Liu Y, Miao HK, Zeng HW *et al.* Nondeterministic probabilistic Petri net — A new method to study qualitative and quantitative behaviors of system. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 28(1): 203–216 Jan. 2013. DOI 10.1007/s11390-013-1323-7

Nondeterministic Probabilistic Petri Net — A New Method to Study Qualitative and Quantitative Behaviors of System

Yang Liu[1,2,3] (刘　阳), Huai-Kou Miao[1] (缪淮扣), *Senior Member*, *CCF*, Hong-Wei Zeng[1] (曾红卫)
Yan Ma[2] (马　艳), and Pan Liu[1] (刘　攀)

[1]*School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China*

[2]*School of Information Science and Technology, Taishan University, Taian 271021, China*

[3]*State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210093, China*

E-mail: liuyang_shu@126.com; {hkmiao, zenghongwei}@shu.edu.cn; {mayan1616, panl008}@163.com

Received December 14, 2011; revised November 30, 2012.

Abstract There are many variants of Petri net at present, and some of them can be used to model system with both function and performance specification, such as stochastic Petri net, generalized stochastic Petri net and probabilistic Petri net. In this paper, we utilize extended Petri net to address the issue of modeling and verifying system with probability and nondeterminism besides function aspects. Using probabilistic Petri net as reference, we propose a new mixed model NPPN (Nondeterministic Probabilistic Petri Net) system, which can model and verify systems with qualitative and quantitative behaviours. Then we develop a kind of process algebra for NPPN system to interpret its algebraic semantics, and an action-based PCTL (Probabilistic Computation Tree Logic) to interpret its logical semantics. Afterwards we present the rules for compositional operation of NPPN system based on NPPN system process algebra, and the model checking algorithm based on the action-based PCTL. In order to put the NPPN system into practice, we develop a friendly and visual tool for modeling, analyzing, simulating, and verifying NPPN system using action-based PCTL. The usefulness and effectiveness of the NPPN system are illustrated by modeling and model checking an elaborate model of travel arrangements workflow.

Keywords nondeterminism, probabilistic Petri net, model checking, action-based probabilistic computation tree logic

1 Introduction

Petri net is proposed by Carl Adam Petri in his Ph.D. thesis "Kommunikation mit Automaten" (Communication with Automata) in 1962. At present, it has been used in many application areas successfully. As a system model to describe the function specification, the advantages of Petri net for the modeling of systems are well-known[1]: it provides a graphically and mathematically founded modeling formalism, which is in contrast to many similar techniques, where only one of these properties is well developed and the other is added in a less systematic way; to date there exists a huge variety of algorithms for the design and analysis of Petri nets and powerful computer tools have been developed to aid this process; it provides mechanisms for abstraction and refinement that are well integrated into the basic model; different variants of Petri net models have been developed that are all related by the basic net formalism which they are built upon. The variants of Petri net models can be summarized as two-dimensional (2-D) direction developments[2]. The one is the vertical direction which is from condition/event Petri net (C/E net) to place/transition Petri net (P/T net), then to high level Petri net (HL net) (e.g., predicate/transition Petri net, colored Petri net, and fuzzy Petri net); and the other is the horizontal direction which is from Petri net without parameter to timed Petri net (TPN)[3-4], stochastic Petri net (SPN)[5-7]and generalized stochastic Petri net (GSPN)[8], from ordinary directed arc to inhibitory arc and alterable arc, from atomic transition to sub-net transition, and so on. The guideline of Petri net developments is the general net theory[9] which was put forward by Carl Adam Petri in the 1980s. The development process and the general net theory of Petri net are shown in Fig.1, and our research work focuses

Regular Paper

This work was supported by the National Natural Science Foundation of China under Grant Nos. 60970007, 61073050 and 61170044, the National Basic Research 973 Program of China under Grant No. 2007CB310800, the Shanghai Leading Academic Discipline Project of China under Grant No. J50103, and the Natural Science Foundation of Shandong Province of China under Grant No. ZR2012FQ013.

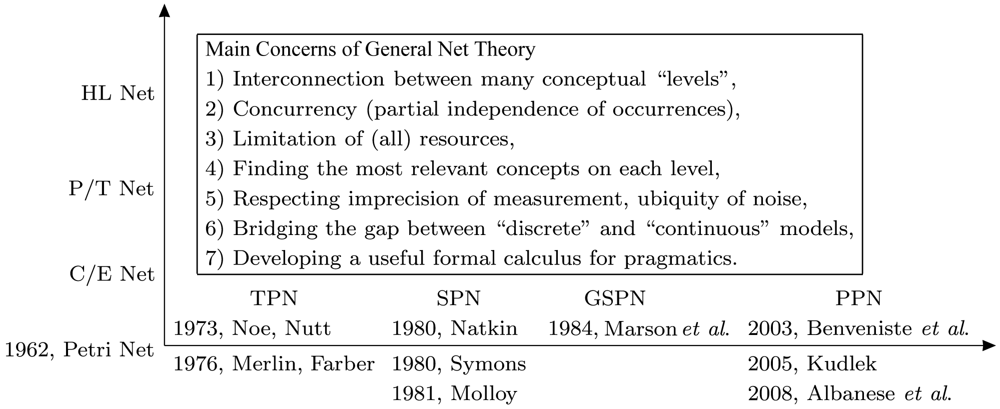

Main Concerns of General Net Theory
1) Interconnection between many conceptual "levels",
2) Concurrency (partial independence of occurrences),
3) Limitation of (all) resources,
4) Finding the most relevant concepts on each level,
5) Respecting imprecision of measurement, ubiquity of noise,
6) Bridging the gap between "discrete" and "continuous" models,
7) Developing a useful formal calculus for pragmatics.
HL Net
P/T Net
C/E Net
1962, Petri Net
TPN
1973, Noe, Nutt
1976, Merlin, Farber
SPN
1980, Natkin
1980, Symons
1981, Molloy
GSPN
1984, Marson *et al.*
PPN
2003, Benveniste *et al.*
2005, Kudlek
2008, Albanese *et al.*

Fig.1. Development process of Petri net and the general net theory.

on the probabilistic Petri net (PPN) of the horizontal coordinate.

In recent decades, time parameters are introduced into Petri net in various ways, which make its descriptive ability extend to both function and performance of the system from only function aspect. There are two most common ways to introduce the time parameter into Petri net. One way is to associate a time parameter with the place; the other is to associate a time parameter with the transition of Petri net. A Petri net is called stochastic Petri net, when it associates a time parameter with the period of the transition from being enabled to firing. The original stochastic Petri net proposal assumes atomic firings, exponentially distributed firing times, and a race execution policy. That is to say, when multiple transitions are simultaneously enabled, the race policy selects the transition with the statistical minimum delay to fire[10]. However, there are some situations where we need to account for the probabilistic behavior rather than keep track of time. Many systems exhibit probabilistic behavior, either added explicitly to assess or predict the performance, or implicitly, through interaction with the environment. So, probability aspects are essential for, among others: randomized algorithm, modeling unreliable and unpredictable system behavior, and model-based performance evaluation[11]. For that reason, probabilistic Petri net is proposed in [12-14][1]. In the theory of probabilistic Petri net, probability is associated with places or arcs, and the transition is described in some probability distribution. Therefore, the transition firing of probabilistic Petri net is no longer nondeterministic, and the nondeterminism is substituted by probability. It is effective to process the sequential transitions in that manner; nevertheless some aspects of a system may not be probabilistic and should not be modeled probabilistically, such as concurrency, underspecification, unknown environments, and abstraction. These aspects listed above are the nondeterminism of a system. The notion of nondeterminism is essential for some distributed system, and it is useful to model different important concepts that could not be expressed by probability. They can be summarized as follows[15]:

1) *Implementation Freedom.* An interactive process can be viewed as an abstract specification, and nondeterminism represents implementation freedom. That is, if for some place there are two transitions that can be chosen nondeterministically, then an implementation may have just one of the two transitions.

2) *Scheduling Freedom.* This is the classical use of nondeterminism in an interleaving semantics. Several processes run in parallel and there is a freedom in the choice of which process performs the next transition.

3) *External Environment.* External actions represent interaction possibilities with some external process, or more generally an external environment, by means of synchronization. The interaction capabilities of this environment influence how the choice is determined.

However, a probabilistic Petri net cannot exhibit nondeterministic behavior, since each possible transition at any time has a specific probability. Therefore, in order to present nondeterminism of probabilistic system model, we make alterations of probabilistic Petri net using the following ways:

1) combine probability with transition orthogonally,

2) maintain the nondeterministic choice of the system.

This paper is further organized as follows: Section 1 provides some theoretical concepts about both Petri net and probability measure theory. In Section 2, we define the nondeterministic probabilistic Petri net system and

[1] Another definition for probabilistic Petri net is: Varacca D, Nielsen M. Probabilistic Petri nets and Mazurkiewicz equivalence (Draft), February 2003. http://www.pps.univ-paris-diderot.fr/~varacca/papers/shortpetri.ps.

its algebraic and logical semantics. In Section 3, we present how to compose operations of nondeterministic probabilistic Petri net system. Section 4 designs the algorithm of model checking nondeterministic probabilistic Petri net system. In Section 5, we apply the nondeterministic probabilistic Petri net system to model and model check a nontrivial example. The paper ends with a short summary and conclusions.

2 Preliminaries

2.1 Related Concepts of Petri Net

Definition 1 (Directed Net). *A directed net is a triple $N = (S, T; F)$, the necessary and sufficient conditions of which are as follows*:

1) $S \cap T = \phi$,

2) $S \cup T \neq \phi$,

3) $F \subseteq S \times T \cup T \times S$,

4) $dom(F) = \{x | \exists y : (x, y) \in F\}$,
$cod(F) = \{y | \exists x : (x, y) \in F\}$,
$dom(F) \cup cod(F) = S \cup T$,

where S-element and T-element are called the places set and transitions set respectively, F-element is called the flow relation for the set of arcs, $dom(F)$ is the domain of F, and $cod(F)$ is the range of F.

Definition 2 (Net System). *A net system is defined as a tuple $\Sigma = (S, T; F, K, W, M_0)$, where*

1) *$N = (S, T; F)$ is a directed net, and called the basic net of Σ,*

2) *K, W, and M_0 is the capacity function, weight function, and initial marking respectively.*

Actually, in 1962, the system model used by Carl Adam Petri is the net system of $K = \omega$ and $W = 1$[16]. In the 70s of the 20th century, Holt named this system model as the Petri net, and at the same time, the net and net system were referred as Petri nets without distinction[16]; while unless otherwise mentioned, all through this paper we distinguish between the net and the net system using the names of them.

Definition 3 (Pure Net). *N is a pure net, $X = S \cup T$, if $\forall x \in X$: ${}^{\bullet}x \cap x^{\bullet} = \phi$ for ${}^{\bullet}x = \{y | (y, x) \in F\}$ and $x^{\bullet} = \{y | (x, y) \in F\}$.*

2.2 Measure Theory

Definition 4 (σ-Algebra). *A σ-algebra is a pair $(Outc, \zeta)$ where $Outc$ is a nonempty set and $\zeta \subseteq 2^{Outc}$ is a set consisting of subsets of $Outc$ that contains the empty set and is closed under complementation and countable unions.*

The elements of *Outc* are often called outcomes, while the elements of ζ are called events.

Definition 5 (Probability Measure). *A probability measure on $(Outc, \zeta)$ is function P: $\zeta \rightarrow [0, 1]$ such that $P(Outc) = 1$, and if $(E_n)_{n\geqslant 1}$ is a family of pairwise disjoint events $E_n \in \zeta$, then $P(\bigcup\limits_{n\geqslant 1} E_n) = \sum\limits_{n\geqslant 1} P(E_n)$.*

Definition 6 (Probability Space). *A probability space[17-18] is a σ-algebra equipped with a probability measure, i.e., it is a triple $(Outc, \zeta, P)$ where $(Outc, \zeta)$ is a σ-algebra and P a probability measure on $(Outc, \zeta)$.*

3 Nondeterministic Probabilistic Petri Net and Its Semantics

3.1 Definition of Nondeterministic Probabilistic Petri Net

Nondeterministic probabilistic Petri net is a kind of extended Petri net, in which transitions are endued with probability, and it keeps the characteristic of nondeterminism. The formal definition of it is as follows.

Definition 7 (Nondeterministic Probabilistic Petri Net). *A nondeterministic probabilistic Petri net (NPPN) is defined as a tuple $N = (S, T; F, f; C)$, where*

1) *$T = (Trans, Pt)$, Trans denotes the transition act, $Pt \in [0, 1]$, which denotes the success probability of the transition; T is the act transition (AT) with the probability equaling 1, or the probabilistic act transition (PAT) with an act satisfying a certain probability distribution, or the probability transition (PT) without any act. If $Pt = 0$, it means the transition is invalid.*

2) *$S \cap T = \phi$, $S \cup T \neq \phi$, $F \subseteq S \times T \cup T \times S$, which is the flow relation of net, and $N = (S, AT, F)$ is the pure net, where AT is the act transition with the probability equaling 1.*

3) *$f = f_T \cup f_S \cup f_{S\times T} \cup f_{T\times S}$, f_T: $T \rightarrow 2^{Trans\times Pt}$, f_S: $S \rightarrow [0, 1]$, $f_{T\times S}$: $T \times S \rightarrow [0, 1]$, $f_{S\times T}$: $S \times T \rightarrow [0, 1]$, and the value of $f_{S\times T}$ is determined by the nondeterminism of transitions, the value of $f_{T\times S}$ can be obtained from the value of $\sigma_{Pt}(f_T(t))$, the value of f_S except the initial place can be computed according to the value of $f_{S\times T}$ and $f_{T\times S}$.*

4) *$\forall t \in T$, $\exists s \in S$ $f_{T\times S}(t \times s) = 1 - \sigma_{Pt}(f_T(t))$. $f_{T\times S}(t \times s) = 0$, if $\sigma_{Pt}(f_T(t)) = 1$, which represents flow relation (t, s) and place s are invalid.*

5) *C is the set of nondeterminism classes, and each nondeterminism class is a set comprised of (s, t_i). If $\{(s, t_1), (s, t_2), \ldots, (s, t_n)\} \in C$, then $\sum\limits_{i=1}^{n} f_{S\times T}(s, t_i) = 1$.*

N is called finite if S and T are finite, and the size of N is the number of places and transitions plus the number of pairs (s, t) with $f_{S\times T} > 0$ and (t, s) with $f_{T\times S} > 0$. Nondeterministic probabilistic Petri net can be viewed as a variant of Petri net that permits both probabilistic and nondeterministic choices.

Definition 8 (Nondeterministic Probabilistic Petri Net System). *A nondeterministic probabilistic Petri* *net system (NPPN system) is defined as a tuple* $\sum = (S, T; F, f; C, K, W, M_0)$, *where*

1) $N = (S, T; F, f; C)$ *is the nondeterministic probabilistic Petri net, and* N *is called the basic net of* $\sum$;

2) K, W, M_0 *are the capacity function, weight function, and marking function respectively.* M_0 *is the special case of the marking function* M, *and* $K = 1$, $W = 1$, $M \leqslant 1$;

3) $\forall s_i \in S$, $\exists M_0(s_i) = 1$, *and* $\sum\limits_{i=1}^{n} f_S(s_i) = 1$[2].

The relationship between the NPPN system and Petri net system is as follows: If $\sigma_{Pt}(f_T(t)) = 1 \wedge f_S(s) = 1 \wedge f_{T\times S}(t \times s) = 1 \wedge f_{S\times T}(s \times t) = 1$ and $\{(s,\ t)\} \in C$ are true for $\forall t \in T$, $\forall s \in S$, NPPN is the classical Petri net. And the relationship between the NPPN system and probabilistic Petri net system is as follows: If $\sigma_{Pt}(f_T(t)) = 1 \wedge f_S(s) = 1 \wedge f_{T\times S}(t \times s)$ is true for $\forall t \in T$, $\forall s \in S$, and $\exists (s \times t) \in (S \times T) \sum f_{S\times T}(s \times t) = 1$ is true for $\forall s \in S$, NPPN is the probabilistic Petri net.

Definition 9 (Transition Firing Rule). *The transition firing rules of NPPN system are*:

1) $\forall t \in T$, t *is enabled at the marking* M, *represented by* $M[t>$, *if* $s \in^{\bullet} t$, *then* $M(s) = 1$ *for* $\forall s \in S$;

2) $\forall t \in T$, t *is enabled iff* $\sum_{s\in S} f_{T\times S}(t \times s) = 1$;

3) $\sigma_{Pt}(f_T(t))$ *is related to the current place* s *and the transition* t *itself which* $s \xrightarrow{t} s'$, *but not other places or transitions which can reach the place* s;

4) *A new marking* M' *is reached from* M *with* t, *such that* $\forall s \in S$,

$$M'(s) = \begin{cases} M(s) - 1, & \text{if } s \in^{\bullet} t - t^{\bullet}, \\ M(s) + 1, & \text{if } s \in t^{\bullet} -^{\bullet} t \text{ and} \\ & f_{T\times S}(t \times s) = \sigma_{Pt}(f_T(t)), \\ M(s), & \text{otherwise}, \end{cases}$$

or

$$M'(s) = \begin{cases} M(s) - 1, & \text{if } s \in^{\bullet} t - t^{\bullet}, \\ M(s) + 1, & \text{if } s \in t^{\bullet} -^{\bullet} t \text{ and} \\ & f_{T\times S}(t \times s) = 1 - \sigma_{Pt}(f_T(t)), \\ M(s), & \text{otherwise}. \end{cases}$$

For computing the probability after transition firing, we divide the NPPN system into three basic structures, and present the calculation formulae of probability respectively.

1) *Nondeterministic Choice.* The sum of a certain class of nondeterministic choice probability is 1. As shown in Fig.2, let t_1 and t_2 belong to one nondeterministic choice class α, and the remainders belong to the other class β, the calculation formulae of place probability are:

$$f_S(s_k) = f_S(s) \times f_{S\times T}(s \times t_k) \times f_{T\times S}(t_k \times s_k),$$

where $k = 1, \ldots, n$, $f_{T\times S}(t_k \times s_k) = \sigma_{Pt}(f_T(t_k))$, $f_{S\times T}(s \times t_1) + f_{S\times T}(s \times t_2) = 1$ and $\sum\limits_{i=3}^{n} f_{S\times T}(s \times t_i) = 1$;

$$f_S(s'_k) = f_S(s) \times f_{S\times T}(s \times t_k) \times f_{T\times S}(t_k \times s'_k),$$

where $k = 1, \ldots, n$, $f_{T\times S}(t_k \times s'_k) = 1 - \sigma_{Pt}(f_T(t_k))$, $f_{S\times T}(s \times t_1) + f_{S\times T}(s \times t_2) = 1$ and $\sum\limits_{i=3}^{n} f_{S\times T}(s \times t_i) = 1$.

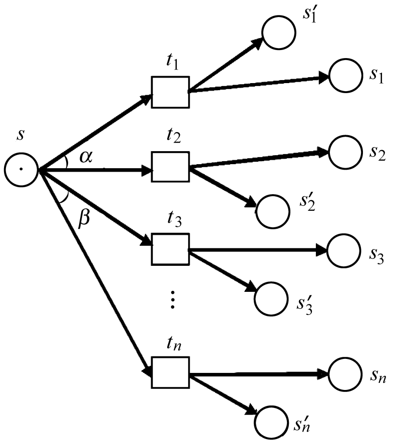


Fig.2. Nourendeterministic choice structure.

2) *Concurrency.* As shown in Fig.3, let $f_S(s) = 1$, the calculation formulae of place probability are:

$$f_S(s_n) = f_S(s) \times f_{S\times T}(s \times t) \times f_{T\times S}(t \times s_k),$$

where $k = 1, \ldots, n$, $f_{S\times T}(s \times t) = 1$, $f_{T\times S}(t \times s_n) = \sigma_{Pt}(f_T(t))$;

$$f_S(s') = f_S(s) \times f_{S\times T}(s \times t) \times f_{T\times S}(t \times s'),$$

where $f_{S\times T}(s \times t) = 1$, $f_{T\times S}(t \times s') = 1 - \sigma_{Pt}(f_T(t))$.

3) *Synchronization.* As shown in Fig.4, the calculation formulae of place probability are:

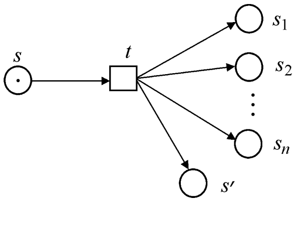


Fig.3. Concurrent structure.

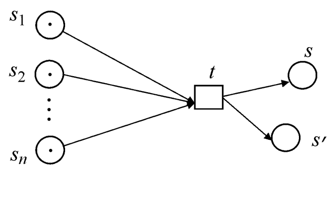


Fig.4. Synchronous structure.

[2] For the sake of simplicity, we just consider a single initial place. Theoretically, the expressive power of NPPN with more than one initial place is equal to that of the NPPN with only one initial place, i.e., the expressive power of NPPN is not associated with the number of initial place.

$$f_S(s)=\prod_{i=1}^{n} f_S(s_i)\times f_{S\times T}(s_i\times t)\times f_{T\times S}(t{\times}s),$$

where $f_{S\times T}(s_i\times t)=f_S(s_i)$, $f_{T\times S}(t{\times}s)=\sigma_{Pt}(f_T(t))$, and $\sum\limits_{i=1}^{n} f_S(s_i)=1$;

$$f_S(s')=\prod_{i=1}^{n} f_S(s_i)\times f_{S\times T}(s_i\times t)\times f_{T\times S'}(t{\times}s'),$$

where $f_{S\times T}(s_i{\times}t)=f_S(s_i)$, $f_{T\times S'}(t{\times}s')=1-\sigma_{Pt}(f_T)$, and $\sum\limits_{i=1}^{n} f_S(s_i)=1$.

3.2 Algebraic Semantics

In order to analyze and operate NPPN system, we develop an algebra to interpret the algebraic semantics for NPPN system, which is called the NPPN system process algebra (NPPNA). NPPNA is the formal semantics of language level of NPPN system, and it is the probability extension of process algebra actually. On the foundation of occurrence net of classical Petri net[16], we present the concept of NPPN system process firstly and then NPPN system process algebra.

Definition 10 (NPPN System Process). *Let $CN=(B,E;Z)$ be the occurrence net, and $\sum=(S,T;F,f;C,K,W,M_0)$ is NPPN system, if $\exists\gamma : CN\to\sum$, such that*

1) $\gamma(B)\subseteq S\wedge\gamma(E)=T\wedge\forall(x,y)\in Z:$

$$\gamma(x,y)=(\gamma(x),\gamma(y))\in F;$$

2) $\forall e\in E:\gamma({}^\bullet e)={}^\bullet\gamma(e)\wedge\gamma(e^\bullet)=\gamma(e)^\bullet$*;*

3) $\forall e\in E:\exists\gamma(e)\in T\wedge f_T(e)=f_T(\gamma(e))$;

4) $\forall b_1,b_2\in B:b_1\neq b_2\wedge\gamma(b_1)=\gamma(b_2)\Rightarrow$ ${}^\bullet b_1\neq{}^\bullet b_2\wedge b_1^\bullet\neq b_2^\bullet$;

5) $\forall s\in S:|\{b|{}^\bullet b=\phi\wedge\gamma(b)=s\}|\leqslant M(s)$;

then (CN,γ) is called the NPPN system process of $\sum$.

On the basis of Definition 10, the NPPN system process algebra is defined as follows.

Definition 11 (NPPN System Process Algebra). *Let pa denote the probability, a denote the act, and A denote the external act, NPPN system process algebra (NPPNA) can be expressed using BNF (Backus Normal Form):* $P::=0|1|pa\cdot P|a\cdot P|(a,pa)\cdot P|P;P|P+P|P\backslash A|P|[A]|P$.

Compared with the traditional process algebra, $pa{\cdot}P$ and $(a,pa){\cdot}P$ are the additional operations of NPPNA, which correspond to probability transition and probabilistic act transition, respectively. There are three forms of prefixal act in NPPNA: 1) a, which is the same with traditional process algebra CCS[19] and CSP[20]; 2) pa, the operational rules of which are shown in Table 1; 3) (a,pa), according to the definition of NPPN, pa can be viewed as (a,pa) with $a=$ null or a is the silent act, the operational rules of which are similar to pa with substituting (a,pa) for pa.

Table 1. Operational Rules for Probability Operator of NPPNA

Atomic Operator	$\dfrac{}{(pa)\cdot P\xrightarrow{pa}P}$	
Sequential Composition	$\dfrac{P_1\xrightarrow{pa}P_1'}{P_1;P_2\xrightarrow{pa}P_1'}$	$\dfrac{P_2\xrightarrow{pa}P_2'}{P_1;P_2\xrightarrow{pa}P_1'}$
Choice Composition	$\dfrac{P_1\xrightarrow{pa}P_1'\wedge P_2\xrightarrow{A}}{P_1+P_2\xrightarrow{pa}P_1'}$	
	$\dfrac{P_2\xrightarrow{pa}P_2'\wedge P_1\xrightarrow{A}}{P_1+P_2\xrightarrow{pa}P_1'}$	
Parallel Composition	$\dfrac{P_1\xrightarrow{pa}P_1'\wedge P_2\xrightarrow{A}}{P_1\lvert[A]\rvert P_2\xrightarrow{pa}P_1'\lvert[A]\rvert P_2}$	
	$\dfrac{P_2\xrightarrow{pa}P_2'\wedge P_1\xrightarrow{A}}{P_1\lvert[A]\rvert P_2\xrightarrow{pa}P_1\lvert[A]\rvert P_2'}$	
Abstract Operator	$\dfrac{P_1\xrightarrow{pa}P_1'}{P_1\backslash A\xrightarrow{pa}P_1'\backslash A}$	

Note: $P_2\xrightarrow{A}$ and $P_1\xrightarrow{A}$ denote process P_2 and P_1 start to execute the external act A respectively.

Based on the operational rules of Table 1, we can derive the expansion formula of parallel operator with the interleaving semantics:

Let $P=\sum_i(a,\ pa)_iP_i$ and $P'=\sum_j(a,\ pa)_jP_j'$, then

$$\begin{aligned}P|[A]|P'=&\sum_{(a,pa)_i\notin A}(a,pa)_i(P_i|[A]|P')+\\&\sum_{(a,pa)_j\notin A}(a,pa)_j(P|[A]|P_j')+\\&\sum_{(a,pa)_i=(a,pa)_j\in A}(a,pa)_i(P_i|[A]|P_j').\end{aligned}$$

3.3 Logical Semantics

3.3.1 *Execution Path of NPPN System and Its Probability Measure*

Execution path is the basis of analyzing the function and performance of NPPN system. We use transition sequence as the base for defining the execution path of NPPN system, and then present how to measure probability for it.

Definition 12 (Transition Sequence)[16].

1) *$M_0t_1M_1t_2M_2\ldots t_nM_n$ is the finite occurrence sequence of NPPN system, if and only if, for $\forall i$, $i=1,2,3,\ldots,n$, $\exists M_{i-1}[t_i>M_i$, where M_i is the marking function, M_0 is the initial marking of the NPPN system, n is called the length of the finite occurrence sequence.*

2) *$M_0t_1M_1t_2M_2 \ldots t_nM_n \ldots$ is the infinite occurrence sequence of NPPN system, if and only if, for $\forall i$, $i = 1, 2, 3, \ldots$, $\exists M_{i-1}[t_i > M_i$.*

3) *If $M_0t_1M_1t_2M_2 \ldots t_nM_n$ is the finite occurrence sequence of NPPN system, we denote finite transition sequence by $\tau = t_1t_2 \ldots t_n$, the length of which is n.*

4) *If $M_0t_1M_1t_2M_2 \ldots t_nM_n \ldots$ is the infinite occurrence sequence of NPPN system, we denote infinite transition sequence by $\tau' = t_1t_2 \ldots t_n \ldots$.*

Definition 13 (Execution Path). *Let $\tau = t_0t_1 \ldots t_n$ be a transition sequence, and let $s_0 \in {}^\bullet t_0$, $s_1 \in t_0^\bullet \cap {}^\bullet t_1, \ldots, s_n \in t_{n-1}^\bullet \cap {}^\bullet t_n$, $s_{n+1} \in t_n^\bullet$, then $\pi = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots \to s_n \xrightarrow{t_n} s_{n+1}$ is an execution path of the NPPN system $\sum$. π is the infinite execution path of $\sum$, if τ is the infinite transition sequence.*

If $s_i = \phi$, i.e., $t_{i-1}^\bullet \cap {}^\bullet t_i = \phi$, then s_i and successive path from s_i are invalid according to the definition of NPPN system. For any execution path π, $\pi[i]$ denotes the $(i+1)$-th place of the path, i.e., $\pi[i] = s_i$.

Definition 14 (Cylinder Set). *The cylinder set of $\hat{\pi} = s_0 \xrightarrow{t_1} s_1 \xrightarrow{t_2} \ldots \to s_k \in Paths_{\mathrm{finite}}(\sum)$ is defined as $Cylinder(\hat{\pi}) = \{\pi \in Paths(\sum) | \hat{\pi} \in pref(\pi)\}$, where $Paths(\sum)$ denotes the set of all infinite execution paths, $Paths_{\mathrm{finite}}(\sum)$ denotes the set of all finite execution paths, and $pref(\pi)$ denotes the set of prefix paths of the path π.*

The σ-algebra associated with $\sum$ is generated by the cylinder set which is spanned by the finite path fragments in $\sum$.

Definition 15 (Probability Matrix Among Places). *If $\exists t(s_i \xrightarrow{t} s_j)$, place s_j is directly reached from s_i, and the probability from s_i to s_j is $Pr(s_i, s_j) = f_{S\times T}(s_i, t) \times f_{T\times S}(t, s_j)$, then the element a_{ij} of probability matrix $\boldsymbol{Ma}$ of the NPPN system can be expressed as follows,*

$$a_{ij} = \begin{cases} Pr(s_i, s_j), & \text{if } \exists t(s_i \xrightarrow{t} s_j), \\ 0, & \text{otherwise}, \end{cases}$$
$$i \in \{0, 1, 2, 3, \ldots, n\}, \quad j \in \{0, 1, 2, 3, \ldots, m\}.$$

Definition 16 (Probability Measure of Execution Path). *Let $Path(s)$ denote the all execution paths from s, $\Omega = Path(s)$ denote the sample space of execution path, and $E = \{\hat{\pi} | \hat{\pi}$ is the prefix of $\pi \wedge \pi \in Path(s)\}$ denote the event set, then the probability of a finite execution path $\hat{\pi} = s_0 \xrightarrow{t_1} s_1 \xrightarrow{t_2} \cdots \xrightarrow{t_k} s_k$ is defined as follows,*

$$P_{s_0}(\hat{\pi}) = \begin{cases} 1, & \text{if } \hat{\pi} = s_0 \\ Pr(s_0, s_1) \times Pr(s_1,\ s_2) \times \cdots \times & \\ Pr(s_{k-1},\ s_k), & \text{if } \hat{\pi} \neq s_0. \end{cases}$$

Definition 17 (Probability Measure of Time-Step). *The probability of starting from place s to s' in k time-steps is defined as*

$$\begin{aligned} \boldsymbol{Prob}_{s,k}(s') &= \sum_{s'' \in S} Pr(s'', s') \times \boldsymbol{Prob}_{s,\,k-1}(s'') \\ &= \boldsymbol{Prob}_{s,\,k-1}(s, s'') \cdot \boldsymbol{Ma}. \end{aligned}$$

The probability vector of starting from place s to all places in k time-steps is $\boldsymbol{Prob}_{s,k}(S) = \boldsymbol{Prob}_{s,0}(S) \cdot \boldsymbol{Ma}^k$.

3.3.2 Action-Based PCTL

Temporal logic is a powerful means to specify complex requirements that a system has to satisfy, and the most commonly used of which are the temporal logic LTL (linear temporal logic), CTL (computation tree logic), and PLTL[21]. Later, PCTL[22-23], RTCTL[24] and CSL[25] were established on the basis of CTL, which are not only used to specify system function but also to reason about the performance of probability or time.

Based on the analysis of execution path and probability measurement of path, we extend PCTL with action[26-27] to interpret the logical semantics for NPPN system, and name it as the action-based PCTL (aPCTL). aPCTL is the action-based probabilistic computation tree logic, which is transformed from PCTL by the following two aspects: 1) remove the atomic state proposition of PCTL state formulae, 2) augment the action proposition in the PCTL path formulae. The formal definition of aPCTL is as follows.

Definition 18 (Syntax of aPCTL). *Let $p \in [0, 1]$, which indicates probability, $\propto \in \{\leqslant, \geqslant, <, >\}$, which indicates comparison operator, state formulae Φ of aPCTL can be expressed by BNF, $\Phi ::= true|\Phi_1 \wedge \Phi_2|\neg\Phi|\mathrm{P}_{\propto_p}[\varphi]$; Let $k \in IN \cup \{\infty\} \cup \{0\}$, which indicates the number of step, $A \subset Act$, $B \subset Act$, where Act indicates set of the acts, IN means the set of positive integers, Φ_1 and Φ_2 are state formulae. Path formulae φ of aPCTL can be expressed by BNF, $\varphi ::= \Phi_{1A}U^{\leqslant k}\Phi_2|\Phi_{1A}U_B^{\leqslant k+1}\Phi_2$.*

In the state formulae, the probabilistic operator $\mathrm{P}_{\propto_p}[\varphi]$ replaces the CTL path quantifiers $A\varphi$ and $E\varphi$, which means the probability of path formula φ satisfies $\propto p$. $\mathrm{P}_{\propto_p}[\varphi]$ can express the state formulae $AG\varphi$ and $EF\varphi$ by selecting the external probabilities, i.e., $AG\varphi = \mathrm{P}_{\geqslant 1}[G\varphi]$, $EF\varphi = \mathrm{P}_{>0}[F\varphi]$. The other Boolean connectives such as $\vee$, $\to$, and $\leftrightarrow$ can be derived in the following way: $\Phi_1 \vee \Phi_2 = \neg(\neg\Phi_1 \wedge \neg\Phi_2)$, $\Phi_1 \to \Phi_2 = \neg\Phi_1 \vee \Phi_2$, $\Phi_1 \leftrightarrow \Phi_2 = (\neg\Phi_1 \vee \Phi_2) \wedge (\Phi_2 \vee \neg\Phi_1)$.

In the state formulae, the path formula $\Phi_{1A}U^{\leqslant k}\Phi_2$ is the variant of $\Phi_1U^{\leqslant k}\Phi_2$ in PCTL path formula. $\Phi_{1A}U^{\leqslant k}\Phi_2$ asserts that a Φ_2-state is eventually reached via visiting only Φ_1-state before, and in the process of visiting the following conditions have to be satisfied: a) only taking transitions from the act set A, and b) going from the beginning of the path until reaching the $\it\Phi_2$-state should last at most k steps. The path formula $\Phi_{1A}U_B^{\leqslant k+1}\Phi_2$ is the variant of $\Phi_{1A}U^{\leqslant k}\Phi_2$, which also asserts that a $\it\Phi_2$-state is eventually reached via visiting only Φ_1-state before, and in the process of visiting the following conditions have to be satisfied: a) only taking transitions from the act set A, b) going from the beginning of the path until reaching the Φ_2-state should last at most k steps, c) a transition to a Φ_2-state is actually made, and d) this transition belongs to the act set B. The condition c) is the reason for that the step-bound of $\Phi_{1A}U_B^{\leqslant k+1}\Phi_2$ is $k+1$ and the step-bound of $\Phi_{1A}U^{\leqslant k}\Phi_2$ is k. $\Phi_{1A}U^{\leqslant k}\Phi_2$ can express the CTL formula $\Phi_1 U\Phi_2$ by selecting the special value of A and k, i.e., $\Phi_1 U\Phi_2 = \Phi_{1_{\mathrm{Act}}}U^{<\infty}\Phi_2$. The other operation of path formulae can be derived as follows:

1) *Operator X*. In the aPCTL path formula operator X is based on act transition, which can be derived from the path formula $\Phi_{1A}U_B^{\leqslant k+1}\Phi_2$:

$$X_A\Phi = \mathrm{true}_\phi U_A^{\leqslant k+1}\Phi, \quad k=0; X\Phi = X_{\mathrm{Act}}\Phi.$$

2) *Operator F*. Operator F can be derived from the path formulae $\Phi_{1A}U^{\leqslant k}\Phi_2$ and $\Phi_{1A}U_B^{\leqslant k+1}\Phi_2$:

$${}_AF^{\leqslant k}\Phi = \mathrm{true}_A U^{\leqslant k}\Phi; {}_AF\Phi = {}_AF^{<\infty}\Phi; F\Phi =_{\mathrm{Act}} F\Phi;$$
$${}_AF_B^{\leqslant k+1}\Phi = \mathrm{true}_A U_B^{\leqslant k+1}\Phi; {}_AF_B\Phi = {}_AF^{<\infty}{}_B\Phi.$$

3) *Operator G*. Operator G is the dual operator of F, which can be derived from operator F:

$$\mathrm{P}_{\propto_p}[{}_AG^{\leqslant k}\Phi] = \neg\mathrm{P}_{\propto_p}[{}_AF^{\leqslant k}\neg\Phi],$$
$$\mathrm{P}_{\propto_p}[{}_AG_B^{\leqslant k+1}\Phi] = \neg\mathrm{P}_{\propto_p}[{}_AF_B^{\leqslant k+1}\neg\Phi].$$

Definition 19 (Semantics of aPCTL). *Let $\sum$ be an NPPN system, the aPCTL state and path formulae are interpreted over the places and execution paths. Let $Sat(\Phi) = \{s \in S | s \models \Phi\}$, the satisfaction relation for state formulae are defined by*:

$$\begin{aligned}
&s \models \mathrm{true}, && \forall s \in S,\\
&s \models \neg\Phi, && \text{iff } s \not\models \Phi,\\
&s \models \Phi_1 \wedge \Phi_2, && \text{iff } s \models \Phi_1 \wedge s \models \Phi_2,\\
&s \models \mathrm{P}_{\propto p}[\varphi], && \text{iff } P_s(\pi | \pi \in Path(s) \wedge \pi \models \varphi) \propto p.
\end{aligned}$$

The satisfaction relations for path formulae are defined by:

$$\begin{aligned}
&\pi \models \Phi_{1A}U^{\leqslant k}\Phi_2, \quad \text{iff} \quad \exists 0 \leqslant i \leqslant k,\\
&(\pi(i) \models \Phi_2 \wedge (\forall 0 \leqslant j < i(\pi(j) \models \Phi_1))\wedge\\
&\qquad (\pi(j) \xrightarrow{A} \pi(j+1))),
\end{aligned}$$

$$\begin{aligned}
&\pi \models \Phi_{1A}U_B^{\leqslant k+1}\Phi_2, \quad \text{iff} \quad \exists 0 < i \leqslant k+1,\\
&(\pi(i) \models \Phi_2 \wedge (\forall 0 \leqslant j < i(\pi(j) \models \Phi_1))\wedge\\
&\qquad (\pi(j) \xrightarrow{A} \pi(j+1))\wedge\\
&\qquad (\pi(i-1) = \Phi_1) \wedge (\pi(i-1) \xrightarrow{B} \pi(i))),
\end{aligned}$$

where $\pi = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots \xrightarrow{t_{n-1}} s_n \xrightarrow{t_n} s_{n+1} \ldots$, i is an integer and not less than 0, and $\pi[i] = s_i$.

Theorem 1. *The path set described by path formula φ of aPCTL is measurable.*

Proof. According to the syntax of aPCTL, there are two situations for φ:

1) $\varphi = \Phi_{1A}U^{\leqslant k}\Phi_2$.

The set $\{\pi \in Path(s) | \pi \models \varphi\}$ for φ can be obtained by the union of the cylinder sets $Cylinder(\varphi) = \{\pi \in Paths(\sum) | \varphi \in pref(\pi)\}$, i.e., $Cylinder(s_0s_1s_2\ldots s_k)$ where $s_k \models \Phi_2$, $s_i \models \Phi_1$ and $t_i \in A$ for $s_0 \xrightarrow{t_1} s_1 \xrightarrow{t_2} \ldots \xrightarrow{t_k} s_k$, $0 \leqslant i < k$.

2) $\varphi = \Phi_{1A}U_B^{\leqslant k+1}\Phi_2$.

The set $\{\pi \in Path(s) | \pi \models \varphi\}$ for φ can be obtained by the union of the cylinder sets $Cylinder(\varphi) = \{\pi \in Paths(\sum) | \varphi \in pref(\pi)\}$, i.e., $Cylinder(s_0s_1s_2\ldots s_k)$ where $s_k \models \Phi_2$, $s_i \models \Phi_1$, $t_i \in A$ and $t_{k-1} \in B$ for $s_0 \xrightarrow{t_1} s_1 \xrightarrow{t_2} \ldots \xrightarrow{t_k} s_k$, $0 \leqslant i < k$.

Because $Cylinder(s_0s_1s_2\ldots s_k)$ belongs to the σ-algebra associated with defined in Definition 14, $Cylinder(s_0s_1s_2\ldots s_k)$ is measurable. In line with the characteristics of σ-algebra, the union set of $Cylinder(s_0s_1s_2\ldots s_k)$ is measurable too.

The other operators of path formula can be derived form 1) and 2), so the path set described by each formula φ of aPCTL is measurable. □

The NPPN system $\sum$ satisfies aPCTL state formula Φ if and only if Φ holds in all initial places of $\sum$, i.e., $\sum \models \Phi$ if and only if $\forall s \models \Phi$, where s is the initial place of $\sum$.

4 Compositional Operation of NPPN System

In this section we will analyze the compositional operation of NPPN system at the level of algebraic semantics. Compositional operation is a very useful method to model systems. When using NPPN system to model complex systems, it is usually to construct NPPN system models of sub-systems using bottom-up approach, then the NPPN system models of sub-systems form the complete NPPN system model of the system by compositional operation. According to the forming manner of sub-systems, we divide the compositional operation of NPPN systems into three kinds of operations: true concurrency composition, synchronous communication composition, asynchronous communication composition, and present the compositional operation rules.

1) *True Concurrency Composition.* True concurrency composition means that there is not the same action execution, information exchange, or dependent relation between NPPN systems $\sum_1$ and $\sum_2$, i.e., $\sum_1$ and $\sum_2$ can execute in any order, which is denoted as $C_{\rm T}$. Because the operation rules of K, W and M for true concurrency composition are the same with the classical Petri net, we do not consider them here. Let $\sum_1 = (S_1, T_1; F_1, f_1; C_1)$, $\sum_2 = (S_2, T_2; F_2, f_2; C_2)$, where $S_1 \cap S_2 = \phi$, $T_1 \cap T_2 = \phi$, then the true concurrency compositional net system $\sum = \sum_1 C_{\rm T} \sum_2 = \sum_1 + \sum_2$, $\sum_2 + \sum_1 = (S, T; F, f; C)$, where

$$\begin{aligned}
&S = S_1 \cup S_2 \cup \{s_1, s_2\};\\
&T = T_1 \cup T_2 \cup \{t_1, t_2, t_3, t_4, t_5, t_6\};\\
&C = C_1 \cup C_2 \cup \{(t_1), (t_2)\};\\
&F = F_1 \cup F_2 \cup \{(s_1, t_1), (s_1, t_2), (t_1, s_{11}), (t_2, s_{21}),\\
&(s_{12}, t_3), (s_{22}, t_4), (t_3, s_{21}), (t_4, s_{11}), (s_{22}, t_5),\\
&(s_{12}, t_6), (t_5, s_2), (t_6, s_2)\};\\
&f_S = f_{S_1} \cup f_{S_2} \cup \{f_S(s_1), f_S(s_2)\};\\
&f_T = f_{T_1} \cup f_{T_2} \cup \{f_T(t_1), f_T(t_2), f_T(t_3), f_T(t_4),\\
&\qquad f_T(t_5), f_T(t_6)\};\\
&f_{S\times T} = f_{S_1\times T_1} \cup f_{S_2\times T_2} \cup \{f_{S\times T}(s_1, t_1),\\
&\qquad\quad f_{S\times T}(s_1, t_2), f_{S\times T}(s_{12}, t_3),\\
&\qquad\quad f_{S\times T}(s_{22}, t_4), f_{S\times T}(s_{22}, t_5),\\
&\qquad\quad f_{S\times T}(s_{12}, t_6)\};\\
&f_{T\times S} = f_{T_1\times S_1} \cup f_{T_2\times S_2} \cup \{f_{T\times S}(t_1, s_{11}),\\
&\qquad\quad f_{T\times S}(t_2, s_{21}), f_{T\times S}(t_3, s_{21}),\\
&\qquad\quad f_{T\times S}(t_4, s_{11}), f_{T\times S}(t_5, s_2),\\
&\qquad\quad f_{T\times S}(t_6, s_2)\};\\
&t_1 = t_2 = t_3 = t_4 = t_5 = t_6 = (\text{null}, 1);\\
&f_S(s_1) = 1, \quad f_{S\times T}(s_1, t_1) = 1, \quad f_{S\times T}(s_1, t_2) = 1.
\end{aligned}$$

As shown in Fig.5, we use the free choice net to implement the compositional operation of true concurrency, and for $\forall s_i \in S_1$, $\forall s_k \in S_2$, the act semantics of it is as follows:

$$\frac{s_i \xrightarrow{t} s_i'}{(s_i, s_k) \xrightarrow{t} (s_i', s_k)} \text{ or } \frac{s_k \xrightarrow{t} s_k'}{(s_i, s_2) \xrightarrow{t} (s_i, s_k')}.$$

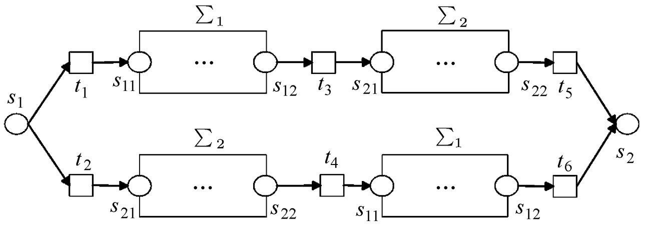


Fig.5. True concurrency composition.

2) *Synchronous Communication Composition.* Synchronous communication composition means that there is interrelated action execution to exchange information between NPPN systems $\sum_1$ and $\sum_2$, which is denoted as $C_{\rm S}$. Because the operation rules for K, W and M of synchronous composition are the same with the classical Petri net, we do not consider them here. Let $\sum_1 = (S_1, T_1; F_1, f_1; C_1)$, $\sum_2 = (S_2, T_2; F_2, f_2; C_2)$, where $S_1 \cap S_2 = \phi$ and $T_1 \cap T_2 \neq \phi$, min( ) be the minimum function, then the synchronous communication compositional net system $\sum = \sum_1 C_{\rm S} \sum_2 = (S, T; F, f; C)$, where

$$\begin{aligned}
&S = S_1 \cup S_2;\\
&C = C_1 \cup C_2;\\
&t' = T_1 \cap T_2;\\
&T = T_1/\{t_1\} \cup T_2/\{t_2\} \cup t';\\
&F = F_1/\{(s_{11}, t_1), (t_1, s_{12})\} \cup F_2/\{(s_{21}, t_2), (t_2, s_{22})\} \cup\\
&\qquad \{(s_{11}, t'), (s_{21}, t'), (t', s_{12}), (t', s_{22})\};\\
&f_S = f_{S_1} \cup f_{S_2};\\
&f_T = f_{T_1}/\{f_{T_1}(t_1)\} \cup f_{T_2}/\{f_{T_2}(t_2)\} \cup \{f_T(t')\};\\
&f_{S\times T} = (f_{S_1\times T_1}/\{f_{S\times T}(s_{11}, t_1)\}) \cup \{f_{S\times T}(s_{11}, t')\} \cup\\
&\qquad\qquad (f_{S_2\times T_2}/\{f_{S\times T}(s_{21}, t_2)\}) \cup \{f_{S\times T}(s_{21}, t')\};\\
&f_{T\times S} = (f_{T_1\times S_1}/\{f_{T_1\times S_1}(t_1, s_{12})\}) \cup\\
&\qquad\qquad \{f_{T_1\times S_1}(t', s_{12})\} \cup\\
&\qquad\qquad (f_{T_2\times S_2}/\{f_{T_2\times S_2}(t_2, s_{22})\}) \cup\\
&\qquad\qquad \{f_{T_2\times S_2}(t', s_{22})\}\\
&\sigma_{Pt}(f_T(t')) = \min(\sigma_{Pt}(f_T(t_1)), \sigma_{Pt}(f_T(t_2)));\\
&f_{S\times T}(s_{11}, t') = f_{S\times T}(s_{11}, t_1);\\
&f_{S\times T}(s_{21}, t') = f_{S\times T}(s_{21}, t_2);\\
&f_{T_1\times S_1}(t', s_{12}) = f_{T_1\times S_1}(t_1, s_{12});\\
&f_{T_2\times S_2}(t', s_{22}) = f_{T_2\times S_2}(t_2, s_{22}).
\end{aligned}$$

As shown in Fig.6, we use fusion of transitions of subnets to implement the compositional operation of synchronous communication, and for $\forall s_i \in S_1$, $\forall s_k \in S_2$, the act semantics of it is as follows,

$$\begin{cases}
\dfrac{s_i \xrightarrow{t} s_i' \wedge s_k \xrightarrow{t} s_k'}{(s_i, s_k) \xrightarrow{t} (s_i', s_k')}, \quad \text{if } t \in T_1 \cap T_2,\\[2ex]
\dfrac{s_i \xrightarrow{t} s_i'}{(s_i, s_k) \xrightarrow{t} (s_i', s_k)} \text{ or } \dfrac{s_k \xrightarrow{t} s_k'}{(s_i, s_k) \xrightarrow{t} (s_i, s_k')},\\
\text{if } t \in F/\{t'\}.
\end{cases}$$

3) *Asynchronous Communication Composition.* Asynchronous communication composition means that there is the sharing place to exchange information between NPPN systems $\sum_1$ and $\sum_2$, which is denoted as $C_{\rm A}$. Because the operation rules for K, W and M of

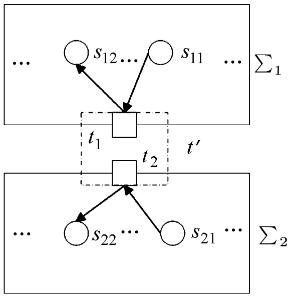


Fig.6. Synchronous communication composition.

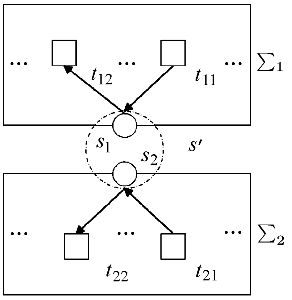


Fig.7. Asynchronous communication composition.

asynchronous composition are the same with the classical Petri net, we do not consider them here. Let $\sum_1 = (S_1, T_1; F_1, f_1; C_1)$, $\sum_2 = (S_2, T_2; F_2, f_2; C_2)$, where $S_1 \cap S_2 \neq \phi$ and $T_1 \cap T_2 = \phi$, min( ) be the minimum function, then the asynchronous communication compositional net system $\sum = \sum_1 C_{\rm A} \sum_2 = (S, T; F, f; C)$, where

$$s' = S_1 \cap S_2;$$
$$S = S_1/\{s_1\} \cup S_2/\{s_2\} \cup \{s'\};$$
$$T = T_1 \cup T_2;$$
$$C = C_1 \cup C_2;$$
$$F = F_1/\{(t_{11}, s_1), (s_1, t_{12})\} \cup F_2/\{(t_{21}, s_2), (s_2, t_{22})\} \cup \{(t_{11}, s'), (t_{21}, s'), (s', t_{12}), (s', t_{22})\};$$
$$f_S(s) = (f_1/\{f_S(s_1)\}) \cup (f_2/\{f_S(s_2)\}) \cup \{f_S(s')\};$$
$$f_T = f_{T_1} \cup f_{T_2};$$
$$f_{S\times T} = (f_{S_1\times T_1}/\{f_{S\times T}(s_1, t_{12})\}) \cup \{f_{S\times T}(s', t_{12})\} \cup (f_{S_2\times T_2}/\{f_{S\times T}(s_2, t_{22})\}) \cup \{f_{S\times T}(s', t_{22})\};$$
$$f_{T\times S} = (f_{T_1\times S_1}/\{f_{T_1\times S_1}(t_{11}, s_1)\}) \cup \{f_{T\times S}(t_{11}, s')\} \cup (f_{T_2\times S_2}/\{f_{T_2\times S_2}(t_{21}, s_2)\}) \cup \{f_{T\times S}(t_{21}, s')\};$$
$$f_S(s') = \min(f_S(s_1), f_S(s_2));$$
$$f_{S\times T}(s', t_{12}) = f_{S\times T}(s_1, t_{12});$$
$$f_{S\times T}(s', t_{22}) = f_{S\times T}(s_2, t_{22});$$
$$f_{T\times S}(t_{11}, s') = f_{T_1\times S_1}(t_{11}, s_1);$$
$$f_{T\times S}(t_{21}, s') = f_{T_2\times S_2}(t_{21}, s_2).$$

As shown in Fig.7, we use fusion of places of subnets to implement the compositional operation of asynchronous communication, and for $\forall s_i \in S_1$, $\forall s_k \in S_2$, the act semantics of it is as follows,

$$\frac{s_i \xrightarrow{t} s'_i}{(s_i, s_k) \xrightarrow{t} (s'_i, s_k)} \text{ or } \frac{s_k \xrightarrow{t} s'_k}{(s_i, s_k) \xrightarrow{t} (s_i, s'_k)}.$$

5 Model Checking NPPN System

5.1 Problem Statement

Given an NPPN system model $\sum$ of the system under consideration, a desired property described by aPCTL formula Φ, and the initial place s_0, the model checking is to decide whether the NPPN system model with the initial place s_0 satisfies the aPCTL formula, i.e., $(\sum, s_0) \models \Phi$ is true or not. The above belongs to the local semantic model checking problem. While in order to compute whether all the other places satisfy Φ or not, we use global semantic model checking algorithm to implement it:

1) recursively computing place set which satisfies semantics of the sub-formulae of Φ;

2) construct set Y of all the places which satisfy the formula Φ;

3) decide whether the initial place s_0 belongs to the set Y or not, i.e., if $s_0 \in Y$, $(\sum, s_0) \models \Phi$ is true, else $(\sum, s_0) \models \Phi$ is false.

5.2 Analysis of the Solution

Because the aPCTL formula Φ is comprised of state formulae and path formulae, the solution process can be divided into two steps. Let $Sat(\Phi) = \{s \in S | s \models \Phi\}$:

1) *Satisfiability Computation of Non-Probabilistic State Formulae.*

$$\begin{aligned} Sat(\text{true}) =& \{s|s \models \text{true}\}, \\ Sat(\Phi_1 \wedge \Phi_2) =& \{s|s \models \Phi_1 \wedge \Phi_2\} \\ =& \{s|s \models \Phi_1\} \cap \{s|s \models \Phi_2\} \\ =& Sat(\Phi_1) \cap Sat(\Phi_2), \\ Sat(\neg\Phi) =& \{s|s \models \neg\Phi\} \\ =& Sat(\text{true}) - \{s|s \models \Phi\} \\ =& Sat(\text{true}) - Sat(\Phi). \end{aligned}$$

The satisfiability computation of the other formulae can be derived by the above formulae.

2) *Satisfiability Computation of Probabilistic Path.*

Due to the nondeterminism of the NPPN system, a path of model $\sum$ may have several probability values. We use $P_s^{\min}$ and $P_s^{\max}$ to indicate the minimum value and the maximum value over the nondeterminism set C respectively, i.e., $P_s^{\min}(\varphi) = \inf\limits_{C} P_s(\varphi)$,

$P_s^{\max}(\varphi) = \sup\limits_{C} P_s(\varphi)$. To compute which one of them is determined by the operator $\propto$. If $\propto \in \{\leqslant, <\}$, we should compute $P_s^{\max}$ over the nondeterminism set C, else compute $P_s^{\min}$. For the sake of simplicity, we take $P_s^{\min}$ for the example in the following subsections, and the computing process of $P_s^{\max}$ is similar to that of $P_s^{\min}$.

For the formula $Sat(\mathrm{P}_{\propto p}[\Phi_{1A}U^{\leqslant k}\Phi_2]) = \{s \in Sat(\mathrm{true}) | P_s^{\min}(\Phi_{1A}U^{\leqslant k}\Phi_2) \propto p\}$, it need to compute $P_s^{\min}(\Phi_{1A}U^{\leqslant k}\Phi_2) \propto p$ for each place s of $Sat(\mathrm{true})$. $P_s^{\min}$ can be computed as follows,

a) $P_s^{\min}(\Phi_{1A}U^{\leqslant k}\Phi_2) = 1$, if $s \in Sat(\Phi_2)$;

b) $P_s^{\min}(\Phi_{1A}U^{\leqslant k}\Phi_2) = 0$,
if $s \in Sat(\mathrm{true}) - (Sat(\Phi_1) \cup Sat(\Phi_2))$;

c) $P_s^{\min}(\Phi_{1A}U^{\leqslant k}\Phi_2) = 0$,
if $s \in Sat(\Phi_1) \cup Sat(\Phi_2) - Sat(\Phi_2)$ and $k = 0$;

d) $P_s^{\min}(\Phi_{1A}U^{\leqslant k}\Phi_2) = 0$,
if $s \in Sat(\Phi_1) \cup Sat(\Phi_2) - Sat(\Phi_2)$, $k > 0$
and $\neg(\exists t, s \xrightarrow{t} s' \wedge \sigma_{\mathrm{Trans}}(t) \in A)$;

e) $P_s^{\min}(\Phi_{1A}U^{\leqslant k}\Phi_2) = \sum\limits_{s' \in S} Pr(s, s') \bullet P_{s'}^{\min}(\Phi_{1A}U^{\leqslant k-1}\Phi_2) = \boldsymbol{Ma'} \bullet P_s^{\min}(\Phi_{1A}U^{\leqslant k-1}\Phi_2) = (\boldsymbol{Ma'})^k \bullet P_{s'}^{\min}(\Phi_{1A}U^{\leqslant 0}\Phi_2)$,
if $s \in Sat(\Phi_1) \cup Sat(\Phi_2) - Sat(\Phi_2)$, $k > 0$
and $\forall t\ (s \xrightarrow{t} s') \rightarrow (\sigma_{\mathrm{Trans}}(t) \in A)$, where

$$Pr(s,\ s') = \begin{cases} Pr(s,\ s'), & \text{if } s \in Sat(\Phi_1) \cup Sat(\Phi_2) - \\ & \quad Sat(\Phi_2), \\ 1, & \text{if } s \in Sat(\Phi_2) \wedge s = s', \\ 0, & \text{otherwise.} \end{cases}$$

Probability matrix $\boldsymbol{Ma'}$ is composed by $Pr(s,\ s')$.

For the formula $Sat(\mathrm{P}_{\propto p}[\Phi_{1A}U_B^{\leqslant k+1}\Phi_2]) = \{s \in Sat(\mathrm{true}) | P_s^{\min}(\Phi_{1A}U_B^{\leqslant k+1}\Phi_2) \propto p\}$, we also need to compute $P_s^{\min}(\Phi_{1A}U^{\leqslant k}\Phi_2) \propto p$ for each place s of $Sat(\mathrm{true})$. $P_s^{\min}$ can be computed as follows,

f) $P_s^{\min}(\Phi_{1A}U_B^{\leqslant k+1}\Phi_2) = 0$, if $s \not\models \Phi_1$;

g) $P_s^{\min}(\Phi_{1A}U_B^{\leqslant k+1}\Phi_2) = f_S(s) \times \sigma_{Pt}(t) \times f_{T\times S}(t, s')$, if $(s \models \Phi_1) \wedge \exists t(s \xrightarrow{t} s') \wedge (s' \models \Phi_2) \wedge (\sigma_{\mathrm{Tans}}(t) \in B)$;

h) $P_s^{\min}(\Phi_{1A}U_B^{\leqslant k+1}\Phi_2) =$
$\min(\sum\limits_{s' \in S}(f_S(s) \times \sigma_{Pt}(t) \times f_{T\times S}(t, s')))+$
$\sum\limits_{s' \in S} Pr(s,\ s') \times P_{s'}^{\min}(\Phi_{1A}U_B^{\leqslant k}\Phi_2)$, if $\exists t \exists s'((s' \models \Phi_2) \wedge ((s \xrightarrow{t} s') \rightarrow (\sigma_{\mathrm{Trans}}(t) \in B))) \wedge\ \exists t \exists s''((s'' \models \Phi_1) \wedge ((s \xrightarrow{t} s'') \rightarrow (\sigma_{\mathrm{Trans}}(t) \in A))) \wedge (A \cap B = \phi)$;

i) $P_s^{\min}(\Phi_{1A}U_B^{\leqslant k+1}\Phi_2) =$
$\min(\sum\limits_{s' \in S}(f_S(s) \times \sigma_{Pt}(t) \times f_{T\times S}(t, s')))+$
$\sum\limits_{s' \in S} Pr(s,\ s') \times P_{s'}^{\min}(\Phi_{1A}U_B^{\leqslant k}\Phi_2)-$
$\sum\limits_{s' \in S} Pr_{A\cap B}(s, s') \times P_{s'}^{\min}(\Phi_{1A}U_B^{\leqslant k}\Phi_2)$,

where $Pr_{A\cap B}(s, s')$ denotes the value of $Pr(s, s')$ when $s \xrightarrow{t} s'$, and $t \in A \cap B$, if $\exists t \exists s'((s' \models \Phi_2) \wedge ((s \xrightarrow{t} s') \rightarrow (\sigma_{\mathrm{Trans}}(t) \in B))) \wedge\ \exists t \exists s''((s'' \models \Phi_1) \wedge ((s \xrightarrow{t} s'') \rightarrow (\sigma_{\mathrm{Trans}}(t) \in A))) \wedge (A \cap B \neq \phi)$;

j) Others, $P_s^{\min}(\Phi_{1A}U_B^{\leqslant k+1}\Phi_2) = 0$.

The method to compute h) and i) is similar to e).

5.3 Model Checking Algorithm

Based on the above analyses, we present the complete algorithm of aPCTL model checking NPPN system. Take $P_s^{\min}$ for the example, the algorithm is shown in Fig.8.

```
Input: NPPN model $\sum$, initial place $s_0$, aPCTL formula $\Phi$
Output: model $\sum$ satisfies the formula $\Phi$ or not
MC_NPPN ($\sum, s_0, \Phi$){
$Sat(\Phi)$ = Computing ($\sum, s_0, \Phi$);
If ($s_0 \in Sat(\Phi)$) print ("model $\sum$ satisfies the formula $\Phi$");
  else print ("model $\sum$ does not satisfy the formula $\Phi$");
}
Computing ($\sum, s_0, \Phi$){
  Translate the formula $\Phi$ into the forms of $\Phi_1 \wedge \Phi_2$, $\neg\Phi$,
  or $P_{\propto p}[]$
  Parse the syntax tree of $\Phi$ by recursively depth-first
  traversing it
  switch ($\Phi$)
    case true: return $S$
    case $\Phi_1 \wedge \Phi_2$: return Computing($M, s_0, \Phi_1$)$\cap$
      Computing($M, s_0, \Phi_2$)
    case $\neg\Phi$: return Sat (true)-Computing ($M, s_0, \Phi$)
    case $P_{\propto p}[\Phi_{1A}U^{\leqslant k}\Phi_2]$: return $\{s \in S | P_s^{\min}(\Phi_{1A}U^{\leqslant k}\Phi_2)$
      $\propto p\}$
    case $P_{\propto p}[\Phi_{1A}U_B^{\leqslant k+1}\Phi_2]$: return $\{s \in S | P_s^{\min}(\Phi_{1A}U_B^{\leqslant k+1}$
      $\Phi_2) \propto p\}$
}
```

Fig.8. aPCTL model checking NPPN system algorithm.

The time complexity of the algorithm is analysed as follows. Let the number of flow relation be $|F|$ and the number of sub-formulae included in formula Φ be $|\Phi|$, each sub-formula has to execute satisfiability computation once, so the execution number of computation is $|\Phi|$; the time for transforming logic operation into flow relation set operation is $O(|F|)$; therefore, the time complexity of the model checking algorithm proposed in this paper is $O(|\Phi|(|F| + k_{\max}))$, where $k_{\max}$ means the maximum value of k.

6 Tool Support and Case Study

Based on the open source software PIPE 2[28], we develop the tool NPNMV (Nondeterministic Probabilistic Petri Net Modeling and Verification) for modelling, analysis, simulation and verification of NPPN systems. To the best of our knowledge, it is the first tool to model

Petri net with probability and nondeterminism. Besides the traditional function of modeling and analysis of Petri net which PIPE 2 owns, with the tool NPNMV we can, among others:

1) model the NPPN, for instance, setup the probability of transitions and arcs, and divide the nondeterministic class set;

2) analyse the NPPN model, such as the structural properties of siphon, trap, and probability matrix, the behavioural properties of fairness, liveness, persistency, and the probabilistic reachability marking graph;

3) qualitatively and quantitatively verify the NPPN model with aPCTL model checking algorithm.

Workflow area is a successful application of Petri net[29], in this section we take the travel arrangement workflow for example to discuss the modeling and verification of workflow model with NPPN system using the tool NPNMV. Assuming it is feasible to reach the destination by aircraft or ship, we divide the travel arrangement workflow into the following subtasks roughly: get visa, book hotel, book airline ticket and book ship ticket. The abstract NPPN systems corresponding to the above subtasks are shown in Fig.9, Fig.10, Fig.11, and Fig.12 respectively, and the attribute "pt" of the transition is the probability of successful implementation.

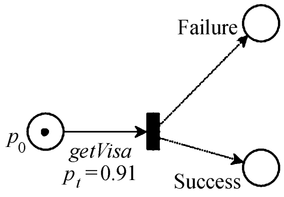


Fig.9. NPPN system of getting visa.

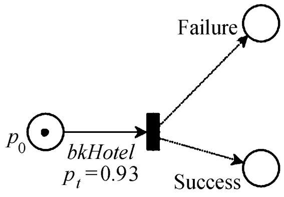


Fig.10. NPPN system of booking hotel.

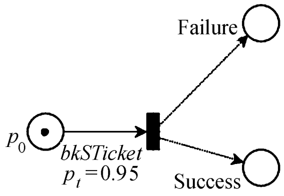


Fig.11. NPPN system of booking ship ticket.

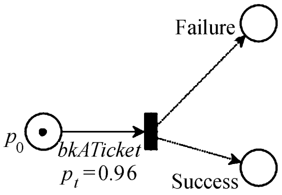
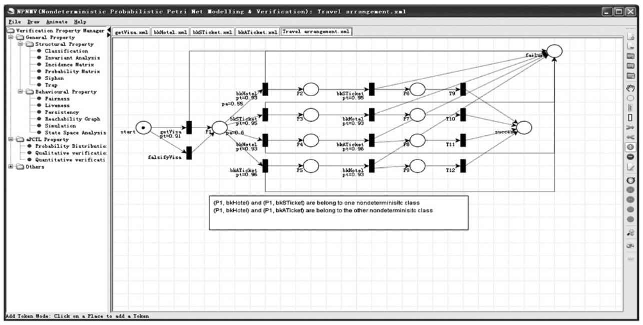


Fig.12. NPPN system of booking airline ticket.

According to the semantics and rules for compositional operation of NPPN system, the composed NPPN system of travel arrangement is shown in Fig.13, and the probability matrix of it is shown in Fig.14. In Fig.13, we use the only one *failure* place to present all failure places; the transition PT (represents the transitions T9, T10, T11, and T12) are probability transition with the probability 1 and action is null; as with the transition *getVisa*, transition *falsifyVisa* can also reach the place s_1, the use of transition *falsifyVisa* is just to illuminate that aPCTL is more convenient than PCTL, and it is not used as a practical transition for probabilistic model checking. That is to say, the PCTL formula cannot succinctly express that reaching place s_1 is the transition *getVisa* or *falsifyVisa*. However, we can use aPCTL Φ formula to do it, i.e., $\Phi = \text{true}U_{getVisa}\text{true}$. In Fig.14, forwards probability matrix, backwards matrix, combined probability matrix, and the transition probability vector are presented by the BFS (breadth-first traversal) of the travel arrangement NPPN system.



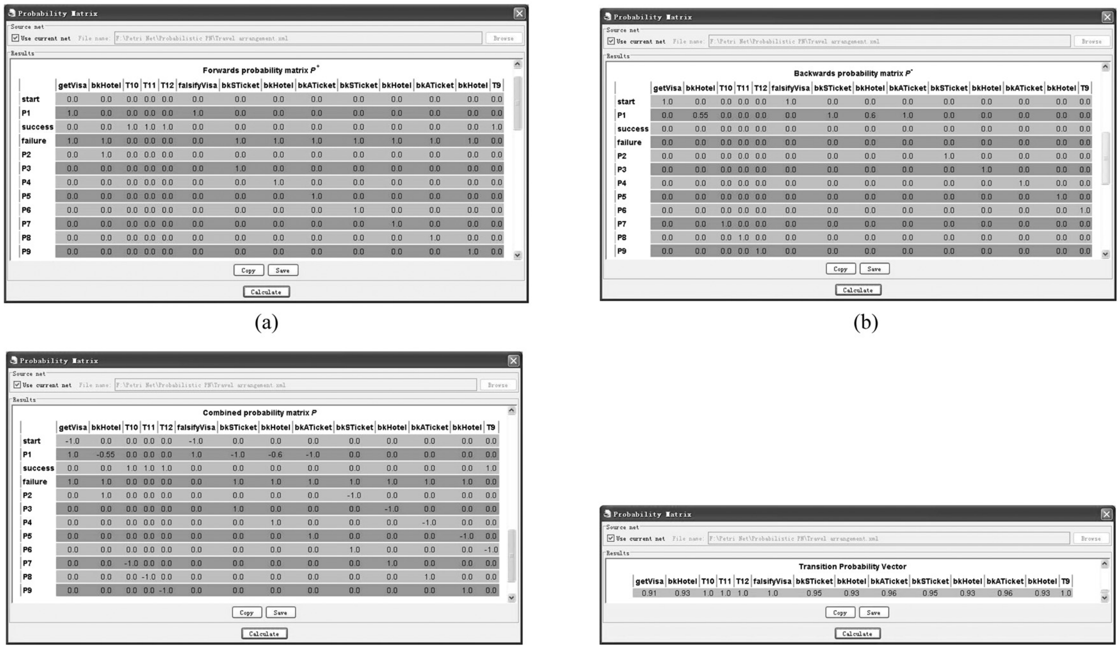
(P1, bkHotel) and (P1, bkSTicket) are belong to one nondeterminisitc class
(P1, bkHotel) and (P1, bkATicket) are belong to the other nondeterminisitc class

Add Token Mode: Click on a Place to add a Token

Fig.13. Travel arrangement NPPN system.

Next, we introduce the quantitative verification process for the property that the probability of reaching place *success* is not less than *threshold*. The property formula satisfied by the NPPN system can be formulated as $\text{P}_{\geqslant threshold}[\Phi F_{PT}\text{true}]$, and the verification problem can be implemented by model checking $(\sum, \ start) \models P^{\min}_{start}(\Phi F_{PT}\text{true}) \geqslant threshold$, and let *threshold*=0.75. The quantitative verification result is shown in Fig.15, and the resolution process of it is as follows.

1) According to the syntax of aPCTL and operator



☑ Use current net File name: F:\Petri Net\Probabilistic PN\Travel arrangement.xml

Forwards probability matrix P^+

	getVisa	bkHotel	T10	T11	T12	falsifyVisa	bkSTicket	bkHotel	bkATicket	bkSTicket	bkHotel	bkATicket	bkHotel	T9
start	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P1	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
success	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
failure	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0
P2	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P3	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
P5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
P6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
P7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
P8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
P9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0

(a)

☑ Use current net File name: F:\Petri Net\Probabilistic PN\Travel arrangement.xml

Backwards probability matrix P^-

	getVisa	bkHotel	T10	T11	T12	falsifyVisa	bkSTicket	bkHotel	bkATicket	bkSTicket	bkHotel	bkATicket	bkHotel	T9
start	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P1	0.0	0.55	0.0	0.0	0.0	0.0	1.0	0.6	1.0	0.0	0.0	0.0	0.0	0.0
success	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
failure	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
P3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
P4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
P5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
P6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
P7	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P8	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P9	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

(b)

☑ Use current net File name: F:\Petri Net\Probabilistic PN\Travel arrangement.xml

Combined probability matrix P

	getVisa	bkHotel	T10	T11	T12	falsifyVisa	bkSTicket	bkHotel	bkATicket	bkSTicket	bkHotel	bkATicket	bkHotel	T9
start	-1.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P1	1.0	-0.55	0.0	0.0	0.0	1.0	-1.0	-0.6	-1.0	0.0	0.0	0.0	0.0	0.0
success	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
failure	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0
P2	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0
P3	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0
P4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	-1.0	0.0	0.0
P5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	-1.0	0.0
P6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	-1.0
P7	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
P8	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
P9	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0

(c)

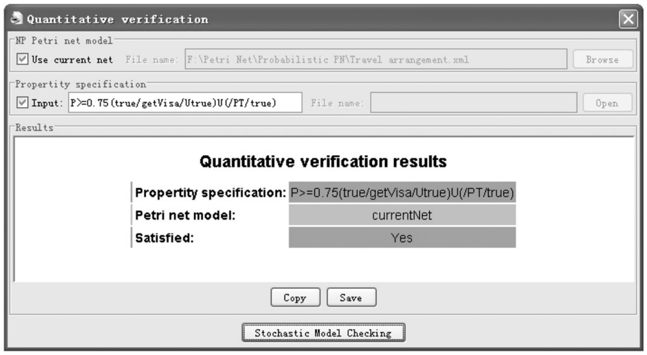
☑ Use current net File name: F:\Petri Net\Probabilistic PN\Travel arrangement.xml

Transition Probability Vector

getVisa	bkHotel	T10	T11	T12	falsifyVisa	bkSTicket	bkHotel	bkATicket	bkSTicket	bkHotel	bkATicket	bkHotel	T9
0.91	0.93	1.0	1.0	1.0	1.0	0.95	0.93	0.96	0.95	0.93	0.96	0.93	1.0

(d)

Fig.14. Probability matrices of travel arrangement NPPN system. (a) Forwards probability matrix. (b) Backwards matrix. (c) Combined probability matrix. (d) Transition probability vector.



☑ Use current net File name: F:\Petri Net\Probabilistic PN\Travel arrangement.xml

☑ Input: P>=0.75(true/getVisa/Utrue)U(/PT/true) File name:

Quantitative verification results

Propertity specification:	P>=0.75(true/getVisa/Utrue)U(/PT/true)
Petri net model:	currentNet
Satisfied:	Yes

Fig.15. Quantitative verification result of travel arrangement NPPN system.

F, transforming the property formula into the operator U formula, i.e.,

$$\Phi F_{PT}\text{true} = \Phi U^{\leqslant k}_{PT}\text{true}.$$

2) In the light of the satisfiability computation of probabilistic path in Subsection 4.1, we compute the value of $P^{\min}_{start}(\Phi U^{\leqslant k}_{PT}\text{true})$ over the nondeterminism α and β,

$$\begin{aligned}
&\min\{0.55 \times \sigma_{Pt}(f_T(bookHotel))\times\\
&\quad\sigma_{Pt}(f_T(bookSTicket))+\\
&\quad 0.45 \times \sigma_{Pt}(f_T(bookSTicket)) \times \sigma_{Pt}(f_T(bookHotel)),\\
&\quad 0.6 \times \sigma_{Pt}(f_T(bookHotel)) \times \sigma_{Pt}(f_T(bookATicket))+\\
&\quad 0.4 \times \sigma_{Pt}(f_T(bookATicket)) \times \sigma_{Pt}(f_T(bookHotel))\}\\
&=0.8835.
\end{aligned}$$

3) Compute $P^{\min}_{start}({}_{getVisa}F_{PT}\text{true})$, $P^{\min}_{start}({}_{getVisa}F_{PT}\text{true}) = 0.803985$. So, the NPPN system satisfies the property formula $\mathrm{P}_{\geqslant threshold}[{}_{getVisa}F_{PT}\text{true}]$, according to the model checking algorithm.

6 Related Work

In the Petri net community, Varacca and Nielsen proposed the notion of probabilistic net systems[3], whose semantics is defined in terms of probabilistic languages, and they extended the notion of Mazurkiewicz equivalence to prove the soundness of their definition. Then they used the above idea to the denotational universe together with Winskel, and presented the concept of probabilistic event structures[30-31]. Kudlek introduced probability into Petri nets to increase their power[13], which was done via reachability sets of Petri nets, and proved that the probabilistic Petri nets are more powerful than ordinary Petri nets. Albanese, Chellappa and others presented the concept of a constrained probabilistic Petri net, and they used it to

[3] Probabilistic Petri nets and Mazurkiewicz equivalence. Draft, http://www.pps.univ-paris-diderot.fr/∼varacca/papers/shortpetri.ps, February 2003.

specify human activities that can be executed in a multiplicity of ways[12]. And then they developed the computational framework for human activity representation based on the constrained probabilistic Petri net.

Compared with the above work, we proposed a new notion of nondeterministic probabilistic Petri net on the condition of maintaining the characteristics of nondeterminism of Petri net, and we presented its algebraic and logical semantics based on the occurrence net and transition sequence respectively. And then we analyzed the composing operation and model checking algorithm of nondeterministic probabilistic Petri net.

In the performance evaluation area, Hermanns addressed the issue of compositional specification and analysis of Markov chains[32], Martin presented the methods to model check nondeterministic and randomly timed systems[33]. Based on principles known from process algebra, Hermams developed an algebra of Interactive Markov chains (IMCs) arising as an orthogonal extension of both continuous-time Markov chains and process algebra. Interactive Markov chains combine interactive processes and Markovian chains, which can be seen as the extended Markovian chains. So, from that point of view, IMC is the method to model and analyse systems at the low level. Compared with IMC, the NPPN model is a high level method to model and verify systems with nondeterminism and probability, and we believe that there may be a certain relationship between them which will be a very interesting research topic.

7 Conclusions

In this paper we introduced the probability measure theory into traditional Petri net system, ensuring the nondeterministic characteristics, and named it as the NPPN system which can model and verify the qualitative and quantitative behaviour aspects of systems simultaneously. Then, we gave its algebraic semantics and logical semantics, and by making use of them we presented the compositional operation rules and model checking algorithm of NPPN system respectively. In order to illustrate the usefulness of NPPN system, we took travel arrangement workflow for example to model and model check travel arrangement NPPN system. For the future, we will pay more attention to lower the time complexity and improve the efficiency of the model checking algorithm, which will involve the following topics, such as, stochastic game semantics, bounded model checking, satisfiability modulo theory.

References

[1] Girault C, Valk R. Petri Nets for System Engineering: A Guide to Modeling, Verification, and Application. Springer-Verlag, 2003.

[2] Lin C. Stochastic Petri Net and System Performance Evaluation (2nd Edition). Tsinghua University Press, 2005, pp.1-2. (in Chinese)

[3] Noe J D, Nutt G J. Macro e-nets representation of parallel systems. *IEEE Transactions on Computers*, 1973, C-22(8): 718-727.

[4] Merlin P M, Farber D J. Recoverability of communication protocols: Implications of a theoretical study. *IEEE Transactions on Communications*, 1976, 24(9): 1036-1043.

[5] Molloy M K. On the integration of delay and throughput measures in distributed processing models [Ph.D. Thesis]. University of California, Los Angeles, USA, 1981.

[6] Natkin S. Les reseaux de PETRI stochastiques et leur application à l'évaluation des systèmes informatiques [Ph.D. Thesis]. CNAM, Paris, France, 1980. (In French)

[7] Symons F J W. Introduction to numerical Petri nets, a general graphical model of concurrent processing systems. *Australian Telecommunications Research*, 1980, 14(1): 28-33.

[8] Marsan M A, Conte G, Balbo G. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems*, 1984, 2(2): 93-122.

[9] Petri C A. Introduction to general net theory. In *Lecture Notes in Computer Science 84*, Brauer W (ed.), Springer-Verlag, 1980, pp.1-19.

[10] Balbo G. Introduction to generalized stochastic Petri net. In *Proc. the 7th Int. Conf. Formal Methods for Performance Evaluation*, May 2007, pp.83-131.

[11] Baier C, Katoen J P. Principles of Model Checking. MIT Press, 2008.

[12] Albanese M, Chellappa R, Moscato V, Picariello A, Subrahmanian V S, Turaga P, Udrea O. A constrained probabilistic Petri net framework for human activity detection in video. *IEEE Transactions on Multimedia*, 2008, 10(8): 1429-1443.

[13] Kudlek M. Probability in Petri nets. *Fundamenta Informaticae — Concurrency Specification and Programming*, 2005, 67(1/3): 121-130.

[14] Benveniste A, Fabre E, Haar S. Markov nets: Probabilistic models for distributed and concurrent systems. *IEEE Transactions on Automatic Control*, 2003, 48(11): 1936-1950.

[15] Segala R. Modeling and verification of randomized distributed real-time systems [Ph.D. Thesis]. Massachusetts Institute of Technology, Cambridge, USA, 1995.

[16] Yuan C Y. Principle and Application of Petri Net. Beijing: Publishing House of Electronics Industry, 2005, pp.66-78. (in Chinese)

[17] Han T T. Diagnosis, synthesis and analysis of probabilistic models [Ph.D. Thesis]. RWTH Aachen University, Germany, 2009.

[18] Ash R B, Doléans-Dade C A. Probability and Measure Theory (2nd edition). Academic Press, 2000, pp.3-10.

[19] Milner R. Communication and Concurrency. Prentice-Hall, 1989, pp.10-36.

[20] Hoare C A R. Communicating Sequential Processes. Prentice-Hall, 1985, pp.81-100.

[21] Heljanko K, Junttila T, Latvala T. Incremental and complete bounded model checking for full PLTL. In *Proc. the 17th International Conference on Computer Aided Verification*, July 2005, pp.98-111.

[22] Hansson H, Jonsson B. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 1994, 6(5): 512-535.

[23] Bianco A, de Alfaro L. Model checking of probabilistic and nondeterministic systems. In *Proc. the 15th Conference on Foundations of Software Technology and Theoretical Computer Science*, December 1995, pp.499-513.

[24] Emerson E A, Mok A K, Sistla A P, Srinivasan J. Quantitative temporal reasoning. *Real Time Systems*, 1992, 4(4): 331-352.

[25] Baier C, Katoen J P, Hermanns H. Approximate symbolic model checking of continuous-time Markov chains. In *Proc. the 10th International Conference on Concurrency Theory*, August 1999, pp.146-162.

[26] Kindler E, Vesper T. ESTL: A temporal logic for events and states. In *Proc. the 19th International Conference of Application and Theory of Petri Nets*, June 1998, pp.365-384.

[27] Feng L, Kwiatkowska M, Parker D. Compositional verification of probabilistic systems using learning. In *Proc. the 7th International Conference on Quantitative Evaluation of Systems*, September 2010, pp.133-142.

[28] Bonet P, Llado C M, Puijaner R, Knottenbelt W J. PIPE v2.5: A Petri net tool for performance modelling. In *Proc. the 23rd Latin American Conference on Informatics*, October 2007.

[29] Yuan C Y, Zhao W, Zhang S K, Huang Y. A three-layer model for business processes: Process logic, case semantics and workflow management. *Journal of Computer Science and Technology*, 2007, 22(3): 410-425.

[30] Varacca D. Probability, nondeterminism and concurrency: Two denotational models for probabilistic computation [Ph.D. Thesis]. University of Aarhus, Aarhus, Denmark, 2003.

[31] Varacca D, Völzer H, Winskel G. Probabilistic event structures and domains. *Theoretical Computer Science — Concurrency Theory*, 2006, 358(2): 173-199,

[32] Hermanns H. Interactive markov chains: The quest for quantified quality. In *Lecture Notes in Computer Science 2428*, 2002, pp.57-87.

[33] Neuhäusser M R. Model checking nondeterministic and randomly timed systems [Ph.D. Thesis]. RWTH Aachen University, Germany, 2010.

Yang Liu received the M.Sc. degree in computer application from Qufu Normal University in 2007 and the Ph.D. degree in computer application from Shanghai University in 2012. He joined in Taishan University in 2007, and now he is also a post-doc researcher in the State Key Laboratory for Novel Software Technology at Nanjing University. His research interests include Petri nets and applications, service-oriented computing, and software verification.

Huai-Kou Miao received the M.Sc. degree in computer application technology from Shanghai University of Science and Technology, China, in 1986. He is currently a professor in the School of Computer Engineering and Science Technology at Shanghai University, China. His research interests include software formal methods and software engineering.

Hong-Wei Zeng is a professor at the School of Computer Engineering and Science, Shanghai University. He received his Ph.D. degree in control theory and control engineering from Shanghai University in 2008. His current research interests include formal method, formal verification and software testing.

Yan Ma received the M.Sc. degree in computer science and technology from Jiangnan University, Wuxi, China, in 2007. Her research interests include evolutionary computation and software engineering.

Pan Liu received the M.Sc. degree in computer software and theory from Nanchang University in 2006 and the Ph.D. degree in computer application from Shanghai University in 2011. Now he is a lecturer at College of Computer Engineering and Science, Shanghai Business School. HE has published papers in some well-known international Journals and IEEE conferences. His main research interests include software testing, model-based testing, formal method, and algorithm design.