Liveness, Fairness, and Recurrence in Petri Nets

Ekkart Kindler*
Humboldt-Universität zu Berlin
Institut für Informatik
D-10099 Berlin
Germany

Wil van der Aalst†
Eindhoven University of Technology
Department of Mathematics and Computing Science
P.O. Box 513
5600 MB Eindhoven
The Netherlands

Keywords: Concurrency, Distributed Systems, Petri Nets, Theory of Computation.

Introduction

In Petri net theory, a transition is called *live* if from every reachable marking there starts a computation in which the transition occurs. Another important property of a transition is the recurrent occurrence of the transition in every computation. In that case, we call the transition *recurrent*. Though *liveness*[1] and *recurrence* of a transition are similar in spirit, there is a big difference: Liveness only requires the possibility of transition occurrences whereas recurrence requires transition occurrences in every computation.

Obviously, recurrence implies liveness. In this paper, we will investigate the reverse direction of this implication. In particular, we will characterize a class of Petri nets for which liveness implies recurrence. Basically, this class is *asymmetric choice nets* [7, 9] which is often also called *extended simple nets* [8, 6]. Since we extend the class with respect to loops we call the class *extended asymmetric choice nets* rather than 'extended extended simple nets'.

This result is important for the following reason: Recurrence is an important property in many application areas. Usually, recurrence is proven by temporal logic or by model checking. For extended asymmetric choice nets, it is sufficient to prove liveness by standard techniques of Petri net theory. More concretely, the result is important for the verification of business processes. In fact, our quest for the relation between liveness and recurrence started with the question whether the *soundness* of a workflow process [1] guarantees its proper termination. The results reported in this paper say that, for extended asymmetric choice nets, soundness does guarantee termination. Moreover, detailed analysis of workflow processes has shown that all workflows encountered can be modelled by extended asymmetric choice nets.

1 Some examples

Before a formal presentation of our results, let us give some examples which illustrate the basic concepts and the basic result. Figure 1 shows a simple system net Σ_1 which is live; i.e. from each reachable marking, each transition can somehow be enabled. However, there is an infinite computation

*email: kindler@informatik.hu-berlin.de
†email: wsinwa@win.tue.nl

[1] We use liveness in the Petri net sense throughout this paper. This concept is not equivalent to the concept of liveness introduced by Lamport [11, 5]. Actually, recurrence is a liveness property in Lamport's terminology.

in which transition d does never occur:

$$[A] \xrightarrow{a} [B] \xrightarrow{b} [C] \xrightarrow{c} [A] \xrightarrow{a} \ldots$$

In this representation of a computation, a marking is represented within square brackets; the occurrence of a transition which results in another marking is indicated by arrows inscribed by the corresponding transition. The above computation shows that there are computations of a live system net in which not all transitions do occur infinitely often. In fact, d does not occur at all. So, Σ_1 is not recurrent in the above computation.

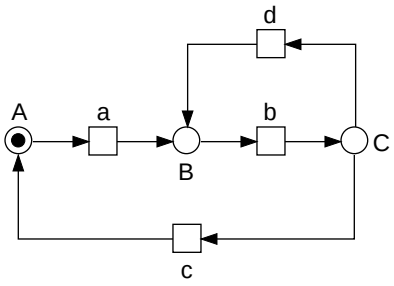

Figure 1: A live system net Σ_1

The above computation, however, might be considered to be illegal because transition d was enabled over and over again (i.e. infinitely often) but was never chosen—clearly an *unfair* behaviour. If we only consider *fair* computations of Σ_1, each transition occurs over and over again. So, we define recurrence with respect to fair computations in the formal part of this paper. There are even simpler computations of Σ_1 in which no transition is recurrent: finite computations and in particular the computation which consist of the initial marking only. We also exclude these computations from our considerations. Technically, this *progress assumption* is subsumed by the definition of fairness.

Now, we might expect that a live system net is always recurrent with respect to fair computations. This, however, is not true. Figure 2 shows a counter-example. The system net Σ_2 is live. Furthermore, there is a computation in which transition e does not occur:

$$[A, C] \xrightarrow{d} [A, D] \xrightarrow{a} [B, D] \xrightarrow{b} [A, D] \xrightarrow{c} [A, C] \xrightarrow{d} \ldots$$

Nevertheless, this computation is reasonably[2] fair because transition e is never enabled in this computation. Altogether, Σ_2 shows that liveness does not imply recurrence—even when restricted to fair computations.

[2] A reasonable fairness concept does not enforce the occurrence of a transition which is never enabled.

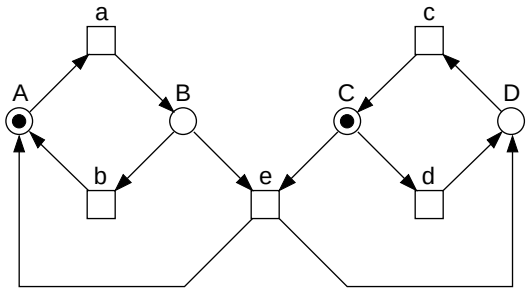

Figure 2: Another live system net Σ_2

The reason why Σ_2 is not recurrent is that transition e has *two conflict places* B and C. With respect to place B, transition e is in conflict with transition b; with respect to place C, it is in conflict with transition d. In the above computation, the transitions b and d *conspire* [6] against transition e. In this paper, we show that a live system net is recurrent if every transition has at most one conflict place. This class is often called *simple nets*. This result is proven in Sect. 3 after introducing the mathematical prerequisites in Sect. 2. In Sect. 4, we generalize this result to extended asymmetric choice nets and discuss some limitations.

2 Basic concepts

Now, we introduce and formalize the basic concepts. In Sect. 2.1, we present concepts from Petri net theory which are all well-established [13, 14, 7, 9]. In Sect. 2.2, we present computations, fair computations, and the concept of recurrence, which are not standard in Petri net theory.

2.1 Petri nets

A Petri net consists of a finite set of *places* P, a finite set of *transitions* T, and a *flow relation* F which relates transitions and places. A place is graphically represented by a circle, a transition is represented by a square, and the flow relation is represented by arrows between the corresponding elements (cf. Fig. 1 and Fig. 2).

A *marking* of a net associates a natural number

with each place. This number represents the number of *tokens* residing at that place. Graphically, a token is represented by a black dot. In a textual representation, we use multiset notation for markings, e.g. $[A, C, A]$ represents a marking with two tokens at place A, a single token at place C, and no tokens on all other places.

Definition 1 (Net, marking, system net)
Let P and T be two finite and disjoint sets and let F be a relation $F \subseteq (P \times T) \cup (T \times P)$. Then, we call $N = (P, T, F)$ a net. *A mapping $M : P \longrightarrow \mathbb{N}$ is called a* marking *of N. A net N equipped with a marking M is called a* system net *$\Sigma = (N, M)$ and M is called the* initial marking *of Σ.*

The places and transitions of a net are also called the *elements* of the net. For a given element x, the preset ${}^\bullet x$ denotes all elements which have an arc towards x; the postset $x^\bullet$ denotes all those elements which have an arc coming from x.

Definition 2 *Let $N = (P, T, F)$ be a net.*

1. *For an element $x \in P \cup T$, we define the* preset *of x by ${}^\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$. We define the* postset *of x by $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$.*

2. *A place $p \in P$ is called a* conflict place *of N if it has more than one transition in its postset; i.e. if $|p^\bullet| > 1$. A conflict place p is called a* conflict place *of a transition t if $p \in {}^\bullet t$.*

   The net N is called simple *if every transition of N has at most one conflict place.*

3. *A set $S \subseteq P$ is called a* siphon *of N if for every transition $t \in T$ with $t^\bullet \cap S \neq \emptyset$ we also have ${}^\bullet t \cap S \neq \emptyset$. A siphon S is* unmarked *at a marking M if for every $s \in S$ we have $M(s) = 0$.*

The definition of a siphon is structural. Still, there are behavioural consequences. Once unmarked, a siphon remains unmarked forever, because every transition which could produce a token on S also needs a token on S. We formalize this property in Prop. 5 after the formal definition of the behavioural concepts.

A transition is *enabled* at a marking if every place in its preset is marked. An enabled transition may *occur* in which case one token is removed from every place in the transition's preset and one token is added to every place in the transition's postset.

Definition 3 (Behaviour of nets)
Let $N = (P, T, F)$ be a net and let M be a marking of N.

1. *A transition $t \in T$* is enabled *at M if for every $p \in {}^\bullet t$ we have $M(p) \geq 1$.*

2. *If transition $t \in T$ is enabled at M, it may* occur *and its occurrence changes the marking into the* successor marking *M' defined by*

$$M'(p) = \begin{cases} M(p) & \text{if } p \notin {}^\bullet t \text{ and } p \notin t^\bullet \\ M(p) & \text{if } p \in {}^\bullet t \text{ and } p \in t^\bullet \\ M(p) - 1 & \text{if } p \in {}^\bullet t \text{ and } p \notin t^\bullet \\ M(p) + 1 & \text{if } p \notin {}^\bullet t \text{ and } p \in t^\bullet \end{cases}.$$

   Then, we write $M \xrightarrow{t} M'$.

3. *A marking M' is* reachable from *M if there exists a (possibly empty) sequence of markings $M_1, M_2, \ldots, M_{n-1}$ and a sequence of transitions $t_1, t_2, \ldots, t_n$ such that we have $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \rightarrow \ldots \xrightarrow{t_n} M'$.*

With these basic concepts, we are able to define liveness of a Petri net.

Definition 4 (Liveness)
Let $\Sigma = (N, M)$ be a system net.

1. *A transition t of Σ is* live *if for every marking M_1 which is reachable from M there exists a marking M_2 which enables t and is reachable from M_1.*

2. *Σ is* live *if every transition of Σ is live.*

Now, we come back to the already mentioned behavioural property of siphons, which are well-known in Petri net theory (e.g. [9]).

Proposition 5 *Let $\Sigma = (N, M)$ be a system net and let S be a siphon of N.*

1. *Let M_1 be a marking in which S is unmarked. Then, S is unmarked at every marking M_2 reachable from M_1.*

2. *Let M_1 be a marking which is reachable from M and at which S is unmarked. If there exists a transition t with ${}^\bullet t \cap S \neq \emptyset$, then Σ is not live.*

2.2 Computations and recurrence

A computation of a system is a finite or infinite sequence of transition occurrences. As already mentioned in the examples, we additionally impose a fairness assumption on computations. A computation in which a transition is enabled over and over again but does not occur any more from some point on, is *unfair*. This fairness requirement is usually called *strong fairness* [10, 4].

Definition 6 (Fair computation)
Let $\Sigma = (N, M)$ be a system net. A finite or infinite sequence $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_3} M_3 \xrightarrow{t_4} \ldots$ is a computation *of Σ. The computation is* fair with respect to a transition *t if either t occurs infinitely often or there exists a* position *i from which on t is never enabled again (i.e. no marking M_j of the computation with $j \geq i$ enables transition t). A computation is a* fair computation *of Σ if the computation is fair with respect to every transition of Σ.*

Note that we consider infinite as well as finite computations. By the above definition, a finite computation is only fair if no transition is enabled at its final marking. This way, fairness subsumes the progress assumption, which guarantees that a computation does not stop as long as transition occurrences are possible. Moreover, every finite computation of a system net can be extended to a (possibly infinite) fair computation of the system net. This property is called *feasibility* [4]. In particular, feasibility guarantees that every system net has fair computations.

Definition 7 (Recurrence) *Let Σ be a system net. A transition t of Σ is* recurrent *if it occurs infinitely often in every fair computation of Σ. The system net Σ is* recurrent *if every transition of Σ is recurrent.*

Obviously, recurrence implies liveness.

Proposition 8 *A recurrent system net is live.*

Proof: Let us assume that Σ is a recurrent system net. It is sufficient to show that every transition t of Σ can be enabled from every reachable marking of Σ. For every reachable marking M', there exists a finite (possibly unfair) computation

$$M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} \ldots \xrightarrow{t_n} M_n = M'$$

of Σ with $M' = M_n$. Due to feasibility of the fairness requirement, there exists an extension of this computation which is fair. Since Σ is recurrent, transition t occurs infinitely often in this fair extension. Thus, we know that t can be enabled from M'. □

In the rest of this paper, we consider the reverse direction of Prop. 8.

3 Liveness and recurrence

In this section, we prove for simple nets (i.e. for nets in which every transition has at most one conflict place) that liveness implies recurrence. This requirement is illustrated in Fig. 3.

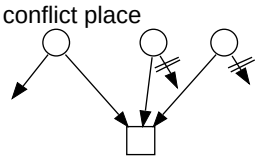

Figure 3: Illustration of the requirement

The following observation is pivotal in the proof: Let us consider a transition t with at most one conflict place and consider a fair computation in which t occurs only finitely many times. Then, there is a place p in its preset which is unmarked forever in this computation from some point on. We call this place the *scapegoat* which is responsible for t not occurring any more.

In the rest of this section, we formalize scapegoats (Def. 9), we show that simple nets have scapegoats (Lemma 10), and we prove for nets with scapegoats that liveness implies recurrence (Theorem 11).

Definition 9 (Scapegoat) *Let $\Sigma = (N, M)$ be a system net, let t be a transition of Σ, and let $\sigma = M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_3} \ldots$ be a computation. A*

place $p \in {}^\bullet t$ is called scapegoat for t in σ *if there exists a position i in σ such that p is unmarked at every marking M_j of σ with $j \geq i$.*

We say that t has scapegoats *if in each fair computation σ in which t is not recurrent there exists a scapegoat for t in σ.*

Let us consider our introductory example Σ_1 from Fig. 1, again. In the computation

$$[A] \xrightarrow{a} [B] \xrightarrow{b} [C] \xrightarrow{d} [B] \xrightarrow{b} [C] \xrightarrow{d} \ldots$$

place A is a scapegoat for transition a. Note, that there is no scapegoat for transition c; but this does not contradict our above observation because the computation is not fair with respect to c. In general, a transition with at most one conflict place has scapegoats which is proven in the following lemma.

Lemma 10 *Let Σ be a system net and let t be a transition with at most one conflict place. Then, t has scapegoats.*

Proof: Let $\sigma = M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_3} \ldots$ be a fair computation in which t is not recurrent. We will show that t has a scapegoat.

Since t is not recurrent and the computation is fair, there is a position i in σ from which on t is never enabled again. If one of the places in the preset of t except for the conflict place remains unmarked forever, then t has a scapegoat.

If each place in the preset of t except for the conflict place is marked at some position $j \geq i$, then there is a position k such that all these places are (and remain) simultaneously marked (t is the only transition that can remove tokens from these places). Since t is never enabled beyond position i, the conflict place is never marked beyond position k and is therefore a scapegoat. □

Now, it remains to show that a live system net in which every transition has scapegoats is recurrent.

Theorem 11 *Let Σ be a system net in which every transition has a scapegoat. Σ is live if and only if Σ is recurrent.*

Proof:

"$\Leftarrow$" Prop. 8

"$\Rightarrow$" Let us assume to the contrary, that Σ has scapegoats for each transition and is not recurrent. Then, we show that Σ is not live. To this end, we consider a fair computation σ in which some transition t is not recurrent. Now, we construct a siphon S with ${}^\bullet t \cap S \neq \emptyset$ which is unmarked in some marking of the computation. By Prop. 5 (2), Σ is not live.

Let us consider a suffix of σ which satisfies the following requirements:

1. No non-recurrent transition of σ occurs in the suffix.
2. For each non-recurrent transition of σ, there exists a scapegoat for t which is not marked in any marking of the suffix.

Since the net is finite, such a suffix exists.

Let S be the set of places which are unmarked in all markings of this suffix. Then, S is a siphon of Σ for the following reason: Let t' be a transition of Σ with $t'^\bullet \cap S \neq \emptyset$. Clearly, t' is not recurrent in σ. Therefore, t' has scapegoats in σ. According to the definition of the suffix, one of these scapegoats $p \in {}^\bullet t'$ is unmarked in all markings of the suffix; therefore, we have $p \in S$. Thus, we have ${}^\bullet t' \cap S \neq \emptyset$. By definition, S is unmarked in a reachable marking (e.g. in the first marking of the suffix).

□

Altogether, we have shown that for simple nets liveness implies recurrence.

4 Extensions and Limitations

Though simple nets occur in some practical applications, this class is rather restrictive. In this section, we generalize the result to extended asymmetric choice nets.

Asymmetric choice nets have been introduced as a generalization of free choice nets for which some properties of free choice nets are still valid [9]. In an asymmetric choice net, for a given transition t, the *conflict sets* with two other transitions t_1 and t_2 are included in either way: ${}^\bullet t \cap {}^\bullet t_1 \subseteq {}^\bullet t \cap {}^\bullet t_2$ or ${}^\bullet t \cap {}^\bullet t_2 \subseteq {}^\bullet t \cap {}^\bullet t_1$. Therefore, the conflict sets of

t with other transitions form an ascending chain as illustrated in Fig. 4. An equivalent definition which relates postsets of places will be given in Def. 15.

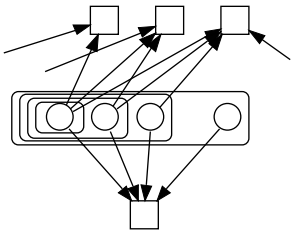

Figure 4: Asymmetric choice nets

In extended asymmetric choice nets, there may be loops which violate the ascending chain of conflict sets.

Definition 12 (Extended asymmetric choice) *Let $N = (P, T, F)$ be a net. For each transition $t \in T$ we define a relation $\rightsquigarrow_t$ on the places ${}^\bullet t$ as follows: For $p_1, p_2 \in {}^\bullet t$, we have $p_1 \rightsquigarrow_t p_2$ if and only if there exists a transition $t' \in T$ such that $p_1 \in {}^\bullet t' \setminus t'^\bullet$ and $p_2 \notin {}^\bullet t'$.*

A net is called an extended asymmetric choice (EAC) net *if for every transition $t \in T$ the relation $\rightsquigarrow_t$ is acyclic.*

In order to prove that live extended asymmetric choice nets are also recurrent, we prove that every transition of an extended asymmetric choice net has scapegoats. The rest follows by Theorem 11.

Theorem 13 (EAC nets have scapegoats) *Let Σ be a an extended asymmetric choice system net. Then, every transition of Σ has scapegoats.*

Proof: Let σ be a fair computation in which transition t has no scapegoats; we show that t is recurrent in σ. Since σ is fair, it is sufficient to prove that in σ transition t is enabled over and over again.

Since N is extended asymmetric choice, $\rightsquigarrow_t$ is acyclic. Therefore, we can arrange the places in the preset of t in a sequence $p_1\, p_2 \ldots p_n$ such that for each $j, k \in \{1, 2, \ldots, n\}$ with $j < k$ we have $p_j \not\rightsquigarrow_t p_k$. By definition of $\rightsquigarrow_t$, we know that for $j < k$ and each transition t' with $p_j \in {}^\bullet t' \setminus t'^\bullet$ we also have $p_k \in {}^\bullet t'$; i.e. every transition which removes a token from a place p_j also needs a token from all places p_k with $k > j$.

Next, we show by induction on $i = 1, 2, \ldots n$ that the set of places $\{p_1, \ldots, p_i\}$ are simultaneously marked over and over again.

$i = 1$: Since t has no scapegoats in σ, we know that $p_1 \in {}^\bullet t$ is marked in σ over and over again (otherwise p_1 is a scapegoat for t).

$i \to i + 1$: Let us assume by induction hypothesis that $\{p_1, \ldots, p_i\}$ is simultaneously marked over and over again. Since t has no scapegoats in σ, place $p_{i+1} \in {}^\bullet t$ is also marked over and over again.

The only reason for $\{p_1, \ldots, p_{i+1}\}$ not being marked simultaneously over an over again, is that some transition t' removes a token from some place p_j with $j \leq i$ before p_{i+1} is marked. In that case, we have $p_j \in {}^\bullet t' \setminus t'^\bullet$. Due to the arrangement of the places and by $i + 1 > j$ we also have $p_{i+1} \in {}^\bullet t'$. Thus, no t' can remove a token from $\{p_1, \ldots, p_i\}$ before p_{i+1} is marked. Therefore, $\{p_1, \ldots, p_{i+1}\}$ is simultaneously marked over and over again.

Thus, t is enabled over and over again in σ. □

In combination, Theorem 11 and Theorem 13 give us the following corollary.

Corollary 14 *Let Σ be an extended asymmetric choice system net. Σ is live if and only if Σ is recurrent.*

Up to now, we have only considered finite nets. The proofs of all theorems make use of the finiteness of the considered net. Indeed, Theorem 11 and Corollary 14 do not apply to infinite nets—not even to infinite free choice nets. Figure 5 shows a counter example. The system net Σ_3 is asymmetric choice and live, but it is not recurrent.

Readers familiar to the definition of asymmetric choice might observe that the definition of extended asymmetric choice nets has a quite different structure than the definition of asymmetric choice. For a naive extension of asymmetric choice, however, the property of Corollary 14 does

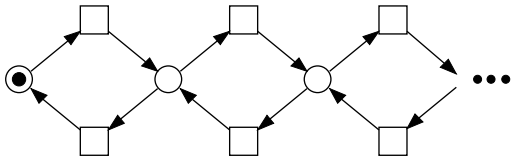

Figure 5: A live but not recurrent net Σ_3

not hold. In order to clarify this point, we compare the concepts of *asymmetric choice* (AC), *extended asymmetric choice* (EAC), and a *naive extension of asymmetric choice* (NEAC). Note that we naively extend the place-wise definition of asymmetric choice in the following. Similarly, we could also naively extend the transition-wise definition of asymmetric choice (see beginning of Sect. 4 and [2])—with the same effect.

Definition 15 *Let $N = (P, T, F)$ be a net.*

1. *Net N is an* asymmetric choice (AC) net *if for each two places $p_1, p_2 \in P$ with $p_1^\bullet \cap p_2^\bullet \neq \emptyset$ we have $p_1^\bullet \subseteq p_2^\bullet$ or $p_2^\bullet \subseteq p_1^\bullet$.*
2. *Net N is a* naively extended asymmetric choice (NEAC) net *if for each two places $p_1, p_2 \in P$ with $p_1^\bullet \cap p_2^\bullet \neq \emptyset$ we have $p_1^\bullet \setminus {}^\bullet p_1 \subseteq p_2^\bullet$ or $p_2^\bullet \setminus {}^\bullet p_2 \subseteq p_1^\bullet$.*

It can be proven [2] that AC implies EAC and that EAC implies NEAC. The reverse directions of these implications, however, do not hold. Fig. 6 shows a system net Σ_4 which is NEAC but not EAC because $\rightsquigarrow_t$ has a cycle. Furthermore, this example shows that Corollary 14 does not apply to NEAC nets: Σ_4 is live but not recurrent[3].

5 Conclusion

We have shown that for extended asymmetric choice nets liveness implies recurrence. The proof uses techniques which are similar to proof techniques used in classical free choice theory [9]. In particular, the concept of scapegoats and the relation to recurrence is analog to the concept of *place-liveness* and liveness in free choice theory. Scapegoats and recurrence are just a re-formulation of

[3] Think of the following infinite sequence of transition occurrences: $t_1\,(t_2\,t_1'\,t_3\,t_2'\,t_1\,t_3')^\omega$. The corresponding computation is fair but t does not occur.

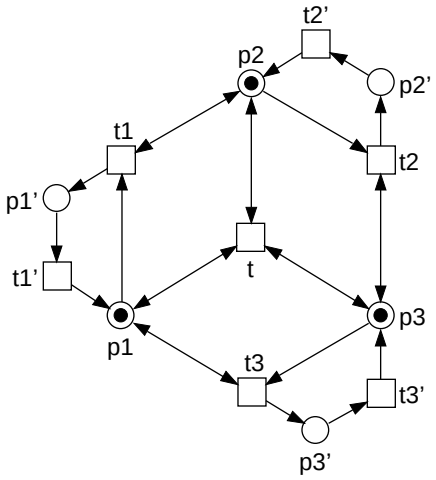

Figure 6: A live but not recurrent NEAC net Σ_4

place-liveness and liveness based on computations rather than on the reachability graph. On a more abstract level, we have converted a branching-time result from free choice theory to a linear-time result. We believe, that many other concepts and results from free choice theory can be converted from branching-time to linear-time when fairness is assumed.

Though classical free choice theory mainly deals with branching-time properties, there has been some work which is similar to our work. Thiagarajan and Voss [15] show that for free choice nets *global fairness* can be implemented by *local fairness*. Global fairness is equivalent to recurrence. Local fairness, however, is different form our concept of fairness. In particular, the result from [15] does not apply to asymmetric choice nets.

Best [6] introduces a hierarchy of fairness concepts, which are called k-fairness for each $k \in \mathbb{N}$ and ∞-fairness. Basically, a computation σ is not k-fair with respect to a transition t if there are infinitely many positions in σ from which t could be enabled within k steps, but t does not occur infinitely often in σ. So, 0-fairness is equivalent to our concept of fairness. Best shows that ∞-fairness is equivalent to k-fairness for all k. Moreover, he shows that the fairness hierarchy collapses for asymmetric choice nets. The bridge between Best's and our results is the following: We conjecture that liveness of a net is equivalent to the

requirement that all ∞-fair computations of a net are recurrent. With this conjecture, Corollary 14 follows from Best's theorems. But, this conjecture is neither mentioned in [6] nor proven in our paper. We preferred to give a direct proof of our theorems.

Best's concept of ∞-fairness has also been introduced as *hyperfairness* in [3] and [12]. We believe that hyperfairness is an important concept in the context of fault-tolerant distributed computing. Still, hyperfairness should be clearly distinguished from fairness since there are no general schedulers for hyperfairness (cf. [3]). A detailed discussion of this issue, however, is left to a forthcoming paper.

Acknowledgments We would like to thank Jörg Desel, Wolfgang Reisig, Hagen Völzer, and one anonymous referee for their helpful comments on preliminary versions of this paper. Furthermore, we gratefully acknowledge the discussions of the 'Kaffeerunde' at Humboldt University.

References

1. Wil van der Aalst. The application of Petri nets to workflow management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.
2. Wil van der Aalst, Ekkart Kindler, and Jörg Desel. Beyond asymmetric choice: A note on some extensions. *Petri Net Newsletter*, 55:3–13, October 1998.
3. Paul C. Attie, Nissim Francez, and Orna Grumberg. Fairness and hyperfairness in multi-party interactions. *Distributed Computing*, 6:245–254, 1993.
4. Krysztof R. Apt, Nissim Francez, and Shmuel Katz. Appraising fairness in languages for distributed programming. *Distributed Computing*, 2:226–241, 1988.
5. Bowen Alpern and Fred B. Schneider. Defining liveness. *Information Processing Letters*, 21:181–185, October 1985.
6. Eike Best. Fairness and conspiracies. *Information Processing Letters*, 18:215–220, 1984.
7. Eike Best. Structure theory of Petri nets: the free choice hiatus. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Central Models and Their Properties*, *LNCS* 254, pages 168–205. Springer-Verlag, 1987.
8. E. Best and M.W. Shields. Some equivalence results for free choice nets and simple nets, and on the periodicity of live free choice nets. In G. Ausiello and M. Protasi, editors, *Proceedings of CAAP '83*, *LNCS* 159, pages 141–154. Springer-Verlag, 1983.
9. Jörg Desel and Javier Esparza. *Free Choice Petri Nets*. Cambridge University Press, 1995.
10. Nissim Francez. *Fairness*. Texts and Monographs in Computer Science. Springer-Verlag, 1986.
11. Leslie Lamport. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering*, SE-3(2):125–143, March 1977.
12. Leslie Lamport. Fairness and hyperfairness. SRC Research Report 152, Digital, Systems Research Center, March 1998.
13. James L. Peterson. *Petri Net Theory And The Modeling of Systems*. Prentice-Hall, 1981.
14. Wolfgang Reisig. *Petri Nets*, *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1985.
15. P.S. Thiagarajan and K. Voss. A fresh look at free choice nets. *Information and Control*, 61(2):85–113, May 1984.