Jason实验报告

201300086 史浩男

一、代码运行方法

（Terminal中使用绝对路径）

家政机器人问题（1-2）：

```
❯ java -jar "D:\coding\jason\bin\jason" "C:\Users\Shawn\Desktop\NJUAI20\MAS\jason-AgentSpeak\code 1-2\domestic-robot\DomesticRobot.mas2j"
```

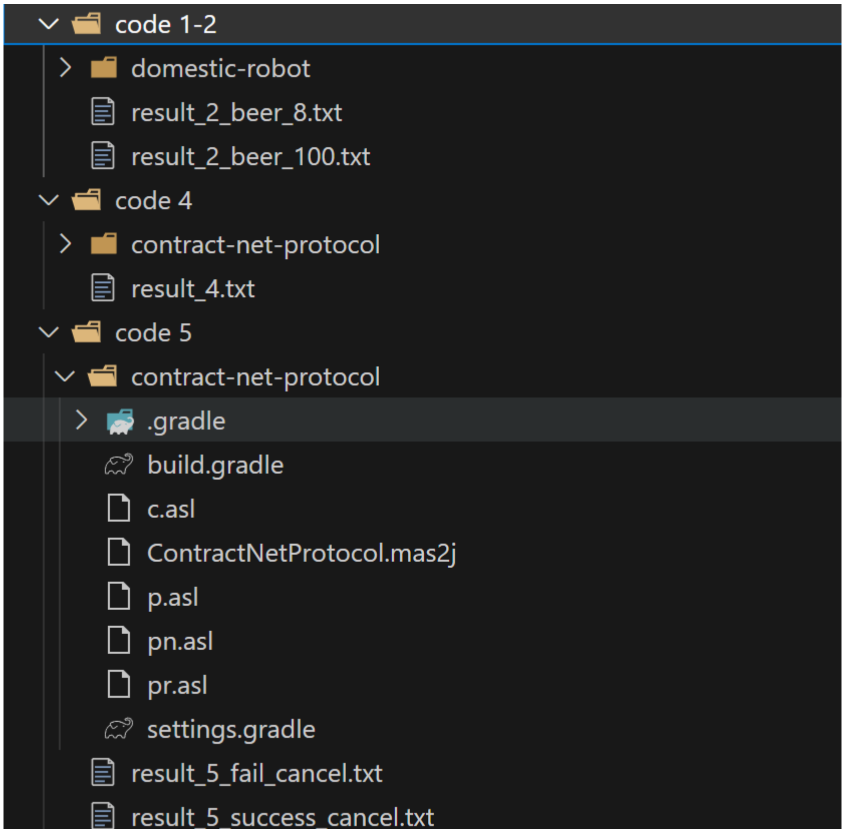
合同网撤销（4）：

```
❯ java -jar "D:\coding\jason\bin\jason" "C:\Users\Shawn\Desktop\NJUAI20\MAS\jason-AgentSpeak\code 4\contract-net-protocol\ContractNetProtocol.mas2j"
```

合同网取消（5）：

```
❯ java -jar "D:\coding\jason\bin\jason" "C:\Users\Shawn\Desktop\NJUAI20\MAS\jason-AgentSpeak\code 5\contract-net-protocol\ContractNetProtocol.mas2j"
```

二、代码结构



```
code 1-2
  domestic-robot
  result_2_beer_8.txt
  result_2_beer_100.txt
code 4
  contract-net-protocol
  result_4.txt
code 5
  contract-net-protocol
    .gradle
    build.gradle
    c.asl
    ContractNetProtocol.mas2j
    p.asl
    pn.asl
    pr.asl
    settings.gradle
  result_5_fail_cancel.txt
  result_5_success_cancel.txt
```

前两题、第四题、第五题代码分别在不同的文件夹中

- 其中初始库存为8或100的运行结果在`result_2_beer_8.txt`和`result_2_beer_100.txt`中
- 第四题运行结果在`result_4.txt`
- 第五题取消成功和失败的运行结果在`result_5_success_cancel.txt`和`result_5_fail_cancel.txt`

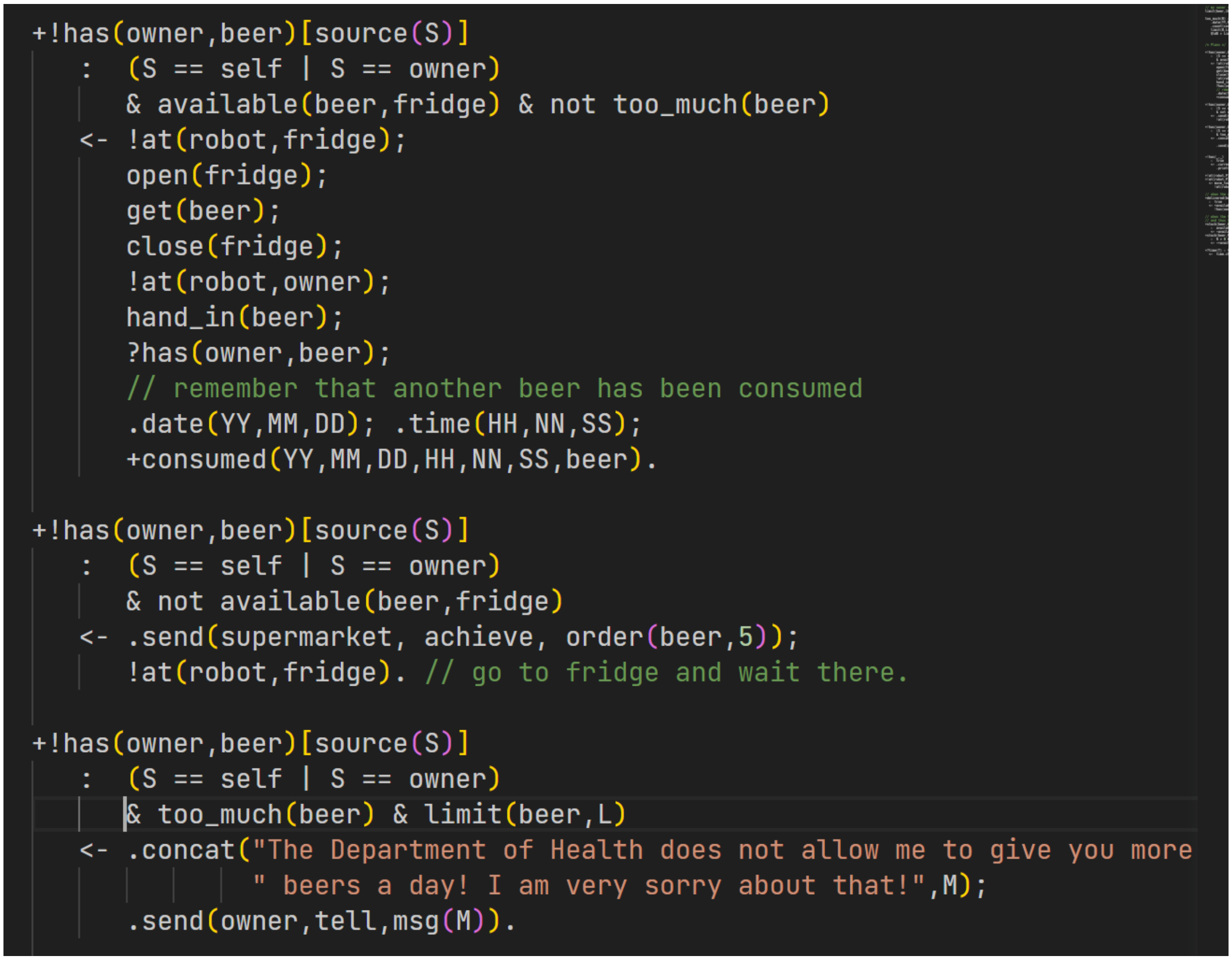
三、题目与解析

1、家政--添加确认目标来源的条件

修改内容：

在条件语句中都加上[source(S)]：(S == self | S == owner)

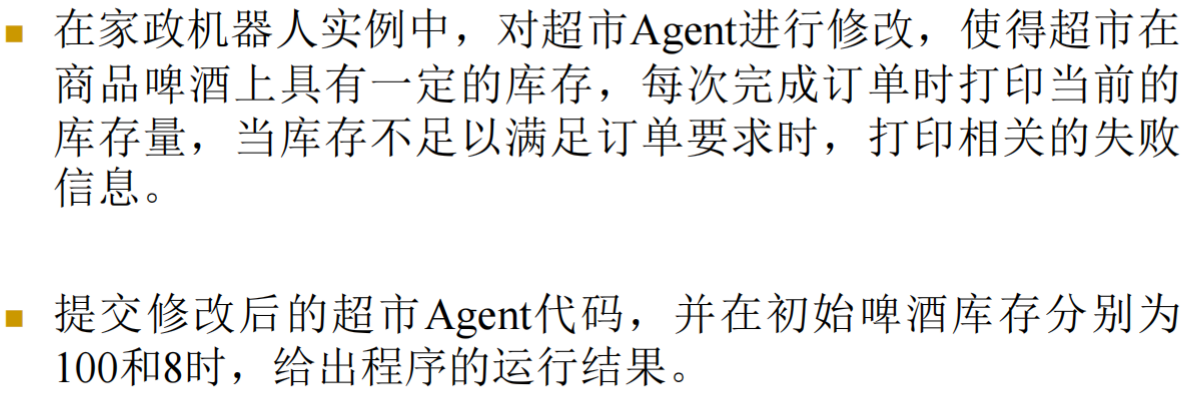
修改后代码：



```
+!has(owner,beer)[source(S)]
   :  (S == self | S == owner)
      & available(beer,fridge) & not too_much(beer)
   <- !at(robot,fridge);
      open(fridge);
      get(beer);
      close(fridge);
      !at(robot,owner);
      hand_in(beer);
      ?has(owner,beer);
      // remember that another beer has been consumed
      .date(YY,MM,DD); .time(HH,NN,SS);
      +consumed(YY,MM,DD,HH,NN,SS,beer).

+!has(owner,beer)[source(S)]
   :  (S == self | S == owner)
      & not available(beer,fridge)
   <- .send(supermarket, achieve, order(beer,5));
      !at(robot,fridge). // go to fridge and wait there.

+!has(owner,beer)[source(S)]
   :  (S == self | S == owner)
      & too_much(beer) & limit(beer,L)
   <- .concat("The Department of Health does not allow me to give you more
              " beers a day! I am very sorry about that!",M);
      .send(owner,tell,msg(M)).
```

这样修改可以防止其他攻击者等意外情况操纵Agent

2、家政--修改超市Agent



- 在家政机器人实例中，对超市Agent进行修改，使得超市在商品啤酒上具有一定的库存，每次完成订单时打印当前的库存量，当库存不足以满足订单要求时，打印相关的失败信息。

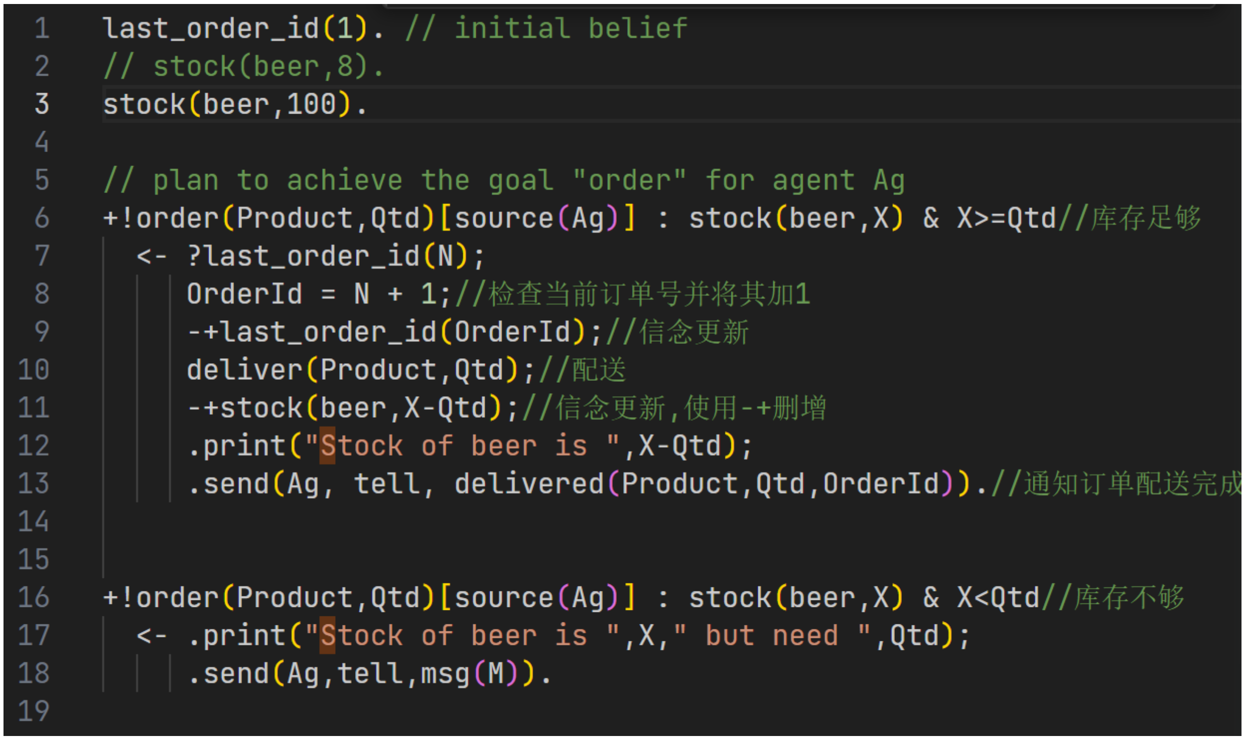
- 提交修改后的超市Agent代码，并在初始啤酒库存分别为100和8时，给出程序的运行结果。

初始库存为8时：

```
[supermarket] doing: deliver(beer,5)
[supermarket] Stock of beer is 3
[supermarket] Stock of beer is 3 but need 5
```

初始库存为100时：

```
[supermarket] doing: deliver(beer,5)
[supermarket] Stock of beer is 95
[supermarket] doing: deliver(beer,5)
[supermarket] Stock of beer is 90
[owner] Message from robot: The Department of Health does not allow me to give you
more than 10 beers a day! I am very sorry about that!
```

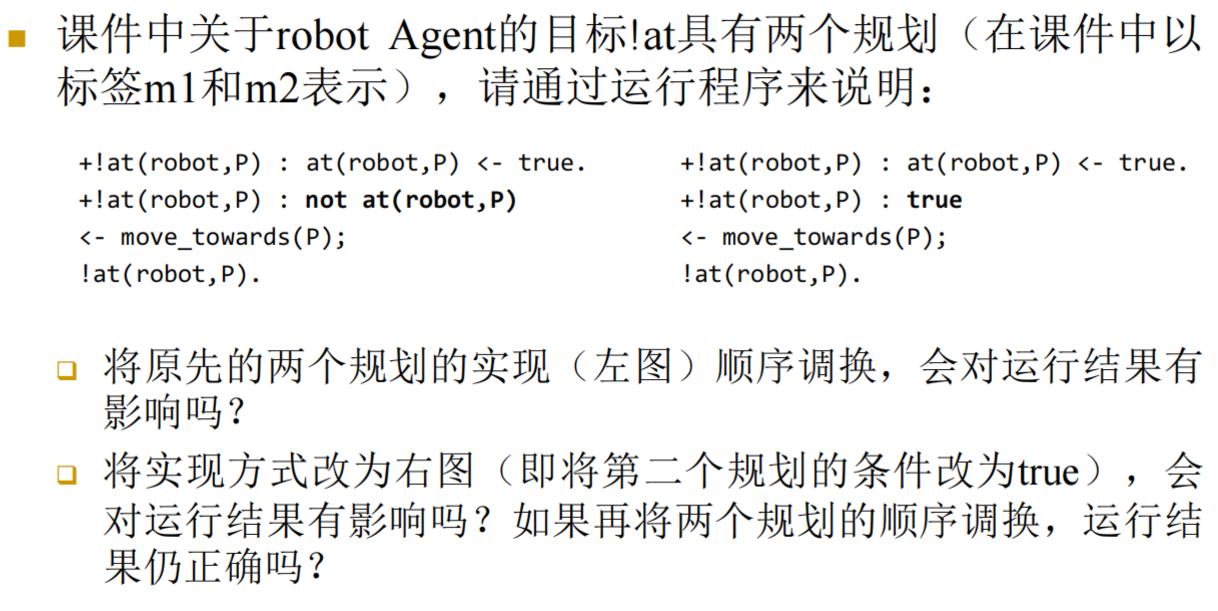


```
last_order_id(1). // initial belief
// stock(beer,8).
stock(beer,100).

// plan to achieve the goal "order" for agent Ag
+!order(Product,Qtd)[source(Ag)] : stock(beer,X) & X>=Qtd//库存足够
  <- ?last_order_id(N);
     OrderId = N + 1;//检查当前订单号并将其加1
     -+last_order_id(OrderId);//信念更新
     deliver(Product,Qtd);//配送
     -+stock(beer,X-Qtd);//信念更新,使用-+删增
     .print("Stock of beer is ",X-Qtd);
     .send(Ag, tell, delivered(Product,Qtd,OrderId)).//通知订单配送完成


+!order(Product,Qtd)[source(Ag)] : stock(beer,X) & X<Qtd//库存不够
  <- .print("Stock of beer is ",X," but need ",Qtd);
     .send(Ag,tell,msg(M)).
```

修改内容：

- 记录库存中beer数量
- 更新信念，采用-+stock的方式，一键增删
- 增加库存不够的判断
- 报错注意：每行结尾用；还是.要小心

3、家政--规划顺序调换



- 课件中关于robot Agent的目标!at具有两个规划（在课件中以标签m1和m2表示），请通过运行程序来说明：

```
+!at(robot,P) : at(robot,P) <- true.
+!at(robot,P) : not at(robot,P)
<- move_towards(P);
!at(robot,P).
```

```
+!at(robot,P) : at(robot,P) <- true.
+!at(robot,P) : true
<- move_towards(P);
!at(robot,P).
```

  - ❑ 将原先的两个规划的实现（左图）顺序调换，会对运行结果有影响吗？
  - ❑ 将实现方式改为右图（即将第二个规划的条件改为true），会对运行结果有影响吗？如果再将两个规划的顺序调换，运行结果仍正确吗？

(1)

没影响，因为这两条是并列的，互斥的（就像if和else语句一样），先执行哪个都一样。

(2)

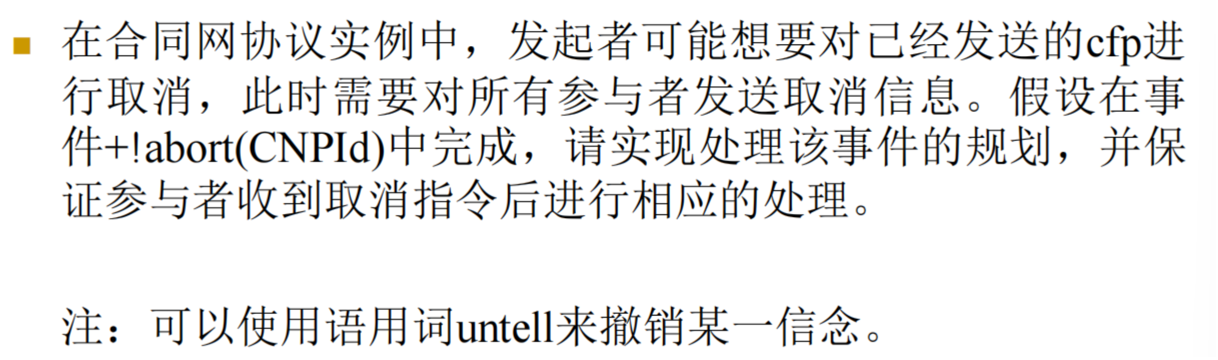
如果m1在前面，则没影响。

- 因为优先执行m1，优先级的限制使这部分程序总是有机会执行true并结束。而且执行m2时仍能达到移动到指定地点P的目的。

如果m2在前面，则有影响。

- 因为优先执行m2，程序会在m2陷入死循环

4、合同网--发送者取消



- 在合同网协议实例中，发起者可能想要对已经发送的cfp进行取消，此时需要对所有参与者发送取消信息。假设在事件+!abort(CNPId)中完成，请实现处理该事件的规划，并保证参与者收到取消指令后进行相应的处理。

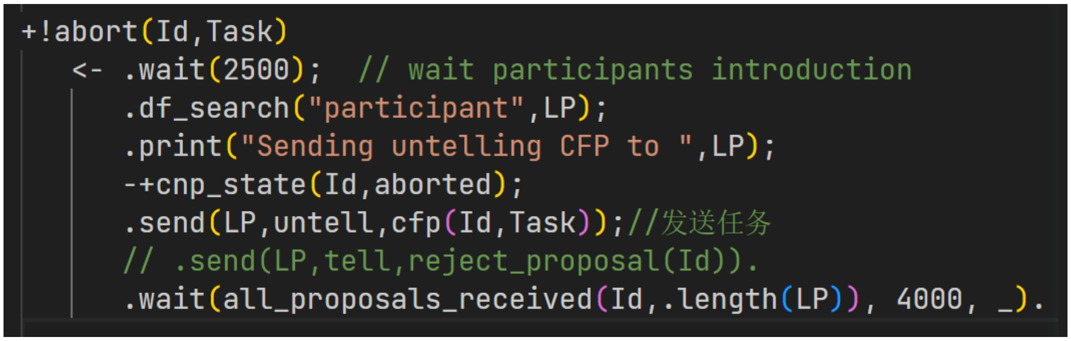
注：可以使用语用词untell来撤销某一信念。

我选择撤销第二个任务进行实验

```
!abort(2,banana).
```

增加事件：

- 增加`cnp_state(Id,aborted)`，用于记录任务状态

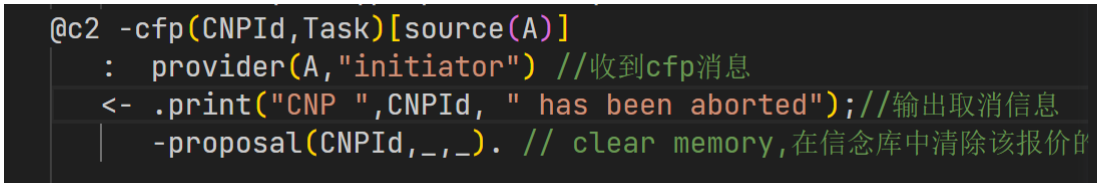


```
+!abort(Id,Task)
    <- .wait(2500);  // wait participants introduction
       .df_search("participant",LP);
       .print("Sending untelling CFP to ",LP);
       -+cnp_state(Id,aborted);
       .send(LP,untell,cfp(Id,Task));//发送任务
       // .send(LP,tell,reject_proposal(Id)).
       .wait(all_proposals_received(Id,.length(LP)), 4000, _).
```

增加应答：

untell对应的应答是减号-，不是加号+

此处传递的`cfp(Id,Task)`要注意一致，才能起到untell的目的



```
@c2 -cfp(CNPId,Task)[source(A)]
    :  provider(A,"initiator") //收到cfp消息
    <- .print("CNP ",CNPId, " has been aborted");//输出取消信息
       -proposal(CNPId,_,_). // clear memory,在信念库中清除该报价
```

输出

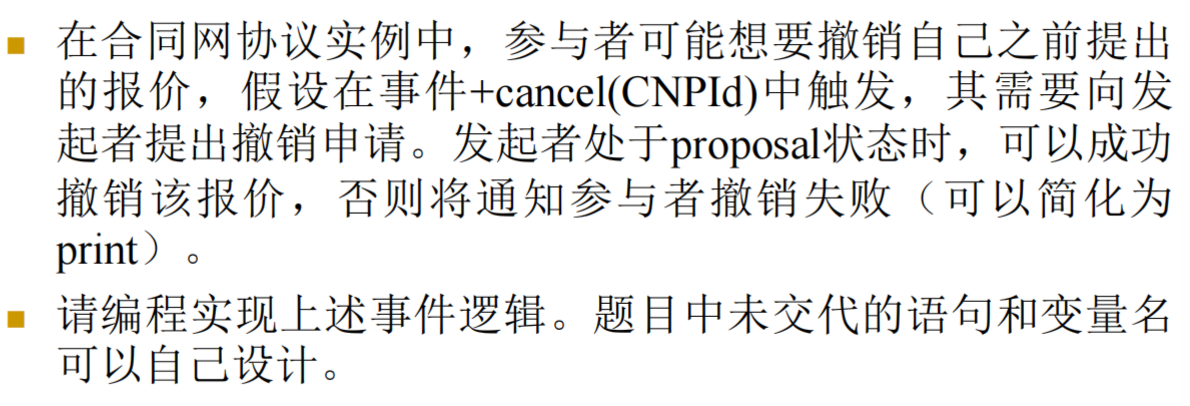
将任务二取消：

```
[c] Sending untelling CFP to [p1,pr,p2,p3,pn]
[p3] CNP 2 has been aborted
[p1] CNP 2 has been aborted
[p2] CNP 2 has been aborted
[c] Aborted 2 has no winner
```

任务一正常进行：

```
[c] Offers are
[offer(108.50486394141676,p1),offer(103.12538945976335,p3),offer(106.91108295940049,
p2)]
[c] Winner is p3 with 103.12538945976335
[p1] I lost CNP 1.
[p3] My proposal '103.12538945976335' won CNP 1 for fix(computer)!
[p2] I lost CNP 1.
```

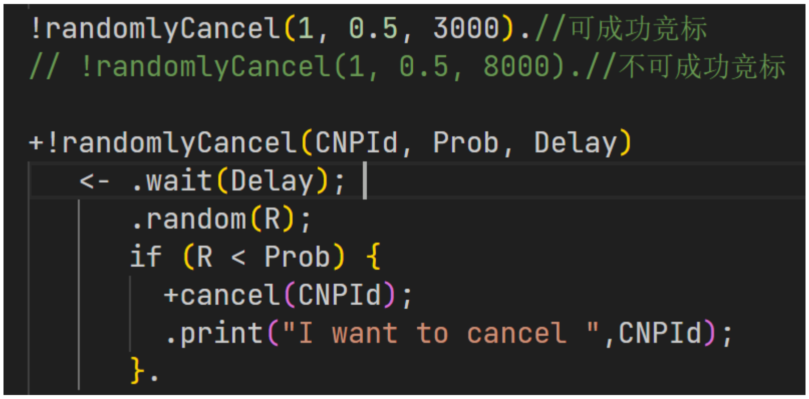
5、合同网--参与者撤销报价



- 在合同网协议实例中，参与者可能想要撤销自己之前提出的报价，假设在事件+cancel(CNPId)中触发，其需要向发起者提出撤销申请。发起者处于proposal状态时，可以成功撤销该报价，否则将通知参与者撤销失败（可以简化为print）。
- 请编程实现上述事件逻辑。题目中未交代的语句和变量名可以自己设计。

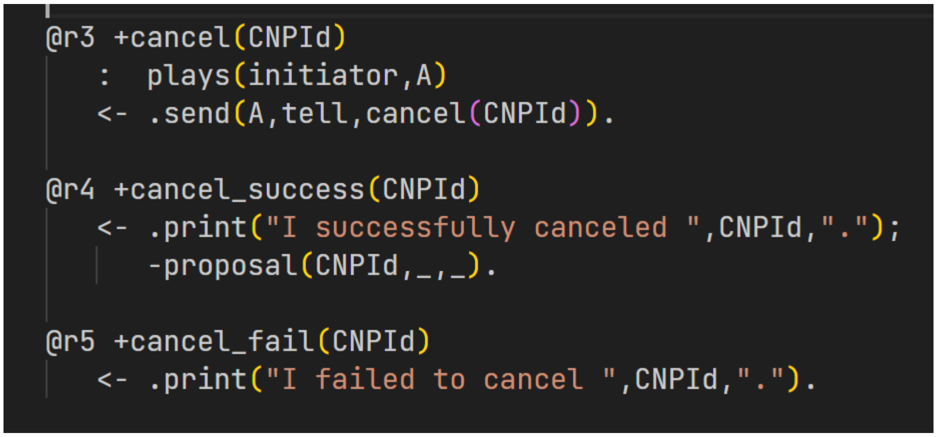
随机取消

每个参与者都有50%概率取消自己的报价



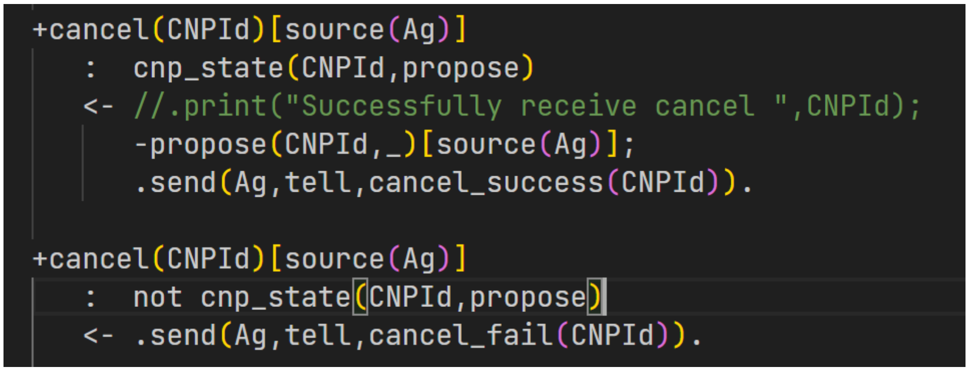
```
!randomlyCancel(1, 0.5, 3000).//可成功竞标
// !randomlyCancel(1, 0.5, 8000).//不可成功竞标

+!randomlyCancel(CNPId, Prob, Delay)
   <- .wait(Delay);
      .random(R);
      if (R < Prob) {
        +cancel(CNPId);
        .print("I want to cancel ",CNPId);
      }.
```

发送撤销申请并接收结果



```
@r3 +cancel(CNPId)
   :  plays(initiator,A)
   <- .send(A,tell,cancel(CNPId)).

@r4 +cancel_success(CNPId)
   <- .print("I successfully canceled ",CNPId,".");
      -proposal(CNPId,_,_).

@r5 +cancel_fail(CNPId)
   <- .print("I failed to cancel ",CNPId,".").
```

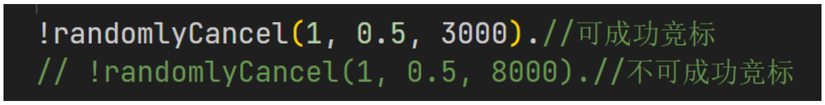
发起者接收消息



```
+cancel(CNPId)[source(Ag)]
   :  cnp_state(CNPId,propose)
   <- //.print("Successfully receive cancel ",CNPId);
      -propose(CNPId,_)[source(Ag)];
      .send(Ag,tell,cancel_success(CNPId)).

+cancel(CNPId)[source(Ag)]
   :  not cnp_state(CNPId,propose)
   <- .send(Ag,tell,cancel_fail(CNPId)).
```

结果输出1

将等待时间调整至8s，p2玩家提出cancel成功



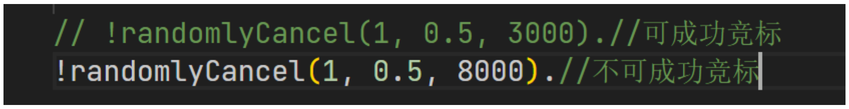
```
!randomlyCancel(1, 0.5, 3000).//可成功竞标
// !randomlyCancel(1, 0.5, 8000).//不可成功竞标
```

任务1的竞标情况如下：（观察到p2的竞标确实没有被收集到）

```
[p2] I want to cancel 1
[p2] I successfully canceled 1.
[c] Offers are [offer(109.23661992039372,p1),offer(103.01030872001407,p3)]
[c] Winner is p3 with 103.01030872001407
[p1] I lost CNP 1.
[p3] My proposal '103.01030872001407' won CNP 1 for fix(computer)!
```

结果输出2

将等待时间调整至8s，p2和p3玩家提出cancel失败



```
// !randomlyCancel(1, 0.5, 3000).//可成功竞标
!randomlyCancel(1, 0.5, 8000).//不可成功竞标
```

任务1的竞标情况如下：（正常进行，收集到了所有玩家的竞标）

```
[c] Offers are
[offer(109.04502303463701,p3),offer(100.72460935319542,p1),offer(102.7291853423984,p
2)]
[c] Winner is p1 with 100.72460935319542
[p3] I lost CNP 1.
[p2] I lost CNP 1.
[p1] My proposal '100.72460935319542' won CNP 1 for fix(computer)!
[p2] I want to cancel 1
[p3] I want to cancel 1
[p2] I failed to cancel 1.
[p3] I failed to cancel 1.
```