Uppaal 4.0教程

2006年11月28日更新

Gerd Behrmann，Alexandre David和Kim G.Larsen

丹麦奥尔堡大学计算机科学系

{behrmann，adavid，kgl} @ cs.auc.dk。

抽象。这是有关工具Uppaal的教程文件。它的目标是

简要介绍了定时自动机的实现方式

工具，以展示其界面，并说明如何使用该工具。的

本文的贡献是提供参考实例和建模

模式。

1引言

Uppaal是用于验证由以下人员共同开发的实时系统的工具箱

乌普萨拉大学和奥尔堡大学。它已成功应用于

案例研究，从通信协议到多媒体应用

[35,55,24,23,34,43,54,44,30]。该工具旨在验证可以

建模为定时自动机网络，扩展了整数变量，结构化数据类型，用户定义的函数和通道同步。

Uppaal的第一版于1995年发布[52]。从那以后

在不断发展中[21,5,13,​​10,26,27]。实验和改进包括数据结构[53]，部分订单减少[20]，分布式版本

Uppaal [17,9]的指导和最低成本可达性[15,51,16]的工作

UML状态图[29]，加速技术[38]和新的数据结构

和内存减少[18,14]。版本4.0 [12]带来对称性降低[36]，

主流的扫描线方法[49]，新的抽象技术[11]，优先级[28]和用户定义的功能。 Uppaal还获得了相关博士学位。论文[50,57,45,56,19,25,32,8,31]。它具有Java用户

接口和用C ++编写的验证引擎。可在以下位置免费获得

http://www.uppaal.com/。

本教程介绍了定时自动机网络和定时的风格

第2节在Uppaal中使用了自动机。第3节介绍了工具本身，

第4、5和6节介绍了三个示例。最后，第7节

介绍了Uppaal经常使用的常见建模模式。

2 Uppaal中的定时自动机

模型检查器Uppaal基于定时自动机理论[4]（请参阅[42]

自动机理论）及其建模语言提供了其他功能，例如

作为有界整数变量和紧迫性。 Uppaal的查询语言用于指定要检查的属性，它是TCTL（定时计算树）的子集

逻辑）[39,3]。在本节中，我们介绍建模和查询语言

，我们对定时自动机中的时间进行了直观的解释。

2.1建模语言

定时自动机网络定时自动机是一种有限状态机

扩展了时钟变量。它使用密集时间模型，其中时钟变量

计算为实数。所有时钟同步进行。在Uppaal，

一个系统被建模为多个并行的此类定时自动机的网络。

该模型进一步扩展了有界离散变量，这些变量是

状态。这些变量在编程语言中的用法如下：

书面的，并且要接受常见的算术运算。系统状态

由所有自动机的位置，时钟值和

离散变量。每个自动机可能会激发优势（有时会误导

称为过渡）或与另一个自动机1同步，

导致一个新的状态。

图1（a）显示了一个定时自动机对简单灯的建模。灯

具有三个位置：关闭，低和明亮。如果用户按下按钮，即

与按？同步，然后指示灯点亮。如果用户按下

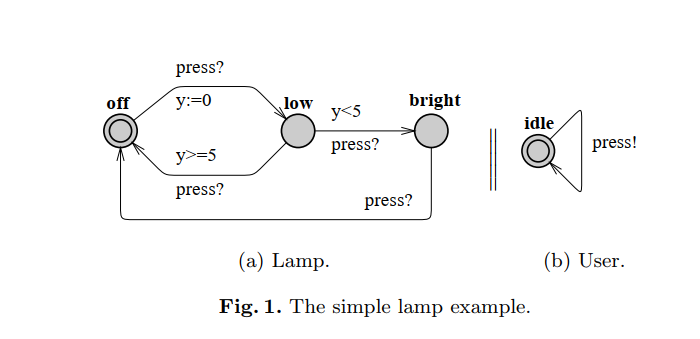
再按一次按钮，指示灯熄灭。但是，如果用户快速快速

按下按钮两次，灯泡点亮并变亮。用户

模型如图1（b）所示。用户可以随时随意按下按钮

甚至根本不按按钮。灯泡的时钟y用于检测是否

用户使用快（y <5）或慢（y> = 5）。



我们给出了基本定时的语法和语义的基本定义

自动机。在下文中，我们将跳过定时自动机的更丰富风味

在Uppaal中受支持，即具有整数变量和紧急扩展名

和指定的位置。有关其他信息，请参考帮助

广播同步情况下的1个或多个自动机，定时的另一个扩展

Uppaal中的自动机。

工具内的2菜单。我们使用以下符号：C是一组时钟，

B（C）是x或c形式的简单条件下的合集。

x−y⊲⊳c，其中x，y∈C，c∈N和⊲⊳∈{<，≤，=，≥，>}。定时自动机是

有限有向图，上面注有条件，且条件为非负

实际价值的时钟。

定义1（定时自动机（TA））。定时自动机是一个元组

（L，l0，C，A，E，I），其中L是一组位置，l0∈L是初始位置，

C是一组时钟，A是一组动作，协同动作和内部τ动作，

E⊆L×A×B（C）×2C×L是具有动作的位置之间的一组边，

一个警卫和一组要重置的时钟，并且I：L→B（C）将不变式分配给

位置。 </ s> </ s> </ s>

在图1的先前示例中，y：= 0是时钟y的复位，标签

按？然后按！表示动作-协作（此处为频道同步）。

现在，我们定义定时自动机的语义。时钟估值是

从时钟组到非负实数的函数u：C→R≥0。让RC

是所有时钟估值的集合。对于所有x∈C，令u0（x）=0。我们将滥用

通过将警卫和不变量视为时钟估值集来表示

u∈I（l）表示u满足I（l）

