计算机科学
COMPUTER SCIENCE

基于CPN的供应链合约的形式化验证

郑红, 钱诗慧, 刘泽润, 杜漫

引用本文

郑红, 钱诗慧, 刘泽润, 杜漫.基于CPN的供应链合约的形式化验证[J]. 计算机科学, 2023, 50(6A): 220300220-7.

ZHENG Hong, QIAN Shihui, LIU Zerun, DU Wen. Formal Verification of Supply Chain Contract Based on Coloured Petri Nets [J]. Computer Science, 2023, 50(6A): 220300220-7.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220300220

基于 CPN 的供应链合约的形式化验证

郑　红[1]　钱诗慧[1]　刘泽润[1]　杜　浸[2]
1 华东理工大学信息科学与工程学院　上海 200237
2 电信科学技术第一研究所　上海 200032

摘　要　智能合约的安全性对于区块链在供应链领域的应用尤为重要。目前,大多数对智能合约的形式化验证工作集中于漏洞检测,对于如何在部署上链前生成安全的智能合约的关注仍然比较少,如何有效规范地将特定领域的属性安全地映射为智能合约代码存在难点。因此,提出在编写合约前基于 CPN(Coloured Petri Net)对供应链业务逻辑进行形式化规范并构建双层仿真模型,以图形化界面描述交易状态变化,进行形式化验证和状态分析,从而在建模阶段就减少逻辑漏洞。最后,提供了一种从 CPN 建模语言到 Solidity 编写的合约的转换方法,以提高智能合约的安全性和可靠性。

关键词:智能合约;形式化方法;模型检查;CPN;供应链
中图法分类号　TP311

Formal Verification of Supply Chain Contract Based on Coloured Petri Nets

ZHENG Hong[1],QIAN Shihui[1],LIU Zerun[1] and DU Wen[2]
1 School of Information Science and Engineering,East China University of Science and Technology,Shanghai 200237,China
2 First Research Institute of Telecommunications Technology,Shanghai 200032,China

Abstract　The security of smart contracts is particularly vital to the application of blockchain in the supply chain field. Currently, most of formal verification work on smart contracts focuses on vulnerability detection,and there is still relatively little attention to how to generate secure smart contracts before deploying them on chain,and there are difficulties in how to effectively and standardly map the properties of specific fields to smart contracts. Therefore,this paper proposes formal specification of supply chain business logic based on coloured Petri Net(CPN) before writing contracts and constructing a two-layer simulation model with a graphical interface to describe transaction state changes for formal verification and state analysis,thus reducing logic vulnerabilities at the modeling stage. Finally,a conversion method from the CPN modeling language to contracts written in Solidity is provided to improve the security and reliability of smart contracts.

Keywords　Smart contract,Formal methods,Model checking,CPN,Supply chain

1　引言

随着区块链技术的爆发式发展,以太坊等区块链计算基础架构为智能合约提供了可信的执行环境,从而可以执行可跟踪、不可逆转的交易,在提供高可用性、可审计性、透明和中立等特性的同时,减少或消除了审查、第三方介入和对手方风险。目前,智能合约可以理解为以确定性方式运行在区块链平台上的不可变的计算机程序,可以处理信息,接收、储存和发送价值。智能合约的应用为保险、金融衍生品、贸易融资甚至知识产权、医疗保险记录等众多领域带来了重大变化。然而,所有智能合约都是公开可见的,任何人都能构造一个交易与其进行交互。一旦智能合约中存在的漏洞被利用,链上的数字资产将面临不可预测的风险。区块链领域重大黑客事件有 The DAO[1]、冻结用户资金的 Parity Multisig Wallet [2]以及 the King of the Ether Throne[3]等。因此,安全性是编写智能合约必须考量的因素之一[4]。在过去几年中,越来越多的研究人员已经采用形式化方法对智能合约的安全性问题进行研究。

形式化验证智能合约的技术主要有模型检查和定理证明,而验证技术的选择主要取决于形式化模型及分析对象的规范。CPN[5-6]是一种基于状态的形式化方法,适用于区块链应用系统的业务流程分析,能够检查工作流特定的属性如死锁和可达性等,使得在工作流建模阶段避免了此类潜在逻辑漏洞。因此,本文提出基于 CPN 对美国石油学会 API(American Petroleum Institute)[7]供应链系统的业务流程进行形式化分析和分层建模,通过有色网中令牌的流动仿真区块链应用系统业务逻辑中交易状态的变化,验证模型是否满足安全属性。再将经过形式化验证的系统业务逻辑转换为相应的智能合约,一方面能够更直观地理解能源石化供应链系统的运作方式,另一方面保证了智能合约功能的正确性和运行时的安全性,推动了区块链在能源石化领域的健康发展。

2　相关工作

形式化方法严格可靠的模型检查或定理证明能够大大提高智能合约的安全性,越来越多的学者开始关注形式化方法

基金项目:2019 年度上海市信息化发展(大数据发展)专项资金项目(201901043)
This work was supported by the 2019 Shanghai Informatization Development(Big Data Development) Special Fund Project(201901043).
通信作者:郑红(zhenghong@ecust.edu.cn)

在智能合约安全性验证方面的应用。

基于定理证明形式化验证智能合约的安全性是典型的方法，但将形式化语言转化为编程语言的过程需要大量的人工投入，自动化程度低。例如，Hirai[8]迈出了形式化 EVM 语义的第一步，提出使用定理证明器 Isabelle/HOL 来手动证明智能合约的某些安全属性和不变性。此同样地，Amani 等[9]在字节码级别使用逻辑框架 Isabelle/HOL 来验证以太坊智能合约的字节码程序的正确性。工具只采用了 EVM 部分不完整的语义，导致遗漏了合约字节码的某些关键行为如合约的循环调用。而模型检查方法将智能合约构建为有限状态机，然后逐一检查状态序列的所有性质，其验证过程是自动的，可以高效执行。文献[10]采用 NuSMV 工具对基于以太坊的智能合约建模，以验证应用程序的实现是否符合其规范，将需要检查的属性形式化为计算树逻辑。由于输入语言的限制，NuSMV 模型无法精确建模。文献[11]以 Azure 区块链的智能合约为例，定义了具体的规定和合约的语义一致性检查问题，并使用 Boogie 工具链开发出一种新的形式化验证工具 VERISOL。但文中未提及运行时间，可以推断该方法不适用于复杂的合约。文献[12]提出使用抽象解释和符号模型快速验证合约的正确性和公平性等安全性，并基于 LLVM 生成了规则的形式化验证工具 ZEUS。但在将合约代码转换为 LLVM 的过程中，无法表达出智能合约程序的所有语义。文献[13-14]中提出的方法显示了 Petri 网在评估安全系统（如访问控制）的代码生成方面能够显著地减少代码开发时间和错误。Duo 等[15]提出了一种层次智能合约建模方法来分析智能合约的安全性，改进了字节码的程序逻辑规则，应用 Hoare 条件建立了 CPN 模型，并引入自定义调用库和基于回溯的路径推导算法，作者侧重于提高 CPN 模型动态仿真的效率。文献[16]提出了一种基于 CPN 工具的形式化验证方法，用于识别区块链系统中智能合约的逻辑漏洞。作者将可能的攻击者加入智能合约进行建模并执行智能合约模型以验证功能的正确性。但该模型只针对特定智能合约中的两个关键函数进行建模，而未验证与其他合约之间交互的安全性。Dong 等[17]使用 CPN 分别对具有拒绝服务攻击漏洞众筹合约整体、无攻击操作和有攻击操作进行建模，侧重于定位合约逻辑漏洞。文献[18]使用 CPN 形式化方法对悬赏合约建模，验证合约中的交易顺序漏洞。从目前的研究成果来看，基于 CPN 的形式化方法具有良好的语义描述且具有图形界面，能够十分直观地对智能合约程序的状态变化和数据流进行安全性分析。

到目前为止，大多数成果是关于形式化验证已部署的智能合约，鲜有关于部署合约前通过形式化方法验证区块链中系统的业务逻辑来生成安全的智能合约的研究，也未发现使用 CPN 建模的形式化方法对石油供应链智能合约安全分析的工作。本文旨在帮助开发人员创建安全的智能合约，而不是修复现有的合约。通过美国石油学会的供应链系统的案例，抽象出其面向多参与方交互信息的业务逻辑，使用 CPN 建模并检查模型中的安全属性，验证无误后再将其直接转换为安全的智能合约。

3 基于 CPN 的供应链合约建模

API 供应链系统的主要焦点是对石油、天然气等产品和各参与方资金流动进行审计并溯源。采用区块链技术能够降低交易的成本，提高交易的安全性和透明性。但是，链上部署的智能合约可能存在安全漏洞，使得 API 供应链系统面临黑客盗取资金、资产冻结等风险。基于区块链的 API 供应链系统业务流程由若干个任务组成，实现该系统所需开发的智能合约将由多个标准化的智能合约集成，每一个智能合约只完成一笔交易内的部分流程。因此，本节根据 API 供应链系统业务逻辑进行形式化规范，使用 CPN Tools 构建了一个双层模型，仿真 API 供应链系统业务流程的部分子任务，然后对各执行流进行形式化验证和状态分析。

3.1 形式化规范

API 供应链系统涉及的参与方有很多，如炼油厂、供应商、运输商、销售点、金融机构以及监管部门等，本模型中涉及到的的参与方为供应商和销售点。顶层模型是主合约的整体结构，如图 1 所示。底层是 Payment 层和 Update 层，分别实现了参与方交易和更新监管部门的指标的功能。

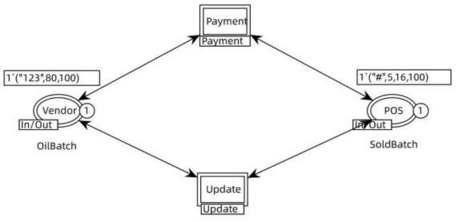

图 1 顶层模型

Fig. 1 Top-level model

CPN 有色网是在经典 Petri 网的基础上对 Token 进行颜色的扩展，将 Token 进行分类来实现对网系统的折叠。合约模型 M 用一个八元组表示，$M=(P,T,F,S,G,E,C,t)$：

(1)P 是一个库所的有限集；

(2)T 是一个变迁的有限集，其中 $P\cap T=\varnothing$；

(3)$F\subseteq(P\times T)\cup(T\times P)$ 是一个弧的有限集，满足 $P\cap F=T\cap F=\varnothing$。输入库所的颜色设置必须与该位置的输出弧上的变量一致，以确保变迁顺利进行；

(4)S 是一个非空的有色集，称为颜色集；

(5)G 是一个守卫函数，定义在 T 上，其中 $\forall t\in T$ 有 $Type[G(t)]=Boolean\wedge Type(Var(G(t)))\subseteq S$；

(6)E 是一个弧表达式函数，定义在 F 上，其中 $\forall f\in F$ 有 $Type[E(f)]=C(p)_{MS}\wedge Type(Var(E(f)))\subseteq S$；

(7)C 是一个颜色函数，定义为 $\forall p\in P$ 都有 $C(p)\in S$；

(8)t 代表某一刻时间。

分析 API 供应链中应满足的安全属性后，抽象模型的属性约束如下：

(1)销售点处交易的石油单价必须遵循监管部门对价格的规定。

(2)当供应商供给的石油数量少于销售点的需求量时，交易终止。

(3)当销售点的账户余额不足时，该用户无法进行采购石油、支付等操作。

(4)一旦销售点处收到供应商的产品，销售点必须及时付款给对方。

(5)供应商账户余额和销售点的账户余额的总和是固定值。

(6)供应商每日石油产量必须在监管部门设定的阈值之下。

(7)当供应商的供给量多于销售点需求量一定数量时，监管部门需要重新调整供应商的每日产量限制和交易的石油单价。

在后续的模型分析过程中，我们通过验证该模型是否满足上述安全属性来判断所建模型的逻辑是否正确。

3.2　形式化建模

3.2.1　顶层模型

顶层模型如图 1 所示，库所 Vendor 代表供应商，库所 POS 代表销售点，Payment 和 Update 是两个替换变迁，分别表示供应商和销售点的转账交易以及监管部门更新指标的状态变化。

3.2.2　Payment 模型

Payment 模型是 Payment 替换变迁的底层实现，如图 2 所示，该模型有 15 个库所和 4 个变迁，其中，Vendor 和 POS 是连接顶层的接口。库所 OilBatchID 是供应商提供的某批次石油 ID，V_Amount 库所是该批次石油的数量，V_Balance 库所是供应商账户余额，SoldBatchID 库所是销售点收到的某批次石油 ID，S_Amount 库所是销售点的石油需求量，Price 库所是由监管部门设定的石油交易单价，S_Balance 库所是销售点的账户余额。假定销售点未进行交易时的 soldBatchID＝“#”。

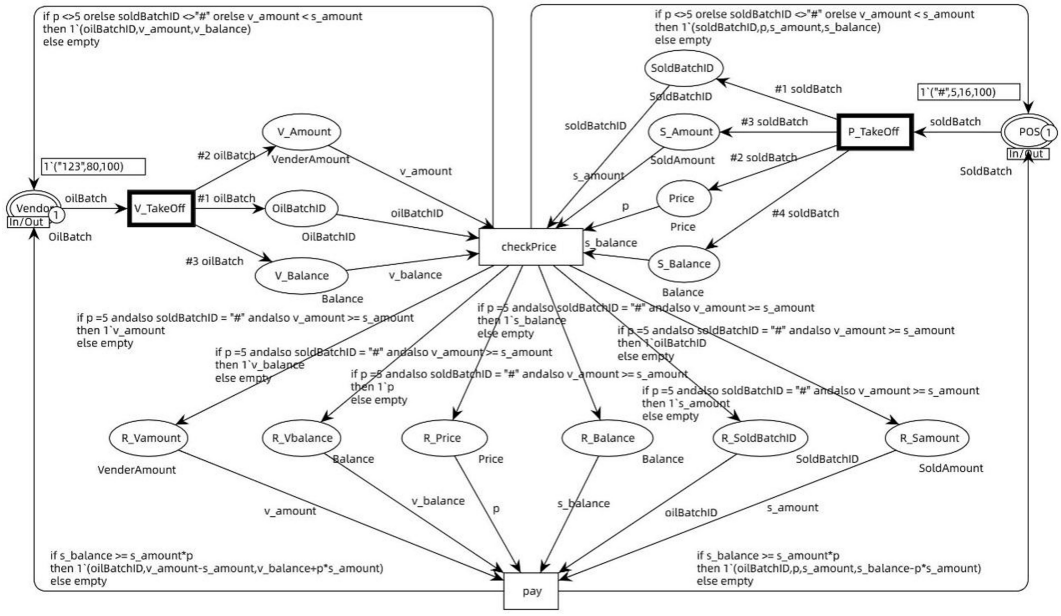

图 2　Payment 模型

Fig. 2　Model of Payment

基于 Payment 模型，使用 CPN 仿真工具执行一次供应商与销售点的转账交易过程，包括以下步骤：

步骤 1　初始标识为 M_0，$M_0[V\_TakeOff\rangle M_1$。V_TakeOff 变迁发生，状态从 M_0 转为 M_1。此时，$E(Vendor, V\_TakeOff)\langle oilBatch\rangle = 1'\{oilBatchID=\text{“123”}, v\_amount=80, v\_balance=100\}$，$G(V\_TakeOff)=\text{true}$。该变迁的输入弧绑定了供应商的某批 $oilBatchID$ 为“123”的石油批次变量，使得供应商的该批次石油信息被提取。

步骤 2　$M_1[P\_TakeOff\rangle M_2$。P_TakeOff 变迁发生，状态从 M_1 转为 M_2。此时，$E(POS, P\_TakeOff)\langle soldBatch\rangle = 1'\{soldBatchID=\text{“\#”}, price=5, s\_amount=16, s\_balance=100\}$，$G(P\_TakeOff)=\text{true}$。该变迁的输入弧绑定了销售点账户的信息，变迁点火后信息被提取。

步骤 3　$M_2[checkPrice\rangle M_3$。checkPrice 变迁点火，状态从 M_2 转为 M_3。此时，供应商的账户信息、待销售的一批石油信息、销售点的账户信息以及销售点的需求情况等信息都作为该变迁的输入数据，$G(checkPrice) < p=5\ andalso\ soldBatch=\text{“\#”}\ andalso\ v\_amount >= s\_amount >= \text{true}$。这里将判断业务流程是否满足属性约束(1)和约束(2)，以及判断该批石油是否处于未出售状态，只有这 3 个约束条件均满足时，checkPrice 变迁才能发生。因为销售点的交易单价符合监管部门的价格 5，供应商的供应量大于需求量，所以该有色网能继续执行。

步骤 4　$M_3[pay\rangle M_4$。pay 变迁点火，状态从 M_3 转为 M_4。此时，依旧在输入弧上传递交易参与方的所有信息，$G(Pay)=\text{true}$。这里的输出弧将判断销售点的账户余额是否大于该批石油的总价，若满足条件即满足属性约束(3)，则销售点收货且付款给供应商，将销售点的 80 个资产转给供应商。若销售点账户及时付款且各账户资金变化没有错误，则满足属性约束(4)和约束(5)。

因此，一次供应商与销售点的转账交易对应的安全发生序列为($M_0M_1M_2M_3M_4$)。若一次交易完成，则无法到达 M_2 状态；若销售点的交易单价不符合监管部门规定的价格，或者供应商的供应量小于需求量，则无法到达 M_3 状态；若销售点的账户余额不足以支付，则无法到达 M_4 状态。

3.2.3　Update 模型

Update 模型是 Update 替换变迁的底层实现，通过该模型检查供应商和销售点在完成一次交易后的供需情况，以便监管部门及时更新生产数量和交易定价规则。如图 3 所示，该模型有 12 个库所和 3 个变迁，其中，Vendor 和 POS 是连接顶层的接口。Prod_Limit 库所是供应商每日石油总产量的最大阈值。FixedPrice 库所是监管部门所设定的销售点必须遵循的石油单价。checked 库所容纳的令牌为布尔类型，用于记录交易是否完成检查。Need 库所容纳的令牌也是布尔类型，用于记录监管部门是否需要更新生产和买卖规则。其余 6 个库所是供应商和销售点的账户信息以及进行的某笔交易的状态信息。该模型在 Payment 模型成功执行以后才可发生，其

初始标识如下：

-var oilBatch=1′{oilBatchID="123", v_amount=64, v_balance=180};

-var soldBatch=1′{soldBatchID="123", price=5, s_amount=16, s_balance=20};

-var c=1′unchecked;

-var n=1′no;

-var prod_limit=1′500;

-var fix_price=1′5;

基于 Update 模型，使用 CPN 仿真工具执行一次检查并更新石油产量和交易单价的操作，包括以下步骤：

步骤 1 $M_4[U\_TakeOff\rangle M_5$。U_TakeOff 变迁点火，状态从M_4转为M_5。此时 $E(Vendor, U\_TakeOff)\langle oilBatchID\rangle$="123"，$E(POS, U\_TakeOff)\langle soldBatchID\rangle$="123"，$E(Checked, U\_TakeOff)\langle c\rangle = unchecked$，$G(U\_TakeOff)\langle oilBatchID = soldBatchID\ andalso\ c = unchecked\rangle$ = true。因为输入弧上绑定的变量能使 U_TakeOff 变迁，所以能够到达下一个标识M_5。

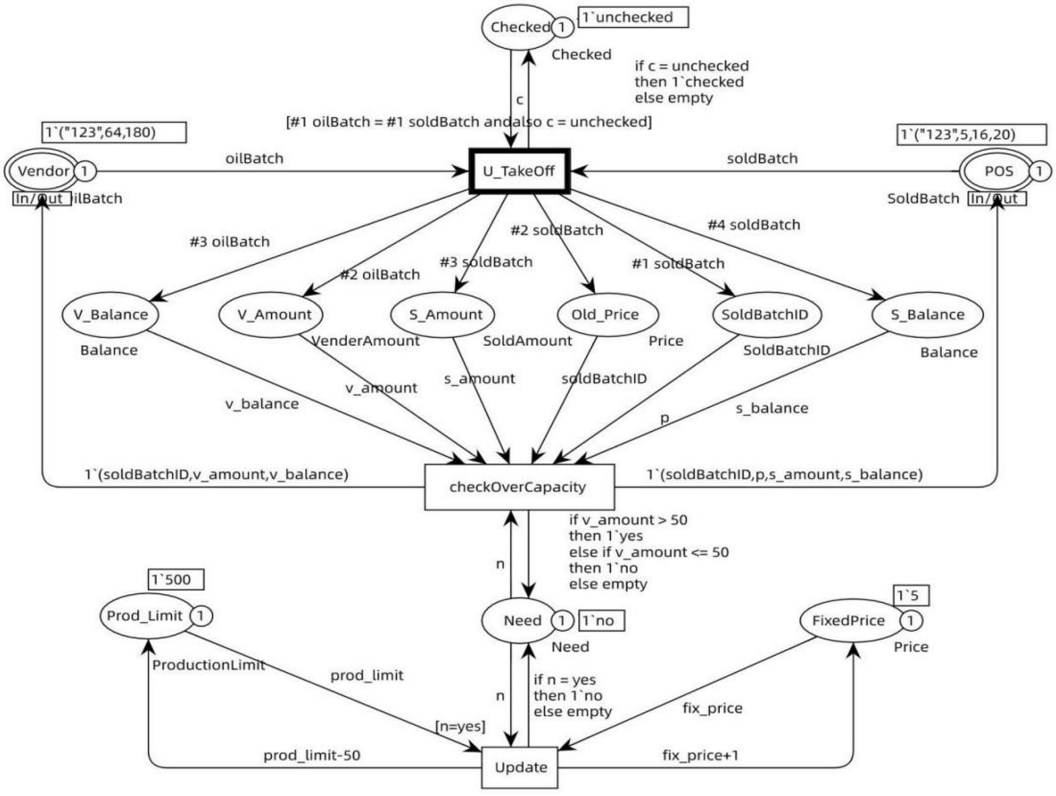

图 3 Update 模型

Fig. 3 Model of Update

步骤 2 $M_5[checkOverCapacity\rangle M_6$。CheckOverCapacity 变迁发生，状态从M_5转为M_6。此时经过一次交易后的 *v_amount* 变量的值为供应商剩余的产品数量，若它大于监管部门设定的供应量超出需求量的最大阈值，则供需不平衡。在输出弧上输出一个值为 *yes* 的变量，表示需要更新买卖和交易规定。

步骤 3 $M_6[Update\rangle M_7$。Update 变迁点火，状态从M_6转为M_7。此时 $E(Prod\_Limit, Update)<prod\_limit)=500$，$E(FixedPrice, Update)\langle fix\_price\rangle=5$，$E(Need, Update)\langle n\rangle = yes$，$G(Update)\langle n=yes\rangle$ = true。变迁发生后，供应商每日产量的阈值减小为 450，石油单价上调为 6。该步骤满足属性约束(3)。

因此，监管部门更新指标的过程对应的安全发生序列为($M_4M_5M_6M_7$)。若交易供需平衡时，无需调整监管部门设定的供应量超出需求量的最大阈值，则无法到达M_6状态；供需平衡的情况下也无需调整供应商每日产量和交易价格，无法到达M_7状态。

4 模型检查

在本节中，使用 CPN Tools 中的 State Space Tools 分析所建 CPN 模型的可达性、有界性、活性等关键动态性质，其行为路径可通过可达标识图来表达，以分析网系统中的状态变化和变迁发生序列的情况，验证模型的正确性从而确保模型对应语义的智能合约的安全性。

性质 1(活性) 通过表 1 所列的状态空间报告可以看到该模型没有死标识和死变迁，所有变迁都能被点火，说明模型不存在死锁。P_TakeOff, V_TakeOff, checkPrice 是 3 个活变迁，即在任何可达标识下，这 3 个变迁都存在于可发生序列中。P_TakeOff 和 V_TakeOff 变迁用于提取参与方的相关信息，在交易之前点火 checkPrice 变迁判断参与方的交易数据是否满足标准条件。实际供应链系统中只要有正在交易的参与方，就需要进行该条件判断。因此，模型中的这 3 个活变迁是合理的。

性质 2(有界性) 由于篇幅所限，表 1 列出了部分库所内令牌数的最大值和最小值，完整的报告中所有库所均有界，即 CPN 模型可以映射为一个有限状态机。

性质 3(公平性) 该性质反映两个变迁之间的相互关系，报告表明 CPN 模型满足公平性，没有无尽发生序列，反映了被模拟的供应链系统部分业务执行流程中没有饥饿性问题。

表 1　CPN 状态空间报告

Table 1　State space report of CPN

Liveness Properties	Boundedness Properties			Fairness Properties
Dead Markings	Best Integer Bounds			No infinite occurrence sequences
None		Upper	Lower	
Dead Transition Instances	Payment'S_Amount 1	1	0	
None	Payment'S_Balance 1	1	0	
Live Transition Instances	Payment'SoldBatchID 1	1	0	
Paym ent'P_TakeOff 1	Payment'V_Amount 1	1	0	
Payment'V_TakeOff 1	Payment'V_Balance 1	1	0	
Payment'checkPrice 1	Top'POS 1	1	0	
	Top'Vendor 1	1	0	
	Update'Checked 1	1	1	
	Update'FixedPrice 1	0	0	
	Update'Need 1	1	1	
	Update'Prod_Limit 1	1	1	
	Update'S_Amount 1	0	0	
	Update'S_Balance 1	0	0	
	Update'SoldBatchID 1	0	0	

从图 4 所示的可达标识图可以看到模型没有发生无限序列。每个节点中上方的数字是节点编号，下方冒号前的数字表示该节点前驱节点的个数，冒号后的数字表示该节点后继节点的个数。状态 1 时，供应商账户资产为 100，该批石油供给量为 80。销售点账户资产为 100，石油需求量为 16，监管部门规定的石油单价为 5。由于还未进行交易，因此销售商未获得该批石油的 ID，soldBatchID 为“123”。直到转账操作成功后，模型到达状态 6。此时，销售商获取到了这批石油的 ID，供应商和销售商的资产分别为 180 和 20，而 API 供应链合约总资产为 200。通过分析，Payment 模型的转账过程是正确的。再进行更新操作后，模型到达状态 12，此时供应商的每日最大产量限制和销售点处的石油买卖单价都得到正确的更新。通过分析所建模型状态空间，可知该模型仿真的业务流程是满足功能需求并且逻辑正确的。

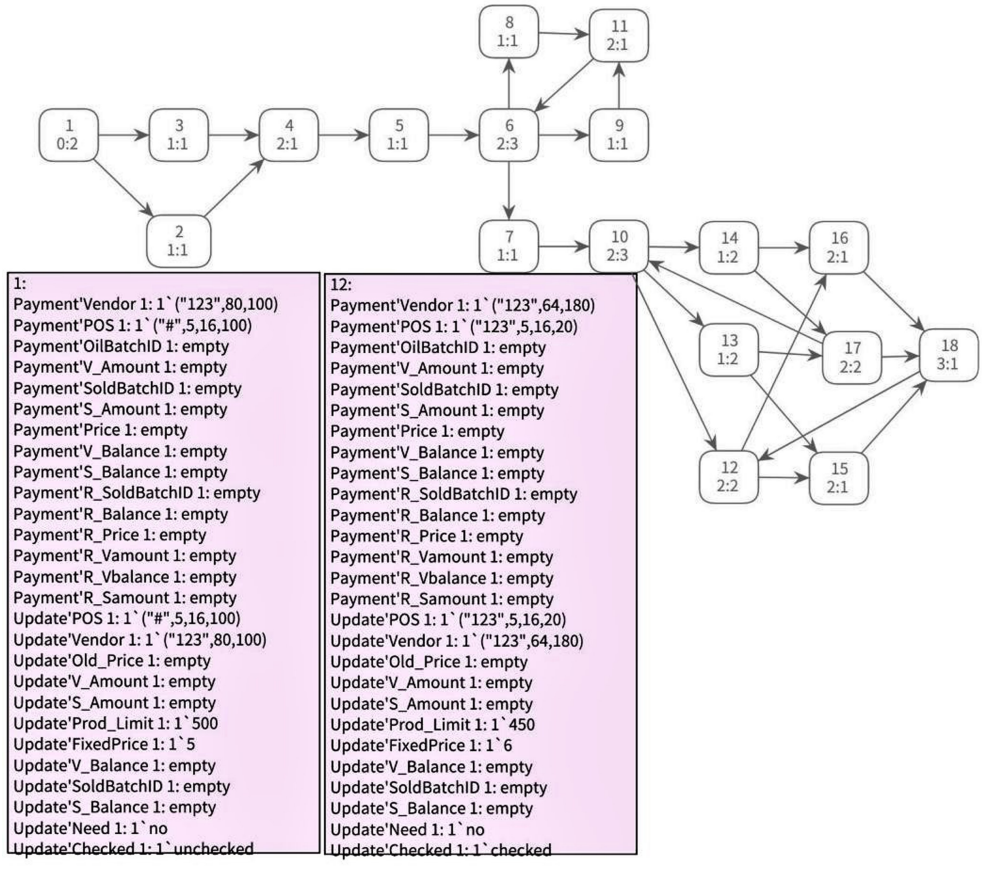

图 4　可达标识图

Fig. 4　Reachable marking graph

5　合约实现

本文提出的模型描述了石油供应链系统中面向供应商和销售点的一个简单交易流程，经过模型正确性分析后，可将模型转换为供应商的一部分安全业务合约代码。

将 CPN 模型转换为 Solidity 合约代码的过程主要包括

两部分，一部分是颜色集到合约变量之间的转换；另一部分是变迁和相应防卫表达式到合约函数的转换，即控制流程的转换与处理。

模型中颜色集表示变量数据结构的描述，令牌的流转表示程序中变量的值的传递。对于简单有色集，直接将颜色集声明转换为对应类型的合约状态变量。例如 CPN 中表示某账户余额的 BALANCE 颜色集，声明的数据类型为 uint，定义的 var 变量为 balance，则将该颜色集转换为 Solidity 代码中的 uint 类型的变量 balance。此外，CPN ML 中还有 globref 类型的变量，将其转换为 Solidity 语言中的 constant 类型的常量。对于复杂颜色集，将其转换为 Solidity 合约中的结构体或数组。例如表示销售点采购的某批次石油的颜色集 SoldBatch，声明的结构是 SoldBatchID * p:Price * b:Balance，即一个 record 类型的多元组，则可以把该颜色集转换为 Solidity 代码中的结构体 SoldBatch，其中包含的变量类型与组成复杂颜色集的各单颜色集下的变量一一对应。

模型中所有的发生序列能够表示部署链上的供应链合约的执行变化，接下来分析如何将模型中带防卫表达式的变迁以及与变迁相连的输入弧和输出弧表达式映射为合约中的函数，即控制流程的转换。

任何 Petri 网模型都由顺序、并发、选择和循环 4 种基本结构及其组合构成，因此，模型中这 4 种基本结构的转换是合约代码中方法的转换的基础。以模型中的选择结构为例，只要该结构中的输入库所含有 Token，那么任意分支均处于使能状态，根据防卫表达式的限定，选择一个分支触发。图 2 中 Payment 模型中的 checkPrice 变迁相关的选择结构可以转换为图 5 所示 Solidity 代码中的条件语句。若使能 checkPrice 变迁，Token 经过输出弧表达式 $p=5$，即销售点的石油交易单价符合监管部门设定的价格 5，那么转换为第 5 行的 else 表达式；否则，Token 经过输出弧表达式 $p<>5$，转换为第 3 行的 if 表达式，表示不满足形式化规范的安全属性(6)。弧表达式改变 Token 值并将数据流转到下一个状态，映射为对应条件下的语句，并给 msg.sender 即调用者地址发送交易是否符合规范的通知。修改相应变量的值即改变参与方的状态。

```
1.      function checkPrice(string calldata soldBatchID, int mainFixedPrice) public {
2.          int price = soldBatchMap[soldBatchID].price;
3.          if (price != mainFixedPrice) {
4.              emit info(msg.sender, "Violation of gallon price.");
5.          } else {
6.              emit info(msg.sender, "No violations.");
7.          }
8.      }
```

图 5 API 供应链合约的 checkPrice 方法

Fig. 5 checkPrice method of API supply chain contract

结束语 区块链技术在供应链领域的应用得到越来越多的关注。针对目前智能合约漏洞引起攻击事件频发的现状，本文提出了一个安全开发供应链智能合约的形式化方法。首先，结合区块链的运行机制和交易特点，采用 CPN 对 API 供应链系统的业务逻辑进行形式化规范与分层建模仿真，给出基于区块链技术的 API 供应链的业务流程的动态表达，以图形化界面直观地描述智能合约执行的交易状态变化，以方便跟踪所有执行路径。其次，通过形式化安全验证，分析出模型满足安全属性的发生序列。最后，给出从 CPN 模型直接转换为部署以太坊的 Solidity 合约代码的方法，提高了编写智能合约的安全性。此外，本文侧重于对智能合约的需求规范开发进行功能验证。考虑到智能合约能够被外部账户和其他合约调用，黑客可能利用该特性进行攻击，后续将在最终的集成开发中对合约交互进行安全性验证。

参 考 文 献

[1] DHILLON V, METCALF D, HOOPER M. The DAO hacked [M]// Blockchain Enabled Applications. 2017: 67-78.

[2] STEINER J. Security is a process: A postmortem on the parity multi-sig library self-destruct [EB/OL]. https://www.parity.io/blog/a-postmortem-on-the-parity-multi-sig-library-self-destruct.

[3] ATZEI N, BARTOLETTI M, CIMOLI T. A survey of attacks on ethereum smart contracts (sok) [C]// International Conference on Principles of Security and Trust. Berlin: Springer, 2017: 164-186.

[4] ZHANG X H, SUN L L. Attribute Proxy Re-encryption for Ciphertext Storage Sharing Scheme on Blockchain [J]. Journal of System Simulation, 2020, 32(6): 1009-1020.

[5] JENSEN K, KRISTENSEN L M, WELLS L. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems [J]. International Journal on Software Tools for Technology Transfer, 2007, 9(3): 213-254.

[6] LI H, SUN T, WANG X R, et al. Testing the concurrent behavior of systems based on CPN [J]. Computer Science, 2016, 43(1): 218-225.

[7] The American PetroleumInstitute(API). Energy: Understanding Our Oil Supply Chain [EB/OL]. https://www.api.org/-/media/Files/Policy/Safety/API-Oil-Supply-Chain.pdf.

[8] HIRAI Y. Defining the ethereum virtual machine for interactive theorem provers [C]// International Conference on Financial Cryptography and Data Security. Cham: Springer, 2017: 520-535.

[9] AMANI S, BÉGEL M, BORTIN M, et al. Towards verifying ethereum smart contract bytecode in Isabelle/HOL [C]// Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs. 2018: 66-77.

[10] NEHAI Z, PIRIOU P Y, DAUMAS F. Model-checking of smart contracts [C]// 2018 IEEE International Conference on Internet of Things (iThings) and IEEE GreenComputing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). IEEE, 2018: 980-987.

[11] WANG Y, LAHIRI S K, CHEN S, et al. Formal specification and verification of smart contracts for azure blockchain [J]. arXiv: 1812.08829, 2018.

[12] KALRA S, GOEL S, DHAWAN M, et al. Zeus: Analyzing safe-

ty of smart contracts[C]// Ndss. 2018:1-12.

[13] MORTENSEN K H. Automatic code generation method based on coloured petri net models applied on an access control system [C]// International Conference on Application and Theory of Petri Nets. Berlin:Springer,2000:367-386.

[14] PHILIPPI S. Automatic code generation from high-level Petri-Nets for model driven systems engineering[J]. Journal of Systems and Software,2006,79(10):1444-1455.

[15] WANG D,HUANG X,MA X F. Formal analysis of smart contract based on colored petri nets[J]. IEEE Intelligent Systems,2020,35(3):19-30.

[16] LIU Z,LIU J. Formal verification of blockchain smart contract based on colored petri net models[C]// 2019 IEEE 43rd Annual Computer Software and Applications Conference(COMPSAC). IEEE,2019:555-560.

[17] DONG C Y,TAN L. Formal Validation of Auction Smart Contract Based on CPN Model[J]. Journal of Chinese Computer Systems,2020,41(11):2292-2297.

[18] ZHENG H,LIU Z R,HUANG J H,et al. Verification of Transaction Ordering Dependence Vulnerability of Smart Contract Based on CPN[J]. Journal of System Simulation,2022,34(7):1629-163.

ZHENG Hong, born in 1973, Ph.D, associate professor, master supervisor, is a member of China Computer Federation. Her main research interests include formal verifivation and blockchain.