函数极限的高阶逻辑形式化建模与验证

赵春娜 赵 刚
(云南大学信息学院 昆明 650500)

摘 要 在高阶逻辑定理证明器中研究了函数无穷远处极限的形式化建模和验证,包括函数无穷远处极限定义的形式化模型,函数极限相关性质的建模与验证,有唯一性、保不等式性、绝对值函数在无穷远处的极限、极限等价性、常函数极限等.函数无穷远处极限的高阶逻辑定义是利用拓扑极限方式定义的,并在实数域内利用集合关系等验证定理.根据集合有序关系定理验证了唯一性.利用差值和绝对值的高阶逻辑性质验证极限为零的属性.变量与常数之和与积的极限高阶逻辑定理也通过已验证定理和高阶逻辑策略验证了.建立了函数极限四则运算的高阶逻辑模型,并验证了函数极限加法、函数极限减法、函数极限乘法、函数极限与常数乘法、函数极限除法的高阶逻辑定理.也建立了函数积分极限的高阶逻辑形式化模型,验证了函数积分极限上限绝对值定理、函数积分极限上限可加定理、函数积分极限上限可乘定理.在此基础上,建立了拉普拉斯变换卷积定理的高阶逻辑形式化建模与验证.最后,对电阻-电感电路中的电流进行了高阶逻辑形式化建模与验证,建立了单位阶跃信号和电路中电流的高阶逻辑形式化定义,并验证了其正确性.该实例验证表明了函数极限和相关性质的高阶逻辑形式化模型的正确性,为后续控制系统的形式化分析奠定了良好的基础.

关键词 函数极限;高阶逻辑;形式化验证;定理证明;卷积
中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2020.02119

Formal Modeling and Verification of Function Limit in Higher-Order Logic

ZHAO Chun-Na ZHAO Gang
(*School of Information Science and Engineering, Yunnan University, Kunming* 650500)

Abstract Function limit of infinity is studied in higher order logic theorem prover in this paper. Formal modeling and verification of function limit are created in higher order logic. Higher order logic model of function limit plays a significant role in the formalization of fractional order systems. Firstly, based on higher order logic models of set and number, formal model of function limit definition and its related properties are proposed in this paper. These attributes include the basic nature of function limit—uniqueness, inequality preserving property, the absolute value function limit when solving its positive infinity limit, limit equivalence, constant function limit, etc. Higher order logic definition of function limit is defined by topological limit way. Then higher order logic theorem is verified based on sets and relations in the real domain. The uniqueness theorem is validated on account of the sets the orderly relations theorem. The inequality preserving property is verified through the higher order logic validation strategy. The zero limit theorem is tested by higher order logic properties of difference and absolute value. The limit equivalence and constant function limit theorems are verified in higher order logic. The limit of the sum of variable and constant and the limit of the product of variable and constant also are validated based on the higher order logic validation strategy and the verified theorems. Higher order logic model of arithmetic of function limit is also established in higher order logic theorem prover. And some related

收稿日期:2019-11-16;在线发布日期:2020-05-18.本课题得到国家自然科学基金(61862062,61104035)资助.**赵春娜**,博士,教授,主要研究领域为高阶逻辑形式化验证、分数阶系统.E-mail:chunnazhao@163.com.**赵 刚**,硕士,主要研究方向为高阶逻辑形式化验证.

theorems are verified. The function limit addition higher order logic axiom is validated on the basis of higher order logic model of function addition. The function limit subtraction higher order logic proposition is verified on the grounds of higher order logic model of function subtraction. The function limit multiplication higher order logic theorem is validated in the light of higher order logic model of function multiplication. Then higher order logic theorems of multiplication of function limit and constants and function limit division are verified in higher order logic theorem prover according to the verified theorems and some validation strategies. The higher order logic formal modeling and verification of function integral limit is also proposed in this paper. The upper limit absolute value of function integral limit theorem is verified on the grounds of some higher order logic attributes of function limit and absolute value. The upper limit addition theorem and upper limit multiplication theorem of function integral limit is verified on the grounds of some higher order logic validation strategies. On these grounds, formal modeling and verification of Laplace transformation convolution theorem are established in higher order logic theorem prover. Last but not least, formal modeling and verification of the current in Resistor-inductor circuit has been discussed as an instance. The higher order logic formal models of unit step signal and current in the circuit are proposed based on the verified definitions in higher order logic theorem prover. The current is correct and complete by the formal verification. Results show that the correctness of higher order logic formal models of function limit and the related properties. It lays a good foundation for the follow-up formal analysis of control systems.

Keywords function limit; higher order logic; formal verification; theorem proving; convolution

1 引 言

人工智能一直是科技界的一个努力方向,能够与人类互动的机器人更是人们研究的热点之一. 2015 年 7 月,德国大众工厂发生“机器人杀人”事件. 提高机器人的可靠性和安全性是机器人应用的至关重要的问题. 机器人系统的可靠性主要依赖于其控制系统. 形式化方法是基于数学方法描述目标系统属性的一种技术,它提供了更完备的验证结果,为交互式机器人的安全验证提供了坚实的基础. 高阶逻辑定理证明方法是形式化方法中一种严谨的验证方法. 在高阶逻辑定理证明方法中,机器人核心控制系统的验证需要很多定理库的支持,其中拉普拉斯变换的形式化是一个重要的内容. 拉普拉斯变换的形式化是以函数极限和相关性质的高阶逻辑形式化为基础的.

函数极限是数学中的基本概念之一,也是拉普拉斯变换等的理论基础,本文在高阶逻辑定理证明器中研究了函数正无穷大时极限及相关性质的形式化建模和验证,为控制系统的高阶逻辑验证提供基础理论. 第 2 节介绍高阶逻辑定理证明器;第 3 节建立函数正无穷大时极限的高阶逻辑模型;第 4 节对函数极限的基本性质、四则运算和函数积分极限的一些性质进行形式化建模和验证;第 5 节对拉普拉斯变换卷积定理建立高阶逻辑形式化模型;第 6 节应用本文的内容对电阻-电感电路(简称 RL 电路)中的电流进行高阶逻辑形式化建模与验证.

2 高阶逻辑定理证明器

定理证明是形式化方法的一种,它是一种用数学逻辑公式来表达系统及其属性的技术,通过对现实中的物理模型提取属性性质转换为数学模型,再由数学模型转换为逻辑模型,并在相关定理证明器中进行描述,从而得到一个形式化系统,找到某属性的一个证明过程. 定理证明是一种交互式分析技术. 它的分析原理是为系统的需求规范和设计实现建立逻辑模型,然后通过形式化定理验证两者的关系. 如果该定理通过证明是正确的,则说明实现和需求之间是等价的或是蕴含的. 定理证明克服了等价性验证需要建立标准模型和模型检测处理复杂系统会产生空间爆炸的约束的问题. 由于定理证明器可以表达所有可以逻辑化的东西,它已用于多个领域的可靠性分析,如应用高阶逻辑来验证操作系统的安全需求[1];

文献[2]用高阶逻辑表达线性时态逻辑和区间时态逻辑，并以实例说明它在硬件设计验证中的应用；应用定理证明也可以来验证多机器人路径规划的安全性[3]；文献[4]在高阶逻辑证明工具 HOL-Light 中建立了几何代数系统的形式化模型等等. 定理证明虽然可以表达所有可逻辑化的系统，但是证明定理时需要人工引导，于是要求定理证明器的使用者熟悉逻辑推理并且拥有一定的推理经验，这是定理证明难以大众化的原因. 目前，定理证明主要应用于一些系统关键性质的分析，在特殊领域里定理证明发挥了巨大优势. 定理证明技术逐渐成为形式化验证技术的重点研究方向，在研究和应用上都拥有巨大的发展潜力.

定理证明本质是，基于系统的公理及推导规则来为定理寻找证明[5]. 当演绎推理和数学定理的手工推导证明，变化成为符号演算的过程技术，并且可以在计算机上自动进行时，定理证明就成为当今软件工程领域中一种非常重要的形式验证技术[6]，即定理证明系统. 在运用定理证明的方法进行系统设计验证时，辅助手工推导的计算机程序被称为机械定理证明器，它和自动定理证明器组成了自动化程度不同的定理证明器. 现在，具有各种特点的定理证明系统已成为教育、学术及工业界的有力工具. 这些定理证明系统拥有各种不同的特性，主要系统有 ACL2、Coq、Lego、Isabelle、HOL 和 PVS 等. ACL2 是由早期用于软件验证的定理证明器 Boyer-Moore 发展来的，ACL2 从设计上支持基于归纳逻辑理论的自动推理，可应用于软件或硬件系统的验证. ACL2 定理证明器的核心是基于项重写系统. Coq 支持数学断言的表达式，机械化地对这些断言执行检查，辅助寻找正式证明，并从其形式化描述的构造性证明中提取出可验证的程序. Coq 基于归纳构造演算，是构造演算的一种衍生理论.

HOL(Higher-Order Logic)是定理证明中的高阶逻辑定理证明器，由 Gordon 于 20 世纪 80 年代中期在英国剑桥大学创建的高阶逻辑系统[7-8]，其主要特点是通过 ML(Meta Language 是一个通用的函数式编程语言)语言实现高度的可编程性. 本文采用的版本是 HOL4，该工具是利用缜密的数学逻辑来实现工业验证的精确性. ML 是一种强类型函数程序设计语言，是比较经典的函数式语言，其所有对象的类型都必须在编译的时候静态分析决定. ML 提供的类型有单元(unit)、布尔型(bool)、整型(int)、字符串型(string)、实数(real)、元组(tuple)、记录(record)和列表(list). ML 还支持模式匹配、意外处理、类型引用、多态性以及递归数据结构. 它拥有自然的语法和较少的基本概念，其理论基础是 λ 演算，语言实现严谨、高效且易于理解，用户可以容易写出清晰可靠的程序. 大多数著名的推理系统都是用 ML 语言编写的. ML 语言作为一种函数式编程语言，减少了指针的使用，并提供了灵活的表达方式，有助于管理复杂的对象. ML 编译器循环地进行“输入-求值-输出”，用户标准的操作方式是一条一条地输入 ML 表达式或者声明，让 ML 编译器去处理. ML 编译器处理的过程包括类型检查、编译、执行. ML 家族有好几种语言，主要的两种语言是 Caml 和标准 ML，标准 ML 语言被简称为 SML，或者直接称为 ML. ML 语言结合了函数式编程语言和命令式编程语言的特点，这是它得以广泛应用的重要原因.

在定理证明器 HOL4 中进行形式化证明时，如要使用定理库里相关的定义和定理，需要先用 load 和 open 语句加载并打开相应的定理库. HOL4 有非常丰富的定理库，并且由于其庞大的用户基础，定理库会越来越丰富. 随着 HOL4 定理库越来越完善，其应用也越来越广泛. 加拿大 Concordia 大学的 Siddique 等人在高阶逻辑定理证明器 HOL 中验证了分数阶微积分 RL 定义[9]和 Gamma 函数[10]；Liu 等人在高阶逻辑定理证明器 HOL 中验证了马尔科夫链[11]，为系统的状态验证提供基础. Kumar 等人利用 HOL 分析了 DNA 中的纳米生物规模的化学信息处理动态[12]，开发利用全息理解分子尺度生物化学计算行为的新形式. Ahmed 等人在 HOL 中对可靠性框图进行了形式化验证[13]，对串并联、平行、嵌套等建立了形式化模型，并验证了云计算中心的一个通用的虚拟数据的可靠性. 巴基斯坦的 Sardar 等人对于分布式动态热管理系统，应用高阶逻辑方法验证系统的性能和时间属性[14]，包括热性能、温度范围、达到热稳定性时间等. 文献[15]在 HOL4 中研究了一个双臂机器人的避碰规划算法的高阶逻辑形式化验证，并根据验证结果改进了算法.

本文的研究动机是在分数阶PID(比例 Proportion、积分 Integral、微分 Derivative)控制系统高阶逻辑形式化验证的目标下产生的. 分数阶 PID 控制系统的高阶逻辑形式化验证，需要分数阶拉普拉斯变换的相关理论和方法. 分数阶拉普拉斯变换是在拉普

拉斯变换基础上的拓展，因此要对拉普拉斯变换基础内容性质定理进行高阶逻辑形式化验证. Wang 等人在 Coq 中通过复变函数的形式化研究了拉普拉斯变换的形式化[16]. Taqdees 等人在 HOL Light 中通过多重微积分来验证拉普拉斯变换的形式化[17]. 将拉普拉斯变换拓展到分数阶拉普拉斯变换，需要用到函数极限的高阶逻辑形式化. Weber 在 Isabelle/HOL 中给出了函数在固定点处极限的定义[18]. 还有学者在 Isabelle/HOL 和 Coq 中给出了函数极限的形式化研究[19-20]，但仅是函数趋于固定点的极限定义，没有相关性质验证. 拉普拉斯变换拓展研究需要函数在正无穷大时极限，及其相关性质属性等. 由于研究团队之前的分数阶微积分定义和性质等验证工作是建立在 HOL 上的，因此本文在 HOL 中建立了函数在正无穷大时极限的定义和性质等定理，以便后续工作的持续开展. 在高阶逻辑定理证明器 HOL 中已包括一些基本定理库，有实数库[21]，超越函数库、自然数库[22]、极限库[23]、链表库[24]等，目前在 HOL 中不存在函数在正无穷大时极限的高阶逻辑形式. 本文的验证基于首都师范大学团队研发的复数库[25]、gauge 积分库[26]和函数库中的一部分内容.

3 函数极限定义的建模与验证

函数无穷远处极限：设函数 $f: D\subset R\to R$ 是一个定义在实数上的函数，并在某个开区间$\{x>A\}$上有定义. L 是一个给定的实数. 如果对任意的正实数 ε，都存在一个正实数 X，使得对任意的实数 x，只要 $x\geq X$，就有 $|f(x)-L|\leq\varepsilon$，那么就称 L 是函数 f 在 x 趋于正无穷大时的极限，或简称 L 为 f 在正无穷处的极限，记为 $\lim\limits_{x\to+\infty} f(x)=L$ 或 $f(x)\to A(x\to+\infty)$. 反之则称 L 不是 f 在 x 趋于正无穷大时的极限.

在 HOL 中，首先对实数域函数趋于正无穷的符号“→”进行形式化描述，给出如下定义：

定义 1. 函数无穷远处极限定义

```
val tends_real_real=
⊢ ∀f f0.f→f0⇔(f tends f0) (mtop mrl, $>=)
```

定义中(f: real→real)为实数域中要取正无穷极限的函数，(f0: real)则为该函数在正无穷远处的极限值. 而符号“→”则用来表示中缀操作，它的前项为要取无穷极限的实数域函数，而它的后项则为该函数在无穷远处所取得的极限值. 因为 HOL 中其它极限(包括序列极限以及函数在某点上极限)的定义都使用等价于拓扑极限的方式，所以在本文中也采用这种定义方式. 在拓扑学中，网是序列的广义化，用来统一极限不同的概念和将其广义至任意的拓扑空间. 根据拓扑中网极限的定义，定义函数在实数域的极限. 式中 tends 是网的定义，后面则表示满足该网的一种条件. mtop 为是将度量转换为拓扑的函数，mrl 为实数直线上的度量定义，$>= 则表示中缀操作，是一个 real→real→bool 的集合运算符，而这个集合中的所有元素都满足偏序关系，这里主要用于确定取极限时的方向，取负无穷时则会用 $≤.

根据上述对实数域函数在正无穷处取极限的形式化描述定义，下面则对上述的定义进行形式化证明如下：

定理 1. 函数无穷远处极限

```
val FUNC_POS=
⊢ ∀f f0.f→f0⇔∀e.0<e ⇒?X.∀x.x≥&X⇒
abs(f x−f0)<e
```

该定理等价符号“⇔”的左边部分为实数域函数无穷极限的形式化定义，右边部分为实数域函数在正无穷处极限的文字定义. 在这里，利用 HOL 的高阶逻辑对其进行形式化等价性证明.

证明. 首先将符号“→”进行重写，用定义进行替换，得到(f tends f0)(mtop mrl, $>=)，然后通过度量拓扑中极限的特征定理 MTOP_TENDS 对目标进行重写. 可将(f tends f0)(mtop d, $>=)替换成一个类似等价符号右边的形式，因为该定理就是将度量拓扑的定义展开，并证明了集合满足某种关系，这里即为偏序关系. 接着将实数域中的特殊函数类型及极限值类型用度量拓扑中的任意类型进行替换即可.

对于上面的符号 $>=在集合域中满足有向集的关系，这里将其以定理的形式给出证明，以方便后面使用. 其高阶逻辑形式化证明如下：

定理 2. 集合有序关系

```
val DORDER_RNGE=⊢ dorder $>=
```

证明. dorder 为 HOL 系统中有向集合的定义. 在数学中，有向集合(也叫有向预序或过滤集合)是一个具有序关系[27](自反及传递之二元关系≤)的非空集合 A，而且每一对元素都会有个上界[28]，亦即对于 A 中任意两个元素 a 和 b，存在着 A 中的一个元素 c(不必然不同于 a,b)，使得 $a\leq c$ 和 $b\leq c$

(有向性). 有向集合是非空全序集合的一般化. 在拓扑中它们用来定义一般化序列的网，并联合在数学分析中用到的各种极限的概念. 将 dorder 定义重写，并且利用实数在关系 $>= 上具有自反性的性质来证明. 会得到目标式子"?z. ∀w. z≤w⇒x≤w∧y≤w"，若要证明 w 既大于 x 又大于 y，则需要证明 w 大于 x，y 中较大的一个即可. 因此再增加一个前件用来讨论当 x≥y 时，以及 y≥x 时的两种情况. 因此可以证明当 x≥y 时，取 z 为 x 值；当 y≥x 时，取 z 为 y 值，两种不同情况下都成立.

对于实函数无穷极限取值的高阶逻辑形式化定义有：

定义 2.　实函数无穷极限值定义

val f_lim=⊢ ∀f. f_lim f=@l. f→l:thm

该定义的返回值并不是一个定义形式，而是函数在正无穷远处的极限值，这个定义将在后续形式化建模函数极限其它性质以及形式化拉普拉斯变换中会使用到. 实际控制系统的验证中还需要实函数在正无穷处取极限时的一些相关性质，其形式化验证如下.

4　函数极限相关性质的建模与验证

4.1　基本性质

(1) 唯一性

若极限 $\lim_{x\to+\infty} f(x)$ 存在，则此极限是唯一的. 其形式化定理如下：

定理 3.　唯一性

val FUNC_UNIQ=

⊢ ∀x x1 x2. x→x1∧x→x2⇒(x1=x2)

证明.　首先根据目标将定义的函数无穷极限的定义进行重写；其次，根据 HOL 中度量拓扑中网的极限唯一性定理 MTOP_TENDS_UNIQ 进行 MATCH 证明. 这时，则需要证明关系集合：$>= 是一个有向集. 这就是上面已经证明的定理 2，将之前证明的引理 DORDER_RNGE 进行 MATCH 接收写入策略.　　证毕.

(2) 保不等式性

如果两个函数 f，g 在正无穷远处有极限，即 $\lim_{x\to+\infty} f(x)=L$，$\lim_{x\to+\infty} g(x)=M$，并且在给定任意实数 ε，当 x>ε 时，f(x)≥g(x)，则有 L≥M. 其形式化定理如下：

定理 4.　保不等式性

val FUNC_LE=

⊢ ∀f g l m.

f→l∧g→m∧(?X. ∀x. x≥X⇒f x≤g x)⇒

l≤m

证明.　首先使用 GEN_TAC 将目标中的全称量词去掉；其次需要使用 MP_TAC 引入一个新的假设条件，这个假设条件是度量拓扑中网的极限的比较定理 NET_LE. 因为这个定理的形式是对任意类型的二元关系集合，所以还需要使用 geq($>=: real→real→bool) 对该定理进行实例化关系类型，则最后使用的策略为：MP_TAC(ISPEC geq NET_LE). 因为引入的前件是一个已经证明过的定理的实例化，所以在此不需要进行证明，可直接引入. 最后先使用 tends_real_real 对目标中的函数在正无穷极限的定义进行重写；然后使用 geq 和有向集的定理 DORDER_RNGE 对目标进行重写，再使用实数的自反性 REAL_LE_REFL 对目标进行证明，可得到跟前件的形式相似的目标；最后使用匹配接收策略 MATCH_ACCEPT_TAC 可完成证明.　　证毕.

(3) 极限为零

如果函数 f 在无穷远处存在极限，并且极限为零，即 $\lim_{x\to+\infty} f(x)=0$，则其绝对值函数在无穷远处的极限也为零，即 $|\lim_{x\to+\infty} f(x)|=0$，反之也成立. 其形式化定理如下所示：

定理 5.　极限为零

val FUNC_ABS=⊢ ∀f. (λx. abs(f x))→0⇔f→0

证明.　首先使用 GEN_TAC 将目标中的全称量词去掉，其次使用函数正无穷极限的定理 FUNC_POS 重写目标，将会得到函数正无穷极限定义的形式目标. 接着使用 BETA_TAC 将目标中的 λ 函数去掉，然后使用任意实数与零之间的差值关系定理 REAL_SUB_RZERO 及任意实数的绝对值的绝对值与其绝对值相等的定理：ABS_ABS 对目标进行重写即可证明完成.　　证毕.

(4) 极限等价性

若函数 f 在正无穷远处存在极限，则该函数在正无穷远处的极限值为其在正无穷远处的取值.

定理 6.　极限等价性

val LIM_FUNC_EQ=⊢ ∀g l. f→l⇒(f_lim f=l)

证明.　该定理即是证明了之前定义的 f_lim 的正确性，即当函数 f 在正无穷远处存在极限，则

可推出 f_lim f 的值即是该函数的极限值.首先将目标中的全称量词去掉并将 f_lim 的定义进行重写,因为 f_lim 定义中使用了选择操作符@,所以在证明的时候需要使用消除选择操作符的策略:SELECT_ELIM_TAC,将选择操作符@变为存在(?)和任意(∀)两个形式的目标.此时可使用策略 CONJ_TAC 将合取形式的目标分成两个子目标的形式.两个子目标的前件都为函数 f 的正无穷处极限为 l,第一个子目标为证明存在一个实数 x 使得函数 f 的正无穷处的极限为该实数,而第二个子目标则为对任意实数 x 若函数 f 的极限为 x,则 x 与 l 相等.第一个子目标很好证明,只需要将用 l 代替 x,然后使用前件重写目标即可.在第二个子目标中会发现,函数 f 有两个正无穷极限值 x 和 l,然后需要证明 x 和 l 相等,这与之前证明的函数在正无穷处的极限唯一性定理一致,因此可以直接用自动证明策略 PROVE_TAC[FUNC_UNIQ](并将极限唯一性定理代入)可完成证明. 证毕.

(5)常函数极限

若函数 f 为常函数,则其在正无穷远处的极限值为函数本身.

定理 7. 常函数极限

val FUNC_CONST=⊢ ∀k.(λx.k)→k.

证明. 首先将目标中的全称量词去除,然后用函数无穷极限的定理:FUNC_POS 对目标进行重写,再分别使用定理:REAL_SUB_REFL 和 ABS_0 将目标中的"abs(k−k)"变为"0".此时的目标为:"∀e. 0<e⇒?X. ∀x.x≥&X⇒0<e",发现目标中的前件和结论都为"0<e",那么就需要去除全称量词,并使用策略 DISCH_TAC 将目标中的前件放在条件队列中,最后用自动重写策略 ASM_REWRITE_TAC[]即可完成证明. 证毕.

(6)变量与常数之和的极限

若函数 f 在正无穷远处存在极限,即 $\lim\limits_{x\to+\infty} f(x)=L$,则有任意常数 a 使得下式成立:

$$\lim_{x\to+\infty} f(a+x)=L.$$

其形式化描述:

定理 8. 变量与常数之和的极限

val LIM_FUNC_LAM_ADD=

⊢ ∀f l a.(λt.f t)→l⇔(λt.f (a+t))→l.

(7)变量与常数之积的极限

若函数 f 在正无穷远处存在极限,即 $\lim\limits_{x\to+\infty} f(x)=L$,则有任意常数 $a>0$ 使得下式成立:

$$\lim_{x\to+\infty} f(a*x)=L.$$

其形式化描述:

定理 9. 变量与常数之积的极限

val LIM_FUNC_LAM_MUL=

⊢ ∀f l a.a>0∧(λt.f t)→l⇔(λt.f(a*t))→l.

这两个定理通过消去全称量词,并用函数无穷极限的定理对目标重写即可证明.

4.2 函数极限四则运算的建模与验证

极限形式化语言只能证明极限,不能求极限.对于简单函数的极限问题,可以使用比较容易的证明方法证明其极限存在或不存在,但对于一些形式比较复杂的函数,就不太容易证明.因此,函数正无穷极限的运算法则的形式化十分必要,它对于证明复杂函数无穷极限的问题至关重要.在上面正无穷极限的形式化模型的基础上,这里将对正无穷极限的运算法则进行形式化建模及验证.

在对函数在正无穷远处的极限的运算法则进行形式化描述和证明中,形式化过程包括两种形式化内容.一种是在推出结果的形式中不带有"="而是使用"→"的形式化,这是最基本的形式化证明;另外一种则是基于上一种证明的进一层,是在推导结果中使用"="和 f_lim 的形式,而这种形式是在后面函数极限的应用形式化建模中所需要的.如若跳过第一层形式的证明,直接到第二层,则会有很多证明代码的冗余.因此,为了能够更加有效地组织这些证明逻辑、过程以及代码形式,最终抽出第一层形成单独的定理形式,这样在后续的证明过程中可减少不必要的重复证明.

(1)函数极限加法

若函数 f 和 g 在正无穷远处都存在极限,即 $\lim\limits_{x\to+\infty} f(x)=L$, $\lim\limits_{x\to+\infty} g(x)=M$,则有函数 f 和 g 的和在正无穷处存在极限并且:

$$\lim_{x\to+\infty}(f(x)+g(x))=L+M.$$

其形式化描述:

定理 10. 函数加法

val FUNC_ADD=

⊢ ∀f g l m.f→l∧g→m⇒(λt.f t+g t)→(l+m).

证明. 首先将全称量词去掉,然后将 tends_real_real 的定义进行重写,去掉符号"→",根据目标的形式使用 MATCH 策略匹配定理 NET_ADD,目标会变成"dorder $>=",此时需要使用前面证

明过的定理 2：DORDER_RNGE 来说明“$>=”满足有向集关系，用 MATCH 接收策略将该定理代入即可.　　证毕.

定理 11.　函数极限加法

val LIM_FUNC_ADD=

⊢ ∀$f\ g\ l\ m.f\to l\wedge g\to m\Rightarrow$(f_lim($\lambda x.f\,x+g\,x$)=$l+m$).

证明.　由于极限的运算是用 f_lim 定义写的，所以首先去除全称量词及重写 f_lim 的定义，由于 f_lim 定义中使用了选择操作运算符@，因此证明需要使用策略 SELECT_ELIM_TAC，这样就可以将目标转换为两个目标的合取形式，一个是证明存在性，一个证明等价性. 使用 CONJ_TAC 将目标中的合取形式变为两个子目标. 第一个子目标，是证明两个函数的和在无穷远处存在极限，根据前件中的条件函数 f 和 g 在无穷远处的极限分别为 l 和 m，因此只需要证明两个函数的和在正无穷远处的极限值为 $l+m$ 即可. 那么就需要 MATCH_MP_TAC 对上面已经证明完的定理 FUNC_ADD 进行匹配，然后重写目标即可. 第二个子目标，去掉全称量词后发现它的目标形式与之前证明函数极限的唯一性定理 FUNC_UNIQ 很相似. 此时需要使用 MATCH 策略匹配该定理，则目标变成了“$?x'.x'\to x\wedge x'\to(l+m)$”的形式. 先证明目标合取式的第一部分，即证明存在性，显然存在的 x' 应该为($\lambda x.f\,x+g\,x$)，所以根据前件的条件可以消去目标合取式的第一部分；而目标的第二部分与上面第一层证明极限和的形式一样，所以使用 MATCH 策略匹配定理 FUNC_ADD 即可.　　证毕.

(2) 函数极限减法

若函数 f 和 g 在正无穷远处都存在极限，即 $\lim\limits_{x\to+\infty}f(x)=L$，$\lim\limits_{x\to+\infty}g(x)=M$，则有函数 f 和 g 的差在正无穷处存在极限并且：

$$\lim_{x\to+\infty}(f(x)-g(x))=L-M.$$

其形式化描述：

定理 12.　函数减法

val FUNC_SUB=

⊢ ∀$f\ g\ l\ m.f\to l\wedge g\to m\Rightarrow(\lambda t.f\,t-g\,t)\to(l-m)$.

该定理的证明方法与定理 10 是类似的.

定理 13.　函数极限减法

val LIM_FUNC_SUB=

⊢ ∀$f\ g\ l\ m.f\to l\wedge g\to m\Rightarrow$(f_lim($\lambda x.f\,x-g\,x$)=$l-m$).

该定理的证明方法与定理 11 是类似的.

(3) 函数极限乘法

若函数 f 和 g 在正无穷远处都存在极限，即 $\lim\limits_{x\to+\infty}f(x)=L$，$\lim\limits_{x\to+\infty}g(x)=M$，则有函数 f 和 g 的积在正无穷处存在极限并且：

$$\lim_{x\to+\infty}(f(x)\times g(x))=L\times M.$$

其形式化描述：

定理 14.　函数乘法

val FUNC_MUL=

⊢ ∀$f\ g\ l\ m.f\to l\wedge g\to m\Rightarrow(\lambda t.f\,t*g\,t)\to(l*m)$.

定理 15.　函数极限乘法

val LIM_FUNC_MUL=

⊢ ∀$f\ g\ l\ m.f\to l\wedge g\to m\Rightarrow$(f_lim($\lambda x.f\,x*g\,x$)=$l*m$).

证明.　通过使用策略 SELECT_ELIM_TAC，可以将目标转换为两个目标的合取形式，一个是证明存在性，一个证明等价性. 使用 CONJ_TAC 将目标中的合取形式变为两个子目标. 第一个子目标，是证明两个函数的乘积在无穷远处存在极限. 根据前件中的条件函数 f 和 g 在无穷远处的极限分别为 l 和 m，因此只需要证明两个函数的和在正无穷远处的极限值为 $m\times l$ 即可. 那么就需要 MATCH_MP_TAC 对上面已经证明完的定理 FUNC_MUL 进行匹配，然后重写目标即可. 第二个子目标，需要使用 MATCH 策略匹配该定理，则目标变成了“$?x'.x'\to x\wedge x'\to(l*m)$”的形式，先证明目标合取式的第一部分，证明存在性，显然存在的 x' 应该为($\lambda x.f\,x*g\,x$)，所以根据前件的条件可以消去目标合取式的第一部分；目标的第二部分使用 MATCH 策略匹配定理 FUNC_MUL 即可证明.　　证毕.

(4) 函数极限与常数乘法

若函数 f 在正无穷远处存在极限，即 $\lim\limits_{x\to+\infty}f(x)=L$，则有任意常数 a，使得 a 与 f 的乘积在正无穷处存在极限并且：

$$\lim_{x\to+\infty}a\times f(x)=a\times L.$$

其形式化描述：

定理 16.　函数极限与常数乘法

val LIM_FUNC_CMUL=

⊢ ∀$f\ l\ a.f\to l\Rightarrow$(f_lim($\lambda x.a*f\,x$)=$a*l$);thm.

证明.　由于该目标的形式与上一个定理：函数极限的乘法运算很相似，因此可以借助定理：LIM_

FUNC_MUL 来进行证明.那么就需要将目标的形式变成与所用定理一样的形式,在定理 LIM_FUNC_MUL 中,是两个函数乘积的形式,而在证明目标中则是一个常数与一个函数的乘积形式,因此需要使用定位策略及添加 lambda 函数的方法将常数改写为常函数的形式:CONV_TAC((LAND_CONV o EXACT_CONV)[X_BETA_CONV (--'x:real'--)(--'a:real'--)]).应用到要证明的定理中,则左侧即可表示为,对目标中的常数 a 添加一个自变量为 x 的 lambda 函数.此时的目标形式为:f_lim($\lambda x.(\lambda x.a)\ x * g\ x$)$=a*l$.此时便可以使用 MATCH 策略匹配定理 LIM_FUNC_MUL,然后使用常函数极限定理 FUNC_CONST 进行重写,即可完成证明. 证毕.

(5)函数极限除法

若函数 f 和 g 在正无穷远处都存在极限,即 $\lim\limits_{x\to+\infty} f(x)=L$, $\lim\limits_{x\to+\infty} g(x)=M$,并且 $M\neq 0$,则有函数 f 和 g 的商在正无穷处存在极限并且:

$$\lim_{x\to+\infty}(f(x)/g(x))=L/M.$$

其形式化描述:

定理 17. 函数极限除法

```
val FUNC_DIV=
  ⊢ ∀f g l m.f→l ∧ g→m ∧ m≠&0⇒
  (λt.f t/g t)→(l/m).
```

证明. 首先将全称量词去掉,然后将 tends_real_real 的定义进行重写,去掉符号"→",根据目标的形式使用 MATCH 策略匹配定理 NET_DIV,目标会变成"dorder $>=",此时需要使用前面证明过的定理 2:DORDER_RNGE 来说明"$>="满足有向集关系,即用 MATCH 接收策略将该定理代入即可. 证毕.

4.3 函数积分极限的高阶逻辑形式化建模与验证

(1)正无穷函数积分上限取绝对值的建模与验证

若函数 f 在以任意常数 a 为积分下限存在正无穷积分 L,即 $\lim\limits_{x\to+\infty}\int_a^x f(x)=L$,则有该函数在以 a 为积分下限,积分上限为取正无穷变量的绝对值的极限为 L,即

$$\lim_{x\to+\infty}\int_a^{|x|} f(x)=L.$$

该定理表示,函数积分上限取极限时,与其上限的绝对值取极限的结果一样.其形式化描述如下:

定理 18. 函数积分极限上限绝对值定理

```
val LIM_FUNC_BOUND_ABS=
  ⊢ ∀f a p.(λt.integral(a,t) f)→p⇔
  (λt.integral(a, abs t) f)→p.
```

证明. 首先用极限定理 FUNC_POS 重写,然后使用等价策略(EQ_TAC)将等价性证明变换成两个蕴含式证明.第一个子目标的蕴含式证明:由于目标中存在两个蕴含式且它们之间的关系为蕴含关系,每个蕴含式中都有一个 e,所以先将两个 e 实例化为同一个 e,主要使用 SPEC,即 MP_TAC o SPEC "e:real".目标式中有存在量词的证明,实际上在 $0<t$ 时,abs $t=t$.所以,两个蕴含式中的 X,即存在量词,可以写成一个.此时,目标变为"abs(integral(a,abs x) $f-p$)$<e$".而前件中有三个条件分别为"$x\geq\&X$","$\forall x.x\geq\&X\Rightarrow$ abs(integral(a,x) $f-p$)$<e$"和"$0<e$".从条件 2 中可以发现,其成立的条件为"$x\geq\&X$".对目标来说,它的成立条件形式应该为"abs $x\geq\&X$".所以使用目标中的前件匹配结果,即可得到目标:abs $x\geq\&X$,经过简单的变换,根据条件即可完成第一个子目标的证明.

第二个子目标的证明思路与第一个子目标很相似,但第二个子目标中的蕴含式与第一个子目标中的蕴含式刚好相反,条件和结果进行了调换,所以在证明到目标形式为"abs(integral(a,x) $f-p$)$<e$"的时候,有所不同.此时的前件条件中有"$x\geq\&X$","$\forall x.x\geq\&X\Rightarrow$ abs(integral(a,abs x) $f-p$)$<e$"和"$0<e$".此时条件 2 中的目标中有"abs x"的形式,而条件中的前件却为"$x\geq\&X$",并没有绝对值 abs 的形式.而目标中也没有绝对值 abs 的使用.为了能使用前件中第二个条件,需要将目标的形式改写为与前件第二个条件的目标相似的形式.即用 abs x 代替目标中的 x,将目标变为"abs(integral(a,abs x) $f-p$)$<e$".而前件中第一个条件为"$x\geq\&X$",x 为实数,而 X 为自然数,因此会有"$\&X$"的形式,是将自然数取实数的表示方法.由此可知,若一个实数大于等于一个自然数,那么这个实数与其绝对值是相等的.

证明步骤如下:

首先证明"$0\leq x$",需要使用实数小于等于的传递定理 REAL_LE_TRANS,找到一个处于 0 和 x 的中间值,那就是自然数 X.于是分别证明"$\&X\leq x$"和"$0\leq X$",第一个目标用重写即可(前件条件中有该目标的形式).第二个目标使用定理 ZERO_LESS_EQ(表示自然数不小于零)即可证明.

其次证明"$x=$abs x",使用定理 ABS_REFL(实数绝对值与其本身相等成立的条件,即该实数不小于零),这就需要用到前面证明的目标"$0\leq x$",然

后将条件进行重写即可完成.

至此已经构造出了需要的目标形式“abs(integral(a,abs x) $f-p$)$<e$”,然后使用条件匹配策略FIRST_ASSUM HO_MATCH_MP_TAC,可得到目标为“$x\geq\&X$”,将前件中一样的条件进行重写即可完成证明.　　证毕.

(2)正无穷函数积分上限与常数之和的建模与验证

若函数 f 在以任意常数 a 为积分下限存在正无穷积分 L,即 $\lim\limits_{x\to+\infty}\int_a^x f(x)=L$,则有该函数在以 a 为积分下限,积分上限为取正无穷变量与任意常数 b 的和的极限为 L,即

$$\lim_{x\to+\infty}\int_a^{b+x} f(x)=L.$$

该定理表示,函数积分上限取极限时,与其上限与一常数的和取极限的结果一样.形式化描述如下:

定理 19.　函数积分极限上限可加定理

```
val LIM_FUNC_BOUND_ADD=
  ⊢ ∀f a p b.(λt. integral(a,t) f)→p⇔
  (λt. integral(a, b+t) f)→p
```

证明.　首先将极限的定理进行重写,然后使用等价策略EQ_TAC进行目标重写,将等价性证明变换成为两个蕴含式的证明.第一个子目标的蕴含式证明:由于目标中存在两个蕴含式且它们之间的关系为蕴含关系,主要使用SPEC.目标式中有存在量词的证明,但是目标中的两个 X 则不相同,前件中的存在量词用 X 表示,则目标中的 X 应该用 clg(&(X:num)$-b$:real)表示(clg 表上取整).此时,目标变为“$\forall x.x\geq\&clg(\&X-b)\Rightarrow$ abs((λt. integral(a,$b+t$) f) $x-p$)$<e$”,再将目标中的前件放下去,则前件中有三个条件分别为“$x\geq\&clg(\&X-b)$”,“$\forall x.x\geq\&X\Rightarrow$abs(integral($a$,$x$) $f-p$)$<e$”和“$0<e$”.从条件2中可以发现,其成立的条件为“$x\geq\&X$”.对目标来说,它的成立的条件形式应该为“$b+x\geq\&X$”.所以使用目标中的前件匹配结果,即可得到目标:$b+x\geq\&X$,经过简单的变换,主要会用到上取整定理:LE_NUM_CEILING($\forall x.x\leq\&clg\ x$),根据条件即可完成第一个子目标的证明.第二个子目标的证明:由于第二个子目标的形式跟第一个子目标很相似,所以证明的思路也基本一致.但是第二个子目标中的蕴含式与第一个子目标中的蕴含式刚好相反,条件和结果进行了调换,所以在证明到目标形式为“abs(integral(a,x) $f-p$)$<e$”的时候,有所不同.此时的前件条件中有“$x\geq\&X$”,“$\forall x.x\geq\&X\Rightarrow$abs(integral($a$,abs x) $f-p$)$<e$”和“$0<e$”.此时条件2中的目标中有“$b+x$”的形式,而条件中的前件却为“$x\geq\&X$”,并没有和的形式.而目标中也没有和的使用.为了能使用前件中第二个条件,需要将目标的形式改写为与前件第二个条件的目标相似的形式.即用 $b+(x-b)$ 代替目标中的 x,将目标变为“abs(integral(a,$b+$($x-b$)) $f-p$)$<e$”.而前件中第一个条件为“$x\geq\&clg(\&X+b)$”,x 为实数,而 X 为自然数,因此会有“$\&X$”的形式,是将自然数取实数的表示方法,又使用实数上取整 clg,所以又变成自然数了,然后再用符号“&”即可变为实数的形式.由于“$x=b+(x-b)$”的证明比较简单,这里就不赘述了.目标替换完成后,使用条件匹配策略 FIRST_ASSUM HO_MATCH_MP_TAC,可得到目标为“$x-b\geq\&X$”.由于前件的第一个条件中有“$\&X+b$”的形式,所以将目标转换为“$\&X+b\leq x$”.然后使用 MATCH 匹配实数小于等于的传递定理 REAL_LE_TRANS,则处于目标中间的一个实数为“$\&clg(\&X+b)$”,使得目标“$\&X+b\leq\&clg(\&X+b)\wedge\&clg(\&X+b)\leq x$”成立.最后使用上取整定理 LE_NUM_CEILING,并将目标进行重写,即可完成证明.　　证毕.

(3)正无穷函数积分上限与非负常数之积的建模与验证

若函数 f 在以任意常数 a 为积分下限存在正无穷积分 L,即 $\lim\limits_{x\to+\infty}\int_a^x f(x)=L$,则有该函数在以 a 为积分下限,积分上限为取正无穷变量与任意非负常数 b 的积的极限为 L,即

$$\lim_{x\to+\infty}\int_a^{b*x} f(x)=L.$$

该定理表示,函数积分上限取极限时,与其上限与任一非负常数的乘积取极限的结果一样.形式化描述如下:

定理 20.　函数积分极限上限可乘定理

```
val LIM_FUNC_BOUND_CMUL=
  ⊢ ∀f a p b.&0<b⇒((λt. integral(a, t) f)→
  p⇔(λt. integral(a,b*t) f)→p).
```

该定理的证明思路与定理19的证明思路很相似,首先将极限的定理进行重写,然后使用等价策略 EQ_TAC 进行目标重写,将等价性证明变换成为两个蕴

含式的证明.再分别证明两个蕴含式的子目标即可.

证毕.

验证了函数极限的上述性质定理后,则可以证明拉普拉斯变换的性质定理,为控制系统的高阶逻辑验证提供基础理论支撑.

5 拉普拉斯变换卷积定理的建模与验证

卷积是一种积分变换的数学方法,在许多方面得到了广泛应用.统计学中,加权的滑动平均是一种卷积.概率论中,两个统计独立变量的和的概率密度函数是两个变量的概率密度函数的卷积.声学中,回声可以用源声与一个反映各种反射效应的函数的卷积表示.电子工程与信号处理中,任一个线性系统的输出都可以通过将输入信号与系统函数做卷积获得.物理学中,任何一个线性系统都存在卷积.高斯变换就是用高斯函数对图像进行卷积.

卷积最重要的一种情况,就是在信号与线性系统或数字信号处理中的卷积定理.利用该定理,可以将时间域或空间域中的卷积运算等价为频率域的相乘运算,从而利用快速算法,实现有效的计算,节省运算代价.

拉普拉斯变换卷积定理是指,函数卷积的拉普拉斯变换是函数拉普拉斯变换的乘积.即一个域中的卷积相当于另一个域中的乘积,例如时域中的卷积就对应于频域中的乘积.

$$L(f(x)*g(x))=L(f(x))\times L(g(x)),$$

其中,L(Laplace 的简写)表示的是拉普拉斯变换.拉普拉斯变换的卷积性质不仅能够用来求出某些函数的拉氏逆变换,而且在线性系统的研究中起着重要作用.文献[29]中已验证了拉普拉斯变换定义和存在条件.

根据卷积定义,两个函数 $f,g:R\to C$ 的卷积可用 $f*g$ 表示.拉普拉斯变换的卷积定义如下:

$$(f*g)(t)=\int_{-\infty}^{+\infty}f(\tau)g(t-\tau)\mathrm{d}\tau=\int_{0}^{t}f(\tau)g(t-\tau)\,\mathrm{d}\tau.$$

根据拉普拉斯变换定义,$t<0$ 时 $f(t)=0$.则当 $\tau<0$ 或 $t-\tau<0$ 时,上述定义中的积分值为零,因此在 $t<0$ 时,$(f*g)(t)=0$.

下面为在 HOL4 中对拉普拉斯变换卷积定义的形式化:

定义 3. 卷积定义

```
L_convolution_def
  ⊢ ∀f_1 f_2 t. L_convolution f_1 f_2 t=
  integral(0,abs t) (λ tau. f_1 tau * f_2 (t−tau))
```

这里,假定函数 f 和 g 是分段光滑,容易证明拉普拉斯变换卷积存在.实际上,当固定 $t>0$ 时,以 τ 为自变量的函数 $\tau\to f(\tau)g(t-\tau)$ 同样也是分段光滑的,而且这样的函数在区间 $[0,t]$ 上总是可积.因此,在 $t\in R$ 时,两个分段光滑的函数 f 和 g 的卷积一定存在.拉普拉斯变换卷积定理:$L[f_1(t)*f_2(t)]=L(s)G(s)$.式中,函数 f 和 g 满足拉普拉斯变换的条件,且 $L[f_1(t)]=L(s)$,$L[f_2(t)]=G(s)$.则拉普拉斯变换卷积定理在 HOL4 中的验证如下:

定理 21. 卷积定理

```
val L_TRANS_CONVOLUTION=
  ⊢ ∀f' f'' f_1 f_2 t M c b w t.
  L_exists_condition f_1 b w ∧
  L_exists_condition f_2 b w ∧
  (t<0⇒(f_1 t=0) ∧ (f_2 t=0))⇒
  (lap_trans(λx. L_convolution f_1 f_2 x) b w=
  lap_trans f_1 b w * lap_trans f_2 b w)
```

其中,函数 lap_trans 和 L_exists_condition 是拉普拉斯变换和存在条件的高阶逻辑形式化表示,详细解释参见文献[29].函数 lap_trans 和 L_exists_condition 的高阶逻辑形式如下:

```
val lap_trans_def=
  ∀f b w. lap_trans f b w=
  (f_lim(λt. integral(&0,abs t)
  (λt. f t * exp(−(b * t)) * cos(w * t))),
  f_lim(λt. integral(&0, abs t)
  (λt. −f t * exp(−(b * t)) * sin(w * t))))
val L_exists_condition_def=
  ∀f b w. L_precondition f b w⇔
  ∃M c. ∀t. &0≤t ∧ &0<M ∧ &0<c ∧ c<b⇒
  abs(f t)≤M * exp(c * t) ∧ f contl t
```

在卷积定理中,当 $t<0$ 时,$f_1(t)=f_2(t)=0$,因此在平面 $Re\ s>c$ 上有

$$L(s)G(s)=\int_{0}^{+\infty}f_1(t)e^{-st}\,\mathrm{d}t\int_{0}^{+\infty}f_2(u)e^{-su}\,\mathrm{d}u.$$

从上式可以看出,第二个积分与 t 无关,所以可以将上式写成:

$$L(s)G(s)=\int_{0}^{+\infty}\left(\int_{0}^{+\infty}f_1(t)f_2(u)e^{-s(u+t)}\,\mathrm{d}u\right)\mathrm{d}t.$$

现在可以使用新变量 τ 代替 $t+u$，则可得到：

$$L(s)G(s)=\int_0^{+\infty}\left(\int_0^{+\infty}f_1(t)f_2(\tau-t)e^{-s\tau}\mathrm{d}\tau\right)\mathrm{d}t.$$

根据定理中的条件，将上式积分顺序改变，可以得到：

$$L(s)G(s)=\int_0^{+\infty}e^{-s\tau}\left(\int_0^{+\infty}f_1(t)f_2(\tau-t)\mathrm{d}t\right)\mathrm{d}\tau.$$

如上式所示，其内积分为函数的卷积结果. 因此，$L[f_1(t)*f_2(t)]$存在，并且 $L[f_1(t)*f_2(t)]=L(s)G(s)$.

对上述定理的证明，首先需要将拉普拉斯变换的定义以及卷积的定义重写，并且将等式中复变函数中的实部和虚部分开证明. 本文就以实部的证明进行简要说明，当证明到如下等式：

```
f_lim(λt'. integral(0, abs t')
 (λtau. f_1 tau * integral(tau, abs t'+tau)
 (λt. f_2(t−tau)*(exp(−(b*t))*cos(w*t)))))=
   f_lim(λt. integral(0, abs t)
   (λt. f_1 t * exp(−(b*t)) * cos(w*t))) *
   f_lim(λt. integral(0, abs t)
   (λt. f_2 t * exp(−(b*t)) * cos(w*t)))−
   f_lim(λt. integral(0, abs t)
   (λt. −f_1 t * exp(−(b*t)) * sin(w*t))) *
   f_lim(λt. integral(0, abs t)
   (λt. −f_2 t * exp(−(b*t)) * sin(w*t))).
```

可使用换元积分性质 INTEGRATION_BY_SUBST 对等式左边的内积分进行变换，可将 integral(*tau*, abs $t'+tau$)(λt. f_2 $(t-tau)$ * (exp(−(b*t)) * cos(w*t)))变换为 integral(0, abs t')(λt. f_2 t * (exp(−(b * (t+tau))) * cos(w * (t+tau)))). 使用该性质定理就需要证明目标等式中上下限大小比较的问题，以及左式和右式中上下限使用换元时相等问题. 最重要的是需要证明积分函数存在原函数，在抽象形式化中，并不知道在实际进行验证时是对什么样的模型进行验证，也无法确定该模型提取出的原函数的形式，所以在这里使用定义(L_COS_DIFF)将原函数写在前件中. 后面在使用该定理验证实际应用时，可将前件中原函数的定义用实际中的原函数替换，这样就保证了实际验证的正确性.

根据余弦的二倍角公式(COS_ADD)将 cos(w * (t+tau))化简为 cos(w * t) * cos(w * tau) − sin(w * t) * sin(w * tau). 同时根据指数函数的性质(EXP_ADD)可将 $e^{-(b*(t+tau))}$ 化简为 $e^{-(b*t)}*e^{-(b*tau)}$. 根据积分性质“两函数分别可积，则函数之差的积分与函数积分之差相等(INTEGRAL_SUB)”，将等式左边中的内积分分成两个内积分之差的形式. 再将等式左边外层的积分进行分解，将函数之差的积分写成函数积分之差的形式. 可得

```
f_lim(λt'. integral(0, abs t')
 (λtau. f_1 tau * (exp(−(b*tau)) * cos(w*tau))) *
 integral(0, abs t')
 (λt. f_2 t * (exp(−(b*t)) * cos(w*t)))−
 integral(0, abs t')
 (λtau. −(f_1 tau * (exp(−(b*tau)) *
 sin(w*tau)))) *
 integral(0, abs t')
 (λt. −(f_2 t * (exp(−(b*t)) * sin(w*t))))) =
   f_lim(λt. integral(0, abs t)
   (λt. f_1 t * exp(−(b*t)) * cos(w*t))) *
   f_lim(λt. integral(0, abs t)
   (λt. f_2 t * exp(−(b*t)) * cos(w*t)))−
   f_lim(λt. integral(0, abs t)
   (λt. −f_1 t * exp(−(b*t)) * sin(w*t))) *
   f_lim(λt. integral(0, abs t)
   (λt. −f_2 t * exp(−(b*t)) * sin(w*t))).
```

此时，即可将等式左边的极限写成单个积分极限的形式，用来对应等式的右边形式. 这里主要会用到引理 FUNC_SUB 极限减法运算以及引理 FUNC_MUL 极限乘法运算.

6　RL 电路电流的高阶逻辑形式化建模与验证

RL 电路，全称电阻-电感电路，或称 RL 滤波器、RL 网络，是最简单的无限脉冲响应电子滤波器. 它由一个电阻器、一个电感元件串联或并联组成，并由电压源驱动. 本文验证的是 RL 串联电路. 日光灯电路实际上就相当于一个 RL 电路，它是由电阻元件和电感元件组成，这类电路是很多复杂系统中的组成部分.

RL 电路作为最基本的电子原件的一种组合，在各种电路中使用的非常多. 这里将对 RL 电路中的电流进行高阶逻辑形式化验证. 验证过程中会用到函数极限和拉普拉斯变换的高阶逻辑模型和性质. 这里将从模型、需要验证的性质、形式化验证过程这三方面来描述.

(1)模型

本文讨论的 RL 电路如图 1 所示.

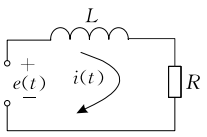

图 1 时域 RL 电路

图中,激励信号为单位阶跃 $u(t)=e(t)$,根据拉普拉斯变换的卷积定理以及相关性质,对该电路电流 $i(t)=\frac{1}{R}(1-e^{\frac{R}{L}t})$进行高阶逻辑验证.

上述时域电路图可等效频域电路图,如图 2 所示.

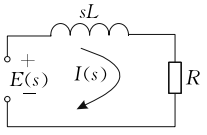

图 2 频域 RL 电路

根据图 2 所示,可知图中电流在频域的表达式应为:$I(s)=\frac{1}{R+sL}E(s)$. $\frac{1}{R+sL}$为频域电路中的导纳(阻抗的倒数),可以写成$\frac{1}{L}\cdot\frac{1}{s-\left(-\frac{R}{L}\right)}$. 电压的拉普拉斯变换(频域表示):$E(s)=L[u(t)]=\frac{1}{s}$.

(2)需要验证的性质

根据拉普拉斯变换的位移性质[29],若 $L[f(t)]=L(s)$,则有 $L[e^{at}f(t)]=L(s-a)$,则可以得到导纳(频域)的拉氏逆变换为$\frac{1}{L}e^{-\frac{R}{L}t}$,这个结果可以根据拉普拉斯变换进行验证. 则将电流在频域中的表达式进行拉氏逆变换,及根据卷积定理,得到结果为

$$i(t)=\frac{1}{L}e^{-\frac{R}{L}t}*u(t)=\int_0^t u(\tau)\cdot\frac{1}{L}e^{-\frac{R}{L}(t-\tau)}\mathrm{d}\tau$$
$$=\frac{1}{R}e^{\frac{R}{L}(t-\tau)}\Big|_0^t=\frac{1}{R}(1-e^{\frac{R}{L}t}).$$

下面将通过高阶逻辑形式化验证这个性质,由此验证电流结果的正确性.

(3)形式化验证

首先在 HOL4 中建立输入信号单位阶跃响应的高阶逻辑模型.

定义 4. 单位阶跃信号的形式化定义:

```
val u_f=
  ⊢ ∀t.u_f t=if t<0 then 0 else 1:thm
```

验证 $f'(t)=\frac{1}{L}e^{-\frac{R}{L}t}$ 的拉普拉斯变换的结果为 $f(t)=\frac{1}{R+sL}$.

引理 1. 阶跃函数频域表示

```
val L_TRANS_INSTANCE_UF_LEMMA=
  ⊢ ∀M c b w.
  0<M∧0<c∧c<b⇒
  (lap_trans (λt.u_f t) b w=
  (b/(b*b+w*w),−(b/(b*b+w*w)))).
```

上面的引理为证明阶跃函数的拉普拉斯变换的结果,即为 $L[u(t)]=\frac{1}{s}$,当 $u(t)=\begin{cases}0, & t<0\\ 1, & t\geq 0\end{cases}$时,$L[u(t)]=\frac{1}{s}$.

导纳的表示证明:

下面是根据上面阶跃函数的引理证明,可以推理出,某函数 f(即为上述 RL 电路在时域中的导纳)的拉普拉斯变换若为$\frac{1}{L}\cdot\frac{1}{s-\left(-\frac{R}{L}\right)}$,那么该结果可以写成 $L(s-a)$的形式,即 a 为$-\frac{R}{L}$,再根据拉普拉斯变换的位移性质 $L[e^{at}f(t)]=L(s-a)$,可将其变为 $L[e^{at}f(t)]$.

证明过程如下所示.

引理 2. 导纳等价性表示证明 1

```
val L_TRANS_INSTANCE_LEMMA=
  ⊢ ∀M c b w R_c.
  0<M∧0<c∧c<b∧0<L⇒
  (lap_trans (λt.1/L*exp(−(R_c /L)*t)) b w=
  lap_trans (λt.1/L)(b−−(R_c /L)) w).
```

上述的证明为说明导纳(频域)在使用拉普拉斯变换表示时存在两种等价的表示方法,即使用拉普拉斯变换的位移性质.

引理 3. 导纳等价性表示证明 2

```
val L_TRANS_ INSTANCE _ADMIN=
  ⊢ ∀M c b w R_c t L.
  0<M∧0<c∧c<b∧0<L∧0<R_c⇒
  (lap_trans(λt.1/L) (b−−(R_c /L))w)=
  1/L*1/(b−(−(R_c /L))).
```

上述引理证明了,导纳的另一种表示方法(根据位移性质),证明了导纳在上述推导的正确性.

引理 4. 电流在时域中表达式

下面则为证明电流在时域的表达式，是根据上面导纳拉氏逆变换的结果进行证明的.

```
val L_TRANS_INSTANCE_LEMMA2=
  ⊢ ∀M c b w R_c L.
  0<M∧0<c∧c<b∧0<L∧0<R_c∧
  (∀t. abs((λt. 1/L * exp(−(R_c /L) * t)) t)⇐
  M * exp(c * t))∧
  (t<0⇒((λt. 1/L * exp(−(R_c /L) * t)) t=0))∧
  (∀t. abs((λt. u_f t) t)≤
  M * exp(c * t))∧(∀t. (λt. u_f t) contl t)⇒
  (lap_trans (λt. u_f t) b w * lap_trans (λt. 1/L)
  (b−−(R_c /L)) w=
  lap_trans (λt. 1/R_c * (1−exp(−(R_c /L) * t)))
  b w).
```

下面是对电流在时域中的表达式结果进行证明，使用的是积分的方法，证明的结果与上面使用拉普拉斯变换的卷积定理证明的结果一样，这同时也说明了拉普拉斯变换卷积定理的形式化证明的正确性和本例中在推理电流在时域中的表达式是正确的.

```
val L_TRANS_INSTANCE_LEMMA3=
  ⊢ ∀x R_c L.
  0<x∧0<R_c∧0<L⇒
  (integral(0,x) (λtau. 1/L * exp(−(R_c /L) *
  (x−tau)))=1/R_c * (1−exp(−(R_c /L) * x))).
```

7　结　论

形式化方法以严格的数学化和机械化方法为基础来规约、构建和验证计算系统，是改善和确保计算系统质量的重要方法，已广泛应用于软硬件验证中. 高阶逻辑形式化验证需要验证者具有专业知识，并采用必要的策略和手段，交互式引导系统完成验证. 对于分数阶控制系统的高阶逻辑形式化验证，首先要完善一些必要的数学理论和性质等，再通过反复推敲各种需要的定义，最终在一系列定理和性质的基础上验证控制系统的精准性、时效性.

函数正无穷大时极限是拉普拉斯变换形式化的基础，拉普拉斯变换形式化验证是控制系统形式化验证的重要部分. 本文验证的函数正无穷大时极限及其性质将是控制系统验证平台的重要理论基础. 后续将研究分数阶拉普拉斯变换的高阶逻辑形式化验证，最后验证分数阶控制系统的优越性能. 在本文的基础上，进一步验证机器人分数阶控制系统的安全稳定性能，为机器人产业的发展提供有力保障.

参　考　文　献

[1] Qian Zhen-Jiang, Huang Hao, Song Fang-Min. Research on consistency verification of formal design and security requirements for operating system. Chinese Journal of Computers, 2014, 37(5): 1082-1099(in Chinese)
(钱振江，黄皓，宋方敏. 操作系统形式化设计与安全需求的一致性验证研究. 计算机学报，2014，37(5)：1082-1099)

[2] Han Jun-Gang. Expressing temporal logic in higher-order logic and its applications. Chinese Journal of Computers, 1993, 16(12): 925-930(in Chinese)
(韩俊刚. 用高阶逻辑表达时态逻辑及其应用. 计算机学报，1993，16(12)：925-930)

[3] Liu Tao, Wang Shu-Ling, Zhan Nai-Jun. Safety verification of trajectory planning for multiple robots. Journal of Software, 2017, 28(5): 1118-1127(in Chinese)
(刘涛，王淑灵，詹乃军. 多机器人路径规划的安全性验证. 软件学报，2017，28(5)：1118-1127)

[4] Ma Sha, Shi Zhi-Ping, Li Li-Ming, et al. Formalization of geometric algebra theories in higher order logic. Journal of Software, 2016, 27(3): 497-516(in Chinese)
(马莎，施智平，李黎明等. 几何代数的高阶逻辑形式化. 软件学报，2016，27(3)：497-516)

[5] Abbasi N. Formal Reliability Analysis using Higher-Order Logic Theorem Proving[Ph. D. dissertation]. Department of Electrical and Computer Engineering, Concordia University, Montreal, Quebec, Canada, 2012

[6] Schumann J M. Automated Theorem Proving in Software Engineering. New York, USA: Springer, 2001

[7] Gordon M J C, Melham T F. Introduction to HOL: A Theorem Proving Environment for Higher Order Logic. New York, USA: Cambridge University Press, 1993

[8] Harrison J. Handbook of Practical Logic and Automated Reasoning. Cambridge, UK: Cambridge University Press, 2009

[9] Siddique U, Hasan O. Formal analysis of fractional order systems in HOL. Formal Methods in Computer-Aided Design, 2011: 163-170

[10] Siddique U, Hasan O. On the formalization of the gamma function in HOL. Journal of Automated Reasoning, 2014, 53: 407-429

[11] Liu L, Hasan O, Tahar S. Formal reasoning about finite-state discrete-time Markov chains in HOL. Journal of Computer Science and Technology, 2013, 28(2): 217-231

[12] Kumar N, da Cruz N C, Rangel E C. DNA for nano-bio scale computation of chemical formalisms using Higher Order Logic (HOL) and analysis using an interdisciplinary approach. Materials Research, 2014, 17(6): 1391-1396

[13] Ahmed W, Hasan O, Tahar S. Formalization of reliability block diagrams in higher-order logic. Journal of Applied Logic, 2016, 18: 19-41

[14] Sardar M U, Hasan O, Shafique M, Henkel J. Theorem proving based formal verification of distributed dynamic thermal management schemes. Journal of Parallel and Distributed Computing, 2017, 100: 157-171

[15] Li L, Shi Z, Guan Y, et al. Formal verification of a collision-free algorithm for a dual-arm robot in HOL4//Proceedings of the Robotics and Automation (ICRA). Hong Kong, China, 2014: 1380-1385

[16] Wang Y F, Chen G. Formalization of Laplace transform in Coq//Proceedings of the 2017 Fourth International Conference on Dependable Systems and Their Applications (Dsa 2017). Beijing, China, 2017: 13-21

[17] Taqdees S H, Hasan O. Formalization of Laplace transform using the multivariable calculus theory of HOL-light// McMillan K, Middeldorp A, Voronkov A eds. Logic for Programming, Artificial Intelligence, and Reasoning. LNCS, vol. 8312. Berlin Heidelberg: Springer-Verlag, 2013: 744-758

[18] Weber T. Bounded model generation for isabelle/HOL. Electronic Notes in Theoretical Computer Science, 2005, 125: 103-116

[19] Puitg F, Dufourd J F. Formalizing mathematics in higher-order logic: A case study in geometric modelling. Theoretical Computer Science, 2000, 234: 1-57

[20] Rashid A, Hasan O. Formal analysis of continuous-time systems using Fourier transform. Journal of Symbolic Computation, 2019, 90: 65-88

[21] Cardell-Oliver R. The formal verification of hard real-time systems [Ph. D. dissertation]. University of Cambridge, Cambridge, UK, 1992

[22] Harrison J. Formalized mathematics. Turku Centre for Computer Science: Technical Report 36, 1996

[23] Slind K, Norrish M. A brief overview of HOL4//Proceedings of the 21st International Conference on Theorem Proving in Higher Order Logics. Berlin, Germany: Springer-Verlag, 2008: 28-32

[24] Harrison J. A HOL theory of Euclidean space//Hurd J, Melham T F, eds. Theorem Proving in Higher Order Logics (TPHOLs 2005). Lecture Notes in Computer Science 3603. Berlin Heidelberg: Springer-Verlag, 2005: 114-129

[25] Shi Zhi-Ping, Li Li-Ming, Guan Yong, et al. Formalization of the complex number theory in HOL4. Applications of Mathematics, 2013, 7(1): 279-286

[26] Gu Wei-Qing, Shi Zhi-Ping, Guan Yong, et al. Formalization of Gauge integration theory in HOL4. Computer Science, 2013, 40(2): 191-194(in Chinese)
(谷伟卿，施智平，关永等. Gauge 积分在 HOL4 中的形式化. 计算机科学，2013，2：191-194)

[27] Schröder B S W. Ordered Sets: An Introduction. Boston, USA: Birkhäuser, 2002

[28] Davey B A, Priestley H A. Introduction to Lattices and Order. New York: Cambridge University Press, 2002

[29] Zhao Gang, Zhao Chun-Na, Guan Yong, et al. Formalization of Laplace transform calculus in HOL4. Journal of Chinese Computer Systems, 2014,35(9): 2177-2181(in Chinese)
(赵刚，赵春娜，关永等. 拉普拉斯变换微积分性质在 HOL4 中的形式化. 小型微型计算机系统，2014，35(9)：2177-2181)

ZHAO Chun-Na, Ph. D., professor. Her main research interests include higher order logic verification and fractional order systems.

ZHAO Gang, M. S. His main research interest is higher order logic verification.

Background

This paper surveys the field of higher order logic formal verification. Higher order logic formal verification is in the stage of development. The researchers from all over the world work on higher order logic theorem library, and then verification of related practical problems on the ground of the library.

This work is the partial content of higher order logic formal modeling and verification of fractional order systems. Formal modeling and verification of function limit are created in higher order logic. Based on higher order logic models of set and number, formal model of function limit definition and its related properties—uniqueness, inequality preserving property, the absolute value function limit when solving its positive infinity limit, limit equivalence, constant function limit, etc—are proposed in this paper. Higher order logic model of arithmetic of function limit is also established in

higher order logic theorem prover. And some related theorems are verified. The higher order logic formal modeling and verification of function integral limit is also proposed in this paper. Formal modeling and verification of Laplace transformation convolution theorem is established in higher order logic theorem prover. Formal modeling and verification of the current in Resistor-inductor circuit has been discussed as an instance. The higher order logic formal models of unit step signal and current in the circuit are proposed based on the verified definitions in higher order logic theorem prover.

The work attributes to the project "Formal Analysis and Verification of Fractional Order PID Controller Based on Higher Order Logic", which is supported by the National Natural Science Foundation of China (No. 61862062).

Fractional order control system is a milestone in the history of fractional order control theory. It is able to improve the control precision and accuracy of the system and get more robust control results. Theorem proving formal verification method can be used for any system that can be expressed by mathematical model. It is the ideal verification method because it is not subject to limits on state numbers. This project researches formal verification and modeling of fractional order control system. The research results will enhance to improve the control performance of control system, to achieve the complete verification of fractional order control systems, to ensure the reliability and security of robot control systems.

Our research group has been working on higher order logic formal verification of fractional order systems. Fractional order calculus has been verified in higher order logic theorem prover. Related works were published in good-reputation journals and conferences, such as Theoretical Computer Science, ISA Transactions, ICRA.

This paper provides the basic mathematical foundation library for control system. It lays a good foundation for the follow-up formal analysis of fractional order control systems.