ELSEVIER

Theoretical Computer Science 211 (1999) 311–337

Theoretical Computer Science

Multisets and structural congruence of the pi-calculus with replication

Joost Engelfriet*,[1], Tjalling Gelsema

Department of Computer Science, Leiden University, P.O. Box 9512, 2300 RA Leiden, Netherlands

Received January 1995; revised October 1996
Communicated by G. Rozenberg

Abstract

In the π-calculus with replication, two processes are multiset congruent if they have the same semantics in the multiset transition system Mπ. It is proved that (extended) structural congruence is the same as multiset congruence, and that it is decidable.

Keywords: Multiset; Congruence; Pi-calculus; Petri nets

1. Introduction

This paper is a sequel to [2]. Therefore, the reader is assumed to be familiar with the concepts and results of [2]. The results of the present paper were already announced and discussed in the introduction of [2]. In the present introduction we recall some facts from [2] and discuss some specific aspects of the results and proofs of this paper.

A particularly elegant version of the π-calculus was presented in [6]. The main aspects in which it differs from the usual π-calculus of [8] are twofold.

First, it has replication as an operation on processes, rather than recursion. The replication $!P$ of a process P consists of the parallel composition of infinitely many copies of P. Replication can be used to simulate recursion, but is much easier to handle theoretically. Thus, it seems to be more basic than recursion. In a certain sense, from the point of view of formal language theory, replication is similar to the Kleene star operation in regular expressions, whereas recursion is similar to context-free grammars.

Second, a natural relation of structural congruence between process terms is defined, and used to present the axioms and rules of the transition system of the π-calculus in a compact way. The idea is that process terms are descriptions of processes, and that these processes are characterized by their "spatial" structure (see also [7]). In other

* Corresponding author. E-mail: engelfriet@wi.leidenuniv.nl.

[1] The research of this author was supported by the Esprit Basic Working Group No. 6067 CALIBAN.

PII: S0304-3975(97)00179-5

words, two process terms describe the same process if and only if they have the same structure, and such process terms are said to be structurally congruent. Compactness of the axioms and rules is obtained by the natural stipulation that structurally congruent process terms have the same behaviour (in fact, it is the behaviour of the process that they describe). As an example, process terms $(P \mid Q) \mid R$ and $Q \mid (R \mid P)$ are structurally congruent because they both describe a process consisting of three subprocesses P, Q, and R that are placed in parallel, i.e., the parallel composition of P, Q, and R. Structural congruence is defined in [6] to be the smallest congruence that satisfies eight such structural laws.

These two main aspects are closely related. Replication is a typically structural operation on processes, just like parallel composition. In fact, it is "just" an infinite version of parallel composition. On the other hand, recursion is usually viewed as a behavioural construction (although it is also possible to view it structurally, by unfolding the recursion). Thus, the use of replication fits well in the structural approach; in fact, replication can be completely described by structural laws, and no separate transition rules are needed.

Restriction, of the scope of a name to a (sub)process, is also usually viewed as a structural operation. Certain subprocesses of a process "know" the name and others do not, and this can be seen as "spatial" information. In fact, in [5], the operations of CCS are divided into static (or structural) operations and dynamic (or behavioural) operations. The static operations are parallel composition, restriction, and relabeling, and the dynamic operations are prefix (or guard), sum, and recursion. Since the π-calculus of [6] has no relabeling or sum, and dynamic recursion is replaced by static replication, only the use of guards is dynamic in this version of the π-calculus.

In [3, 4] the process terms of CCS are interpreted as "flowgraphs" that correspond intuitively to their structure, i.e., to the processes they describe (see also the informal use of flowgraphs in [5, 8]). Structural laws for the process terms of CCS (called "laws of flow") are given that are sound and complete with respect to the flowgraph interpretation. To be more precise, two such process terms have the same flowgraph if and only if they are in the smallest congruence (with respect to the static operations) that satisfies the given structural laws. Also, process terms with the same flowgraph have the same behaviour, where, in this case, "same" means "strongly bisimilar". The corresponding laws for strong bisimilarity are the "static laws" in Section 3.4 of [5].

In [2] a "multiset semantics" (or Petri net semantics) of the π-calculus of [6] is given that is closely related to the structural approach. A transition system Mπ is defined of which the states are "solutions", which are multisets of molecules, and a "molecule" is a guarded solution (the chemical terminology is taken from the Chemical Abstract Machine of [1]). A multiset of molecules can be viewed as a "spatial" distribution of molecules, where several copies of the same molecule may be present in the solution. In accordance with the operation of replication, there may even be infinitely many such copies. Moreover, a semantic mapping is defined that associates with each process term of the π-calculus a state of Mπ. It is proved in [2] that the semantic mapping is a

strong bisimulation between a process term and its corresponding multiset in Mπ. Thus, from the interleaving point of view they have the same behaviour. The gain is that, intuitively, the behaviour of the multiset in Mπ is the "true concurrency" behaviour of the process term.

Two process terms are said to be multiset congruent if they correspond to the same multiset in Mπ. It is claimed in [2] that the multiset corresponding to a process term represents its "spatial" structure, i.e., the process it describes. In fact, the multisets can be viewed as a kind of nested flowgraphs ("nested" because multiset congruence is a congruence for all operations of the π-calculus rather than only the static ones). It turned out in [2] that, indeed, multiset congruence and structural congruence (as defined in [6]) are closely related, but unfortunately not as closely as one would wish. It was shown that structurally congruent process terms are multiset congruent, but not vice versa. However, the failure of the reverse direction seemed to be due to the omission of a few natural structural laws that from an intuitive point of view should be valid. An example of such a law is the structural equivalence of $!P \mid !P$ and $!P$. This is basically a cardinality law which expresses that adding infinitely many copies of P to infinitely many copies of P still leaves you with infinitely many copies of P. In this paper we show that, indeed, after the addition of a few such natural structural laws (as proposed in [2]), structural congruence and multiset congruence are the same. This means that the laws of structural congruence are now sound and complete with respect to the multiset semantics (or rather, its structural part). It is the analogue of the results of [4] for the π-calculus. As a second result, we show that multiset congruence, and hence (extended) structural congruence, is decidable. Clearly, any "good" notion of static, structural equivalence should be decidable. Thus, the two results of this paper support the thesis that the general notion of structure of processes, as introduced in [3–5] for CCS and in [6] for the π-calculus, is a natural one.

The main technical concept in the proof of the two main results is that of a "connected" solution. Each molecule in a solution "knows" a number of names (i.e., communication links). Some of these names are public (or global), i.e., known to all molecules, whereas others are secret (or local), i.e., known to a restricted number of molecules only. The secret names are also called "new" names, because they are new, unique, names that are introduced by the semantical mapping as a result of the restrictions that occur in the process term. As an example, a replication of a restriction gives infinitely many copies of the restriction, each of which should have its own "new" name. Let us now say that two molecules are "related" if they "know" at least one common "new" name. This is a natural relationship, because two such molecules are into the same secret. We will say that a solution is connected if the resulting graph of molecules with their relationships is connected, in the usual sense. More in general, any solution can be divided into "connected components". This division gives an intuitive picture of clusters of relationships. A process term is said to be connected if its semantics in Mπ is a connected solution. The proof of the main results uses a normal form lemma: every process term is structurally congruent with one that only replicates connected subterms. Intuitively, if a connected component corresponds to a "module"

that does a certain job, then it is natural to replicate such a module. The normal form lemma says that the use of replication can be restricted to such cases.

Whereas the main aim of [2] was to present a natural multiset semantics of the small π-calculus that should reflect the true concurrent *behaviour* of its process terms, the main aim of this paper is to show that this multiset semantics also reflects the spatial *structure* of the process terms of the small π-calculus, presenting a quite involved proof of the completeness and decidability of (extended) structural congruence, based on the (new) notion of a connected process.

The structure of the paper is as follows. In Section 2 some definitions and results from [6, 2] are recalled. Section 2 contains in particular the full definition of structural congruence, with the addition of the new structural laws. In Section 3 we discuss a few basic properties of multisets in general that are needed in the formal proofs. Section 4 introduces the notion of a connected solution and investigates the relationship between connectedness and multiset union. Section 5 contains the above-mentioned normal form of process terms. In Section 6 the two main results are proved: structural congruence and multiset congruence are the same, and decidable.

2. Preliminaries

Although the reader is assumed to be familiar with the basic definitions in [2], we briefly recall some of them. The reader who wishes to skip this section should realize that the new laws (2.4) and (3.2)–(3.5) below are added to the structural congruence of [6], and should glance at the new "copy of" terminology introduced just before Lemma 2.

$\mathbf{N} = \{1, 2, 3, \ldots\}$ is the set of positive natural numbers, and N is the infinite set of names. The set of all x that satisfy property $p(x)$, will be denoted $\{x : p(x)\}$ (rather than $\{x \mid p(x)\}$, to avoid confusion with parallel composition).

The syntax for process terms of the "small" π-calculus from [6] is

$$P ::= \bar{x}y.P, \quad x(y).P, \quad \mathbf{0}, \quad P \mid P, \quad !P, \quad (vy)P$$

where x and y are names in N.

The strings $\bar{x}y$ and $x(y)$ are called guards (over N), and the terms $\bar{x}y.P$ and $x(y).P$ are guarded terms. For processes P and Q, $P \mid Q$ is the parallel composition of P and Q, $(vy)P$ is the restriction of y to P, $!P$ is the replication of P, and $\mathbf{0}$ is the inactive or zero process. The y in $x(y).P$ and in $(vy)P$ binds all free occurrences of y in P. For process P, $\mathrm{fn}(P)$ denotes the set of names that occur free in process P, and $P[z/y]$ denotes the result of substituting z for all free occurrences of y in P.

In [2], $P \equiv Q$ denotes the structural congruence of P and Q as defined in [6]. However, as discussed in [2], we propose to extend that congruence with a number of new laws. Thus, in this paper we denote by $\equiv$ the so extended structural congruence, and we are dealing with what is called in [2] the *extended* small π-calculus. For clearness sake, we present the full definition of $\equiv$.

Structural congruence, denoted $\equiv$, is the smallest congruence over the set of all process terms such that

(α) $P \equiv Q$ whenever P and Q are α-convertible,

(1.1) $P \mid \mathbf{0} \equiv P$,

(1.2) $P \mid Q \equiv Q \mid P$,

(1.3) $P \mid (Q \mid R) \equiv (P \mid Q) \mid R$,

(2.1) $(vx)(vy)P \equiv (vy)(vx)P$,

(2.2) $(vx)P \equiv P$
provided $x \notin \mathrm{fn}(P)$,

(2.3) $(vx)(P \mid Q) \equiv P \mid (vx)Q$
provided $x \notin \mathrm{fn}(P)$,

(2.4) $(vx)g.P \equiv g.(vx)P$
provided x does not occur in g,

(3.1) $!P \equiv P \mid !P$,

(3.2) $!(P \mid Q) \equiv !P \mid !Q$,

(3.3) $!!P \equiv !P$,

(3.4) $!\mathbf{0} \equiv \mathbf{0}$, and

(3.5) $!P \mid !P \equiv !P$.

Structural laws (2.4) and (3.2)–(3.5) are new with respect to [6]. Law (3.5) was not mentioned in [2]. The reason is that, as pointed out by one of the referees of [2], it can be proved from the others, as follows: $!P \equiv !!P \equiv !(P \mid !P) \equiv !P \mid !!P \equiv !P \mid !P$, by structural laws (3.3), (3.1), (3.2), and (3.3), respectively. It is included in the above list for its usefulness.

We note here that in [3–5] a stronger version of law (2.3) is used, of which the analogue in the π-calculus would be the following law (2.3)′: $(vx)(P \mid Q) \equiv (vx)P \mid (vx)Q$ provided P and Q cannot communicate along the link x (formalized in a straightforward way). Clearly, law (2.3)′, together with law (2.2), implies law (2.3). The results to be proved in this paper show that law (2.3)′ does not follow from the above laws, e.g., it is not true that $(vx)(x(y).\mathbf{0} \mid x(y).\mathbf{0}) \equiv (vx)(x(y).\mathbf{0}) \mid (vx)(x(y).\mathbf{0})$. It seems to be a matter of taste whether or not to accept law (2.3)′ as a structural law. Accepting

it, we conjecture that the results of this paper can still be shown, with an appropriate, but probably rather artificial, change of the multiset semantics of the small π-calculus (i.e., of the semantic relation $\Rightarrow$ given below).

In this paper we do not need to consider the transition system of the small π-calculus, as we are interested in structural congruence only. Similarly, we need not consider the transitions of the multiset transition system Mπ.

The elements of Mπ are *solutions*, which are multisets of *molecules*. A molecule is a pair $g.S$, where S is a solution and g is a schematic guard, i.e., a string of the form $x(-)$ or $\bar{x}y$ where x and y are names (in N), or new names (in New, with $\mathrm{New} \cap \mathrm{N} = \emptyset$), or positive natural numbers (in $\mathbf{N}$). For a guard $x(y)$ over $\mathrm{N} \cup \mathrm{New}$, we denote by $x(y).S$ the molecule $x(-).\mathrm{inc}(S)[1/y]$, where $\mathrm{inc}(S)$ denotes the result of increasing all natural numbers in S by one, and, in general, $S[u/v]$ denotes the result of substituting u for every occurrence of v in S.

For a solution (or molecule) S, $\mathrm{fn}(S)$ is the set of all names and new names in $\mathrm{N} \cup \mathrm{New}$ that occur in S, and $\mathrm{new}(S) = \mathrm{fn}(S) \cap \mathrm{New}$ is the set of new names that occur in S. For a solution S, $\mathrm{new}(S) = \bigcup_{m \in S} \mathrm{new}(m)$ and $\mathrm{new}(g.S) = \mathrm{new}(g) \cup \mathrm{new}(S)$, where $\mathrm{new}(x(-)) = \{x\} \cap \mathrm{New}$ and $\mathrm{new}(\bar{x}y) = \{x, y\} \cap \mathrm{New}$. A similar statement holds for 'fn'.

The semantic relation $P \Rightarrow S$ is defined between the process terms P of the small π-calculus and the solutions S of the multiset π-calculus Mπ. It is the smallest relation such that

(S0) $\mathbf{0} \Rightarrow \emptyset$

(S1) If $P_1 \Rightarrow S_1$ and $P_2 \Rightarrow S_2$, then $P_1 \mid P_2 \Rightarrow S_1 \cup S_2$
provided $\mathrm{new}(S_1) \cap \mathrm{new}(S_2) = \emptyset$

(S2) If $P \Rightarrow S$, then $(\nu x)P \Rightarrow S[n/x]$
provided $n \in \mathrm{New} - \mathrm{new}(S)$

(S3) If $P \Rightarrow S$ and g is a guard over N, then $g.P \Rightarrow \{g.S\}$

(S4) If $P \Rightarrow S_i$ for all $i \in \mathbf{N}$, then $!P \Rightarrow \bigcup_{i \in \mathbf{N}} S_i$
provided $\mathrm{new}(S_i) \cap \mathrm{new}(S_j) = \emptyset$ for all $i \neq j$.

Note that $S_1 \cup S_2$ and $\bigcup_{i \in \mathbf{N}} S_i$ are unions of multisets, with addition of multiplicities. Note also that if $P \Rightarrow S$ then $\mathrm{fn}(P) = \mathrm{fn}(S) \cap \mathrm{N}$.

Process terms P and Q are *multiset congruent*, denoted $P \equiv_m Q$, if P and Q have the same multiset semantics in Mπ, i.e., if $\{S : P \Rightarrow S\} = \{S : Q \Rightarrow S\}$. It was shown in Theorem B and Lemma 9 of [2] that structurally congruent processes are multiset congruent.

Lemma 1. *For process terms P and Q, if $P \equiv Q$, then $P \equiv_m Q$.*

The first main result of this paper is the other direction of this lemma: processes that are multiset congruent, are also structurally congruent. This proves that multiset congruence $\equiv_m$ and structural congruence $\equiv$ are the same. The second main result is the decidability of $\equiv$. These results were announced as statements (B′) and (C′), respectively, in the introduction of [2].

We will say that a solution S' is a *copy of* a solution S if there exists a bijection $f:\mathrm{new}(S)\to\mathrm{new}(S')$ such that $f(S)=S'$ (where $f(S)$ is the result of replacing every occurrence of a new name n by $f(n)$). As observed in [2] it is easy to show that 'copy of' is an equivalence relation. It is proved in Lemma 5 of [2] that for each process term P, the set $\{S:P\Rightarrow S\}$ is an equivalence class of this relation.

Lemma 2. *If $P\Rightarrow S$, then $P\Rightarrow S'$ if and only if S' is a copy of S.*

We will need the easy fact that the copy relation is preserved by taking the union of solutions with disjoint sets of new names. It was implicitly used in the proof of the above lemma in [2].

Lemma 3. *Let I be a countable index set, and let S_i and S_i', $i\in I$, be solutions such that the new(S_i) are mutually disjoint and the new(S_i') are mutually disjoint. If S_i' is a copy of S_i for every $i\in I$, then $\bigcup_{i\in I}S_i'$ is a copy of $\bigcup_{i\in I}S_i$.*

Proof. If $f_i(S_i)=S_i'$ for bijections $f_i:\mathrm{new}(S_i)\to\mathrm{new}(S_i')$, and f is defined such that its restriction to new(S_i) is f_i, then $f(\bigcup_{i\in I}S_i)=\bigcup_{i\in I}f(S_i)=\bigcup_{i\in I}S_i'$. Note that $\mathrm{new}(\bigcup_{i\in I}S_i)=\bigcup_{i\in I}\mathrm{new}(S_i)$ and similarly for the S_i'. □

3. Some basic properties of multisets

We will need some more basic properties of multisets. Since there does not seem to be a standard reference to such properties, we also discuss their proofs. This section may be skipped by the reader familiar with multisets.

Recall from [2] that a multiset S is a countable set D_S together with a mapping $\phi_S:D_S\to\mathbf{N}\cup\{\omega\}$ that defines the multiplicity of the elements of D_S in S (where $\mathbf{N}=\{1,2,3,\ldots\}$ and ω stands for countably infinite multiplicity). Union of multisets is defined in the obvious way, adding the multiplicities of each element (but note that in the literature this is often called the sum of multisets, in which case union is defined by taking the maximum of multiplicities).

We first explicitly state three, closely related, basic properties of multiset union that have already been used in [2]. Let I and J be countable index sets, let S_i be a multiset for each $i\in I$, and let $T_{i,j}$ be a multiset for each $i\in I$ and $j\in J$.

(a) Renaming the index set. If $\psi:J\to I$ is a bijection, then

$$\bigcup_{i\in I}S_i=\bigcup_{j\in J}S_{\psi(j)}.$$

(b) General commutativity and associativity.
If $I = \bigcup_{j\in J} I_j$ and the I_j are mutually disjoint, then

$$\bigcup_{i\in I} S_i = \bigcup_{j\in J}(\bigcup_{i\in I_j} S_i).$$

(c) Interchanging unions. $\bigcup_{i\in I}(\bigcup_{j\in J} T_{i,j}) = \bigcup_{j\in J}(\bigcup_{i\in I} T_{i,j})$.

Property (b) can be proved directly from the definition of multiset union, property (a) is a special case of (b), taking $I_j = \{\psi(j)\}$, and property (c) can easily be proved from the other two, by showing that both sides of the equation equal $\bigcup_{(i,j)\in I\times J} T_{i,j}$. We note that property (a) allows one to write a union $\bigcup_{i\in I} S_i$ with any index set J that has the same cardinality as I. In particular, J can always be taken disjoint with any other given set.

Next we consider a different, well-known, way of viewing a multiset, viz. as an indexed family of objects. Let D be a set, and let, for every i in some countable index set I, d_i be an element of D. Then, intuitively, the family $\{d_i\}_{i\in I}$ of elements of D represents the multiset $\bigcup_{i\in I}\{d_i\}$. Note that different families can represent the same multiset. In fact, intuitively, a multiset is an indexed family for which the identity of the indices in I is irrelevant. This leads us to investigating the properties of multiset unions of singleton sets. More formally, a family $\{d_i\}_{i\in I}$ is determined by the function $f: I \to D$ with $f(i) = d_i$. Thus, from now on we will consider multiset unions of the form $\bigcup_{i\in I}\{f(i)\}$ where f is such a function.

It follows from the definition of multiset union that $S = \bigcup_{i\in I}\{f(i)\}$ if and only if $D_S = f(I)$, the usual range of f, and $\phi_S(d) = \#f^{-1}(d)$, the cardinality of the set $f^{-1}(d) \subseteq I$ (where ω stands for $\aleph_0$). Thus, the multiplicity of d in S is the number of indices i with $f(i) = d$. From this it should be clear that every multiset S can be written as a union of singletons in at least one way: define $I \subseteq D_S \times \mathbf{N}$ by $I = \{(d,k): d \in D_S, 1 \leqslant k \leqslant \phi_S(d)\}$, where $k \leqslant \omega$ for every $k \in \mathbf{N}$, and define $f(d,k) = d$; then $S = \bigcup_{i\in I}\{f(i)\}$.

We now observe that two unions of singletons represent the same multiset if and only if they are obtained from each other by a renaming of the index set. In other words, for singleton multisets S_i the reverse of (a) is true.

Lemma 4. *$\bigcup_{i\in I}\{f(i)\} = \bigcup_{j\in J}\{g(j)\}$ if and only if there exists a bijection $\psi: J \to I$ such that $g(j) = f(\psi(j))$ for every $j \in J$.*

Proof. The if direction is a special case of (a). To show the only-if direction let $S = \bigcup_{i\in I}\{f(i)\} = \bigcup_{j\in J}\{g(j)\}$. Then, for every $d \in D_S$, $\#f^{-1}(d) = \#g^{-1}(d)$. Hence there is a bijection $\psi_d: g^{-1}(d) \to f^{-1}(d)$, for every $d \in D_S$. Then the function $\psi = \bigcup_{d\in D_S} \psi_d$ is a bijection from J to I such that $g(j) = f(\psi(j))$ for every $j \in J$. □

This lemma expresses the above-mentioned fact that a multiset is an indexed family for which the identity of the indices is irrelevant. This means that multisets over D are isomorphism classes of families of elements of D, where an isomorphism between two families is a mapping ψ as above.

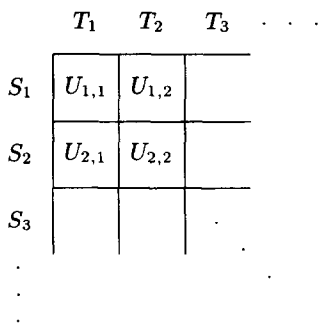

Fig. 1. Division of a multiset.

After characterizing the equality of two unions of singletons, we now characterize the equality of two unions of which one is a union of singletons. This turns out to be the reverse of property (b) for singleton multisets S_i.

Lemma 5. *$\bigcup_{i\in I}\{f(i)\}=\bigcup_{j\in J} T_j$ if and only if there exist mutually disjoint sets I_j such that $I=\bigcup_{j\in J} I_j$ and $T_j=\bigcup_{i\in I_j}\{f(i)\}$ for every $j\in J$.*

Proof. The if direction is a special case of (b). To show the only-if direction let $\bigcup_{i\in I}\{f(i)\}=\bigcup_{j\in J} T_j$. For each $j\in J$ we represent T_j as a union of singletons, with index set K_j. By (a) we may assume the K_j to be mutually disjoint. Hence $\bigcup_{j\in J} T_j=\bigcup_{j\in J}\bigcup_{k\in K_j}\{g(k)\}$ for some function g that is defined for every $k\in K=\bigcup_{j\in J} K_j$. Hence, by (b), $\bigcup_{j\in J} T_j=\bigcup_{k\in K}\{g(k)\}$ and so $\bigcup_{i\in I}\{f(i)\}=\bigcup_{k\in K}\{g(k)\}$. By Lemma 4 there is a bijection $\psi: K\to I$ such that $g(k)=f(\psi(k))$ for every $k\in K$. Hence $T_j=\bigcup_{k\in K_j}\{g(k)\}=\bigcup_{k\in K_j}\{f(\psi(k))\}$. And so, by (a), $T_j=\bigcup_{i\in I_j}\{f(i)\}$ where $I_j=\psi(K_j)$. □

Finally we characterize the equality of two arbitrary unions of multisets. This turns out to be the reverse of (c). Fig. 1 illustrates a multiset that can be viewed as a union in two ways.

Lemma 6. *$\bigcup_{i\in I} S_i=\bigcup_{j\in J} T_j$ if and only if there exist multisets $U_{i,j}$ such that $S_i=\bigcup_{j\in J} U_{i,j}$ and $T_j=\bigcup_{i\in I} U_{i,j}$ for every $i\in I$ and $j\in J$.*

Proof. The if direction is property (c). To show the only-if direction let $\bigcup_{i\in I} S_i=\bigcup_{j\in J} T_j=\bigcup_{k\in K}\{f(k)\}$. By Lemma 5 there exist mutually disjoint sets Y_i and mutually disjoint sets Z_j such that $K=\bigcup_{i\in I} Y_i=\bigcup_{j\in J} Z_j$, $S_i=\bigcup_{k\in Y_i}\{f(k)\}$ for every $i\in I$, and $T_j=\bigcup_{k\in Z_j}\{f(k)\}$ for every $j\in J$. Now let $U_{i,j}=\bigcup_{k\in Y_i\cap Z_j}\{f(k)\}$. Then $\bigcup_{j\in J} U_{i,j}=\bigcup_{j\in J}\bigcup_{k\in Y_i\cap Z_j}\{f(k)\}=\bigcup_{k\in Y_i}\{f(k)\}=S_i$ by property (b) because $Y_i=\bigcup_{j\in J}(Y_i\cap Z_j)$. A similar computation shows that $\bigcup_{i\in I} U_{i,j}=T_j$. □

Note that $U_{i,j}=S_i\cap T_j$ in the special case that the S_i are mutually disjoint sets, and similarly for the T_j (cf. the Venn-diagram in Fig. 1). However, in general the $U_{i,j}$

are not unique, as can be seen from the following trivial example. Let $I=J=\{1,2\}$ and let $S_i=T_j=\{a,b\}$ for all i and j. Then $S_1\cup S_2=T_1\cup T_2=\{a,a,b,b\}$. Now the requirements of the lemma hold for $U_{1,1}=U_{2,2}=\{a,b\}$ and $U_{1,2}=U_{2,1}=\emptyset$, but they also hold for $U_{1,1}=U_{2,2}=\{a\}$ and $U_{1,2}=U_{2,1}=\{b\}$.

4. Connected solutions

The main technical concept to be used in the proof of the main results is that of a connected solution. A solution S of Mπ is *connected* if there do not exist nonempty solutions S_1 and S_2 such that $S=S_1\cup S_2$ and $\mathrm{new}(S_1)\cap\mathrm{new}(S_2)=\emptyset$. Intuitively this means that all molecules of the solution are connected to each other through a chain of "relationships", where we say that two molecules m_1 and m_2 are "related" if $\mathrm{new}(m_1)\cap\mathrm{new}(m_2)\neq\emptyset$, i.e., if they both make use of at least one common local link. Recall that new names are always introduced as a result of restriction, which defines a local scope for a name.

Note that, trivially, any singleton solution $\{m\}$ is connected; the doubleton solution $\{m,m\}$ is connected if and only if $\mathrm{new}(m)\neq\emptyset$.

The notion of connectedness will be used as follows in the proof of the main results in Section 6. Roughly speaking, we will prove that $P\equiv_m Q$ implies $P\equiv Q$ by induction on the syntactical structure of P and Q. Consider the case that $P=!P'$ and $Q=!Q'$ and assume that $P\equiv_m Q$. Now we would like to prove that $P'\equiv_m Q'$, because then $P'\equiv Q'$ by induction, and hence $!P'\equiv !Q'$ by congruence. Since P and Q are multiset congruent, $\bigcup_{i\in\mathbf{N}} S_i=\bigcup_{j\in\mathbf{N}} T_j$, where the S_i are "disjoint" meanings of P' and the T_j are "disjoint" meanings of Q', in the sense that their sets of new names are disjoint. In general, by Lemma 6, this means that each S_i is cut into "disjoint" pieces by the T_j's, and vice versa. Thus, there is no relationship between the S_i and the T_j. If, however, we would know that the S_i and T_j are connected, in the above sense, then they could not be cut into non-trivial pieces, and hence we would be able to conclude that the S_i and T_j are equal (in fact, they are the "connected components" of the solution). This would then imply that $P'\equiv_m Q'$. To this aim, we will show in the next section that every process term is structurally congruent with one in which, roughly speaking, only connected solutions are replicated. In this section we investigate some fundamental properties of connected solutions and of the "connected components" of a solution.

Connectedness is preserved by taking copies (see the end of Section 2 for the notion of a 'copy' of a solution).

Lemma 7. *If S is a connected solution, and S' is a copy of S, then S' is a connected solution.*

Proof. By definition of 'copy', there is a bijection $h:\mathrm{new}(S')\to\mathrm{new}(S)$ such that $h(S')=S$. If S'_1 and S'_2 are nonempty solutions such that $S'=S'_1\cup S'_2$ and $\mathrm{new}(S'_1)\cap\mathrm{new}(S'_2)=\emptyset$, then $S_1=h(S'_1)$ and $S_2=h(S'_2)$ are nonempty solutions such that $S=h(S'_1\cup S'_2)=S_1\cup S_2$ and $\mathrm{new}(S_1)\cap\mathrm{new}(S_2)=h(\mathrm{new}(S'_1))\cap h(\mathrm{new}(S'_2))=h(\mathrm{new}(S'_1)\cap$

$\text{new}(S_2')) = \emptyset$, because $\text{new}(h(T)) = h(\text{new}(T))$ for every solution T, and h is a bijection. □

From this lemma and Lemma 2 it follows that the notion of connectedness can be carried over from solutions to process terms. If $P \Rightarrow S$ and S is connected, then we say that P is a *connected process term*. Note that every guarded term $g.P$ is connected because $g.P \Rightarrow \{g.S\}$ and $\{g.S\}$ is a singleton.

We will need a number of results that relate connectedness with multiset union. The first follows immediately from the definition of connectedness.

Lemma 8. *Let S be a nonempty connected solution. If $S = \bigcup_{i\in I} S_i$ and the new(S_i) are mutually disjoint, then there exists $j \in I$ such that $S_j = S$ and $S_i = \emptyset$ for $i \neq j$.*

We now compare two "disjoint" unions of solutions, one of which is a union of nonempty connected solutions. We obtain, for a particular case, the reverse of property (b) in Section 3 (see also Lemma 5).

Lemma 9. *Let S_i, $i \in I$, and T_j, $j \in J$, be solutions, such that the new(S_i) are mutually disjoint and the new(T_j) are mutually disjoint. Let, moreover, S_i be connected and nonempty. Then $\bigcup_{i\in I} S_i = \bigcup_{j\in J} T_j$ if and only if there exist mutually disjoint sets I_j such that $I = \bigcup_{j\in J} I_j$ and $T_j = \bigcup_{i\in I_j} S_i$ for every $j \in J$.*

Proof. The if direction is a special case of property (b) of Section 3. To show the only-if direction, assume that $\bigcup_{i\in I} S_i = \bigcup_{j\in J} T_j$. By Lemma 6 there exist solutions $U_{i,j}$ such that $S_i = \bigcup_{j\in J} U_{i,j}$ and $T_j = \bigcup_{i\in I} U_{i,j}$. This implies that $\text{new}(U_{i,j}) \subseteq \text{new}(S_i) \cap \text{new}(T_j)$, and hence the new$(U_{i,j})$ are mutually disjoint. Thus, Lemma 8 implies that for every i there exists j such that $S_i = U_{i,j}$ and $U_{i,k} = \emptyset$ for $k \neq j$. This means that S_i "belongs completely" to T_j. Note that the j is unique, because S_i is nonempty. Define, for $j \in J$, $I_j = \{i \in I : S_i = U_{i,j}\}$. Then $I = \bigcup_{j\in J} I_j$, the I_j are mutually disjoint, and $T_j = \bigcup_{i\in I} U_{i,j} = \bigcup_{i\in I_j} S_i$ for every $j \in J$. □

Next we compare two "disjoint" unions of nonempty connected solutions. We obtain, for a particular case, the reverse of property (a) in Section 3 (see also Lemma 4). Intuitively it means that there is essentially at most one way to divide a solution into "disjoint connected parts", as discussed in the beginning of this section.

Lemma 10. *Let S_i, $i \in I$, and T_j, $j \in J$, be nonempty connected solutions, such that the new(S_i) are mutually disjoint and the new(T_j) are mutually disjoint.*

Then $\bigcup_{i\in I} S_i = \bigcup_{j\in J} T_j$ if and only if there exists a bijection $\psi : J \to I$ such that $T_j = S_{\psi(j)}$ for every $j \in J$.

Proof. The if direction is by property (a) of Section 3. To show the only-if direction, assume that $\bigcup_{i\in I} S_i = \bigcup_{j\in J} T_j$. By Lemma 9, there exist mutually disjoint sets I_j such that $I = \bigcup_{j\in J} I_j$ and $T_j = \bigcup_{i\in I_j} S_i$ for every $j \in J$. Since T_j is connected and the S_i

are nonempty, Lemma 8 implies that I_j is a singleton. Define $\psi: J \to I$ such that $I_j = \{\psi(j)\}$. Clearly, ψ is a bijection and $T_j = S_{\psi(j)}$. □

We now show that every solution can be divided into "disjoint connected parts" in at least one way.

Lemma 11. *For every solution S there exist nonempty connected solutions S_i, $i \in I$, such that $S = \bigcup_{i \in I} S_i$ and the new(S_i) are mutually disjoint.*

Proof. Let $S = \bigcup_{k \in K} \{f(k)\}$, where $f: I \to \mathrm{Mol}$ is an indexed family of molecules, and consider the undirected graph $G = (V, E)$ such that $V = K$ and $E = \{(k, k') : \mathrm{new}(f(k)) \cap \mathrm{new}(f(k')) \neq \emptyset\}$. Thus, G models the "relationships" between the (indexed) molecules of S. Let $\{K_i : i \in I\}$ be the set of connected components of G, with $K = \bigcup_{i \in I} K_i$ and the K_i are mutually disjoint. Define $S_i = \bigcup_{k \in K_i} \{f(k)\}$. Clearly, the $\mathrm{new}(S_i) = \bigcup_{k \in K_i} \mathrm{new}(f(k))$ are mutually disjoint: if $k \in K_i$ and $k' \in K_j$, with $i \neq j$, then $\mathrm{new}(f(k)) \cap \mathrm{new}(f(k')) = \emptyset$, because k and k' belong to different connected components of G. Also, S_i is connected: if $S_i = S_{i,1} \cup S_{i,2}$ for nonempty solutions $S_{i,1}$ and $S_{i,2}$, then, by Lemma 5, there is a partition $K_{i,1}, K_{i,2}$ of K_i such that $S_{i,j} = \bigcup_{k \in K_{i,j}} \{f(k)\}$ for $j = 1, 2$; since K_i is a connected component, there is an edge (k_1, k_2) in G between $K_{i,1}$ and $K_{i,2}$, which implies that $\mathrm{new}(S_{i,1}) \cap \mathrm{new}(S_{i,2}) \neq \emptyset$. Finally, $\bigcup_{i \in I} S_i = \bigcup_{i \in I} \bigcup_{k \in K_i} \{f(k)\} = \bigcup_{k \in K} \{f(k)\} = S$ by property (b) of Section 3. □

Altogether we have shown in Lemmas 10 and 11 that there is essentially one way to divide a solution into "disjoint connected parts". This allows us to define the family of connected components of a solution, as follows. For a solution S, if $S = \bigcup_{i \in I} S_i$ for nonempty connected solutions S_i with mutually disjoint $\mathrm{new}(S_i)$, then we say that the S_i, $i \in I$, are the *connected components* of S. Note that if $S = \emptyset$ then $I = \emptyset$. Note also that for distinct i and j, D_{S_i} and D_{S_j} need not be disjoint; more precisely, if $m \in D_{S_i} \cap D_{S_j}$, then $\mathrm{new}(m) = \emptyset$, and hence $S_i = S_j = \{m\}$. Note finally that Lemma 9 can now be understood as follows: if $S = \bigcup_{j \in J} T_j$ and the $\mathrm{new}(T_j)$ are mutually disjoint, then each T_j is a union of connected components of S.

We will need some parameters of a solution S that are directly related to its connected components: the number of connected components of S, the multiplicity of a solution in the family of connected components of S, and the copy-width of S. These parameters have values in $\{0\} \cup \mathbf{N} \cup \{\omega\}$.

Let $S = \bigcup_{i \in I} S_i$ where the S_i are the connected components of S. It is an easy consequence of Lemmas 10 and 11 that the following definitions are valid. The *number of connected components of S* is $\mathrm{conn}(S) = \#I$, the cardinality of the index set I. For a solution S', the *multiplicity of S' in S* is $\mathrm{mult}(S', S) = \#\{i \in I : S_i \text{ is a copy of } S'\}$. Note that, by Lemma 7, $\mathrm{mult}(S', S)$ can only be non-zero if S' is connected. The 'mult' function counts the number of times that S' and its copies occur as connected component of S. The *copy-width* of S is $\mathrm{copy}(S) = \max\{\mathrm{mult}(S', S) : S' \in \mathrm{Sol}, \mathrm{mult}(S', S) \neq \omega\}$. Obviously, $\mathrm{copy}(S) = \max\{\mathrm{mult}(S_i, S) : i \in I, \mathrm{mult}(S_i, S) \neq \omega\}$. Thus, the copy-width of S is the maximal multiplicity of a connected component of S, where only finite

multiplicities are taken into account. Note that $\mathrm{copy}(S)=0$ if $\mathrm{mult}(S_i,S)=\omega$ for all $i\in I$, and that $\mathrm{copy}(S)=\omega$ if the numbers $\mathrm{mult}(S_i,S)$, $i\in I$, are unbounded. It will be shown in Lemma 22 that $\mathrm{copy}(S)\neq\omega$ for every solution S that is the semantics of a process term.

As an example, let P be a connected process term with $P\Rightarrow S'$ and $S'\neq\emptyset$. If $!P\Rightarrow S_1$, then $\mathrm{mult}(S',S_1)=\omega$ because, by Lemma 2 and (S4), S_1 has infinitely many connected components and each of them is a copy of S'. Note that $\mathrm{conn}(S_1)=\omega$ and $\mathrm{copy}(S_1)=0$. Now let Q be another connected process term with $Q\Rightarrow T$, such that $T\neq\emptyset$ and T is not a copy of S', and let $P\mid P\mid !Q\Rightarrow S_2$. Then $\mathrm{mult}(S',S_2)=2$, $\mathrm{mult}(T,S_2)=\omega$, $\mathrm{conn}(S_2)=\omega$, and $\mathrm{copy}(S_2)=2$.

The functions 'conn', 'mult', and 'copy' behave well with respect to multiset union.

Lemma 12. *If the $\mathrm{new}(S_i)$ are mutually disjoint, then*

$$\mathit{conn}\Big(\bigcup_{i\in I}S_i\Big)=\sum_{i\in I}\mathit{conn}(S_i),$$

$$\mathit{mult}\Big(S',\bigcup_{i\in I}S_i\Big)=\sum_{i\in I}\mathit{mult}(S',S_i),\ \textit{and}$$

$$\mathit{copy}\Big(\bigcup_{i\in I}S_i\Big)\leqslant\sum_{i\in I}\mathit{copy}(S_i).$$

Proof. Let $S=\bigcup_{i\in I}S_i$. Let $S_i=\bigcup_{j\in J_i}T_j$ where the T_j, $j\in J_i$, are the connected components of S_i. By renaming the index sets we may assume the J_i to be mutually disjoint. Then $S=\bigcup_{i\in I}\bigcup_{j\in J_i}T_j=\bigcup_{j\in J}T_j$ with $J=\bigcup_{i\in I}J_i$. Clearly, the T_j, $j\in J$, are the connected components of S. Hence $\mathrm{conn}(S)=\#J=\sum_{i\in I}\#J_i=\sum_{i\in I}\mathrm{conn}(S_i)$.

Similarly, for any solution S', $\mathrm{mult}(S',S)=\#\{j\in J: T_j \text{ is a copy of } S'\}=\sum_{i\in I}\#\{j\in J_i: T_j \text{ is a copy of } S'\}=\sum_{i\in I}\mathrm{mult}(S',S_i)$.

Consequently, if $\mathrm{mult}(S',S)\neq\omega$, then $\mathrm{mult}(S',S_i)\neq\omega$ for all $i\in I$. And so $\mathrm{copy}(S)\leqslant\max\{\sum_{i\in I}\mathrm{mult}(S',S_i): S'\in\mathrm{Sol},\mathrm{mult}(S',S_i)\neq\omega\}\leqslant\sum_{i\in I}\max\{\mathrm{mult}(S',S_i): S'\in\mathrm{Sol},\mathrm{mult}(S',S_i)\neq\omega\}=\sum_{i\in I}\mathrm{copy}(S_i)$. □

We have seen that the notion of connectedness can be carried over from solutions to process terms. The next lemma is needed to show that the functions 'conn' and 'copy' can be carried over in the same way.

Lemma 13. *If S' is a copy of S, then $\mathit{conn}(S')=\mathit{conn}(S)$ and $\mathit{copy}(S')=\mathit{copy}(S)$.*

Proof. Let $f(S)=S'$ for a bijection $f:\mathrm{new}(S)\to\mathrm{new}(S')$, and let $S=\bigcup_{i\in I}S_i$ where the S_i are the connected components of S. Then $S'=f(\bigcup_{i\in I}S_i)=\bigcup_{i\in I}f(S_i)$. By Lemma 7, $f(S_i)$ is connected. Since $\mathrm{new}(S)=\bigcup_{i\in I}\mathrm{new}(S_i)$, f is a bijection between $\mathrm{new}(S_i)$ and $\mathrm{new}(f(S_i))$. Hence the $f(S_i)$ are the connected components of S', and so $\mathrm{conn}(S')=\#I=\mathrm{conn}(S)$. Now note that $f(S_i)$ is a copy of S_i for every $i\in I$. This implies that, for any solution T, $\mathrm{mult}(T,S)=\#\{i\in I: S_i \text{ is a copy of } T\}=\#\{i\in I: f(S_i) \text{ is a copy of } T\}=\mathrm{mult}(T,S')$. Hence $\mathrm{copy}(S')=\mathrm{copy}(S)$. □

If $P \Rightarrow S$ then we define the *number of connected components* of P to be $\mathrm{conn}(P) = \mathrm{conn}(S)$, and the *copy-width* of P to be $\mathrm{copy}(P) = \mathrm{copy}(S)$.

In the next lemma we show that if $\mathrm{mult}(S', S) = \omega$, then arbitrary copies of S' can be added to S, resulting in a copy of S. This is similar to structural laws (3.1) and (3.5).

Lemma 14. *Let S and T be solutions with $new(S) \cap new(T) = \emptyset$. Let $T = \bigcup_{k \in K} S_k$ where the S_k, $k \in K$, are the connected components of T.*

If $mult(S_k, S) = \omega$ for every $k \in K$, then $S \cup T$ is a copy of S.

Proof. Let $S = \bigcup_{i \in I} S_i$ with $I \cap K = \emptyset$ and the S_i, $i \in I$, are the connected components of S.

We first consider the easy case that all S_k and S_i are copies of each other. Since $\mathrm{mult}(S_k, S) = \omega$ for $k \in K$, $\#I = \omega$. Hence there is a bijection $\psi : I \to I \cup K$. Since, for every $i \in I$, $S_{\psi(i)}$ is a copy of S_i (and all $\mathrm{new}(S_i)$ and $\mathrm{new}(S_k)$ are disjoint), Lemma 3 implies that $\bigcup_{i \in I} S_{\psi(i)}$ is a copy of $\bigcup_{i \in I} S_i$. Since $\bigcup_{i \in I} S_{\psi(i)} = \bigcup_{j \in I \cup K} S_j = \bigcup_{i \in I} S_i \cup \bigcup_{k \in K} S_k = S \cup T$ by properties (a) and (b) of Section 3, respectively, $S \cup T$ is a copy of S.

The general case is just the obvious simultaneous combination of any number of applications of the easy case. The 'copy of' relation induces a partition of K into equivalence classes, where k and k' are equivalent iff S_k is a copy of $S_{k'}$. Thus $K = \bigcup_{j \in J} K_j$, with mutually disjoint K_j, and S_k is a copy of $S_{k'}$ iff k and k' are in the same K_j. Similarly for S, $I = \bigcup_{l \in L} I_l$, with mutually disjoint I_l, and S_i is a copy of $S_{i'}$ iff i and i' are in the same I_l. Since, for every $k \in K$, $\mathrm{mult}(S_k, S) = \omega$, we may assume (by renaming the index set) that $J \subseteq L$ and, for every $j \in J$, $\#I_j = \omega$ and S_k is a copy of S_i for every $k \in K_j$ and $i \in I_j$. By the easy case considered above, for every $j \in J$, $\bigcup_{i \in I_j} S_i$ is a copy of $\bigcup_{i \in I_j} S_i \cup \bigcup_{k \in K_j} S_k$. Hence, by Lemma 3 $\bigcup_{j \in J} \bigcup_{i \in I_j} S_i$ is a copy of $\bigcup_{j \in J} \bigcup_{i \in I_j} S_i \cup \bigcup_{j \in J} \bigcup_{k \in K_j} S_k$ and $\bigcup_{l \in L} \bigcup_{i \in I_l} S_i$ is a copy of $\bigcup_{l \in L} \bigcup_{i \in I_l} S_i \cup \bigcup_{j \in J} \bigcup_{k \in K_j} S_k$, i.e., S is a copy of $S \cup T$. □

This lemma is used in the next one, which can be viewed as a strengthening of Lemma 6 for solutions with disjoint sets of new names, in the case that $I = J = \{1, 2\}$. We show that there exist $U_{i,j}$ that have at most the same copy-width as the given solutions S_i and T_j. However, the $U_{i,j}$ only add up to copies of the S_i and T_j. Again, Fig. 1 illustrates the situation.

Lemma 15. *Let $m \in \mathbf{N}$, and let S_i and T_j, with $i, j \in \{1, 2\}$, be solutions such that $S_1 \cup S_2 = T_1 \cup T_2$, with $new(S_1) \cap new(S_2) = \emptyset$ and $new(T_1) \cap new(T_2) = \emptyset$. If $copy(S_i) \leqslant m$ and $copy(T_j) \leqslant m$, for all i and j, then there exist four solutions $U_{i,j}$ such that $copy(U_{i,j}) \leqslant m$, $U_{i,1} \cup U_{i,2}$ is a copy of S_i, and $U_{1,j} \cup U_{2,j}$ is a copy of T_j.*

Proof. It follows from Lemma 6 (with $I = J = \{1, 2\}$) that solutions $U'_{i,j}$ exist such that $U'_{i,1} \cup U'_{i,2} = S_i$ and $U'_{1,j} \cup U'_{2,j} = T_j$. Note that, as in the proof of Lemma 9, the $\mathrm{new}(U'_{i,j})$ are mutually disjoint. Let $U'_{i,j} = \bigcup_{k \in K_{i,j}} V_k$ for mutually disjoint index sets $K_{i,j}$, where the V_k, $k \in K_{i,j}$, are the connected components of $U'_{i,j}$.

It need not be true that $\text{copy}(U'_{i,j}) \leqslant m$. To reach this goal, the idea is just to drop the "wrong" connected components from $U'_{i,j}$, i.e., the components V_k with $\text{mult}(V_k, U'_{i,j}) > m$ and $\text{mult}(V_k, U'_{i,j}) \neq \omega$. Thus, let $L_{i,j} = \{k \in K_{i,j} : m < \text{mult}(V_k, U'_{i,j}) < \omega\}$, $W_{i,j} = \bigcup_{k \in L_{i,j}} V_k$, and $U_{i,j} = \bigcup_{k \in K_{i,j} - L_{i,j}} V_k$ (and so $U'_{i,j} = U_{i,j} \cup W_{i,j}$). We now claim that the $U_{i,j}$ satisfy the requirements of the lemma. Obviously, by definition, $\text{copy}(U_{i,j}) \leqslant m$. It remains to show that $U_{i,1} \cup U_{i,2}$ is a copy of S_i and that $U_{1,j} \cup U_{2,j}$ is a copy of T_j. We only prove the first statement; the proof of the second statement is symmetrical.

Consider an arbitrary $k \in L_{i,j}$, i.e., $m < \text{mult}(V_k, U'_{i,j}) < \omega$. We claim that $\text{mult}(V_k, U'_{i,3-j}) = \omega$. To see this, note that $U'_{i,j} \cup U'_{i,3-j} = S_i$ and hence, by Lemma 12, $\text{mult}(V_k, S_i) = \text{mult}(V_k, U'_{i,j}) + \text{mult}(V_k, U'_{i,3-j})$. Thus, it would follow from $\text{mult}(V_k, U'_{i,3-j}) < \omega$ that $m < \text{mult}(V_k, S_i) < \omega$, which contradicts the fact that $\text{copy}(S_i) \leqslant m$. Hence $\text{mult}(V_k, U'_{i,3-j}) = \omega$, which, by definition of $U_{i,3-j}$ and $U_{3-i,j}$, implies that $\text{mult}(V_k, U_{i,3-j}) = \omega$. Hence, we have shown that $\text{mult}(V_k, U_{i,3-j}) = \omega$ for every connected component V_k of $W_{i,j}$.

Now $S_i = U'_{i,1} \cup U'_{i,2} = (U_{i,1} \cup W_{i,1}) \cup (U_{i,2} \cup W_{i,2}) = (U_{i,1} \cup W_{i,2}) \cup (U_{i,2} \cup W_{i,1})$. By the above, Lemma 14 is applicable and gives that $U_{i,1} \cup W_{i,2}$ is a copy of $U_{i,1}$ and $U_{i,2} \cup W_{i,1}$ is a copy of $U_{i,2}$. Hence, by Lemma 3, S_i is a copy of $U_{i,1} \cup U_{i,2}$. □

We end this section by giving a condition ensuring that multiset union preserves connectedness.

Lemma 16. *Let $S = \bigcup_{i \in I} S_i$. If each S_i is connected, and $\text{new}(S_i) \cap \text{new}(S_j) \neq \emptyset$ for all $i \neq j$, then S is connected.*

Proof. If $S = T_1 \cup T_2$ for nonempty solutions T_1 and T_2 then, by Lemma 6, there are solutions $U_{i,j}$ such that $S_i = U_{i,1} \cup U_{i,2}$ for $i \in I$, and $T_j = \bigcup_{i \in I} U_{i,j}$ for $j = 1, 2$. If there exists $i \in I$ such that both $U_{i,1}$ and $U_{i,2}$ are nonempty, then, by connectedness of S_i, $\text{new}(U_{i,1}) \cap \text{new}(U_{i,2}) \neq \emptyset$ and so $\text{new}(T_1) \cap \text{new}(T_2) \neq \emptyset$. Otherwise, there is a partition I_1, I_2 of I such that $T_j = \bigcup_{i \in I_j} S_i$ for $j = 1, 2$. For $i_1 \in I_1$ and $i_2 \in I_2$, $\text{new}(S_{i_1}) \cap \text{new}(S_{i_2}) \neq \emptyset$ implies that $\text{new}(T_1) \cap \text{new}(T_2) \neq \emptyset$. □

5. Connected process terms

We now turn to properties of connected process terms. First, we use the last lemma of the previous section to show that connectedness of process terms is preserved under certain conditions.

We will write $P_1 \mid P_2 \mid \cdots \mid P_k$ for any process term that is obtained from the process term $(\cdots((P_1 \mid P_2) \mid P_3) \mid \cdots \mid P_{k-1}) \mid P_k$ by structural law (1.3), i.e., by associativity of parallel composition.

Lemma 17. *If, for every $1 \leqslant i \leqslant k$, $x \in fn(P_i)$ and either P_i is connected or $P_i = !Q_i$ for some connected Q_i, then $(\nu x)(P_1 \mid \cdots \mid P_k)$ is connected.*

Proof. By repeated use of (S1) and (S4), and the use of (a) and (b) in Section 3, we obtain that $P_1 \mid \cdots \mid P_k \Rightarrow \bigcup_{j\in J} S_j$ where for each j there is a connected process term R_j such that $x \in \mathrm{fn}(R_j)$ and $R_j \Rightarrow S_j$ (clearly, $R_j = P_i$ or $R_j = Q_i$ for some i). Hence, by (S2), $(vx)(P_1 \mid \cdots \mid P_k) \Rightarrow (\bigcup_{j\in J} S_j)[n/x] = \bigcup_{j\in J} (S_j[n/x])$ for some $n \in \mathrm{New}$. We have to show that $\bigcup_{j\in J} (S_j[n/x])$ is connected. Since $x \in \mathrm{fn}(R_j)$, $x \in \mathrm{fn}(S_j)$ and so $n \in \mathrm{new}(S_j[n/x])$ for every j. Thus, by Lemma 16, it now suffices to show that $S_j[n/x]$ is connected for every j. Since R_j is connected, we know that S_j is connected. Observe now that, in general, if S is connected, then so is $S[n/x]$. In fact, let $S = \bigcup_{i\in I}\{f(i)\}$. Then $S[n/x] = \bigcup_{i\in I}\{f(i)[n/x]\}$. Suppose that $S[n/x] = S_1 \cup S_2$ for nonempty solutions S_1 and S_2. Then, by Lemma 5, there is a partition I_1, I_2 of I such that $S_j = \bigcup_{i\in I_j}\{f(i)[n/x]\}$ for $j = 1,2$. Then $S = S_1' \cup S_2'$ with $S_j' = \bigcup_{i\in I_j}\{f(i)\}$. Since S is connected, there are $i_1 \in I_1$ and $i_2 \in I_2$ such that $\mathrm{new}(f(i_1)) \cap \mathrm{new}(f(i_2)) \neq \emptyset$. Hence, since, for any molecule m, $\mathrm{new}(m) \subseteq \mathrm{new}(m[n/x])$, $\mathrm{new}(f(i_1)[n/x]) \cap \mathrm{new}(f(i_2)[n/x]) \neq \emptyset$, and so $\mathrm{new}(S_1) \cap \mathrm{new}(S_2) \neq \emptyset$. □

The next result gives an important normal form for process terms. We show, as discussed in the beginning of the previous section, that for every process term there is a structurally congruent one in which only connected subprocesses are replicated. We also need the natural property that all its restricted subprocesses are connected. Additionally, to avoid empty subprocesses, we remove $\mathbf{0}$ as much as possible.

We say that a process term P is *subconnected* if (i) Q is connected for every subterm $!Q$ of P, (ii) each subterm $(vx)Q$ of P is connected, and (iii) P does not contain subterms of the form $\mathbf{0} \mid Q$, $Q \mid \mathbf{0}$, $(vx)\mathbf{0}$, or $!\mathbf{0}$. Note that connectedness and subconnectedness are incomparable properties.

Lemma 18. *For every process term P, a subconnected process term P' can be computed such that $P \equiv P'$.*

Proof. We compute P' by induction on the syntactical structure of P. The cases $P = \mathbf{0}$ and $P = g.P_1$ are easy. For $P = \mathbf{0}$, $P' = \mathbf{0}$, and for $P = g.P_1$, $P' = g.P_1'$ where, by induction, P_1' is a subconnected term with $P_1 \equiv P_1'$.

Let $P = P_1 \mid P_2$. By induction, subconnected P_1' and P_2' have been computed such that $P_1 \equiv P_1'$ and $P_2 \equiv P_2'$. Then $P = P_1 \mid P_2 \equiv P_1' \mid P_2'$. Take $P' = P_1' \mid P_2'$ if both P_1' and P_2' are non-zero, $P' = P_1'$ if $P_2' = \mathbf{0}$, and $P' = P_2'$ if $P_1' = \mathbf{0}$, and use structural laws (1.1) and (1.2).

Let $P = !P_1$. By induction a subconnected P_1' has been computed such that $P_1 \equiv P_1'$. Since $P \equiv !P_1'$, it now suffices to compute a subconnected process term P' that is structurally congruent with $!P_1'$. If $P_1' = \mathbf{0}$ then take $P' = \mathbf{0}$, using structural law (3.4). Otherwise $P_1' \equiv Q_1 \mid \cdots \mid Q_m \mid !R_1 \mid \cdots \mid !R_n$ where the Q_i are not parallel compositions and not replications (any non-zero term can be written in this form, using structural laws (1.2) and (1.3) only). Then every Q_i is connected, because it is either a guarded term (which is always connected) or a restriction (which is connected because P_1' is subconnected). Now $!P_1' \equiv !(Q_1 \mid \cdots \mid Q_m \mid !R_1 \mid \cdots \mid !R_n) \equiv !Q_1 \mid \cdots \mid !Q_m \mid !!R_1 \mid \cdots \mid !!R_n \equiv$

$!Q_1 \mid \cdots \mid !Q_m \mid !R_1 \mid \cdots \mid !R_n = P'$ by structural laws (3.2) and (3.3), respectively. Obviously, since P_1' is subconnected and the Q_i are connected and non-zero, P' is subconnected.

Let $P = (\nu x)P_1$. As in the previous case, P_1 is structurally congruent with a subconnected P_1' and it suffices to compute a subconnected process term P' that is structurally congruent with $(\nu x)P_1'$. If $P_1' = \mathbf{0}$ then take $P' = \mathbf{0}$, and use structural law (2.2). Otherwise $P_1' \equiv Q_1 \mid \cdots \mid Q_m \mid R_1 \mid \cdots \mid R_n$ where the Q_i and R_j are not parallel compositions, $x \notin \mathrm{fn}(Q_i)$, and $x \in \mathrm{fn}(R_j)$. Then $(\nu x)P_1' \equiv (\nu x)(Q_1 \mid \cdots \mid Q_m \mid R_1 \mid \cdots \mid R_n) \equiv Q_1 \mid \cdots \mid Q_m \mid (\nu x)(R_1 \mid \cdots \mid R_n) = P'$ by structural law (2.2), if $n = 0$, or (2.3), if $n > 0$. By an argument similar to the one in the previous case, every R_j is either connected or the replication of a connected term. Hence $(\nu x)(R_1 \mid \cdots \mid R_n)$ is connected by Lemma 17. Together with the fact that P_1' is subconnected, this shows that P' is subconnected. □

As an example, let $P(x, y) = \bar{x}y.\mathbf{0}$, $Q(x) = x(u).\bar{u}x.\mathbf{0}$, $R(y) = y(u).\mathbf{0}$, and $R'(y) = y(u).(!(\nu x)\mathbf{0} \mid !!\mathbf{0})$. Then $R(y)$ is subconnected and $R(y) \equiv R'(y)$. Consider the process term $P = (\nu x)!(\nu y)(!(P(x, y) \mid Q(x)) \mid R'(y))$. A subconnected process term P' with $P' \equiv P$ is computed as follows:

$$\begin{aligned} P &\equiv (\nu x)!(\nu y)(!Q(x) \mid !P(x, y) \mid R(y)) \\ &\equiv (\nu x)!(!Q(x) \mid (\nu y)(!P(x, y) \mid R(y))) \\ &\equiv (\nu x)(!Q(x) \mid !(\nu y)(!P(x, y) \mid R(y))) = P'. \end{aligned}$$

Note that the subconnected normal form is not unique, i.e., there exist distinct subconnected process terms that are structurally congruent, e.g., the terms $(\nu x)(Q(x) \mid (\nu y)(P(x, y) \mid R(y)))$ and $(\nu y)(R(y) \mid (\nu x)(P(x, y) \mid Q(x)))$.

By a similar argument as in the above proof of Lemma 18, we obtain the following corollary.

Corollary 19. *For every process term P, a process term P' can be computed such that $P \equiv P'$ and either $P' = \mathbf{0}$ or $P' = P_1 \mid \cdots \mid P_n \mid !P_{n+1} \mid \cdots \mid !P_{n+k}$ with $n, k \geqslant 0$, $n + k \geqslant 1$, P_i is connected, subconnected, and non-zero.*

Note that, as can be seen from the proof of Lemma 18, in a subconnected process P both the replications and the restrictions are nested as deeply as possible in P. Moreover, either $P = \mathbf{0}$ or every $\mathbf{0}$ occurs in a guarded subterm $g.\mathbf{0}$. This implies that the meaning of a non-zero subconnected process term is nonempty, as formally shown in the next lemma.

Lemma 20. *For a subconnected P, $P \Rightarrow \emptyset$ if and only if $P = \mathbf{0}$.*

Proof. The proof of the only-if direction is by induction on the syntactical structure of P. For $P = \mathbf{0}$ it is trivial.

Let $P = Q_1 \mid Q_2$. By (S1), there are solutions T_1 and T_2 such that $Q_1 \Rightarrow T_1$, $Q_2 \Rightarrow T_2$, and $T_1 \cup T_2 = \emptyset$. Then $T_1 = \emptyset$ and $T_2 = \emptyset$. Hence, by induction, $Q_1 = \mathbf{0}$ and $Q_2 = \mathbf{0}$, and so $P = \mathbf{0} \mid \mathbf{0}$. This contradicts the fact that P is subconnected. Hence this case cannot occur.

Let $P=(\nu x)Q$. By (S2), $Q \Rightarrow T$ and $T[n/x]=\emptyset$. Then $T=\emptyset$. By induction, $Q=\mathbf{0}$, and so $P=(\nu x)Q=(\nu x)\mathbf{0}$. This contradicts again the fact that P is subconnected.

Let $P=g.Q$. This case cannot occur because, by (S3), $g.Q \Rightarrow \{g.T\}$ for some T, but $\{g.T\} \neq \emptyset$.

Let $P=!Q$. By (S4), there are solutions T_i such that $Q \Rightarrow T_i$ and $\bigcup_{i\in\mathbf{N}} T_i=\emptyset$. Then $T_i=\emptyset$ for all i. By induction $Q=\mathbf{0}$, and so $P=!Q=!\mathbf{0}$, contradicting the subconnectedness of P. □

Finally we will show that for each process term P we can compute $\mathrm{conn}(P)$ and we can compute an upper bound for $\mathrm{copy}(P)$. To do this we need the following properties of $\mathrm{conn}(P)$ and $\mathrm{copy}(P)$, which easily follow from Lemma 12, (S1), (S4), and Lemma 2.

Lemma 21. *Let P be a connected process term such that $P \not\equiv \mathbf{0}$, and let P_1 and P_2 be arbitrary process terms. Then*

(1) $conn(P)=copy(P)=1$,

(2) $conn(P_1 \mid P_2)=conn(P_1)+conn(P_2)$,
$copy(P_1 \mid P_2) \leqslant copy(P_1)+copy(P_2)$, *and*

(3) $conn(!P)=\omega$ *and* $copy(!P)=0$.

Lemma 22. *For every process term P, $conn(P) \in \{0\} \cup \mathbf{N} \cup \{\omega\}$ can be computed, and a number $c(P) \in \{0\} \cup \mathbf{N}$ can be computed such that $copy(P) \leqslant c(P)$.*

Proof. For given P, first compute a process term P' such that $P' \equiv P$, as in Corollary 19. Obviously, by Lemma 1, $\mathrm{conn}(P)=\mathrm{conn}(P')$ and $\mathrm{copy}(P)=\mathrm{copy}(P')$. If $P'=\mathbf{0}$, then $\mathrm{conn}(P')=0$ and $\mathrm{copy}(P')=0$ (i.e., we can take $c(P)=0$). Otherwise $P'=P_1 \mid \cdots \mid P_n \mid !P_{n+1} \mid \cdots \mid !P_{n+k}$ where the P_i are connected, subconnected, and nonzero. By Lemma 20, $P_i \not\equiv \mathbf{0}$. Hence, by Lemma 21 (1,3), $\mathrm{conn}(P_i)=\mathrm{copy}(P_i)=1$, $\mathrm{conn}(!P_i)=\omega$, and $\mathrm{copy}(!P_i)=0$. Consequently, by Lemma 21 (2), if $k \geqslant 1$ then $\mathrm{conn}(P')=\omega$, and if $k=0$ then $\mathrm{conn}(P')=n$. Moreover, $\mathrm{copy}(P') \leqslant n$ (i.e., we can take $c(P)=n$). □

Note that this implies that it is decidable whether or not a process term P is connected (viz., if $\mathrm{conn}(P) \leqslant 1$). Hence, subconnectedness of P is also decidable.

We finally note that if $P \equiv P'=P_1 \mid \cdots \mid P_n \mid !P_{n+1} \mid \cdots \mid !P_{n+k}$ as in Corollary 19, then the P_i represent the connected components of P, in the sense that if $P \Rightarrow S$ and $P_i \Rightarrow S_i$, then every connected component of S is a copy of some S_i (cf. the proof of Lemma 17).

6. The main results

In this section we prove the two main results. The proof of the first main result – if $P \equiv_m Q$, then $P \equiv Q$ – is by induction on the syntactical structure of Q. We will

show that, for given P and $Q \neq \mathbf{0}$, there exist terms P_i and Q_i, $1 \leqslant i \leqslant n$, and a boolean function f of n boolean arguments, such that

(I1) if $P \equiv_m Q$, then $f(P_1 \equiv_m Q_1, \ldots, P_n \equiv_m Q_n)$,
(I2) if $f(P_1 \equiv Q_1, \ldots, P_n \equiv Q_n)$, then $P \equiv Q$, and
(I3) the Q_i are direct subterms of Q.

Together with the proof for the case that $Q = \mathbf{0}$, this clearly proves the first main result. Moreover, together with the decidability of $P \equiv \mathbf{0}$, it also shows that $\equiv$ is decidable. This is because, in fact, the P_i, Q_i, and f can be effectively constructed from P and Q. Note that, after proving the first main result, we know that $\equiv_m$ and $\equiv$ are equal, and hence statements (I1) and (I2) turn into the characterization

(I4) $P \equiv Q$ if and only if $f(P_1 \equiv Q_1, \ldots, P_n \equiv Q_n)$.

Thus the truth value of $P \equiv Q$ can be computed by a recursive boolean function procedure with two arguments P and Q, of which the body contains finitely many recursive calls. Since the second argument of each recursive call is smaller than Q, the function procedure always halts.

We have chosen to present the proofs of the two main results simultaneously, in the above way. Without the decidability of $\equiv$, the proof of the first main result can be simplified by allowing the boolean functions to have infinitely many arguments, and, in fact, we first had such a proof. The notion of copy-width has been introduced explicitly to deal with decidability, and in particular to produce the finitary versions of Lemmas 28 and 29 as presented below.

We start with the basis of the inductive proof described above: the case that $Q = \mathbf{0}$. This follows rather directly from Lemmas 18 and 20.

Lemma 23. (1) $P \equiv_m \mathbf{0}$ *if and only if* $P \equiv \mathbf{0}$.
(2) *It is decidable, for a process term* P, *whether or not* $P \equiv \mathbf{0}$.

Proof. (1) The if-direction is by Lemma 1. Now assume that $P \equiv_m \mathbf{0}$. By Lemma 18 there is a subconnected P' such that $P' \equiv P$. By Lemma 1, $P' \equiv_m P$ and so $P' \equiv_m \mathbf{0}$. Now Lemma 20 implies that $P' = \mathbf{0}$ and hence $P \equiv \mathbf{0}$.

(2) By Lemma 18, a subconnected P' can be computed such that $P' \equiv P$. Thus, $P \equiv \mathbf{0}$ if and only if $P' \equiv \mathbf{0}$, and, by (1) and Lemma 20, $P' \equiv \mathbf{0}$ if and only if $P' = \mathbf{0}$. □

The induction step for the case that $Q = (vx)Q'$ is shown in the next lemma. We prove that if $P \equiv_m (vx)Q'$, then there exists a P' such that $P \equiv (vx)P'$ and $P' \equiv_m Q'$. Moreover, we compute (from P and x only) a finite set of possible P'.

Lemma 24. *For every process term* P *and every name* x *a finite set* $res(P,x)$ *of process terms can be computed such that*
(1) $P \equiv (vx)P'$ *for every* $P' \in res(P,x)$, *and*
(2) *if* $P \Rightarrow S[n/x]$, *with* $n \in New - new(S)$, *then there exists* $P' \in res(P,x)$ *such that* $P' \Rightarrow S$.

Proof. In the case that $x \in \mathrm{fn}(P)$, we define $\mathrm{res}(P,x) = \emptyset$. In fact, if $P \Rightarrow S[n/x]$ then $x \notin \mathrm{fn}(P)$. In what follows we assume that $x \notin \mathrm{fn}(P)$.

Intuitively, the set $\mathrm{res}(P,x)$ consists of all P' that are obtained from P by "moving a restriction outermost", as explained in [4, 6] (but note that in [6] only unguarded restrictions could be moved outermost, due to the omission of structural law (2.4)). The formal definition of $\mathrm{res}(P,x)$ will be inductive.

We first observe that it suffices to prove the statement of the lemma for the case that x does not occur at all in P (neither free nor bound). In fact, using α-conversion (i.e., structural law (α)) to rename all bound occurrences of x in P, a term $\overline{P}$ can be constructed such that $P \equiv \overline{P}$. We now define $\mathrm{res}(P,x)$ to be $\mathrm{res}(\overline{P},x)$. Note that by Lemma 1, $P \equiv_m \overline{P}$; hence $P \Rightarrow S[n/x]$ implies $\overline{P} \Rightarrow S[n/x]$.

The computation, and its correctness, for process terms P in which x does not occur, is by induction on the syntactical structure of P.

Let $P = \mathbf{0}$. Define $\mathrm{res}(\mathbf{0},x) = \{\mathbf{0}\}$. Then $\mathbf{0} \equiv (vx)\mathbf{0}$ by structural law (2.2). Moreover, if $P \Rightarrow S[n/x]$, then $S[n/x] = \emptyset$ and so $S = \emptyset$. Hence there exists $P' \in \mathrm{res}(\mathbf{0},x)$ such that $P' \Rightarrow S$.

Let $P = Q_1 \mid Q_2$. Define $\mathrm{res}(Q_1 \mid Q_2, x) = \{Q_1' \mid Q_2 \colon Q_1' \in \mathrm{res}(Q_1,x)\} \cup \{Q_1 \mid Q_2' \colon Q_2' \in \mathrm{res}(Q_2,x)\}$. To show (1), consider some $P' = Q_1' \mid Q_2 \in \mathrm{res}(Q_1 \mid Q_2, x)$, with $Q_1' \in \mathrm{res}(Q_1,x)$. By the induction hypothesis for Q_1, $Q_1 \equiv (vx)Q_1'$. Hence $P = Q_1 \mid Q_2 \equiv (vx)Q_1' \mid Q_2 \equiv (vx)(Q_1' \mid Q_2) \equiv (vx)P'$ by structural laws (1.2) and (2.3), because $x \notin \mathrm{fn}(Q_2)$. The proof for $Q_1 \mid Q_2' \in \mathrm{res}(Q_1 \mid Q_2, x)$ is symmetric. To show (2), assume that $P \Rightarrow S[n/x]$. Then $Q_i \Rightarrow T_i$ with disjoint $\mathrm{new}(T_i)$, and $S[n/x] = T_1 \cup T_2$. Note that x does not occur in Q_i and so $x \notin \mathrm{fn}(T_i)$. Since the $\mathrm{new}(T_i)$ are disjoint, $n \notin \mathrm{new}(T_1)$ or $n \notin \mathrm{new}(T_2)$. Assume that $n \notin \mathrm{new}(T_1)$; the other case is symmetric. Since $n \notin \mathrm{new}(S)$, $S = S[n/x][x/n] = (T_1 \cup T_2)[x/n] = T_1[x/n] \cup T_2[x/n] = T_1 \cup T_2[x/n]$. Also $T_2 = T_2[x/n][n/x]$, because $x \notin \mathrm{fn}(T_2)$. Hence, by the induction hypothesis for Q_2, there exists $Q_2' \in \mathrm{res}(Q_2,x)$ such that $Q_2' \Rightarrow T_2[x/n]$. Take $P' = Q_1 \mid Q_2' \in \mathrm{res}(P,x)$. Then $P' = Q_1 \mid Q_2' \Rightarrow T_1 \cup T_2[x/n] = S$.

Let $P = (vy)Q$. Define $\mathrm{res}((vy)Q,x) = \{Q[x/y]\} \cup \{(vy)Q' \colon Q' \in \mathrm{res}(Q,x)\}$. To show (1), consider first $P' = Q[x/y] \in \mathrm{res}((vy)Q,x)$. Then, $P = (vy)Q \equiv (vx)Q[x/y] = (vx)P'$ by structural law (α), because x does not occur in Q. Now consider some $P' = (vy)Q'$ with $Q' \in \mathrm{res}(Q,x)$. By induction, $Q \equiv (vx)Q'$. Then, by structural law (2.1), $P = (vy)Q \equiv (vy)(vx)Q' \equiv (vx)(vy)Q' = (vx)P'$. To show (2), let $P \Rightarrow S[n/x]$. Then $Q \Rightarrow T$ and $S[n/x] = T[m/y]$, with $m \notin \mathrm{new}(T)$. Since we assume that x does not occur in P, $x \neq y$. We first consider the case that $m = n$. Then $S = T[m/y][x/n] = T[x/y]$. Take $P' = Q[x/y] \in \mathrm{res}(P,x)$. By Lemma 6(1) of [2], $Q \Rightarrow T$ implies that $Q[x/y] \Rightarrow T[x/y]$, i.e., $P' \Rightarrow S$. Assume now that $m \neq n$. Then $T = S[n/x][y/m] = S[y/m][n/x]$. By induction there exists $Q' \in \mathrm{res}(Q,x)$ such that $Q' \Rightarrow S[y/m]$. Take $P' = (vy)Q' \in \mathrm{res}(P,x)$. Then $P' = (vy)Q' \Rightarrow S[y/m][m/y] = S$. Note that $y \notin \mathrm{fn}(S)$ because $T[m/y] = S[n/x]$.

Let $P = g.Q$. Define $\mathrm{res}(g.Q,x) = \{g.Q' \colon Q' \in \mathrm{res}(Q,x)\}$. To show (1), consider some $P' = g.Q'$ with $Q' \in \mathrm{res}(Q,x)$. Then $P = g.Q \equiv g.(vx)Q' \equiv (vx)g.Q' = (vx)P'$ by induction and structural law (2.4). Note that x does not occur in g because it does not occur in P by assumption. To show (2), assume that $P \Rightarrow S[n/x]$. Then $Q \Rightarrow T$

and $S[n/x]=\{g.T\}$. By our assumption, x does not occur in $g.Q$ and $x\notin \mathrm{fn}(T)$. Hence $S=\{g.T\}[x/n]=\{g.T[x/n]\}$ and $T=T[x/n][n/x]$. By induction there exists $Q'\in \mathrm{res}(Q,x)$ such that $Q'\Rightarrow T[x/n]$. Take $P'=g.Q'\in \mathrm{res}(P,x)$. Then $P'=g.Q'\Rightarrow \{g.T[x/n]\}=S$.

Let $P=!Q$. Define $\mathrm{res}(!Q,x)=\{Q'\mid !Q : Q'\in \mathrm{res}(Q,x)\}$. To show (1), consider some $P'=Q'\mid !Q$ with $Q'\in \mathrm{res}(Q,x)$. Then $P=!Q\equiv Q\mid !Q\equiv (vx)Q'\mid !Q\equiv (vx)(Q'\mid !Q)=(vx)P'$ by induction and structural laws (3.1) and (2.3). To show (2), assume that $P\Rightarrow S[n/x]$. Then $Q\Rightarrow T_i$, with mutually disjoint $\mathrm{new}(T_i)$, and $S[n/x]=\bigcup_{i\in \mathbf{N}} T_i$. Note that $x\notin \mathrm{fn}(T_i)$. Since the $\mathrm{new}(T_i)$ are mutually disjoint, there exists j such that $n\notin \mathrm{new}(T_i)$ for all $i\neq j$. By renaming the index set we may assume that $j=1$. Now $S=(\bigcup_{i\in \mathbf{N}} T_i)[x/n]=\bigcup_{i\in \mathbf{N}} T_i[x/n]=T_1[x/n]\cup \bigcup_{i\in \mathbf{N}} T_{i+1}$. Since $x\notin \mathrm{fn}(T_1)$, $T_1=T_1[x/n][n/x]$. Hence, by induction there exists $Q'\in \mathrm{res}(Q,x)$ such that $Q'\Rightarrow T_1[x/n]$. Take $P'=Q'\mid !Q\in \mathrm{res}(P,x)$. Then $P'=Q'\mid !Q\Rightarrow T_1[x/n]\cup \bigcup_{i\in \mathbf{N}} T_{i+1}=S$. □

We observe here that it is essential that the proof of the main results is by induction on the structure of Q: the P' in $\mathrm{res}(P,x)$ may be much larger than P, as can be seen in the proof of Lemma 24 for the case of replication, where structural law (3.1) is used.

As a corollary we obtain results (I1)–(I3), as discussed in the beginning of this section, for the case that Q is a restriction.

Lemma 25. (1) *If* $P\equiv_m (vx)Q'$, *then there exists* $P'\in res(P,x)$ *such that* $P'\equiv_m Q'$.
(2) *If there exists* $P'\in res(P,x)$ *such that* $P'\equiv Q'$, *then* $P\equiv (vx)Q'$.

Proof. (1) Let $P\equiv_m (vx)Q'$. Take S such that $Q'\Rightarrow S$ (see Lemma 4 of [2]). Then $(vx)Q'\Rightarrow S[n/x]$ with $n\in \mathrm{New}-\mathrm{new}(S)$. Hence $P\Rightarrow S[n/x]$. Now Lemma 24 (2) implies that there exists $P'\in \mathrm{res}(P,x)$ such that $P'\Rightarrow S$. Then $P'\equiv_m Q'$ by Lemma 2.
(2) This is immediate from Lemma 24 (1). □

Next we prove the induction step for the case that $Q=g.Q'$. The idea is similar to the case of restriction. We prove that if $P\equiv_m g.Q'$, then there exists a P' such that $P\equiv g.P'$ and $P'\equiv_m Q'$. A finite set of possible P' can be computed from P and g.

Lemma 26. *For every process term* P *and every guard* g *over* N *a finite set* $gua(P,g)$ *of process terms can be computed such that*
(1) $P\equiv g.P'$ *for every* $P'\in gua(P,g)$, *and*
(2) *if* $P\Rightarrow \{g.S\}$, *then there exists* $P'\in gua(P,g)$ *such that* $P'\Rightarrow S$.

Proof. We first observe that, by a similar argument as the one in the proof of Lemma 24, it suffices to prove the statement of the lemma for process terms P such that if y occurs bound in g, then y does not occur bound in P. For similar reasons we may, in addition, restrict ourselves to process terms P that are subconnected. This is because, by Lemma 18, a subconnected term can be constructed that is structurally congruent (and hence also multiset congruent) with P. It is easy to check in the proof

of Lemma 18 that the construction does not change the bound names of P. For such process terms P, the computation of $\mathrm{gua}(P,g)$ is, as in Lemma 24, by induction on the syntactical structure of P. Note that every subterm of P satisfies the same restrictions, and hence satisfies the induction hypothesis.

For $P=\mathbf{0}$ we define $\mathrm{gua}(P,g)=\emptyset$, because $\{g.S\}\neq\emptyset$.

Let $P=Q_1\mid Q_2$. Also in this case we define $\mathrm{gua}(P,g)=\emptyset$. In fact, assume that $P\Rightarrow\{g.S\}$. Then $Q_i\Rightarrow T_i$ and $\{g.S\}=T_1\cup T_2$. Assume that $T_1=\{g.S\}$ and $T_2=\emptyset$ (the other case is symmetric). Since Q_2 is subconnected, it follows from Lemma 20 that $Q_2=\mathbf{0}$. Hence $P=Q_1\mid\mathbf{0}$. This contradicts the fact that P is subconnected.

Let $P=(vy)Q$. By our assumption, y does not occur bound in g. If y occurs free in g, then we define $\mathrm{gua}((vy)Q,g)=\emptyset$. Otherwise we define $\mathrm{gua}((vy)Q,g)=\{(vy)Q'\colon Q'\in\mathrm{gua}(Q,g)\}$. To show (1), assume that y does not occur free in g (and hence does not occur at all in g), and consider some $P'=(vy)Q'\in\mathrm{gua}(P,g)$ with $Q'\in\mathrm{gua}(Q,g)$. Then $P=(vy)Q\equiv(vy)g.Q'\equiv g.(vy)Q'$ by induction and structural law (2.4). To show (2), assume that $P\Rightarrow\{g.S\}$. Then $Q\Rightarrow T$ and $\{g.S\}=T[n/y]$, with $n\notin\mathrm{new}(T)$. Hence $y\notin\mathrm{fn}(g.S)$, and so y does not occur at all in g. Now $T=\{g.S\}[y/n]=\{g.S[y/n]\}$. By induction there exists $Q'\in\mathrm{gua}(Q,g)$ such that $Q'\Rightarrow S[y/n]$. Now take $P'=(vy)Q'\in\mathrm{gua}(P,g)$. Then $P'=(vy)Q'\Rightarrow S[y/n][n/y]=S$.

Let $P=h.Q$ where h is a guard. We first consider the case that g has no bound name. If $g=h$, then we define $\mathrm{gua}(h.Q,g)=\{Q\}$, otherwise $\mathrm{gua}(h.Q,g)=\emptyset$. Clearly, if $g=h$, then $g.P'=g.Q=h.Q\equiv P$. Also, if $P\Rightarrow\{g.S\}$, then $Q\Rightarrow T$ and $\{g.S\}=\{h.T\}$, and so $g=h$ and $S=T$; hence $P'=Q\Rightarrow S$. Now assume that g does contain a bound name, say $g=x(y)$. If $h=x(v)$ for some name v, then we define $\mathrm{gua}(h.Q,g)=\{Q[y/v]\}$, otherwise $\mathrm{gua}(h.Q,g)=\emptyset$. To show (1), assume that $h=x(v)$ and consider $P'=Q[y/v]$. Now $P=x(v).Q\equiv x(y).Q[y/v]=g.P'$ by structural law (α); note that y does not occur in Q by the assumption at the beginning of this proof. To show (2), assume that $P\Rightarrow\{g.S\}$. Then $Q\Rightarrow T$ and $\{g.S\}=\{h.T\}$. Hence $h=x(v)$ for some v, and $x(y).S=x(v).T$, i.e., $x(-).\mathrm{inc}(S)[1/y]=x(-).\mathrm{inc}(T)[1/v]$. Consequently $\mathrm{inc}(S)[1/y]=\mathrm{inc}(T)[1/v]$ and so $\mathrm{inc}(S)=\mathrm{inc}(T)[1/v][y/1]=\mathrm{inc}(T)[y/v]=\mathrm{inc}(T[y/v])$, which shows that $S=T[y/v]$. Consider $P'=Q[y/v]\in\mathrm{gua}(P,g)$. By Lemma 6(1) of [2], $Q\Rightarrow T$ implies $P'=Q[y/v]\Rightarrow T[y/v]=S$.

Let $P=!Q$. Define $\mathrm{gua}(!Q,g)=\emptyset$. In fact, if $P\Rightarrow\{g.S\}$, then $Q\Rightarrow T_i$ with mutually disjoint $\mathrm{new}(T_i)$, and $\{g.S\}=\bigcup_{i\in\mathbf{N}}T_i$. This case cannot occur. In fact, since P is subconnected, $Q\neq\mathbf{0}$ and hence, by Lemma 20, $T_i\neq\emptyset$. Consequently $\mathrm{conn}(T_i)\geqslant 1$ and so, by Lemma 12, $\mathrm{conn}(\bigcup_{i\in\mathbf{N}}T_i)=\omega$. But $\mathrm{conn}(\{g.S\})=1$.

We observe that it follows from the definition of $\mathrm{gua}(P,g)$ that, in fact, it is always either a singleton or empty. □

As for restriction, we now obtain from Lemma 26 the results (I1)–(I3) for the case that Q is guarded, as a corollary. The proof is similar to the one of Lemma 25.

Lemma 27. (1) *If $P\equiv_m g.Q'$, then there exists $P'\in gua(P,g)$ such that $P'\equiv_m Q'$.*

(2) *If there exists $P'\in gua(P,g)$ such that $P'\equiv Q'$, then $P\equiv g.Q'$.*

We now turn to the induction step for the case that $Q = Q_1 \mid Q_2$. This case (and the one for replication) is more complicated than the previous ones, due to the necessity to use the concept of connectedness. Similarly to the previous two cases, we prove that if $P \equiv_m Q_1 \mid Q_2$, then there exist P_1 and P_2 such that $P \equiv P_1 \mid P_2$, $P_1 \equiv_m Q_1$, and $P_2 \equiv_m Q_2$. A finite set of possible pairs (P_1, P_2) can be computed from P and from upper bounds for copy(Q_1) and copy(Q_2). Recall that for a solution S, copy(S) is its copy-width. In the next lemma a bound is put on the copy-width of the solutions, to guarantee finiteness of the set of pairs (P_1, P_2). This bound is obtained from the upper bounds for copy(Q_1) and copy(Q_2) (which can be computed according to Lemma 22).

Lemma 28. *For every process term P and every number $k \in \mathbf{N}$ a finite set comp(P,k) of pairs of process terms can be computed such that*

(1) $P \equiv P_1 \mid P_2$ *for every pair* $(P_1, P_2) \in$ comp(P,k), *and*

(2) *if* $P \Rightarrow S_1 \cup S_2$, *with* new$(S_1) \cap$ new$(S_2) = \emptyset$ *and, for* $i = 1,2$, copy$(S_i) \leqslant k$, *then there exists* $(P_1, P_2) \in$ comp(P,k) *such that* $P_1 \Rightarrow S_1$ *and* $P_2 \Rightarrow S_2$.

Proof. As in the proof of Lemma 26, we may restrict ourselves to subconnected process terms P. For subconnected P, the computation of comp(P,k) is by induction on the syntactical structure of P.

The cases $P = \mathbf{0}$, $P = (\nu x)Q$, and $P = g.Q$ are treated in one stroke, using the fact that P is connected. We define comp$(P,k) = \{(P, \mathbf{0}), (\mathbf{0}, P)\}$. Then (1) follows from structural law (1.1). To show (2), assume that $P \Rightarrow S_1 \cup S_2$, with new$(S_1) \cap$ new$(S_2) = \emptyset$. Since P is connected, $S_1 \cup S_2$ is connected. By Lemma 8, either $S_1 = \emptyset$ or $S_2 = \emptyset$ (or both). Assume that $S_2 = \emptyset$; the other case is symmetric. Take $P_1 = P$ and $P_2 = \mathbf{0}$.

Let $P = Q_1 \mid Q_2$. Let $k' = \max\{k, c(Q_1), c(Q_2)\}$, where $c(Q_i)$ is the upper bound of copy(Q_i) which can be computed according to Lemma 22. Define comp$(Q_1 \mid Q_2, k) =$ $\{(R_{1,1} \mid R_{1,2}, R_{2,1} \mid R_{2,2}) : (R_{1,1}, R_{2,1}) \in$ comp(Q_1, k') and $(R_{1,2}, R_{2,2}) \in$ comp$(Q_2, k')\}$. Then (1) is proved as follows: $P = Q_1 \mid Q_2 \equiv (R_{1,1} \mid R_{2,1}) \mid (R_{1,2} \mid R_{2,2}) \equiv (R_{1,1} \mid R_{1,2}) \mid (R_{2,1} \mid R_{2,2})$ by the induction hypotheses for Q_1 and Q_2 and by structural laws (1.2) and (1.3). To show (2), assume that $P \Rightarrow S_1 \cup S_2$, with new$(S_1) \cap$ new$(S_2) = \emptyset$ and, for $i = 1,2$, copy$(S_i) \leqslant k$. Then, by (S1), there are T_1 and T_2 such that $Q_1 \Rightarrow T_1$, $Q_2 \Rightarrow T_2$, new$(T_1) \cap$ new$(T_2) = \emptyset$, and $S_1 \cup S_2 = T_1 \cup T_2$. Note that copy$(S_i) \leqslant k \leqslant k'$ and copy$(T_j) =$ copy$(Q_j) \leqslant c(Q_j) \leqslant k'$. Hence, by Lemma 15 (with $m = k'$) there exist $U_{i,j}$, $i,j \in \{1,2\}$, such that copy$(U_{i,j}) \leqslant k'$, $U_{i,1} \cup U_{i,2}$ is a copy of S_i, and $U_{1,j} \cup U_{2,j}$ is a copy of T_j. By Lemma 2, $Q_j \Rightarrow U_{1,j} \cup U_{2,j}$. Since new$(U_{i,j}) \subseteq$ new$(S_i) \cap$ new(T_j), the new$(U_{i,j})$ are mutually disjoint. Therefore, by induction, there are process terms $R_{i,j}$ such that $(R_{1,j}, R_{2,j}) \in$ comp(Q_j, k') and $R_{i,j} \Rightarrow U_{i,j}$. By (S1), $R_{i,1} \mid R_{i,2} \Rightarrow U_{i,1} \cup U_{i,2}$, and so, by Lemma 2, $R_{i,1} \mid R_{i,2} \Rightarrow S_i$, i.e., $R_{1,1} \mid R_{1,2} \Rightarrow S_1$ and $R_{2,1} \mid R_{2,2} \Rightarrow S_2$. Take $(P_1, P_2) =$ $(R_{1,1} \mid R_{1,2}, R_{2,1} \mid R_{2,2}) \in$ comp(P,k).

Let $P = !Q$. We will use Q^m to denote $\overbrace{Q \mid \cdots \mid Q}^{m}$. Define comp$(!Q,k) = \{(!Q, \mathbf{0}),$ $(\mathbf{0}, !Q),$ $(!Q, !Q)\} \cup \{(Q^m, !Q) : 1 \leqslant m \leqslant k\} \cup \{(!Q, Q^m) : 1 \leqslant m \leqslant k\}$. Then (1) follows from structural laws (1.1), (3.5), and m times (3.1). To show (2), assume that $P \Rightarrow S_1 \cup S_2$,

with $\text{new}(S_1) \cap \text{new}(S_2) = \emptyset$ and, for $i = 1,2$, $\text{copy}(S_i) \leqslant k$. By (S4) there exist T_i, $i \in \mathbf{N}$, such that $Q \Rightarrow T_i$, the $\text{new}(T_i)$ are mutually disjoint, and $S_1 \cup S_2 = \bigcup_{i \in \mathbf{N}} T_i$. Note that, by Lemma 2, the T_i are all copies of each other. Since P is subconnected, Q is connected, and hence T_i is connected. Moreover, $Q \neq \mathbf{0}$ and so, by Lemma 20, $T_i \neq \emptyset$. This means that the T_i, $i \in \mathbf{N}$, are the connected components of $S_1 \cup S_2$. From Lemma 9 (with S and T interchanged) we conclude that there is an index set $K \subseteq \mathbf{N}$ such that $S_1 = \bigcup_{i \in K} T_i$ and $S_2 = \bigcup_{i \in K'} T_i$, where $K' = \mathbf{N} - K$. Depending on the nature of K there are a number of cases.

(1) $K = \emptyset$. Then $S_1 = \emptyset$. Take $P_1 = \mathbf{0}$ and $P_2 = !Q$. The case that $K' = \emptyset$ is similar.

(2) K is a finite, nonempty, set. By renaming the index set we may assume that $K = \{1, \ldots, m\}$. Then $S_1 = T_1 \cup \cdots \cup T_m$ and $S_2 = \bigcup_{i \in \mathbf{N}} T_{m+i}$. Since $T_1, \ldots, T_m$ are the connected components of S_1 and they are all copies of each other, $\text{copy}(S_1) = m$. Hence $m \leqslant k$. Take $P_1 = Q^m$ and $P_2 = !Q$. Then $(P_1, P_2) \in \text{comp}(!Q, k)$. Moreover, by (S1) and (S4), $P_1 \Rightarrow S_1$ and $P_2 \Rightarrow S_2$, respectively. The case that K' is finite, is similar.

(3) Both K and K' are infinite. By a renaming of the index set, S_1 and S_2 can be written as $S_1 = \bigcup_{i \in \mathbf{N}} T_{2i}$ and $S_2 = \bigcup_{i \in \mathbf{N}} T_{2i-1}$. Take $(P_1, P_2) = (!Q, !Q) \in \text{comp}(!Q, k)$. Then, by (S4), $P_1 \Rightarrow S_1$ and $P_2 \Rightarrow S_2$. □

As before, we obtain results (I1)–(I3) for the case that Q is a parallel composition.

Lemma 29. *Let Q_1 and Q_2 be process terms, and let $k = max\{c(Q_1), c(Q_2)\}$.*

(1) If $P \equiv_m Q_1 \mid Q_2$, then there exists $(P_1, P_2) \in comp(P, k)$ such that $P_1 \equiv_m Q_1$ and $P_2 \equiv_m Q_2$.

(2) If there exists $(P_1, P_2) \in comp(P, k)$ such that $P_1 \equiv Q_1$ and $P_2 \equiv Q_2$, then $P \equiv Q_1 \mid Q_2$.

Proof. (1) Consider a solution S such that $Q_1 \mid Q_2 \Rightarrow S$. Then there are S_1 and S_2 such that $S = S_1 \cup S_2$, $Q_i \Rightarrow S_i$, and the $\text{new}(S_i)$ are disjoint. Since $P \equiv_m Q_1 \mid Q_2$, $P \Rightarrow S_1 \cup S_2$. Note that $\text{copy}(S_i) = \text{copy}(Q_i) \leqslant c(Q_i) \leqslant k$. We now obtain from Lemma 28 (2) that $P_i \Rightarrow S_i$ and so $P_i \equiv_m Q_i$.

(2) Immediate from Lemma 28 (1). □

Note that the number k mentioned in Lemma 29, can be computed from Q_1 and Q_2, as should be clear from Lemma 22.

The last induction step is for the case that $Q = !Q'$. The next lemma could have been formulated in the same style as Lemmas 24, 26, and 28, replacing $!Q'$ by its corresponding solution. However, in this case it turns out to be easier to keep $!Q'$ in the formulation of the lemma. We will write R for Q'. As in the proof of Lemma 28, we denote $\overbrace{R \mid \cdots \mid R}^{n}$ by R^n.

Lemma 30. *For every process term P, a finite set $rep(P)$ of process terms can be computed such that for every subconnected process term $!R$ the following two state-*

ments hold.

(1) *If* $P' \equiv R$ *for all* $P' \in rep(P)$, *then* $P \equiv !R$ *or* $P \equiv R^n$ *for some* $n \geqslant 1$.

(2) *If* $P \equiv_m !R$ *or* $P \equiv_m R^n$ *for some* $n \geqslant 1$, *then* $P' \equiv_m R$ *for all* $P' \in rep(P)$.

Proof. Since $!R$ is subconnected, R is connected and non-zero. Thus, by Lemma 20, $\mathrm{conn}(R) = 1$. Hence $\mathrm{conn}(R^n) = n$ and $\mathrm{conn}(!R) = \omega$ by Lemma 21.

As in the proof of Lemmas 26 and 28, we may restrict ourselves to subconnected P, and the proof is by induction on the syntactical structure of P. For the sake of intuition we note that we will define $\mathrm{rep}(P)$ in such a way that if $P = P_1 \mid \cdots \mid P_n \mid !P_{n+1} \mid \cdots \mid !P_{n+k}$ (as in Corollary 19), then $\mathrm{rep}(P) = \{P_1, \ldots, P_{n+k}\}$. Thus, the elements of $\mathrm{rep}(P)$ represent the connected components of P, as observed at the end of Section 5.

For the cases $P = \mathbf{0}$, $P = (\nu x)Q$, and $P = g.Q$, we define $\mathrm{rep}(P) = \{P\}$. Then (1) is obvious (with $n = 1$). To show (2), note that in these cases P is connected, and so $\mathrm{conn}(P) \leqslant 1$. Then $P \equiv_m !R$ is impossible because $\mathrm{conn}(!R) = \omega$, and $P \equiv_m R^n$ is only possible for $n = 1$. Hence $P \equiv_m R$.

Let $P = Q_1 \mid Q_2$. Define $\mathrm{rep}(Q_1 \mid Q_2) = \mathrm{rep}(Q_1) \cup \mathrm{rep}(Q_2)$. To show (1), assume that $P' \equiv R$ for all $P' \in \mathrm{rep}(Q_1 \mid Q_2)$. By induction, $Q_i \equiv !R$ or $Q_i \equiv R^{n_i}$ for some $n_i \geqslant 1$. Then either $P \equiv !R \mid !R \equiv !R$ by structural law (3.5), or $P \equiv R^{n_1} \mid !R \equiv !R$ by repeated use of structural law (3.1), or symmetrically $P \equiv !R \mid R^{n_2} \equiv !R$, or $P \equiv R^{n_1} \mid R^{n_2} \equiv R^{n_1+n_2}$. To show (2), assume that $P \equiv_m !R$ or $P \equiv_m R^n$ for some $n \geqslant 1$. By (S1) or (S4), there exist solutions S_i with mutually disjoint $\mathrm{new}(S_i)$, such that $P \Rightarrow \bigcup_{i \in I} S_i$ and $R \Rightarrow S_i$, where I is either finite or $I = \mathbf{N}$. Hence, by (S1), there are solutions T_1 and T_2 with disjoint $\mathrm{new}(T_j)$, such that $Q_j \Rightarrow T_j$ and $\bigcup_{i \in I} S_i = T_1 \cup T_2$. Since $\mathrm{conn}(R) = 1$, S_i is connected and nonempty. Lemma 9 now implies that there exist disjoint index sets I_j such that $I = I_1 \cup I_2$ and $T_j = \bigcup_{i \in I_j} S_i$. Since P is subconnected, Q_j is non-zero and so T_j is nonempty. Hence I_j is nonempty. Now, if $\#I_j = \omega$, then $Q_j \equiv_m !R$, and if $\#I_j = n$, then $Q_j \equiv_m R^n$. By induction, $P' \equiv_m R$ for all $P' \in \mathrm{rep}(Q_j)$ and so $P' \equiv_m R$ for all $P' \in \mathrm{rep}(Q_1 \mid Q_2)$.

Let $P = !Q$. Define $\mathrm{rep}(!Q) = \mathrm{rep}(Q)$. To show (1), assume that $P' \equiv R$ for all $P' \in \mathrm{rep}(P) = \mathrm{rep}(Q)$. By induction, $Q \equiv !R$ or $Q \equiv R^n$ for some $n \geqslant 1$. Then either $P \equiv !!R \equiv !R$ by structural law (3.3), or $P \equiv !R^n \equiv (!R)^n \equiv !R$ by structural laws (3.2) and (3.5), respectively. To show (2), assume that $P \equiv_m !R$ or $P \equiv_m R^n$ for some $n \geqslant 1$. Since P is subconnected, Q is connected and non-zero. Hence $\mathrm{conn}(P) = \omega$, and so $P \equiv_m !R$, i.e., $!Q \equiv_m !R$. By (S4), there exist solutions S_i with mutually disjoint $\mathrm{new}(S_i)$ and solutions T_j with mutually disjoint $\mathrm{new}(T_j)$ such that $\bigcup_{i \in \mathbf{N}} S_i = \bigcup_{j \in \mathbf{N}} T_j$, $R \Rightarrow S_i$ and $Q \Rightarrow T_j$. Since R and Q are connected and non-zero, S_i and T_j are connected and nonempty. It now follows from Lemma 10 that there exists a bijection $\psi : \mathbf{N} \to \mathbf{N}$ such that $T_j = S_{\psi(j)}$ for every $j \in \mathbf{N}$. Hence $Q \equiv_m R$, i.e., $Q \equiv_m R^n$ with $n = 1$. Consequently, by induction, $P' \equiv_m R$ for all $P' \in \mathrm{rep}(Q) = \mathrm{rep}(P)$. □

Again we obtain results (I1)–(I3), for $Q = !Q'$, but for subconnected Q only.

Lemma 31. *Let* $!Q'$ *be a subconnected process term.*

(1) *If* $P \equiv_m !Q'$, *then* $conn(P) = \omega$ *and* $P' \equiv_m Q'$ *for all* $P' \in rep(P)$.
(2) *If* $conn(P) = \omega$ *and* $P' \equiv Q'$ *for all* $P' \in rep(P)$, *then* $P \equiv !Q'$.

Proof. (1) and (2) follow directly from Lemma 30 (2) and (1). Note that, in general, $R \equiv R'$ implies $R \equiv_m R'$ implies $\mathrm{conn}(R) = \mathrm{conn}(R')$. Note also that, as shown at the start of the proof of Lemma 30, $\mathrm{conn}((Q')^n) = n$ and $\mathrm{conn}(!Q') = \omega$. □

Note that the truth value of $\mathrm{conn}(P) = \omega$ can be computed from P, by Lemma 22.
Everything has now been prepared for the proof of the first main result of this paper.

Lemma 32. *If* $P \equiv_m Q$ *then* $P \equiv Q$.

Proof. It suffices to show this result for subconnected Q. In fact, by Lemma 18, there is a subconnected Q' such that $Q \equiv Q'$, and so $Q \equiv_m Q'$ by Lemma 1.
The proof is by induction on the syntactical structure of Q, assuming that Q is subconnected. It follows immediately from Lemmas 23(1), 25, 27, 29, and 31. Note that the subconnectedness is used in Lemma 31. □

This last result, together with Lemma 1, proves that (extended) structural congruence and multiset congruence are the same.

Theorem 33. $P \equiv Q$ *if and only if* $P \equiv_m Q$.

From the introduction of this section, Lemmas 23 (2), 25, 27, 29, 31, and all computability arguments (in particular the computability of $\mathrm{res}(P,x)$, $\mathrm{gua}(P,g)$, $\mathrm{comp}(P,k)$, and $\mathrm{rep}(P)$, as shown in Lemmas 24, 26, 28, and 30, respectively) we obtain the second main result of this paper: the decidability of (extended) structural congruence.

Theorem 34. *It is decidable, for process terms* P *and* Q, *whether or not* $P \equiv Q$.

In Lemma 22 we have shown that for every process term P an upper bound for the number $\mathrm{copy}(P)$ can be computed. Using the decidability of structural congruence, we can now show that $\mathrm{copy}(P)$ can be determined precisely, by the following algorithm. Note first that the multiplicity function 'mult' can also be carried over from solutions to process terms, cf. Lemma 13: if $P \Rightarrow S$ and $Q \Rightarrow S'$, then $\mathrm{mult}(Q,P) = \mathrm{mult}(S',S)$. To compute $\mathrm{copy}(P)$ we may clearly assume P to be subconnected and of the form $P = P_1 \mid \cdots \mid P_n \mid !P_{n+1} \mid \cdots \mid !P_{n+k}$, as in Corollary 19. As observed at the end of Section 5, the P_i represent the connected components of P. This implies that $\mathrm{copy}(P) = \max\{\mathrm{mult}(P_i,P) : 1 \leqslant i \leqslant n, \mathrm{mult}(P_i,P) \neq \omega\}$. Note that $\mathrm{mult}(P_{n+j}, P) = \omega$ for $1 \leqslant j \leqslant k$. Now, by Lemma 2, $\mathrm{mult}(P_i,P) = \omega$ if there exists $1 \leqslant j \leqslant k$ such that $P_i \equiv_m P_{n+j}$, and otherwise $\mathrm{mult}(P_i,P) = \#\{k \in \{1,\ldots,n\} : P_i \equiv_m P_k\}$. This shows that $\mathrm{copy}(P)$ is computable. Note that it also shows that $\mathrm{mult}(Q,P)$ is computable: if there is a P_i such that $Q \equiv_m P_i$ then $\mathrm{mult}(Q,P) = \mathrm{mult}(P_i,P)$, and otherwise $\mathrm{mult}(Q,P) = 0$.

Since, by the results of [2], $P \equiv Q$ implies that P and Q are strongly bisimilar (in the transition system of the small π-calculus), structural congruence is a decidable sufficient condition for strong bisimilarity. Note that $P \equiv_M Q$ implies $P \equiv Q$ but not the other way around, where $\equiv_M$ is the original structural congruence of [6]. Thus, from this point of view, the decidability of $\equiv$ is more interesting than the one of $\equiv_M$. Nevertheless one might be interested in the decidability of $\equiv_M$. We conjecture that $\equiv_M$ is decidable, more precisely, that there is a computable transformation 'tra' of process terms such that $P \equiv_M Q$ if and only if $\mathrm{tra}(P) = \mathrm{tra}(Q)$.

Acknowledgements

We thank the referees for their constructive remarks.

References

[1] G. Berry, G. Boudol, The chemical abstract machine, Theoret. Comput. Sci. 96 (1992) 217–248.
[2] J. Engelfriet, A multiset semantics for the pi-calculus with replication, Theoret. Comput. Sci. 153 (1996) 65–94.
[3] G. Milne, R. Milner, Concurrent processes and their syntax, J. ACM 26 (1979) 302–321.
[4] R. Milner, Flowgraphs and flow algebras, J. ACM 26 (1979) 794–818.
[5] R. Milner, Communication and Concurrency, Prentice-Hall, Englewood Cliffs, NJ, 1989.
[6] R. Milner, Functions as processes, Math. Struct. Comput. Sci. 2 (1992) 119–141.
[7] R. Milner, Turing Award Lecture: Elements of interaction, Comm. ACM 36 (1993) 78–89.
[8] R. Milner, J. Parrow, D. Walker, A calculus of mobile processes, Inform. Comput. 100 (1992) 1–77.