基于事件集的反应系统模型的验证

张 磊[1]，马光胜[2]
(1. 黑龙江东方学院 计算机科学与电气工程学部，黑龙江 哈尔滨 150008；
2. 哈尔滨工程大学 计算机科学与技术学院，黑龙江 哈尔滨 150003)

摘 要：在总结前人工作的基础上，提出了一种有效检测并发或反应系统的动态行为模型中违反安全属性的方法，目的是减少为检测违反安全属性所需检测的状态数量，验证过程包括构造一个由所有独立状态图组成的全局状态空间图，并遍历这个全局状态空间图中的状态以便检测安全协议。首先读待验证的安全属性和可能会违反这些属性的相关事件集，构造全局状态空间图只考虑相关事件产生的状态转换。使用该方法验证了“火车道口”系统，减少了 59% 的搜索空间。

关键词：反应系统；安全属性；UML 状态空间图；相关事件集；状态转换
中图分类号：P315.69 **文献标识码**：B **文章编号**：1674-7720(2012)08-0013-03

Verification of model for reactive systems based-on event set

Zhang Lei[1], Ma Guangsheng[2]
(1. Institute of computer science and Electric Engineering, Heilongjiang East Academy, Harbin 150086, China;
2. Department of Computer and Technology, Harbin Engineering University, Harbin 150001, China)

Abstract: A new method to detecting effectively safety violation in dynamic behavior model is proposed in this paper based on the study of predecessors' work, which aim at reducing the number of states to be traversed for finding a property violation. The verification process involves building a global state space graph from these independent statechart. The authors applied the technology to the GRC(Generalized Railroad Crossing) systems and reduced space state by 59%。

Key words: reactive systems; safety property; UML state space diagram; relevant event set; state transition

本文在没有其他模型检测工具的情况下使用 UML 状态图验证已建模的反应系统。反应系统在这里认为是面向状态并对外部或内部行动做出反应，反应有可能产生状态或行为的变化，一个反应系统(事件驱动)的行为由一系列的状态、事件和行为集所规范。

1 提出的验证技术

1.1 假设

假定正在考虑中的系统有多个合作的对象，这些对象通过事件相互联系。每个对象的动态行为都用 UML 状态图建模。这些对象在接收一个正确的外部或内部产生事件及相应的保护条件变为真实状态发生改变。要验证的属性用时态逻辑表示并由符号 ϕ 代表。验证过程包括每个 UML 状态图到一个元组$\{S_i,E_i,T_i,I_i\}$形式的转换。其中 i 代表对象，S_i 代表非空有限的状态集，E_i 代表事件集，$T_i \subseteq S_i \times S_i$ 是一套转换集。$I_i \subseteq S_i$ 是一套初始状态集。让 E_t 为总的事件集即 $E_t=\{E_1 \cup E_2 \cdots E_n\}$，其中 n 是系统对象的数量[1]。

1.2 基于事件的验证方法

一旦在 E_t 中所有事件都发生时就合并所有对象的状态转换来建造系统的状态空间，在状态空间 (on the fly)的状态图中找到表示为$\neg \varphi$ 的错误状态(否定行为)，如果终止了更深层的状态空间的搜索，就会演示出错误轨迹(反例)。

本节中描述的算法[2]如下：

(1)一个事件是相关的如果：

①存在着一个与这个事件相关的转换并且当前状态是一个错误状态($\neg \varphi$)

②存在于一个与这个事件相关的转换并且下一个状态为一个错误状态($\neg \varphi$)

(2)一套事件是相关的如果：

存在着一套与这些事件相关的转换，并且能将对象从最初状态转到错误状态($\neg \varphi$)。即将对象从一个对

象的最初状态转到错误状态的事件集合叫做相关的事件集。

相关事件集计算完，每个对象的 UML 状态图都转换成这种元组$\{S_i, Er_i, T_i, I_i\}$，其中Er_i代表联系对象O_i的相关事件集，Er_t代表总的相关事件集，即$Er_t=\{Er_1 \cup Er_2 \cup \cdots Er_n\}$。搜索状态空间只考虑在总的相关事件集$Er_t$中的事件。一旦到达了错误状态或访问了所有的状态，就终止搜索状态图。

2 实例研究

(1)火车道口问题是实时系统中的一个典型问题。火车道口系统用来操纵道口的栏杆。对于两个铁轨上的门位于区域 A 上，火车在两个铁轨(T_1,T_2)上任意方向运行[3]。图 1 中显示了已经定位的传感器(S_1,S_2,S_3,S_4 和 S_5)。传感器表明当火车行驶到区域 A、进入 RC、离开区域 A 或退出 RC。传感器 S5 表明门是关着的还是开着的。“占用期间”指 RC 上有一个或更多火车的时期。系统被期望满足如下的属性：

①道口在所有占用期间是关闭的(安全)；

②如果占用期间没有火车，道口是开放的(实用性)；

③道口在尽可能的时间里是开放的(活性)。

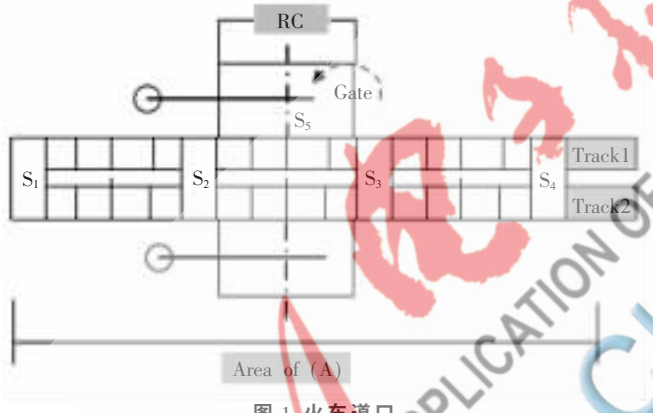

图 1 火车道口

(2)GRC 的 UML 状态图模型

对象道口栏杆和铁轨的动态行为用 UML 状态图规范化了，如图 2 所示，栏杆的 UML 状态图显示了一个最初状态和 4 个简单状态，即开着、正关闭、关闭和正打开。栏杆通过正打开和正关闭对外界信号做出反应[4]。每个正交区域都有一个最初状态和 5 个简单状态。

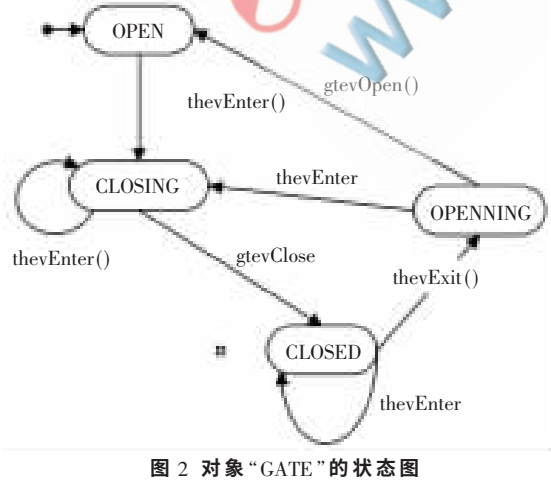

图 2 对象“GATE”的状态图

2.1 状态空间的构建

对象铁轨有两个正交状态 Track1 和 Track2。对象栏杆有 4 个局部状态，Track1 有 5 个局部状态，Track2 有 5 个局部状态。GRC 系统的 U 包括(4×5×5)100 个状态。通常模型限定可到达状态的数量。表 1 显示了所有可能的状态。

表 1 所有可能的状态

S1.NO.	Gate status	Track1 status	Track2 status
S1.	Open	Notrain	Notrain
S2.	Open	Notrain	Approaching
S3.	Open	Notrain	Crossing
S4.	Open	Notrain	Leaving
S5.	Open	Approaching	Notrain
S6.	Open	Approaching	Approaching
S7.	Open	Approaching	Crossing
S8.	Open	Approaching	Leaving
S9.	Open	Crossing	Notrain
S10.	Open	Crossing	Approaching
…	…	…	…
S40.	Opening	Crossing	Notrain
S41.	Opening	Crossing	Approaching
S42.	Opening	Crossing	Leaving
S43.	Opening	Leaving	Notrain
S44.	Opening	Leaving	Approaching
S45.	Opening	Leaving	Crossing
S46.	Opening	Leaving	Leaving

2.2 基于事件的算法应用到火车道口系统

在 GRC 模型中要检测的安全属性“当火车在 Track1 或 Track2 上的 RC 时，道口始终关闭”[5]，时态逻辑表示为：(T1.Crossing ∨ T2.Crossing)⇒G.Closed，如果成立则此模型有漏洞，产生一个反例/错误轨迹[3]，否定形式表示为：(T1.Crossing ∨ T2.Crossing)⇒¬ (G.Closed)，这就意味着火车通过时大门开着或正开着或正关闭的状态中。

图 3 中，状态搜索从最初状态 S_1 开始，由事件“tkevarrive”引发的后续状态为 S_2、S_5、S_6，随便选择状态 S_2 进行更深层的搜索[6]，直到到达状态 S_{15}，它不响应任何相关事件，所以回来遍历状态 S_{29} 后，会产生由事件“tkevexit”引发状态 S_{42}。状态 S_{42} 是一个错误状态，因为它违背了安全属性(即当一个火车经过道口时，大门是开着的)，一旦状态搜索终止，反例和错误轨迹就能产生(如图 4)，产生反例的路径长度为 6。同理如果遍历 S_{29} 后又遍历 S_{45}，也会违背安全属性，也会产生路径长度为 6 的反例。

3 结果及讨论

3.1 GRC 模型的改进

图 4 中的错误轨迹描绘出栏杆打开时，当一个火车

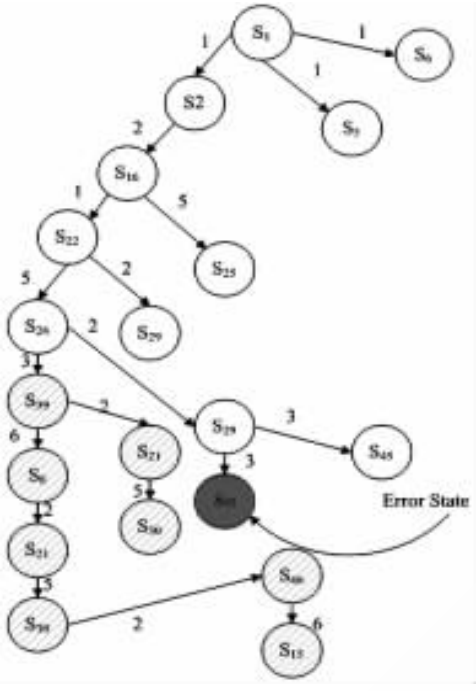

图 3 搜索状态空间

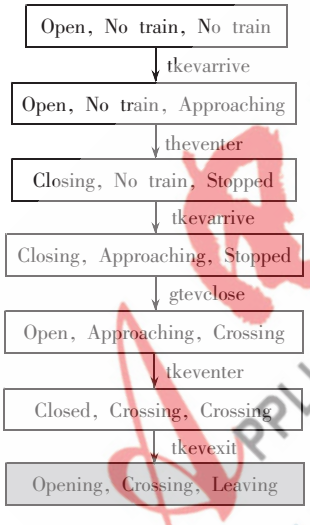

图 4 反例

穿过 RC 时就会导致错误的状态，模型中的这个漏洞可以通过保证占用期间没有火车后才允许栏杆打开的情况下予以避免[7]，对象栏杆的正确的 UML 状态图。将全局变量“train count”加进了模型中，它当每次火车进入道口时递增，火车离开道口时递减。

3.2 算法的执行

通过在状态搜索期间减少遍历状态数量方面验证该算法，在带有 6 个状态的 GRC 的 UML 状态模型中，由每个对象的状态图组合成相关事件集构成的状态空间后，即状态空间中仅由 19 个状态组成，在检测违反安全属性方面，将基于相关事件的算法应用到 GRC 系统后，只搜索遍历整个状态空间的 41%，状态空间大大减少，并产生长度为 6 的反例（见表 2）。

表 2 算法的执行

状态空间	搜索状态数	错误路径长度
46	19	06

参考文献

[1] 周清雷，姬莉霞，王艳梅. 基于 UPPAAL 的实时系统模型验证[J]. 计算机应用，2004，24(09)：129-131.

[2] 李勇，李宣东，郑国梁. 实时系统时段性质的模型检验[J]. 计算机科学，2002，29(11)：165-167.

[3] 徐雨波，晏荣杰. 一种基于有限精度时间自动机的模型检测工具[J]. 计算机应用研究，2006(05)：121-125.

[4] LANGE E. The degree of realism of gis-based virtual landscapes：Implications for spatial planning[C]. In：D. Fritsch and R.S piller(eds)，Photogrammertric Week'99，Herbert Wichmann Verlag，Heidelberg，1999：367-374.

[5] HENZINGER T A，JHALA R，MAJUMDAR R，et al. Software verification with blast[C]. in Proc. of 10th SPIN Workshop on Model Checking Software(SPIN)，LNCS 2648. Springer-Verlag，2003：235-239.

[6] BEER I，BEN-DAVID S，EISNER C，et al. Rulebase-an industryoriented formal verification tool[C]. in Proc. of 33rd Design Automation Conference(DAC). Association for Computing Machinery，1996：655-660.

[7] MIKK E，LAKHNECH Y，HOLZMANN G，et al. Implementing statecharts in promela/spin[C]. in Proc. of 2nd IEEE workshop on industrial strength formal specification techniques WIFT'98，1998：90-101.

（收稿日期：2012-01-11）

作者简介：

张磊，女，1979 年生，硕士研究生，主要研究方向：模型检测。

马光胜，男，1949 年生，博士生导师，主要研究方向：计算机辅助设计。