

混成系统

操作符

实时系统

Timed RAISE项目

(25)

82-84

混成系统的描述和设计与 Timed RAISE 项目^{*}

Specification and Verification of Hybrid System and the Timed RAISE Project

李黎

TP316.2

TP311.5

(中国科技大学计算机科学技术系 合肥 230026)

Abstract Hybrid systems are real-time systems consisting of both continuous and discrete components. This paper briefly introduces hybrid systems and their design methods, gives an overview of existent research on semantic model, property specification, operational design and verification. RAISE is a formal method for specification, design and development of software systems. The Timed RAISE project aims to add real-time features to RAISE for design of hybrid systems. We give a description of this project.

Keywords Hybrid system, RAISE, Formal method

1 引言

混成系统是包含连续部件和离散部件的实时系统。其中连续部件是满足一定微分方程的物理设备,离散部件是运行程序的计算机及受程序控制的感应器和主动器。混成系统的典型例子有:机器人控制系统、计算机辅助制造设备、飞行控制系统等。

混成系统的设计一般分为连续的几个相关步骤。首先分析系统需求及环境中运行进程的特性,形式化地描述基本的物理模型,确定系统的整体描述。然后直接确定或通过控制论的方法综合出控制策略,并证明这一策略在所确定的物理模型下正确地实现整个系统。从控制策略可以得到系统各部件的形式化描述,从这些描述产生的高级程序被翻译为所选的计算机器语言。为了将计算机嵌入它所控制的系统,可能需要增加一些特定的硬件。为了保证系统的可靠性,需要验证所有的实现须满足它们的描述。我们主要关注软件部分的描述与验证。

混成系统要求同时满足逻辑正确性和时间正确性。从某种意义上讲,时间正确性可以看作是逻辑正确性在时间轴上的扩展。物理设备的时间行为只能被描述为连续的实值函数,而控制软件和硬件却使用时间行为的离散模型。所以,为了有效地分析、综合、描述与验证混成系统,需要一个时间行为的统一模型,它是连续和离散时间模型的混合体。

混成系统有两类重要的时间属性:基于时间点的

属性和基于时间区间的属性,前者仅依赖于个别的系统状态,而基于时间区间的属性依赖于整个时间区间的系统状态,如极限、微分、积分等。

2 语义模型

在分析混成系统时,由于其连续行为和离散行为的相互作用,出现了两种语义模型:采样语义和连续语义。在采样模型中,系统的行为由许多离散时间点上的系统状态(称为“快照”)组成;而在连续模型中,系统的行为由一个映射每个时间点到系统状态的函数给出。

在采样模型中,有限的时间区间中只有有限数量的快照,因此可能遗漏系统的某些关键事件。为了解决这一问题,有人提出了“重要事件”的概念,这些重要事件是系统必须采样的。但即使如此,我们也可能遗漏一些有趣的事件。

连续语义可以分为两类:一类是密集语义和超密集语义,另一类是连续区间语义。在密集和超密集语义中,快照代表离散时刻,而其中间的部分是由无数连续时刻组成的连续区间。在连续区间语义中没有离散时刻,所有的系统状态值都由一组分段连续的函数给出,密集语义和超密集语义的区别是,密集语义和采样语义一样,允许同一时间点上有一个快照,而超密集语义只允许同一时间点上有一个快照。

3 性质描述

时态逻辑是分析混成系统的最抽象方法。为了形

* 本文由国家自然科学基金69773025和博士点基金98035828支持

式化地描述混成系统及其部件的性质,出现了各种各样的时态逻辑。其中有:线性时间时态逻辑、实时逻辑、动作时态逻辑和区间逻辑。

线性时间时态逻辑使用了采样语义和超密集语义。在实时逻辑中,实时时钟被作为一种特殊类型的连续变量。动作时态逻辑没有固定的操作符来描述实时或混成系统,所有描述实时性质所需的操作符都由动作时态逻辑和普通的数学来定义。区间逻辑被用于描述并行系统,因为它提供的操作符能很容易地描述在一定环境下必须满足的性质。基于区间的逻辑的优点是它能自然地表达经过任意区间的发展和变化。而在基于时间点的逻辑中,为了表达同样的性质,常常需要引入额外的辅助变量来记录所关心区间开始点的状态。

最早的区间逻辑是用于硬件验证的区间时态逻辑,它使用有限区间和采样语义,其区间由“chop”操作符连接。其它的区间逻辑有:命题线性时间时态逻辑、命题时态区间逻辑、区间描述逻辑和将来区间逻辑。其中后两者使用子区间操作符而不是“chop”操作符。

持续时间演算^[1]是一种为描述和验证实时系统设计的实时区间逻辑。它使用“chop”操作符,其语义模型是连续区间语义。由于状态持续时间是实时系统行为的一个重要度量,持续时间演算引入一个持续时间操作符“ $\int$ ”来度量区间中一个命题为真的持续时间。例如,公式 $t \geq 60 \Rightarrow 20 \int \text{Leak} \leq t$ 表示,如果时间区间的长度至少是60个时间单位,那么布尔状态变量 Leak 在这一区间中为真的时间部分不能超过区间长度的1/20。

状态间距离是实时系统行为的另一个重要度量,对此在实时逻辑、度量时态逻辑、显式时钟时态逻辑和区间时态逻辑中都有深入的研究。然而,状态持续时间比状态间距离的表达能力更强,前者可以表达后者,反之却不行。

扩展的持续时间演算^[2]将持续时间演算扩展到可以用于混成系统。它允许描述在区间的左右端点上的状态值及其微分,这对于分析包含连续和离散状态的混成系统、描述基于时间点和基于时间区间的性质是非常有用的。持续时间演算的证明系统也被扩展到允许数学分析的结果直接用于逻辑中,这就提供了一个与传统控制理论的灵活接口。

另一个与扩展的持续时间演算类似的逻辑是混成时态逻辑,它允许无限的区间。它们都将持续时间操作符作为从微分导出的操作符,混成时态逻辑隐式地使用,而扩展的持续时间演算显式地定义这一操作符并给出它满足的公理集。

除了以上的时态逻辑以外,还有一些其它的方法

来形式化描述混成系统及其部件的性质,其中之一是断言演算,它将传统的霍尔逻辑加入时间特性,并修改其解释以描述非终止计算。这种方法不够抽象,更适用于具体的程序推导。

4 操作设计与验证

混成系统软件部件的操作设计通常使用时间过程代数,其中包括:实时通讯过程代数、时间过程的代数、时间通讯顺序过程、时序通讯系统演算、时序过程语言等,它们都是在原有的过程代数基础上引入时间构件。大多数时间过程代数的操作都有以下特点。

- 系统由彼此合作的顺序部件或过程构成,每个部件有一个语义上的时间变量作为它的状态之一。时间变量定义在适当的时间模型上,并有一个二元操作“+”,部件通过执行一些不花时间的原子动作,或者通过增加时间变量让时间前进来改变其状态。

- 只有当所有的部件时间变量同步增加,但必须取得所有部件的同意。

- 执行序列分为两步,在第一步中,部件可能独立地或彼此配合地执行任意长的有限数量动作序列。在第二步中,各部件彼此配合让时间前进。

为了验证软件部件的操作设计满足其性质描述,可以使用推导的方法,在证明系统中将过程代数的每一个操作与代表其所满足行为的谓词联系起来。对于有限状态的系统,可以将时间过程代数转换为带时间的状态转变系统,然后使用模型检查的方法。

5 Timed RAISE 项目

时态逻辑和时间过程代数能很好地描述时间性质和算法,但它们不支持数据结构和中大型系统的模块化构造,因此,我们需要象 Z、VDM、扩展的 ML 及 RAISE 描述语言这样的形式化语言。

5.1 RAISE

RAISE (Rigorous Approach to Industrial Software Engineering, 工业软件工程的严密方法) 不仅仅是一种形式化描述语言,还是一种形式化开发方法和一个工具集合。RAISE 描述语言^[1]是软件系统的形式化描述与设计开发中最通用语言之一。RAISE 描述语言适用的范围很广,既能用于初始的、非常抽象的描述,也能用于表示适合于转换为程序语言的、抽象级较低的设计,因而,RAISE 方法^[4]包含:

- 确定抽象描述
- 由抽象描述相继产生更具体的描述
- 证明产生的正确性
- 将最后的描述转换为程序语言

使用 RAISE 的优点是,从系统描述到设计的大部

分构造都能用同一种形式化方法,这有助于对产生步骤的正确性和其它重要性质的准确分析。RAISE 描述语言的结构化功能可以支持分解与重用。

RAISE 描述是软件系统的数学模型,不同开发层次的描述之间的区别在于其不同的抽象层次,RAISE 描述语言支持以下的描述类型:面向模型的描述;面向性质的描述;函数式描述;命令式描述;并行描述;隐式定义;显式定义。

RAISE 带有以下工具:

- 用于描述、产生关系、证明和证明约束的库
- 带语法和类型检查的编辑器
- 将 RAISE 描述语言转换为 Ada 和 C++ 的翻译工具

• 格式打印工具

• 证明工具

5.2 Timed RAISE 项目

RAISE 方法被证明对许多软件系统的工业开发是非常有用的。但是,RAISE 没有描述实时需求的特性,不能用于混成系统。给 RAISE 加入实时特性不仅是理论计算机科学研究者感兴趣的课题,也是工业界 RAISE 用户的需要。

持续时间演算是为描述和分析软件和时间硬件的特性而设计的。但是,它不支持数据结构,算法和中大型系统的模块化构造。

Timed RAISE 项目的目标就是结合 RAISE 和(扩展的)持续时间演算,以描述、设计和开发混成系统。这一项目是由联合国大学国际软件技术研究所和丹麦工业大学信息技术系提出的,作者在联合国大学国际软件技术研究所期间,参与了这一项目。这一项目包括三个方面:

• RAISE 描述语言的扩展:需要在 RAISE 描述语言中加入描述时间的特性。当前它包括一个表示时间的类型及一个描述延迟的等待表达式。需要定义这些扩展的静态和动态语义。扩展后的语言称为 Timed RAISE 描述语言。

• 结合 Timed RAISE 描述语言与持续时间演算:

需要定义 Timed RAISE 描述语言与持续时间演算之间的关系。一种办法是用 Timed RAISE 语言来定义持续时间演算,但这种办法使得对持续时间演算的公式推导比较复杂。另一种办法是用持续时间演算来定义 Timed RAISE 描述语言的语义。这只需要是表达其时间特性的部分语义,因为其它部分可以使用现有的 RAISE 描述语言证明理论。当前采用的就是这种方法。

• 方法的结合:需要将 RAISE 和持续时间演算的开发方法结合起来,以支持 Timed RAISE 描述的生成。结合后方法的有效性和易用性将证明前两个方面所作出的选择的正确性。

目前,已经定义了 Timed RAISE 描述语言的语法和操作语义^[5]、指称语义^[6]。

结束语 混成系统是应用非常广泛的实时系统,为了增强其可靠性和易维护性,采用了许多形式化方法。Timed RAISE 项目结合工业界使用的形式化方法中较通用的 RAISE 方法与实时系统中比较流行的持续时间演算,对于混成系统的描述、设计与开发有很大的意义。

参考文献

- 1 Zhou Chaochen, et al. A calculus of durations. Information Processing Letters, 1991, 40(5): 269~276
- 2 Zhou Chaochen, et al. An extended duration calculus for hybrid systems. In: Grossman R L, et al. eds. Hybrid Systems, volume 736 of LNCS, Springer-Verlag, 1993: 36~59
- 3 The RAISE Language Group. The RAISE Specification Language. Prentice Hall International(UK)Ltd., 1992
- 4 The RAISE Method Group. The RAISE Development Method. Prentice Hall International(UK)Ltd., 1992
- 5 George C, et al. An operational semantics for Timed RAISE. [Research Report 149] UNU/IIST, P.O. Box 3058, Macau, Nov. 1998
- 6 Li Li, He Jifeng. A denotational semantics of Timed RSL using duration calculus. In: Proc. 6th Interl. Conf. on Real-Time Computing Systems and Applications, 1999

好機遇 好去處 環境優良 事業有成

北京大学志在建成世界一流大学

计算机科学技术系

誠邀優秀中青年學者加盟