Some Remarks on Definability of Process Graphs

Clemens Grabmayer[1,⋆], Jan Willem Klop[2], and Bas Luttik[3]

[1] Department of Computer Science, Vrije Universiteit Amsterdam,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
clemens@cs.vu.nl
http://cs.vu.nl/~clemens

[2] Vrije Universiteit and CWI Amsterdam, and Radboud Universiteit Nijmegen
Postal address: Department of Computer Science, Vrije Universiteit Amsterdam,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
jwk@cs.vu.nl
http://www.cs.vu.nl/~jwk

[3] Technische Universiteit Eindhoven, and CWI Amsterdam
Postal address: Department of Mathematics and Computer Science, TU/e,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
s.p.luttik@tue.nl
http://www.win.tue.nl/~luttik

Abstract. We propose the notions of "density" and "connectivity" of infinite process graphs and investigate them in the context of the well-known process algebras BPA and BPP. For a process graph G, the density function in a state s maps a natural number n to the number of states of G with distance less or equal to n from s. The connectivity of a process graph G in a state s is a measure for how many different ways "of going from s to infinity" exist in G.

For BPA-graphs we discuss some tentative findings about the notions density and connectivity, and indicate how they can be used to establish some non-definability results, stating that certain process graphs are not BPA-graphs, and stronger, not even BPA-definable. For BPP-graphs, which are associated with processes from the class of Basic Parallel Processes (BPP), we prove that their densities are at most polynomial. And we use this fact for showing that the paradigmatic process Queue is not expressible in BPP.

1 Introduction

An important topic in process theory is the issue of expressiveness or definability. There is a family of results, especially in the context of ACP [5,3], to the effect that a particular process can or cannot be defined by a finite recursive specification using a certain set of process operations.

A typical example is the process Stack. It can be defined by a finite recursive specification over BPA (for the axioms of BPA see Table 1), but not by a

⋆ Supported by the Netherlands Organisation for Scientific Research NWO in project *GeoProc*("Geometry of Processes", Nr. 612.000.313).

C. Baier and H. Hermanns (Eds.): CONCUR 2006, LNCS 4137, pp. 16–36, 2006.

Table 1. BPA (Basic Process Algebra), left, and PA (Process Algebra), on the right

$x + y$	$= y + x$
$x + (y + z)$	$= (x + y) + z$
$x + x$	$= x$
$(x + y) \cdot z$	$= x \cdot z + y \cdot z$
$(x \cdot y) \cdot z$	$= x \cdot (y \cdot z)$

$x + y$	$= y + x$
$x + (y + z)$	$= (x + y) + z$
$x + x$	$= x$
$(x + y) \cdot z$	$= x \cdot z + y \cdot z$
$(x \cdot y) \cdot z$	$= x \cdot (y \cdot z)$
$x \parallel y$	$= x \lfloor\!\lfloor y + y \lfloor\!\lfloor x$
$\mathrm{a} \lfloor\!\lfloor x$	$= \mathrm{a} \cdot x$
$\mathrm{a} \cdot x \lfloor\!\lfloor y$	$= \mathrm{a} \cdot (x \parallel y)$
$(x + y) \lfloor\!\lfloor z$	$= x \lfloor\!\lfloor z + y \lfloor\!\lfloor z$

Table 2. Stack, an infinite linear and a finite non-linear BPA-specification

$\mathrm{S}_\lambda = 0{\cdot}\mathrm{S}_0 + 1{\cdot}\mathrm{S}_1$
$\mathrm{S}_{\mathrm{d}\sigma} = 0{\cdot}\mathrm{S}_{0\mathrm{d}\sigma} + 1{\cdot}\mathrm{S}_{1\mathrm{d}\sigma} + \underline{\mathrm{d}}{\cdot}\mathrm{S}_\sigma$
(for $d = 0$ or $d = 1$, and any string σ)

$\mathrm{S} = \mathrm{T}{\cdot}\mathrm{S}$
$\mathrm{T} = 0{\cdot}\mathrm{T}_0 + 1{\cdot}\mathrm{T}_1$
$\mathrm{T}_0 = \underline{0} + \mathrm{T}{\cdot}\mathrm{T}_0$
$\mathrm{T}_1 = \underline{1} + \mathrm{T}{\cdot}\mathrm{T}_1$

finite recursive specification over BCCSP (which has action prefixing instead of sequential composition). See the infinite linear and the finite non-linear specification of Stack in Table 2. Moreover, it has been shown in [6] that a recursive BPA specification defining Stack has more than one equation; Stack cannot be defined by a single equation over BPA. Another example, well-known in the ACP literature, is that the process Bag cannot be defined by a finite recursive BPA specification, while it can be defined using a parallel operator as is present in the process algebra BPP [12] and in the axiom system PA[4] (see Table 1). The process Queue cannot be defined by a finite recursive specification in PA [7]; neither can it be defined in ACP with handshaking communicating [1]. Further well-known results are that communication adds, in the presence of global renaming operations, to the strength of PA (again [1]), and that abstraction via the τ-action and corresponding τ-laws further increase the expressive strength.

An appealing way of representing processes is by means of labeled transition graphs. In this paper, we propose to deal with expressiveness questions by considering geometric aspects of the labeled transition graphs associated with a class of recursive specifications. For instance, it is not hard to show that the labeled transition graphs associated with finite recursive BCCSP specifications have finitely many non-bisimilar vertices (modulo bisimulation). From this it follows that the process Stack is not definable by means of a finite recursive BCCSP specification, for the labeled transition graph of Stack depicted in Figure 1 has infinitely many non-bisimilar vertices.

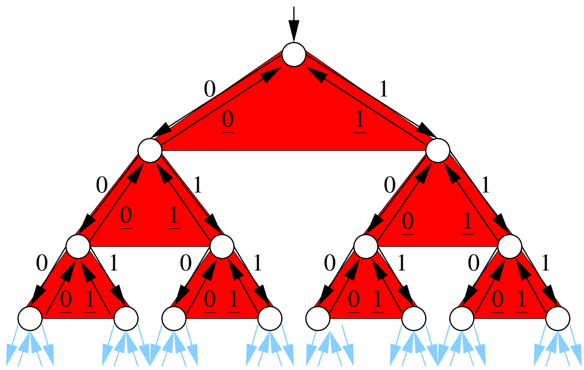

Fig. 1. The labeled transition graph STACK representing the process Stack

The main contribution of the present paper is the definition of two key concepts of the graph structure of processes, namely the density and the connectivity. For a process graph G, the density function maps a state s and a natural number n to the number of states of G with distance less or equal to n from s. The connectivity of a process graph G in a state s is a measure for how many different ways "of going from s to infinity" exist in G. Thus both density and connectivity are initially locally defined notions. It turns out that under certain conditions on a process graph both measures have also a global meaning.

The issue of properties of process graphs was also taken up by other researchers. In particular Caucal [10], Caucal and Montfort [9], and Burkart et al. [8] obtained sophisticated results from the mere appearance of the graphs. Although our primary motivation is the definability issue, one can imagine that progress in understanding the "geometry of processes" may have other benefits as well. Recently, there are major advances in visualising large state spaces of a plethora of processes (see work of Groote and van Ham reported in [15]). The process graphs of such large state spaces exhibit many interesting geometric phenomena that are at this moment largely unexplored. A better insight into the geometric structure of such processes is very likely to increase our awareness and intuition for such processes, with its obvious significance for verification applications.

We consider this study as a step towards a geometry of processes. It is to be expected that much more key notions will emerge. But already the present two parameters of graphs, density and connectivity, enable us to give "high-level" proofs of some non-definability theorems that before were obtained by intricate ad hoc proofs. Actually, the present note is not a first step towards a geometry of processes. Apart from the work already mentioned ([7,9,8]), one may also view the seminal paper [18] of Muller and Schupp to point in the direction of a geometrical study of processes, and likewise, there is a rich tradition of work on graphs, pattern graphs, graph grammars, and so on. There are also historical roots in the topological notion of Freudenthal, "ends" of topological spaces, followed up by the notion of context-free graph of Muller and Schupp.

2 Preliminaries on Labeled Transition Graphs

In this paper, we take the stance that a process is mathematically modeled as a rooted labeled transition graph. We fix the set A of *actions* that will be used as labels of edges in our graphs.

Definition 2.1. A *labeled transition graph* is a pair $(S, \rightarrow)$ consisting of a set S of *vertices* (or: *states*), and a *transition relation* $\rightarrow \subseteq S \times A \times S$. A *rooted* labeled transition graph is a triple $T = (S, \rightarrow, r)$ with $(S, \rightarrow)$ a labeled transition graph and $r \in S$ a distinguished state r called the *root* of T.

A labeled transition graph can be thought of as an edge-labeled directed graph with $\rightarrow$ as the set of labeled directed edges. It can also be thought of as an edge-labeled undirected graph if the direction of the edges, implied by the ordering of the triples in $\rightarrow$, is simply ignored. We shall rely on both views in the remainder of this paper. We proceed to define several general notions for labeled transition graphs. First we discuss the notions that depend on the direction of the edges, and then we discuss the notions that do not take the direction of the edges into account. Throughout, we fix a rooted labeled transition graph $T = (S, \rightarrow, r)$ (but most of our notions actually do not depend on the declaration of a distinguished root).

2.1 Labeled Transition Graphs as Directed Graphs

We write $s \xrightarrow{a} s'$ for $\langle s, a, s' \rangle \in \rightarrow$, and $s \rightarrow s'$ if there exists $a \in A$ such that $s \xrightarrow{a} s'$.

A *directed path* from a state s to a state s' in T is a sequence of states $s_0, \ldots, s_n$ such that $s = s_0 \rightarrow \cdots \rightarrow s_n = s'$. If there exists a path from s to s', then we also say that s' is *reachable* from s. It is convenient to take the number of transitions associated with a path as the length of the path (so the length of the path $s_0, \ldots, s_n$ is n, and not $n + 1$).

A state s is *normed* if there is a directed path from s to a state s' without outgoing transitions; the length of the shortest such path is called the *norm* of s. A labeled transition graph is *normed* if all its states are normed.

A state $s \in S$ is called a *coroot* of T if it has no outgoing transitions and there is a path to s from every other state in S. Note that if T has a coroot, then it is clearly unique, and, moreover, T is normed.

Let $T_i = (S_i, \rightarrow_i, r_i)$ $(i = 1, 2)$ be rooted labeled transition graphs. A binary relation $\mathcal{R} \subseteq S_1 \times S_2$ is a *bisimulation* between T_1 and T_2 if $s_1 \mathrel{\mathcal{R}} s_2$ implies for all $a \in A$:

(i) if $s_1 \xrightarrow{a} s_1'$, then there exists $s_2' \in S_2$ such that $s_2 \xrightarrow{a} s_2'$ and $s_1' \mathrel{\mathcal{R}} s_2'$;
(ii) if $s_2 \xrightarrow{a} s_2'$, then there exists $s_1' \in S_2$ such that $s_1 \xrightarrow{a} s_1'$ and $s_1' \mathrel{\mathcal{R}} s_2'$.

We write $s_1 \leftrightarroweq s_2$ if there exists a bisimulation $\mathcal{R}$ such that $s_1 \mathrel{\mathcal{R}} s_2$. Furthermore, we write $T_1 \leftrightarroweq T_2$ if there exists a bisimulation relation $\mathcal{R}$ such that $r_1 \mathrel{\mathcal{R}} r_2$.

A *self-bisimulation* on T is a bisimulation between T and itself. The rooted labeled transition graph T is *canonical* if every state is reachable from the root and the diagonal on S (i.e., the binary relation $\{\langle s, s\rangle \mid s \in S\}$) is the only self-bisimulation on T.

An *isomorphism* between T_1 and T_2 is a transition-preserving bijection between the subsets of states of T_1 and T_2 that are reachable from their respective roots. If there exists an isomorphism between T_1 and T_2, then we say that they are *isomorphic* (notation: $T_1 \simeq T_2$).

2.2 Labeled Transition Graphs as Undirected Graphs

We write $s \leftrightarrow s'$ if $s \rightarrow s'$ or $s' \rightarrow s$. An *undirected path* between states s and s' is a sequence of states $s_0, \ldots, s_n$ such that $s = s_0 \leftrightarrow \cdots \leftrightarrow s_n = s'$. Two states s and s' are *connected* if there exists an undirected path between s and s'; a labeled transition graph is *connected* if any two states are connected.

The *distance* $d(s, s')$ of states s and s' is the length of the shortest undirected path between s and s' if s and s' are connected, and ∞ otherwise. Clearly, distance is commutative, i.e., $d(s, s') = d(s', s)$. The degree $deg(s)$ of a state s is the cardinality of the set of directed edges that have s as their source or as their target, that is, we let $deg(s) = \left|\left\{\, \langle s, a, s'\rangle,\, \langle s', a, s\rangle \mid s \xrightarrow{a} s' \text{ or } s' \xrightarrow{a} s \right\}\right|$. If every state in T has a finite degree, then we say that T is *locally finite*. For all states s in a locally finite labeled transition graph the set $\{s' \mid d(s, s') \leq n\}$ of states at a distance less or equal some $n \geq 0$ is finite.

A labeled transition graph $T' = (S', \rightarrow')$ is a *subgraph* of T (notation: $T' \subseteq T$) if $S' \subseteq S$ and $\rightarrow' \subseteq \rightarrow$. A *connected component* of T is a maximal connected subgraph of T, i.e., it is a connected subgraph T' of T and, for all T'' such that $T' \subseteq T'' \subseteq T$, either $T'' = T'$ or T'' is not connected.

2.3 Density and Connectivity

We introduce the notions of density and connectivity to classify labeled transition graphs according to their geometrical structure. The density of a labeled transition graph T in a state s is a function that describes the dependency on n of the number of states inside a sphere with radius n around s. The connectivity of a labeled graph T in a state s is the limit, as n tends to infinity, of the number of infinite connected parts into which T splits outside of a sphere around s with radius n.

Let $n \in \mathbb{N}$ and s a state of T. By $In(s, n, T)$ and $Out(s, n, T)$ we mean the subgraphs of T that result by removing all states with distance greater than n from s, and respectively, with distance less than n from s, i.e. we let

$$In(s, n, T) = (S_{\text{in}}(s, n), \rightarrow_{\text{in}}) , \qquad Out(s, n, T) = (S_{\text{out}}(s, n), \rightarrow_{\text{out}}) ,$$

with

$$S_{\text{in}}(s, n) = \{s' \in S \mid d(s, s') \leq n\} , \quad S_{\text{out}} = \{s' \in S \mid d(s, s') \geq n\} ,$$

$$\rightarrow_{\text{in}} = \{\langle s_1, a, s_2\rangle \in \rightarrow \mid a \in A \ \&\ s_1, s_2 \in S_{\text{in}}(s, n)\} ,$$

$$\rightarrow_{\text{out}} = \{\langle s_1, a, s_2\rangle \in \rightarrow \mid a \in A \ \&\ s_1, s_2 \in S_{\text{out}}(s, n)\} .$$

Definition 2.2. Suppose that T is a locally finite labeled transition graph. The *(undirected) density* in a state s of T is the function $\boldsymbol{d}_s : \mathbb{N} \to \mathbb{N}$ defined by

$$\boldsymbol{d}_s(n) = |S_{\text{in}}(s, n)| \;,$$

which maps every natural number n to the number of states of the subgraph $In(s, n, T)$ of T. The *directed density* in a state s of T is the function $\boldsymbol{d}_s^{\rightarrow} : \mathbb{N} \to \mathbb{N}$ defined by

$$\boldsymbol{d}_s^{\rightarrow}(n) = \left\{ s' \mid \text{there is a path of length} \leq n \text{ from } s \text{ to } s' \right\} .$$

The *density* $\boldsymbol{d}_T$ (*directed density* $\boldsymbol{d}_T^{\rightarrow}$) of T is the density (the directed density) in the root of T. (Usually, T will be clear from the context, and then we shall drop the subscript T and simply write $\boldsymbol{d}$ and $\boldsymbol{d}^{\rightarrow}$ to denote the density, and respectively, the directed density of T.)

From this definition it is obvious that, for a locally finite and connected graph T, and for all states of s of T, the directed-density function $\boldsymbol{d}_s^{\rightarrow}$ of T in s is bounded by the density function $\boldsymbol{d}_s$ of T in s.

We shall generally be interested in bounds on the growth of the density function $\boldsymbol{d}$ locally in a vertex, or globally for all vertices. Let $f : \mathbb{N} \to \mathbb{R}$ a monotone increasing function, and s a state of T. We say that f is an *upper bound on the density of T in s* if and only if

$$(\exists\, n_0 \in \mathbb{N})\, (\forall n \in \mathbb{N}) \left[\, n \geq n_0 \;\;\Rightarrow\;\; \boldsymbol{d}_s(n) \leq f(n) \,\right] . \tag{1}$$

holds, that is iff $\boldsymbol{d}_s$ is almost everywhere bounded by f. We call f a *uniform upper bound* on the density of T if and only if, for all $s \in S$, f is an upper bound on the density of T in s. Analogously, upper bounds are defined for directed-density functions.

A function $f : \mathbb{N} \to \mathbb{R}$ is called *constant*, *linear*, *polynomial*, or *exponential* if $f \in \Theta(1)$, $f \in \Theta(n)$, $f \in \Theta(n^c)$ for some $c \geq 1$, or $f \in \Theta(c^n)$ for some $c > 1$, respectively. We agree to say that *the density of T is linear* (*polynomial*, or *exponential*) if and only if, for all $s \in S$, there exists a linear (and respectively, polynomial, or exponential) upper bound, but not a constant (and respectively, linear, or polynomial) upper bound on the density of T in s. We say that *the density of T is constant* iff, for all $s \in S$, there is a constant upper bound on the density of T in s. This agreement, which also applies for the density of T in a state s and for the directed density of T, is intended to allow succinct formulations of some of our statements. However, it has the consequence that some density functions are categorised imprecisely: graphs with super-linear, non-polynomial density functions like $n \log n$ are said to have polynomial density, and graphs with super-polynomial, but not-exponential functions like $n^{\log n}$ are agreed to have exponential density.

Without proof we now give a proposition, which relates the global property of a locally finite and connected labeled transition graph T to have linear, constant, polynomial, or exponential (undirected) density to the local property of T in a state s to have linear, constant, polynomial, or exponential (undirected) density in s, respectively.

Proposition 2.3. *Let T be a locally finite and connected labeled transition graph, and let s be a state of T. Then the density of T is linear (constant, polynomial, or exponential) if and only if the density of T in s is linear (or respectively, constant, polynomial, or exponential).*

Proposition 2.3 makes it possible to determine the "degree of growth" of the density in an arbitrary state of a locally finite and connected labeled transition graph by only considering the density in the root. For example, by a glance at the process graph STACK for the process Stack in Figure 1, one can recognise that STACK has exponential density in all of its states.

Now we are going to introduce the "connectivity" of a labeled transition graph T in a state s as the limit, as n goes to infinity, of the number of infinite connected components of $Out(s, n, T)$, the subgraph of T consisting of all states with distance greater or equal to n from s, and of all edges of T linking such states. This definition coincides with the definition of the "number of ends" of a locally finite, rooted graph that is used by Muller and Schupp in [17]. It seems to have played a motivating role for the concept of "context-free" graphs that has been introduced by the mentioned authors later in [18]. The "theory of ends", from which the definition of the "number of ends" stems, originated with Freudenthal's dissertation, on which [14] is based.

It is convenient to have notation for the set of all those connected components of a labeled transition graph that have infinitely many states; we define

$$icc(T) = \{T' \mid T' \text{ is an infinite connected component of } T\} .$$

(We say that a labeled transition graph is *infinite* if it has infinitely many states.) For the definition below of our connectivity measure it is important to note the following fact: For all locally finite and connected labeled transition graphs T and states s of T, the function $g : \mathbb{N} \to \mathbb{N},\ n \mapsto |icc(Out(s, n, T))|$ is well-defined and non-decreasing. This implies that the function g possesses a limit in $\mathbb{N} \cup \{\infty\}$, which we call the "connectivity of T in s."

Definition 2.4. Let T be a connected and locally finite rooted labeled transition graph and s a state of T. We define the *connectivity* $\boldsymbol{c}_s$ *of* T *in* s by

$$\boldsymbol{c}_s = \lim_{n\to\infty} |icc(Out(s, n, T))| \ \in \mathbb{N} \cup \{\infty\} .$$

By the *connectivity* $\boldsymbol{c}_T$ *of* T we mean the connectivity of T in its root.

Without proof we state the following proposition, which expresses the fact that, in locally finite and connected graphs, connectivity is a global concept that does not need to be relativised to individual states of a labeled transition graph. For such labeled transition graphs T, the connectivity of T is equal to the connectivity in every state of T.

Proposition 2.5. *Let T be a locally finite and connected labeled graph. Then, for all states s_1 and s_2 of T, $\boldsymbol{c}_{s_1} = \boldsymbol{c}_{s_2}$ holds, that is, the connectivity $\boldsymbol{c}_{s_1}$ of T in s_1 coincides with the connectivity $\boldsymbol{c}_{s_2}$ of T in s_2 (and hence also with the connectivity $\boldsymbol{c}_T$ of T, the connectivity of T in its root).*

Finally, it is important to note that both of the concepts "density function" and "connectivity" of labeled transition graphs are invariant under isomorphism, but not under bisimilarity.

3 BPA-Graphs and BPP-Graphs

BPA-graphs and BPP-graphs are the labeled transition graphs of processes definable in the process algebras BPA (Basic Process Algebra, [4]) and BPP (Basic Parallel Processes, [13]). Starting with work by Caucal and Montfort (see [9] and [11]) there have emerged a number of formal characterisations of the transition graphs of processes in well-known process algebras as the transition graphs described by certain labeled rewrite systems. These characterisations provide alternative definitions, which are widely used since, of process graphs belonging to process algebras like BPA and BPP. A particularly elegant framework is that of process rewrite systems due to Mayr in [16].

To limit technicalities in the two cases of classes of process graphs studied here, BPA-graphs and BPP-graphs, we base the definitions on the somewhat simpler framework of "labeled rewrite systems" (following the exposition in [8]).

Definition 3.1. An *alphabetic labeled (string) rewrite system* is a triple $\mathcal{R} = (V, \Sigma, R)$ where V is an *alphabet* (or set of *nonterminals*), and $R \subseteq V \times \Sigma \times V^*$ is a finite set of *rewrite rules*. We will generally denote a rewrite system $\mathcal{R} = (V, \Sigma, R)$ simply by R if V and Σ are clear from the context; rules $\langle u, a, v \rangle$ will generally be denoted as transitions $u \xrightarrow{a} v$.

Let (V, Σ, R) be a labeled rewrite system. Then the *prefix rewriting relation* $\mapsto$ of R is defined by

$$\mapsto \;=\; \big\{ \langle uw, a, vw \rangle \mid \langle u, a, v \rangle \in R,\; w \in V^* \big\} .$$

We extend $\mapsto$ to a "more-step prefix rewriting relation" $\mapsto^* \subseteq V^* \times \Sigma^* \times V^*$ by defining, for all $v, v' \in V^*$, a more-step transition $v \mapsto^* v'$ to be possible if and only if $v \mapsto u_1 \mapsto u_2 \mapsto \ldots \mapsto u_{n-1} \mapsto v'$ holds for some $u_1, \ldots, u_{n-1} \in V^*$. By the labeled transition graph *generated by* $\mapsto$ *from* u we mean the rooted labeled transition graph

$$\mathcal{T}(\mapsto, u) \;=\; \big(\big\{ v \in V^* \mid u \mapsto^* v \big\}, \mapsto, u \big) .$$

For an example, we consider the alphabetic rewrite system (V, Σ, R) with $V = \{A, B, C\}$, $\Sigma = \{a, b\}$, and set of rules

$$R = \{ A \xrightarrow{a} \lambda,\; A \xrightarrow{b} AB,\; B \xrightarrow{a} \lambda,\; B \xrightarrow{b} BC,\; C \xrightarrow{a} \lambda \} . \tag{2}$$

The transition graph $\mathcal{T}(\mapsto, u)$, which we call TEMPLE, is illustrated in Figure 2. Alternatively, this graph is defined by the recursive specification

$$\langle \mathrm{A} \mid \mathrm{A} = \mathrm{a} + \mathrm{b}\,\mathrm{AB},\; \mathrm{B} = \mathrm{a} + \mathrm{b}\,\mathrm{BC},\; \mathrm{C} = \mathrm{a} \rangle \tag{3}$$

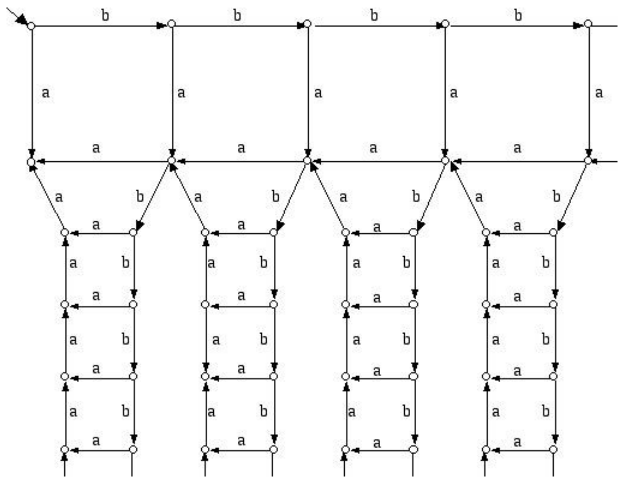

Fig. 2. The labeled transition graph TEMPLE

in the process algebra BPA, where the set of recursion equations is in "restricted Greibach normal form". The relationship indicated for this example between guarded recursive specifications in BPA and alphabetic rewrite systems justifies the following definition, by which the transition graph TEMPLE can be seen to be a "BPA-graph".

Definition 3.2. A rooted labeled transition graph $(T, \rightarrow, r)$ is called a BPA-*graph* iff there exists an alphabetic rewrite system (V, Σ, R) and $u \in V$ such that $(T, \rightarrow, r)$ is isomorphic to $\mathcal{T}(\mapsto, u)$.

For introducing transition graphs associated with recursive specifications in the process algebra BPP (of "Basic Parallel Processes"), we need some notation on multisets. Let V be a set. We denote by

$$\mathcal{M}(V) \;=\; \{\tilde{u} \mid \tilde{u} : V \rightarrow \mathbb{N},\ \tilde{u}(v) > 0 \text{ for finitely many } v \in V\}$$

the set of finite multisets over V. By $\oplus$ and $\ominus$ we denote the operations multiset union and multiset difference on the set $\mathcal{M}(V)$. For all $\tilde{u} \in \mathcal{M}(V)$, we let $|\tilde{u}| = \sum_{X \in V} \tilde{u}(X)$ the number of elements of the multiset $\tilde{u}$. Furthermore we designate by $\mathrm{ms}(w)$, for all $w \in V^*$, the multiset in $\mathcal{M}(V)$ that maps every $v \in V$ to the number of occurrences of v in w.

Let (V, Σ, R) be an alphabetic rewrite system. By the *multiset rewriting relation of* R we mean the rewriting relation $\rightsquigarrow \subseteq \mathcal{M}(V) \times \Sigma \times \mathcal{M}(V)$ that is defined, for all $\tilde{u}, \tilde{v} \in \mathcal{M}(\Sigma)$, by

$$\tilde{u} \rightsquigarrow \tilde{v} \iff (\exists \langle X, a, w \rangle \in R)\left[\,\tilde{u}(X) > 0 \;\;\&\;\; \tilde{v} = (\tilde{u} \ominus \mathrm{ms}(X)) \oplus \mathrm{ms}(w)\right]. \quad (4)$$

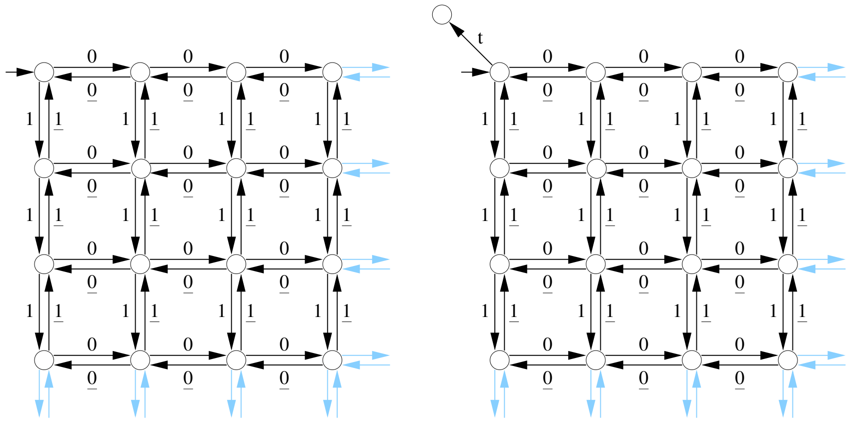

Fig. 3. The canonical process graphs of the process BAG (on the left-hand side), and of a terminating variant BAG_t of BAG (on the right-hand side)

Similar as the prefix-rewriting relation $\mapsto$ of R we extend $\leadsto$, to the "more-step multiset rewriting relation" $(\leadsto)^*$ of R. By the labeled transition graph *generated by from* $\tilde{u}$, where $\tilde{u} \in \mathcal{M}(V)$, we mean the rooted labeled transition graph

$$\mathcal{T}(\leadsto, \tilde{u}) \;=\; (\{\tilde{v} \in \mathcal{M}(V) \mid \tilde{u} \leadsto^* \tilde{v}\}, \leadsto, \tilde{u}) \;.$$

As an example, let (V, Σ, R) be the alphabetic rewrite system with $V = \{X, Y\}$, $\Sigma = \{0, \underline{0}, 1, \underline{1}\}$ and

$$R = \{X \xrightarrow{0} XY,\; X \xrightarrow{1} XZ,\; Y \xrightarrow{\underline{0}} \lambda,\; Z \xrightarrow{\underline{1}} \lambda\} \;.$$

The transition graph $\mathcal{T}(\leadsto, \mathrm{ms}(X))$ is the transition graph BAG on the left in Figure 3. This graph can also be specified by the recursive specification $\langle \mathrm{X} \mid \mathrm{X} = 0\,(\mathrm{X} \parallel \mathrm{Y}) + 1\,(\mathrm{X} \parallel \mathrm{Z}),\; \mathrm{Y} = \underline{0},\; \mathrm{Z} = \underline{1}\rangle$ in the process theory BPP, where $\parallel$ denotes the operator "merge" for parallel composition. Associativity and commutativity of $\parallel$ are the reason why BPP-specifications can be adequately formalised by multiset rewrite relations based on alphabetic rewrite relations. The relationship indicated here between guarded recursive specifications in BPP and alphabetic rewrite systems justifies the following definition, by which the process graph BAG can be recognised to be a "BPP-graph".

Definition 3.3. A rooted labeled transition graph $(T, \to, r)$ is called a *BPP-graph* if and only if there exists an alphabetic rewrite system (V, Σ, R) and some $u \in V$ such that $(T, \to, r)$ is isomorphic to $\mathcal{T}(\leadsto, \mathrm{ms}(u))$.

4 Density and Connectivity of BPA-Graphs

In this section we will discuss the notions of density and connectivity as they are found in BPA-graphs. We start with an experimental approach, by considering a number of examples.

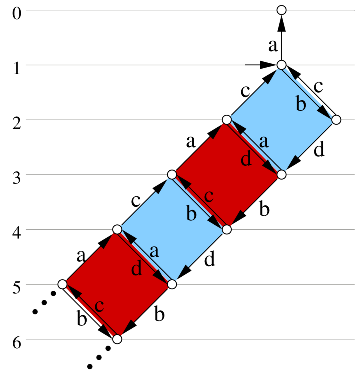

Fig. 4. The labeled transition graph RAILS

4.1 Examples

Example 4.1. The first example is the BPA-graph STACK in Figure 1 we encountered in Section 1. Its specification, both the easy to understand infinite one and the somewhat more sophisticated finite specification, have been given in Table 2 in the Introduction. It is straightforward to draw the infinite process graph corresponding to it. From the illustration in Figure 1 it is easy to read off the salient properties of the Stack graph:

- it is canonical (two different nodes are not bisimilar, i.e., they are the roots of subgraphs that are not bisimilar, or in our definition above, there is no non-trivial self-bisimulation of the whole graph);
- it is not normed, since all maximal traces are infinite (in other words, there is no coroot);
- the graph is not a tree since there are cycles, but it has a striking tree-like appearance which moreover is in some sense periodical, to be explained more precisely below;
- the density is exponential;
- the connectivity is infinite: cutting off a prefix of depth n, there arise 2^{n+1} icc's, so the limit for growing n is infinite.

Example 4.2. The BPA-graph in Figure 4, which we call RAILS, belongs to the guarded recursive BPA-specification $\langle \mathrm{X} \mid \mathrm{X} = \mathrm{a} + \mathrm{b}\,\mathrm{YX},\ \mathrm{Y} = \mathrm{c} + \mathrm{d}\,\mathrm{XY} \rangle$. Equivalently, it is given by the alphabetic rewrite system with set of rules $\left\{X \xrightarrow{a} \lambda,\ X \xrightarrow{b} YX,\ Y \xrightarrow{c} \lambda,\ Y \xrightarrow{d} XY\right\}$. The graph RAILS has the following properties:

- it is not canonical; nodes on the same level (i.e. distance to the coroot that is the highest node displayed) are bisimilar. So the graph can be compressed to its unique canonical form by identifying the nodes in a horizontal direction;
- the graph is normed;
- the density is linear;
- the connectivity is 1.

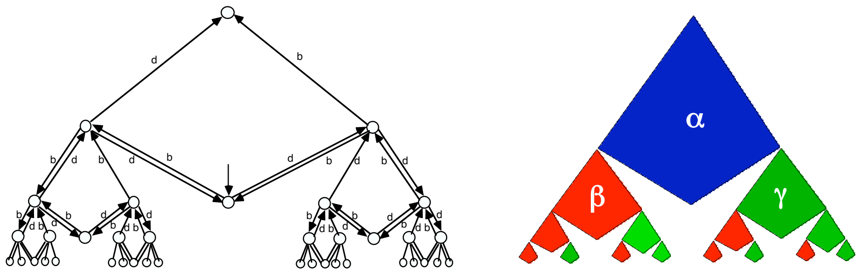

Fig. 5. The labeled transition graph KITES

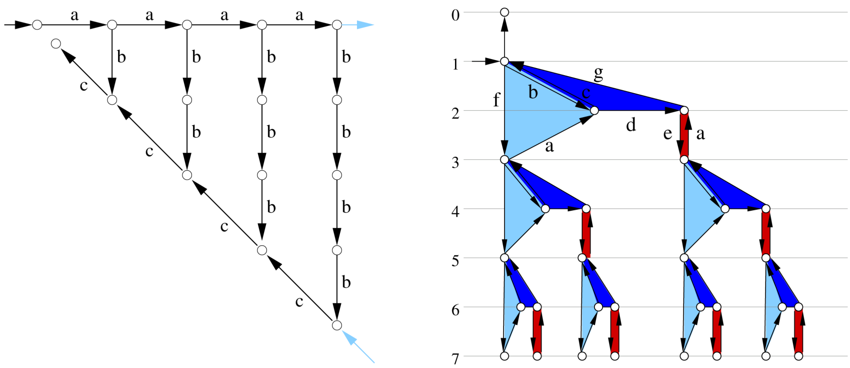

Fig. 6. The labeled transition graphsTRIANGLE and BUTTERFLIES

Example 4.3. KITES, in Figure 5. This is a canonical and normed BPA-graph, with exponential density and infinite connectivity. It corresponds to the recursive BPA-specification $\langle \mathrm{X} \mid \mathrm{X} = \mathrm{bY} + \mathrm{dZ},\ \mathrm{Y} = \mathrm{d} + \mathrm{dX} + \mathrm{bYY},\ \mathrm{Z} = \mathrm{b} + \mathrm{bX} + \mathrm{dZZ}\rangle$; its finite traces form the context-free language consisting of words containing as many b's as d's. Again it has a periodical tree-like decomposition.

Example 4.4. BUTTERFLIES, in Figure 6.

This BPA-graph has the recursive BPA-specification $\langle \mathrm{X} \,|\, \mathrm{X} = \mathrm{a} + \mathrm{b\,Y} + \mathrm{f\,XY},\ \mathrm{Y} = \mathrm{c\,X} + \mathrm{d\,Z},\ \mathrm{Z} = \mathrm{g\,X} + \mathrm{e\,XZ}\rangle$. The characteristics are as for KITES.

So far, our sequence of experiments revealed BPA-graphs that have either linear or exponential density. How about the graph BAG, or BAG_t in Figure 3? According to our discussion in the Introduction, BAG is not a BPA-graph. Clearly, it has polynomial (quadratic) density. Its connectivity is 1. It is not normed, but it is canonical. Here one could jump to the guess that all BPA-graphs have density linear or exponential—and not polynomial. However, Didier Caucal pointed out to us that, surprisingly, there are BPA-graphs with polynomial density. His beautiful example is displayed in Figure 2, as the graph called TEMPLE. It corresponds to the rewrite system with rules (2) and to the recursive

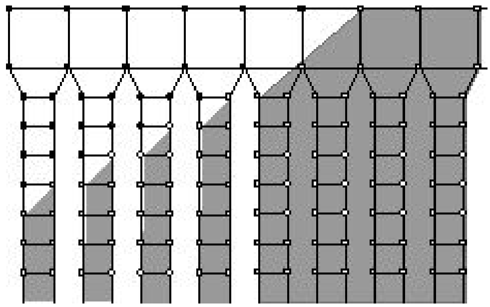

Fig. 7. Determining the connectivity of the graph TEMPLE

specification (3). It is a normed, canonical graph with quadratic density and infinite connectivity. That the connectivity is infinite, is clear from Figure 7, where the shadowy part, obtained by removing the prefix of depth 6, consists of 5 icc's; it is easy to see that the number of icc's grows to infinity with the prefix depth of nodes that are removed.

4.2 Periodic Decomposition

In the present note, we will not dwell on this phenomenon extensively, but refer instead to [2]. All BPA-graphs displayed above, including the quadratically dense TEMPLE, display a tree-like periodical decomposition, which was first observed and proved in [2]. That is, there are finitely many graph fragments that are strung together in a regular way.

So, for RAILS the graph fragment structure is described by the recursion term $\langle\alpha|\alpha = c(\alpha)\rangle$, where c stands for "connected to"; for STACK we have $\langle\alpha|\alpha = c(\alpha,\alpha)\rangle$; for TEMPLE we have $\langle\alpha|\alpha = c(\beta,\alpha),\ \beta = c(\beta)\rangle$; and for KITES, $\langle\alpha|\alpha = c(\beta,\gamma),\ \beta = c(\beta,\gamma),\ \gamma = c(\beta,\gamma)\rangle$. Alternatively, one may write these recursion terms as μ-terms, obtaining $\mu\alpha.c(\alpha)$, etc.

We continue by mentioning another important feature of BPA-graphs, this time with the restriction to normed graphs.

Theorem 4.5 (Caucal). *The class of normed* BPA*-graphs (or equivalently, the class of* BPA*-graphs with a coroot) is closed with respect to minimisation under bisimulation.*

It is a noteworthy fact that the restriction of normedness in this theorem is crucial: the class of all BPA-graphs is not closed under minimisation (for an example of an—unnormed—BPA-graph with a canonical graph that is not a BPA-graph, see Figure 3 in [8]).

4.3 Relating Density and Connectivity

We now state in Table 3 the relationship between the density and connectivity for BPA-graphs. Here we do not give the full proofs, but merely mention that they are obtained from an analysis of the structural μ-terms that describe tree-like

Table 3. Connectivity-density value pairs $\langle \boldsymbol{c}, \boldsymbol{d} \rangle$ that are possible for BPA-graphs

$\boldsymbol{c}$ versus $\boldsymbol{d}$	constant	linear	polynomial	exponential
0	✓	—	—	—
finite, nonzero	—	✓	—	—
∞	—	—	✓	✓

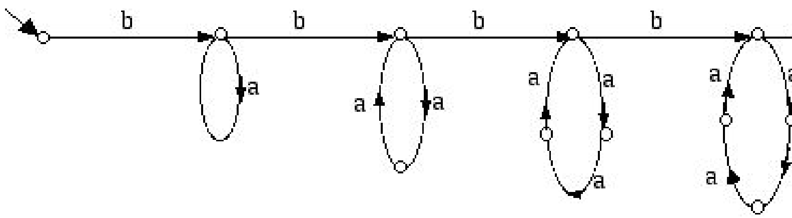

Fig. 8. The labeled transition graph RINGS

periodic decompositions obtained as described in [2]. As a caveat, we mention that the statements concerning Table 3 are still of tentative nature.

We now give an application. Figure 6 contains the graph TRIANGLE. Is it a BPA-graph? It corresponds to the context-sensitive language $\{ a^n b^n c^n \mid n \in \mathbb{N} \}$, which is not context-free; so our suspicion is that it is not a BPA-graph. Indeed this is the case: it is normed, canonical, with quadratic density. However, its connectivity is 1, and according to the table above, it should be infinite. Hence TRIANGLE is not a BPA-graph. Similar for BAG_t.

Now the fact that TRIANGLE and BAG_t are not BPA-graphs, does not yet mean that they are not BPA-definable. It could be that for TRIANGLE there is a BPA-specification E, with process graph $g(E)$, such that the canonical form $\mathrm{can}(g(E)) \leftrightarrows \mathsf{TRIANGLE}$. Since TRIANGLE is canonical, we even have $\mathrm{can}(g(E)) \simeq \mathsf{TRIANGLE}$ (recall that $\simeq$ stands for graph isomorphism). Now TRIANGLE is normed, hence $\mathrm{can}(g(E))$ is normed. So $g(E)$ is normed (since $\mathrm{can}(g(E)) \leftrightarrows g(E)$, and normedness is preserved by bisimilarity). So, we can apply Theorem 4.5 on the normed BPA-graph $g(E)$, and conclude that $\mathrm{can}(g(E))$ is again a BPA-graph; say $\mathrm{can}(g(E)) \simeq g(E')$ for some BPA-specification E'. Now we have that the BPA-graph $g(E') \simeq \mathsf{TRIANGLE}$ hence $g(E')$ has the same connectivity and density as TRIANGLE, namely $\boldsymbol{c} = 1$ and $\boldsymbol{d}$ is quadratic. But this is impossible.

Now let us consider BAG and RINGS, in Figure 8. Both have quadratic density and connectivity 1. They are therefore not BPA-graphs. But are they not BPA-definable? Invoking Caucal's theorem (Theorem 4.5) does not work here, since both graphs are not normed. Here a more powerful theory is needed, and this is found in the notion of "context-free graph" of Muller and Schupp, in conjunction with more recent work of Caucal in [11] and [8].

Suppose that BAG is BPA-definable. Then there exists a recursive specification E in BPA such that BAG is bisimilar with a tree-like periodic graph $g(E)$ as defined by Baeten, Bergstra, and Klop in [2]. Then $g(E)$ is a BPA-graph (in the sense of Definition 3.2).[1]

In [8] Burkart, Caucal, and Steffen have shown that, for *every* BPA-graph T, the canonical graph of T is a "pattern graph", which means that it can be generated from a finite (hyper)graph by a reduction sequence of length ω according to a deterministic (hypergraph) grammar.[2] Since BAG is itself a canonical graph and since therefore BAG is the canonical graph of the BPA-graph $g(E)$, it follows that BAG is a pattern graph.

A theorem due to Caucal in [11] states that all (rooted) pattern graphs of finite degree are "context-free" according to the definition of Muller and Schupp in [18].[3] It follows that BAG is context-free. However, it is not difficult to verify that BAG is actually *not* a context-free graph according to the definition in [18].

In this way we have arrived at a contradiction with our assumption that BAG is definable in BPA. For RINGS the same reasoning applies.

We conclude this section by mentioning a useful fact due to Muller and Schupp in [18] that characterises the class of transition graphs corresponding to recursively defined specifications in the process algebra PDP (containing "Pushdown Processes") as the class of "context-free graphs".

Proposition 4.6. *Every* BPA*-graph is context-free.*

5 Density of BPP-Graphs

In this section we investigate the possible densities of BPP-graphs. We prove that BPP-graphs have at most polynomial density, and apply this result to show that the paradigmatic process Queue, which is definable in the process algebra ACP, cannot be defined in BPP.

For the proof of the mentioned result concerning the density of BPP-graphs the following technical lemma will be essential. This lemma contains a bound on the number of finite multisets with k members over a set with m elements.

Lemma 5.1. Let V be a finite set with $m \in \mathbb{N}\backslash\{0\}$ elements. Then, for all $k \in \mathbb{N}$, the number of multisets over V with k elements is equal to $\binom{m+k-1}{k}$.

[1] In earlier papers of Caucal (e.g. in [9] and [11]) BPA-graphs were known under the name "alphabetic graphs".

[2] "Pattern graphs" according to this definition used by Caucal and Montfort in [9] are called "regular graphs" in the later paper [8] by Burkart, Caucal, and Steffen. Because the use of the attribute "regular" for process graphs could lead to wrong associations, we avoid this terminology from (hyper)graph rewriting here.

[3] Note that the class of "context-free" graphs in Muller and Schupp's definition does not coincide with the graphs associated with "context-free" processes (the class of BPA-graphs), but that it forms a strictly richer class of graphs corresponding to the class of transition graphs of push-down automata.

Furthermore it holds:

$$\Big(k \longmapsto \big|\{\, \tilde{v} \mid \tilde{v} \in \mathcal{M}(V),\, |\tilde{v}| = k \}\big|\Big) \;\in\; O(k^{m-1})\,, \tag{5}$$

$$\Big(k \longmapsto \big|\{\, \tilde{v} \mid \tilde{v} \in \mathcal{M}(V),\, |\tilde{v}| \le k \}\big|\Big) \;\in\; O(k^{m})\,. \tag{6}$$

Proof. Let $V = \{X_1, \ldots, X_m\}$ be a finite set with $m \in \mathbb{N}\backslash\{0\}$ elements. Then, for all $k \in \mathbb{N}$, the number of finite multisets over V with k elements can be computed as follows:

$$\begin{aligned}\big|\{\, \tilde{v} \mid \tilde{v} \in \mathcal{M}(V),\, |\tilde{v}| = k \}\big| &= \big|\{\, (x_1, \ldots, x_m) \mid 0 \le x_i \le k\,, \textstyle\sum_{i=1}^{k} x_i = k \}\big| \\ &= \big|\{\, (x_1, \ldots, x_{k+m-1}) \mid x_i \in \{0,1\},\, \textstyle\sum_{i=1}^{m+k-1} x_i = k \}\big| \\ &= \tbinom{m+k-1}{k}\,. \end{aligned} \tag{7}$$

For all $k \in \mathbb{N}$ with $k \ge m$ it holds:

$$\begin{aligned}\binom{m+k-1}{k} &= \frac{(m+k-1).(m+k-2)\ \ldots\ (m+1).m}{1.2\ \ldots\ (k-1).k} \\ &= \frac{(m+k-1)\ \ldots\ (k+1)}{1\ \ldots\ (m-1)} \;\le\; \frac{(2k)^{m-1}}{(m-1)!}\ .\end{aligned}$$

This implies $\big(k \mapsto \binom{m+k-1}{k}\big) \in O(k^{m-1})$, which in view of (7) demonstrates (5). Furthermore with $C_1 = \sum_{i=0}^{m-1} \binom{m+i-1}{i}$ and $C_2 = \frac{2^{m-1}}{(m-1)!}$ it follows:

$$\begin{aligned}\sum_{i=0}^{k} \tbinom{m+i-1}{i} &= \sum_{i=0}^{m-1} \tbinom{m+i-1}{i} + \sum_{i=m}^{k} \tbinom{m+i-1}{i} \\ &\le\; C_1 + C_2.\sum_{i=0}^{k} i^{m-1} \;\le\; C_1 + C_2.(k+1).k^{m-1} \\ &=\; C_1 + C_2.(k^m + k)\end{aligned}$$

which because of $(k \mapsto k^m + k) \in O(k^m)$ and (7) now demonstrates (6). $\square$

Now we are able to state and prove our result concerning the density of BPP-graphs.

Theorem 5.2. *For every* BPP*-graph there exists a polynomial uniform upper bound on its density.*

Proof. Let $T = (S, \rightarrow, r)$ be a BPP-graph. That is, there exists an alphabetic rewrite system (V, Σ, R) with $V = \{X_1, \ldots, X_m\}$ such that T is isomorphic to the rooted labeled transition graph $\mathcal{T}(\rightsquigarrow, \mathrm{ms}(X_1))$, where $\rightsquigarrow$ is the multiset rewrite relation of R. Since the density of a labeled transition graph is invariant under isomorphism, we may assume, without loss of generality, that T actually *is* the rooted labeled transition graph $\mathcal{T}(\rightsquigarrow, \mathrm{ms}(X_1))$.

Now we let $\tilde{u}$, with $\tilde{u} \in \mathcal{M}(V)$, be an arbitrary state of T and investigate the density of T in $\tilde{u}$.

We let $N = \max\{1, \left(\max_{\langle X,a,v\rangle \in R} \lg(v)\right) - 1\}$. In particular, N is greater or equal to the maximal length of the right-hand side of a rule in R minus one. Therefore the definition of N implies, in view of the definition of $\rightsquigarrow$:

$$\tilde{v}_1 \rightsquigarrow \tilde{v}_2 \implies |\tilde{v}_2| \leq |\tilde{v}_1| + N \qquad \text{(for all } \tilde{v}_1, \tilde{v}_2 \in \mathcal{M}(V)) , \tag{8}$$

$$\tilde{v}_1 \rightsquigarrow \tilde{v}_2 \implies |\tilde{v}_1| \leq |\tilde{v}_2| + 1 \leq |\tilde{v}_2| + N \qquad \text{(for all } \tilde{v}_1, \tilde{v}_2 \in \mathcal{M}(V)) . \tag{9}$$

Now let, for all $n \in \mathbb{N}$, $=^{\leq n}_{(R)}$ be defined by

$$=^{\leq n}_{(R)} = \bigcup_{i=0}^{n} \left(\rightsquigarrow \cup (\rightsquigarrow)^{-1}\right)^{i}$$

as the restriction of the convertibility relation $=_{(R)} = (\rightsquigarrow \cup (\rightsquigarrow)^{-1})^*$ of $\rightsquigarrow$ to conversions of length less or equal to n. Using (8) and (9) it can be proved by induction on n that, for all $n \in \mathbb{N}$,

$$\tilde{v}_1 =^{\leq n}_{(R)} \tilde{v}_2 \implies |\tilde{v}_2| \leq |\tilde{v}_1| + n.N \qquad \text{(for all } \tilde{v}_1, \tilde{v}_2 \in \mathcal{M}(V)) \tag{10}$$

holds. Since $\tilde{v}_1 =^{\leq n}_{(R)} \tilde{v}_2$ holds if and only if $d(\tilde{v}_1, \tilde{v}_2) \leq n$ is the case, (10) entails

$$d(\tilde{v}_1, \tilde{v}_2) \leq n \implies |\tilde{v}_2| \leq |\tilde{v}_1| + n.N \qquad \text{(for all } \tilde{v}_1, \tilde{v}_2 \in \mathcal{M}(V)) ,$$

for all $n \in \mathbb{N}$. Now this implies that, for all $n \in \mathbb{N}$, a superset of the set $S_{\text{in}}(\tilde{u}, n)$ of states of the subgraph $In(\tilde{u}, n, T)$ of T can be found as follows:

$$S_{\text{in}}(\tilde{u}, n) = \left\{\tilde{v} \in \mathcal{M}(V) \mid d(\tilde{u}, \tilde{v}) \leq n\right\} \subseteq \left\{\tilde{v} \in \mathcal{M}(V) \mid |\tilde{v}| \leq |\tilde{u}| + n.N\right\} .$$

By applying Lemma 5.1 now a bound on the density of T in $\tilde{u}$ can be established as follows:

$$\boldsymbol{d}_{\tilde{u}} = (n \mapsto |S_{\text{in}}(\tilde{u}, n)|) \in O((|\tilde{u}| + n.N)^m) \subseteq O(n^m) .$$

Hence there is a polynomial upper bound on the density of T in $\tilde{u}$; moreover, for such an upper bound a polynomial of the order of the number of variables in the alphabetic rewrite system underlying T can be chosen.

Since in this argument $\tilde{u}$ has been an arbitrary state of T, we have established that there is a polynomial which is a uniform upper bound on the density of T. □

Now we turn to the paradigmatic process Queue in the first-in-first-out version with unbounded capacity. An infinite specification of Queue in BPA is given in Table 4. The canonical process graph of QUEUE is sketched, for the case $D = \{0, 1\}$ in Figure 9. Compared with the easier graphs of the paradigmatic processes Stack and Bag, the structure of this canonical graph is more complex. By using Proposition 2.5 it is easy to see from Figure 9 that $\boldsymbol{c}_{\mathsf{QUEUE}} = 1$. Hence the connectivity of QUEUE is the same as that of BAG, but different from that of STACK. On the other hand, we will prove below that the density of QUEUE is greater than that of BAG.

Table 4. Queue, infinite BPA-specification

$$\mathrm{Q} = \mathrm{Q}_\lambda = \sum_{d\in D} \mathrm{r}_1(\mathrm{d}) \cdot \mathrm{Q}_\mathrm{d}$$
$$\mathrm{Q}_{\sigma \mathrm{d}} = \mathrm{s}_2(\mathrm{d}) \cdot \mathrm{Q}_\sigma + \sum_{e\in D} \mathrm{r}_1(\mathrm{e}) \cdot \mathrm{Q}_{\mathrm{e}\sigma\mathrm{d}}$$
(for $d \in D$, and $\sigma \in D^*$)

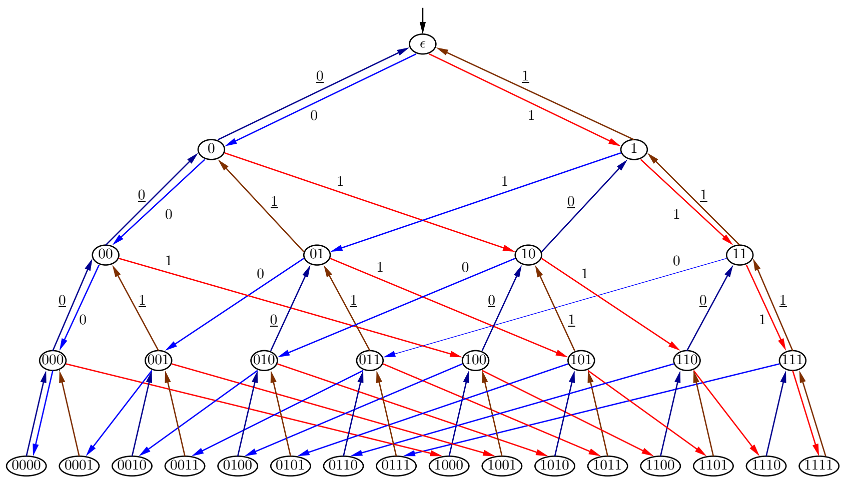

Fig. 9. The canonical process graph QUEUE of Queue

Table 5. Queue, finite ACP-specification with renaming

$$\mathrm{Q} = \sum_{d\in D} \mathrm{r}_1(\mathrm{d})(\rho_{\mathrm{c}_3 \to \mathrm{s}_2} \circ \partial_H)(\rho_{\mathrm{s}_2 \to \mathrm{s}_3}(\mathrm{Q}) \parallel \mathrm{s}_2(\mathrm{d}) \cdot \mathrm{Z})$$
$$\mathrm{Z} = \sum_{d\in D} \mathrm{r}_3(\mathrm{d}) \cdot \mathrm{Z}$$

It is clearly desirable to obtain a finite specification of this process. It was proved by Bergstra and Tiuryn in [7] that neither the process algebra BPA is sufficient for that, nor in fact is its extension PA. Building on this result, Baeten and Bergstra in [1] proved the even stronger statement that Queue cannot be defined in ACP *with handshaking communication* under the weak additional assumption that the pushing and popping actions are not the result of communications. However, in [1] also a finite recursive specification of Queue is given in ACP *with global renaming operators*. This beautiful specification (see Table 5) is originally due to Hoare who used a "chaining"-operation.

Proposition 5.3. *Let D be a finite set with $|D| > 1$. Then the canonical process graph* QUEUE(D) *of* Queue(D) *has exponential density.*

Proof. From the infinite BPA-specification for Queue in Table 4 it is easy to verify that, for all $\sigma_1, \sigma_2 \in D$ with $\sigma_1 \neq \sigma_2$, the subprocesses Q_{σ_1} and Q_{σ_2} of Q in Table 4 cannot be bisimilar: the sequence of popping moves for the processes Q_{σ_1} and Q_{σ_2} must be different. Hence the canonical process graph QUEUE for Queue can be drawn in tree space; for the example of $D = \{0,1\}$, QUEUE(D) is hinted in Figure 9.

We first consider the case $|D| = 2$. Then it is easy to verify that the function $(n \mapsto 2^{n+1} - 3)$ is an asymptotic lower bound on the density of QUEUE (for each vertex v by a sequences of length n of transitions in downwards-direction there are 2^n vertices reachable), and the function $\left(n \mapsto \frac{4}{3}(4^n - 1)\right)$ is an upper bound on the density of QUEUE (as an easy consequence of the summation formula for a geometric series in view of the fact that in QUEUE there are at most four vertices reachable from an arbitrary vertex by a single transition).

In the general case, where $k = |D|$, it is easy to verify that $\left(n \mapsto \frac{k(k^n-1)}{k-1} - 1\right)$ is an asymptotic lower bound on the density of QUEUE, and that $\left(n \mapsto \frac{2k((2k)^n-1)}{2k-1}\right)$ is an upper bound on the density of QUEUE. □

The following theorem states two conditions, canonicity and the existence of an exponential “lower bound” on the directed density, under which a labeled transition graph is not a BPP-graph. In the proof it is shown that these conditions enable to deduce a contradiction with Theorem 5.2, the statement that BPP-graphs have at most exponential density.

Theorem 5.4. *Let T be a rooted process graph that is canonical. Furthermore, there exists an exponential function that is not an upper bound on the directed density of T. Then T is not definable in* BPP*.*

Proof. Let $T = (S, \to, r)$ be a rooted process graph that is canonical and for which there exists an exponential function that is not an upper bound on the directed density of T.

Suppose that T is definable in BPP. Then there exists an alphabetic rewrite system $\mathcal{R} = (V, \Sigma, R)$ with $V = \{X_1, \ldots, X_m\}$ such that for the rooted transition graph $T' = (S', \to', r') = \mathcal{T}(\rightsquigarrow, \mathrm{ms}(X_1))$ it holds:

$$T' \leftrightarroweq T\,. \tag{11}$$

By Theorem 5.2 the density of T' is at most polynomial, and hence there exists a polynomial upper bound p on the density of T' in the state $r' = \mathrm{ms}(X_1)$; hence $\boldsymbol{d}_{r'}(n) \le p(n)$ holds for all but finitely many $n \in \mathbb{N}$. By assumption on T there exists an exponential function $f : \mathbb{N} \to \mathbb{R}$ that is not an upper bound on the directed density of T, which means that there are infinitely many $n \in \mathbb{N}$ such that $f(n) < \boldsymbol{d}_r^{\to}(n)$ holds. As a consequence of the fact that $p(n) < f(n)$ holds for all but finitely many n, $\boldsymbol{d}_{r'}(n) < \boldsymbol{d}_r^{\to}(n)$ must hold for infinitely many n. Due to this we can choose $n_0 \in \mathbb{N}$ with the property that

$$|(S')_{\mathrm{in}}(r', n_0)| = \boldsymbol{d}_{r'}(n_0) \;<\; \boldsymbol{d}_r^{\to}(n_0) = |S_{\mathrm{in}}^{\to}(r, n_0)| \tag{12}$$

holds, where

$$S_{\mathrm{in}}^{\to}(r, n_0) = \left\{ s \in S \mid \text{there is a path of length} \le n_0 \text{ from } r \text{ to } s \right\}.$$

Now it follows from (11) by repeated applications of the back-condition for a bisimulation between T' and T linking r' and r that each state in T with distance less or equal to n_0 from r must be bisimilar to a state in T' with distance less or equal to n_0 from r'. Due to (12), the difference in cardinality between $(S')_{\text{in}}(r', n_0)$ and $S^{\rightarrow}_{\text{in}}(r, n_0)$, it follows that there must exist different states s_1 and s_2 in $S^{\rightarrow}_{\text{in}}(r, n_0)$, and hence of T, and a state s' in $(S')_{\text{in}}(r', n_0)$, and hence of T', such that s' (in T') is bisimilar both to s_1 and s_2 (in T). But this entails that actually s_1 and s_2 are two different bisimilar states of T, which is a contradiction with the assumption that T is a canonical graph.

Therefore the assumption that T is definable in BPP cannot be sustained. □

By using this theorem, we are finally able to show that, for sets D with more than one element, the paradigmatic process Queue(D) is not definable in BPP.

Corollary 5.5. *For all finite sets D with $|D| > 1$,* Queue(D) *is not definable in* BPP*.*

Proof. We assume that D is a finite set with more than one element and that Queue(D) is definable in BPP. This means that there exists a guarded recursive specification E in BPP with Queue(D) as a solution. From the specification E a BPP-graph T can be extracted which has the property that it is bisimilar to the canonical process graph QUEUE(D) of Queue(D), showing that the graph QUEUE(D) is definable in BPP. However, Theorem 5.4 implies, in view of Proposition 5.3, that QUEUE(D) is actually not definable in BPP. We have obtained a contradiction. □

We conclude with a question concerning the possible values of connectivity and the relationship between connectivity and density for BPP-graphs.

Question 5.6. What can be said about the connectivity of BPP*-graphs? Is there a useful concept of "regular decomposition" for* BPP*-graphs? Is there perhaps, similar as for* BPA*-graphs, a correspondence statement that relates density and connectivity also for* BPP*-graphs?*

Acknowledgement. We are very grateful to Didier Caucal for the graph TEMPLE with its specification, and for pointing out its consequences. We thank Henk Barendregt for discussions about this paper and posing the definability questions for the graphs RINGS and TRIANGLE.

References

1. J. C. M. Baeten and J. A. Bergstra. Global renaming operators in concrete process algebra. *Information and Computation*, pages 205–245, 1988.
2. J. C. M. Baeten, J. A. Bergstra, and J. W. Klop. Decidability of bisimulation equivalence for process generating context-free languages. *Journal of the ACM*, 40(3):653–682, 1993.
3. J. C. M. Baeten and W. P. Weijland. *Process Algebra.* Number 18 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1990.

4. J. A. Bergstra and J. W. Klop. The algebra of recursively defined processes and the algebra of regular processes. In J. Paredaens, editor, *Proceedings of ICALP'84*, volume 172 of *LNCS*, pages 82–95. Springer, 1984.
5. J. A. Bergstra and J. W. Klop. Process algebra for synchronous communication. *Information and Control*, 60(1–3):109–137, January/February/March 1984.
6. J. A. Bergstra and J. W. Klop. Process algebra: specification and verification in bisimulation semantics. In M. Hazewinkel, J. K. Lenstra, and L. G. L. T. Meertens, editors, *CWI Monograph 4, Proceedings of the CWI Symposium Mathematics and Computer Science II*, pages 61–94, Amsterdam, 1986. North-Holland.
7. J. A. Bergstra and J. Tiuryn. Process algebra semantics for queues. *Fundamenta Informaticae*, X:213–224, 1987.
8. O. Burkart, D. Caucal, and B. Steffen. Bisimulation collapse and the process taxonomy. In *Proceedings of CONCUR'96*, 1996.
9. D. Caucal and R. Montfort. On the transition graphs of automata and grammars. In *Proceedings of WG 90*, volume 484 of *LNCS*, pages 61–86. Springer, 1990.
10. D. Caucal. Graphes canoniques de graphes algébriques. *Theoret. Inform. and Appl.*, 24(4):339–352, 1990.
11. D. Caucal. On the regular structure of prefix rewriting. *Theoretical Computer Science*, 1992.
12. S. Christensen. *Decidability and Decompostion in Process Algebras.* PhD thesis, University of Edinburgh, 1993.
13. Søren Christensen, Yoram Hirshfeld, and Faron Moller. Bisimulation equivalence is decidable for basic parallel processes. In *CONCUR*, pages 143–157, 1993.
14. H. Freudenthal. Über die Enden topologischer Räume und Gruppen. *Mathematische Zeitschrift*, 33:692–713, 1931.
15. J. F. Groote and F. J. J. van Ham. Interactive visualization of large state spaces. *International Journal on Software Tools for Technology Transfer*, 2005.
16. R. Mayr. Process rewrite systems. *Information and Computation*, 156(1):264–286, 2000.
17. D. E. Muller and P. E. Schupp. Groups, the theory of ends, and context-free languages. *Journal of Computer and System Sciences*, 26:295–310, 1983.
18. D. E. Muller and P. E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theoretical Computer Science*, 37:51–75, 1985.