

# A Partial Supply Simulation Relation and its Proof System in PADS

Xinghua Yao<sup>a,\*</sup>, Hengyang Wu<sup>b</sup>

<sup>a</sup>*School of Electronic and Electric Engineering, Shanghai University of Engineering Science, Shanghai 201620, China*

<sup>b</sup>*School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China*

---

## Abstract

PADS (Process Algebra for Demand and Supply) is a formal framework to analyze hierarchical scheduling in real-time embedded systems. Inspired by the supply simulation relation in PADS, we introduce a partial supply simulation relation in order to describe the fact that an unschedulable task may finish on time. It is more general than the supply simulation relation. Then, we explore some properties of partial supply simulation relation. Furthermore, we establish a proof system for the partial supply simulation relation in a decomposing-composing way, which helps to infer tasks' partial schedulabilities. Finally, it is proved that the proof system is sound and complete with respect to the semantic definition of partial supply simulation relation.

*Keywords:* proof system; process algebra; scheduling analysis; resource demand; resource supply

(Submitted on July 25, 2017; Revised on August 30, 2017; Accepted on September 15, 2017)

(This paper was presented at the Third International Symposium on System and Software Reliability.)

© 2017 Totem Publisher, Inc. All rights reserved.

---

## 1. Introduction

As technologies such as Net of Things and Cyber-Physical System are developed, real-time embedded systems appear in more areas, including health care, space exploration, high-speed rail running and so on. Its complexity also increases. Naturally, how to design a real-time embedded system and how to analyze it has attracted researchers' attention. For such problems, some compositional design and analysis methods are proposed and studied. The method of component-based design is widely used in designing real-time embedded system. Its idea is to decompose a large system into some smaller components and then to compose the components by interfaces. The interface is an abstraction of component. When utilizing component technology to construct real-time embedded systems, some corresponding scheduling problems need to be considered. For instance, is a combination of schedulable components schedulable? To analyze scheduling problems in component-based design, some hierarchical scheduling analysis methods are proposed in [4, 5, 8, 15, 16]. Philippou et al. in [14] introduce a framework, named Process Algebra for Demand and Supply (PADS), for analyzing compositional hierarchical scheduling. In the formal model PADS, task's behavior of requesting resources is described by task processes, and resource provider's supplying behavior are modeled by a supply process. Based on the two kinds of processes, a task being schedulable is defined. Further, the schedulability of tasks and their compositionality could be analyzed.

Philippou et al. in [14] defined task's schedulability by requiring that all the demanding scenarios be satisfied on the principle of maximizing resource utilization. Task being schedulable by supply process means that task could definitely finish successfully with a supply's providing resource. Also, a supply simulation relation is presented and proved to be an equivalent description of task's schedulability. According to the definition of schedulability in [14], an unschedulable task does not mean that it is not possible to finish its execution on time. In fact, there possibly exists a requesting-resource scenario being satisfied in an unschedulable task's description. If such a satisfied scenario is executed, then the unschedulable task finishes on time. This paper will focus on how to describe and analyze such a fact that unschedulable tasks possibly finish timely.

\* Corresponding author.

E-mail address: [xhyaosues@aliyun.com](mailto:xhyaosues@aliyun.com)

Inspired by the definition of supply simulation relation in [14], we introduce a partial supply simulation relation to describe the situation that an unschedulable task possibly finishes successfully. Then, tasks being partially schedulable are defined based on the partial supply simulation relation. As we expect, it is proved that schedulability implies partial schedulability. Secondly, some properties are explored in views of the union operation and the choice operator “+”. Similar to schedulability, relations about partial schedulability between tasks and tasks’ normal forms are discussed. Finally, we investigate how to determine whether a task is partially schedulable by a supply or not. A related proof system is established for solving the problem; it is proved to be sound and complete. For a supply process, it could be reduced into the form without the operator “+” in the first time unit. A task could be also decomposed into several parts according to executable maximal responses under the first grant set in a supply process. After decomposing, each gotten part is with a form of containing no “+” in the first time unit. Choose a part such that its resource requests in the first unit time could be satisfied by a supply. By ignoring the first resource request and the first resource grant, we similarly analyze relations between the remainder of task and the remainder of supply until that the task is reduced into *FIN*. Then, these reduced parts are connected by using operators “+” and “:” in a reverse order of decomposing process.

*Related work.* For the compositional analysis, there are two main lines of research. One is based on real-time scheduling theory, such as [4, 5, 8, 15, 16]. The other is a formal approach such as process algebra approach [2, 9, 10, 11, 12, 13], task automata [1, 6, 7], pre-emptive Petri nets [3], etc. PADS is also a process algebra model. It could model such behaviors as requesting resources and supplying resources. A task is described by resource requests with priorities in each time unit, corresponding to task process in PADS. The behavior of providing resource is modelled by a supply process with resource grant sets. Task being schedulable by a supply is defined by using their “||” combination’s evolving behaviors. It requires that the combination never enter into deadlock and there is no resource request unsatisfied in every execution. About PADS, [14] discusses tasks’ schedulability in several ways. It gives an equivalent characterization about a task being schedulable, investigates how to safely compose schedulable systems, gives a method of generating a supply for a schedulable task set, and discusses tasks’ hierarchy by a demand relation. [18] generalizes the results about task’s hierarchy in [14]. A weak demand relation is introduced and a related reasoning method is given. [17] discusses properties about a supply simulation relation and presents a reasoning system. At present, studies about the formal model PADS are mainly centered on task’s schedulability. Analysing the meaning of non-schedulability is also helpful to understand the schedulability. In this paper, we analyse some examples of unschedulable tasks and discuss the fact that unschedulable tasks maybe finish on time. Based on our observation, a concept of partial supply simulation relation is presented and its properties are explored. Then, a proof system is established.

The rest of this paper is organized as follows. Section 2 restates basic concepts and some results about PADS. In Section 3, we introduce a partial supply simulation relation, and discuss a problem that weather a supply simulation relation is a partial supply simulation relation or not. Then, in Section 4 we analyze related properties. Section 5 gives a reasoning system for the proposed relation. And we make detailed analyses for its soundness and completeness in Section 6. Section 7 makes a conclusion.

## 2. Preliminary

In this section, we will state syntax and semantics of PADS, some needed concepts and results in [14, 17, 18].

### 2.1. Syntax and Semantics of PADS

In PADS, a resource demander, i.e. task, which requests resources to execute in each time unit, is modelled by a task process. A resource provider, which supplies resources, is modelled by a supply process. Task processes and supply processes constitutes a system which runs with requesting-supplying match in every unit time. In [14], all the resources, like  $r$ ,  $r_1$ , form a set, denoted by  $\mathbf{R}$ . Every resource in  $\mathbf{R}$  could be requested. So,  $\mathbf{R}$  is also considered to be a set of resource requests. And other two sorts of resources, i.e. granted resources and consumed resources, are also described in PADS. A granted resource means being available. A resource is consumed only if a supplying action and a requesting action happen simultaneously. For example, given  $r \in \mathbf{R}$ ,  $\bar{r}$  represents that the resource  $r$  is provided, and  $\overleftrightarrow{r}$  says the resource  $r$  being employed. A resource-priority pair consists of a resource and a priority number. For instance,  $(r_2, 3)$  and  $(\overleftrightarrow{r_1}, 2)$  are two resource-priority pairs. The pair  $(r_2, 3)$  says that  $r_2$  is requested at the priority 3, and  $(\overleftrightarrow{r_1}, 2)$  means that  $r_1$  is employed at the priority 2.

According to resources being requested or granted, we define three sets, respectively denoted by  $\text{Act}$ ,  $\text{Act}_R$  and  $\text{Act}_G$ . A set containing requested resources or granted resources or consumed resources is denoted by  $\alpha$  and  $\beta$ . All such sets form a set, denoted by  $\text{Act}$ . A resource request set, labeled by  $\rho$ , contains only demanding resources. The symbol  $\text{Act}_R$  means the set consisting of all the requesting-resource sets. And the set  $\text{Act}_G$ , ranged over by  $\gamma$ , represents the collection of all the providing-resource sets. For example,  $\alpha = \{(r_1, 2), \bar{r}_2, (\overset{\leftrightarrow}{r}_3, 3)\} \in \text{Act}$  contains three kinds of resources, i.e. requested resource  $r_1$  with priority 2, provided resource  $r_2$  and employed resource  $r_3$  at the priority 3. The resource request set  $\rho = \{(r_1, 3), (r_2, 2)\} \in \text{Act}_R$  means that  $r_1$  and  $r_2$  are demanded respectively at priorities 3 and 2. The set  $\gamma = \{\bar{r}_2, \bar{r}_3\} \in \text{Act}_G$  means providing two resources  $r_2$  and  $r_3$ . In order to compute resource's priority, a symbol is introduced. For  $\alpha \in \text{Act}$ , we write  $\pi_\alpha(r)$  for resource  $r$ 's priority in  $\alpha$ . And for any granted resource, we consider that it is provided at the priority of 0. For example,  $\beta = \{(r_1, 2), \bar{r}_2, (\overset{\leftrightarrow}{r}_3, 3)\} \in \text{Act}$ . So, the three resources' priorities in  $\beta$  are as follows:  $\pi_\beta(r_1) = 2$ ,  $\pi_\beta(r_2) = 0$  and  $\pi_\beta(r_3) = 3$ . We write  $\alpha^b$  for the remaining parts after removing all the priorities in  $\alpha$ , e.g.  $\{(r_1, 2), \bar{r}_2, (\overset{\leftrightarrow}{r}_3, 3)\}^b = \{r_1, \bar{r}_2, \overset{\leftrightarrow}{r}_3\}$ , and  $\text{res}(\alpha)$  for the collection of resources appearing in  $\alpha$ , e.g.  $\text{res}(\{(r_1, 2), \bar{r}_2, (\overset{\leftrightarrow}{r}_3, 3)\}) = \{r_1, r_2, r_3\}$ . Given  $\alpha_1, \alpha_2 \in \text{Act}$ ,  $\alpha_1$  and  $\alpha_2$  are called to be compatible mutually if, whenever  $r$  appears in the two sets then one occurrence must be a granted resource and the other a request. Its formal definition is as follows:

$$\text{compatible}(\alpha_1, \alpha_2) = \bigwedge_{r \in \text{res}(\alpha_1), \bar{r} \in \text{res}(\alpha_2)} (r \in \alpha_1^b \wedge \bar{r} \in \alpha_2^b) \vee (r \in \alpha_2^b \wedge \bar{r} \in \alpha_1^b) \quad (1)$$

For compatible sets  $\alpha_1$  and  $\alpha_2$ , an operation “ $\oplus$ ” is defined in the following.

$$\begin{aligned} \alpha_1 \oplus \alpha_2 = & \{(r, p) \in \alpha_1 \cup \alpha_2 \mid \bar{r} \notin \alpha_1 \cup \alpha_2\} \\ & \cup \{\bar{r} \in \alpha_1 \cup \alpha_2 \mid (r, p) \notin \alpha_1 \cup \alpha_2\} \\ & \cup \left\{ (\overset{\leftrightarrow}{r}, p) \mid (r, p) \in \alpha_i, \bar{r} \in \alpha_{3-i}, i \in \{1, 2\} \text{ or } (\overset{\leftrightarrow}{r}, p) \in \alpha_1 \cup \alpha_2 \right\} \end{aligned} \quad (2)$$

Three kinds of processes are defined in PADS, including task process, supply process and timed system process. A timed system process may be a task process, or a supply, or their combination. The syntaxes for these processes are given in the following three formulae, i.e. (3), (4) and (5). The set of tasks is denoted by  $\mathbf{T}$ , the symbol  $\mathbf{S}$  for the set of supplies, and  $\mathbf{P}$  for the timed system process set.

$$T ::= \text{FIN} \mid \rho : T \mid T + T \mid C \quad (3)$$

$$S ::= \text{FIN} \mid \gamma : S \mid S + S \mid D \quad (4)$$

$$P ::= \delta \mid T \mid S \mid P \parallel P \quad (5)$$

Here,  $C$  denotes a task constant, and it is defined in the form  $C \stackrel{\text{def}}{=} T$ , where  $T$  may include appearances of  $C$  or other task constants.  $D$  represents a supply constant, and its definition is similar to task constant.  $\delta$  is a locked system process and could not make any execution.

In PADS, for the above three kinds of processes, their semantics are defined by two transition relations. One is an un-prioritized transition relation “ $\rightarrow$ ”, pronounced “two head right arrow”, which is presented in Table 1. The other is a

prioritized relation “ $\rightarrow$ ”, which will be defined later. For a process  $P \in \mathbf{P}$  and  $\alpha \in \mathbf{Act}$ , the symbol  $P \xrightarrow{\alpha}$  means that there is a process  $Q$  satisfying  $P \rightarrow Q$ .

Table 1. Transition rules

(Idle)	$FIN \xrightarrow{\infty} FIN$				
(ActT)	$\rho : T \xrightarrow{\rho} T$		(ActS)	$\gamma : S \xrightarrow{\gamma} S$	
(SumT)	$\frac{T_i \xrightarrow{\alpha} T, i \in \{1,2\}}{T_1 + T_2 \xrightarrow{\alpha} T}$		(SumS)	$\frac{S_i \xrightarrow{\alpha} S, i \in \{1,2\}}{S_1 + S_2 \xrightarrow{\alpha} S}$	
(ConstT)	$\frac{T \xrightarrow{\alpha} T'}{C \xrightarrow{\alpha} T'}$	$C \stackrel{\text{def}}{=} T$	(ConstS)	$\frac{S \xrightarrow{\alpha} S'}{D \xrightarrow{\alpha} S'}$	$D \stackrel{\text{def}}{=} S$
(Par)	$\frac{P_1 \xrightarrow{\alpha_1} P_1', P_2 \xrightarrow{\alpha_2} P_2'}{P_1 \parallel P_2 \xrightarrow{\alpha_1 \oplus \alpha_2} P_1' \parallel P_2'}$	compatible $(\alpha_1, \alpha_2)$			

Definition 1:<sup>[14]</sup> An binary relation  $\prec \in \mathbf{Act} \times \mathbf{Act}$  is called to be a preemption relation, if for every pair  $\langle \alpha, \beta \rangle \in \prec$ , i.e.  $\alpha \prec \beta$ , one of the following holds:

1.  $\left\{ r \mid r \in \alpha^b \text{ or } \bar{r} \in \alpha^b \right\} = \left\{ r \mid r \in \beta^b \text{ or } \bar{r} \in \beta^b \right\}$ ,  $\alpha^b \cap \mathbf{R} \neq \emptyset$  and  $\beta^b \cap \mathbf{R} = \emptyset$ , that is,  $\alpha$  and  $\beta$  use the same employed and provided resources and  $\alpha$  includes unsatisfied requests whereas  $\beta$  does not.
2.  $\mathbf{res}(\alpha) = \mathbf{res}(\beta)$ ,  $\alpha^b \cap \mathbf{R} = \beta^b \cap \mathbf{R} = \emptyset$  and  $\left\{ r \mid \bar{r} \in \alpha^b \right\} \subset \left\{ r \mid \bar{r} \in \beta^b \right\}$ , that is,  $\alpha$  and  $\beta$  have the same resources, both of them have no requests, but  $\beta$  employs more resources than  $\alpha$ .
3.  $\alpha^b = \beta^b$ , for all  $r \in \mathbf{res}(\alpha)$ ,  $\pi_\alpha(r) \leq \pi_\beta(r)$ , and there exists  $r \in \mathbf{res}(\alpha)$ ,  $\pi_\alpha(r) < \pi_\beta(r)$ , that is,  $\alpha$  and  $\beta$  include the same resources,  $\beta$  shows greater or equal priority when using the resources, and there is one resource which is used at a greater priority in  $\beta$  than in  $\alpha$ .

Intuitively, a resource request/grant/consumption set preempts another if it could utilize resources more sufficiently under the same supplying. In Definition 1, Clause 1 means that an action that could get all the requested resources preempts an action that not all the requests are satisfied under the same resource supplying. Clause 2 says that for two actions in which resources requests are all satisfied, the greedy consuming-more-resources one preempts another one. Clause 3 involves resources' priorities. For two actions  $\alpha$  and  $\beta$  with the same requested resources, granted resources and consumed resources, if each resource in  $\beta$  is operated at a priority equal to or higher than  $\alpha$ , and at least there exists a resource with a higher priority in  $\beta$ , then  $\beta$  preempts  $\alpha$ .

Now, based on the transition relation “ $\rightarrow$ ” in Table 1 we could give the definition of prioritized transition relation “ $\rightarrow$ ”. If  $P \xrightarrow{\alpha} Q$ , and there is no  $\beta$  such that  $P \xrightarrow{\beta}$  and  $\alpha \prec \beta$ , we say that there exists a transition  $P \xrightarrow{\alpha} Q$ , i.e.

$$\frac{P \xrightarrow{\alpha} Q}{P \xrightarrow{\alpha} Q} \quad \text{there is no } \beta \text{ such that } P \xrightarrow{\beta} \text{ and } \alpha \prec \beta \quad (6)$$

According to [14],  $P \xrightarrow{\alpha}$  means that there is a process  $P'$  with  $P \xrightarrow{\alpha} P'$ ,  $P \rightarrow$  signifies that there is  $\alpha$  satisfying  $P \xrightarrow{\alpha}$ , and  $P \Rightarrow P'$  means that there exist  $\alpha_1, \dots, \alpha_n$  and  $P_1, \dots, P_n$ ,  $n \geq 1$ , such that  $P \xrightarrow{\alpha_1} P_1 \xrightarrow{\alpha_2} \dots P_{n-1} \xrightarrow{\alpha_n} P_n = P'$ . Moreover,  $\mathbf{T}^*$  denotes the collection of all the processes with the form  $T_1 \parallel \dots \parallel T_n$ ,

$n \geq 1$ . In other words,  $\mathbf{T}^*$  contains all the task processes and tasks' combinations by the operation “ $\parallel$ ”. Similarly,  $\mathbf{S}^*$  means the set of all the processes with the form  $S_1 \parallel \dots \parallel S_n$ ,  $n \geq 1$ . And  $\sum_{i \in I} T_i$  denotes  $T_{i_1} + \dots + T_{i_n}$ , where  $I = \{i_1, \dots, i_n\}$ .

Definition 2:<sup>[14]</sup> Let  $\alpha, \beta \in \mathbf{Act}$ .

- We write  $\mathbf{sat}(\beta, \alpha)$  if  $\mathbf{res}(\beta) \subseteq \mathbf{res}(\alpha)$ . For  $\rho \in \mathbf{Act}_{\mathbb{R}}$  and  $\gamma \in \mathbf{Act}_{\mathbb{G}}$ , if  $\mathbf{sat}(\rho, \gamma)$  then we say that resource request set  $\rho$  is satisfied by resource grant set  $\gamma$ .
- Given  $P \in \mathbf{P}$ , we write  $\beta \trianglelefteq_P \alpha$  if  $\mathbf{sat}(\beta, \alpha)$  and there is no  $\gamma \in \mathbf{Act}$  satisfying  $P \xrightarrow{\gamma} P'$ ,  $\mathbf{sat}(\gamma, \alpha)$  and either  $\beta^b \subset \gamma^b$  or  $\beta^b = \gamma^b$  and  $\beta < \gamma$ . If  $\beta \trianglelefteq_P \alpha$ , then  $\beta$  is called to be a maximal response of  $P$  in respect of  $\alpha$ .

Example 1: Assume  $T = \{(r_1, 2), (r_2, 2)\} : \{(r_1, 3)\} : \mathit{FIN} + \{(r_2, 3)\} : \mathit{FIN} + \{(r_1, 2), (r_2, 1)\} : \mathit{FIN}$ ,  $\alpha_1 = \{(r_1, 2), (r_2, 2)\}$ ,  $\alpha_2 = \{(r_2, 3)\}$ ,  $\alpha_3 = \{(r_1, 2), (r_2, 1)\}$  and  $\gamma = \overline{\{r_1, r_2, r_3\}}$ . Then, task  $T$  has only two transitions, i.e.  $T \xrightarrow{\alpha_1} \{(r_1, 3)\} : \mathit{FIN}$  and  $T \xrightarrow{\alpha_2} \mathit{FIN}$ . And  $T \xrightarrow{\alpha_3}$ . Because  $\alpha_3 < \alpha_1$ . Besides, we could also get  $\mathbf{sat}(\alpha_1, \gamma)$ ,  $\alpha_2^b = \{r_2\} \subset \alpha_1^b = \{r_1, r_2\}$ . Moreover,  $\alpha_1 \trianglelefteq_T \gamma$ ,  $\alpha_2 \not\trianglelefteq_T \gamma$ , and  $\alpha_3 \trianglelefteq_T \gamma$ .

## 2.2. Supply Simulation Relation

Definition 3:<sup>[14]</sup> A task  $T \in \mathbf{T}^*$  is schedulable by supply  $S \in \mathbf{S}^*$  if any  $T \parallel S \Rightarrow P$  implies: (i)  $P \rightarrow$  and (ii) for every  $P \xrightarrow{\alpha}$  the equation “ $\alpha^b \cap \mathbf{R} = \emptyset$ ” holds.

Definition 3 says, task  $T$  being schedulable by supply  $S$  means that the system  $T \parallel S$  could evolve infinitely, and task  $T$ 's resource request in every evolution procedure could be satisfied by the supply process  $S$ .

Definition 4:<sup>[14]</sup> A binary relation  $\mathcal{S} \subseteq \mathbf{T}^* \times \mathbf{S}^*$  is a supply simulation relation if for every  $(T, S) \in \mathcal{S}$ ,  $S \rightarrow$ , and if  $S \xrightarrow{\alpha} S'$  then

1. there exists  $T \xrightarrow{\beta} T'$  with  $\mathbf{sat}(\beta, \alpha)$  and  $(T', S') \in \mathcal{S}$ , and
2. for any  $\rho$ , if  $T \xrightarrow{\rho} T'$  and  $\rho \trianglelefteq_T \alpha$ , then  $(T', S') \in \mathcal{S}$ .

If there is a supply simulation relation including  $\langle T, S \rangle$ , then we write  $S \models T$ .

Example 2: Let  $T = \{(r_1, 2), (r_2, 3)\} : \{(r_1, 2)\} : \mathit{FIN} + \{(r_2, 3)\} : \{(r_2, 1)\} : \mathit{FIN} + \{(r_1, 3), (r_2, 1)\} : \{(r_3, 3)\} : \mathit{FIN}$ ,  $S = \overline{\{r_1, r_2\}} : \overline{\{r_1, r_3\}} : \mathit{FIN}$ . Then, we may structure a supply simulation relation  $\{(T, S), (T_1, S_1), (T_2, S_1), (T_3, S_2)\}$ , where  $T_1 = \{(r_1, 2)\} : \mathit{FIN}$ ,  $T_2 = \{(r_3, 3)\} : \mathit{FIN}$ ,  $T_3 = \mathit{FIN}$ ,  $S_1 = \overline{\{r_1, r_3\}} : \mathit{FIN}$  and  $S_2 = \mathit{FIN}$ . Thus,  $S \models T$ .

Lemma 1:<sup>[14]</sup> A task  $T \in \mathbf{T}^*$  is schedulable by supply  $S \in \mathbf{S}^*$  if and only if  $S \models T$ .

Lemma 1 shows that a task being schedulable by a supply is equivalently characterized by a supply simulation relation.

Definition 5:<sup>[18]</sup> Let  $T, T' \in \mathbf{T}$ ,  $\gamma \in \mathbf{Act}_{\mathbb{G}}$ . If for any  $\rho \in \mathbf{Act}_{\mathbb{R}}$ ,  $T \xrightarrow{\rho}$  with  $\rho \trianglelefteq_T \gamma$  implies  $T' \xrightarrow{\rho}$  with  $\rho \trianglelefteq_{T'} \gamma$ , then we say, task  $T'$  preserves executable and maximal actions in  $T$  under  $\gamma$ , denoted by  $\trianglelefteq_T^T(\gamma)$ .

## 2.3. Task's Depth

Definition 6:<sup>[17]</sup> Assume task  $T \in \mathbf{T}$  and  $T \neq \mathit{FIN}$ . If an evolution sequence  $s : T \xrightarrow{\rho_1} T_1 \xrightarrow{\rho_2} \dots \xrightarrow{\rho_n} T_n$  satisfies:  $T_n = \mathit{FIN}$  and  $T_i \neq \mathit{FIN}$ ,  $i = 1, \dots, n-1$ , then  $s$  is called to be a terminated evolution sequence of task  $T$ , and the length of  $s$  is  $n$ , denoted by  $L(s) = n$ .

Definition 7:<sup>[17]</sup> Assuming that task  $T \in \mathbb{T}$  has no task constant and  $T \neq FIN$ , task  $T$ 's depth, denoted by  $D(T)$ , is defined to be maximal length of  $T$ 's terminated evolution sequences, i.e.  $D(T) = \max \{L(s) \mid s \text{ is a terminated evolution sequence of } T\}$ .

Furthermore, we set a convention that the task  $FIN$ 's depth is 0, i.e.  $D(FIN) = 0$ .

Proposition 1:<sup>[17]</sup> Let  $T \in \mathbb{T}$  be a task process without containing task constant, and  $T \neq FIN$ . Then,  $D(T) \geq 1$ .

#### 2.4. Task's Normal Form

Definition 8:<sup>[17]</sup> Assume that task  $T \in \mathbb{T}$  has no task constant. Then,  $T$ 's normal form, denoted by  $\mathcal{N}(T)$ , is inductively defined as follows:

1.  $\mathcal{N}(T) = FIN$ , if  $T = FIN$ ;
2.  $\mathcal{N}(T) = \rho : \mathcal{N}(T_1)$ , if  $T = \rho : T_1$ ;
3.  $\mathcal{N}(T) = \sum_{i=1}^{m+n} T_i'$ , if  $T = T_1 + T_2$ . Here,  $\mathcal{N}(T_1) = \sum_{i=1}^m T_{1i}$ ,  $\mathcal{N}(T_2) = \sum_{j=1}^n T_{2j}$ ;  $T_i' = T_{1i}$ ,  $i = 1, \dots, m$ ;  $T_i' = T_{2, i-m}$ ,  $i = m+1, \dots, m+n$ .

For task  $T \in \mathbb{T}$ , if there is a task  $T'$  with  $T = \mathcal{N}(T')$ , then  $T$  is called to be with normal form.

Example 3: Suppose  $T = \rho_2 : FIN + (FIN + \rho_3 : (\rho_1 : FIN + (\emptyset : FIN + \rho_3 : FIN)))$ . Then,  $\mathcal{N}(T) = \rho_2 : FIN + FIN + \rho_3 : (\rho_1 : FIN + \emptyset : FIN + \rho_3 : FIN)$ .

Proposition 2:<sup>[17]</sup> Assume that task  $T \in \mathbb{T}$  has no task constant. Then, task  $T$ 's normal form  $\mathcal{N}(T)$  is unique, and  $\mathcal{N}(T)$ 's form is  $\sum_{i=1}^n T_i$ , where  $T_i$  is  $FIN$  or with the form of  $\rho : T'$ ,  $i = 1, \dots, n$ .

Proposition 3:<sup>[17]</sup> Let  $T \in \mathbb{T}$  be a task without containing task constant.

- (1) If  $T \xrightarrow{\rho} T'$ , then  $\mathcal{N}(T) \xrightarrow{\rho} \mathcal{N}(T')$ .
- (2) If  $\mathcal{N}(T) \xrightarrow{\rho} T_0$ , then there exists  $T \xrightarrow{\rho} T'$  such that  $\mathcal{N}(T') = T_0$ .

Corollary 1:<sup>[17]</sup> Let  $T \in \mathbb{T}$  be a task without containing task constant.

- (1) If  $T \xrightarrow{\rho} T'$ , then  $\mathcal{N}(T) \xrightarrow{\rho} \mathcal{N}(T')$ .
- (2) If  $\mathcal{N}(T) \xrightarrow{\rho} T_0$ , then there exists  $T \xrightarrow{\rho} T'$  such that  $\mathcal{N}(T') = T_0$ .

#### 2.5. Guarded Supply Process

Definition 9:<sup>[17]</sup> Assume supply constant  $D \stackrel{\text{def}}{=} S$ . If  $S$  has only one constant  $D$ , and contains  $FIN$  or a resource grant set  $\gamma \in \text{Act}_{\mathcal{G}}$ , then  $D$  is called to be a guarded supply constant. For any supply process  $S' \in \mathbb{S}$ , if each constant in  $S'$  is guarded then  $S'$  is called to be a guarded supply.

Example 4: Three supply processes  $D_1 \stackrel{\text{def}}{=} \gamma : D_1$ ,  $D_2 \stackrel{\text{def}}{=} FIN + D_2$  and  $D_3 \stackrel{\text{def}}{=} \gamma_1 : (FIN + \gamma_2 : D_3) + \gamma_3 : D_3$  are guarded supply constants. Further,  $S_1 = \gamma_2 : D_1 + FIN$  and  $S_2 = \gamma : D_2 + D_3$  are guarded supplies.

Property 1:<sup>[17]</sup> Assume that  $S \in \mathbb{S}$  is guarded. Then,  $S \rightarrow$ .

Property 2:<sup>[17]</sup> Assume that  $S \in \mathbb{S}$  is guarded, and  $S \xrightarrow{\gamma} S'$ . Then,  $S'$  is still guarded.

### 3. Partial Supply Simulation Relation

Firstly, let's analyze the following example.

Example 5: Let supply  $S = \overline{\{r_1, r_2\}}: \overline{\{r_2\}}: FIN$ , task  $T = \{(r_1, 1)\}: \{(r_3, 2)\}: FIN + \{(r_2, 2)\}: \{(r_2, 1)\}: FIN$ . Then, according to Definition 4 we could get  $S \not\models T$ , i.e.  $T$  is not schedulable by  $S$ .

In fact, in Example 5 under the supply  $S$ , task  $T$  could get its required resources in the first time unit in either the resource requesting scenario  $\{(r_1, 1)\}: \{(r_3, 2)\}: FIN$  or the scenario  $\{(r_2, 2)\}: \{(r_2, 1)\}: FIN$ . However, the resource  $r_3$  demanded in the second time unit by the scenario of  $\{(r_1, 1)\}: \{(r_3, 2)\}: FIN$  could not be supplied by the supply  $S$ . So,  $S \not\models T$ .

However, for the demanding scenario of  $\{(r_2, 2)\}: \{(r_2, 1)\}: FIN$ , the requested resource  $r_2$  in the first time unit could be satisfied by the supply  $S$ , and the required resource  $r_2$  in the second time unit is also supplied by  $S$ . Thus, under the supply  $S$  task  $T$ 's requesting-resource scenario of  $\{(r_2, 2)\}: \{(r_2, 1)\}: FIN$  is satisfied and it could be finished on time.

As we can see from the above analyses, even if a task is not schedulable by a supply, some demanding scenarios may be satisfied. In order to describe this kind of situation, we present a concept of partial supply simulation relation in the following.

Definition 10 (Partial Supply Simulation Relation): A binary relation  $\mathcal{S} \subseteq \mathbf{T}^* \times \mathbf{S}^*$  is a partial supply simulation relation if for any pair  $(T^*, S^*) \in \mathcal{S}$ ,  $S^* \rightarrow \cdot$ , and " $S^* \xrightarrow{\gamma} S_0^*$ " implies that there exists a transition  $T^* \xrightarrow{\gamma} T_0^*$  satisfying  $\rho \triangleleft_{T^*} \gamma$  and  $(T_0^*, S_0^*) \in \mathcal{S}$ .

Definition 11: For  $(T^*, S^*) \in \mathbf{T}^* \times \mathbf{S}^*$ , if there is a partial supply simulation relation including  $(T^*, S^*)$ , then we say, task  $T^*$  is partially schedulable by supply  $S^*$ , denoted by  $S^* \models_{part} T^*$ .

Task  $T^*$  being partially schedulable by supply  $S^*$ , satisfies the following two conditions:

- (1) Supply  $S^*$  could execute transitions;
- (2) For any resource grant set supplied by  $S^*$ ,  $T^*$  has corresponding resource requests. If  $S^*$  supplies the resource grant set  $\gamma$  and evolves into  $S_0^*$ , i.e.  $S^* \xrightarrow{\gamma} S_0^*$ , then  $T^*$  has such a maximal response with respect to  $\gamma$  that its evolved task is still satisfied by  $S_0^*$ .

Example 6: Let task  $T = \{(r_1, 2), (r_2, 3)\}: \{(r_2, 3)\}: FIN + \{(r_1, 3), (r_2, 1)\}: (\{(r_1, 2)\}: FIN + \{(r_3, 2)\}: FIN)$ , supply  $S = \overline{\{r_1, r_2\}}: \overline{\{r_3\}}: FIN$ . Then, it is not difficult to validate that  $(T, S), (\{(r_1, 2)\}: FIN + \{(r_3, 2)\}: FIN, \overline{\{r_3\}}: FIN), (FIN, FIN)$  is a partial supply simulation relation. So,  $T$  is partially schedulable by  $S$ .

In fact,  $T$  in Example 6 has two demanding scenarios, i.e.  $\{(r_1, 2), (r_2, 3)\}: \{(r_2, 3)\}: FIN$  and  $\{(r_1, 3), (r_2, 1)\}: (\{(r_1, 2)\}: FIN + \{(r_3, 2)\}: FIN)$ . In the first time unit resources  $r_1$  and  $r_2$  supplied by  $S$  are just the required resources in the two scenarios of  $T$ . Anyone in the two scenarios may get resources  $r_1$  and  $r_2$  in the competition. For the scenario of  $\{(r_1, 3), (r_2, 1)\}: (\{(r_1, 2)\}: FIN + \{(r_3, 2)\}: FIN)$ , it has two requesting-resources scenarios after getting  $r_1$  and  $r_2$ , which are  $\{(r_1, 2)\}: FIN$  and  $\{(r_3, 2)\}: FIN$ . Because of  $S$  supplying resource  $r_3$  in the second time unit,  $\{(r_3, 2)\}: FIN$  could finish. From the above analysis, we know, task  $T$ 's requesting resources on the execution path  $T \xrightarrow{\{(r_1, 3), (r_2, 1)\}} \{(r_1, 2)\}: FIN + \{(r_3, 2)\}: FIN \xrightarrow{\{(r_3, 2)\}} FIN$  could be supplied on time by supply  $S$ . So, task  $T$  is partially schedulable by supply  $S$ .

Theorem 1: Assume  $\mathcal{S} \subseteq \mathbf{T}^* \times \mathbf{S}^*$ . If  $\mathcal{S}$  is a supply simulation relation, then it is a partial supply simulation relation.

Proof: Let  $(T', S')$  be any ordered pair in  $\mathcal{S}$ . According to Definition 4, we know  $S' \rightarrow$ . Suppose  $S' \xrightarrow{\gamma} S''$ . Then, by the first condition in Definition 4 there exists a transition  $T' \xrightarrow{\rho_1} T'_1$  with  $\text{sat}(\rho_1, \gamma)$ . Furthermore, there exists a transition  $T' \xrightarrow{\rho_2} T'_2$  such that  $\rho_2 \triangleleft_{T'} \gamma$ . According to the second condition in Definition 4, we can get  $(T'_2, S'') \in \mathcal{S}$ .

The above analyses show that  $(T', S')$  as an ordered pair in the relation  $\mathcal{S}$  satisfies conditions in Definition 10. So,  $\mathcal{S}$  is a partial supply simulation relation. Thereby, the theorem is proved.  $\blacksquare$

Theorem 1 tells us that a supply simulation relation is a partial supply simulation relation. Further, according to Lemma 1, we could get the following corollary from Definition 4 and Definition 11.

Corollary 2: Assume supply  $S \in \mathbf{S}^*$  and task  $T \in \mathbf{T}^*$ . If  $S \models T$ , then  $S \models_{\text{part}} T$ .

#### 4. Some Properties

Analyzing properties about the partial supply simulation relation is helpful in understanding task's partial schedulability. In the following, we will present some related results.

Property 3: Let  $\mathcal{S}_i$  be partial supply simulation relation,  $i = 1, 2, \dots, n, \dots$ . Then,  $\bigcup_{i=1}^{+\infty} \mathcal{S}_i$  is a partial supply simulation relation.

Property 4: Assume guarded supply  $S_1 + S_2 \in \mathbf{S}$  and task  $T \in \mathbf{T}$ . Then,  $S_1 + S_2 \models_{\text{part}} T$  if and only if  $S_1 \models_{\text{part}} T$  and  $S_2 \models_{\text{part}} T$ .

Proposition 4: Assume task  $T \in \mathbf{T}^*$ . Then,  $FIN \models_{\text{part}} T$  if and only if  $\emptyset : FIN \models_{\text{part}} T$ .

Property 3 tells us that the union of partial supply simulation relations is still a partial supply simulation relation. Property 4 shows that task's partial schedulability under a supply is independent of the outermost operator “+” in the supply process. Proposition 4 says supply  $FIN$ 's partial scheduling capacity is the same as supply  $\emptyset : FIN$ .

Theorem 2: Assume that task  $T \in \mathbf{T}$  has no task constant, and supply process  $S \in \mathbf{S}^*$ . Then,  $S \models_{\text{part}} T$  if and only if  $S \models_{\text{part}} \mathcal{N}(T)$ .

Proof:  $\Leftarrow$ : By the condition “ $S \models_{\text{part}} \mathcal{N}(T)$ ”, we could assume a partial supply simulation relation  $\mathcal{S}$  with  $(\mathcal{N}(T), S) \in \mathcal{S}$ .

Let  $\mathcal{S}_1 = \{(\mathcal{N}(T'), S') \mid (T', S') \in \mathcal{S}, T' \text{ does not contain any task constant}\}$ . For a task with normal form, its normal form is itself. Then,  $(\mathcal{N}(T), S) \in \mathcal{S}_1$ . Suppose that  $(\mathcal{N}(T_1), S_1) \in \mathcal{S}_1$  is any ordered pair. Here,  $(T_1, S_1) \in \mathcal{S}$ , and  $T_1$  is a task process without containing task constant. Then,  $S_1 \rightarrow$ . Assume  $S_1 \xrightarrow{\gamma} S'_1$ . By the condition “ $(T_1, S_1) \in \mathcal{S}$ ”, there exists a transition  $T_1 \xrightarrow{\rho_1} T'_1$  such that  $\rho_1 \triangleleft_{T_1} \gamma$  and  $(T'_1, S'_1) \in \mathcal{S}$ . Because of  $T_1$  without task constants,  $T'_1$  does not contain task constants. Furthermore,  $(\mathcal{N}(T'_1), S'_1) \in \mathcal{S}_1$ . By Corollary 1,  $\mathcal{N}(T_1) \xrightarrow{\rho_1} \mathcal{N}(T'_1)$  with  $\rho_1 \triangleleft_{\mathcal{N}(T_1)} \gamma_1$ . The pair  $(\mathcal{N}(T_1), S_1)$  in the relation  $\mathcal{S}_1$  satisfies conditions in Definition 10. Moreover,  $\mathcal{S}_1$  is a partial supply simulation relation.

Let  $\mathcal{S}_2 = \{(T'', S'') \mid (\mathcal{N}(T''), S'') \in \mathcal{S}_1\}$ . Then,  $(T, S) \in \mathcal{S}_2$ . Suppose that a pair  $(T_0, S_0) \in \mathcal{S}_2$  is any ordered pair. Then,  $(\mathcal{N}(T_0), S_0) \in \mathcal{S}_1$ . Moreover,  $S_0 \rightarrow$ . Assume  $S_0 \xrightarrow{\gamma_0} S'_0$ . Then, there exists a transition  $\mathcal{N}(T_0) \xrightarrow{\rho'_0} T_{01}$  such that  $\rho'_0 \triangleleft_{\mathcal{N}(T_0)} \gamma_0$  and  $(T_{01}, S'_0) \in \mathcal{S}_1$ . By Corollary 1, there exists a transition  $T_0 \xrightarrow{\rho'_0} T'_{01}$  with  $\mathcal{N}(T'_{01}) = T_{01}$ . Moreover,  $\rho'_0 \triangleleft_{T_0} \gamma_0$ .



Because of  $(T_{01}, S'_0) \in \mathcal{S}_1$ , i.e.  $(\mathcal{N}(T'_{01}), S'_0) \in \mathcal{S}_1$ , we could get  $(T'_{01}, S'_0) \in \mathcal{S}_2$ . As a pair in the relation  $\mathcal{S}_2$ ,  $(T_0, S_0)$  satisfies conditions in Definition 10. Thus,  $\mathcal{S}_2$  is a partial supply simulation relation. By  $(T, S) \in \mathcal{S}_2$ , we have  $S \models_{part} T$ .

$\Rightarrow$  : According to  $S \models_{part} T$ , there is a partial supply simulation relation  $\mathcal{S}$  satisfying  $(T, S) \in \mathcal{S}$ . Let  $\mathcal{S}' = \{(T', S') \in \mathcal{S} \mid T' \text{ does not contain any task constant}\}$ . Because  $T$  does not contain task constants. We could get  $(T, S) \in \mathcal{S}'$ . Suppose that  $(T_1, S_1) \in \mathcal{S}'$  is any ordered pair. By  $(T_1, S_1) \in \mathcal{S}$ , we know  $S_1 \rightarrow$ . Assume  $S_1 \xrightarrow{\gamma_1} S'_1$ . Then by  $(T_1, S_1) \in \mathcal{S}'$ , there exists a transition  $T_1 \xrightarrow{\rho_1} T'_1$  such that  $\rho_1 \triangleleft_{T_1} \gamma_1$  and  $(T'_1, S'_1) \in \mathcal{S}'$ .  $T_1$  does not contain any task constant. So,  $T'_1$  does not too. Moreover,  $(T'_1, S'_1) \in \mathcal{S}'$ . As a pair in  $\mathcal{S}'$ ,  $(T_1, S_1)$  satisfies conditions in Definition 10. Thus,  $\mathcal{S}'$  is a partial supply simulation relation.

Let  $\mathcal{S}'' = \{(\mathcal{N}(T''), S'') \mid (T'', S'') \in \mathcal{S}'\}$ . Because of  $(T, S) \in \mathcal{S}'$ , we could get  $(\mathcal{N}(T), S) \in \mathcal{S}''$ . Suppose that  $(\mathcal{N}(T_2), S_2) \in \mathcal{S}''$  is any ordered pair. Here,  $(T_2, S_2) \in \mathcal{S}'$ . Then,  $S_2 \rightarrow$ . Assume  $S_2 \xrightarrow{\gamma_2} S'_2$ . According to that  $\mathcal{S}'$  is a partial supply simulation relation, there exists a transition  $T_2 \xrightarrow{\rho_2} T'_2$  such that  $\rho_2 \triangleleft_{T_2} \gamma_2$  and  $(T'_2, S'_2) \in \mathcal{S}'$ . By Corollary 1, we could get  $\mathcal{N}(T_2) \xrightarrow{\rho_2} \mathcal{N}(T'_2)$  and  $\rho_2 \triangleleft_{\mathcal{N}(T_2)} \gamma_2$ . Because of  $(\mathcal{N}(T'_2), S'_2) \in \mathcal{S}''$ , the pair  $(\mathcal{N}(T_2), S_2)$  in the relation  $\mathcal{S}''$  satisfies conditions in Definition 10. Thus,  $\mathcal{S}''$  is a partial supply simulation relation. According to  $(\mathcal{N}(T), S) \in \mathcal{S}''$ , we could get  $S \models_{part} \mathcal{N}(T)$ . ■

Theorem 2 shows that a task has the same partial schedulability as its normal form.

## 5. Proof System $P_{\models_{part}}$

This section will explore how to reason about whether there exists a partial supply simulation relation for a task process and a supply process or not. We will solve the problem by establishing a proof system, named  $P_{\models_{part}}$ .

The proof system  $P_{\models_{part}}$  consists of an axiom and six reasoning rules, presented in Table 2. In the following, we informally explain the rules.

- Rule  $R_1$  means that task  $FIN$  is partially schedulable by any guarded supply process. In fact, task  $FIN$  does not need any resource.
- Rule  $R_2$  says, if task  $\rho:T$ 's requesting resources in the first time unit could be supplied by supply process  $\gamma:S$ , and  $T$  is partially schedulable by  $S$ , then  $\rho:T$  is partially schedulable by  $\gamma:S$ .
- Rule  $R_3$  means that if task  $T_1$  could be partially schedulable by supply  $\gamma:S$ , and it could preserve the original executable and maximal action with respect to the resource grant set  $\gamma$ , then the new task  $T_1 + T_2$  could be partially schedulable by  $\gamma:S$ .
- Rule  $R_4$  says that, for a guarded supply constant  $D \stackrel{\text{def}}{=} S$  the supply  $D$ 's partially scheduling capability is the same as the supply  $S$ 's. Because  $D$ 's supplying-resources scenarios are the same as  $S$ 's scenarios.
- Rule  $R_5$  means that supply  $\emptyset:FIN$ 's partial scheduling capacity is the same as supply  $FIN$ 's. Because both supply processes  $FIN$  and  $\emptyset:FIN$  do not supply any resources.
- Rule  $R_6$  originates from Property 4. A task is partially schedulable, respectively by two supply processes, if and only if it is also partially schedulable by their combination of “+”.
- Rule  $R_7$  says, task's partial schedulability is not affected by the combining way of its summands.

Definition 12: Suppose supply process  $S \in \mathbf{S}$ , task process  $T \in \mathbf{T}$ . If “ $S \parallel_{part} T$ ” is obtained after finitely utilizing rules in the proof system  $P_{\models_{part}}$ ,  $S \parallel_{part} T$  is called to be a theorem in  $P_{\models_{part}}$ , denoted by  $\vdash_{P_{\models_{part}}} S \parallel_{part} T$ .

In the following, an example is given in order to illustrate how to apply the rules in  $P_{|=part}$ .

Example 7: Assume task  $T = \{(r_3, 2)\} : \{(r_2, 1)\} : FIN + \{(r_2, 2)\} : (\{(r_1, 1)\} : \{(r_1, 3)\} : FIN + \emptyset : FIN)$ , supply constant  $D = S$ .  
Here,  $S = \overline{\{r_1, r_3\}} : \overline{\{r_1, r_2\}} : FIN + \overline{\{r_2\}} : D$ . Then,  $\vdash_{P_{|=part}} D \Vdash_{part} T$ .

Table 2. Proof system  $P_{|=part}$ 

$R_1$	$\frac{}{S \Vdash_{part} FIN}$
$R_2$	$\frac{S \Vdash_{part} T, \text{sa}(\rho, \gamma)}{\gamma : S \Vdash_{part} \rho : T}$
$R_3$	$\frac{\gamma : S \Vdash_{part} T_1, \triangleleft_{T_1}^{T_1+T_2}(\gamma)}{\gamma : S \Vdash_{part} T_1 + T_2}$
$R_4$	$\frac{\forall \gamma \in \text{Ac}_{\mathbb{E}}, \forall S' \in \mathbf{S}. ((S \xrightarrow{\gamma} S') \vee \gamma : S' \Vdash_{part} T)}{D \Vdash_{part} T}$ Here, $D = S$ is a guarded supply constant.
$R_5$	$\frac{\emptyset : FIN \Vdash_{part} T}{FIN \Vdash_{part} T}$
$R_6$	$\frac{S_1 \Vdash_{part} T, S_2 \Vdash_{part} T}{S_1 + S_2 \Vdash_{part} T}$
$R_7$	$\frac{S \Vdash_{part} \sum_{i=1}^n T_i, \sigma \text{ is a permutation over the set of } \{1, 2, \dots, n\}}{S \Vdash_{part} \sum_{i=1}^n T_{\sigma(i)}}$

Proof of Example 7. According to  $D$ 's definition, we know that  $D$  is a guarded supply constant. So, we could use rules in the proof system  $P_{|=part}$ .

- (1)  $FIN \Vdash_{part} FIN$  Rule  $R_1$
- (2)  $\overline{\{r_1, r_2\}} : FIN \Vdash_{part} \{(r_2, 1)\} : FIN$  Rule  $R_2$  and (1)
- (3)  $\overline{\{r_1, r_3\}} : \overline{\{r_1, r_2\}} : FIN \Vdash_{part} \{(r_3, 2)\} : \{(r_2, 1)\} : FIN$  Rule  $R_2$  and (2)
- (4)  $\overline{\{r_1, r_3\}} : \overline{\{r_1, r_2\}} : FIN \Vdash_{part} T$  Rule  $R_3$  and (3)
- (5)  $\overline{\{r_1, r_2\}} : FIN \Vdash_{part} \{(r_1, 3)\} : FIN$  Rule  $R_2$  and (1)
- (6)  $\overline{\{r_1, r_3\}} : \overline{\{r_1, r_2\}} : FIN \Vdash_{part} \{(r_1, 1)\} : \{(r_1, 3)\} : FIN$  Rule  $R_2$  and (5)
- (7)  $\overline{\{r_1, r_3\}} : \overline{\{r_1, r_2\}} : FIN \Vdash_{part} \{(r_1, 1)\} : \{(r_1, 3)\} : FIN + \emptyset : FIN$  Rule  $R_3$  and (6)
- (8)  $D \Vdash_{part} FIN$  Rule  $R_1$
- (9)  $\overline{\{r_2\}} : D \Vdash_{part} \emptyset : FIN$  Rule  $R_2$  and (8)
- (10)  $\overline{\{r_2\}} : D \Vdash_{part} \emptyset : FIN + \{(r_1, 1)\} : \{(r_1, 3)\} : FIN$  Rule  $R_3$  and (9)
- (11)  $\overline{\{r_2\}} : D \Vdash_{part} \{(r_1, 1)\} : \{(r_1, 3)\} : FIN + \emptyset : FIN$  Rule  $R_7$  and (10)
- (12)  $D \Vdash_{part} \{(r_1, 1)\} : \{(r_1, 3)\} : FIN + \emptyset : FIN$  Rule  $R_4$ , (7) and (11)
- (13)  $\overline{\{r_2\}} : D \Vdash_{part} \{(r_2, 2)\} : (\{(r_1, 1)\} : \{(r_1, 3)\} : FIN + \emptyset : FIN)$  Rule  $R_2$  and (12)
- (14)  $\overline{\{r_2\}} : D \Vdash_{part} \{(r_2, 2)\} : (\{(r_1, 1)\} : \{(r_1, 3)\} : FIN + \emptyset : FIN) + \{(r_3, 2)\} : \{(r_2, 1)\} : FIN$  Rule  $R_3$  and (13)
- (15)  $\overline{\{r_2\}} : D \Vdash_{part} \{(r_3, 2)\} : \{(r_2, 1)\} : FIN + \{(r_2, 2)\} : (\{(r_1, 1)\} : \{(r_1, 3)\} : FIN + \emptyset : FIN)$   
i.e.  $\overline{\{r_2\}} : D \Vdash_{part} T$  Rule  $R_7$  and (14)

(16)  $D \Vdash_{part} T$  Rule  $R_4$ , (4) and (15)

Thus,  $\vdash_{P_{part}} D \Vdash_{part} T$ . ■

## 6. Soundness and Completeness

This section will discuss the soundness and completeness about the system  $P_{part}$ , and give their proofs.

**Theorem 3 (Soundness):** Suppose that  $S \in \mathbf{S}$  is guarded, and task  $T \in \mathbf{T}$ . If  $\vdash_{P_{part}} S \Vdash_{part} T$ , then  $S \models_{part} T$ .

**Proof:** Considering that the space is limited, here we only verify rules  $R_1$  and  $R_4$ . The remaining rules could be proved similarly.

(1) For Rule  $R_1$ , let  $\mathcal{S} = \{(FIN, S_0) \mid S_0 \in \mathbf{S} \text{ is a guarded supply process}\}$ . Then,  $(FIN, S) \in \mathcal{S}$ .

Suppose that  $(FIN, S_0) \in \mathcal{S}$  is any ordered pair. By Property 1 we know  $S_0 \rightarrow$ . Furthermore, we could assume  $S_0 \xrightarrow{\gamma} S'_0$ . By Property 2,  $S'_0$  is also a guarded supply. So,  $(FIN, S'_0) \in \mathcal{S}$ . As we know,  $FIN \xrightarrow{\emptyset} FIN$  and  $\emptyset \trianglelefteq_{FIN} \gamma$ . Thus,  $(FIN, S_0)$  satisfies conditions in Definition 10. And  $\mathcal{S}$  is a partial supply simulation relation. Thereby,  $S \models_{part} FIN$ .

(2) For Rule  $R_4$ , by the precondition that  $D \stackrel{def}{=} S$  is a guarded supply constant, we could suppose  $A = \{(\gamma', S'') \mid S \xrightarrow{\gamma'} S''\}$ . According to syntaxes and semantics of supply process, we could get that  $A$  is a finite set. By Property 1, we know  $|A| \geq 1$ . Let  $A = \{(\gamma_i, S_i) \mid i = 1, \dots, n\}$ ,  $1 \leq n \in \mathbf{N}$ . By conditions in Rule  $R_4$ ,  $\gamma_i : S_i \Vdash_{part} T$ ,  $i = 1, \dots, n$ .

Let us make a hypothesis that “ $\gamma : S' \Vdash_{part} T$ ” in the condition “ $\forall \gamma \in \mathbf{Act}_{\mathcal{G}}. \forall S' \in \mathbf{S} ((S \xrightarrow{\gamma} S') \vee \gamma : S' \Vdash_{part} T)$ ” satisfies  $\gamma : S' \models_{part} T$ . Then, for “ $\gamma_i : S_i \Vdash_{part} T$ ” we have  $\gamma_i : S_i \models_{part} T$ ,  $i = 1, \dots, n$ . By Property 3, suppose that  $\mathcal{S}$  is such a partial supply simulation relation that  $(T, \gamma_i : S_i) \in \mathcal{S}$ ,  $i = 1, \dots, n$ .

Let  $\mathcal{S}' = \mathcal{S} \cup \{(T, D)\}$ . For any ordered pair  $(T_0, S_0) \in \mathcal{S}$ , as a pair in the relation  $\mathcal{S}'$  it satisfies conditions in Definition 10. Because  $\mathcal{S}$  is a partial supply simulation relation.

For pair  $(T, D)$ , according to Property 1 we know  $D \rightarrow$ . Assume  $D \xrightarrow{\gamma'_0} S'_0$ . Then, by operational semantics of supply constant we can get  $S \xrightarrow{\gamma'_0} S'_0$ . So,  $(\gamma'_0, S'_0) \in A$ . According to properties of elements in the set  $A$ , we have  $(T, \gamma'_0 : S'_0) \in \mathcal{S}$ . By Definition of 10, there exists a transition  $T \xrightarrow{\rho} T'$  such that  $\rho \trianglelefteq_T \gamma'_0$  and  $(T', S'_0) \in \mathcal{S}$ , which conjugates the transition  $\gamma'_0 : S'_0 \xrightarrow{\gamma'_0} S'_0$ . Because of  $\mathcal{S} \subseteq \mathcal{S}'$ , the pair  $(T, D)$  in the relation  $\mathcal{S}'$  satisfies conditions in Definition 10.

So, each pair in the relation  $\mathcal{S}'$  satisfies conditions in Definition 10. And  $\mathcal{S}'$  is a partial supply simulation relation. Furthermore,  $D \models_{part} T$ . ■

Theorem 2 shows that tasks  $T$  and  $\mathcal{N}(T)$  have the same partial schedulability. Definition 8 shows how computing normal form  $\mathcal{N}(T)$  is not difficult. So, we could analyze the system  $P_{part}$ 's completeness over a collection of tasks with normal forms.

**Theorem 4 (Completeness):** Suppose that  $T \in \mathbf{T}$  has no task constant and  $S \in \mathbf{S}$  is guarded. If  $S \models_{part} \mathcal{N}(T)$ , then  $\vdash_{P_{part}} S \Vdash_{part} \mathcal{N}(T)$ .

Proof: (1) Base:  $D(T) = 0$ .

On the bases of Proposition 1 and Definition 7, we have  $T = FIN$ . Moreover,  $\mathcal{N}(T) = FIN$ . By using Rule  $R_1$  in the proof system  $P_{=part}^l$ , we can get  $\vdash_{P_{=part}} S \parallel_{-part} FIN$ , i.e.  $\vdash_{P_{=part}} S \parallel_{-part} \mathcal{N}(T)$ .

(2) Induction: We make a hypothesis that the original conclusion holds if  $D(T) \leq k$ . In the following we will prove that it also holds if  $D(T) = k + 1$ .

By Property 4 and Rule  $R_6$ , we could suppose  $S = \sum_{i=1}^m S_i$ . Here,  $S_i$  is the form of  $\gamma_0 : S_0$ , or  $FIN$ , or  $S_i$  is a supply constant,  $i = 1, \dots, m$ . According to Proposition 2, we can suppose  $\mathcal{N}(T) = \sum_{j=1}^n T_j$ . Here,  $T_j$  is the form of  $\rho_0 : T_0$  or  $FIN$ ,  $j = 1, \dots, n$ .

Because of  $S \parallel_{=part} \mathcal{N}(T)$  and Property 4, we know  $S_i \parallel_{=part} \mathcal{N}(T)$ ,  $i = 1, \dots, m$ . For any  $i \in \{1, \dots, m\}$ , we will prove  $\vdash_{P_{=part}} S_i \parallel_{-part} \mathcal{N}(T)$  in three cases. The three cases are respectively Case 1:  $S_i$  is the form of  $\gamma_0 : S_0$ , Case 2:  $S_i = FIN$ , and Case 3:  $S_i$  is a supply constant.

(2-1) Case 1:  $S_i$  is with the form  $\gamma_0 : S_0$ .

Let  $S_i = \gamma' : S'$ . Then,  $S_i \xrightarrow{\gamma'} S'$ . By  $S_i \parallel_{=part} \mathcal{N}(T)$ , there exists a transition  $\mathcal{N}(T) \xrightarrow{\rho_0} T_0$  such that  $\rho_0 \trianglelefteq_{\mathcal{N}(T)} \gamma'$  and  $S' \parallel_{=part} T_0$ . We will analyze Case 1 in the following two cases which are “ $FIN \in \{T_j | j = 1, \dots, n\}$ ” and “ $FIN \notin \{T_j | j = 1, \dots, n\}$ ”.

(2-1-a) If  $FIN \in \{T_j | j = 1, \dots, n\}$  then for the simplicity let  $T_1 = FIN$  and  $T_j = \rho_j : T_{j1}$ ,  $j = 2, \dots, n$ .

If  $\rho_0 = \emptyset$ , then  $\emptyset \trianglelefteq_{\mathcal{N}(T)} \gamma'$ . For the simplicity suppose  $A = \{j \in \mathbb{N} : 2 \leq j \leq n | \mathcal{N}(T) \xrightarrow{\rho_j} \}$  and  $\rho_j \trianglelefteq_{\mathcal{N}(T)} \gamma' = \{2, \dots, p\}$ . Here,  $p \leq n$ . Then,  $\trianglelefteq \uparrow \sum_{j=1}^{l+1} T_j (\gamma')$ ,  $l = 1, \dots, n-1$ . By Rule  $R_1$ ,  $\vdash_{P_{=part}} S_i \parallel_{-part} FIN$ , i.e.  $\vdash_{P_{=part}} S_i \parallel_{-part} T_1$ . By using Rule  $R_3$  for  $n-1$  times,  $\vdash_{P_{=part}} S_i \parallel_{-part} \sum_{j=1}^n T_j$ , i.e.  $\vdash_{P_{=part}} S_i \parallel_{-part} \mathcal{N}(T)$ .

If  $\rho_0 \neq \emptyset$ , then  $\rho_0 : T_0 \in \{\rho_j : T_{j1} | j = 2, \dots, n\}$ . To be simple, let  $\rho_0 : T_0 = \rho_2 : T_{21}$  and  $B = \{j \in \mathbb{N} : 3 \leq j \leq n | \mathcal{N}(T) \xrightarrow{\rho_j} \}$  and  $\rho_j \trianglelefteq_{\mathcal{N}(T)} \gamma' = \{3, \dots, q\}$ . Here,  $q \leq n$ . Then,  $\rho_2 \trianglelefteq_{\mathcal{N}(T)} \gamma'$ ,  $S' \parallel_{=part} T_{21}$ ,  $\trianglelefteq \uparrow \sum_{j=2}^{l+1} T_j (\gamma')$ ,  $l = 2, \dots, n-1$ , and  $\trianglelefteq \uparrow \sum_{j=2}^n T_j (\gamma')$ . Because  $\mathcal{N}(T) \xrightarrow{\rho_0} T_0$ , i.e.  $\mathcal{N}(T) \xrightarrow{\rho_2} T_{21}$ . By the definition of transition relation “ $\rightarrow$ ”,  $\mathcal{N}(T) \xrightarrow{\rho_2} T_{21}$ .

By Proposition 3, there exists a transition  $T \xrightarrow{\rho_2} T'_{21}$  with  $\mathcal{N}(T'_{21}) = T_{21}$ . Then, we make an assertion of  $T \xrightarrow{\rho_2} T'_{21}$ . Otherwise, there exists  $T \xrightarrow{\rho'_2}$  such that  $\rho_2 \prec \rho'_2$ . Furthermore,  $\mathcal{N}(T) \xrightarrow{\rho'_2}$  and  $\rho_2 \prec \rho'_2$ . It contradicts with  $\mathcal{N}(T) \xrightarrow{\rho_2} T_{21}$ . By the transition  $T \xrightarrow{\rho_2} T'_{21}$  with  $\mathcal{N}(T'_{21}) = T_{21}$ , we could get  $D(T'_{21}) < D(T)$ . Moreover,  $D(T'_{21}) < k$ . By the hypothesis, we have  $\vdash_{P_{=part}} S' \parallel_{-part} T_{21}$ . By using Rule  $R_2$ ,  $\vdash_{P_{=part}} \gamma' : S' \parallel_{-part} \rho_2 : T_{21}$ , i.e.  $\vdash_{P_{=part}} S_i \parallel_{-part} T_2$ . After using Rule  $R_3$  for  $n-1$  times, we could get  $\vdash_{P_{=part}} S_i \parallel_{-part} \sum_{j=2}^n T_j + T_1$ . By Rule  $R_7$ ,  $\vdash_{P_{=part}} S_i \parallel_{-part} \sum_{j=1}^n T_j$ , i.e.  $\vdash_{P_{=part}} S_i \parallel_{-part} \mathcal{N}(T)$ .

(2-1-b) If  $FIN \notin \{T_j | j=1, \dots, n\}$ , then we could suppose  $T_j = \rho_j : T_{j1}$ ,  $j=1, \dots, n$ . Thereby,  $\rho_0 : T_0 \in \{\rho_j : T_{j1} | j=1, \dots, n\}$ . To be simple, let  $\rho_0 : T_0 = \rho_1 : T_{11}$ ,  $L = \left\{ j \in \mathbb{N} : 2 \leq j \leq n \mid \mathcal{N}(T) \xrightarrow{\rho_j} \text{ and } \rho_j \trianglelefteq_{\mathcal{N}(T)} \gamma' \right\} = \{2, \dots, u\}$ . Here,  $u \leq n$ . Then,  $\rho_1 \trianglelefteq_{\mathcal{N}(T)} \gamma'$ ,  $S' \models_{part} T_{11}$ ,  $\trianglelefteq \uparrow \sum_{j=1}^{l+1} T_j (\gamma')$ ,  $l=1, \dots, n-1$ . Because of the transition  $\mathcal{N}(T) \xrightarrow{\rho_0} T_0$ , i.e.  $\mathcal{N}(T) \xrightarrow{\rho} T_{11}$ , we could get  $\mathcal{N}(T) \xrightarrow{\rho_1} T_{11}$  by the definition of transition relation “ $\rightarrow$ ”. According to Proposition 3, there exists a transition  $T \xrightarrow{\rho_1} T'_{11}$  with  $\mathcal{N}(T'_{11}) = T_{11}$ . Thus,  $T \xrightarrow{\rho} T'_{11}$  with  $\mathcal{N}(T'_{11}) = T_{11}$ . Otherwise, there exists  $T \xrightarrow{\rho'} T'_{11}$  with  $\rho_1 < \rho'_1$ . By Proposition 3,  $\mathcal{N}(T) \xrightarrow{\rho'_1}$  with  $\rho_1 < \rho'_1$ . It contradicts with  $\mathcal{N}(T) \xrightarrow{\rho_1}$ . By the transition  $T \xrightarrow{\rho_1} T'_{11}$ , we know  $D(T'_{11}) < D(T)$ . Furthermore,  $D(T'_{11}) < k$ . By the hypothesis, we could get  $\vdash_{P_{=part}} S' \Vdash_{part} T_{11}$ . By Rule  $R_2$ ,  $\vdash_{P_{=part}} \gamma' : S' \Vdash_{part} \rho_1 : T_{11}$ , i.e.  $\vdash_{P_{=part}} S_i \Vdash_{part} T_1$ . Then, by using Rule  $R_3$  for  $n-1$  times we could get  $\vdash_{P_{=part}} S_i \Vdash_{part} \sum_{j=1}^n T_j$ , i.e.  $\vdash_{P_{=part}} S_i \Vdash_{part} \mathcal{N}(T)$ .

(2-2) Case 2:  $S_i = FIN$ .

The condition “ $S_i \models_{part} \mathcal{N}(T)$ ” means  $FIN \models_{part} \mathcal{N}(T)$ . By Proposition 4,  $\emptyset : FIN \models_{part} \mathcal{N}(T)$ . According to the previous analyses in Case 1, we have  $\vdash_{P_{=part}} \emptyset : FIN \Vdash_{part} \mathcal{N}(T)$ . In accordance with Rule  $R_5$ ,  $\vdash_{P_{=part}} FIN \Vdash_{part} \mathcal{N}(T)$ , i.e.  $\vdash_{P_{=part}} S_i \Vdash_{part} \mathcal{N}(T)$ .

(2-3) Case 3:  $S_i$  is a supply constant.

According to that  $S$  is guarded, we get that  $S_i$  is a guarded supply constant. Let  $S_i = D$ . Here,  $D = S_{00}$ . The condition “ $S_i \models_{part} \mathcal{N}(T)$ ” means  $D \models_{part} \mathcal{N}(T)$ . Then,  $S_{00} \models_{part} \mathcal{N}(T)$ . Suppose that “ $S_{00} \xrightarrow{\gamma'_0} S'_0$ ” is any transition of  $S_{00}$ . By operational semantics of supply constants, we could get  $D \xrightarrow{\gamma'_0} S'_0$ , i.e.  $S_i \xrightarrow{\gamma'_0} S'_0$ . According to  $S_i \models_{part} \mathcal{N}(T)$ , there exists a transition  $\mathcal{N}(T) \xrightarrow{\rho'_0} T'_0$  such that  $\rho'_0 \trianglelefteq_{\mathcal{N}(T)} \gamma'_0$  and  $S'_0 \models_{part} T'_0$ .

(2-3-a) If  $FIN \in \{T_j | j=1, \dots, n\}$ , then for the simplicity we assume  $T_1 = FIN$  and  $T_j = \rho_j : T_{j1}$ ,  $j=2, \dots, n$ .

If  $\rho'_0 = \emptyset$ , then  $\emptyset \trianglelefteq_{\mathcal{N}(T)} \gamma'_0$ . To be simple, let  $E = \left\{ j \in \mathbb{N} : 2 \leq j \leq n \mid \mathcal{N}(T) \xrightarrow{\rho_j} \text{ and } \rho_j \trianglelefteq_{\mathcal{N}(T)} \gamma'_0 \right\} = \{2, \dots, v\}$ . Here,  $v \leq n$ . Then,  $\trianglelefteq \uparrow \sum_{j=1}^{l+1} T_j (\gamma'_0)$ ,  $l=1, \dots, n-1$ . By Rule  $R_1$ ,  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \Vdash_{part} FIN$ , i.e.  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \Vdash_{part} T_1$ . After using Rule  $R_3$  for  $n-1$  times, we could get  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \Vdash_{part} \sum_{j=1}^n T_j$ , i.e.  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \Vdash_{part} \mathcal{N}(T)$ .

If  $\rho'_0 \neq \emptyset$ , then  $\rho'_0 : T'_0 \in \{\rho_j : T_{j1} | j=2, \dots, n\}$ . For being simple, let  $\rho'_0 : T'_0 = \rho_2 : T_{21}$ ,  $F = \left\{ j \in \mathbb{N} : 3 \leq j \leq n \mid \mathcal{N}(T) \xrightarrow{\rho_j} \text{ and } \rho_j \trianglelefteq_{\mathcal{N}(T)} \gamma'_0 \right\} = \{3, \dots, w\}$ . Here,  $w \leq n$ . Then,  $\rho_2 \trianglelefteq_{\mathcal{N}(T)} \gamma'_0$ ,  $S'_0 \models_{part} T_{21}$ ,  $\trianglelefteq \uparrow \sum_{j=2}^{l+1} T_j (\gamma'_0)$ ,  $l=2, \dots, n-1$ , and  $\trianglelefteq \uparrow \sum_{j=2}^n T_j (\gamma'_0)$ . Because the transition “ $\mathcal{N}(T) \xrightarrow{\rho'_0} T'_0$ ” means  $\mathcal{N}(T) \xrightarrow{\rho_2} T_{21}$ . Then, by the definition of transition relation “ $\rightarrow$ ” we could get  $\mathcal{N}(T) \xrightarrow{\rho_2} T_{21}$ . By Proposition 3, there exists a transition  $T \xrightarrow{\rho_2} T'_{21}$  with  $\mathcal{N}(T'_{21}) = T_{21}$ . Here, we assert that  $T \xrightarrow{\rho_2} T'_{21}$ . Otherwise, there exists  $T \xrightarrow{\rho'_2}$  with  $\rho_2 < \rho'_2$ . By Proposition 3,  $\mathcal{N}(T) \xrightarrow{\rho'_2}$  with  $\rho_2 < \rho'_2$ . It

contradicts with  $\mathcal{N}(T) \xrightarrow{\rho_2}$ . By the transition  $T \xrightarrow{\rho_2} T'_{21}$  with  $\mathcal{N}(T'_{21}) = T_{21}$ , we know  $D(T'_{21}) < D(T)$ . Moreover,  $D(T'_{21}) \leq k$ . By the hypothesis, we could get  $\vdash_{P_{=part}} S'_0 \parallel_{part} T_{21}$ . By Rule  $R_2$ ,  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \parallel_{part} \rho_2 : T_{21}$ , i.e.  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \parallel_{part} T_2$ . After using Rule  $R_3$  for  $n-1$  times, we could get  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \parallel_{part} \sum_{j=2}^n T_j + T_1$ . According to Rule  $R_7$ , we have  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \parallel_{part} \sum_{j=1}^n T_j$ , i.e.  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \parallel_{part} \mathcal{N}(T)$ .

(2-3-b) If  $FIN \notin \{T_j | j=1, \dots, n\}$ , then let  $T_j = \rho_j : T_{j1}$ ,  $j=1, \dots, n$ . Thereby,  $\rho'_0 : T'_0 \in \{\rho_j : T_{j1} | j=1, \dots, n\}$ . For being simple, assume  $\rho'_0 : T'_0 = \rho_1 : T_{11}$ ,  $G = \left\{ j \in \mathbb{N} : 2 \leq j \leq n \mid \mathcal{N}(T) \xrightarrow{\rho_j} \text{ and } \rho_j \trianglelefteq_{\mathcal{N}(T)} \gamma'_0 \right\} = \{2, \dots, x\}$ . Here,  $x \leq n$ . Then,  $\rho_1 \trianglelefteq_{\mathcal{N}(T)} \gamma'_0$ ,  $S'_0 \parallel_{part} T_{11}$ ,  $\trianglelefteq \uparrow \sum_{j=1}^{x+1} T_j (\gamma'_0)$ ,  $l=1, \dots, n-1$ . Because the transition " $\mathcal{N}(T) \xrightarrow{\rho'_0} T'_0$ " means  $\mathcal{N}(T) \xrightarrow{\rho_1} T_{11}$ . By the definition of transition relation " $\rightarrow$ ", we know  $\mathcal{N}(T) \xrightarrow{\rho_1} T_{11}$ . By Proposition 3, there exists such a transition  $T \xrightarrow{\rho_1} T''_{11}$  that  $\mathcal{N}(T''_{11}) = T_{11}$ . We assert that  $T \xrightarrow{\rho_1} T''_{11}$ . Otherwise, there exists  $T \xrightarrow{\rho'_1}$  with  $\rho_1 < \rho'_1$ . By Proposition 3,  $\mathcal{N}(T) \xrightarrow{\rho'_1}$  with  $\rho_1 < \rho'_1$ . It contradicts with  $\mathcal{N}(T) \xrightarrow{\rho_1}$ . By  $T \xrightarrow{\rho_1} T''_{11}$ , we know  $D(T''_{11}) < D(T)$ . Furthermore,  $D(T''_{11}) \leq k$ . By the hypothesis, we could get  $\vdash_{P_{=part}} S'_0 \parallel_{part} T_{11}$ . By using Rule  $R_2$ ,  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \parallel_{part} \rho_1 : T_{11}$ , i.e.  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \parallel_{part} T_1$ . After using Rule  $R_3$  for  $n-1$  times,  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \parallel_{part} \sum_{j=1}^n T_j$ , i.e.  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \parallel_{part} \mathcal{N}(T)$ .

According to the above analyses in (2-3-a) and (2-3-b), for  $S_{00}$ 's any transition  $S_{00} \xrightarrow{\gamma'_0} S'_0$  we have  $\vdash_{P_{=part}} \gamma'_0 : S'_0 \parallel_{part} \mathcal{N}(T)$ . By Rule  $R_4$  and  $D \stackrel{\text{def}}{=} S_{00}$ , we could get  $\vdash_{P_{=part}} D \parallel_{part} \mathcal{N}(T)$ , i.e.  $\vdash_{P_{=part}} S_i \parallel_{part} \mathcal{N}(T)$ .

Combining the analysis results in Case 1, Case 2 and Case 3, we have  $\vdash_{P_{=part}} S_i \parallel_{part} \mathcal{N}(T)$ ,  $i=1, \dots, m$ . Then, by using Rule  $R_6$  for  $m-1$  times we could get  $\vdash_{P_{=part}} \sum_{i=1}^m S_i \parallel_{part} \mathcal{N}(T)$ , i.e.  $\vdash_{P_{=part}} S \parallel_{part} \mathcal{N}(T)$ . Thus, the original proposition holds for the case  $D(T) = k+1$ .

By the above induction analyses in (1) and (2), the proof of original conclusion is completed. ■

Theorem 3 and Theorem 4 tell us that the largest partial supply simulation relation between tasks with normal forms and guarded supplies is the relation  $\parallel_{part}$  derived in the system  $P_{=part}$ . Thus, task  $T$  is partially schedulable by supply  $S$ , which could be converted into proving " $\vdash_{P_{=part}} S \parallel_{part} T$ " in  $P_{=part}$ .

## 7. Conclusions

This paper analyzed unschedulable cases based on the definition of supply simulation relation. We got a fact that unschedulable tasks may finish on time. Then, we proposed a partial supply simulation relation and partial schedulability to describe the fact. Some useful properties about the partial supply simulation relation were gotten. A proof system was established to help reasoning about the existence of the relation. Furthermore, our proof system is proved to be sound and complete. The proof system tells us how to analyze task's partial schedulability. The above results in this paper are helpful to better understand the schedulability. Next, we plan to discuss partial schedulability from the viewpoint of probability and explore hierarchies of partially schedulable tasks.

## Acknowledgements

In the review of this paper, the referees gave us good advices. We are grateful. And we would like to acknowledge supports from National Science Foundation of China (No. 61370100 and 61673258), the Funding Project of Local Capacity Construction of Shanghai Science and Technology Commission (No. 15590501300), the Training Funding Program of Young Teachers in Universities of Shanghai (No. ZZGCD15001), the Anhui Provincial Natural Science Foundation (No. 1708085MF159), and the Natural Science Foundation of the Anhui Higher Education Institutions (No. KJ2017A375).

## References

1. M. Åsberg, P. Pettersson, and T. Nolte, "Modeling, Verification and Synthesis of Two-Tier Hierarchical Fixed-Priority Preemptive Scheduling," in *Proceedings of the 23rd Euromicro Conference on Real-Time Systems*, pp. 172-184, Porto, Portugal, July 2011
2. H. Ben-Abdallah, J.-Y. Choi, D. Clarke, Y. S. Kim, I. Lee, and H.-L. Xie, "A Process Algebraic Approach to the Schedulability Analysis of Real-Time Systems," *Real-Time Systems*, vol. 15, no. 3, pp. 189-219, November 1998
3. G. Bucci, A. Fedeli, L. Sassoli, and E. Vicario, "Modeling Flexible Real Time Systems with Preemptive Time Petri Nets," in *Proceedings of the 15th Euromicro Conference on Real-Time Systems*, pp. 279-286, Porto, Portugal, July 2003
4. A. Easwaran, I. Lee, and O. Sokolsky, "Interface Algebra for Analysis of Hierarchical Real-Time Systems," in *Proceedings of the 2nd International Workshop on Foundations of Interface Technologies (FIT)*, Budapest, Hungary, April 2008
5. X. Feng and A. Mok, "A Model of Hierarchical Real-Time Virtual Resources," in *Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS)*, pp. 26-35, Austin, Texas, USA, December 2002
6. E. Fersman, P. Krcál, P. Pettersson, and Y. Wang, "Task Automata: Schedulability, Decidability and Undecidability," *Information and Computation*, vol. 205, no. 8, pp. 1149-1172, August 2007
7. E. Fersman, P. Pettersson, and Y. Wang, "Timed Automata with Asynchronous Processes: Schedulability and Decidability," in *Proceedings of the 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pp. 67-82, Grenoble, France, April 2002
8. T. A. Henzinger and S. Matic, "An Interface Algebra for Real-Time Components," in *Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pp.253-266, San Jose, California, United States, April 2006
9. I. Lee, P. Brémont-Grégoire, and R. Gerber, "A Process Algebraic Approach to the Specification and Analysis of Resource-Bound Real-Time Systems," *Proceedings of the IEEE*, vol. 82, no. 1, pp. 158-171, 1994
10. I. Lee, A. Philippou, and O. Sokolsky, "Resources in Process Algebra," *Journal of Logic and Algebraic Programming*, vol. 72, no. 1, pp. 98-122, January 2007
11. M. Mousavi, M. Reniers, T. Basten, and M. Chaudron, "PARS: A Process Algebra with Resources and Schedulers," in *Proceedings of the 1st International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, pp.134-150, Marseille, France, September 2003
12. M. Núñez and I. Rodríguez, "PAMR: A Process Algebra for the Management of Resources in Concurrent Systems," in *Proceedings of the 21st International Conference on Formal Techniques for Networked and Distributed Systems (FORTE)*, pp.169-184, Cheju Island, Korea, August 2001
13. J. Park, I. Lee, O. Sokolsky, D. Y. Hwang, S. Ahn, J. - Y. Choi, and I. Kang, "A Process Algebraic Approach to the Schedulability Analysis and Workload Abstraction of Hierarchical Real-Time Systems," *Journal of Logical and Algebraic Methods in Programming*, vol. 92, pp. 1-18, November 2017
14. A. Philippou, I. Lee, and O. Sokolsky, "PADS: An Approach to Modeling Resource Demand and Supply for the Formal Analysis of Hierarchical Scheduling," *Theoretical Computer Science*, vol. 413, no. 1, pp. 2-20, January 2012
15. I. Shin and I. Lee, "Compositional Real-Time Scheduling Framework," in *Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS)*, pp. 57-67, Lisbon, Portugal, December 2004
16. L. Thiele, E. Wandeler, and N. Stoimenov, "Real-Time Interfaces for Composing Real-Time Systems," in *Proceedings of the 6th ACM & IEEE International Conference on Embedded Software (EMSOFT)*, pp. 34-43, Seoul, Korea, October 2006
17. X. Yao and Y. Chen, "A Proof System in Process Algebra for Demand and Supply," in *Proceedings of the 8th International Conference on Software Security and Reliability Companion (SERE-C)*, pp. 228-236. San Francisco, California, USA, June 30-July 2, 2014
18. X. Yao, M. Zhang, and Y. Chen, "A Proof System in PADS," in *Proceedings of the 10th International Colloquium on Theoretical Aspects of Computing (ICTAC)*, pp. 391-408, Shanghai, China, September 2013