Figure 1: Behaviour of the coalition agent $\langle a, g \rangle$ for the selective matching mechanism without approximation. `subgroups(g,min,max)` returns the subgroups of `g` with the size between `min` and `max`.
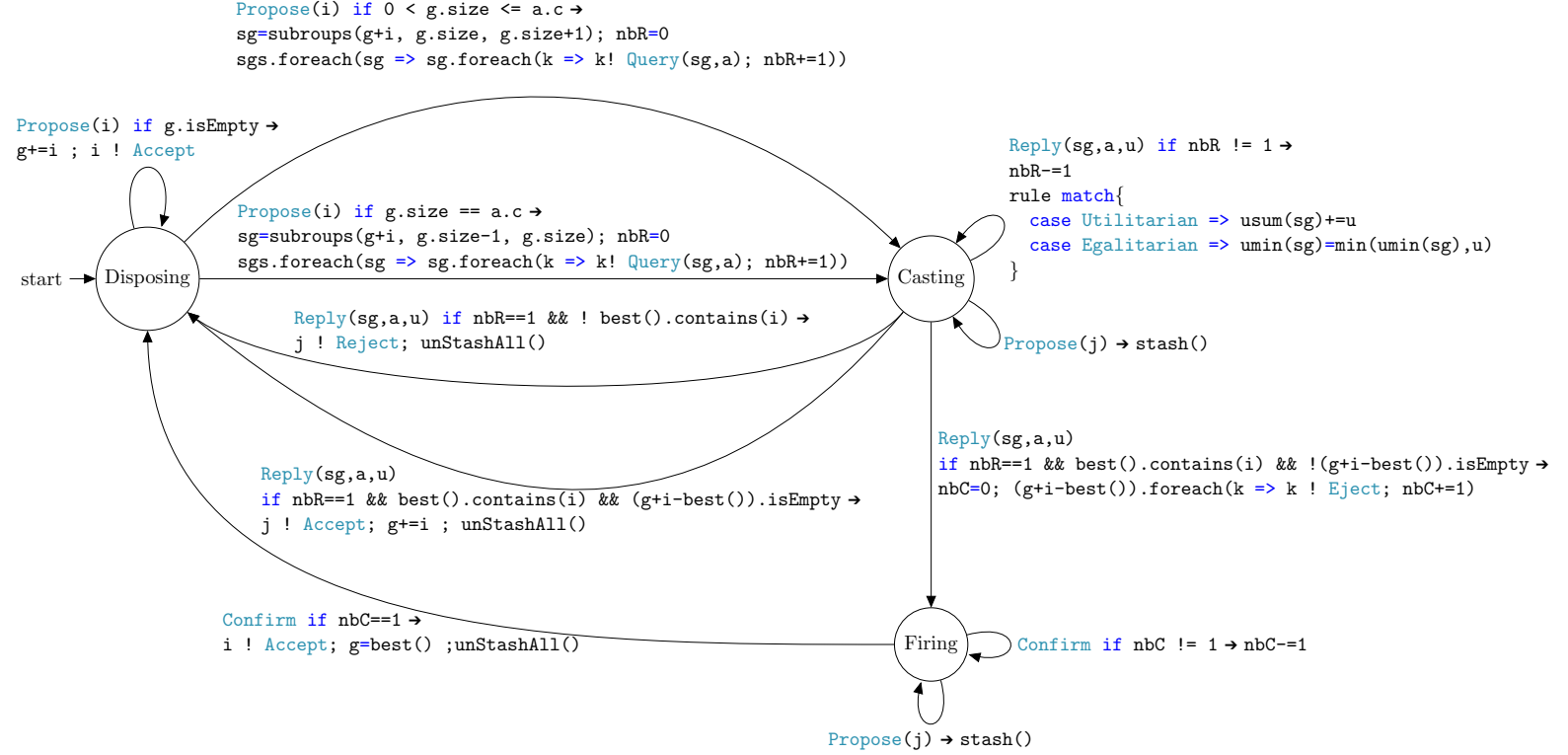
1

Propose(i) if 0 < g.size <= a.c →
sg=subroups(g+i, g.size, g.size+1); nbR=0
sgs.foreach(sg => sg.foreach(k => k! Query(sg,a); nbR+=1))

Propose(i) if g.isEmpty →
g+=i ; i ! Accept

Reply(sg,a,u) if nbR != 1 →
nbR-=1
rule match{
  case Utilitarian => usum(sg)+=u
  case Egalitarian => umin(sg)=min(umin(sg),u)
}

Propose(i) if g.size == a.c →
sg=subroups(g+i, g.size-1, g.size); nbR=0
sgs.foreach(sg => sg.foreach(k => k! Query(sg,a); nbR+=1))

start → Disposing

Casting

Propose(j) → stash()

Reply(sg,a,u) if nbR==1 && ! best().contains(i) →
j ! Reject; unStashAll()

Reply(sg,a,u)
if nbR==1 && best().contains(i) && !(g+i-best()).isEmpty →
nbC=0; (g+i-best()).foreach(k => k ! Eject; nbC+=1)

Reply(sg,a,u)
if nbR==1 && best().contains(i) && (g+i-best()).isEmpty →
j ! Accept; g+=i ; unStashAll()

Confirm if nbC==1 →
i ! Accept; g=best() ;unStashAll()

Firing   Confirm if nbC != 1 → nbC-=1

Propose(j) → stash()

Figure 2: Behaviour of the individual de agent i for the selective matching
mechanism with approximation which excludes no more than one individual
at each step. It distinguishes from Fig. due to the post-conditions of the
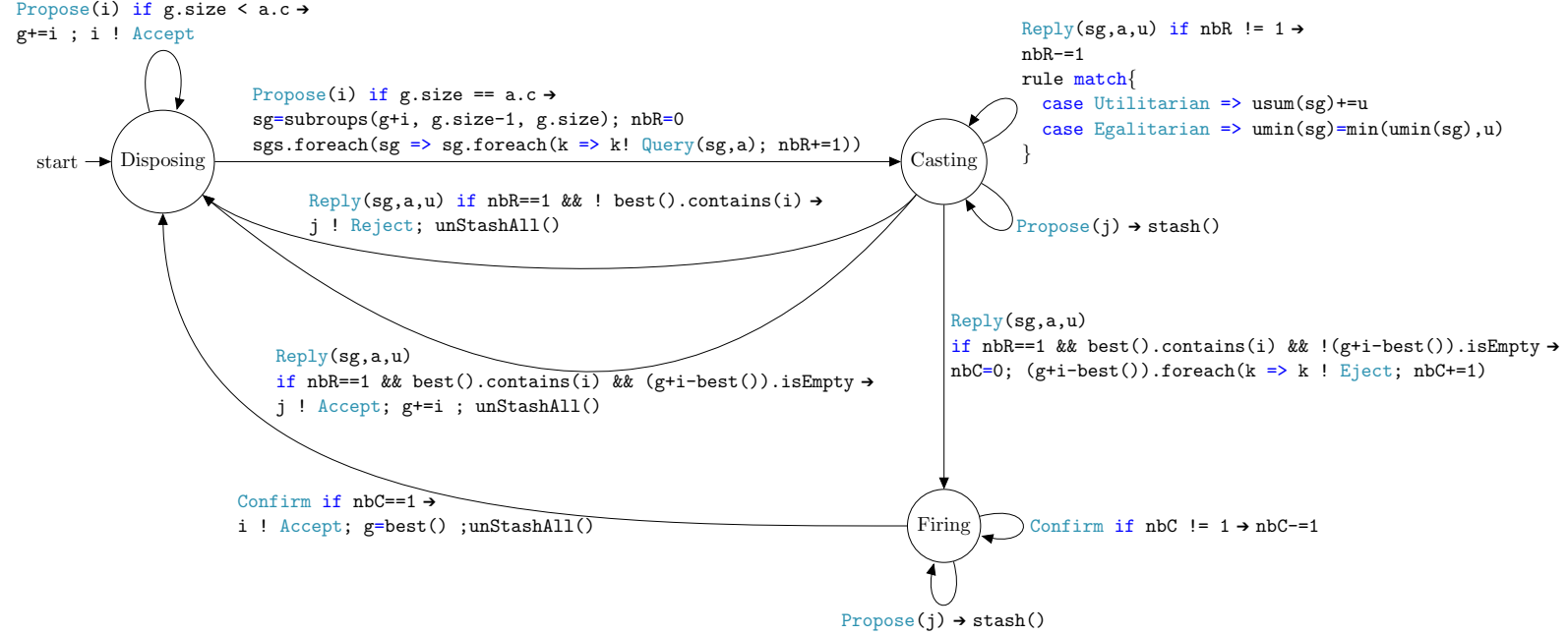outcome transition of the Disposing state.

Propose(i) if g.size < a.c →
g+=i ; i ! Accept

Propose(i) if g.size == a.c →
sg=subroups(g+i, g.size-1, g.size); nbR=0
sgs.foreach(sg => sg.foreach(k => k! Query(sg,a); nbR+=1))

Reply(sg,a,u) if nbR != 1 →
nbR-=1
rule match{
    case Utilitarian => usum(sg)+=u
    case Egalitarian => umin(sg)=min(umin(sg),u)
}

Propose(j) → stash()

start → Disposing   Casting

Reply(sg,a,u) if nbR==1 && ! best().contains(i) →
j ! Reject; unStashAll()

Reply(sg,a,u)
if nbR==1 && best().contains(i) && !(g+i-best()).isEmpty →
nbC=0; (g+i-best()).foreach(k => k ! Eject; nbC+=1)

Reply(sg,a,u)
if nbR==1 && best().contains(i) && (g+i-best()).isEmpty →
j ! Accept; g+=i ; unStashAll()

Confirm if nbC==1 →
i ! Accept; g=best() ;unStashAll()

Firing   Confirm if nbC != 1 → nbC-=1

Propose(j) → stash()

Figure 3: Behaviour of the individual de agent *i* for the inclusive matching making where the social decision rule is assessed only if its capacity constraint is violated and the exclusion of a single individual is envisaged for each individual integration. It distinguishes from Fig. and Fig. due to the post-conditions of the outcome transition of the Disposing state.