Basic Kafka pairwise instance generation results

Constraint 1

run constraint1 {
    #replicates = 0
    #stores = 0
}

Shows completely detached signature instances of Partition and Broker signatures
Note: No replica instances shown because of the `replicates: one Partition` constraint in Replica's signature.

Constraint 2

run constraint2 {
    #replicates = 0
    #stores = 1
}

Inconsistent predicates: Replicas cannot exist with #replicates=0 due to `replicates: one Partition` constraint in Replica's signature. #stores=1 is not possible since there are no replicas

```
Executing "Run constraint2"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   362 vars. 27 primary vars. 746 clauses. 45ms.
   No instance found. Predicate may be inconsistent. 11ms.
```

Constraint 3

run constraint3 {
    #replicates = 0
    #stores >= 2
}

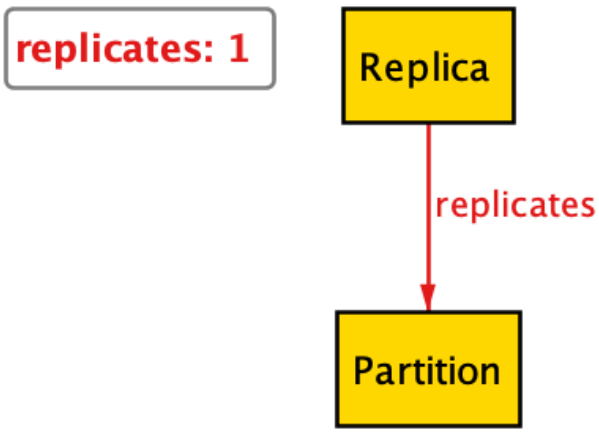
Inconsistent predicates: Replicas cannot exist with #replicates=0 due to `replicates: one Partition` constraint in Replica's signature. #stores >= 2 is not possible since there are no replicas

```
Executing "Run constraint3"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   366 vars. 27 primary vars. 746 clauses. 38ms.
   No instance found. Predicate may be inconsistent. 24ms.
```

Constraint 4

run constraint4 {
    #replicates = 1
    #stores = 0
}

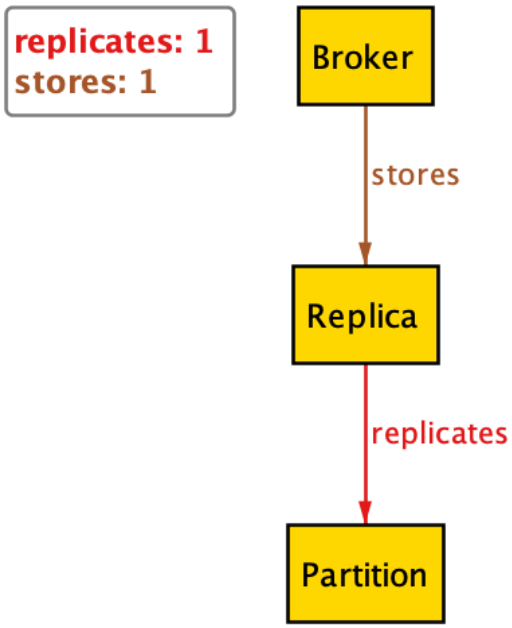
Exactly one simple Replica->Partition relation seen.
Going next will show "Orphaned" Brokers and other Partitions



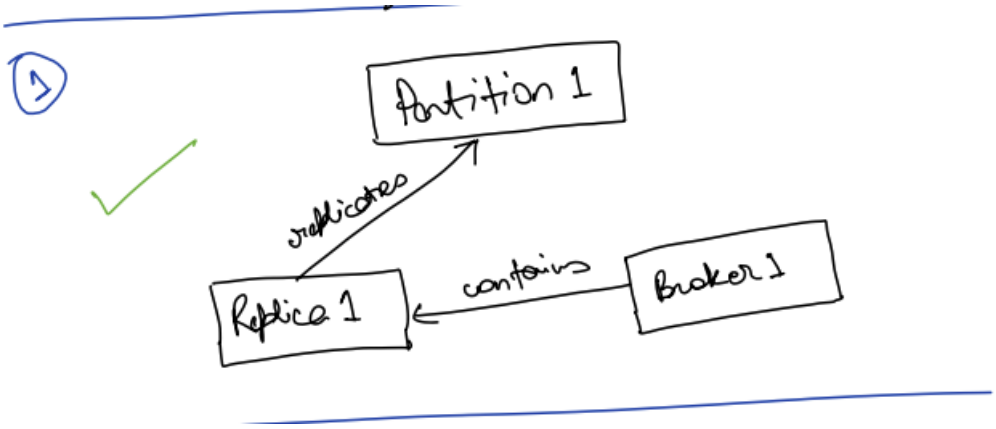
Constraint 5

```
run constraint5 {
        #replicates = 1
        #stores = 1
}
```

Simple Broker --stores--> Replica --replicates--> Partition instance seen. Going next will show other detached objects.



** Same as Instance 1 in pairwise-notes.pdf



Constraint 6

```
run constraint6 {
        #replicates = 1
        #stores >= 2
}
```

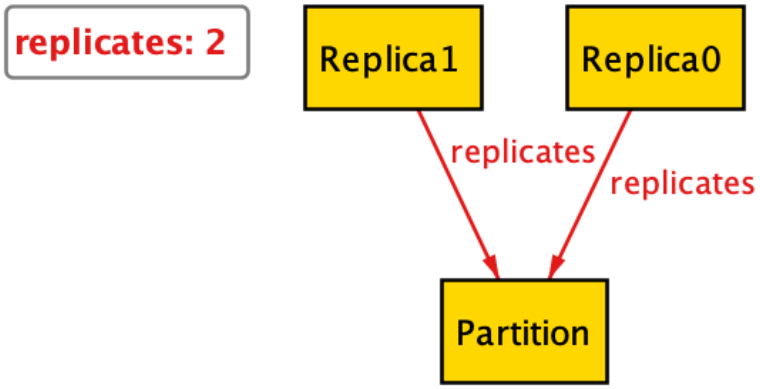
Inconsistent predicates: Exactly one replica exists with #replicates=1 due to `replicates: one Partition`. #stores>=2 is not possible with one replica as multiple brokers cannot store the same replica due to the `disj` in `stores: disj set Replica`.

```
Executing "Run constraint6"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   366 vars. 27 primary vars. 746 clauses. 37ms.
   No instance found. Predicate may be inconsistent. 7ms.
```

Constraint 7

```
run constraint7 {
        #replicates >= 2
        #stores = 0
}
```

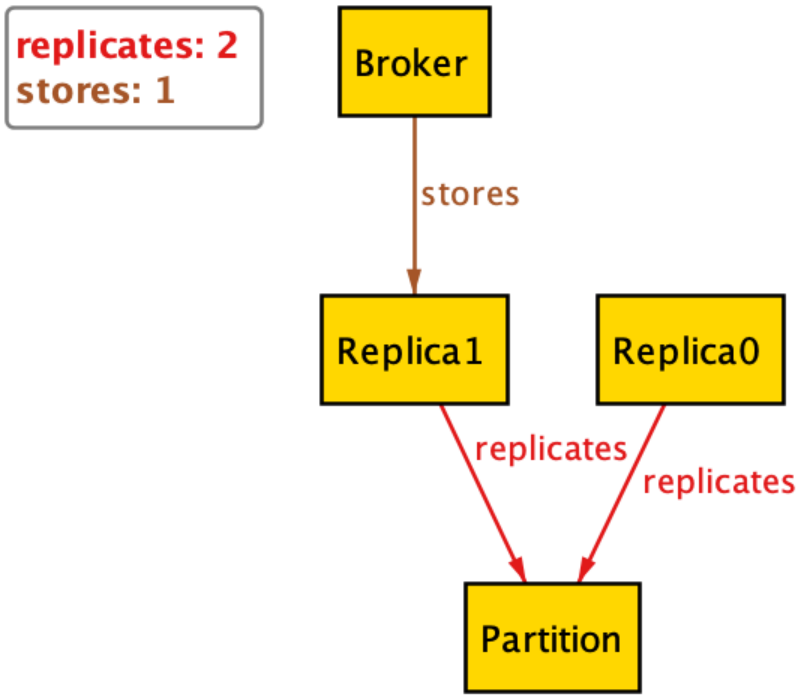
Shows 2 replicas in replicates (related to the same partition). Going next also shows 2+ replicas in replicates (belonging to multiple 2+ partitions)



Constraint 8

```
run constraint8 {
        #replicates >= 2
        #stores = 1
}
```

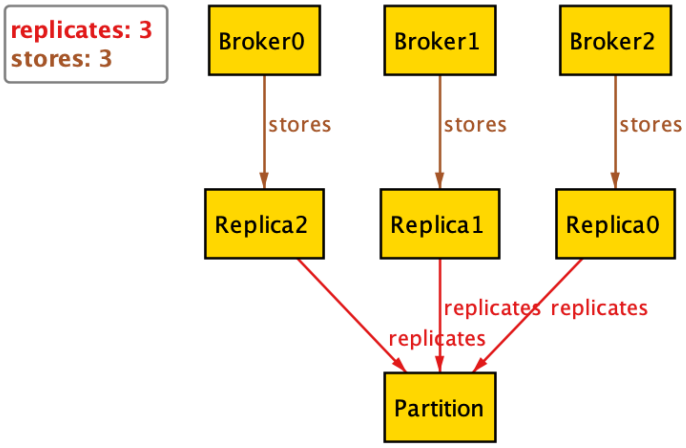
Similar to constraint 7, but with one Broker that stores a replica.



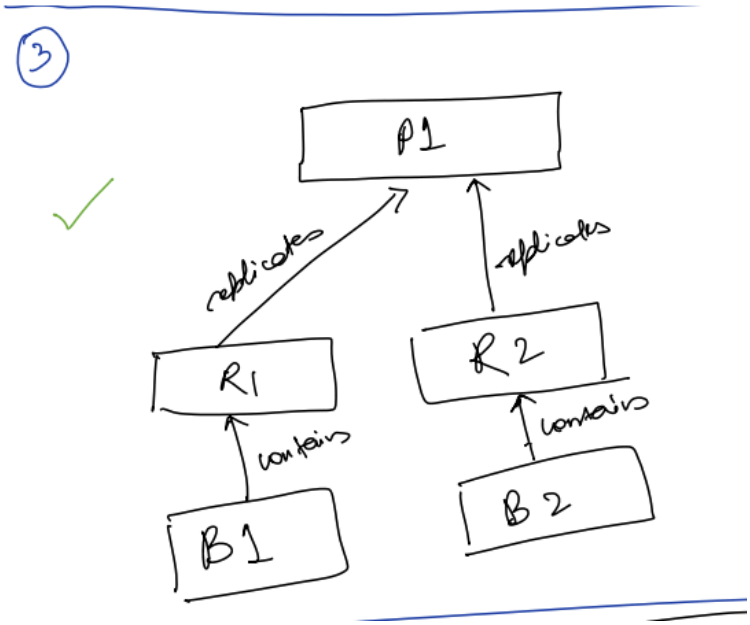
Constraint 9

```
run constraint9 {
        #replicates >= 2
        #stores >= 2
}
```

Similar to constrinat8, but shows 3 replicas and brokers. Each replica is stored in a broker even though there's no constraint that requires all to be stored. All are stored in different brokers because of the global fact `replicaOfSamePartitionOnDiffBrokers`

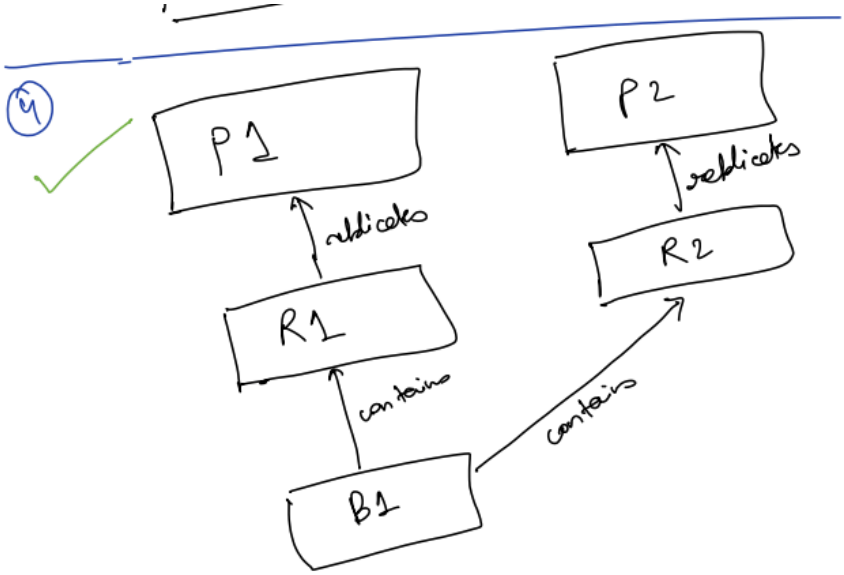


** Similar to Instance 3 in pairwise-notes.pdf



NOTE:

Instance 4 as mentioned in pairwise-notes.pdf was not seen in the above generated instances, where one broker contains multiple replicas of multiple partitions. It would be good to have such an generated instance to see how replicas of multiple partitions are stored.



Instance 2 was not seen because of global fact constraint: `replicaOfSamePartitionOnDiffBrokers`. This is expected.