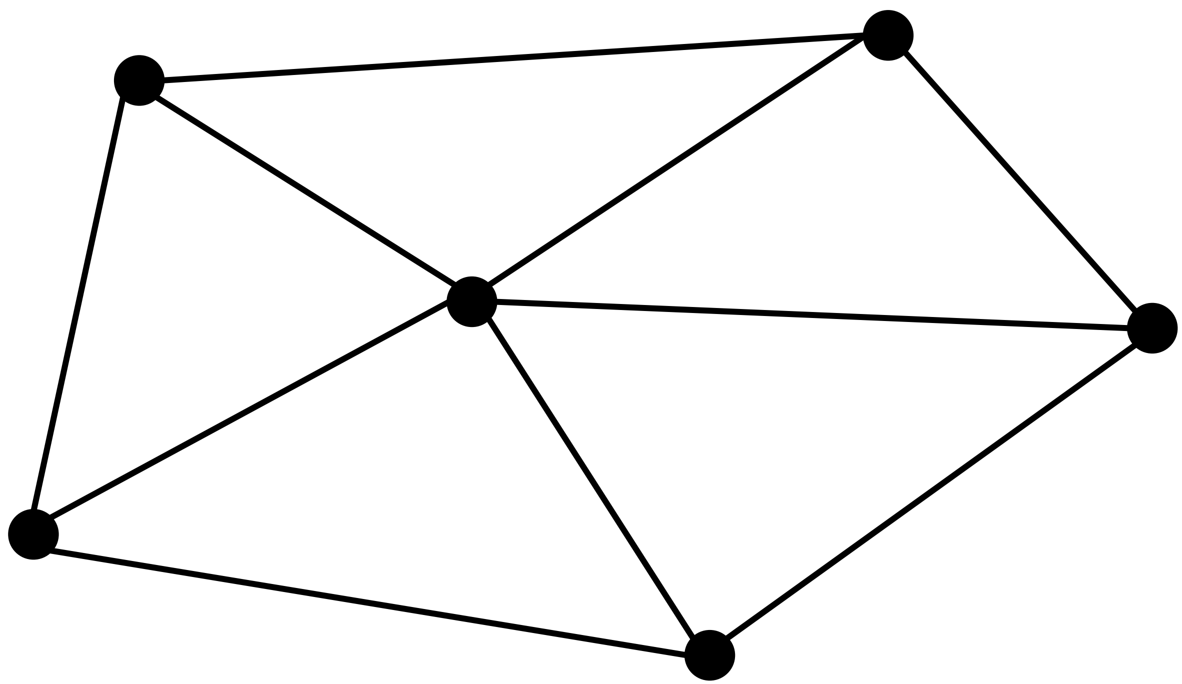
GraphBasics로 정의할 수 있는 것

- 무방향그래프



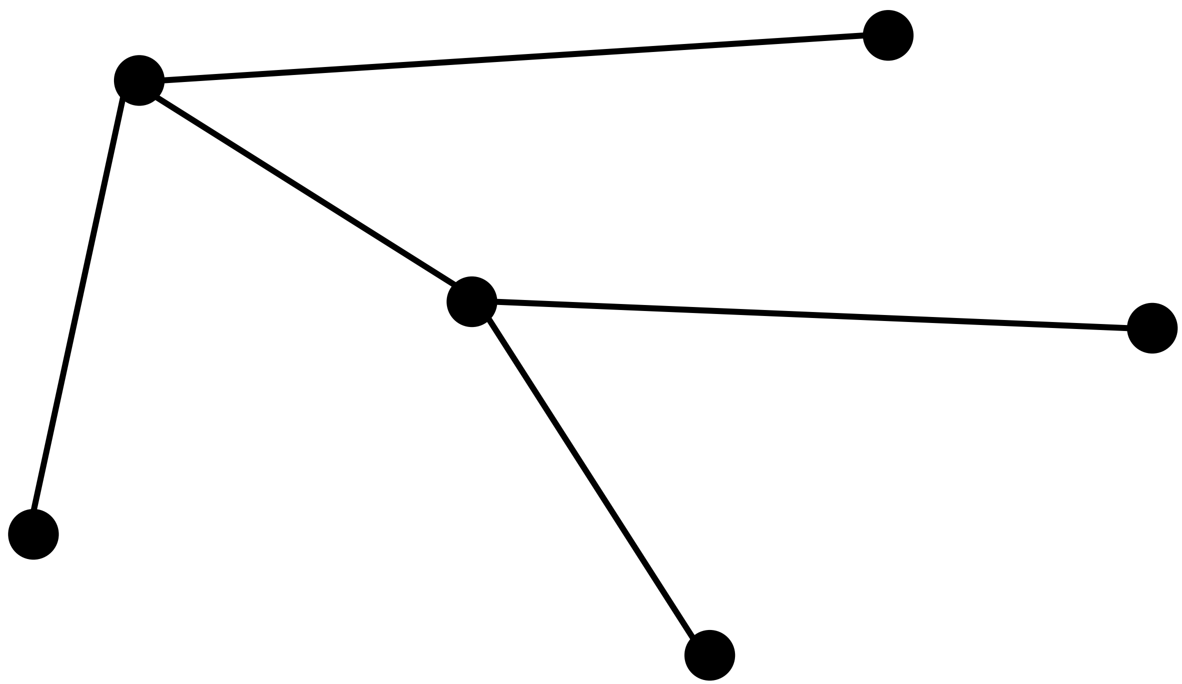
GraphBasics로 정의할 수 있는 것

- 연결그래프



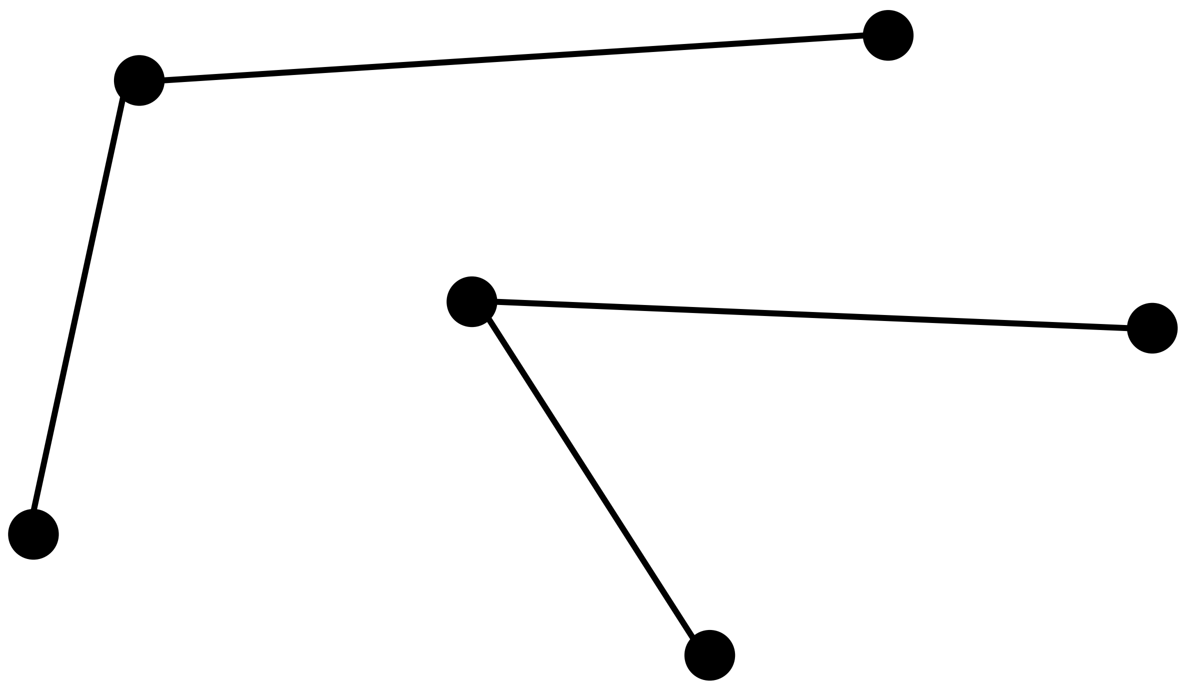
GraphBasics로 정의할 수 있는 것

- 트리



GraphBasics로 정의할 수 있는 것

- 비순환그래프



GraphBasics로 정의할 수 있는 것

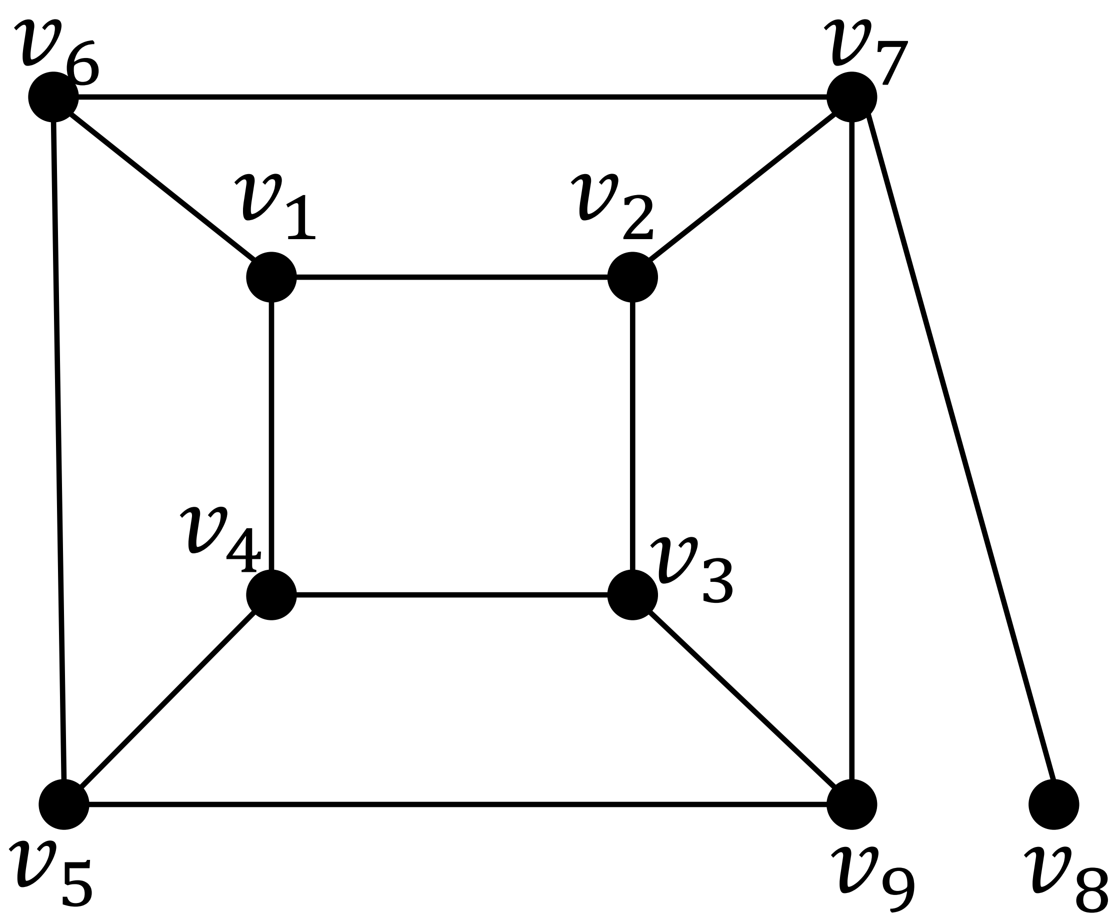
- 이분그래프는 없다.

이분 그래프란?

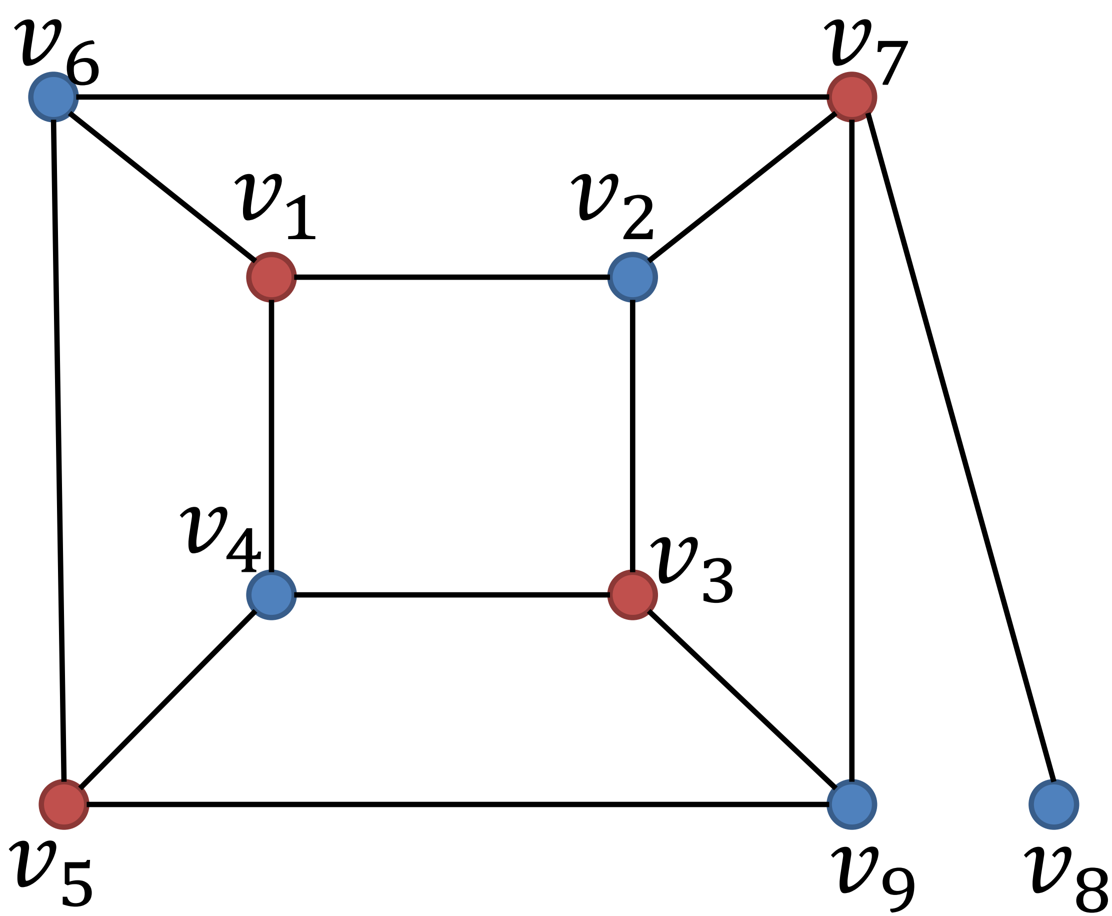
- 수학적 정의

A graph G is called *bipartite* if it is possible to partition the vertex set of G into two subsets, say V_1 and V_2, so that every edge of G joins a vertex of V_1 with a vertex of V_2, and no vertex joins another vertex of its own set.

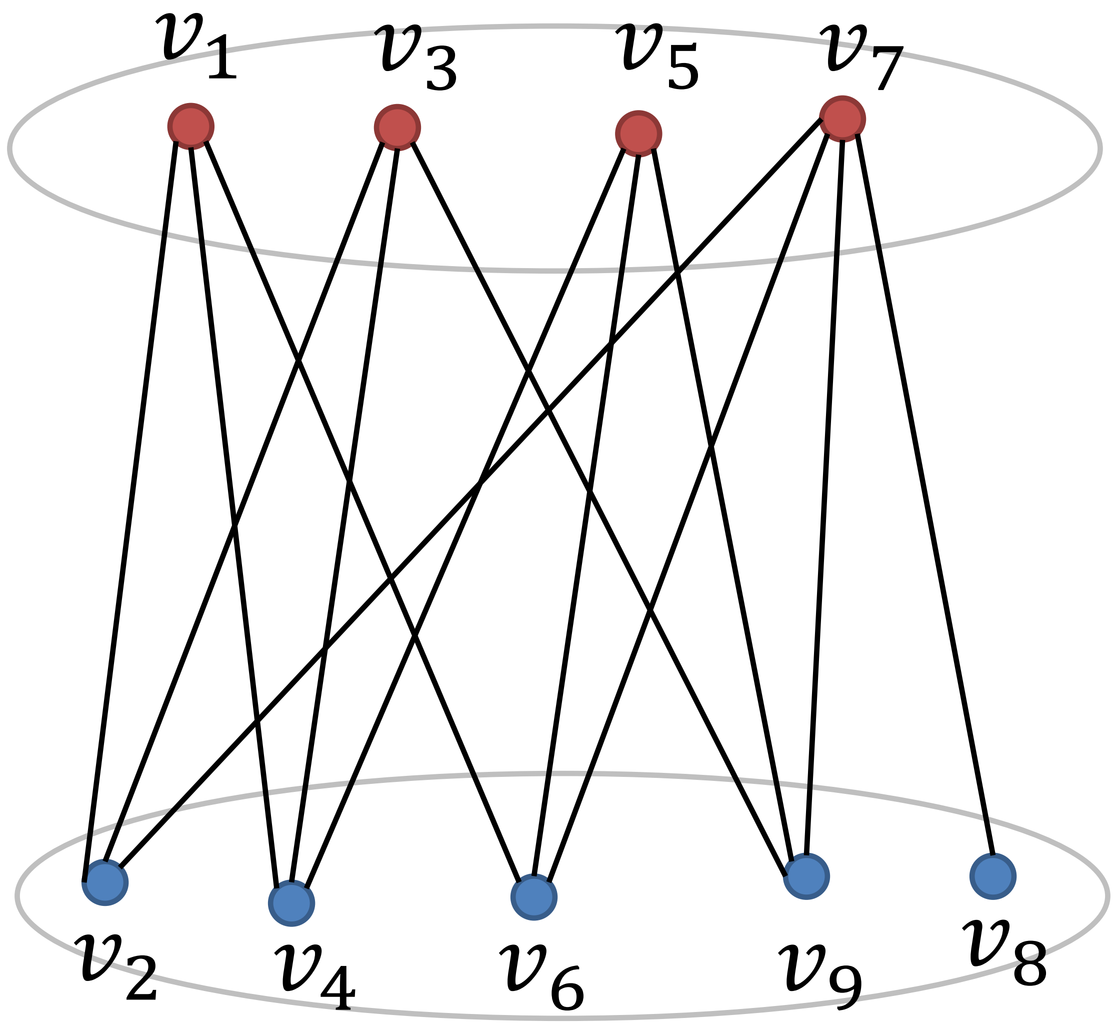
이분 그래프란?



이분 그래프란?

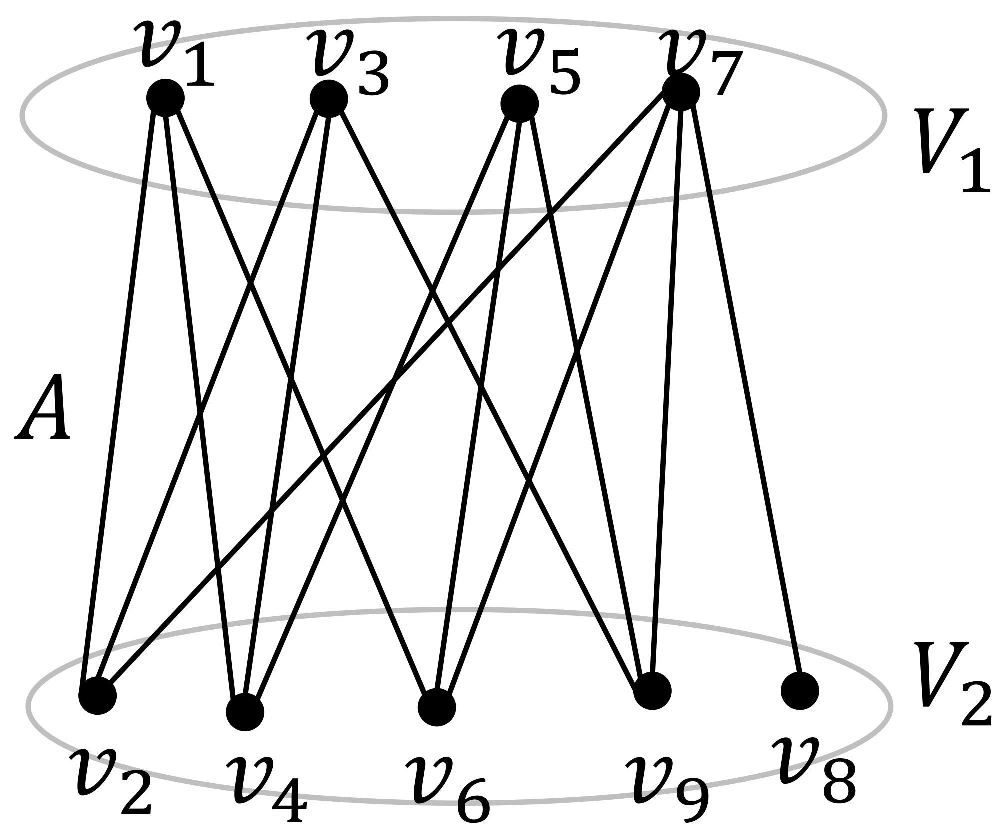


이분 그래프란?



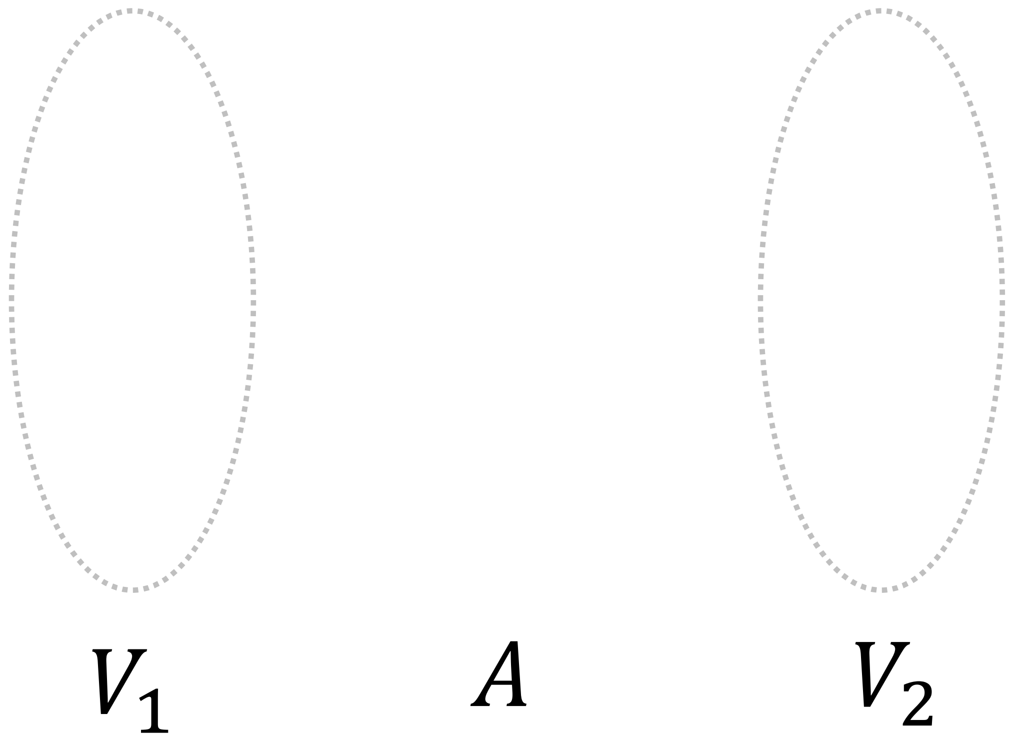
Bigraph

- 정점 집합 V_1, V_2와 간선 집합 A 위에서 Bigraph를 정의



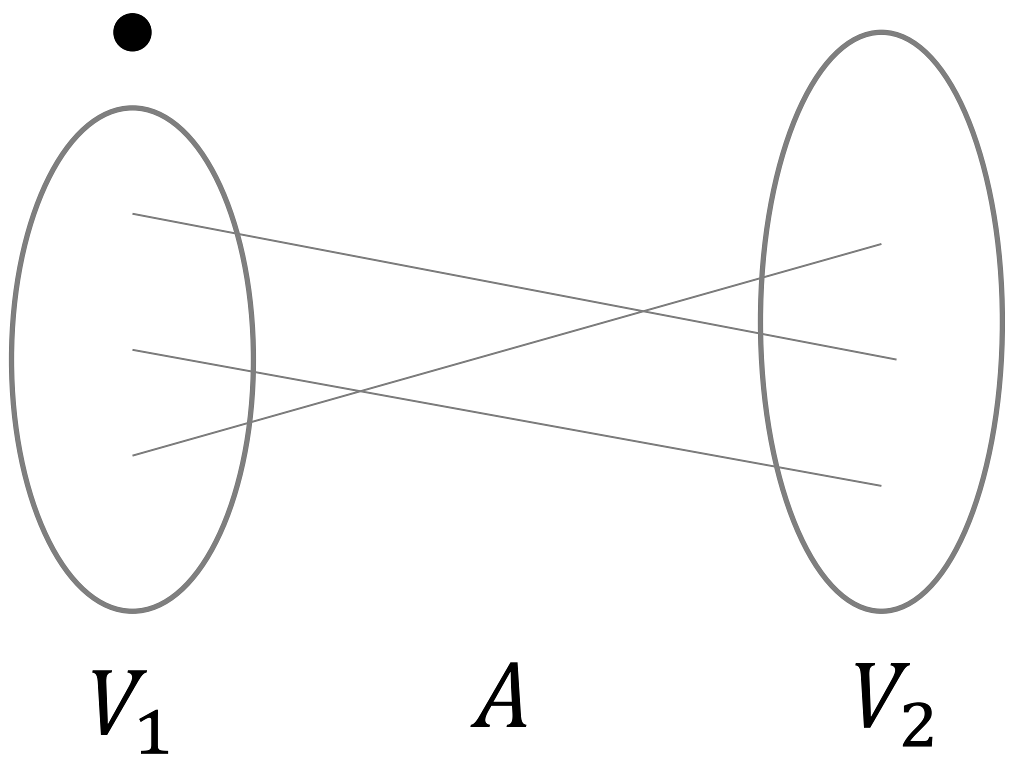
Bigraph

- 공집합



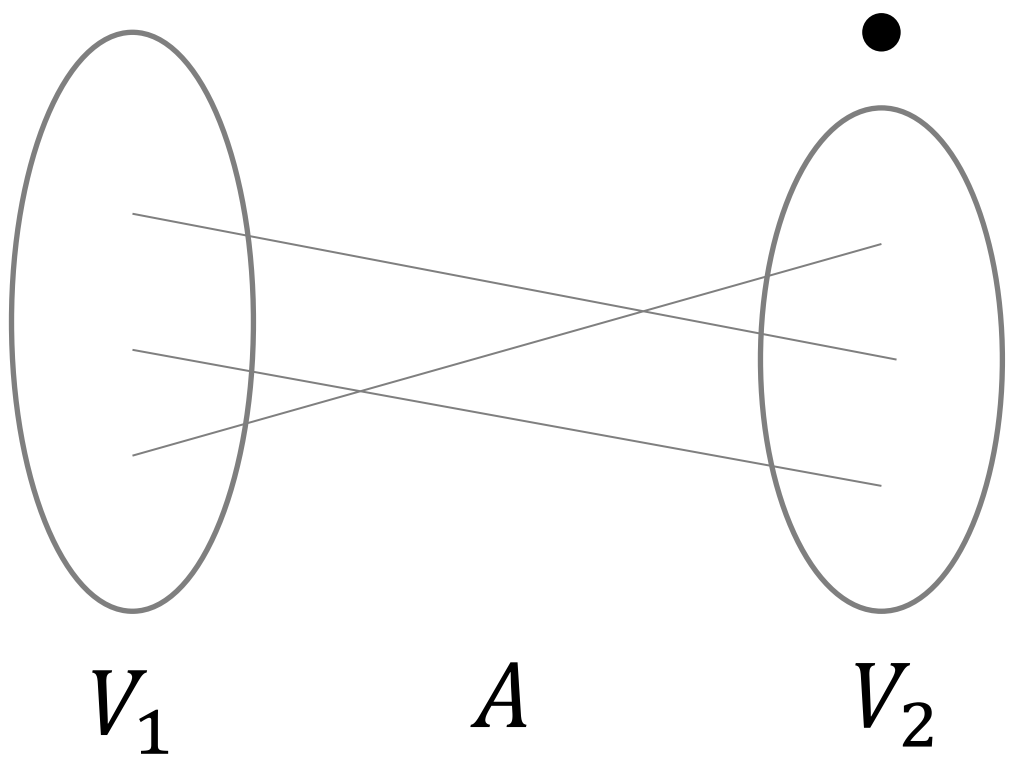
Bigraph

- V_1에 정점 추가



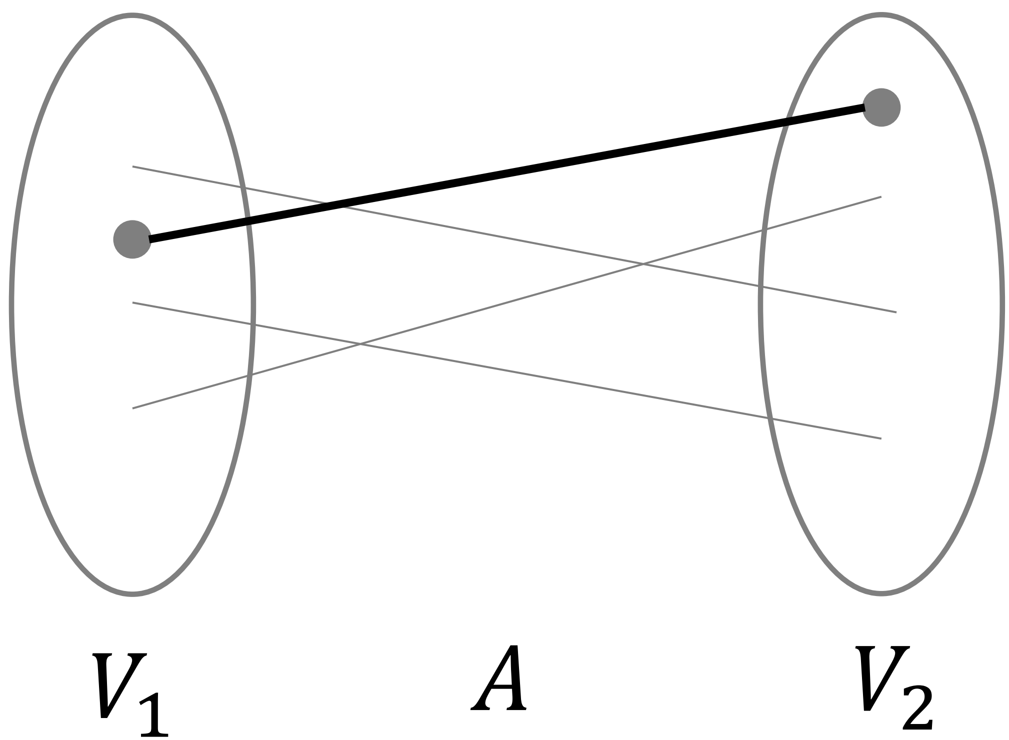
Bigraph

- V_2에 정점 추가



Bigraph

- V_1과 V_2를 잇는 간선 추가



Bigraph

- 공집합
- V_1에 정점 추가
- V_2에 정점 추가
- V_1과 V_2를 잇는 간선 추가

```
Inductive Bigraph : V_set → V_set → A_set → Type :=
  | BG_empty : Bigraph V_empty V_empty A_empty
  | BG_vertex1 :
    ∀ (v1 v2 : V_set) (a : A_set) (d : Bigraph v1 v2 a) (x : Vertex),
      ¬ v1 x → ¬ v2 x →
      Bigraph (V_union (V_single x) v1) v2 a
  | BG_vertex2 :
    ∀ (v1 v2 : V_set) (a : A_set) (d : Bigraph v1 v2 a) (y : Vertex),
      ¬ v1 y → ¬ v2 y →
      Bigraph v1 (V_union (V_single y) v2) a
  | BG_edge :
    ∀ (v1 v2 : V_set) (a : A_set) (d : Bigraph v1 v2 a) (x y : Vertex),
      v1 x → v2 y → x ≠ y →
      ¬ a (A_ends x y) → ¬ a (A_ends y x) →
      Bigraph v1 v2 (A_union (E_set x y) a).
```

Is Bigraph a Bipartite Graph?

Bigraph is a Bipartite Graph

$G = \mathrm{Bigraph}\ V_1\ V_2\ A$ 일 때,

Bigraph is a Bipartite Graph

$G = \text{Bigraph}\ V_1\ V_2\ A$ 일 때,

- V_1, V_2는 G의 정점 집합을 분할한다.

(Bipartition_is_disjoint)

Bigraph is a Bipartite Graph

$G = \text{Bigraph}\ V_1\ V_2\ A$ 일 때,

- V_1, V_2는 G의 정점 집합을 분할한다.

  (Bipartition_is_disjoint)

- 모든 간선은 V_1의 정점과 V_2의 정점을 잇는다.

  (Bigraph_is_bipartite)

Bigraph is a Bipartite Graph

$G =$ Bigraph V_1 V_2 A 일 때,

- V_1, V_2는 G의 정점 집합을 분할한다.

  (Bipartition_is_disjoint)

- 모든 간선은 V_1의 정점과 V_2의 정점을 잇는다.

  (Bigraph_is_bipartite)

- G는 그래프이다. (Bigraph_isa_graph)

앞으로의 할 일

- König's theorem의 증명

앞으로의 할 일

- König's theorem의 증명
- Standard library를 기반으로 한 그래프 정의하기

감사합니다.