Tobacco Smokers' (TS) Problem

Three smokers sit around a table. Each has a permanent supply of precisely one of three resources, namely tobacco, cigarette papers, and matches, but is not permitted to give any of the three resources. The smoker who has the permanent supply of the remaining resource is then in a position to make and smoke a cigarette. On finishing the cigarette this smoker signals the agent, and the agent may then make again available a supply of some two resources.

The smokers are three threads, and the agent can be regarded as a set of three threads. As regards the latter, either none or exactly two of them run at 'anyone time. The problem is to have the six threads cooperate in such a way that deadlock is prevented, e.g., that when the agent supplies paper and matches, it is indeed the smoker with the supply of tobacco who gets both, instead of one or both of these resources being acquired by the other two smokers.

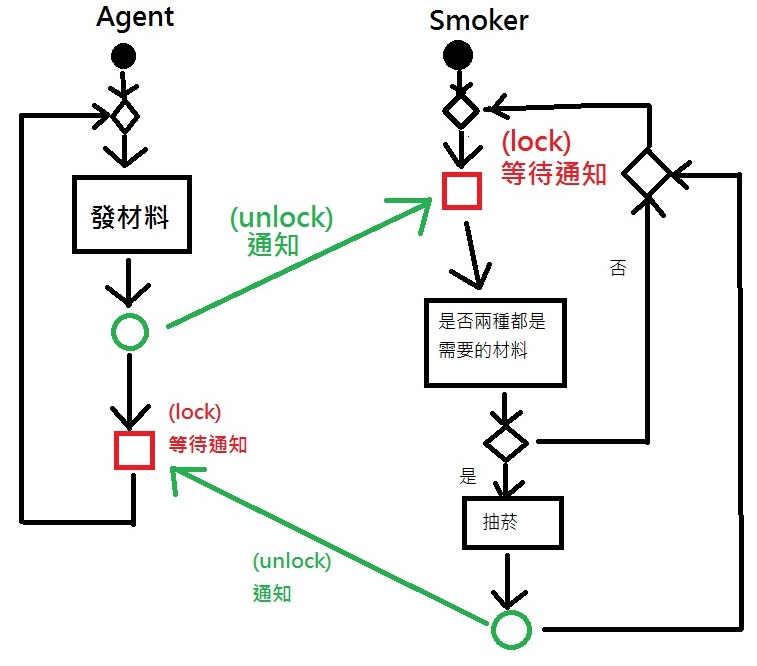
解決方式

三條thread代表三個抽菸者；三條thread代表三種agent發放材料的情況( tobacco、match；tobacco、paper；match、paper)。

0.　 一開始三條smoker threads會被lock住。

1. 利用亂數控制agnet thread，三條agent threads只會有一條發放材料，另外兩條不做任何事情。
2. 發放材料的agent thread執行完後會unlock三條smoker threads，並被lock住。
3. smoker thread確認場上兩種材料都是自己所需的時候，才能拿取；反之則什麼都不做，並被lock住。
4. 拿到材料的smoker thread抽菸，結束後unlock agent thread，並被lock住。
5. 回到第一步驟繼續執行。

流程圖



poison time的使用

放在smoker thread前面，以poison distribution亂數取得一個等待時間，模擬agent通知smoker後，三個smoker隨機到來的情況。

unsigned seed = chrono::system\_clock::now().time\_since\_epoch().count();

default\_random\_engine generator(seed);

poisson\_distribution<int> distribution(2.5);

int number = distribution(generator);

while(number>5||number<0){

number = distribution(generator);

}

this\_thread::sleep\_for(chrono::seconds(number));