Kafka pairwise domain/range cardinality combinations constraint discovery results

Generation of instances by combining every value that the cardinality of the domain set and range set can take, for every relation pair.

- isOneOne defines a 1 to 1 relationship where #domain = #range = 1
- isOneMany defines a 1 to Many relationship where #domain = 1, #range = 2
- isManyOne defines a Many to 1 relationship where #domain = 2, #range = 1
- ***NOTE**:* isManyMany is not strictly equivalent to a many-many relation. It just means that #domain = 2 and #range=2.

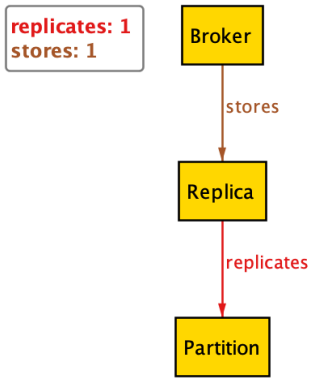
Approach

- The main idea is to constrain any given pair of relations by applying any of the four predicates to each relation: [isOneOne, isOneMany, isManyOne, isManyMany]
- For a pair of relations, there will be total 4 x 4 = 16 constraints
- For `n` relations: total nC2 * 16 constraints = 8n * (n - 1) constraints

Constraint 1

```
run constraint1 {
        isOneOne[replicates]
        isOneOne[stores]
}
```

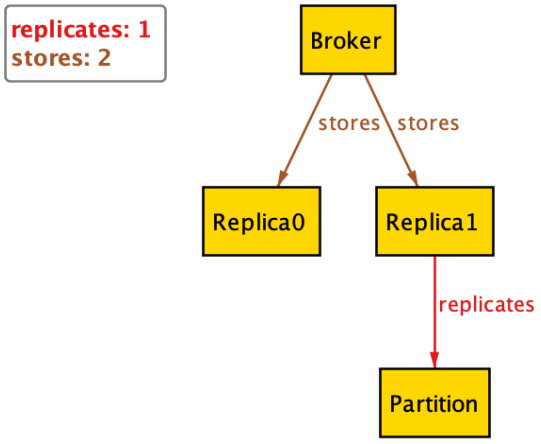
Legal



Constraint 2

```
run constraint2 {
        isOneOne[replicates]
        isOneMany[stores]
}
```

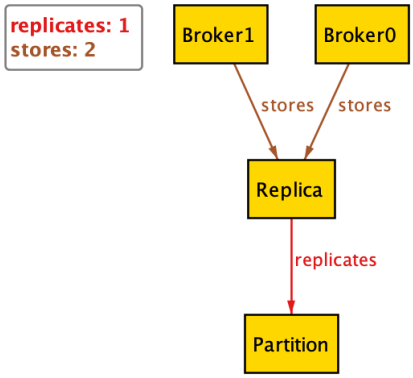
Violation: Replica0 is not associated with a partition



Constraint 3

```
run constraint3 {
        isOneOne[replicates]
        isManyOne[stores]
}
```

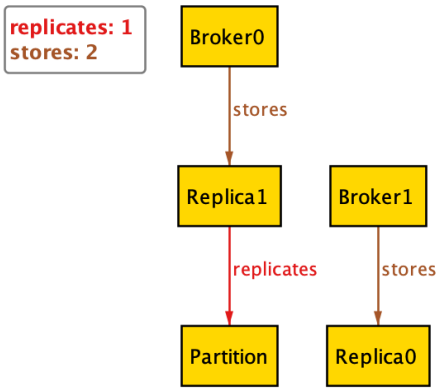
Violation: One replica cannot be stored on multiple brokers



Constraint 4

```
run constraint4 {
        isOneOne[replicates]
        isManyMany[stores]
}
```

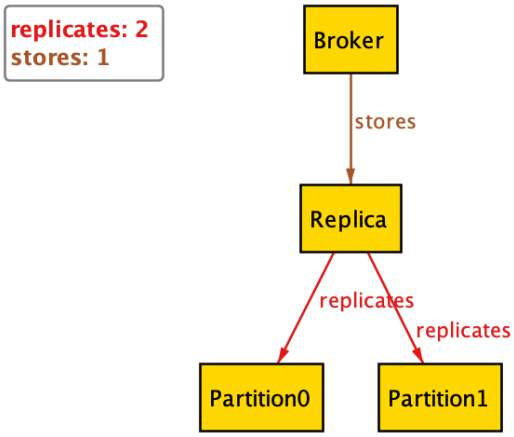
Violation: Replica0 is not associated with a partition



Constraint 5

```
run constraint5 {
        isOneMany[replicates]
        isOneOne[stores]
}
```

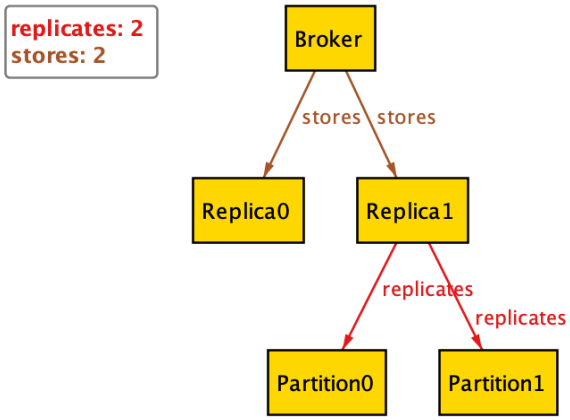
Violation: One replica cannot be associated with two partitions



Constraint 6

```
run constraint6 {
        isOneMany[replicates]
        isOneMany[stores]
}
```

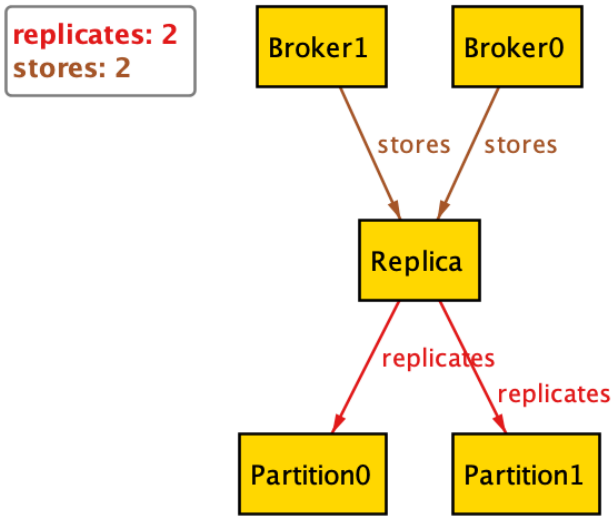
Violation: Replica0 is not associated with a partition



Constraint 7

```
run constraint7 {
        isOneMany[replicates]
        isManyOne[stores]
}
```

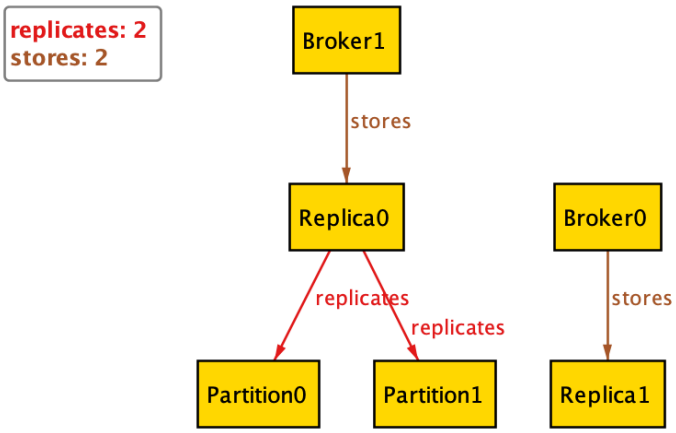
Violation: one replica cannot be stored on multiple brokers, replica cannot be associated with multiple partitions



Constraint 8

```
run constraint8 {
        isOneMany[replicates]
        isManyMany[stores]
}
```

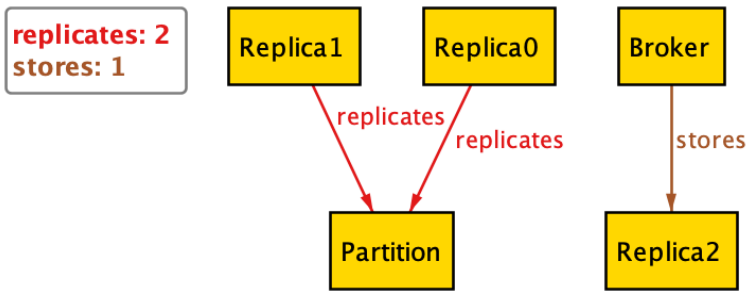
Violation: Replica1 is not associated with a partition, Replica0 associated with multiple partitions



Constraint 9

```
run constraint9 {
        isManyOne[replicates]
        isOneOne[stores]
}
```

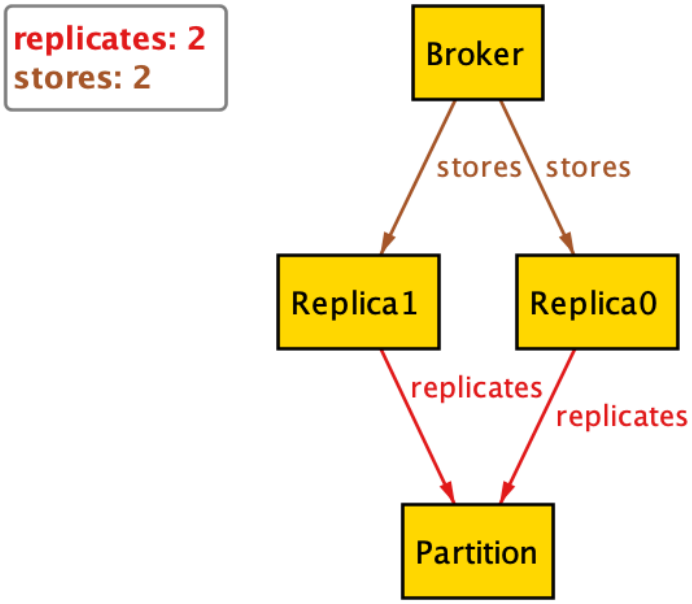
Violation: Replica2 is not associated with a partition



Constraint 10

```
run constraint10 {
        isManyOne[replicates]
        isOneMany[stores]
}
```

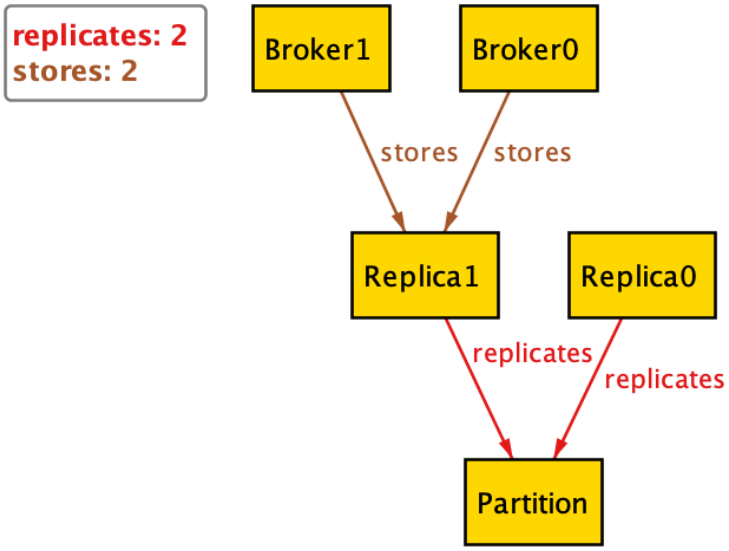
Violation: Two replicas of the same partition should not be stored on the same broker



Constraint 11

```
run constraint11 {
        isManyOne[replicates]
        isManyOne[stores]
}
```

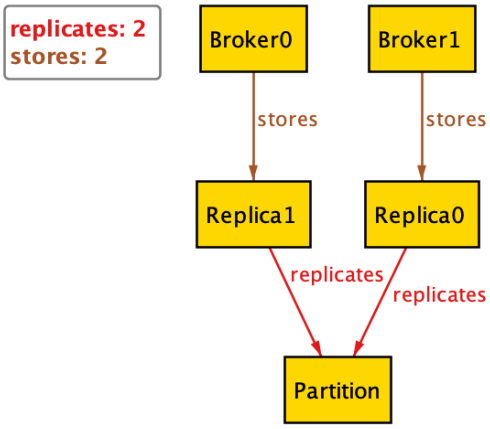
Violation: One replica stored on multiple brokers



Constraint 12

```
run constraint12 {
        isManyOne[replicates]
        isManyMany[stores]
}
```

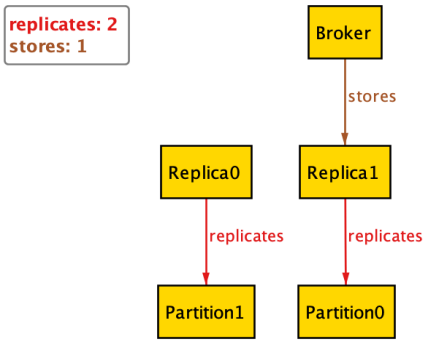
Legal



Constraint 13

```
run constraint13 {
        isManyMany[replicates]
        isOneOne[stores]
}
```

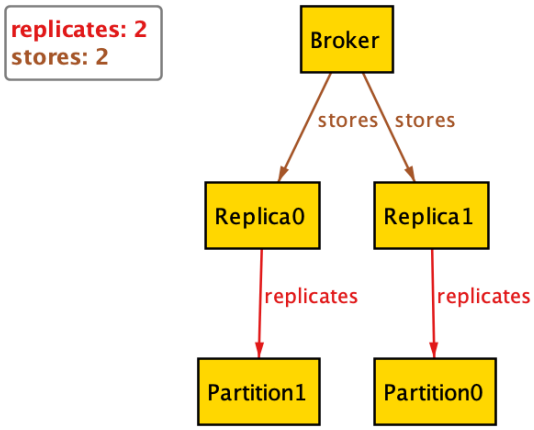
Legal



Constraint 14

```
run constraint14 {
        isManyMany[replicates]
        isOneMany[stores]
}
```

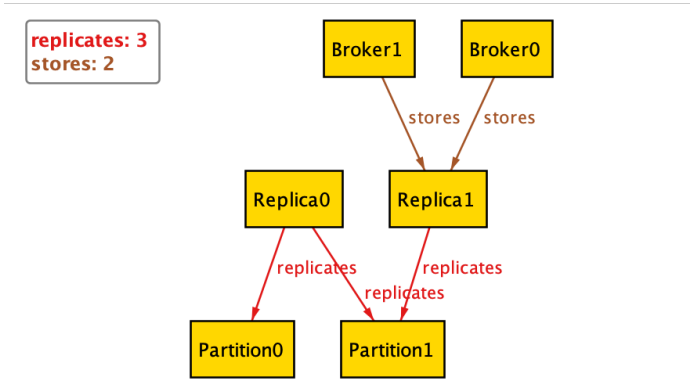
Legal



Constraint 15

```
run constraint15 {
        isManyMany[replicates]
        isManyOne[stores]
}
```

Violation: One replica in multiple brokers, One replica associated with multiple partitions



Constraint 16

```
run constraint16 {
        isManyMany[replicates]
        isManyMany[stores]
}
```

IllLegal

Replica0 replicates two partitions