References

[1] T. Agerwala. A complete model for representing the coordination of asynchronous processes. Computer Research Report 32, Johns Hopkins University, Computer Science Program, July 1974.

[2] H. J. Genrich and P. S. Thiagarajan. A theory of bipolar synchronization schemes. *Theoretical Computer Science*, 30:241–318, 1984.

[3] S. A. Greibach. Remarks on blind and partially blind one-way multicounter machines. *Theoretical Computer Science*, 7:311–324, 1978.

[4] K. Jensen. Coloured Petri nets and the invariant method. *Theoretical Computer Science*, 14:317–336, 1981.

[5] James L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, Inc., 1981.

[6] Wolfgang Reisig. *Petri Nets: An Introduction*. Springer-Verlag, 1985.

two net constructs to be identical. Let $M(s)$ be the token count for color 1, and let $D(s)$ be the token count for color 2; let the 2-color net in the translation have an immediate annihilation policy between tokens of color 1 and color 2. The debit net immediately annihilates tokens and antitokens as defined. Set the initial marking for s in each subnet to the same pair, say (n, m). By the instantaneous annihilation policy it is immediately adjusted so that $(M(s), D(s))$ becomes $(n \dot{-} m, m \dot{-} n)$.

For this proof, we again have the convenience that the transition sets of `A` and `B` are isomorphic. Thus it suffices to show that the same preconditions and postconditions hold for the firing of each transition.

case 1: firing t_1
The preconditions in each net are the same. With respect to s we have that $M_i(s) > 0$ must hold. The postcondition for s in each case is

$$(M_f(s), D_f(s)) = (M_i(s) - 1, D_i(s)).$$

After annihilation adjustment they will be the same.

case 2: firing t_2
The preconditions are equivalent with $\bullet t_2^{\mathrm{A}} - \rhd t_2^{\mathrm{A}} = \bullet t_2^{\mathrm{B}}$. We have $\rhd t_2^{\mathrm{A}} = \{s\}$, a debit arc, representing an always true condition. The postcondition for s in each subnet is

$$M_f(s) = (M_i(s), D_i(s) + 1).$$

In subnet `A` this is by definition of the debit arc, and in subnet `B` it is due to the arc $t_2 \rightarrow s$ producing a token of color 2.

Subnet `A` in Figure 5 universally represents the situational use of a debit arc. We need not consider here self-loops involving a debit arc, i.e., $s \in t\bullet \wedge t \in s \rhd$. Such a structure is semantically void and can simply be removed. By giving a universal reduction of debit nets under an instantaneous annihilation policy to colored nets with an annihilation policy, we have shown the class of such colored nets to be TM-equivalent. □

5 Discussion

We have presented some very initial results for a new P/T net extension we call debit arcs. Our analysis has focused on the expressive power of such nets, and we have shown that different expressive powers are achieved given different annihilation policies. Still, there remains much more to investigate. We would like to study analysis aspects of debit nets, such as *liveness*, *boundness*, and *safeness*, and how these differ with respect to ordinary nets and the standard extensions.

We also want to explore variations on the debit arc extension. One possible extension is debit nets in which debts are not required to stay at the places in which they were created. Instead, antitoken movement rules could be devised to allow transferral of debt around the net. Useful antitoken movement rules can be defined for either annihilation policy discussed in this paper. One simple rule says that, in addition to other valid firings, a transition t that has one or more *antitokens* in each place in $\bullet t$ may fire and create an *antitoken* in each place in $t\bullet$. We conjecture that such a debt transfer rule does not change the recognition power of debit nets under either of the annihilation rules discussed. These ideas constitute good research directions for further development of the properties of this model.

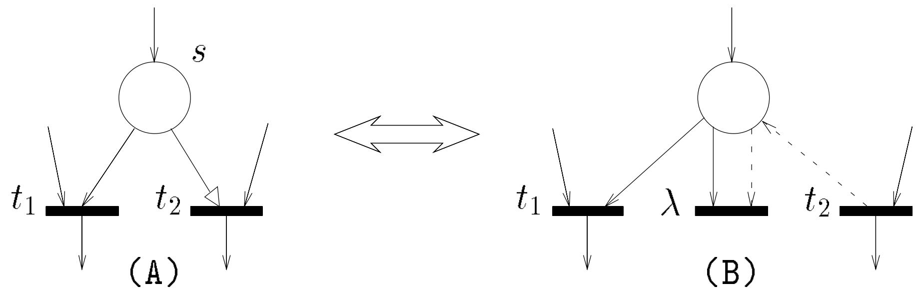

Figure 5: Representing debit arcs in two-color nets.

The preconditions are thus logically equivalent. (We need not consider the case of $M(s) < 0$ as this never occurs.)

To show that INV holds: note that in firing t_3 place s is not involved, so $M(s)$ cannot change. For the invariant to hold, $M(\bar{s})$ cannot change either. On firing t_3 a token is spent and a token is gained at $\bar{s}$, resulting in no net change.

The subnet `A` in Figure 4 universally represents the situational use of an inhibitor arc. Our proof currently holds as we may restrict the discussion to only *pure* nets, which contain no *self-loops* (in particular, self-loops containing an inhibitor arc). Even so, a transformation for this special case is easy to derive. □

4 Relation of debit nets to colored nets

To discuss debit nets as a case of colored nets, we revert to the more general pair notation to represent a debit net state. Likewise, we can consider the state of a 2-color net as a vector of pairs (c_1, c_2) where c_1 is the number of tokens of color 1 at a place and c_2 is the number of tokens of color 2. Conceptually, we can think of color 1 as being equivalent to tokens in the debit net and color 2 as antitokens.

Debit nets under delayed annihilation are equivalent to the standard class of P/T-nets. Token colors in nets have as well been shown to be a notational convenience that do not increase net power [4]. We can view debit nets as a special notation for a class of 2-color nets. This is easily seen by *folding* together places s and $\bar{s}$ from the normal net `B` in Figure 2. We use two colors to distinguish the tokens of s and $\bar{s}$, representing tokens and antitokens respectively.

The equivalence is shown by Figure 5. In the colored net `B` the transition λ still simulates delayed annihilation. The same reasoning used earlier shows these subnets to have the same behavior.

Conversely, we can take the fact that the class of debit nets under the instantaneous annihilation rule is TM-equivalent to prove the following interesting conjecture about colored nets and annihilation:

Theorem 3 The class of colored Petri nets using an instantaneous annihilation rule between any two given colors is TM-equivalent.

Proof:
We show this by providing a universal transformation from Petri nets with debit arcs under instantaneous annihilation (which are TM-equivalent) into a 2-color Petri nets employing an instantaneous annihilation rule between its two colors.

The transformation is represented as in Figure 5, minus the λ transition in `B`. Using the formalism for instantaneous annihilation introduced earlier in Section 3.2, we now show the behavior of these

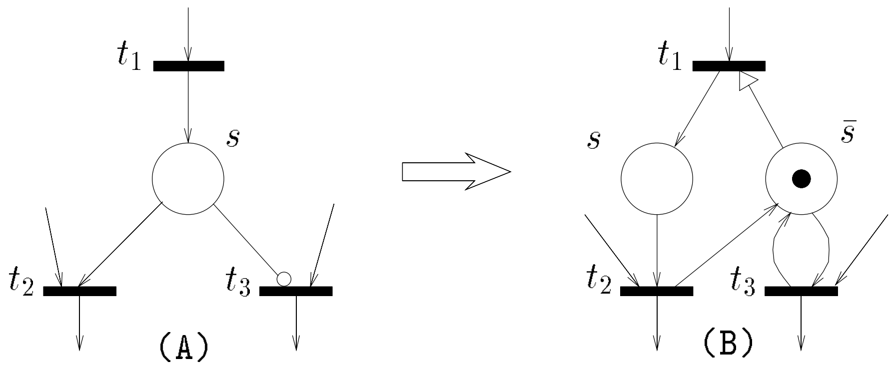

Figure 4: Inhibitor arc as debit net with instantaneous annihilation

where $M(\bar{s})$ will be a negative integer if place s contains more than one token. Without loss of generality, in Figure 4 initially $M(s) = 0$, so initially $M(\bar{s}) = 1$.

We now show that nets A and B have equivalent behaviors. Since their transition sets are isomorphic, it is sufficient to show that the same preconditions and postconditions hold for the firing of each transition.

case 1: firing t_1
The preconditions for firing t_1 are the same with $\bullet t_1^{\mathrm{A}} = \bullet t_1^{\mathrm{B}} - \{\bar{s}\}$. It is the case that t_1^{B} has one more incident arc, the debit arc from $\bar{s}$. However, by definition, this represents an always true condition; it can never prevent t_1 from firing. The postconditions for each with respect to s are the same: $M_f(s) = M_i(s) + 1$. Of course $M_f(\bar{s}) = M_i(\bar{s}) - 1$ even when $M_i(\bar{s}) \leq 0$ by definition.

We now show that INV holds:

$$M_i(s) + M_i(\bar{s}) = 1 \ \wedge \ M_i(s) = M_f(s) - 1 \ \wedge \ M_i(\bar{s}) = M_f(\bar{s}) + 1 \quad \Longrightarrow$$

$$(M_f(s) - 1) + (M_f(\bar{s}) + 1) = 1 \quad \Longrightarrow \quad M_f(s) + M_f(\bar{s}) = 1$$

case 2: firing t_2
The preconditions are identical, $\bullet t_2^{\mathrm{A}} = \bullet t_2^{\mathrm{B}}$, so $M_i(s) > 0$ must hold. The postconditions are the same, $M_f(s) = M_i(s) - 1$, with, additionally, $M_f(\bar{s}) = M_i(\bar{s}) + 1$ in subnet B.

To show that INV holds:

$$M_i(s) + M_i(\bar{s}) = 1 \ \wedge \ M_i(s) = M_f(s) + 1 \ \wedge \ M_i(\bar{s}) = M_f(\bar{s}) - 1 \quad \Longrightarrow$$

$$(M_f(s) + 1) + (M_f(\bar{s}) - 1) = 1 \quad \Longrightarrow \quad M_f(s) + M_f(\bar{s}) = 1$$

case 3: firing t_3
Note that $\bullet t_3^{\mathrm{A}} - \{s\} = \bullet t_3^{\mathrm{B}} - \{\bar{s}\}$. In A the precondition from s for firing t_3 is $M(s) = 0$ as dictated by the inhibitor arc. The precondition from $\bar{s}$ in B is $M(\bar{s}) > 0$. But note that the following holds:

$$M(s) = 0 \wedge INV \quad \Longrightarrow \quad M(\bar{s}) > 0 \quad \textit{and}$$

$$M(s) > 0 \wedge INV \quad \Longrightarrow \quad M(\bar{s}) \leq 0$$

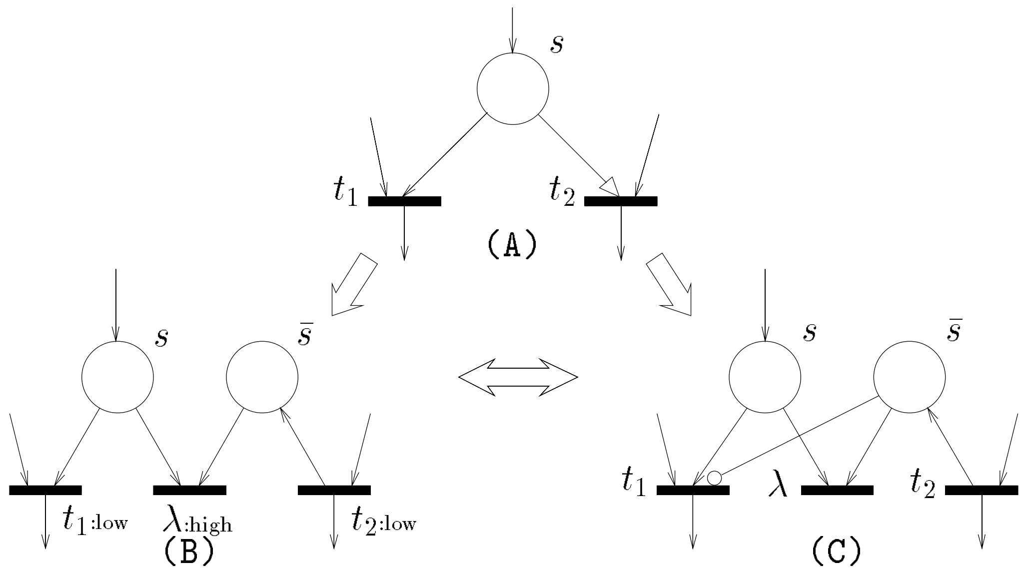

Figure 3: Modeling (A) debit arcs with (B) priorities or (C) inhibitor arcs.

Theorem 2 The class of debit nets under the instantaneous annihilation policy is equivalent to the class of Turing machines (TM-equivalent).

Proof:
We show this by providing a universal transformation from P/T-nets with inhibitor arcs into P/T-nets with debit arcs. This transformation is represented in Figure 4. We prove the behavior of these two net constructs to be identical. Once this is established, we will have demonstrated a clear method for rewriting any Petri net using inhibitor arcs into one using debit arcs. Thus the class of debit nets represents a superset of the class of Petri nets with inhibitor arcs. The class of Petri nets with inhibitor arcs is also known to be TM-equivalent [1]. These facts together with Church's Thesis are sufficient to show that the class of Petri nets with debit arcs is TM-equivalent.

The reasoning for converting a subnet using an inhibitor arc into one using debit arcs is as follows. The place s in net `A` is represented conceptually by a pair of places in net `B`, s and $\bar{s}$. Each time a token enters s, we record that fact by placing a debt in $\bar{s}$. To accomplish this, for each $t \in \bullet s$ we add a debit arc from $\bar{s}$ to t. We represent this simply by transition t_1 without loss of generality. Note that in translating a single inhibitor arc, we require a number of debit arcs equal to $|\bullet s|$.

We also want to record every time s loses a token. Let $s\circ$ be the set of transitions to which s sends an inhibitor arc. We record this by adding an arc from every $t \in s\bullet - s\circ$ to $\bar{s}$. [4] For every $t \in s\circ$ we create arcs $\bar{s} \rightarrow t$ and $t \rightarrow \bar{s}$ as a locking mechanism. This shall replicate the inhibiting function of the inhibitor arc.

We make the following stipulation for the initial marking. Place s should have the same token count in subnet `B` as it has in `A`. Place $\bar{s}$ should be marked so that

$$M(s) + M(\bar{s}) = 1 \qquad (INV)$$

[4] The set $s\circ$ denotes the set of transitions to which s issues an inhibitor arc.

net B it must be the case that $M(\bar{s}) > 0$, and so transition λ will be enabled in B whenever an annihilation is possible in A. In the other direction, whenever λ fires in B (having no effect on the generated string) it is simulated by an annihilation in A. □

3.2 Under instantaneous annihilation

We now consider debit nets under the instantaneous annihilation policy, which means that tokens and antitokens cannot coexist in the same place; whenever a token and antitoken meet, they annihilate. This implies that any place s can only have markings of the form $(n, 0)$ or $(0, n)$ for $n \in \{0, 1, 2 \ldots\}$. We can formalize this annihilation rule by mandating that whenever a state change occurs from a transition firing, the marking (n, m) at place s is immediately adjusted to $(n \dot{-} m, m \dot{-} n)$, where '$x \dot{-} y$' is defined as

$$x \dot{-} y = \begin{cases} x - y, & \text{if } x - y \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

However, since always either $M(s)$ or $D(s)$ is 0, we can dispense with the more general pair notation when employing the instantaneous annihilation rule. It is clearer to represent the marking at each place by a single integer. We may then refer to the state at place s simply as $M(s)$, where $M(s)$ may take on negative values. In the following discussion, let M_i denote the net state immediately before the firing of a given transition and M_f denote the net state immediately after. A debit arc now has very simple semantics. If $s \in \rhd t$ and t is fired, the condition

$$M_f(s) = M_i(s) - 1$$

must hold (assuming $t \notin \bullet s$). Of course, if $s \in \bullet t$ for some t, and $M(s) \leq 0$, then t is not enabled.

In debit nets with instantaneous annihilation, token counts can go negative at designated places. Debit arcs are thus a means by which an event can be allowed to proceed even if all its enabling conditions are not (yet) met; doing so records a debt that is canceled when the enabling condition next holds. At first it is not readily obvious that this increases a net's computational power. For instance, it is not trivial to show how a zero test can be performed in a debit net.

We have previously shown that the class of debit nets under the delayed annihilation rule is, indeed, no more powerful computationally than the class of normal P/T-nets. Thus far, debit arcs have only proven to be a notational convenience. The concept of allowing debts in places alone does nothing to increase the computational power of P/T-nets either. For example, we could allow one to initially mark places with antitokens. This can easily be shown to provide no increase in recognition power. If we adopt an instantaneous annihilation policy, then the initial debts must be paid at a place before it can hold tokens and subsequently enable transitions. This too can easily be shown to be equivalent to the standard P/T-net class. Intuitively, this is because there exists no way to introduce debt dynamically. The only debt is that which exists in the initial marking.

Once we allow a net to have debts created during execution, though, and we adopt the instantaneous annihilation policy, the situation changes. Figure 3 shows a debit subnet A expressed in terms of an equivalent subnet B employing priorities on transitions, and another equivalent subnet C employing inhibitor arcs.

In subnets B and C, net A's place s is represented by two places, s and $\bar{s}$. Place $\bar{s}$ can be thought of as harboring the "antitokens". Since we want tokens and antitokens to annihilate, we add a transition λ to act as a sink. To accomplish *immediate annihilation*, in B we give the added λ transitions a higher priority than the original transitions in the net. Recall that we are simulating a debit net *without* priorities by a regular net *with* priorities. It is sufficient to have two priority classes and to put t_1 and t_2 in the same class, *low*. It is only necessary that a partial order is specified in which $\lambda > t_1$ and $\lambda > t_2$ hold. In C we have $\bar{s}$ inhibit all other transitions until its "antitokens" are all gone. [3]

Intuitively, then, it would seem that instantaneous annihilation does increase the modeling power of debit nets. The following theorem establishes this fact more formally.

[3] Note that if we remove the priorities in net B and the inhibitor arc in C in Figure 3, then the nets have behavior equivalent to the debit net in A under the delayed annihilation policy. We no longer force tokens and "antitokens" to be canceled out before proceeding.

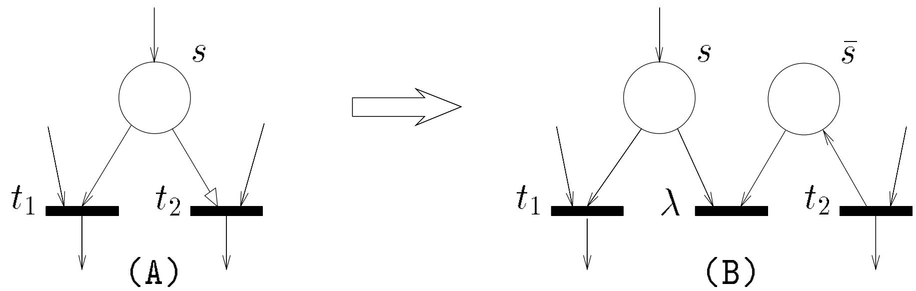

Figure 2: Debit net represented as normal P/T-net.

3.1 Under delayed annihilation

Under the delayed annihilation policy, debit nets offer notational convenience but do not extend the recognition power of P/T-nets.

Theorem 1 The class of debit nets under the delayed annihilation policy is equivalent to the class of normal P/T-nets.

Proof: by double subset.

First, it is trivially the case that all P/T-nets are debit nets (with no debit arcs).

Secondly, we show the other direction with a translation to convert any debit net under delayed annihilation into a normal P/T-net that generates the same language. An example of this translation is shown in Figure 2. In net `B` the firing of t_2 is not constrained by s (but is still constrained by any other of its input places). Firing t_2 records a "debt" in a new place $\bar{s}$. A new transition λ is included to allow cancellation of "debt" tokens in $\bar{s}$ with "regular" tokens in s. In `B` we map a null symbol to λ, and map the same symbols to t_1 and t_2 as in net `A`. Then the language generated by net `A` is the same as the language generated by net `B`, as follows. In net `A` three different events are possible: firing t_1; firing t_2 to make a debt; and annihilation. The event of firing t_2 to consume a token in s is no different from firing t_2 to make a debt and then immediately annihilating the antitoken with the token in s. We now consider the effect of each possible event on the generated string.

case 1: firing t_1
In net `A`, t_1 can fire whenever $M(s) > 0$ and all of its other input places are marked; in net `B`, the preconditions of this event are identical. Thus firing t_1^{A} is simulated by firing t_1^{B}, and *vice versa*. In both cases the same symbol is added to the generated string.

case 2: firing t_2 to make a debt
In net `A`, t_2 can fire whenever $\bullet t_2 - \{s\}$ is marked; in net `B` the same conditions hold (trivially, since $s \notin \bullet t_2$). Thus firing t_2^{A} to make a debt is simulated by firing t_2^{B}, and *vice versa*. In both cases the same symbol is added to the generated string.

case 3: annihilation of token and antitoken
In net `A`, in any state in which $M(s) > 0$ and $D(s) > 0$, an annihilation can occur, adding no symbols to the generated string. First, whenever $M(s) > 0$ in `A` it is the case that $M(s) > 0$ in `B`. Annihilation in `A` is then simulated in `B` by firing transition λ, which also adds no symbols to the generated string. In `A`, $D(s) > 0$ implies that transition t_2^{A} has previously fired; thus, in

debt, also called an *antitoken*. If s is marked, then firing t may either consume a token or create a debt. The choice of whether to create debt or not may be constrained (as described below).

In this paper we will refer to a net containing one or more debit arcs as a *debit net*. To represent debts we extend the definition of *marking* to contain both token and antitoken counts for places:

Definition 2 Marking
A *marking* of a debit net is the normal total function $M : S \rightarrow integer$ to give the token count for each place, along with another total function $D : S \rightarrow integer$ that gives the antitoken count for each place that is the source of a debit arc. [1] A *state* is then the vector of integer pairs $(M(s_i), D(s_i))$ covering all places s_i in the net.

An example of debit arc behavior is shown in Figure 1. The arc between place s_1 and transition t_1 is a debit arc, drawn as a line with an open triangle for an arrowhead. Firing t_1 creates an antitoken in s_1, drawn as an open circle. The figure shows a state change from $[(0,0)(1,0)(0,0)]$ to $[(0,1)(0,0)(1,0)]$. Note that since debit arcs act like normal arcs in the presence of tokens, the state change from $[(1,0)(1,0)(0,0)]$ to $[(0,0)(0,0)(1,0)]$ is valid for the net structure shown. We now define how antitokens can be eliminated from a debit net.

Definition 3 Annihilation
A token and an antitoken residing in the same place can *annihilate* each other. Annihilation is possible in any state in which $M(s) > 0$ and $D(s) > 0$ for some place s. The annihilation causes a state change in which the pair $(M(s), D(s))$ becomes $(M(s) - 1, D(s) - 1)$.

In the remainder of this paper, we will discuss two simple annihilation policies, each leading to different recognition power for debit nets. The first policy is termed *instantaneous annihilation*, and specifies that whenever a token and antitoken become co-resident in a place, they must immediately annihilate. The second policy is termed *delayed annihilation*, and specifies that an annihilation need not occur in the first state it is possible, but may occur in any subsequent state instead. Note that the delayed annihilation policy allows a token to come into a place containing an antitoken, and later leave that place without an annihilation occurring at all. As an example of how annihilation policies can constrain execution state sequences, refer again to the net structure shown in Figure 1, and consider the marking $[(1,1)(1,0)(0,0)]$. Under delayed annihilation, either of states $[(0,1)(0,0)(1,0)]$ or $[(0,0)(1,0)(0,0)]$ is a valid next state. Under instantaneous annihilation, only state $[(0,0)(1,0)(0,0)]$ is a valid next state.

3 Recognition power of debit nets

Several methods are common for ascribing languages to nets, differing in how symbols are associated with transitions and how final states are defined. For this paper, we assume that whenever a transition is fired, whether it thereby creates a debt or not, the symbol mapped to the transition is added to the string being generated. [2] In this case, the three methods described by Peterson [5] for mapping symbols to transitions are still applicable. However the definition of final states can be extended because of our new view of state. Now one must define a final state in terms of both tokens and antitokens in places. Perhaps the simplest (and most obvious) way to do this is to require all debts to be paid in any final state, that is, no antitokens may exist in a final state. This interpretation views a debt as a future enabling of a past event, and it requires that the enabling must actually come about for the generated string to represent a valid computation of the net (since the generated string contains a transition symbol for the event).

[1] By convention we define $D(s)$ to be a constant 0 for any s that is not the source of a debit arc. Having D be a total function is a convenience, and saves a change of notation in later extensions to these early definitions.

[2] There are other possibilities one might consider. For instance, a symbol could also be added to the string whenever a debt is finally paid (an antitoken is annihilated). What symbol is added is another criterion.

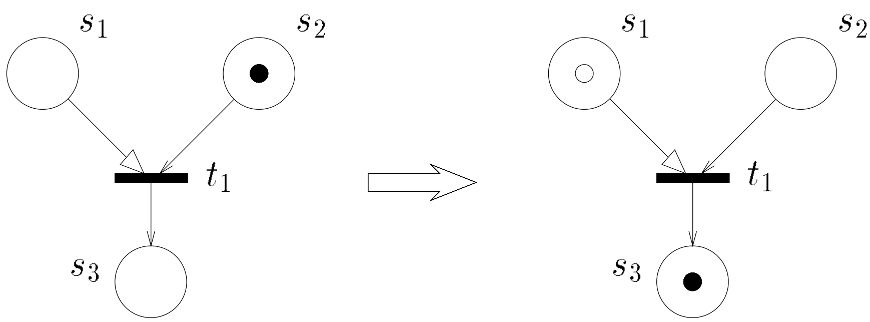

Figure 1: Firing a transition with a debit arc.

naturally, and, of course, in which nets are convenient for their direct representation of parallel activity and control flow.

An advantage of debit net notation is that the place/transition diagram and the debt cancellation policy are separate. A system may be modeled and a debt cancellation policy can be designed (or changed) as is appropriate, without altering the net structure. This is not a possibility with regular Petri nets, nor with any of the standard net extensions such as inhibitor arcs or priorities.

From a theoretical perspective, debit arcs are interesting as they represent a very intuitive, simple extension to Petri nets which can increase the language recognition power of the net. It is also insightful to understand why when the annihilation policy is changed from instantaneous to delayed, the language power reverts to that of regular nets. The debit arc extension may be viewed as more natural than the classic inhibitor arc extension as it does not encode a zero test directly (it does indirectly), while being an equally direct syntactic extension of nets.

As formal language automata, debit nets are similar to the *blind one-way multicounter machines* described by Greibach [3] in that negative token counts are allowed. While our arguments here for the *instantaneous annihilation* execution rule confirm Greibach's results, the *delayed annihilation* execution rule that we examine is not considered in Greibach's results. In addition, debit nets express the negative counter behavior of multicounter machines with P/T-net structure and execution rules, which is a form more directly useful for system modeling than the purely language-theoretic results of Greibach.

Debit nets also bear some resemblance to the *bipolar synchronization (bp)* schemes of Genrich and Thiagarajan [2] in that two token types are explicitly used in each model. In bp schemes, H-tokens represent actions taken and L-tokens explicitly represent actions *not* taken at choice points. One difference is that, for verification reasons, bp schemes are a generalization of marked graphs, a subclass of P/T-nets. Debit nets retain the full modeling generality of P/T-nets, at the cost of being no more (or less) analyzable than the general class. Another distinction is that debt tokens represent actions that *must be* taken in the future, which is not the same designation as an L-token action in bp schemes.

2 Basic definitions

We offer some basic terminology to be used in the subsequent discussion. We start with a basic place/transition net as described in Reisig [6], with unit weights and infinite capacities on all arcs. Within this environment, we define the extended form of arc called a *debit arc*, as follows:

Definition 1 Debit arc
A *debit arc* is a net arc (s,t) that can alter the normal execution behavior of a net. We define the *debit set* of t as $\bullet t \supseteq\ \triangleright t = \{s \mid (s,t)$ is a debit arc in the net $\}$. At any time, t is enabled to fire if $\bullet t -\ \triangleright t$ is marked. Firing t adds a token to each place in $t\bullet$, and removes a token from each place in $\bullet t -\ \triangleright t$. In addition, for each $s \in\ \triangleright t$, if s is not marked then firing t creates in s a

Place/Transition Nets with Debit Arcs

P. David Stotts*
Department of Computer and Information Sciences
University of Florida
Gainesville, FL 32611

Parke Godfrey †
Department of Computer Science
University of Maryland
College Park, MD 20742

Abstract

We add an extension called *debit arcs* to traditional place/transition nets. A debit arc incident upon a transition represents an always true precondition; when the transition fires, a token is subtracted from the place issuing the debit arc, creating an *antitoken* if no tokens are present to subtract. We show that two different policies on how tokens and antitokens *annihilate* produce two classes of automata with different recognition powers.

Key words: Petri nets, place/transition nets, automata theory, formal languages, parallel computation model, colored nets, high-level nets.

CR categories: F.1.1 (Petri nets)

1 Introduction

We introduce an extension called *debit arcs* to traditional place/transition nets (P/T-nets, also known as Petri nets). A debit arc is permitted from a place to a transition, and is distinguished from a regular arc. (Debit arcs are pictorially denoted by hollow arrows in this paper, as in Figure 1.) The destination transition may fire whenever, regardless of any incident debit arcs, but still obeying the standard Petri net rules for fireability. Firing the transition records a *debt*, or an *antitoken*, at each place issuing a debit arc to the transition. A normal token may annihilate with an antitoken within a place, which may be considered as "paying off" the debt. Two natural rules for this token/antitoken annihilation, (*instantaneous* and *delayed*), are examined. The two are shown to create two distinct classes of automata in terms of language recognition power.

In modeling systems, one might wish to specify an event that may proceed even when all of its preconditions are not met. The economic concept of debt, for example, can be thought of as allowing an event to occur (say, the purchase of a good) even when a precondition (having the price available in cash) does not hold. One might further stipulate that the precondition must *eventually* hold. This debit net notation will be beneficial for modeling in applications in which "debts" (events with delayed preconditions) arise

*Supported in part by the Center for Excellence in Space Data and Information Sciences at the NASA Goddard Space Flight Center.
†Supported in part by the U.S. Army Institute for Management Information and Computer Science (AIRMICS).