Nodes:

- 2*2*46[false]: cp=1, balance=(A + -D), p=0, d=D, cur_q=1, ID=H(basiccitizen)
- 3*3*33[true]: cp=1, balance=(A + (R + -D)), p=1, d=(D + -R)
- 3*4*34[false]: cp=1, balance=(A + (R + -D)), p=1, d=(D + -R), cur_q=q', ID=id'
- 4*4*37[false]: cp=0, balance=(A + (R + -D)), p=1, d=(D + -R), cur_q=1, ID=H(basiccitizen)
- 1*5*7[true]: cp=2, balance=(A + -D), p=2, d=(D + -(2 * R))
- 1*6*8[true]: cp=2, balance=(A + -D), p=2, d=(D + -(2 * R)), of=e, ID=gt_id
- 1*7*10[true]: cp=2, balance=(A + -D), p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e', ID'=gt_id'
- 1*8*11[true]: cp=2, balance=(A + -D), p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e', ID'=gt_id'
- 1*9*12[true]: cp=2, balance=(A + -D), p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e', ID'=gt_id'
- 1*10*13[false]: cp=2, balance=(A + -D), p=0, d=((d + -(2 * R)) + (max of of')), of=e, ID=gt_id, of'=e', ID'=gt_id'
- 1*11*15[true]: cp=2, balance=(A + -D), p=0, d=((d + -(2 * R)) + (max of of')), of=e, ID=gt_id, of'=e', ID'=gt_id'
- 1*1*1[true]: cp=2, balance=(A + -D), p=0, d=D
- 1*2*2[false]: cp=2, balance=(A + -D), p=0, d=D, cur_q=q, ID=id
- 1*3*4[true]: cp=2, balance=(A + -D), p=1, d=(D + -R)
- 1*4*5[false]: cp=2, balance=(A + -D), p=1, d=(D + -R), cur_q=q', ID=id'
- 2*4*20[false]: cp=1, balance=(A + -D), p=1, d=(D + -R), cur_q=1, ID=H(basiccitizen)
- 3*5*21[true]: cp=1, balance=(A + (R + -D)), p=2, d=(D + -(2 * R))
- 3*6*22[true]: cp=1, balance=(A + (R + -D)), p=2, d=(D + -(2 * R)), of=e, ID=gt_id
- 3*7*24[true]: cp=1, balance=(A + (R + -D)), p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e', ID'=gt_id'
- 3*8*25[true]: cp=1, balance=(A + (R + -D)), p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e', ID'=gt_id'
- 3*9*26[true]: cp=1, balance=(A + (R + -D)), p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e', ID'=gt_id'
- 3*10*27[false]: cp=1, balance=(A + (R + -D)), p=0, d=((d + -(2 * R)) + (max of of')), of=e, ID=gt_id, of'=e', ID'=gt_id'
- 3*11*29[true]: cp=1, balance=(A + (R + -D)), p=0, d=((d + -(2 * R)) + (max of of')), of=e, ID=gt_id, of'=e', ID'=gt_id'
- 3*1*30[true]: cp=1, balance=(A + (R + -D)), p=0, d=D
- 3*2*31[false]: cp=1, balance=(A + (R + -D)), p=0, d=D, cur_q=q, ID=id
- 4*2*40[false]: cp=0, balance=(A + (R + -D)), p=0, d=D, cur_q=1, ID=H(basiccitizen)

Edge labels:

- tau
- C(garbage_bin).dep[](q',id')
- C(garbage_bin):ID.NOK() (cur_q > 2)
- C(garbage_bin):ID.OK(R) (cur_q = 1)
- C(garbage_bin).bid[](e,gt_id)
- C(garbage_bin):ID.lost(of) ((2 * R) > of)
- C(garbage_bin).bid[](e',gt_id') (of >= (2 * R))
- C(garbage_bin):ID'.LOST(of') (of >= of')
- C(garbage_bin):ID.LOST(of) (of' > of)
- C(garbage_bin):ID.WIN() (of >= of')
- C(garbage_bin):ID'.WIN() (of' >= of)
- tau ((((of >= of') ∧ ~(id = ID)) ∨ ((of' > of) ∧ ~(id = ID'))))
- C(garbage_bin).empty[](id)
- C(garbage_bin):C(incinerator).notify(ID,2) (of >= of')
- C(garbage_bin):C(incinerator').notify(ID',2) (of >= of')
- C(garbage_bin):C(incinerator).save(((max of of') + -(2 * R)))
- C(garbage_bin):ID.OK((2 * R)) (cur_q = 2)
- C(garbage_bin).dep[](q,id)
- C(garbage_bin):ID.OK(R) (cur_q = 1)
- C(garbage_bin):ID.bid[](e,gt_id)
- C(garbage_bin):ID.WIN() (of >= of')
- C(garbage_bin):ID'.WIN() (of' >= of)
- tau (((of >= of') ∧ ~(id = ID)) ∨ ((of' > of) ∧ ~(id = ID')))
- C(garbage_bin):C(incinerator').notify(ID',2) (of' >= of)
- C(garbage_bin):C(incinerator).notify(ID,2) (of >= of')
- C(garbage_bin):C(incinerator).save(((max of of') + -(2 * R)))
- C(garbage_bin):ID.OK((2 * R)) (cur_q = 2)