# Monthly Meeting on October

**Yuichiro Honda**
Morita lab. M1

2016/10/05

1 Previous work

2 Progress

3 Next step

# Last month

1. searched an (polynomial) algorithm to partition two matroids into their common bases
2. found an algorithm to find all common bases in two matroids in $O(n(n^2 + t)\lambda)$

1 Previous work

2 Progress

3 Next step

# References

Komei Fukuda, Makoto Namiki: “Finding all common bases in two matroids”, Discrete Applied Mathmatics 56 (1995) 231-243

# Finding all common bases in two matroids

## Main result

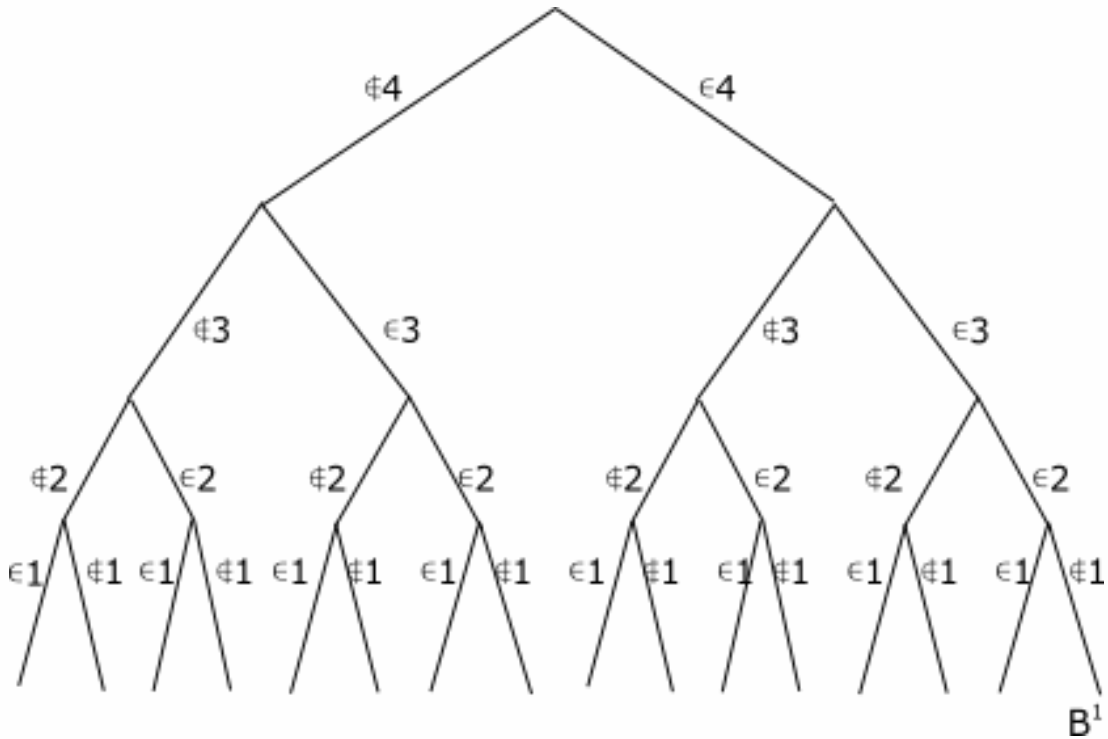
Given two matroids $M_1 = (E, \mathcal{B}_1)$, $M_2 = (E, \mathcal{B}_2)$, and a common base $B^1 \in \mathcal{B}_1 \cap \mathcal{B}_2$, there is an algorithm finding all common bases of them in $O(n(n^2 + t)\lambda)$ where $\lambda$ is number of the bases and $t$ is the time to make one pivot operation.

# Finding all common bases in two matroids

- The algorithm use enumeration tree whose height is $|E| = n$ and the height is indexed in descending order from the root
- In the beginning, the initial comon base $B^1$ is placed at right most position
- $n$-dimentional vector $P \in \{L, R\}^n$ is used to indicate the current position
- also current branch is indicated by $f \in E$
- Each one step of algorithm, $B^k$ goes left updating $B^k$ to $B^{k+1}$ if exists

# Finding all common bases in two matroids

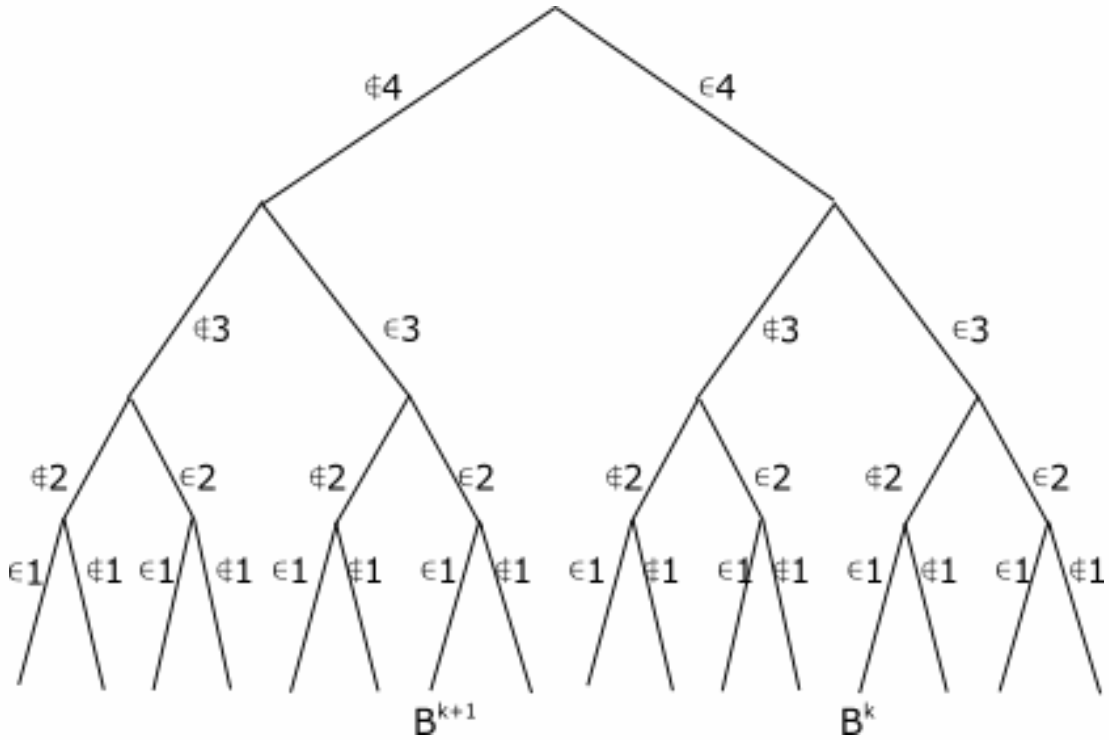
initial state:




$$B^1 = \{2, 3, 4\}$$
$$P = (R\,R\,R\,R),\ f = 1$$

# Finding all common bases in two matroids

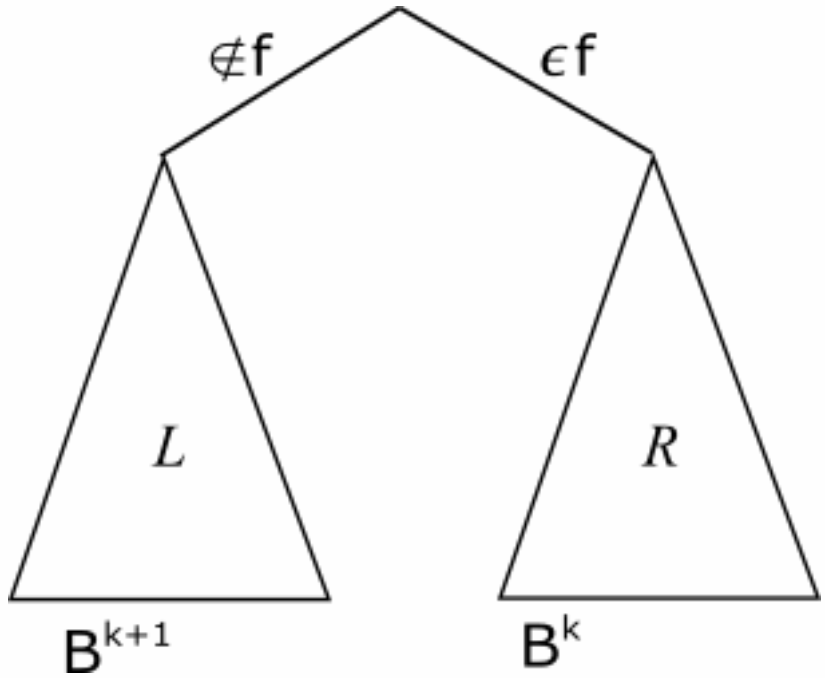
example:




when the current base $B^k = \{1, 3, 4\}$ is placed like above,
$P = (L\,L\,R\,R)$ and $f = 3$

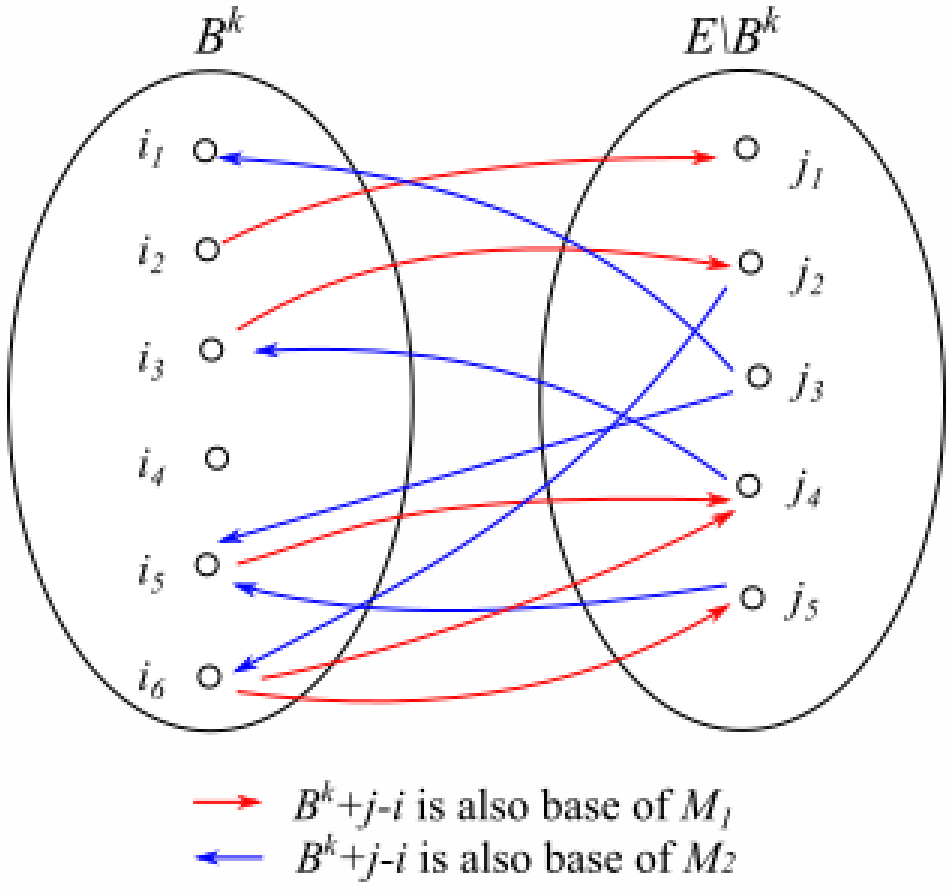
# Finding all common bases in two matroids

next, we need a theorem to reduce the search space like a backtracking



# Finding all common bases in two matroids

bipartite graph $G(B^k)$



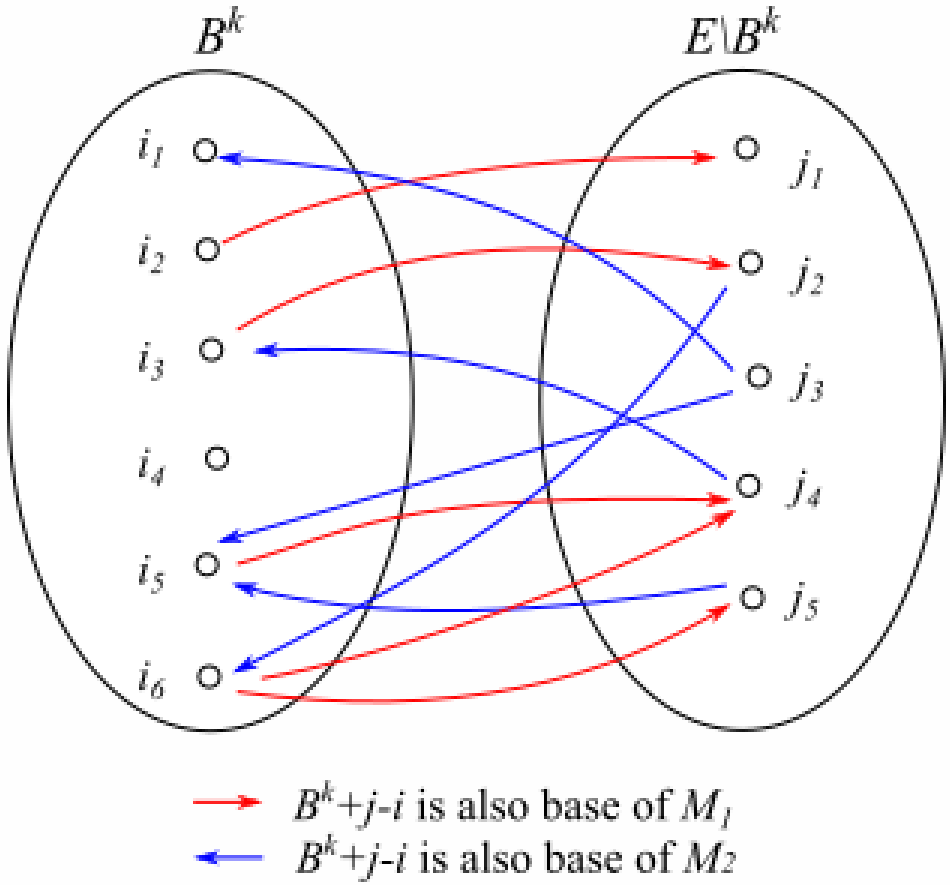
# Finding all common bases in two matroids

## Theorem 1

There exists a common base $B^{k+1}$ in $L$ if and only if there exists a directed cycle $C$ in $G(B^k)$ which contains $f$ and consists of elements in $E$ less than or equal to $f$.

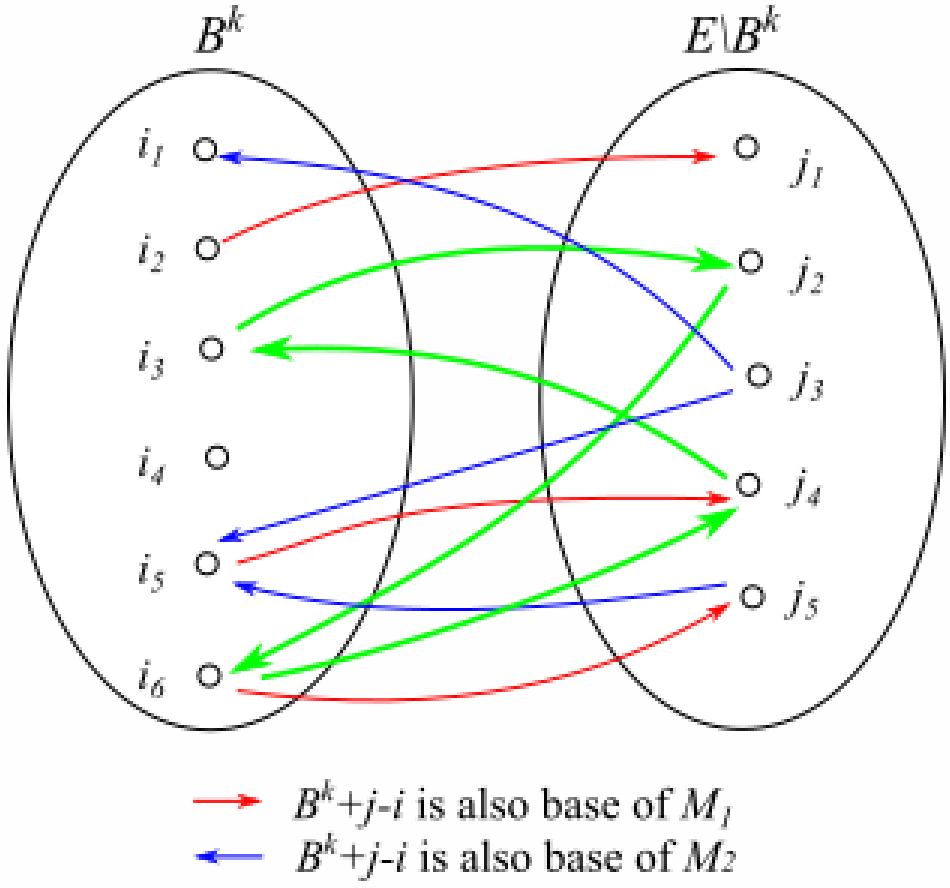
# Finding all common bases in two matroids

bipartite graph $G(B^k)$
when $f = i_6$...

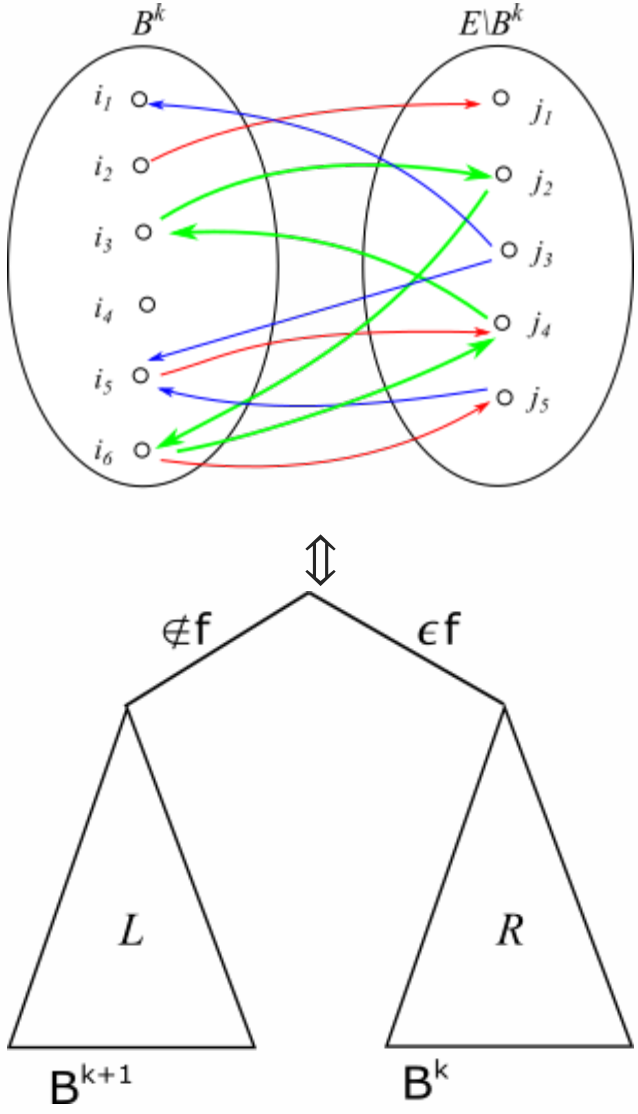


# Finding all common bases in two matroids

bipartite graph $G(B^k)$
when $f = i_6$...



# Finding all common bases in two matroids




$$B^{k+1} = B^k \cup \{j_3,\ j_4\} \setminus \{i_3,\ i_6\} \in \mathcal{B}_1 \cap \mathcal{B}_2$$

# Finding all common bases in two matroids

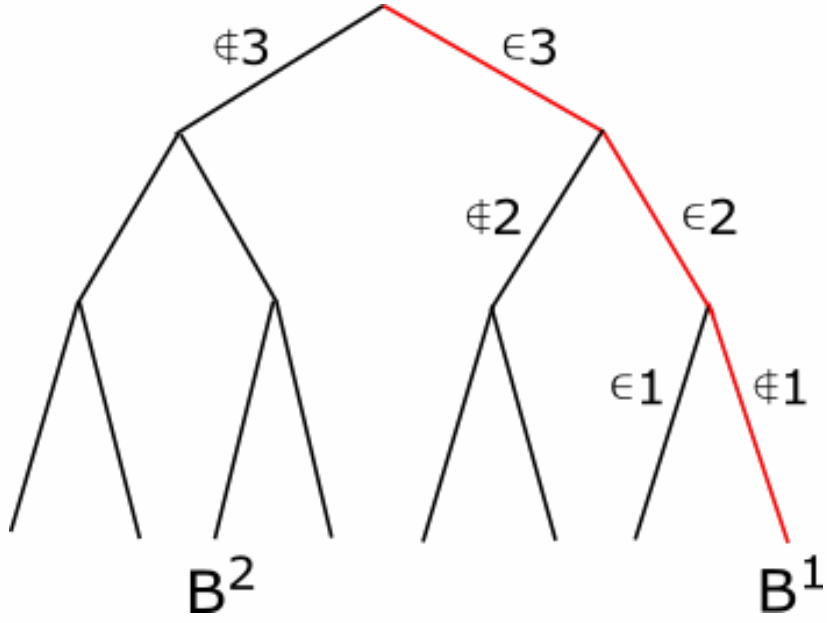
simple algorithm example:

$M_1 = M_2 = (E, \mathcal{B})$

$E = \{1, 2, 3\}, \mathcal{B} = \{\{1, 2\}, \{2, 3\}\}$

$B^1 = \{2, 3\}$

# Finding all common bases in two matroids

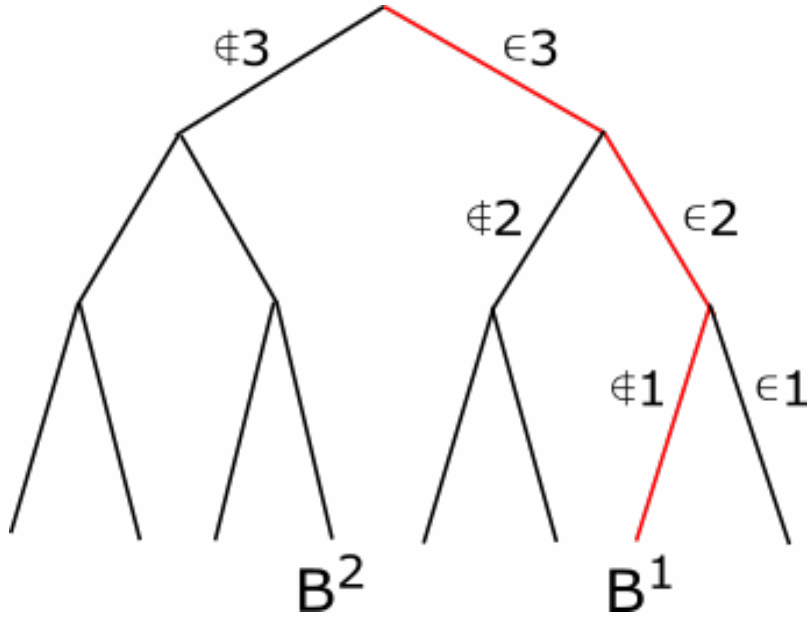



$B^1 = \{2, 3\}$

$P = (R\,R\,R)$

$P_1 = R$ then $f = 1$

# Finding all common bases in two matroids

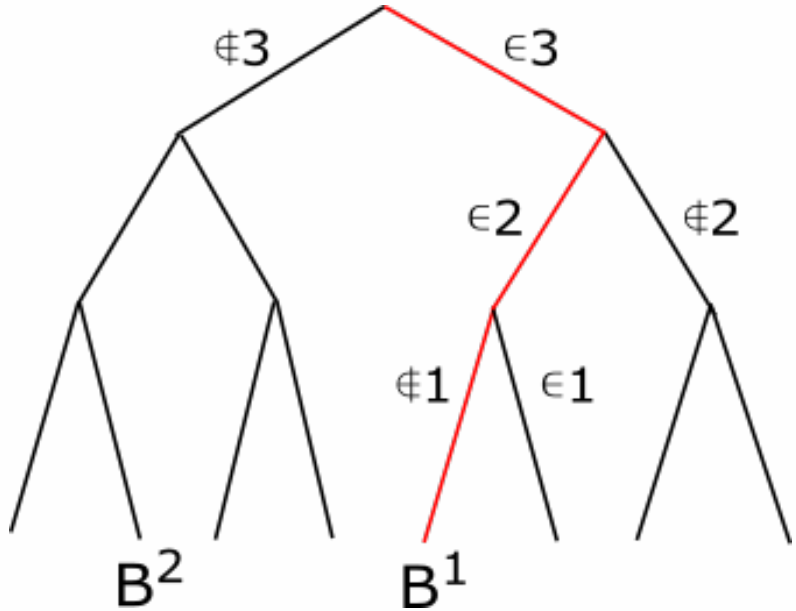



$B^1 = \{2, 3\}$

$P = (L\,R\,R)$

$P_2 = R$ then $f = 2$

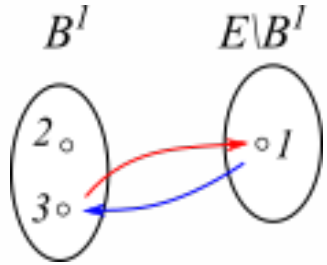
# Finding all common bases in two matroids




$B^1 = \{2, 3\}$

$P = (L\,L\,R)$

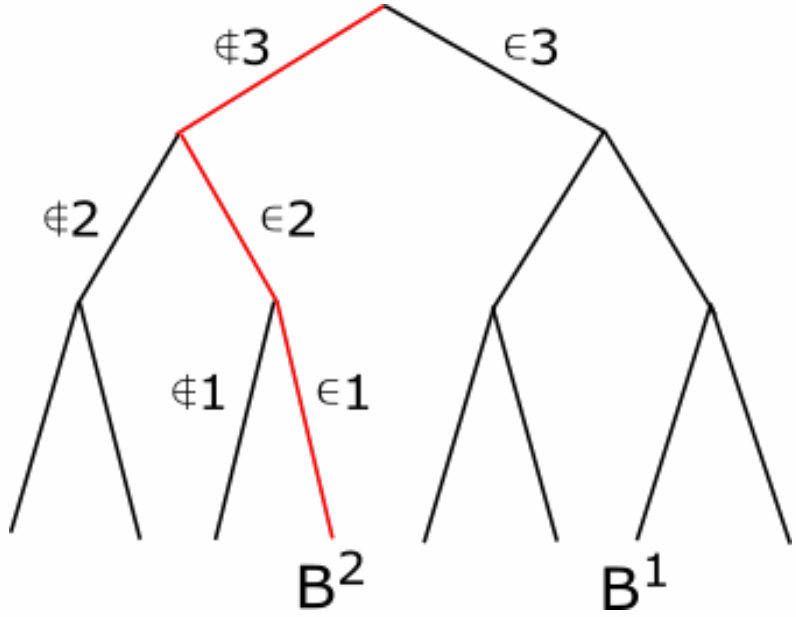
$P_3 = R$ then $f = 3$




there exists an directed cycle $C$ in bipartite graph $G(B^1)$

$\rightarrow$make pivot operation along $C$ and get new base $B^2 = \{1, 2\}$

# Finding all common bases in two matroids

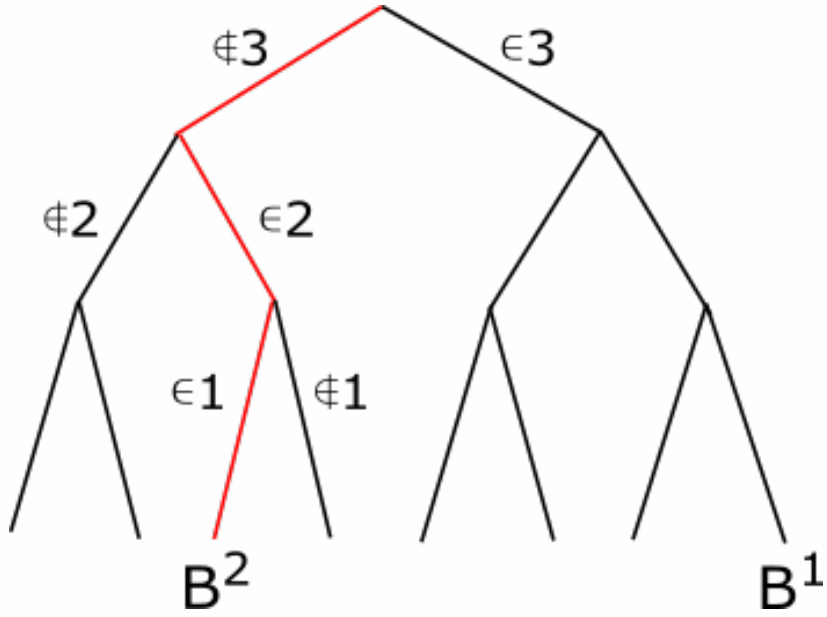



$$B^2 = \{1, 2\}$$

$$P = (R\,R\,L)$$

$$P_1 = R \text{ then } f = 1$$

# Finding all common bases in two matroids

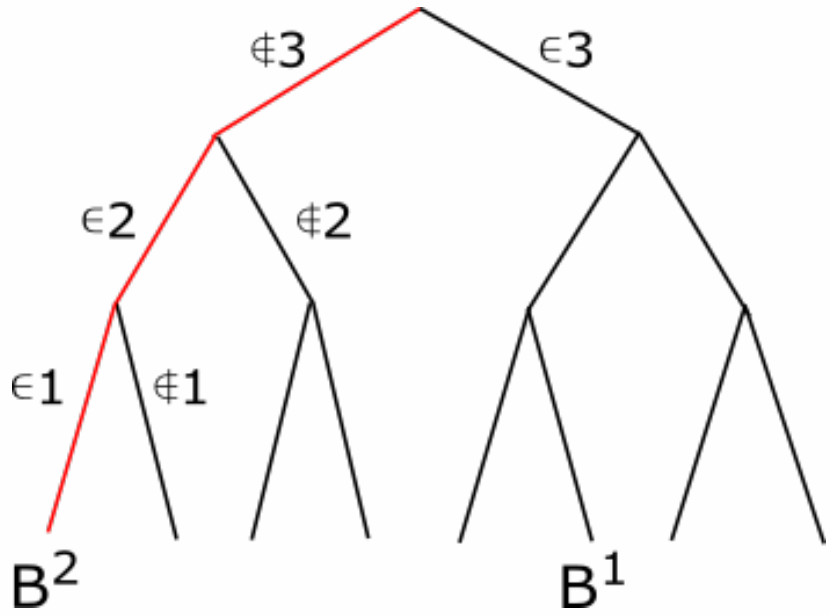



$B^2 = \{1, 2\}$

$P = (L\,R\,L)$

$P_2 = R$ then $f = 2$

# Finding all common bases in two matroids




$B^2 = \{1, 2\}$

$P = (L\,L\,L)$

$f$ doesn't exist

end

# Finding all common bases in two matroids

## alogrithm “Enumerate”

input: $M_1 := (E, \mathcal{B}_1)$, $M_2 := (E, \mathcal{B}_2)$, $B^1 \in \mathcal{B}_1 \cap \mathcal{B}_2$
output: all common bases of $M_1$ and $M_2$

1. $k := 1$, output $B^k$. $P := (R, ..., R)$. (Here, $P \in \{L, R\}^n$)
2. $f := min\{j \mid P_j = R\}$
3. If $f$ doesn’t exist, stop. Else, create $G(B^k)$ and find directed cycle $C$ satisfying theorem 1 with breadth first search.
4. If $C$ doesn’t exist, goto step 2. Else, $k := k + 1$, make pivot operation along $C$ and obtain new $B^k$. $P_f := L$, $\forall j < k$; $P_j := R$, goto step 2.

# Finding all common bases in two matroids

Again, $n = |E|$, $\lambda = |\mathcal{B}_1 \cap \mathcal{B}_2|$, and $t$ is time to generate $G(B^k)$.

## time complexity

We evaluate the time from obtaining $B^k$ to obtaining $B^{k+1}$. Let $A_{max}$ be the maximum number of the arc in $G(B^k)$. Then $C$ in $G(B^k)$ satisfying Theorem 1 can be found in $O(A_{max})$ using BFS. This is iterated at most $n$ times(until finding such $C$). If such $C$ exists, then we pivot elements in $B^k$ along $C$ to get $B^{k+1}$ and generate $G(B^{k+1})$. This takes $O(C_{max}t)$ where $C_{max}$ is the length of the $C$.
Since we know $A_{max} \leq (\frac{n}{2})^2 \cdot 2 = \frac{n^2}{2}$ and $C_{max} \leq 2 \cdot \frac{n}{2} = n$, it takes $O(n^3 + nt)$ from $B^k$ to $B^{k+1}$.
This is iterated $\lambda$ times, so total time complexity is $O((n^3 + nt)\lambda)$.

(note: $t$ depends on the imput form of the matroid. for example, the matroid is given in a form of a map, $t = O(n^2)$, but matroid is given in a form of a matrix, it's more than $n^2$)

1. Previous work
2. Progress
3. Next step

# next month

TODO:

1. learn about matroid intersecrion
2. learn how to partition a matoid into bases
3. tackle two different partition matroids (can be partitioned into their common bases)
4. generalized partition matroid of two different uniform matroids and any matroid