使用Uppaal进行建模和分析

下面是使用模型检查器Uppaal进行建模和分析的练习。

Uppaal是免费下载的，非常易于安装和使用。可以做练习

短时间后由没有任何先验知识的高中或本科生

说明什么是状态图，以及演示基本内容的演示

Uppaal的功能。 （说明应涵盖Uppaal的符号？和！作为输入

并在通道上输出动作（死锁的概念），说明如何编辑状态

图，并显示模拟器和验证程序，包括验证程序如何

产生您可以在模拟器中逐步执行的反例。）

模型检查

模型检查是一种可用于设计和分析的技术

动态系统。模型检查器是一种快速运行的计算机程序

并巧妙地搜索系统的所有可能状态以寻找

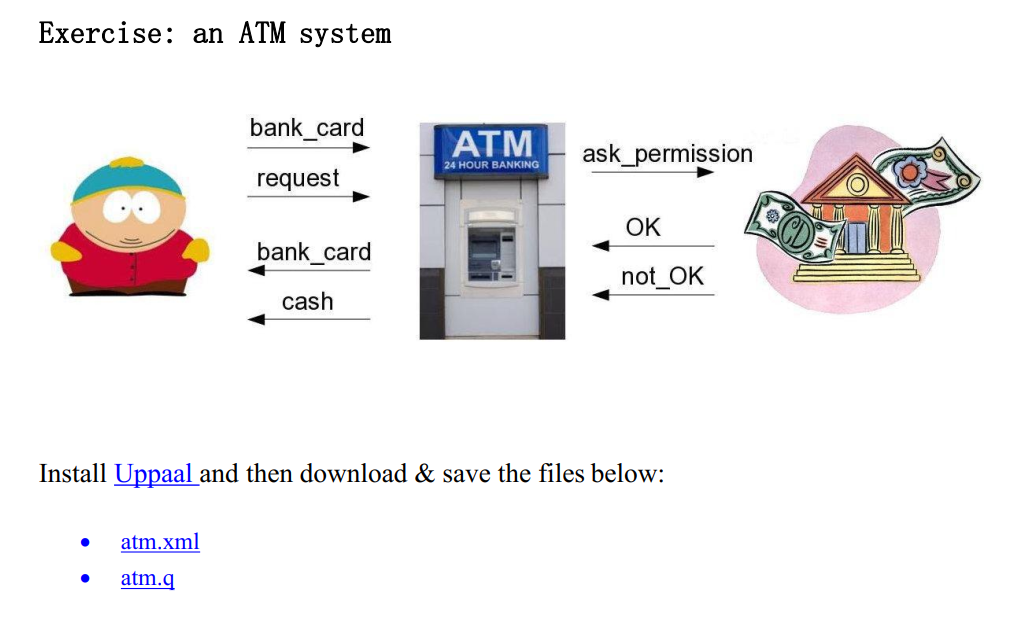
解决问题。实际上，您正在搜寻潜在问题。

模型检查器不适用于实际系统，但可以用于系统模型-

由此得名。 Uppaal使用状态图作为模型。除了图

描述我们要分析的系统，我们还需要描述一些期望的系统

属性。这些属性称为查询。



使用File：Open System在Uppaal：atm.xlm中打开这些文件，并使用File：Import Queries打开atm.q。

文件atm.xml提供了一个ATM（自动柜员机或现金点），

客户和银行。为简单起见，银行只有一台ATM和一名客户。那里

名为Eric的客户在他的电子钱包中携带Eric.cash\_in\_pocket欧元。自动取款机具有

ATM.in\_till欧元兑换。银行是银行的后端系统，并跟踪Eric的

Bank.balance中的余额。

埃里克（Eric）与自动取款机（ATM）互动，从机器中提取现金。 ATM反过来进行通信

在银行，以确保银行正确跟踪Eric银行的余额

帐户。

最初，Eric口袋里没有现金。您可以通过单击左侧的系统在Uppaal中进行检查

Eric，然后单击“声明”。埃里克（Eric）的银行帐户中有80欧元的余额，

而且ATM的收款机里有200欧元。

文件atm.q表示系统的两个属性：即Eric始终拥有80欧元

（因为该模型不包括他花钱的可能性），并且该系统

不应该陷入僵局。

问题1：使用验证程序检查属性。尝试改善模型，直到两者

属性是正确的。对于第一个物业，您必须更换银行。对于第二个属性

您必须更改Eric。 （如果ATM不支付现金但确实退款，Eric应该怎么做？

他的银行卡？）

提示：在菜单选项：诊断跟踪下，选择选项最短以使验证程序

在模拟器中生成反例。

问题2：更改自动取款机，使其最初只有30欧元，而不是200欧元。

自动柜员机模型中仍然存在一些不切实际的问题：

直到模型中可能发生某些事情，这在现实中是不可能的。尝试

弄清楚这是什么。您可以通过模拟模型一段时间或让Uppaal做到这一点

做一个随机模拟。 （提示：埃里克（Eric）想购物，需要大量现金。）

更改自动柜员机的型号以改善此情况。

完成此操作后，将一个查询添加到验证器以检查是否已避免了该问题。

查询应表示对ATM的明显的现实检查。当然，适应模型

仍然不应该陷入僵局！