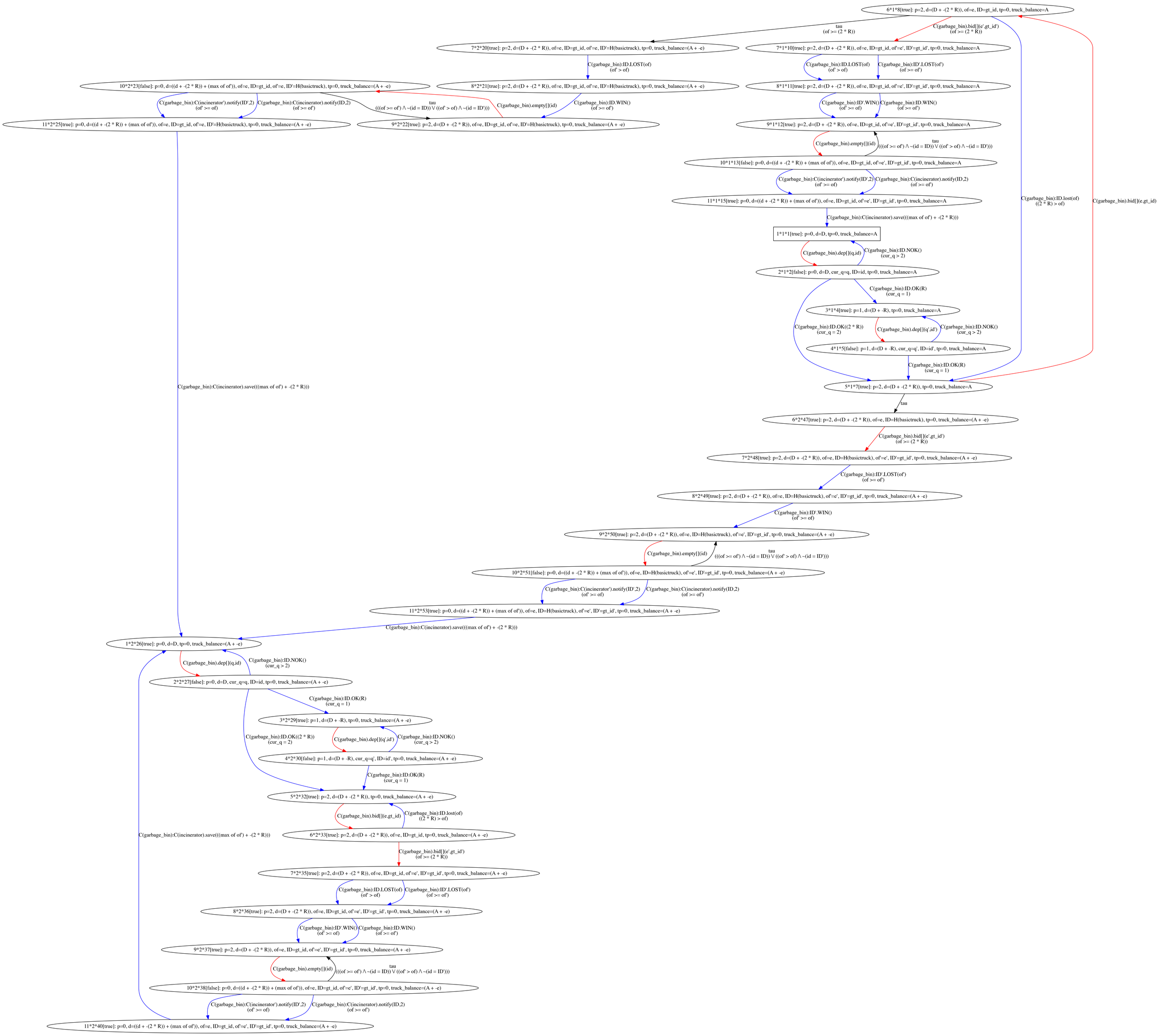
6*1*8[true]: p=2, d=(D + -(2 * R)), of=e, ID=gt_id, tp=0, truck_balance=A
tau
(of >= (2 * R))
C(garbage_bin).bid[](e',gt_id')
(of >= (2 * R))
7*2*20[true]: p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e, ID'=H(basictruck), tp=0, truck_balance=(A + -e)
7*1*10[true]: p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e', ID'=gt_id', tp=0, truck_balance=A
C(garbage_bin):ID.LOST(of)
(of' > of)
C(garbage_bin):ID'.LOST(of')
(of >= of')
10*2*23[false]: p=0, d=((d + -(2 * R)) + (max of of')), of=e, ID=gt_id, of'=e, ID'=H(basictruck), tp=0, truck_balance=(A + -e)
8*2*21[true]: p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e, ID'=H(basictruck), tp=0, truck_balance=(A + -e)
8*1*11[true]: p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e', ID'=gt_id', tp=0, truck_balance=A
C(garbage_bin):C(incinerator').notify(ID',2)
(of' >= of)
C(garbage_bin):C(incinerator).notify(ID,2)
(of >= of')
tau
(((of >= of') ∧ ~(id = ID)) ∨ ((of' > of) ∧ ~(id = ID')))
C(garbage_bin).empty[](id)
C(garbage_bin):ID'.WIN()
(of' >= of)
C(garbage_bin):ID.WIN()
(of >= of')
11*2*25[true]: p=0, d=((d + -(2 * R)) + (max of of')), of=e, ID=gt_id, of'=e, ID'=H(basictruck), tp=0, truck_balance=(A + -e)
9*2*22[true]: p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e, ID'=H(basictruck), tp=0, truck_balance=(A + -e)
9*1*12[true]: p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e', ID'=gt_id', tp=0, truck_balance=A
C(garbage_bin).empty[](id)
tau
(((of >= of') ∧ ~(id = ID)) ∨ ((of' > of) ∧ ~(id = ID')))
10*1*13[false]: p=0, d=((d + -(2 * R)) + (max of of')), of=e, ID=gt_id, of'=e', ID'=gt_id', tp=0, truck_balance=A
C(garbage_bin):C(incinerator').notify(ID',2)
(of' >= of)
C(garbage_bin):C(incinerator).notify(ID,2)
(of >= of')
11*1*15[true]: p=0, d=((d + -(2 * R)) + (max of of')), of=e, ID=gt_id, of'=e', ID'=gt_id', tp=0, truck_balance=A
C(garbage_bin):C(incinerator).save(((max of of') + -(2 * R)))
1*1*1[true]: p=0, d=D, tp=0, truck_balance=A
C(garbage_bin).dep[](q,id)
C(garbage_bin):ID.NOK()
(cur_q > 2)
2*1*2[false]: p=0, d=D, cur_q=q, ID=id, tp=0, truck_balance=A
C(garbage_bin):ID.OK(R)
(cur_q = 1)
3*1*4[true]: p=1, d=(D + -R), tp=0, truck_balance=A
C(garbage_bin):ID.OK((2 * R))
(cur_q = 2)
C(garbage_bin).dep[](q',id')
C(garbage_bin):ID.NOK()
(cur_q > 2)
4*1*5[false]: p=1, d=(D + -R), cur_q=q', ID=id', tp=0, truck_balance=A
C(garbage_bin):ID.OK(R)
(cur_q = 1)
5*1*7[true]: p=2, d=(D + -(2 * R)), tp=0, truck_balance=A
C(garbage_bin):ID.lost(of)
((2 * R) > of)
C(garbage_bin).bid[](e.gt_id)
tau
6*2*47[true]: p=2, d=(D + -(2 * R)), of=e, ID=H(basictruck), tp=0, truck_balance=(A + -e)
C(garbage_bin).bid[](e',gt_id')
(of >= (2 * R))
7*2*48[true]: p=2, d=(D + -(2 * R)), of=e, ID=H(basictruck), of'=e', ID'=gt_id', tp=0, truck_balance=(A + -e)
C(garbage_bin):ID'.LOST(of')
(of >= of')
8*2*49[true]: p=2, d=(D + -(2 * R)), of=e, ID=H(basictruck), of'=e', ID'=gt_id', tp=0, truck_balance=(A + -e)
C(garbage_bin):ID'.WIN()
(of' >= of)
9*2*50[true]: p=2, d=(D + -(2 * R)), of=e, ID=H(basictruck), of'=e', ID'=gt_id', tp=0, truck_balance=(A + -e)
C(garbage_bin).empty[](id)
tau
(((of >= of') ∧ ~(id = ID)) ∨ ((of' > of) ∧ ~(id = ID')))
10*2*51[false]: p=0, d=((d + -(2 * R)) + (max of of')), of=e, ID=H(basictruck), of'=e', ID'=gt_id', tp=0, truck_balance=(A + -e)
C(garbage_bin):C(incinerator').notify(ID',2)
(of' >= of)
C(garbage_bin):C(incinerator).notify(ID,2)
(of >= of')
11*2*53[true]: p=0, d=((d + -(2 * R)) + (max of of')), of=e, ID=H(basictruck), of'=e', ID'=gt_id', tp=0, truck_balance=(A + -e)
C(garbage_bin):C(incinerator).save(((max of of') + -(2 * R)))
1*2*26[true]: p=0, d=D, tp=0, truck_balance=(A + -e)
C(garbage_bin).dep[](q,id)
C(garbage_bin):ID.NOK()
(cur_q > 2)
2*2*27[false]: p=0, d=D, cur_q=q, ID=id, tp=0, truck_balance=(A + -e)
C(garbage_bin):ID.OK(R)
(cur_q = 1)
3*2*29[true]: p=1, d=(D + -R), tp=0, truck_balance=(A + -e)
C(garbage_bin):ID.OK((2 * R))
(cur_q = 2)
C(garbage_bin).dep[](q',id')
C(garbage_bin):ID.NOK()
(cur_q > 2)
4*2*30[false]: p=1, d=(D + -R), cur_q=q', ID=id', tp=0, truck_balance=(A + -e)
C(garbage_bin):ID.OK(R)
(cur_q = 1)
5*2*32[true]: p=2, d=(D + -(2 * R)), tp=0, truck_balance=(A + -e)
C(garbage_bin).bid[](e.gt_id)
C(garbage_bin):ID.lost(of)
((2 * R) > of)
C(garbage_bin):C(incinerator).save(((max of of') + -(2 * R)))
6*2*33[true]: p=2, d=(D + -(2 * R)), of=e, ID=gt_id, tp=0, truck_balance=(A + -e)
C(garbage_bin).bid[](e',gt_id')
(of >= (2 * R))
7*2*35[true]: p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e', ID'=gt_id', tp=0, truck_balance=(A + -e)
C(garbage_bin):ID.LOST(of)
(of' > of)
C(garbage_bin):ID'.LOST(of')
(of >= of')
8*2*36[true]: p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e', ID'=gt_id', tp=0, truck_balance=(A + -e)
C(garbage_bin):ID'.WIN()
(of' >= of)
C(garbage_bin):ID.WIN()
(of >= of')
9*2*37[true]: p=2, d=(D + -(2 * R)), of=e, ID=gt_id, of'=e', ID'=gt_id', tp=0, truck_balance=(A + -e)
C(garbage_bin).empty[](id)
tau
(((of >= of') ∧ ~(id = ID)) ∨ ((of' > of) ∧ ~(id = ID')))
10*2*38[false]: p=0, d=((d + -(2 * R)) + (max of of')), of=e, ID=gt_id, of'=e', ID'=gt_id', tp=0, truck_balance=(A + -e)
C(garbage_bin):C(incinerator').notify(ID',2)
(of' >= of)
C(garbage_bin):C(incinerator).notify(ID,2)
(of >= of')
11*2*40[true]: p=0, d=((d + -(2 * R)) + (max of of')), of=e, ID=gt_id, of'=e', ID'=gt_id', tp=0, truck_balance=(A + -e)