Testing Rules Implementation

Rule 1

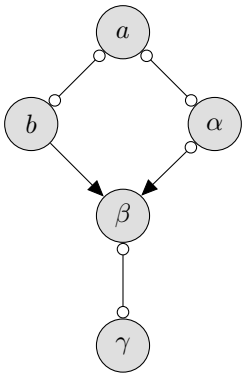

Figure 1: Testing Rule 1

We should see $\beta(3) \to \gamma(4)$ in the new graph.

```
library(LocalFCI)
adj.mat1 <- matrix(c(0,1,1,0,0,
                     1,0,0,2,0,
                     1,0,0,2,0,
                     0,3,1,0,1,
                     0,0,0,1,0),nrow = 5,byrow = TRUE)
adj.mat1.adj <- rule1(adj.mat1)

## Rule 1:
## Orient: 1 *-> 3 o-* 4 as 3 -> 4

cat(adj.mat1.adj$G[4,5]==2,"\n")

## TRUE

cat(adj.mat1.adj$G[5,4]==3,"\n")

## TRUE
```

Rule 2

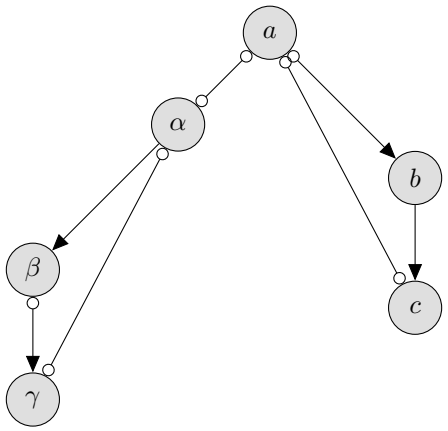

Figure 2: Testing Rule 2

We should see $\alpha(3)\circ\!\!\rightarrow \gamma(5)$ and $a(0)\circ\!\!\rightarrow c(2)$ in the new graph.

```
adj.mat2 <- matrix(c(0,2,1,1,0,0,
                     1,0,2,0,0,0,
                     1,3,0,0,0,0,
                     1,0,0,0,2,1,
                     0,0,0,3,0,2,
                     0,0,0,1,1,0),nrow = 6,byrow = TRUE)
res <- rule2(adj.mat2)

## Rule 2:
## Orient: 0 *-> 1 -> 2 as: 0 *-> 2
## Rule 2:
## Orient: 3 -> 4 *-> 5 as: 3 *-> 5

cat(res$G[1,3]==2,"\n")

## TRUE

cat(res$G[4,6]==2,"\n")

## TRUE
```

Rule 3

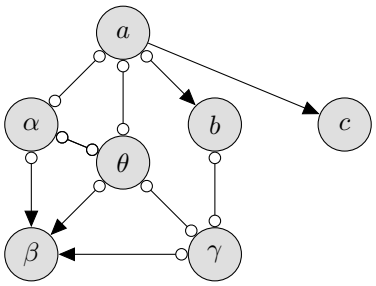

Figure 3: Testing Rule 3

We should see $\theta(6) \circ\!\!\rightarrow \beta(4)$

```
adj.mat3 <- matrix(c(0,2,2,1,0,0,1,
                     1,0,0,0,0,1,0,
                     3,0,0,0,0,0,0,
                     1,0,0,0,2,0,1,
                     0,0,0,1,0,1,1,
                     0,1,0,0,2,0,1,
                     1,0,0,1,1,1,0),nrow = 7,byrow = TRUE)
res <- rule3(adj.mat3)

## Rule 3:
## Orient: 6 *-> 4

cat(res$G[7,5]==2,"\n")

## TRUE
```

Rule 4

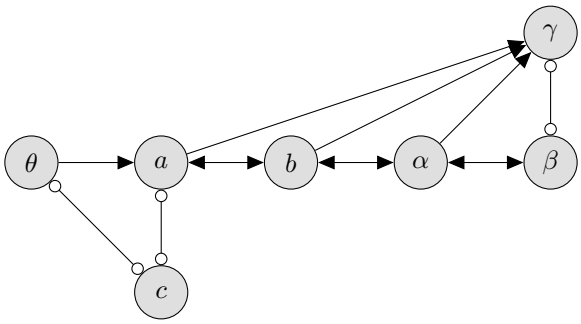

Figure 4: Testing Rule 3

If we let $\beta \in \text{SepSet}(\theta,\gamma)$, then we should have $\beta(4) \to \gamma(5)$.

```
adj.mat4 <- matrix(c(0,2,1,0,0,2,3,
                     2,0,0,2,0,2,0,
                     3,0,0,0,3,0,1,
                     0,2,0,0,2,2,0,
                     0,0,2,2,0,1,0,
                     3,3,0,3,1,0,0,
                     2,0,1,0,0,0,0),nrow = 7,byrow = TRUE)
S <- create_conditioning_sets_efficient_cpp2(seq(0,6))
nodes <- c("a","b","c","alpha","beta","gamma","theta")
S[["6"]][["5"]] <- 4
S[["5"]][["6"]] <- 4

rule4(adj.mat4,S)

## Potential beta: 2 | Potential gamma: 0
## Potential beta: 2 | Potential gamma: 6
## Potential beta: 4 | Potential gamma: 5 | Potential alpha: 3
## Potential values: 1
## Creating path list
## New Path: 3 1
## mpath: 3 1
## Potential values for the path: 0
## Size of old path list: 1
## Size of new path list: 2
## Path 0: 3 1
## Path 1: 3 1 0
## mpath: 3 1 0
## Potential values for the path: 6
## Size of old path list: 2
```

```
## Size of new path list: 3
## Path 0: 3 1
## Path 1: 3 1 0
## Path 2: 3 1 0 6
## mpath: 3 1 0 6
## Minimum Discriminating Path: 6 0 1 3 4 5
## Checking separation...finished
##
## Rule4
## There is a discriminating path between6 and 5 for 4 and 4 is in the SepSet of 5 and 6
##
## Potential beta: 6 | Potential gamma: 2
```