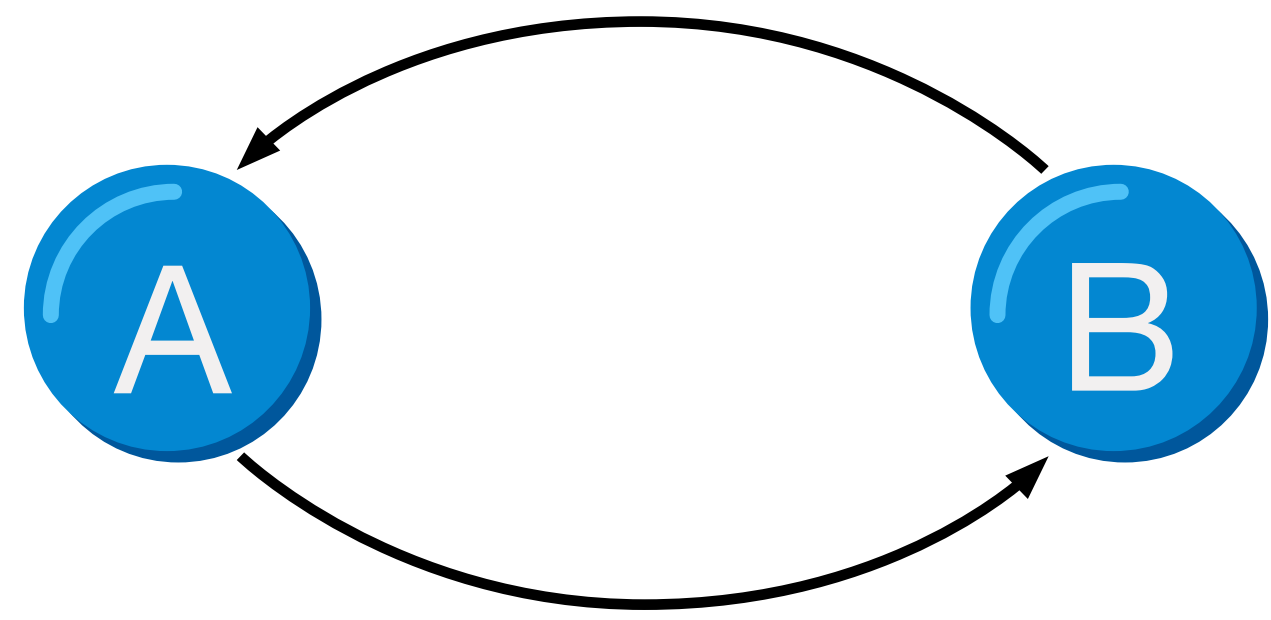


Motif finding DSL

DSL = Domain Specific Language

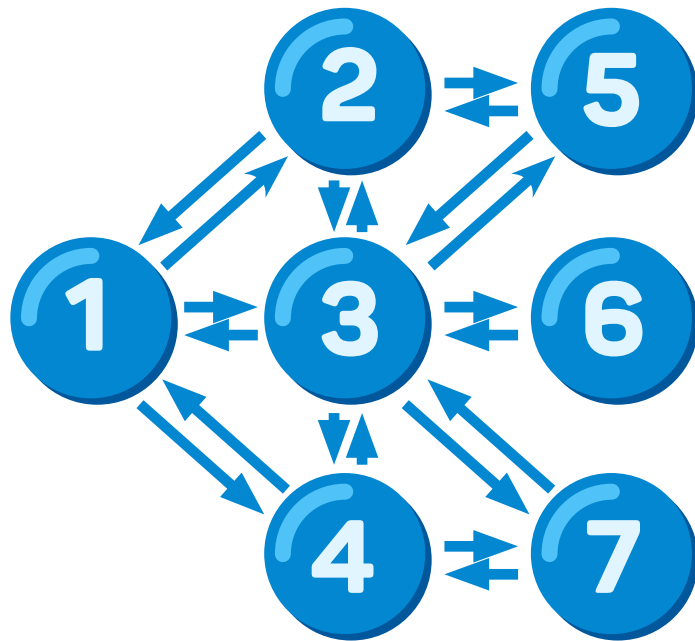
```
graph.find("(a)-[e]->(b); (b)-[e2]->(a)")
```

```
graph.find("(a)-[e]->(b); (b)-[e2]->(a)")
```



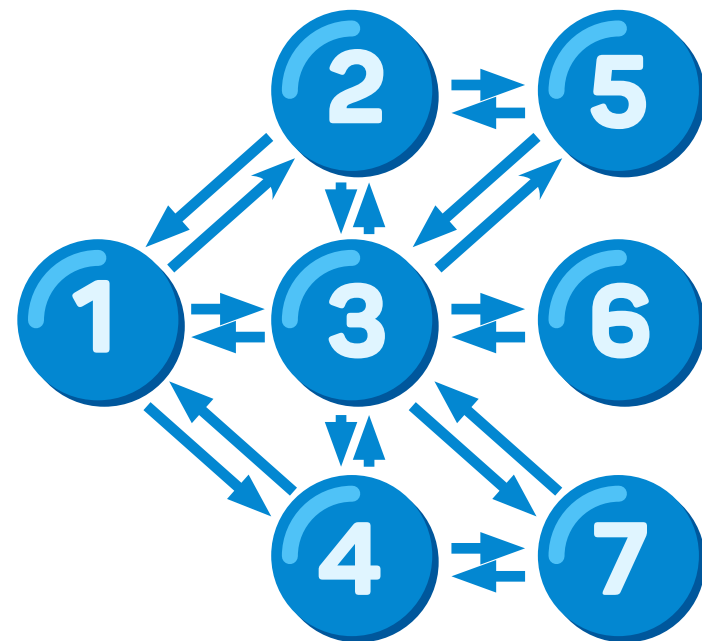
a	e	b	e2
[1,Alex,28,M,MIPT]	[1,2, friend]	[2,Emeli,28,F,MIPT]	[2,1, friend]
[2,Emeli,28,F,MIPT]	[2,4, friend]	[4,Pavel,30,M,MIPT]	[4,2, friend]
[4,Pavel,30,M,MIPT]	[4,5, friend]	[5,Oleg,35,M,MIPT]	[5,4, friend]
[5,Oleg,35,M,MIPT]	[5,1, friend]	[1,Alex,28,M,MIPT]	[1,5, friend]

```
graph.find("(a)-[e]->(b); (b)-[e2]->(a)")
```



```
vertices = sparkSession.createDataFrame([  
  ("1", "Alex", 28, "M", "MIPT"),  
  ("2", "Emeli", 28, "F", "MIPT"),  
  ("3", "Natasha", 27, "F", "SPbSU"),  
  ("4", "Pavel", 30, "M", "MIPT"),  
  ("5", "Oleg", 35, "M", "MIPT"),  
  ("6", "Ivan", 30, "M", "MSU"),  
  ("7", "Ilya", 29, "M", "MSU")  
, ["id", "name", "age", "gender", "university"])
```

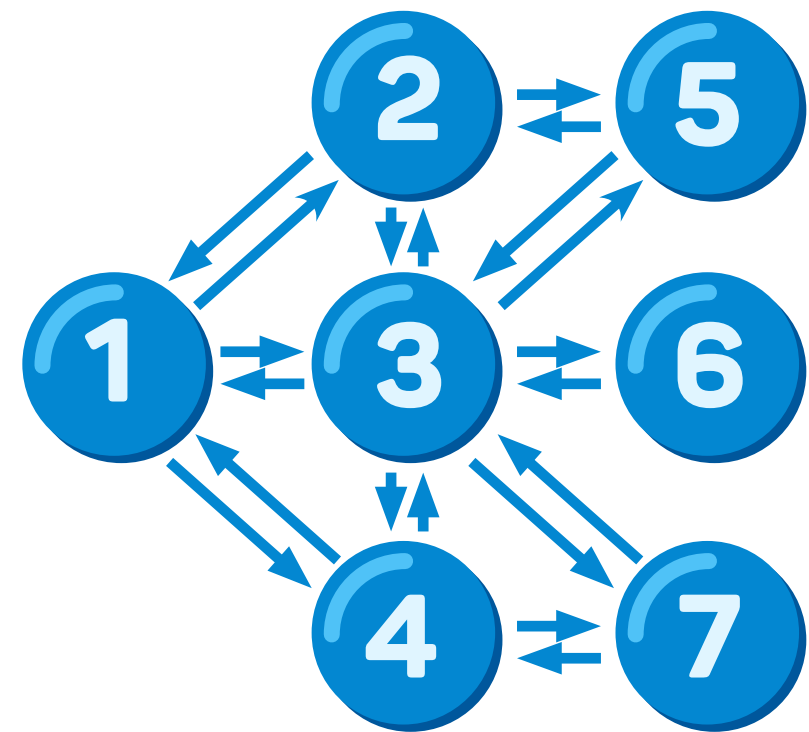
```
graph.find("(a)-[e]->(b); (b)-[e2]->(a)")
```



id	name	age	gender	university
1	Alex	28	M	MIPT
2	Emeli	28	F	MIPT
4	Pavel	30	M	MIPT
5	Oleg	35	M	MIPT

```
vertices = sparkSession.createDataFrame([  
  ("1", "Alex", 28, "M", "MIPT"), ("2", "Emeli", 28, "F", "MIPT"),  
  ("3", "Natasha", 27, "F", "SPbSU"), ("4", "Pavel", 30, "M", "MIPT"),  
  ("5", "Oleg", 35, "M", "MIPT"), ("6", "Ivan", 30, "M", "MSU"),  
  ("7", "Ilya", 29, "M", "MSU")  
], ["id", "name", "age", "gender", "university"])
```

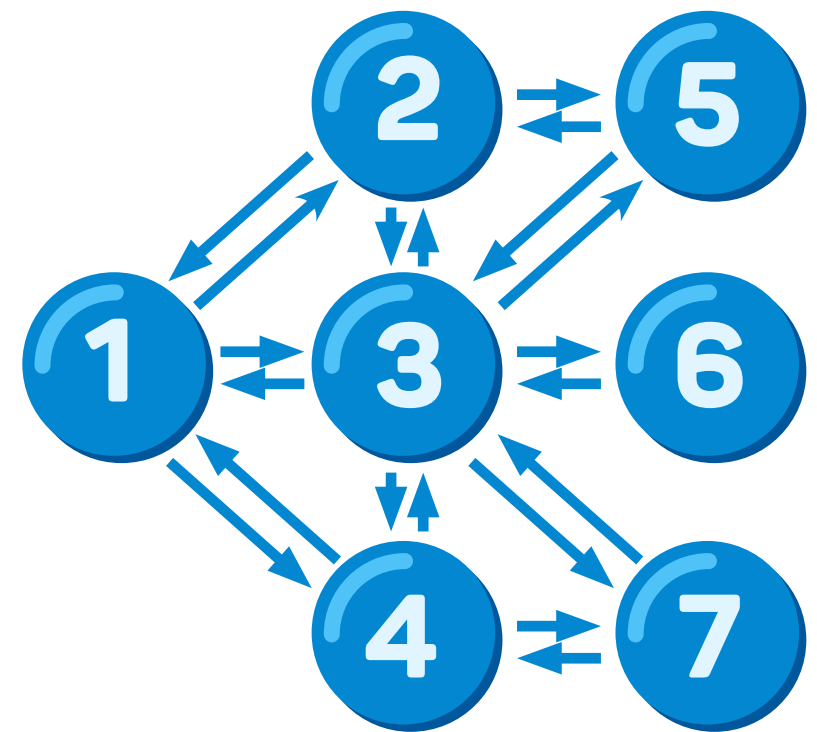
graph.find("(a)-[e]->(b); (b)-[e2]->(a)")



id	name	age	gender	university
1	Alex	28	M	MIPT
2	Emeli	28	F	MIPT
4	Pavel	30	M	MIPT
5	Oleg	35	M	MIPT

a	e	b	e2
[1,Alex,28,M,MIPT]	[1,2, friend]	[2,Emeli,28,F,MIPT]	[2,1, friend]
[2,Emeli,28,F,MIPT]	[2,4, friend]	[4,Pavel,30,M,MIPT]	[4,2, friend]
[4,Pavel,30,M,MIPT]	[4,5, friend]	[5,Oleg,35,M,MIPT]	[5,4, friend]
[5,Oleg,35,M,MIPT]	[5,1, friend]	[1,Alex,28,M,MIPT]	[1,5, friend]

`graph.find("(a)-[e]->(b); (b)-[e2]->(a)")`



id	name	age	gender	university
1	Alex	28	M	MIPT
2	Emeli	28	F	MIPT
4	Pavel	30	M	MIPT
5	Oleg	35	M	MIPT

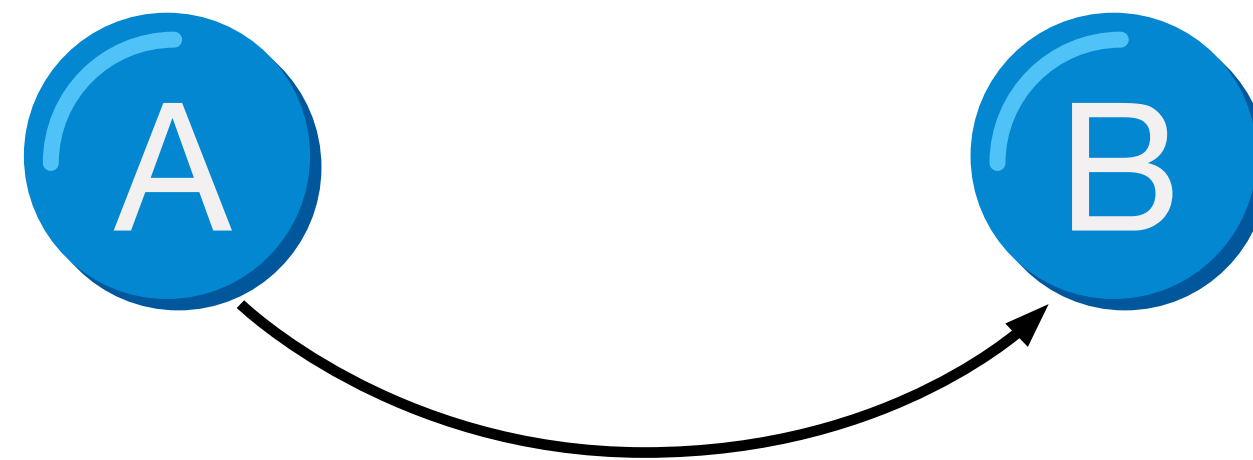
`nestAsCol(df, colName)`

a
[1,Alex,28,M,MIPT]
[2,Emeli,28,F,MIPT]
[4,Pavel,30,M,MIPT]
[5,Oleg,35,M,MIPT]

DSL:

1. Edge

"(a) - [e] -> (b)"

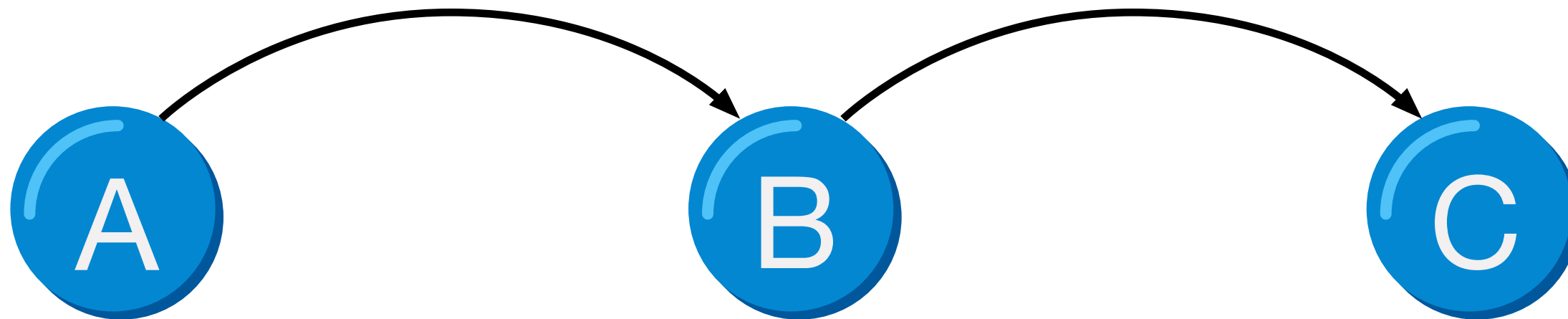


DSL:

1. Edge

2. Union of edges

"(a) - [e] -> (b); (b) - [e2] -> (c)"



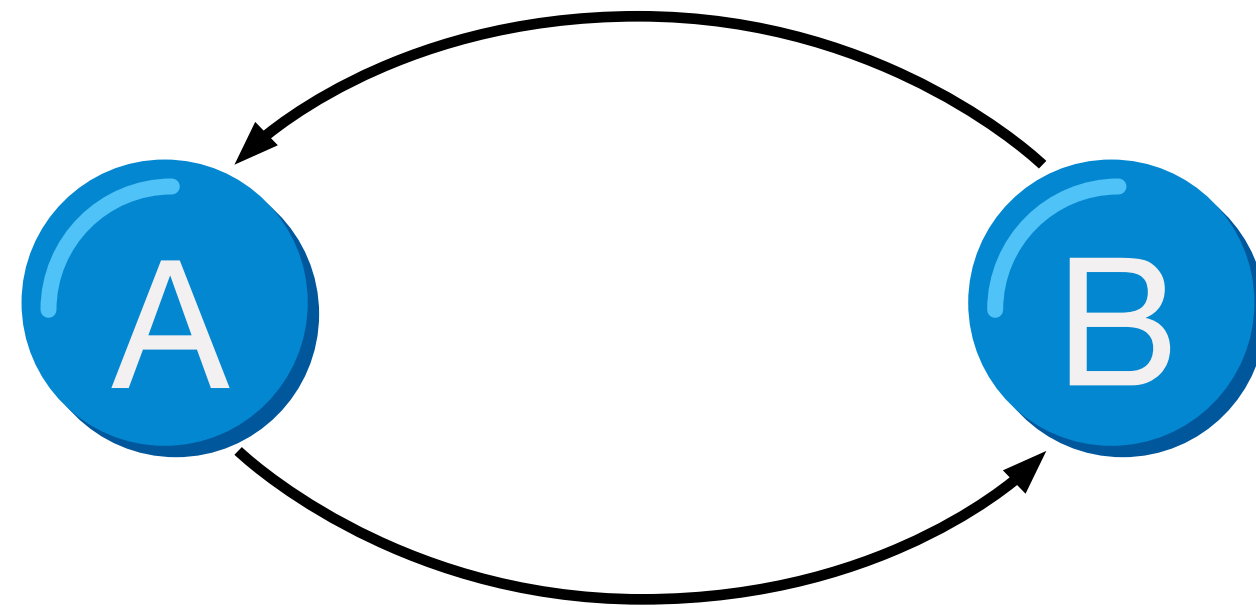
DSL:

1. Edge

2. Union of edges

3. Names

"(a) - [e] -> (b)"



DSL:

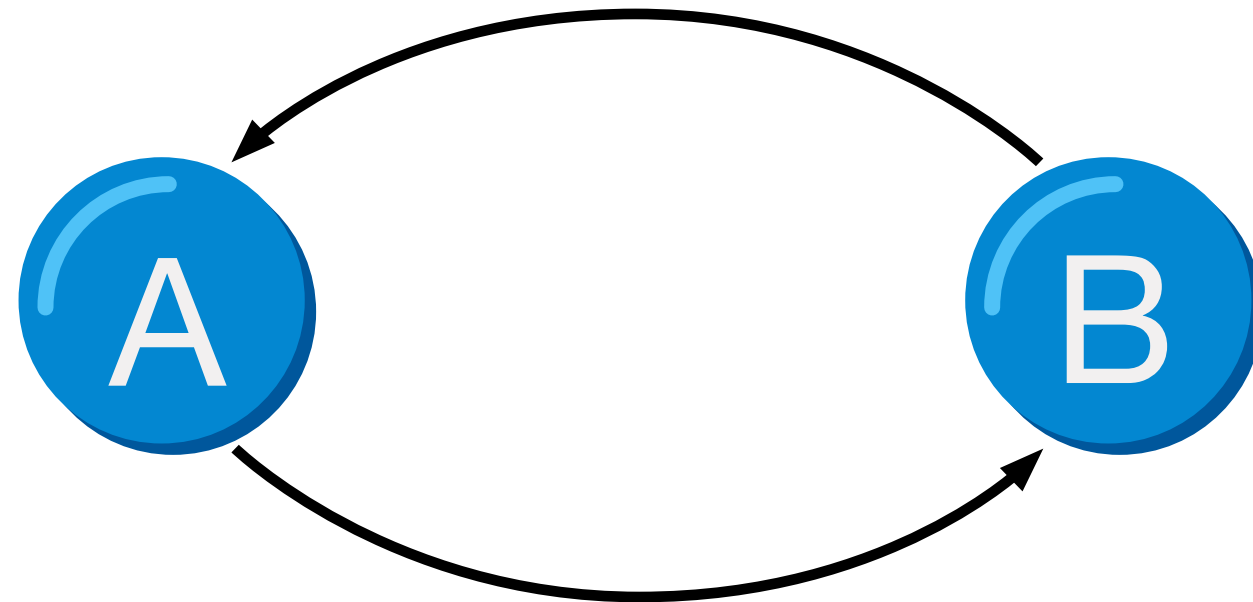
1. Edge

2. Union of edges

3. Names:

1. Identify common elements

"(a) - [e] -> (b); (b) - [e2] -> (a)"



DSL:

1. Edge

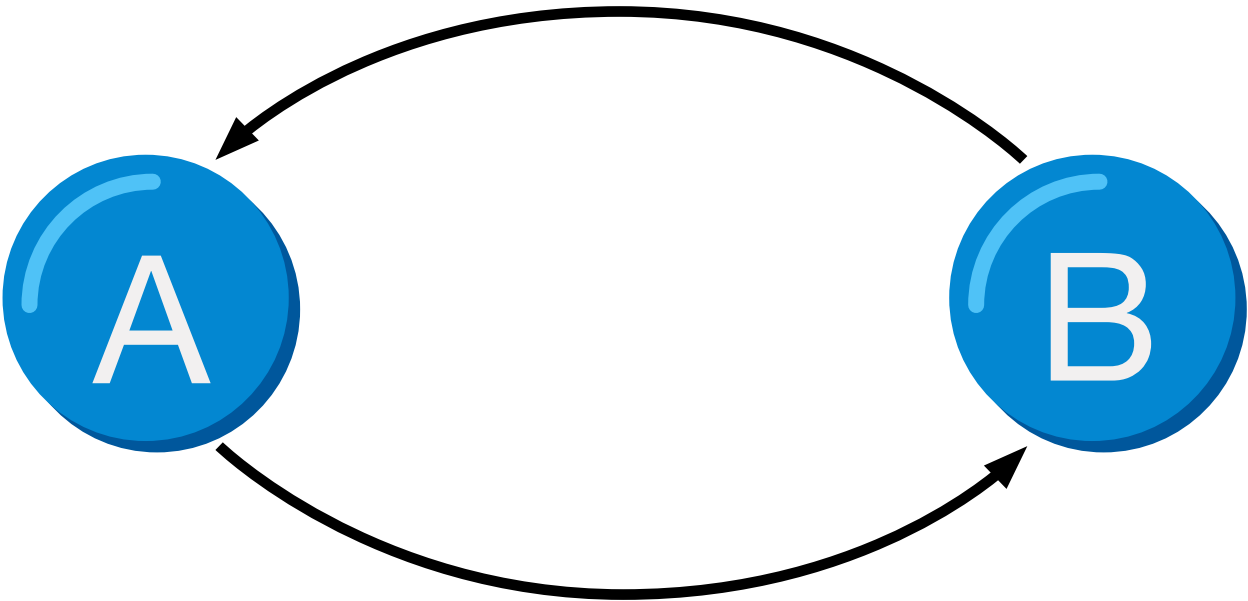
2. Union of edges

3. Names:

1. Identify common elements

2. Identify names of columns in the result DataFrame

"(a) - [e] -> (b); (b) - [e2] -> (a)"



a	e	b	e2
[1,Alex,28,M,MIPT]	[1,2, friend]	[2,Emeli,28,F,MIPT]	[2,1, friend]
[2,Emeli,28,F,MIPT]	[2,4, friend]	[4,Pavel,30,M,MIPT]	[4,2, friend]
[4,Pavel,30,M,MIPT]	[4,5, friend]	[5,Oleg,35,M,MIPT]	[5,4, friend]
[5,Oleg,35,M,MIPT]	[5,1, friend]	[1,Alex,28,M,MIPT]	[1,5, friend]

DSL:

1. Edge

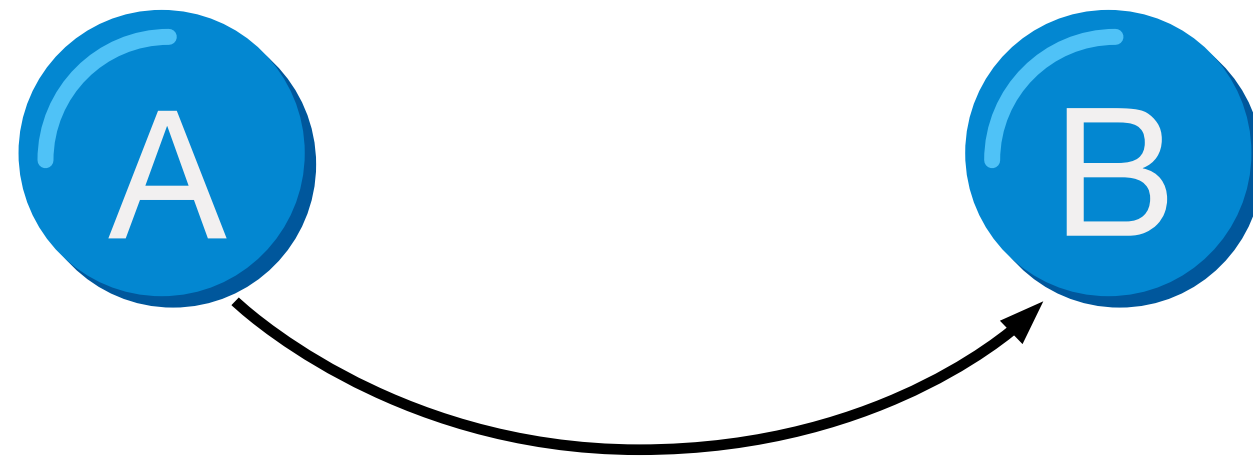
2. Union of edges

3. Names:

- Identify common elements
- Identify names of columns in the result DataFrame

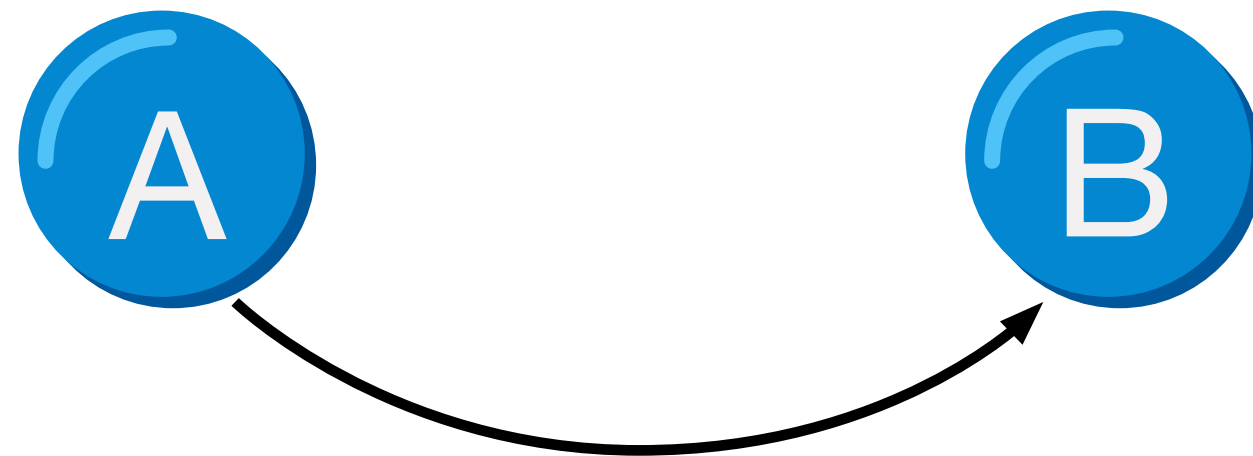
4. Anonymous edges and vertices

"(a) - [] -> (b)"



a	b
[1,Alex,28,M,MIPT]	[2,Emeli,28,F,MIPT]
[2,Emeli,28,F,MIPT]	[4,Pavel,30,M,MIPT]
[4,Pavel,30,M,MIPT]	[5,Oleg,35,M,MIPT]
[5,Oleg,35,M,MIPT]	[1,Alex,28,M,MIPT]

"(a) - [e] -> ()"



a	e
[1,Alex,28,M,MIPT]	[1,2, friend]
[2,Emeli,28,F,MIPT]	[2,4, friend]
[4,Pavel,30,M,MIPT]	[4,5, friend]
[5,Oleg,35,M,MIPT]	[5,1, friend]

DSL:

1. Edge

2. Union of edges

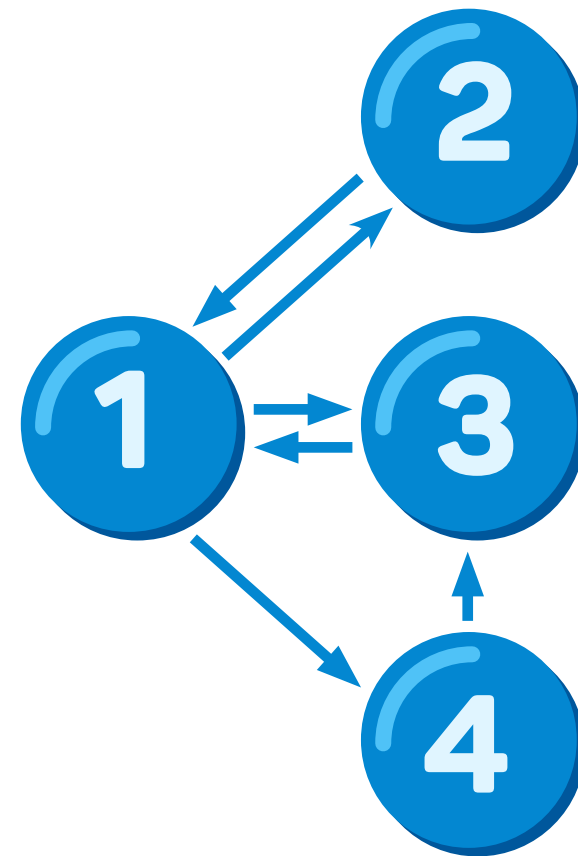
3. Names:

- Identify common elements
- Identify names of columns in the result DataFrame

4. Anonymous edges and vertices

5. Negation

"(a) - [] -> (b); !(b) - [] -> (a)"



a	b
1	4
4	3

Summary

- Syntax of Motif finding Domain Specific Language