计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.181102026

GSOS 算子下共变-异变模拟的公理刻画

李苏婷[1]　　张　严[2]

1 南京航空航天大学计算机科学与技术学院　南京 211106

2 南京林业大学信息科学技术学院　南京 210037

（WY2458969213@163.com）

摘　要　进程的行为理论是进程演算研究的核心内容之一，其侧重于讨论进程间的行为等价和模拟关系。共变-异变模拟(Covariant-Contravariant Simulation，CC-模拟)的概念是对经典(互)模拟概念的推广，它通过区分动作类型，刻画了规范与实现对系统主动、被动和通讯动作在精化关系中的不同要求。行为关系的(前)同余性和公理刻画是进程演算代数特征的集中体现，它们对规范及实现的分析和推理至关重要。一般而言，行为关系(前)同余性的证明和公理系统的构造需要基于不同进程演算系统的结构化操作语义(Structural Operational Semantics，SOS)分别展开。为了避免这类研究工作中的重复劳动，学术界针对一般化 SOS 规则形式的元理论开展了研究，GSOS 是其中被广泛研究的规则形式之一。文中在考量了动作类型的基础上，基于 CC-模拟对 GSOS 规则形式做出扩充，提出了 CC-GSOS 规则类型，证明了 CC-模拟相对于 CC-GSOS 算子具有前同余性，并给出了在这些算子下 CC-模拟的可靠完备公理系统的一般性构造方法。

关键词：GSOS；结构化操作语义(SOS)；进程演算；共变-异变模拟；可靠性；完备性

中图法分类号　TP301

Axiomatizing Covariation-Contravariation Simulation Under GSOS Operators

LI Su-ting[1] and ZHANG Yan[2]

1 College of Computer Science and Technology，Nanjing University of Aeronautics and Astronautics，Nanjing 211106，China

2 College of Information Science and Technology，Nanjing Forestry University，Nanjing 210037，China

Abstract　The behavioral theory of processes is one of the core research contents of process calculus，which focuses on discussing behavioral equivalence or simulation relationships between processes. The Covariant-Contravariant simulation (CC-simulation) is the extension of the (bi)simulation，which distinguishes the types of actions and expresses the different requirement of active，passive and communication actions in refinement relationships between specifications and implementations. The (pre)congruence and axiomatization of behavioral relationships are the concentrated expression of the algebraic features of process calculus，which are essential for the analysis and reasoning of specifications and implementations. In general，the proof of behavioral (pre)congruence and the construction of axiomatic systems need to be based on the Structural Operational Semantics (SOS) of different process calculus systems. In order to avoid duplication of labor in these kinds of research work，the academia has carried out research on the meta-theory of the generalized SOS rule formats，and GSOS is one of the format that have been widely studied. Based on the consideration of action types，this paper extends the GSOS rule format for CC-simulation，proposes CC-GSOS rule format，and proves that the CC-simulation is a precongruent relation relative to CC-GSOS operators，gives a general method for constructing axiomatic system for CC-simulation which is sound and complete.

Keywords　GSOS，Structural operational semantics (SOS)，Process calculus，Covariant-contravariant simulation，Soundness，Completeness

到稿日期：2018-11-03　返修日期：2019-03-17　　本文已加入开放科学计划(OSID)，请扫描上方二维码获取补充信息。

基金项目：国家自然科学基金(61602249)；江苏省普通高校自然科学研究资助项目(17KJB520012)

This work was supported by the National Natural Science Foundation of China (61602249)，Natural Science Foundation of the Jiangsu Higher Education Institutions of China(17KJB520012).

通信作者：张严(zhangyan@njfu.edu.cn)

1 引言

进程演算(如 CCS[1],CSP[2],MEIJE[3])是刻画并发与交互式反应系统行为的原型规范语言,它们都是通过进程项来描述反应式系统的规范及实现,实现是否满足规范则由行为等价或者精化关系来刻画。由于行为关系相对演算系统中算子的(前)同余性是对规范以及实现模块化构造、分析及推理的基础,因此建立行为关系的(前)同余性是进程演算研究的重要内容之一。除此之外,行为关系(前)同余性的证明以及行为关系的公理刻画是进程演算代数特性的集中体现,它为进程间的等价(或模拟)关系的机器定理证明奠定了不可或缺的理论基础。行为关系(前)同余性的证明以及公理刻画都与演算系统的算子有关。一般而言[1,4-5],行为关系同余性的证明以及公理系统的构造需要针对不同的进程演算系统分别开展。为了避免研究工作中的重复劳动,学术界针对结构化操作语义[6]的元理论开展了研究,希望基于规范进程算子操作语义的 SOS 规则形式,给出同余性证明及公理化构造的普适性结论和构造方法。其中,GSOS[7]形式是被广泛研究的规则形式之一,G 表示有卫递归(Guared Recurisoon),是 GSOS 规则的一大特征。Bloom 等[7]对 GSOS 形式的合理性进行了阐述,证明了互模拟关系相对 GSOS 算子的同余性。Aceto 等[8]建立了 GSOS 算子下关于强互模拟关系可靠完备公理系统的一般性构造方法。近年来,针对具体的应用场景发展出了丰富的 GSOS 变形。Argenio 等[9]将 Aceto 和 Bloom 的公理化成果提升到概率操作上,提出了 PGSOS 形式,并基于 PGSOS 定义的概率系统规范(Probabilistic System Specification)给出了互模拟公理系统的构造方法。Klin 等[10]针对随机加权系统(Stochastic and Weighted Transition Systems)构造了 SGSOS。Bonchi 等[11]在 Stream GSOS[12]规则形式下获得了刻画 Mealy 机上开项间迹等价的证明技术。Aceto 等[13]在 GSOS 上研究了输入-输出一致性模拟(Input-Output Conformance Simulation)的逻辑特征及其组合性。Rot[14]在泛代数的范畴上指出,若一个单调的 biGSOS 规范的标记转换系统模型中存在一个带点的 DCPO 结构,则该规范就具有最小模型(Least Model)。

如上所述,行为等价或者模拟关系是刻画系统实现是否满足其规范的一个重要概念。反应式系统遵循刺激-反应计算模式,这类系统只具有被动的行为。经典模拟关系[15]主要是针对这种计算模式的系统间精化关系的数学刻画,它要求规范中被动的行为必须在实现中被模拟。但对于具有主动行为的系统(例如输入/输出自动机[16]),经典的模拟关系是不适用的[4,17]。为此,学术界提出了共变-异变模拟[18]概念,它包含了经典的模拟和互模拟概念。CC-模拟是基于将动作划分为共变、异变和互变 3 种类型而定义的。共变动作表示系统的被动行为,其执行受环境控制,规范中共变动作所标记的转换必须在实现中被模拟,体现了对实现的活性要求;异变动作代表了由系统控制的主动行为,实现中异变动作的执行必须是规范所允许的,它反映了对系统安全性的要求;互变动作的要求体现了经典的互模拟的思想。

本文将针对 CC-模拟,对 GSOS 规则形式做出适当的变形,提出 CC-GSOS 形式。CC-GSOS 是对经典 GSOS 规则的推广,在规则形式的定义中考量了动作的不同类型,并通过实例说明了其对 CC-模拟动作所附加的限制是合理的。在此基础上,本文研究了 CC-GSOS 规则类型相对 CC-模拟的若干元性质,证明了 CC-GSOS 算子下 CC-模拟的前同余性,进而给出了 CC-GSOS 规范下 CC-模拟可靠完备公理系统的一般性构造方法。

本文第 2 节介绍了 GSOS 规则和 GSOS 语言的基本概念;第 3 节介绍了 CC-模拟的概念,提出了 CC-GSOS 的定义,给出了 CC-GSOS 算子对 CC-模拟前同余性的证明及其提出的动机;第 4 节给出了一种通用的公理化 CC-模拟不等式系统的算法,并证明此算法对 CC-模拟的可靠性;第 5 节利用第 4 节给出的公理化算法,分别对(语义)良基和(语义)非良基的系统获取的 CC-模拟公理系统的完备性给出了证明;最后总结全文。

2 基本概念

GSOS 语言是指用 GSOS 规则定义进程项中算子的操作语义的进程演算语言,GSOS 语言的基本概念与文献[7-8]一致。令 V 是可数无限的进程变元集, $x,y,\cdots$ 表示其中的元素。Σ 是型,包含了常元 0、多元算子及其元数。Act_τ 为包含系统内动作 τ 的有限动作标记集, $a,b,\cdots$ 表示 Act_τ 中的元素。与通常的进程演算语言一样,GSOS 语言中进程项的 BNF 范式定义为:

$$P::=0\mid x\mid f(P_1,\cdots,P_l)$$

其中,$x\in V$ 且 $f\in\Sigma$ 是 l 元算子。通常用 $\mathbb{T}(\Sigma)$ 表示基于 Σ 上的所有进程项的集合,$T(\Sigma)$ 表示所有闭项(不含自由变元的进程项)的集合,用 $P,Q\cdots$ 表示进程项。一个 Σ-语境 $C[\vec{x}]$ 是一个进程项,该进程项中的变元不超过 $\vec{x}$ 中的变元。由于语言中不含约束算子,因此可简单地定义 $C[\vec{p}/\vec{x}]$ 是将 $C[(\vec{x})]$ 中的变元 x_i 对应用 P_i 替换后的结果,在不易混淆的情况下也可直接将 $C[\vec{p}/\vec{x})]$ 记作 $C[\vec{p}]$。正转换公式形如 $P\xrightarrow{a}Q$,否转换公式形如 $P\stackrel{\alpha}{\nrightarrow}$,一个转换公式可以为正,也可以为否,用 $\varphi,\phi,\psi,\cdots$ 表示。一个(闭)Σ-替换是一个函数 $\sigma:V\rightarrow T(\Sigma)$,若 t 是一个项、转换公式,或者是一个 GSOS 规则,则用 $t\sigma$ 表示用 σ 替换 t 中每个变元后的结果。

定义 1[8] 一个基于型 Σ 的 GSOS 规则 r 是由一组转换公式作为前提和一个单独的公式作为后承组成的,形如:

$$\frac{\bigcup_{i=1}^{l}\{x_i\xrightarrow{a_{ij}}y_{ij}\mid 1\leqslant j\leqslant m_i\}\cup\bigcup_{i=1}^{l}\{x_i\stackrel{b_{ik}}{\nrightarrow}\mid 1\leqslant k\leqslant n_i\}}{f(x_1,\cdots,x_l)\xrightarrow{c}C[\vec{x},\vec{y}]}\tag{1}$$

其中,所有的变元 x_i 和 y_{ij} $(1\leqslant i\leqslant l,1\leqslant j\leqslant m_i;m_i\geqslant 0,n_i\geqslant 0)$ 都是互不相同的,$f\in\Sigma$ 是 l 元算子,$C[\vec{x},\vec{y}]$ 是 Σ-语境,该语境中所含的变元包含在由所有的 x_i 和 y_ij 组成的集合中, $a_{ij},b_{ik},c\in Act_\tau$。$x_i\xrightarrow{a_{ij}}y_{ij}$ 叫做 r 的正前提,$x_i\stackrel{b_{ik}}{\nrightarrow}$ 叫做 r 的否前提,r 的所有前提组成的集合记作 $prem(r)$,r 的后承记作 $cons(r)$。称 $f(x_1,\cdots,x_l)$ 为 r 的源,称由 GSOS 规则定义的

算子为 GSOS 算子。

一个 GSOS 系统中 G 是一个二元组(Σ_G, R_G)，它刻画了 GSOS 语言定义的标记转换系统[19]中进程行为的转换模型。其中，型 Σ 是有限的，R_G 是一个有限的基于 Σ_G 的 GSOS 规则集。一个转换模型$\rightarrow$是 $T(\Sigma_G)\times Act_\tau \times T(\Sigma_G)$的子集，其中元素$(P,a,Q)$可记为 $P\xrightarrow{a}Q$。一个闭转换公式 φ 在$\rightarrow$中成立或在$\rightarrow$中为真(记作$\rightarrow\vDash\varphi$)，当且仅当 $\varphi\in\rightarrow$。对于一闭转换公式集 H，$\rightarrow\vDash H$ 当且仅当对每个 $\varphi\in H$，有$\rightarrow\vDash\varphi$。$\rightarrow$是 S 的一个模型，当且仅当对任意的 $r\in R_G$ 和闭替换 σ，若$\rightarrow\vDash prem(r\sigma)$则$\rightarrow\vDash cons(r\sigma)$。$\rightarrow$是被 S 支持的，当且仅当对任意的 $\varphi\in\rightarrow$，存在 $r\in R_G$ 和闭替换 σ 使得$\rightarrow\vDash prem(r\sigma)$且 $cons(r\sigma)=\varphi$。如果$\rightarrow$是被 S 支持的且是 S 的模型，则$\rightarrow$是 S 的支持模型。

引理 1[7] 对每一个 GSOS 系统 G，都存在唯一的支持模型，且每一个项 $P\in T(\Sigma_G)$都是分叉有限的。

一个系统规范存在唯一的支持模型是研究该系统规范和实现间的等价或者精化关系性质的前提条件，只有借助于该唯一支持模型的推理才能保证结论是可靠的。后文中用符号$\rightarrow_G$ 表示 G 唯一的支持模型。除此之外，GSOS 系统的分叉有限性质也是后文中公理系统完备性证明的必要条件。

3 CC-模拟

定义 2 令 G 是一个 GSOS 系统，二元关系 $R\subseteq T(\Sigma_G)\times T(\Sigma_G)$是动作集 $A\subseteq Act_\tau$ 上的 CC-模拟关系，当且仅当$(P, Q)\in R$ 蕴含着对所有的 $a\in A$ 满足：

(CCR)如果 $P\xrightarrow{a}_G P'$且 $a\in A^r\cup A^{bi}$，则存在一个 Q'使得 $Q\xrightarrow{a}_G Q'$且$(P',Q')\in R$；

(CCL)如果 $Q\xrightarrow{a}_G Q'$且 $a\in A^l\cup A^{bi}$，则存在一个 P'使得 $P\xrightarrow{a}_G P'$且$(P',Q')\in R$。

其中，A^r，A^l，A^{bi}是对动作集 A 的一个划分，三者互不相交。A^r 表示共变动作集，A^l 表示逆变动作集，A^{bi}是互变动作集。此处定义将 τ 动作和普通可视动作同等看待，因此所定义出的 CC-模拟是一个强的模拟概念。

易验证上述概念是互模拟和模拟概念的推广形式。互模拟关系事实上就是 A^r 和 A^l 取 $\emptyset$，模拟则相当于 A^l 和 A^{bi} 取 $\emptyset$。按照 CC-模拟概念提出的标准，任何正确实现的共变行为都应该具备其规范要求的活跃性，所有异变行为必须禁止超出其规范所规定的安全限度，互变行为应该相互匹配[6-7]。

在 GSOS 系统 G 上，P CC-模拟 Q(记作 $P\geqslant_{(A^r,A^l)_G}Q$)当且仅当存在一个 CC-模拟关系 $R\subseteq T(\Sigma_G)\times T(\Sigma_G)$使得$(P, Q)\in R$。易验证，$\geqslant_{(A^r,A^l)_G}$是系统 G 上最大的 CC-模拟关系。

在公理化一个进程演算语言上的(互)模拟关系之前，不可避免地需要获得一条重要性质——该语言的算子关于互模拟的同余性或模拟的前同余性。该性质对于支持形式规范的模块化构建和推理具有重要意义。当系统具有同余性时，某个项可以用与之行为等价的项进行替换，而整个系统的行为保持不变，有时也把(前)同余性称为可代换性。但是，经典的 GSOS 规则形式下定义的算子不一定都能满足对 CC-模拟关系的前同余性。因此，下文将对 GSOS 规则做出一定的限制，使得限制后的算子能够满足对 CC-模拟的前同余性，并给出例子来说明这些限制的必要性。

定义 3 若一个 GSOS 规则满足以下两个条件：

(1)对于每一个规则中的后承 $f(x_1,\cdots,x_l)\xrightarrow{c}C[\underset{x}{\rightarrow},\underset{y}{\rightarrow}]$，若 $c\in A^r\cup A^{bi}$，则其正前提中的转换动作只属于 $A^r\cup A^{bi}$，其否前提中的转换动作只属于 $A^l\cup A^{bi}$。

(2)对于每一个规则中的后承 $f(x_1,\cdots,x_l)\xrightarrow{c}C[\underset{x}{\rightarrow},\underset{y}{\rightarrow}]$，若 $c\in A^l\cup A^{bi}$，则其正前提中的转换动作只属于 $A^l\cup A^{bi}$，其否前提中的转换动作只属于 $A^r\cup A^{bi}$。

称满足上述条件的 GSOS 规则为 CC-GSOS 规则，称由 CC-GSOS 规则定义的算子为 CC-GSOS 算子。如果一个 GSOS 系统中的所有算子都是 CC-GSOS 算子，那么该 GSOS 系统被称为 CC-GSOS 系统，可见 CC-GSOS$\sqsubseteq$GSOS。CC-GSOS 系统中的算子都能对 CC-模拟保持前同余性，而 GSOS 系统则不一定。

定理 1(前同余性) 假设 G 是一个 CC-GSOS 系统，$f\in\Sigma_G$是 l 元算子，若 $P_i\geqslant_{(A^r,A^l)_G}Q_i$ $(1\leqslant i\leqslant l)$，则 $f(P_1,\cdots,P_l)\geqslant_{(A^r,A^l)_G}(Q_1,\cdots,Q_l)$。

证明：只需验证下述构造的关系 S 是一个 CC-模拟关系，即可证得定理 1。

令 $S_0\overset{\triangle}{=}\geqslant_{(A^r,A^l)_G}$，$S_{n+1}\overset{\triangle}{=}\{(f(P_1,\cdots,P_l),f(Q_1,\cdots,Q_l))\mid 1\leqslant i\leqslant l, l$ 元算子 $f\in\Sigma_G, P_iS_nQ_i\}\cup\geqslant_{(A^r,A^l)_G}$，$S\overset{\triangle}{=}\bigcup_{n<\omega}S_n$，接下来验证每一对$(P,Q)\in S$ 都能满足(CCR)和(CCL)条件。

验证(CCR)条件：令 $P\xrightarrow{a}_G P'$且 $a\in A^r\cup A^{bi}$，根据 S 的构造过程归纳证明，有以下两种情形需要讨论。

(1)$(P,Q)\in\geqslant_{(A^r,A^l)_G}$。该情形显然满足(CCR)条件。

(2)$(P,Q)\in S_n$，$n>0$。此时不失一般性，假设 $P\equiv f(P_1,\cdots,P_l)$，$Q\equiv f(Q_1,\cdots,Q_l)$，$P_iS_{n-1}Q_i(1\leqslant i\leqslant l)$。根据 S 的构造归纳假设，可得 $P_i\geqslant_{(A^r,A^l)_G}Q_i$ $(1\leqslant i\leqslant l)$。因为 $P\xrightarrow{a}_G P'$且 $a\in A^r\cup A^{bi}$，即 $f(P_1,\cdots,P_l)\xrightarrow{a}_G P'$，由$\rightarrow_G$ 是G 的支持模型可知，存在一个形如式(1)且满足定义 3 限制条件的规则 $r\in R_G$ 和一个闭替换 σ 使得：

1)$\sigma(x_i)=P_i$，$1\leqslant i\leqslant l$；

2)$P_i\xrightarrow{a_{ij}}_G P'_{ij}\equiv\sigma(y_{ij})$，$P'_{ij}\in T(\Sigma_G)$，$1\leqslant i\leqslant l$，$1\leqslant j\leqslant m_i$，$a_{ij}\in A^r\cup A^{bi}$；

3)$P_i\overset{b_{ik}}{\nrightarrow}_G$，$1\leqslant i\leqslant l$，$1\leqslant k\leqslant n_i$，$b_{ik}\in A^l\cup A^{bi}$；

4)$P'\equiv\sigma(C[\underset{x}{\rightarrow},\underset{y}{\rightarrow}])$，$c=a$。

由归纳假设得到 $P_i\geqslant_{(A^r,A^l)_G}Q_i(1\leqslant i\leqslant l)$，因此对每个 i $(1\leqslant i\leqslant l)$和 $j(1\leqslant j\leqslant m_i)$，存在 $Q'_{ij}\in T(\Sigma_G)$，使得 $Q_i\xrightarrow{a_{ij}}_G Q'_{ij}$，且 $P'_{ij}\geqslant_{(A^r,A^l)_G}Q'_{ij}$，对每个 $i(1\leqslant i\leqslant l)$，$k(1\leqslant k\leqslant n_i)$且 $b_{ik}\in A^l\cup A^{bi}$，有 $Q_i\overset{b_{ik}}{\nrightarrow}_G$。可构造一个替换函数 δ 使得：对 $1\leqslant i\leqslant l$ 和 $1\leqslant j\leqslant m_i$，有 $\delta(x_i)=Q_i$，$\delta(y_{ij})=Q'_{ij}$。由此构造可知，对任意的变量 $x\in\{x_i\mid 1\leqslant i\leqslant l\}\cup\{y_{ij}\mid 1\leqslant i\leqslant l, 1\leqslant j\leqslant m_i\}$，都满

足$(\sigma(x),\delta(x))\in S$，因此，同样利用 r 规则可得 $f(Q_1,\cdots,Q_l)\xrightarrow{a}_G\delta(C[\underset{x}{\rightarrow},\underset{y}{\rightarrow}])$。

引理 2 令 $t\in\mathbb{R}(\Sigma_G)$，σ 和 δ 是两个替换函数，对 t 中的每个变元 x，如果$(\sigma(x),\delta(x))\in S$，则$(\sigma(t),\delta(t))\in S$。

证明：根据项的结构复杂度归纳，易证引理 2 成立。

根据引理 2 可得$(\sigma(C[\underset{x}{\rightarrow},\underset{y}{\rightarrow}]),\delta(C[\underset{x}{\rightarrow},\underset{y}{\rightarrow}]))\in S$，即$(P',\delta(C[\underset{x}{\rightarrow},\underset{y}{\rightarrow}]))\in S$，因此(CCR)条件满足。(CCL)条件的验证类似。

定理 1 的结论可广泛用于具有 CC-GSOS 类型算子的进程演算系统对互模拟、模拟以及 CC-模拟这类典型的行为关系的(前)同余性质的快速检验，省去了证明过程中大量的重复劳动。

下面将给出 4 个反例来说明定义 3 中对 GSOS 规则添加限制条件的必要性，这些例子都是基于一个 GSOS 系统 G，Σ_G中仅包含 0，+和 a.()算子(三者的操作语义与 CCS 中定义的一致)以及算子();()，下文中的 $a_l\in A^l,a_r\in A^r$。

例 1 假设();()算子的定义规则只有一条，即：

$$\frac{x_2\xrightarrow{b}x_2',x_1\stackrel{a}{\nrightarrow}}{x_1;x_2\xrightarrow{b}x_1;x_2'}(a\in A^r,b\in A^r\cup A^{bi})$$

令 $P\equiv a_l.0$，$P'\equiv a_l.0+a_r.0$，则易知 $P,P'\in T(\Sigma_G)$，$P\geqslant_{(A^r,A^l)_G}P'$。假设 $Q\in T(\Sigma_G)$且 $Q\xrightarrow{b_1}Q'$，$b_1\in A^r\cup A^{bi}$，则因为 P 执行不了 $A^r\cup A^{bi}$ 中的任何动作，所以由上述规则可知有 $P;Q\xrightarrow{b_1}P;Q'$的转换，而 P'则无法运用到上述规则中的 x_1 上，$P';Q$ 无法执行 b_1 转换。因此，$P;Q\ngeqslant_{(A^r,A^l)_G}P';Q$。

例 2 假设算子();()对应的语义规则改为：

$$\frac{x_2\xrightarrow{b}x_2',x_1\stackrel{a}{\nrightarrow}}{x_1;x_2\xrightarrow{b}x_1;x_2'}(a\in A^l,b\in A^l\cup A^{bi})$$

令 $P\equiv a_l.0+a_r.0$，$P'\equiv a_r.0$，则 $P,P'\in T(\Sigma_G)$，$P\geqslant_{(A^r,A^l)_G}P'$。假设 $Q\in T(\Sigma_G)$且 $Q\xrightarrow{b_1}Q'$，$b_1\in A^l\cup A^{bi}$，则根据上述规则有转换 $P';Q\xrightarrow{b_1}P';Q'$，而 $P;Q$ 无法执行 b_1 动作。因此，$P;Q\ngeqslant_{(A^r,A^l)_G}P';Q$。

例 3 继续修改();()的语义规则：

$$\frac{x_1\xrightarrow{a}x_1',x_2\xrightarrow{b}x_2'}{x_1;x_2\xrightarrow{b}x_1';x_2'}(a\in A^l,b\in A^r\cup A^{bi})$$

令 $P\equiv a_l.0+a_r.0$，$P'\equiv a_r.0$，则 $P,P'\in T(\Sigma_G)$，$P\geqslant_{(A^r,A^l)_G}P'$。假设 $Q\in T(\Sigma_G)$且 $Q\xrightarrow{b_1}Q'$，$b_1\in A^r\cup A^{bi}$，则由上述规则有 $P;Q\xrightarrow{b_1}0;Q'$，但 $P';Q$ 无法执行 b_1 转换。因此，$P;Q\ngeqslant_{(A^r,A^l)_G}P';Q$。

例 4 将();()的语义规则更改为：

$$\frac{x_1\xrightarrow{a}x_1',x_2\xrightarrow{b}x_2'}{x_1;x_2\xrightarrow{b}x_1';x_2'}(a\in A^r,b\in A^l\cup A^{bi})$$

令 $P\equiv a_l.0$，$P'\equiv a_l.0+a_r.0$，则易知 $P,P'\in T(\Sigma_G)$，$P\geqslant_{(A^r,A^l)_G}P'$。假设 $Q\in T(\Sigma_G)$且 $Q\xrightarrow{b_1}Q'$，$b_1\in A^l\cup A^{bi}$，类似例 3 可知 $P;Q\ngeqslant_{(A^r,A^l)_G}P';Q$。

例 1 和例 2 说明要保证算子对 CC-模拟的前同余性，则必须限制其 GSOS 规则的否前提和后承上的转换标记所属的动作类型不能同属于 A^l 或者 A^r。例 3 和例 4 则说明规则的正前提的动作类型要与后承的转换动作类型一致，才能保证前同余性。综上可知，定义 3 中的限制条件对 GSOS 算子关于 CC-模拟的前同余性的保证是非常必要的。

定义 4[8](不相交的扩展) 令 H 和 G 是两个 CC-GSOS 系统，如果 H 中的规则和符号包含 G 中的规则和符号，且对属于 G 中的算子不添加新的规则，则 H 称作 G 的不相交的扩展，记作 $G\sqsubseteq H$。

不相交的扩展实质上是表达 H 对 G 的扩展不会对 G 中的项引入新的转换，因此易知对任意的 $P,Q\in T(\Sigma_G)$，$P\geqslant_{(A^r,A^l)_G}Q$ 当且仅当 $P\geqslant_{(A^r,A^l)_H}Q$。

4 公理化

不相交的扩展前的进程项之间的 CC-模拟关系在扩展后依旧保持，而大多数的进程演算系统都是对最基本的有限树系统(记作 FIN)做不相交的扩展，因此我们考虑在 FIN 公理系统上加入扩展部分的算子公理化的结果，从而构成新的公理系统。Aceto 等[8]用类似的思想给出了一种在 GSOS 系统上对互模拟关系公理系统构造的一般性方法。同一个系统关于 CC-模拟的公理系统往往只比其关于互模拟的多了 S_p^r 和 S_p^l 两条公理，因此基于 Aceto 等的研究成果，本文在 CC-GSOS 系统上对 CC-模拟构造了可靠完备公理系统。该方法获得的公理系统的可靠性，利用下面的引理 3 和引理 4 并结合文献[8]中证明的互模拟可靠性结果即可保证，因此后文中的引理 6-引理 8、引理 10 和引理 12 的证明易得，将不再赘述。

FIN 中的进程项 t 的 BNF 定义为：$t::=0\mid t+t\mid a.t$。其中，0 是一个无任何可执行能力的常量算子，+和 $a.()$算子的操作规则与 CCS 中定义的一致，为了书写简便，下文有时会直接用 at 表示 $a.t$。易验证，FIN 是一个 CC-GSOS 系统。为了方便后文引理的描述，还将用算子 $\dagger B$ 对 FIN 做不相交的扩展，扩展后得到的 CC-GSOS 系统用 $G_\dagger$ 表示。对 $\dagger B$ 定义的规则为：

$$\frac{x\xrightarrow{a}y}{x\dagger B\xrightarrow{a}y}(a\notin B)$$

下面将基于公理系统 T_{FIN} 逐步展示本文公理系统构造的过程。下文公理系统中的等式 $x=y$ 表示 $x\geqslant y$ 且 $y\geqslant x$。

引理 3[4] 令 T_{FIN} 表示 FIN 关于$\geqslant_{(A^r,A^l)_{FIN}}$的公理系统，该公理系统包含以下公理：

$$x+y=y+x \tag{B_1}$$
$$(x+y)+z=x+(y+z) \tag{B_2}$$
$$x+x=x \tag{B_3}$$
$$x+0=x \tag{B_4}$$
$$x\geqslant x+a_r.y,a_r\in A^r \tag{S_p^r}$$
$$x+a_l.y\geqslant x,a_l\in A^l \tag{S_p^l}$$

T_{FIN} 关于$\geqslant_{(A^r,A^l)_{FIN}}$是可靠完备的。

引理 4 G 是一个 GSOS 系统，$P,Q\in T(\Sigma_G)$，$\sim_G$ 是 G 上的强互模拟[8]关系（按常规定义），则 $P\sim_G Q$ 蕴含着 $P\geqslant_{(A^r,A^l)_G}Q$。

证明：此证明是平凡的，一对强互模拟的进程一定是强 CC-模拟的。

引理 5 令 $T_\dagger$ 表示一个公理系统，该公理系统是对 T_{FIN} 加入了下面几条公理进行扩展所得：

$$(x+y)\dagger B=(x\dagger B)+(y\dagger B) \tag{2}$$

$$ax\dagger B=ax,\text{如果 } a\notin B \tag{3}$$

$$ax\dagger B=0,\text{如果 } a\in B \tag{4}$$

$$0\dagger B=0 \tag{5}$$

则 $T_\dagger$ 关于$\geqslant_{(A^r,A^l)_{G_\dagger}}$ 是可靠完备的。

证明：根据引理 3 和引理 4 易得可靠性，完备性证明与文献[8](Lemma 3.2)类似，只需对转换的动作区分对待即可。

后文 CC-模拟不等式公理系统构造的过程中需要对 CC-GSOS 系统中的算子做出一定的划分，这将涉及到一些特殊类型的算子，如光滑(smooth)算子、独特(distinctive)算子。该公理化的实现主要分为两大步骤：首先，公理化一些特殊类型的算子；其次，将前者的结果拓展到 CC-GSOS 上任意类型的算子。

4.1 公理化特殊类型的算子

定义 5[8] 一个 GSOS 规则是光滑的，当且仅当它是下面的形式：

$$\frac{\{x_i\xrightarrow{a_i}y_i\mid i\in I\}\cup\{x_i\overset{b_{ik}}{\not\rightarrow}\mid i\in K,1\leqslant k\leqslant n_i\}}{f(x_1,\cdots,x_l)\xrightarrow{c}C[\underset{\boldsymbol{x}}{\rightarrow},\underset{\boldsymbol{y}}{\rightarrow}]} \tag{6}$$

其中，I 和 K 是不相交的集合，$I\cup K=\{1,\cdots,l\}$，且没有 x_i $(i\in I)$出现在 $C[\underset{\boldsymbol{x}}{\rightarrow},\underset{\boldsymbol{y}}{\rightarrow}]$。

对于一个 GSOS 算子，若对其定义的所有规则都是光滑的，则称该算子为光滑算子。

定义 6[8] 一个 GSOS 系统中的光滑算子 f 是独特的，当其满足：对于所有关于 f 的规则，它们的前提中对 f 的同一个参量 x_i 要么都只有正的转换，要么都没有正的转换，且对其中每一对不相同的规则，都存在 f 的一个参量，该参量在这两个规则的前提中做正转换，且这两个正转换是两个不同的转换动作标记的。

由 CC-GSOS⊑GSOS 可知，上述光滑、独特算子的定义也可用于对 CC-GSOS 中算子类型的划分。

引理 6 令 f 是一个 l 元光滑 CC-GSOS 算子，G 是用 f 对 FIN 进行不相交扩展后得到的 CC-GSOS 系统。假设 i 是 f 的一个参量的下标，且 f 的所有规则的前提中对该参量的转换都是正的，则该参量对于算子$+$是满足分配律的，即：

$$f(x_1,\cdots,x_i+y_i,\cdots,x_l)=f(x_1,\cdots,x_i,\cdots,x_l)+f(x_1,\cdots,y_i,\cdots,x_l) \tag{7}$$

对于$\geqslant_{(A^r,A^l)_G}$是可靠的。

引理 7 假设 f 是一个独特且光滑的 CC-GSOS 算子，G 是用 f 对 $G_\dagger$ 做不相交扩展后的 CC-GSOS 系统。式(8)是 f 的一条规则：

$$\frac{\{x_i\xrightarrow{a_i}y_i\mid i\in I\}\cup\{x_i\overset{b_{ik}}{\not\rightarrow}\mid i\in K,1\leqslant k\leqslant n_i\}}{f(\underset{\boldsymbol{x}}{\rightarrow})\xrightarrow{c}C[\underset{\boldsymbol{x}}{\rightarrow},\underset{\boldsymbol{y}}{\rightarrow}]} \tag{8}$$

令 $B_i=\{b_ik\mid 1\leqslant k\leqslant n_i\}$，且

$$P_i\equiv\begin{cases}a_iy_i, & i\in I\\ x_i\dagger B_i, & i\in K\wedge\emptyset\subsetneq B_i\subsetneq Act_\tau\\ x_i, & i\in K\wedge B_i=\emptyset\\ 0, & i\in K\wedge B_i=Act_\tau\end{cases}$$

则：

$$f(\underset{P}{\rightarrow})=c.C[\underset{\boldsymbol{P}}{\rightarrow},\underset{\boldsymbol{y}}{\rightarrow}] \tag{9}$$

对于$\geqslant_{(A^r,A^l)_G}$ 是可靠的，称该等式为动作律(Action Laws)。

引理 8 假设 f 是一个光滑的 CC-GSOS 算子，G 是用 f 对 FIN 做不相交的扩展的系统，且项 $P_i(1\leqslant i\leqslant l)$是 $0,x_i,ax_i$ 或者 ax_i+y_i 形式。如果 f 的每一条规则都是式(6)中的形式，且存在一个索引 i 使得下面两种情况之一成立：1) $i\in I$ 且 $P_i\equiv 0$，或者存在一个 $a\neq a_i$，使得 $P_i\equiv ax_i$；2) $i\in K$，且存在某些 $j,1\leqslant j\leqslant n_i,P_i\equiv b_{ij}+y_i$。则：

$$f(\underset{P}{\rightarrow})=0 \tag{10}$$

对于$\geqslant_{(A^r,A^l)_G}$ 是可靠的，称该等式为无为律(Inaction Laws)。

标准型是完备性证明的核心部分，上面所提出的分配律、动作律和无为律正是为了可以将进程表达式重写成标准型的形式。为了对完备性进行证明，下面引入标准型的概念。

定义 7(标准型) 对于一个基于型 $\Sigma\supseteq\Sigma_{FIN}$ 的进程项 P，若 P 是$\sum a_iP_i$ 形式，则 P 是一个标准型。对于一个公理系统 T，如果存在一个 Σ 的项 Q 是标准型，使得 $T\vdash P=Q$，则称 T 可标准化 P。

引理 9[8] 假设对所有的 $i,a_i\notin B,P\equiv\sum a_iP_i,B\subseteq Act_\tau$，则 $T_\dagger\vdash P=P\dagger B$。

定理 2 假设 G 是一个 CC-GSOS 系统，$G_\dagger\sqsubseteq G$。令 $\Sigma\subseteq\Sigma_G-\Sigma_\dagger$ 是一类 G 上的独特且光滑的算子，T 表示用下面的公理对 $T_\dagger$ 进行扩展后的理论。

对 Σ 中的每个算子 f：

(1)对 f 中的每个参量 x_i，如果该参量在规则的前提中做正转换，则有一个形如式(7)的分配律；

(2)对 f 的每个规则，有一个动作律(9)；

(3)对 f 的每条规则，有一个无为律(10)。

则 T 对$\geqslant_{(A^r,A^l)_G}$是可靠的，且 T 可以标准化 $T(\Sigma\cup\Sigma_\dagger)$中所有的项。

证明：文献[8](Lemma 4.9)基于 GSOS 系统，给出了这些公理的加入构成的对互模拟的可靠性以及可标准化的证明。由于 CC-GSOS⊑GSOS，且由引理 3 和引理 4 易得本定理中的可靠性部分。可标准化部分的证明需要用到 CC-GSOS 算子的前同余性条件，证明过程类似于文献[8]。

4.2 公理化一般的算子

4.1 节针对光滑和光滑且独特的算子给出了对应的一般化公理形式，但是非光滑或者非光滑独特的算子仍需进一步处理。下面将对光滑算子和光滑独特算子之间、非光滑和光

滑算子之间分别建立两个等式，以此对所有的算子获取相应的公理。

引理 10 假设 G 是一个 CC-GSOS 系统，$FIN \sqsubseteq G$，f 是 G 的 l 元光滑算子，则存在一个 G 的不相交扩展 G'，该扩展是加入了一组 l 元光滑独特的 CC-GSOS 算子 $f_1, \cdots, f_n$。因此，对所有长度为 l 的 $\vec{x}$，有：

$$f(\vec{x}) = f_1(\vec{x}) + \cdots + f_n(\vec{x}) \tag{11}$$

对 $\geqslant_{(A^r, A^l)_{G'}}$ 是可靠的。

引理 11 假设 G 是一个 CC-GSOS 系统，$G_{\dagger} \sqsubseteq G$，$\Sigma \subseteq \Sigma_G - \Sigma_{\dagger}$ 是一类 G 上的光滑算子，则存在：

(1) 对于一个 G'，$G \sqsubseteq G'$，且 G' 是对 G 加入了有限的一类 Σ' 上的独特光滑的 CC-GSOS 算子，其中 $\Sigma' = \Sigma_{G'} - \Sigma_G$；

(2) 对于一个有限的公理系统 T，T 是对 $T_{\dagger}$ 的扩展，使得 T 对 $\geqslant_{(A^r, A^l)_{G'}}$ 是可靠的，且 T 对 $T(\Sigma' \cup \Sigma \cup \Sigma_{\dagger})$ 中所有的项可标准化。

证明：直接由引理 10 和定理 2 可得。

引理 12 假设 G 是一个 CC-GSOS 系统，$P = f(\vec{z})$ 和 $Q = f'(\vec{v})$ 是两个 Σ_G 上的项，且其中所含的变元不出现在 R_G 中。若 f 和 f' 的规则之间存在一对一的关系，使得每当 f 的一个规则 ρ 和 f' 的一个规则 ρ' 关联，则除了它们的源之外，$\rho\langle\vec{z}, \vec{x}\rangle$ 和 $\rho'\langle\vec{v}, \vec{y}\rangle$ 是一样的（假设 ρ 的源是 $f(\vec{x})$，ρ' 的源是 $f'(\vec{y})$），那么 $P =_{(A^r, A^l)_G} Q$。

引理 13 假设 f 是一个 CC-GSOS 系统 G 中的一个 l 元非光滑算子，则存在一个 l' 元的光滑 CC-GSOS 算子 f'、含 l 个不同变元的向量 $\vec{z}$ 和 l' 个不同变元的向量 $\vec{v}$，使得：

$$f(\vec{z}) = f'(\vec{v}) \tag{12}$$

对 $\geqslant_{(A^r, A^l)_G}$ 是可靠的。其中，G' 是 G 的一个不相交扩展。

证明：该证明类似于文献[8]（Lemma 4.13）。

下面将正式提出 CC-GSOS 系统上 CC-模拟公理系统的通用构造算法。

算法 1 CC-GSOS 算子下 CC-GSOS 系统构造的一般性算法

输入：一个 CC-GSOS 系统 G

输出：一个 CC-GSOS 系统 G′ 和一个有限公理系统 T，其中 $G \sqsubseteq G'$，T 对 $\geqslant_{(A^r, A^l)_{G'}}$ 是可靠的且 T 可以公理化 $T(\Sigma_{G'})$ 中所有的项。

步骤 1 如果 $G \sqsubseteq G_{\dagger}$，则将 $G_{\dagger}$ 的不相交拷贝添加到 G。

步骤 2 对每个非光滑的算子 f，运用引理 13 构造一个 f 光滑版本 f′，用 f′ 扩展系统。将式(12)的所有实际运用结果加入到 $T_{\dagger}$。

步骤 3 对于上面所得系统中的每个光滑非独特算子 $f \notin \Sigma_{\dagger}$，运用引理 10 生成光滑独特的算子 $f_1, \cdots, f_n$，则加入这些算子即可得到要构造的 G′。将式(12)的实际运用结果全都加入公理系统中。

步骤 4 将所有 $\Sigma_{G'} - \Sigma_{\dagger}$ 中的光滑独特算子运用定理 2 获得的等式加入到步骤 3 获得的公理系统中，即得到要输出的公理系统。

定理 3（可靠性） 令 G 是一个 CC-GSOS 系统，G' 和 T 分别是由算法 1 获得的 G 的不相交的扩展和相应的有限公理系统，则 T 对 $\geqslant_{(A^r, A^l)_{G'}}$ 是可靠的，且 T 可以公理化 $T(\Sigma_{G'})$ 中所有的项。

证明：直接由上述引理和定理可得。

4.3 算法实例

为了更好地展现算法 1 的应用，我们将利用算法 1 的公理化优先级算子 θ 来获取 $G_\theta \sqsupseteq_{FIN}$ 的公理系统。θ 算子的定义规则需要依赖于一个 Act_τ 上的偏序关系 $>$，$>$ 满足对任意的 $a \in A^r$，若 $b > a$ 则 $b \in A^l$。θ 算子的 SOS 规则如下：

$$\frac{x \xrightarrow{a} y,\ x \stackrel{b}{\nrightarrow} (\text{对所有的 } b > a, a \in A^r, b \in A^l)}{\theta(x) \xrightarrow{a} \theta(y)}$$

步骤 1 由于 $G_\theta \sqsubseteq G_{\dagger}$，首先需要 $\dagger$ 将算子加入到 G_θ 中。

步骤 2 由于 θ 是一个非光滑算子，根据算法 1，我们需要使用一个新的算子 Δ 来构造一个 θ 光滑版本。Δ 的规则如下：

$$\frac{x \xrightarrow{a} x',\ y \stackrel{b}{\nrightarrow} (\text{对所有的 } b > a, a \in A^r, b \in A^l)}{x \Delta y \xrightarrow{a} \theta(x')}$$

可知 Δ 是一个独特且光滑的操作符。θ 和 Δ 的关系可以用公式表示为：

$$\theta(x) = x \Delta x \tag{T1}$$

步骤 3 由于 Δ 已经是光滑独特的，因此可直接执行算法 1 中的步骤 4，利用定理 2 公理化 Δ 可得以下 5 条公理：

$$(x+y)\Delta z = x\Delta z + y\Delta z \tag{D1}$$

$$ax + y = a.\theta(x)\text{，若 } a \text{ 是最大值} \tag{A1}$$

$$ax\Delta(y\ \dagger\{b \in Act \mid b > a\}) = a.\theta(x)\text{，若 } a \text{ 不是最大值} \tag{A2}$$

$$0\Delta x = 0 \tag{I1}$$

$$ax\Delta(bx + z) = 0\text{，若 } b > a \tag{I2}$$

将上述公理(T1)，(D1)，(A1)，(A2)，(I1)和(I2)加入到引理 5 中给出的 $G_{\dagger}$ 的公理系统 $T_{\dagger}$ 中，即可得到 G_θ 的公理系统。

5 完备性

通常，进程演算中的组合(II)、限制(\)等算子组成的进程项，都可以被化成一个标准型的形式，从而消除这些算子，但递归进程项通常不能转化成标准型。因此，本节将根据进程是否(语义)良基来分情况讨论和证明算法 1 获得的公理系统的完备性。(语义)良基的进程可以化成标准型，这些标准型的项都是 FIN 中的项，(语义)良基系统的完备性可运用引理 3 中 FIN 的公理系统完备性得到。对于非语义良基系统，则运用一个无限的规则来证明其完备性。

5.1 良基 CC-GSOS 系统的完备性

定义 8 G 是一个 CC-GSOS 系统，项 $P \in T(\Sigma_G)$ 是(语义)良基的，当且仅当在 $T(\Sigma_G)$ 和 Act_τ 中不存在无限的转换序列 $P_0 \xrightarrow{a_0}_G P_1 \xrightarrow{a_1}_G P_2 \xrightarrow{a_2}_G \cdots$。$G$ 是(语义)良基的，当且仅当 $T(\Sigma_G)$ 中所有的项都是(语义)良基的。

可见，含递归算子的进程是(语义)非良基的，良基的 CC-GSOS 系统包含了大多数有限字母表上不含递归算子的子语言[8]，因此我们对这类系统的研究是有意义的。

定理 4 G 是一个 CC-GSOS 系统，$FIN \sqsubseteq G$，T 是对 T_{FIN} 进行了扩展的公理系统，T 对 $\geqslant_{(A^r, A^l)_G}$ 可靠，且可以标准化 $T(\Sigma_G)$ 中所有的项。假设 P, Q 是 $T(\Sigma_G)$ 中(语义)良基的项，

则$P \geqslant_{(A^r,A^l)_G} Q$ 当且仅当 $T \vdash P \geqslant Q$。

证明：因$\rightarrow_G$ 是分叉有限的，则可以对每一个良基的项 P 设置一个自然数表示的项的深度 $depth(P)$，该深度代表了该进程项可能连续进行的最大转换数。

令 P,Q 是 $T(\Sigma_G)$中（语义）良基的项且 $P \geqslant_{(A^r,A^l)_G} Q$，由 T 可以标准化 $T(\Sigma_G)$中所有的项，则存在一些 P_i, Q_j 满足 $P \equiv \sum a_i P_i$ 且 $Q \equiv \sum b_j Q_j$。因 $depth(P_i) < depth(P)$ 且 $depth(Q_j) < depth(Q)$，根据项的深度归纳易得：存在一个 $T(\Sigma_{FIN})$中的项 P'使得 $T \vdash P = P'$ 且 $P' \equiv \sum a_i P_i'$；同样地，存在一个 $T(\Sigma_{FIN})$中的项 Q'，$T \vdash Q = Q'$ 且 $Q' \equiv \sum b_j Q_j'$。由 T 的可靠性和引理 3 中 T_{FIN} 的完备性，可证 T 的完备性。

文献[8]说明了算法 1 对 GSOS 系统做不相交扩展时是可以保持系统（语义）良基性的，因此，由 CC-GSOS⊆GSOS 和定理 4 可知（语义）良基的 CC-GSOS 系统用算法 1 获得的公理系统关于 CC-模拟是可靠完备的。一般地，语义良基是不可判定的，而语法良基可以判定，我们可以利用语法良基来判定语义良基[8]。

5.2 非良基 CC-GSOS 系统的完备性

出于对一些简单的带有递归理论的考虑，需要加入一些推理规则到我们的公理系统中，使得算法 1 获得的公理对所有的 CC-GSOS 系统的 CC-模拟都是完备的。在分叉有限的标记转换系统中，支持一个强大的归纳原则——近似归纳原则[20]。由引理 1 可知所有的 CC-GSOS 系统都是分叉有限的，因此可以运用近似归纳原则（AIP）。

文献[8]用算子“/”对原 AIP 原则中的一元算子 $\pi_n(.)$做了改造，我们将直接使用“/”继续表达 CC-模拟的公理系统所需要的 AIP。我们的表达需要将“/”的规则限定在 CC-GSOS 规则形式的要求上，且使用“≥”替换“＝”。对算子“/”的定义规则的形式为：

$$\frac{x \xrightarrow{a} x',\ y \xrightarrow{b} y'}{x/y \xrightarrow{a} x'/y'}$$

其满足定义 3 中对前提和后承的标记动作的限制。

修改后的 AIP 为：

$$\frac{x/b^n \geqslant y/b^n\ (\text{对所有 } n \in N)}{x \geqslant y}$$

其中，$b^n \overset{\triangle}{=} b \cdots b.0$（$n$ 个 b），b 是任意的动作，n 个 b 可能不同。

可见，算子“/”是光滑且独特的，且对 CC-模拟保持前同余。运用前面的引理，我们可以获得一组关于算子“/”的公理：

$$(x+y)/z = (x/z + (y/z)) \tag{13}$$

$$x/(y+z) = (x/y) + (x/z) \tag{14}$$

$$ax/by = a(x/y) \tag{15}$$

$$0/y = 0 \tag{16}$$

$$x/0 = 0 \tag{17}$$

$G_/$表示用算子“/”对任意 CC-GSOS 系统 G 的不相交扩展。

命题 1 AIP 对$\geqslant_{(A^r,A^l)_{FIN_/}}$ 是可靠的。

证明：该证明类似于文献[20]中定理 2.5.8 的证明，且需要用到 CC-GSOS 系统的分叉有限条件。

引理 14 设 G'是一个 CC-GSOS 系统，且 G'和 T 分别是运用算法 1 获得的 $G_/$的不相交扩展以及有限的公理系统，则对所有的 $P \in T(\Sigma_{G'})$和 $n \in N$，都存在一个 $Q \in T(\Sigma_{FIN})$使得 $T \vdash P/b^n = Q$。

证明：根据构造可知包含了公理(13)－(17)。此证明只需根据 n 做归纳，利用公理(13)－(17)以及定理 3，上述结论易得。

定理 5 设 G'是一个 CC-GSOS 系统，且 G'和 T 分别是运用算法 1 获得的 G 的不相交扩展以及有限的公理系统，则对任意的 $P,Q \in T(\Sigma_{G'})$有：

$$P \geqslant_{(A^r,A^l)_{G'}} Q$$

当且仅当：

$$T, AIP \vdash P \geqslant Q \tag{18}$$

证明：（必要性）可靠性可直接由定理 3 和命题 1 得到。

（充分性）由可靠性和 AIP 可知，$P \geqslant_{(A^r,A^l)_{G'}} Q$ 蕴含着对所有的 $n \in N$，$P/b^n \geqslant_{(A^r,A^l)_{G'}} Q/b^n$，因此只需要证明对所有的 $n \in N$，$T \vdash \frac{P}{b^n} \geqslant \frac{Q}{b^n}$，则再利用 AIP 即可得完备性。选择固定一个任意的 n，根据引理 14，存在一个项 $P', Q' \in T(\Sigma_{FIN})$，使得 $T \vdash P/b^n = P'$且 $T \vdash Q/b^n = Q'$。根据可靠性和$\geqslant_{(A^r,A^l)_{G'}}$ 的传递性可知 $P' \geqslant_{(A^r,A^l)_{G'}} Q'$。根据算法 1 的构造有 $G_/ \sqsubseteq G'$，因此 $FIN \sqsubseteq G'$，故 $P' \geqslant_{(A^r,A^l)_{FIN}} Q'$。利用引理 3 中的 T_{FIN} 且 $T_{FIN} \subseteq T$ 可知：$T \vdash P/b^n \geqslant Q/b^n$。因为 n 是任意选择的，所以此证明得证。

综上，定理 4 和定理 5 证明了算法 1 所获得的 CC-模拟的公理系统对良基 CC-GSOS 系统或者非良基 CC-GSOS 系统都是可靠完备的。

结束语 本文基于 GSOS 规则类型的框架，针对 CC-模拟关系进行研究，主要做出的贡献如下：

(1)对经典 GSOS 规则形式做出扩充，针对 CC-模拟提出了 CC-GSOS 规则类型，该形式的算子必须满足定义 3 中的条件限制。定理 1 证明了 CC-GSOS 规则形式下的算子对 CC-模拟的前同余性，并给出了 CC-GSOS 形式算子对满足 CC-模拟前同余性要求的必要性论证。

(2)定理 3 给出了一种对 CC-GSOS 系统构造 CC-模拟不等式公理系统的通用算法，并证明了利用该算法构造的公理系统的可靠性。定理 4、定理 5 证明了该算法构造的公理系统关于 CC-模拟对于良基或者非良基系统都是可靠完备的。

GSOS 是一种被广泛研究的 SOS 规则框架类型，本文的研究成果将对这类规则定义的进程演算系统的行为关系理论的研究提供一种普适性的参考结论。根据本文的研究成果，凡是符合 GSOS 规则类型的系统，都可以通过检查其算子是否匹配 CC-GSOS 形式来判定该算子对 CC-模拟的前同余性。套用算法 1 的框架，任何用 CC-GSOS 规则定义的进程演算系统都可以得到相应的关于 CC-模拟的可靠完备公理系统。由于 CC-模拟概念包容了互模拟和模拟，因此还可以将本公理系统的构造方法推广到对互模拟以及模拟关系的可靠完备公理系统的构造中，以省去这类研究工作的重复劳动。

参 考 文 献

[1] MILNER R. Communication and concurrency[M]. Prentice-Hall International, Englewood Cliffs, 1989.

[2] HOARE C A R. Communicating sequential processes[J]. Communications of the Acm, 1978, 21(21): 666-677.

[3] AUSTRY D, BOUDOL G. Algèbre de processus et synchronisation[J]. Theoretical Computer Science, 1984, 30(1): 91-131.

[4] FÁBREGAS I, FRUTOS-ESCRIG D, PALOMINO M. Equational Characterization of Covariant-Contravariant Simulation and Conformance Simulation Semantics[C]// The Workshop on Structural Operational Semantics. 2010: 1-14.

[5] ESCRIG F, DAVID D, RODRÍGUEZ, et al. Ready to preorder: an algebraic and general proof[J]. Journal of Logic & Algebraic Programming, 2009, 78(7): 539-551.

[6] GROOTE J F, VAANDRAGER F. Structured operational semantics and bisimulation as a congruence[J]. Information & Computation, 1992, 100(2): 202-260.

[7] BLOOM B, ISTRAIL S, MEYER A R. Bisimulation Can't be Traced: Preliminary Report[C]// ACM Symposium on Principales of Programming Languages. ACM, 1987: 229-239.

[8] ACETO L, BLOOM B, VAANDRAGER F. Turning SOS rules into equations[C]// Proceedings of the Seventh IEEE Symposium on Logic in Computer Science, 1992 (LICS'92). IEEE, 1994: 113-124.

[9] D'ARGENIO P R, GEBLER D, LEE M D. Axiomatizing Bisimulation Equivalences and Metrics from Probabilistic SOS Rules [C] // International Conference on Foundations of Software Science and Computation Structures. 2014: 289-303.

[10] KLIN B, SASSONE V. Structural operational semantics for stochastic and weighted transition systems [J]. Information & Computation, 2013, 227(2): 58-83.

[11] BONCHI F, VAN BUSSEL T, LEE M D, et al. Bisimilarity of open terms in stream GSOS[M] // Fundamentals of Software Engineering. Cham: Springer, 2019: 35-50.

[12] KLIN B. Bialgebras for structural operational semantics: An introduction[J]. Theoretical Computer Science, 2011, 412(38): 5043-5069.

[13] ACETO L, FÁBREGAS I, GREGORIO-RODRÍGUEZ C, et al. Logical Characterisations and Compositionality of Input-Output Conformance Simulation [M] // SOFSEM 2017: Theory and Practice of Computer Science. Cham: Springer, 2017: 37-48.

[14] ROT J. Distributive Laws for Monotone Specifications [C]// Combined Workshop on Expressiveness in Concurrency and Structural Operational Semantics (EXPRESS/SOS 2017). EPTCS, 2017: 83-97.

[15] GLABBEEK V R J. The Linear Time-Branching Time Spectrum I-The Semantics of Concrete, Sequential Processes[M]// Handbook of Process Algebra. Elsevier, 2001: 3-99.

[16] LYNCH N. I/O automata: A model for discrete event systems [C]// Annual Conference on Information Sciences and Systems. Princeton, N J, 1988: 29-38.

[17] FÁBREGAS I, ESCRIG D D F, PALOMINO M. Logics for Contravariant Simulations[M]// Formal Techniques for Distributed Systems. Berlin: Springer, 2010: 224-231.

[18] FÁBREGAS I, ESCRIG D D F, PALOMINO M. Non-strongly Stable Orders Also Define Interesting Simulation Relations [C]// International Conference on Algebra and Coalgebra in Computer Science. Springer-Verlag, 2009: 221-235.

[19] KELLER R M. Formal verification of parallel programs[J]. Comm Acm, 1976, 19(7): 371-384.

[20] BAETEN J C M, WEIJLAND W P. Process algebra[M]. Cambridge: Cambridge University Press, 1990.

LI Su-ting, born in 1994, postgraduate. Her main research interests include process algebra and logics in computer science.

ZHANG Yan, born in 1983, Ph.D, lecturer. His main research interests include concurrent theory and formal methods.