Linear Constraint Systems as High-level Nets*

Eike Best
Institut für Informatik, Universität Hildesheim, Marienburger Platz 22,
D-31141 Hildesheim, Germany. Email: e.best@informatik.uni-hildesheim.de

Catuscia Palamidessi
DISI, Università di Genova, Via Dodecaneso, 35,
I-16146 Genova, Italy. Email: catuscia@disi.unige.it

Abstract. Linear constraint systems are simple deductive systems based on the main underlying idea of linear logic: hypotheses represent physical resources which are consumed by the entailment relation. For such systems, we define back-and-forth translations into a class of high-level Petri nets. Using the specific properties of that class of nets, and previous results about the complexity of the reachability problem for such nets, we examine the complexity of the entailment problem for finitely generated linear constraint systems and we show that it is NP-complete.

1 Introduction

Constraint programming is one of the most successful paradigms for processing declarative information. One of the reasons of its success is that it leads quite naturally to a model for concurrency, as shown in [16, 17]. In this model, agents interact via a *shared store* containing all the information generated by the system. The basic communication actions are the *tell*, which updates the store by adding new information (i.e. a new constraint), and the *ask*, which tests whether the store entails a certain constraint. The ask can be successful, in which case the agent proceeds, or unsuccessful, in which case the agent suspends (waits for the other agents to enrich the store). Concerning the tell, there are two variants: the *atomic tell*, which blocks if the new constraint makes the store inconsistent, and the *eventual tell*, which just adds a constraint without any consistency check. It is relatively easy to prove that the first mechanism leads to a concurrent paradigm which is strictly more expressive [3]. Intuitively this is because the atomic tell is a test-and-set action, which cannot be simulated by test (ask) and set (eventual tell) separately. On the other hand, the particular kind of test-and-set performed by the atomic tell is not very suitable for distributed systems. In fact the test for consistency depends on the *global* content of the store, hence only one such action can be performed at a time.

In [4] we have proposed a concurrent constraint paradigm aimed to combine expressiveness and parallelism. The idea is to allow a test-and-set action (*get*) which tests (part of) the store for entailment, like the ask, but, when successful, it removes from the store the entailing information. This is similar to the interaction

* This work is supported by the HCM project EXPRESS.

mechanism used in Linda [12, 6], where the store is seen as a *blackboard* in which agents add and remove information. However, in Linda there is no structure on the information, i.e. the tokens of information are essentially unrelated (apart from the relation given by variable instantiation). On the contrary, our paradigm assumes an underlying inference mechanism which allows for some deductive processing of information at the level of the store.

The standard notion of constraint system is based on the laws of classical logic, and does not support naturally the get operation. For instance, the entailment relation is monotonic, i.e. if a constraint c entails a constraint d, then also c plus c', for any c', entails d. Thus the notion of *entailing information* as it is here is not suitable to our purposes. One could define it as "the minimal information which entails ...", but also the concept of minimality is not natural here, since c plus c is regarded as equivalent to c. In order to model naturally the get operation, therefore, it is convenient to adopt a notion of constraint system based on a logic like linear logic, where items of information are seen as physical resources which are *consumed* by the entailment relation, and where the notion of entailing information, as we need it, is primitive.

Linear constraint systems are simple deduction systems which are to linear logic as constraint systems are to classical logic. Technically, the main difference is that an element (of a system) is defined as a multiset of constraints, instead than a set. The following example, from [4], illustrates the main idea.

Example 1. Let $available(x)$ denote the situation in which the theater place number x is available, and let $mpc(y)$ denote the fact that Mr. y makes a phone call to reserve a theater place. Then we want to have an inference of the form

$$\{available(x), mpc(y)\} \;\vdash\; booked(x, y)$$

where $booked(x, y)$ means that Mr. y has booked place x. Now, assume that the price of a place is \$50. If a person has booked a place, and he has the money to pay for it, then he can take the place:

$$\{booked(x, y), has(y, \$50)\} \;\vdash\; takes(y, x).$$

If we have, say, two places available with numbers 1 and 2, then the initial situation is $\{available(1),\ available(2)\}$. Now, suppose that Mr. A and Ms. B want to go to the theater, make phone calls, and have \$50 to pay the place. The store becomes $\{available(1),\ available(2),\ mpc(A),\ mpc(B),\ has(A, \$50),\ has(B, \$50)\}$. The two possible final stores, nondeterministically generated by the inference system, are $\{takes(A, 1),\ takes(B, 2)\}$ or $\{takes(A, 2),\ takes(B, 1)\}$. If a third person Ms. C makes now a phone call, she cannot reserve a place, because there are no places available anymore.

The reason why constraint systems in literature are only a subset of classical logic is that, for practical uses, it is essential that the entailment relation is decidable and that it is computed efficiently. Two questions which then arise, about the notion of linear constraint system, are:

Under what conditions the entailment problem is decidable and "not too hard"?, and

Does the definition of entailment lead to a natural distributed model, i.e. does it present some natural degree of parallelism?

In this paper we investigate these two aspects.

We attack the problem by using the formalism of Petri nets, which are particularly convenient for our purposes, since they represent a natural model of distributed computation, and complexity questions about them have been extensively investigated. Moreover, the relationship between linear logic and Petri nets has been investigated in [5, 8], albeit not focussing on constraint systems. The main idea is to regard constraints as tokens, and entailment as transitions. A similar interpretation was considered also in [13], where it was shown that (standard) constraint systems can be naturally modeled as Contextual Nets. Note that in the standard notion of entailment there is no "consumption of resources", hence the standard model of Petri Nets (where a transition consumes its pre-condition and generates its post-conditions) is more suited for linear constraint systems rather than for standard constraint systems. In the above mentioned work, in fact, the authors propose an enrichment of Petri Nets with the notion of *context*, i.e. a condition which must be present for the transition, but it is not consumed by it.

We prove that there is a complete correspondence between a particular class of high-level nets and a class of linear constraint systems which we call *finitely generated*. By using this correspondence, and previous results about the complexity of the reachability problem, we prove that the entailment problem in such systems is NP-complete.

The paper is organized as follows. In Section 2 we recall the notion of linear constraint system and we define the property of being finitely generated. In Section 3 we recall the definition of high-level net, we illustrate it with an example, and show the correspondence with ordinary Petri Nets. Furthermore we define the class of T-out-sequential, quasi-elementary high-level nets, and we prove that for this class of nets the reachability problem is NP-complete (by using previous results about T-out-sequential Petri nets). In Section 4 we define the translation from finitely generated linear systems into this class of nets and vice versa. In Section 5 we prove the NP-completeness of the entailment relation in such constraint systems. In Section 6 we discuss linear constraint system with variables, and an alternative translation into high-level nets. This section also describes a benefit of the high-level (as opposed to a low-level) translation; in this way, the distributed character and the structure of the constraint system can be be preserved almost one-to-one. Section 7 concludes.

2 Linear constraint systems

Before introducing the notion of linear constraint system, it is worth recalling the notion of standard constraint system. Actually in literature this notion has been formalized in various ways, which differ for degree of abstraction. We illustrate here the notion of Saraswat, Rinard and Panangaden [17], which is the most abstract we know of, and which is closely related to the notion of *information system* by Scott [18].

Definition 1. [17]

- A *basic constraint system* is a pair $\langle B, \vdash_B \rangle$ where B is a set of *basic constraints*, c, d ..., and the *basic entailment relation* $\vdash_B \subseteq \mathcal{P}_{\mathrm{f}}(B) \times B$ (where $\mathcal{P}_{\mathrm{f}}(B)$ is the set of the finite subsets of B) satisfies:
  (i) if $c \in s$ then $s \vdash_B c$,
  (ii) if $s \vdash_B c_1, \ldots, s \vdash_B c_n$, and $\{c_1, \ldots, c_n\} \vdash_B c$, then $s \vdash_B c$.
- The constraint system induced by $\langle B, \vdash_B \rangle$ is a pair $\langle \mathcal{P}_{\mathrm{c}}(B), \vdash \rangle$ where $\mathcal{P}_{\mathrm{c}}(B)$ is the set of subsets of B closed under entailment, i.e. $\mathcal{P}_{\mathrm{c}}(B) = \{s \subseteq B \mid \text{if } s' \subseteq s \text{ and } s' \vdash_B c \text{ then } c \in s\}$, and $\vdash \subseteq \mathcal{P}_{\mathrm{c}}(B) \times \mathcal{P}_{\mathrm{c}}(B)$ is defined by $s \vdash s'$ iff for any $c \in s'$ there exists $s'' \subseteq s$ such that $s'' \vdash_B c$.

It can be proved that $\langle \mathcal{P}_{\mathrm{c}}(B), \vdash \rangle$ is an algebraic lattice with ordering relation given by the inverse of $\vdash$. If we regard B as a set of propositions, and $c \vdash_B d$ as an axiom stating that proposition c implies proposition d, then we can consider the algebraic elements of $\mathcal{P}_{\mathrm{c}}(B)$ as particular propositional formulas on B (the formulas obtained by conjunction of basic propositions), modulo logical equivalence. Conjunction corresponds to the join operator. In other words, this construction can be considered (limited to the algebraic part) as a subset of the classical propositional calculus, subset in the sense that certain operators, like disjunction and negation, are usually not expressible.

In [17] this construction is further enriched with the addition of logical variables and (a weak form of) existential quantifier. This is done by means of *cylindrification operators* and *diagonal elements*, and by imposing (some of) the axioms of cylindrical algebras. The resulting system can be regarded (still limited to the algebraic part) as a subset of a weak form of first order logic; subset in the same sense as above, and weak in the sense that not all the laws of first order logic hold here.

We illustrate now the notion of *linear constraint system*, as defined in [4]. For the sake of simplicity, we do not consider here the mechanism of variable instantiation (or else we assume that the basic entailment relation is closed under variable instantiation). In Section 6 we discuss this mechanism.

As announced in the introduction, the aim is to model a constraint as a physical resource, which can be produced in several copies and each copy of which disappears once it is consumed, i.e. used as a hypothesis in the entailment relation. In order to represent the idea that the same resource can have multiple copies, it is natural to define the elements of the constraint system to be multisets instead than sets. Furthermore the underlying logic must be modified. The derivation of a (basic) constraint by the entailment relation *consumes the hypothesis*, hence, if the derivation of a constraint uses twice the same information, then this information must be represented twice in the hypothesis. Therefore the entailment relation must be defined from multisets of constraints to constraints.

In the following, $\mathcal{M}(X)$ and $\mathcal{M}_{\mathrm{f}}(X)$ stand for the set of multisets and for the set of finite multisets of a set X, respectively; $\oplus$ is the multiset sum, $\ominus$ is the multiset difference. We refer to the appendix for the formal definition of these notions.

Definition 2. [4]

- A *basic linear constraint system* is a pair $\langle B, \vdash_B \rangle$ where B is a set of *basic constraints* and the *basic entailment relation* $\vdash_B \subseteq \mathcal{M}_{\mathrm{f}}(B) \times B$ satisfies:
  (i) $\{c\} \vdash_B c$,
  (ii) if $\mu_1 \vdash_B c_1, \ldots, \mu_n \vdash_B c_n$ and $\{c_1, \ldots, c_n\} \vdash_B d$, then $\mu_1 \oplus \ldots \oplus \mu_n \vdash_B d$.
- The *linear constraint system* induced by $\langle B, \vdash_B \rangle$ is the structure $\langle \mathcal{M}(B), \vdash \rangle$, where $\vdash \subseteq \mathcal{M}(B) \times \mathcal{M}(B)$ is the minimal relation such that

$$\text{if} \quad \forall i \in I.\ \mu_i \vdash_B c_i \quad \text{then} \quad \bigoplus_{i \in I} \mu_i \vdash \bigoplus_{i \in I} \{c_i\}.$$

Note that we cannot define the elements of a constraint system to be multisets closed under entailment. In fact if we take a definition like in 1, then a set of constraints and its closure are not equivalent, because they represent different sets of resources.

A linear constraint system does not have all the properties as a constraint system; it is not a complete algebraic lattice and not even a lattice (for instance it does not necessarily have a minimum element). However it is a preorder, and therefore can be transformed into a partial order by factorising it with respect to the associated equivalence relation.

Lemma 3. $\vdash$ *is a preorder on* $\mathcal{M}(S) \times \mathcal{M}(S)$.

Proof

Reflexivity. Assume $\mu = \bigoplus_{i \in I}\{c_i\}$. By Definition 2 (i) we have that for each $i \in I$, $\{c_i\} \vdash_B c_i$ holds. Hence by definition $\bigoplus_{i \in I}\{c_i\} \vdash \bigoplus_{i \in I}\{c_i\}$ holds.

Transitivity. Assume $\mu \vdash \mu'$ and $\mu' \vdash \mu''$. We have to show that $\mu \vdash \mu''$. Assume $\mu'' = \bigoplus_{i \in I}\{c_i\}$. Then there exist a collection of multisets $\{\mu'_i \mid i \in I\}$ such that $\mu' = \bigoplus_{i \in I} \mu'_i$ and for each $i \in I$, $\mu'_i \vdash_B c_i$ holds. Let, for $i \in I$, $\mu'_i = \bigoplus_{j \in J_i}\{d_{ij}\}$. We have that there exists a collection of multisets $\{\mu_{ij} \mid j \in J_i\}$ such that $\mu = \bigoplus_{i \in I} \bigoplus_{j \in J_i} \mu_{ij}$ and for each $i \in I, j \in J_i$, $\mu_{ij} \vdash_B d_{ij}$ holds. By Definition 2 (ii), we get that for each $i \in I$, $\bigoplus_{j \in J_i} \mu_{ij} \vdash_B c_i$ holds. Hence by definition we obtain $\bigoplus_{i \in I} \bigoplus_{j \in J_i} \mu_{ij} \vdash \bigoplus_{i \in I}\{c_i\}$. ∎

The relation $\vdash$ is generally infinite, even when B is finite, and even when restricted to $\mathcal{M}_{\mathrm{f}}(B)$ modulo equivalence. Of course, from a practical point of view we are mainly interested in linear constraint systems which have a decidable and "not too hard" entailment relation. To this end, we introduce the notion of *finitely generated* linear constraint systems. We will show that for this class of systems the entailment relation is in fact decidable and NP complete. Intuitively, a linear constraint systems is finitely generated if the basic entailment relation is obtained by starting from a finite relation and closing it up with respect to Conditions (i) and (ii) of Definition 2.

Definition 4. A linear constraint system is *finitely generated* if its base is a pair $\langle B, \vdash_B \rangle$ where B is a finite set, and $\vdash_B$ is defined by the following inductive system, for some finite $\vdash_0 \subseteq \mathcal{M}_f(B) \times B$:

$$(1)\ \frac{}{\{c\} \vdash_B c} \qquad (2)\ \frac{\mu \vdash_0 c}{\mu \vdash_B c} \qquad (3)\ \frac{\mu_1 \vdash_B c_1 \ , \ldots \ , \mu_n \vdash_B c_n \ , \ \{c_1, \ldots, c_n\} \vdash_B d}{\mu_1 \oplus \ldots \oplus \mu_n \vdash_B d}$$

We conclude this section by showing a property of finitely generated constraint systems, which will be useful for studying the correspondence with high-level nets.

Definition 5. Let $\langle B, \vdash_B \rangle$ be a basic constraint system finitely generated by $\vdash_0$. Define $\rightsquigarrow$ as the relation induced by the following rule:

$$\frac{\mu \in \mathcal{M}_{\mathrm{f}}(B) \ , \ \{c_1, \ldots, c_n\} \vdash_0 d}{\{c_1, \ldots, c_n\} \oplus \mu \rightsquigarrow \{d\} \oplus \mu}$$

Proposition 6. *Let $\langle \mathcal{M}(B), \vdash \rangle$ be a finitely generated linear constraint system with generating relation $\vdash_0$. Let μ and μ' be two finite multisets of B. Then $\mu \vdash \mu'$ if and only if $\mu \rightsquigarrow^* \mu'$, where $\rightsquigarrow^*$ is the reflexive and transitive closure of $\rightsquigarrow$.*

Proof The if part is easy. To prove the only-if part, we use the following lemma:

Lemma 7. *Let μ, μ' be two finite multisets of B, let $c \in B$, and assume that $\mu \vdash_B c$. Then $\mu \oplus \mu' \rightsquigarrow^* \{c\} \oplus \mu'$ holds.*

Proof By induction on the definition of $\vdash_B$. There are three cases, depending whether $\mu \vdash_B c$ is obtained from Rule (1), Rule (2) or Rule (3) respectively.

(1) In this case $\mu = \{c\}$. Just observe that $\{c\} \oplus \mu' \rightsquigarrow^0 \{c\} \oplus \mu'$.
(2) In this case $\mu \vdash_0 c$. Then $\mu \oplus \mu' \rightsquigarrow \{c\} \oplus \mu'$ holds by definition.
(3) In this case there exist $\mu_1, \ldots, \mu_n$, $c_1, \ldots, c_n$ such that $\mu_1 \vdash_B c_1, \ldots, \mu_n \vdash_B c_n$, $\{c_1, \ldots, c_n\} \vdash_B c$, and $\mu = \mu_1 \oplus \ldots \oplus \mu_n$. By inductive hypothesis we have
   - for any finite multisets $\mu'_1, \ldots \mu'_n$, for any $i \in \{1, \ldots, n\}$, $\mu_i \oplus \mu'_i \rightsquigarrow^* \{c_i\} \oplus \mu'_i$ holds, and
   - for any finite multiset μ'', $\{c_1, \ldots, c_n\} \oplus \mu'' \rightsquigarrow \{c\} \oplus \mu''$ holds.

   Then define, for $i \in \{1, \ldots, n\}$, $\mu'_i = \mu_{i+1} \oplus \ldots \oplus \mu_n \oplus \{c_1, \ldots, c_{i-1}\} \oplus \mu'$, and $\mu'' = \mu'$. We have that, for each $i \in \{1, \ldots, n\}$, $\mu_i \oplus \ldots \oplus \mu_n \oplus \{c_1, \ldots, c_{i-1}\} \oplus \mu' \rightsquigarrow^* \mu_{i+1} \oplus \ldots \oplus \mu_n \oplus \{c_1, \ldots, c_i\} \oplus \mu'$, and $\{c_1, \ldots, c_n\} \oplus \mu' \rightsquigarrow^* \{c\} \oplus \mu'$. Finally, apply the transitivity of $\rightsquigarrow^*$. ∎

We prove now Proposition 6. Let $\mu' = \{c_1, \ldots, c_n\}$. Then there are $\mu_1, \ldots, \mu_n$ such that $\mu = \mu_1 \oplus \ldots \oplus \mu_n$ and $\mu_1 \vdash_B c_1, \ldots, \mu_n \vdash_B c_n$. By previous lemma we have, for each $i \in \{1, \ldots, n\}$, $\mu_i \oplus \ldots \oplus \mu_n \oplus \{c_1, \ldots, c_{i-1}\} \rightsquigarrow^* \mu_{i+1} \oplus \ldots \oplus \mu_n \oplus \{c_1, \ldots, c_i\}$. Hence apply the transitivity of $\rightsquigarrow^*$. ∎

3 High-level Petri nets

Rather than use the full power of Pr/T-nets [1, 10], colored nets [11] or algebraic nets [15, 19], we consider a somewhat restricted model: M-nets [2].

Let $x, y, z, \ldots$ (letters near the end of the alphabet) be variables and let $a, b, c, \ldots$ (letters near the beginning of the alphabet) be values. A *high-level net*, for our present purpose, is a structure $N = (S, T, \iota)$. As usual, S is the set of places and T is the set of transitions (satisfying $S \cap T = \emptyset$). The function ι associates:

- With every place $s \in S$ a nonempty set of values (its *type*).
- With every pair (s, t) and (t, s) (with $s \in S$ and $t \in T$) a possibly empty multiset of variables.
- With every transition t a predicate involving variables from the set $Var(t)$,

where $Var(t)$ is defined as the set of variables occurring in a multiset $\iota(s, t)$ or $\iota(t, s')$, for some $s, s' \in S$. Function ι may be considered to satisfy these properties:

$$s \neq s' \Rightarrow \iota(s) \cap \iota(s') = \emptyset \quad \text{and} \quad t \neq t' \Rightarrow Var(t) \cap Var(t') = \emptyset,$$

since it follows from standard high-level net theory that they are no restriction of generality; they can always be forced to be true, provided enough values and variables are available in the universe.

A *marking* M of N is a function associating with every place s a finite multiset $M(s)$ on $\iota(s)$. A *binding of* t, where t is a transition, is a function σ associating a value with every variable in $Var(t)$.[2] A binding of t is *legal* if variables get assigned values from the types of the places they correspond to, i.e. if x is a variable occurring in $\iota(s, t)$ or in $\iota(t, s)$, then $\sigma(x) \in \iota(s)$. In what follows we will tacitly assume all bindings to be legal. A (legal) binding σ of t is *enabling (for t)* if the predicate $\iota(t)$ evaluates to *true* under σ. A marking M *activates* t *under* σ *(or 'in mode* σ*')* if σ enables t and, moreover, for all (s, t), the multiset of values $\iota(s, t)[\sigma]$ (i.e., σ applied to every individual element of $\iota(s, t)$) is a sub-multiset of $M(s)$. If M activates t in mode σ then t may *occur in mode* σ, yielding a successor marking M' which is calculated as follows: for every place s,

$$M'(s) \quad = \quad (M(s) \ominus (\iota(s, t)[\sigma])) \oplus (\iota(t, s)[\sigma]).$$

Figure 1 exemplifies the notions just defined. The net on the left-hand side of this figure is $(\{s_1, s_2, s_3\}, \{t\}, \iota)$ with

$$\begin{array}{ll}
\iota(s_1) \;\; = \{0, 1\} & \iota(s_2, t) = \{u\} \\
\iota(s_2) \;\; = \{2, 3, 4\} & \iota(t, s_2) = \emptyset \\
\iota(s_3) \;\; = \{5\} & \iota(s_3, t) = \emptyset \\
\iota(s_1, t) = \{x, x, z\} & \iota(t, s_3) = \{y\} \\
\iota(t, s_1) = \emptyset & \iota(t) \quad\;\; = (z{=}0 \wedge u{=}3 \wedge y{=}5).
\end{array}$$

[2] In the following, we use standard notation relating to bindings and substitutions; thus, if X is some object then $X[\sigma]$ denotes X where σ is applied (elementwise or otherwise) to X.

The same net is also shown on the right-hand side of the figure. The marking M on the left-hand side is $M(s_1) = \{0, 1, 1\}$, $M(s_2) = \{2, 2, 3\}$ and $M(s_3) = \{5\}$, while the marking M' (of the same net) on the right-hand side is $M'(s_1) = \emptyset$, $M'(s_2) = \{2, 2\}$ and $M'(s_3) = \{5, 5\}$. Consider four bindings for t:

$$\sigma_1 : x{=}0, z{=}1, u{=}3, y{=}0 \qquad \sigma_3 : x{=}0, z{=}0, u{=}3, y{=}5$$
$$\sigma_2 : x{=}0, z{=}1, u{=}3, y{=}5 \qquad \sigma_4 : x{=}1, z{=}0, u{=}3, y{=}5.$$

The first binding σ_1 is not legal because the value of y is not in the type of the corresponding place. The second binding is legal but not enabling for t because the predicate does not evaluate to *true* (due to z being bound to 1 rather than 0). The third binding σ_3 is legal and enabling for t, but the marking M on the left-hand side does not activate t in mode σ_3 (due to there being insufficiently many tokens 0 on place s_1). Finally, σ_4 is legal, enabling for t, and such that t is activated by M in mode σ_4. Thus, t can occur in mode σ_4 starting from M. If it does, then the marking M' shown on the right-hand side of the figure is obtained after its occurrence.

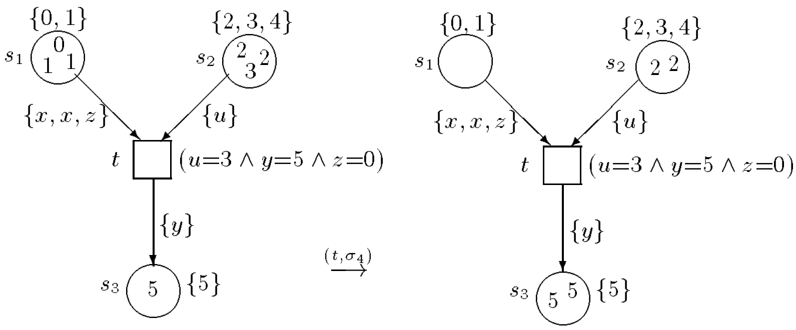

Fig. 1. Illustrating some basic notions of high-level nets

Next, we define the unfolding of a high-level net into a place/transition net. A (weighted) place/transition net is a triple (S, T, W) where S and T are sets as above and W is a function from $(S{\times}T){\cup}(T{\times}S)$ to $\mathbb{N}$ attaching a weight (possibly 0) to every arc. Let $N = (S, T, \iota)$ be a high-level net as defined above. The *unfolding* of N, $U(N)$, is a place/transition net $U(N) = (S_u, T_u, W_u)$ defined as follows:

- $S_u = \{s_v \mid s{\in}S \text{ and } v{\in}\iota(s)\}$.
- $T_u = \{t_\sigma \mid t{\in}T \text{ and } \sigma \text{ is an enabling binding of } t\}$.
- $W_u(s_v, t_\sigma) = |\, \iota(s,t)[\sigma]|_v \,|$ and $W_u(t_\sigma, s_v) = |\, \iota(t,s)[\sigma]|_v \,|$.[3]

Moreover, a marking M of N corresponds to a marking M_u of $U(N)$, which is a function from places to natural numbers as usual, in a unique and canonical way, namely as $M_u(s_v) = |\, M(s)|_v \,|$ for all $s_v \in S_u$. Figure 2 shows the unfolding

[3] Compare the appendix for the notion of the restriction $\mu|_v$ of a multiset μ.

corresponding to the net on the left-hand side of Figure 1.[4] Note that transition t of Figure 1 has been split into t_1 and t_2. This is due to the fact that t has exactly two enabling bindings which are discriminated by $x{=}0$ and $x{=}1$, respectively, namely $x{=}0, z{=}0, u{=}3, y{=}5$ and $x{=}1, z{=}0, u{=}3, y{=}5$. It is left to the reader to compute the unfolding of the net shown on the right-hand side of Figure 1.[5] It should be added that every place/transition system can be interpreted canonically as a special high-level system (the specialisation consists in that every place type is a singleton and every transition predicate has precisely one enabling binding); however for historical reasons it is not usual to view place/transition nets in this way.

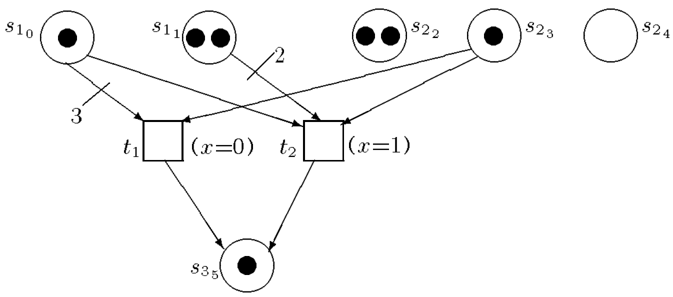

Fig. 2. Unfolding of the net shown on the left-hand side of Figure 1

We are interested in certain classes of high-level nets and place/transition nets. Let a place/transition net (S, T, W) be called *T-out-sequential* if for all $t \in T$, $\sum_{s \in S} W(t, s) = 1$ and *T-in-sequential* if for all $t \in T$, $\sum_{s \in S} W(s, t) = 1$. T-out-sequential nets are the class of interest in connection with linear constraint programming while T-in-sequential nets are the class of nets used for translating BPP (Basic Parallel Processes) of process algebra [7, 9].

Proposition 8. *The problem*

REACH-1: Given a finite T-out-sequential place/transition net N and two markings M and M' of N, is M' reachable from M?

is NP-complete.

Proof Let $N'{=}(S, T, W')$ be the reverse of $N{=}(S, T, W)$, i.e. $W'(t, s){=}W(s, t)$ and $W'(s, t){=}W(t, s)$ for all $s{\in}S$ and $t{\in}T$. From the transition rule of place/transition nets [14] it follows that M' is reachable from M in N iff M is reachable from M' in N'. Moreover, N' is a T-in-sequential net. Reachability in T-in-sequential nets has been shown to be NP-complete in [9]. The NP-completeness of REACH-1 follows immediately. ∎

The definition of being T-out-sequential can be lifted to high-level nets. Let $N = (S, T, \iota)$ be a high-level net. Then N is called T-out-sequential if for all

[4] Arc weights greater than 1 have been indicated explicitly.

[5] Hint: the structure of the net is the same, the only difference is in the marking.

transitions t there is exactly one place s' with $|\iota(t,s')| = 1$ while for all the other places, $|\iota(t,s)| = 0$. It is clear that if N is T-out-sequential then so is its unfolding $U(N)$.[6] However, the previous proposition cannot be transferred immediately because it may be the case that an exponential number of transitions is created in the unfolding of a (T-out-sequential) high-level net. Hence we consider nets with the following property. A high-level net is called *quasi-elementary* if every transition has exactly one enabling binding.[7] Note that via its interpretation as a high-level net, every place/transition net is always quasi-elementary. A high-level net is *finite* iff S is finite, T is finite and all place types $\iota(s)$, for $s \in S$, are finite sets. If N is finite then so is its unfolding $U(N)$.

Corollary 9. *The problem*

> REACH-2: Given a finite, T-out-sequential, quasi-elementary high-level net N and two markings M and M' of N, is M' reachable from M?

is NP-complete.

Proof By the set of assumptions, it is possible to construct the unfolding $U(N)$ of N in polynomial time. By standard high-level net theory [2, 11], M' is reachable from M in N iff the unfolded marking $U(M')$ is reachable from $U(M)$ in $U(N)$. The result follows immediately from Proposition 8. ■

4 Linear constraint systems and high-level nets

This section has two parts. First we define a translation from finitely generated linear constraint systems to high-level Petri nets (Section 4.1). Then we define a converse translation from quasi-elementary T-out-sequential high-level nets into linear constraint systems (Section 4.2). The two translations are not converses of each other. For instance, starting out with an arbitrary net N satisfying the conditions, deriving the linear constraint system $C(N)$, and then deriving back a net $N(C(N))$, will always end up with a one-place net even if the original one had more places than that.

4.1 Translating a linear constraint system into a high-level net

Let a finitely generated linear constraint system $C = \langle \mathcal{M}(B), \vdash \rangle$ be given and let $\vdash_0 \subseteq \mathcal{M}_{\mathrm{f}}(B) \times B$ be its generating relation. We associate a high-level net $N = N(C)$ with C as follows. N has exactly one place, say s, with type B. For every element of $\vdash_0$ there is a transition t. Suppose $(\{c_1, \ldots, c_n\}, d) \in \vdash_0$. Then the corresponding transition t has the following connections: $\iota(s,t) = \{x_1, \ldots, x_n\}$, where the x_i's are variables occurring with the same multiplicities as specified by the multiset $\{c_1, \ldots, c_n\}$, i.e. if c_i is the same value as c_j then x_i is the same variable as x_j. Moreover, $\iota(t,s) = \{y\}$. Finally, $\iota(t) = ((\forall i, 1{\le}i{\le}n{:}\, x_i{=}c_i) \wedge (y{=}d))$.

6 Note that this happens to be the case in Figures 1 and 2.

7 The weaker property of having a polynomially bounded number of enabling bindings is sufficient for the next corollary, but in the following, we will be interested only in nets for which this number is actually 1. The number 0 is uninteresting because transitions with no enabling bindings are dead.

Example 2. Let $B = \{a, \mathit{true}, \mathit{false}\}$ and

$$\vdash_0 = \{ (\{a,a\}, \mathit{false}), (\emptyset, \mathit{true}), (\{\mathit{false}\}, a), (\{\mathit{false}\}, \mathit{true}), (\{\mathit{false}\}, \mathit{false}) \}.$$

We do not show the relation $\vdash$, i.e. we do not give the system $C = \langle \mathcal{M}(B), \vdash \rangle$ generated by $\vdash_0$ explicitly.[8] The net $N(C)$ corresponding to this system has one place with a value set of size three and five transitions corresponding to the five members of $\vdash_0$. It is shown (with initial marking $\{a, a\}$, which is not specified by the construction) in Figure 3. To verify Proposition 10 below, the reader may check that with this initial marking (for example) transition t_1 is enabled, and after occurrence of t_1, t_3 is enabled and can occur yielding $\{a\}$, and that the inference system allows a corresponding deduction $\{a, a\} \vdash \{a\}$.

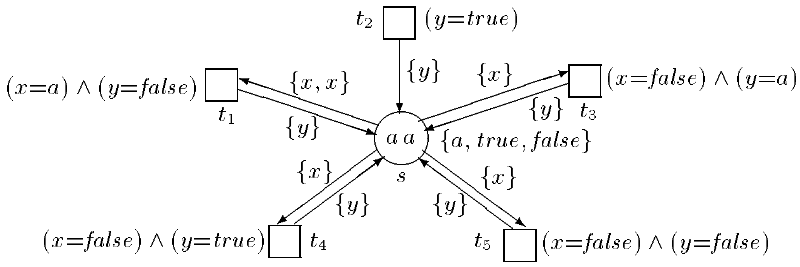

Fig. 3. A net $N(C)$ arising from the construction defined in the text

Note that any high-level net $N(C)$ arising from the above construction is finite, T-out-sequential and quasi-elementary. It is finite because there is only one place with (by assumption) finite type B, and moreover, there are $\mid \vdash_0 \mid$ transitions (where by assumption, $\vdash_0$ is a finite relation). It is T-out-sequential because, by construction, every transition t has exactly one output arc with label $\iota(t, s) = \{y\}$, i.e. a singleton set. It is quasi-elementary because every predicate of a transition specifies exactly the (only) enabling binding of that transition. Moreover, every multiset μ of B is at the same time also a marking of s, and therefore (since s is the only place) also a marking of $N(C)$. Furthermore:

Proposition 10. *Let μ and μ' be two multisets on B. Then $\mu \vdash \mu'$ in C iff μ' is reachable from μ in $N(C)$.*

Proof ($\Rightarrow$:) Assume $\mu \vdash \mu'$ in C. By Proposition 6($\Rightarrow$),

$$\mu = \mu_0 \rightsquigarrow \mu_1 \rightsquigarrow \ldots \rightsquigarrow \mu_m = \mu'$$

for some $m \geq 0$. For every $1 \leq j \leq m$, multiset μ_j arises from multiset μ_{j-1} by an application of some member, say $(\{c_1, \ldots, c_n\}, d)$, of $\vdash_0$ by the inference rule defined in Definition 5. According to that rule, $\{c_1, \ldots, c_n\}$ is a submultiset of μ_{j-1}. This implies, by the construction of $N(C)$, that the transition, say t_j,

[8] Note that $\vdash$ is infinite, and $\vdash_B$ too.

which corresponds to $(\{c_1, \ldots, c_n\}, d)$ is activated at marking μ_{j-1} in mode σ_j assigning to the variables $x_1, \ldots, x_n$ and y of t_j the values $c_1, \ldots, c_n$ and d, respectively. Hence t_j can occur in mode σ_j at marking μ_{j-1}, and the marking obtained after the occurrence of t_j is $(\mu_{j-1} \ominus \{c_1, \ldots, c_n\}) \oplus \{d\}$, which, by Definition 5, coincides with μ_j. Inductively, this proves that μ' is reachable from μ in $N(C)$.

($\Leftarrow$:) All arguments of part ($\Rightarrow$:) can be reverted using Proposition 6($\Leftarrow$) instead of ($\Rightarrow$). ■

4.2 Translating a high-level net into a linear constraint system

Let a finite, T-out-sequential, quasi-elementary high-level net N be given. We associate a linear constraint system $C(N)$ with N as follows. Assume (without loss of generality) that all values occurring in N are different from *true* and *false*. Assume further, again without loss of generality, that the place types of N are disjoint. Then define the set B as the union of all place types of N. (Alternatively, but equivalently, we could define B as the set of all $s(v)$ where s is a place name of N and v is an element of $\iota(s)$.)

The relation $\vdash_0$ is defined as follows. Every transition t of N generates exactly one element of $\vdash_0$. Let t be a transition of N. By T-out-sequentiality, t has exactly one place s' such that $\iota(t, s')$ is a singleton, say $\iota(t, s') = \{y\}$. Let $\{x_1, \ldots, x_n\}$ be the multiset sum of all multisets $\iota(s, t)$, for $s \in S$. Let σ be an enabling binding of t (by quasi-elementariness, σ exists and is unique). Then $(\{x_1, \ldots, x_n\})[\sigma]$ is a well-defined multiset of values and $y[\sigma]$ is a well-defined value. By definition, let $((\{x_1, \ldots, x_n\})[\sigma], y[\sigma]) \in \vdash_0$.

Example 3. Let N be the net shown in Figure 2. Then $C(N)$ is the linear constraint system generated by the following set B and relation $\vdash_0$:

$$B = \{0, 1, 2, 3, 4, 5, \mathit{true}, \mathit{false}\} \quad \text{and} \quad \vdash_0 = \{\ (\{0, 0, 0, 3\}, 5), (\{0, 1, 1, 3\}, 5)\}.$$

The two pairs of $\vdash_0$ come from the two transitions t_1 and t_2 of Figure 2. In order to check Proposition 11 below, consider, the inference $\{0, 1, 1, 2, 2, 3, 5\} \vdash \{2, 2, 5, 5\}$ coming from the second pair in $\vdash_0$. This corresponds to the occurrence of transition t_2 in the initial marking shown in Figure 2.

Note that any constraint system $C(N)$ arising from this construction is finitely generated. The set B of $C(N)$ is finite since, by assumption, all place types of N are finite. The relation $\vdash_0$ has only finitely many elements as a consequence of the sets T (of N) and B (of $C(N)$) being finite.

Let M be a marking of N associating a multiset of values with every place s of N. From M we construct a multiset $\mu(M)$ of B as the multiset sum of all $M(s)$, for all places s. Then:

Proposition 11. *Let M and M' be two markings of N. Then M' is reachable from M in N iff $\mu(M) \vdash \mu(M')$ in $C(N)$.*

Proof ($\Rightarrow$:) Assume M' can be reached from M in N. By the transition rule for high-level nets,

$$M = M_0 \stackrel{(t_1,\sigma_1)}{\longrightarrow} M_1 \stackrel{(t_2,\sigma_2)}{\longrightarrow} \ldots \stackrel{(t_m,\sigma_m)}{\longrightarrow} M_m = M'$$

where $M_{j-1} \stackrel{(t_j,\sigma_j)}{\longrightarrow} M_j$ means that t_j is activated in mode σ_j at M_{j-1} and M_j is the successor marking of M_{j-1} after occurrence of t_j in mode σ_j.[9] Suppose $\{x_1, \ldots, x_n\}$ and $\{y\}$ are the relevant multisets of t_j used in constructing the corresponding element of $\vdash_0$ (which generates $C(N)$). Then this element is

$$(\{x_1, \ldots, x_n\}[\sigma_j], y[\sigma_j]) \; = \; (\{c_1, \ldots, c_n\}, d).$$

By the fact that M_{j-1} activates t_j we get $\{c_1, \ldots, c_n\} \subseteq \mu(M_{j-1})$. Moreover, $\mu(M_j) = (\mu(M_{j-1}) \ominus \{c_1, \ldots, c_n\}) \oplus \{d\}$ because of the way M_j is calculated from M_{j-1} by the transition rule of high-level nets. Hence this inference rule can be applied and yields $\mu(M_j) \leadsto (\mu(M_{j-1})$. Inductively, we get $\mu(M) \leadsto \ldots \leadsto \mu(M')$, and an appeal to Proposition 6($\Leftarrow$) yields $\mu(M) \vdash \mu(M')$ as required.

($\Leftarrow$:) All arguments of part ($\Rightarrow$:) can be reverted using Proposition 6($\Rightarrow$) instead of ($\Leftarrow$) and ignoring all applications of rule $(\emptyset, \mathit{true}) \in \vdash_0$. ■

5 NP-completeness of linear entailment

We now show that the entailment problem in a finitely generated linear constraint system is NP-complete.

Proposition 12. *The problem*

ENTAIL: Given a finitely generated linear constraint system $\langle \mathcal{M}(B), \vdash \rangle$ and two finite multisets μ and μ' of B, does $\mu \vdash \mu'$?

is NP-complete.

Proof The proof is done in two steps: ENTAIL is NP-hard and ENTAIL is a member of NP.

NP-hardness is shown by reduction from the problem REACH-1. Let an instance of REACH-1, i.e. a finite T-out-sequential place/transition net $N = (S, T, W)$ and two markings M and M' of N, be given. To check whether M' is reachable from M in N, first turn N into a (quasi-elementary) high-level net, then apply the construction defined in Section 4.2[10] and finally check whether or not $\mu(M) \vdash \mu(M')$ in the resulting linear constraint system. By Proposition 11 this yields 'yes' if and only if M' is reachable from M in N. Hence the NP-hardness of REACH-1 implies the NP-hardness of ENTAIL.

Membership in NP is shown by reduction to the problem REACH-2. Let an instance of ENTAIL, i.e. a finitely generated linear constraint system $C = \langle \mathcal{M}(B), \vdash \rangle$ and two finite multisets μ and μ' of B, be given. A nondeterministic

[9] Actually we could omit the modes at this point since they are uniquely determined by quasi-elementariness.

[10] Both constructions can be done in polynomial time.

polynomial-time algorithm to test whether or not $\mu \vdash \mu'$ can be defined as follows. First apply the construction defined in Section 4.1 to obtain a quasi-elementary T-out-sequential high-level net $N(C)$ from C,[11] then apply the polynomial-time nondeterministic algorithm, which exists by the fact that REACH-2 is in NP, to check whether or not μ' is reachable from μ in $N(C)$. By Proposition 10, this is the case if and only if $\mu \vdash \mu'$ in C. Hence the fact that REACH-2 is in NP implies that ENTAIL is in NP. ■

6 Translating constraint systems with variables

In example 1, inference involves instantiation of variables in the relation $\vdash$. We did not introduce this mechanism in previous chapters; however the constraint system in example 1 can be transformed into a constraint system of the kind treated before by considering its relation $\vdash$ as a schema for representing $\vdash_0$: the pairs in $\vdash_0$ are obtained by instantiating the pairs of $\vdash$ on all possible elements of the domain (possibly typed, i.e. parted in places and persons). If the domain is finite, also $\vdash_0$ is finite.

When we have extra structure on the constraint system, such as the predicate-variable-structure of example 1, it can be reasonable to define a slightly different kind of translation into high-level nets. It may be the case that such a translation represents more realistically the distributed nature of the inference system. For instance, for the translation of example 1, instead of having a single place we could attach a different place to every predicate, having as value sets the range of individuals it applies to. A net of this type corresponding to example 1 is shown in Figure 4. Note that it satisfies the property of being T-out-sequential.

In translating example 1 into the net shown in Figure 4, we have assumed that there exist two (finite) sets $\mathbf{A} = \{1, 2, \ldots\}$ of theater places and $\mathbf{P} = \{A, B, \ldots\}$ of persons. The typed of places $s_1, \ldots, s_5$ in Figure 4 are constructed from these sets in a fashion appropriate to the predicates defined in example 1. The initial marking is M_0 shown in the following table:

$$
\begin{array}{llllll}
M_0: & s_1 \mapsto \{1,2\} & s_2 \mapsto \emptyset & s_3 \mapsto \emptyset & s_4 \mapsto \emptyset & s_5 \mapsto \emptyset \\
M_1: & s_1 \mapsto \{1,2\} & s_2 \mapsto \{A,B\} & s_3 \mapsto \emptyset & s_4 \mapsto \{A,B\} & s_5 \mapsto \emptyset \\
M_2: & s_1 \mapsto \emptyset & s_2 \mapsto \emptyset & s_3 \mapsto \{(1,A),(2,B)\} & s_4 \mapsto \{A,B\} & s_5 \mapsto \emptyset \\
M_2': & s_1 \mapsto \emptyset & s_2 \mapsto \emptyset & s_3 \mapsto \{(1,B),(2,A)\} & s_4 \mapsto \{A,B\} & s_5 \mapsto \emptyset \\
M_3: & s_1 \mapsto \emptyset & s_2 \mapsto \emptyset & s_3 \mapsto \emptyset & s_4 \mapsto \emptyset & s_5 \mapsto \{(A,1),(B,2)\} \\
M_3': & s_1 \mapsto \emptyset & s_2 \mapsto \emptyset & s_3 \mapsto \emptyset & s_4 \mapsto \emptyset & s_5 \mapsto \{(A,2),(B,1)\}.
\end{array}
$$

M_0 corresponds to the initial store specified in example 1, and transitions t_2 and t_3 model the making of phone calls and earning money, respectively (alternatively, the set of persons having made phone calls and having earned fifty dollars could be coded into the initial marking). Moreover, it can be checked that the two changes of the store specified in that example correspond to two occurrence sequences in

[11] This is again a polynomial-time construction.

the net, namely, in abbreviated form,

$$M_0 \xrightarrow{t_2,t_3} M_1 \xrightarrow{t_1,t_1} M_2 \xrightarrow{t_4,t_4} M_3 \text{ and } M_0 \xrightarrow{t_2,t_2,t_3,t_3} M_1 \xrightarrow{t_1,t_1} M_2' \xrightarrow{t_4,t_4} M_3',$$

respectively, leading to the same final stores specified above.

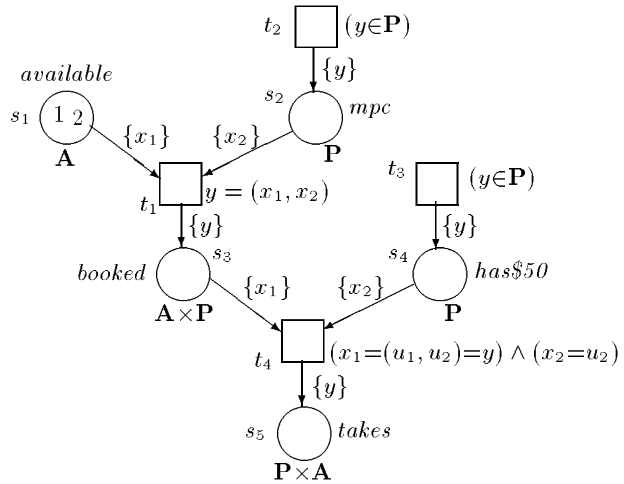

Fig. 4. A net corresponding to example 1

7 Conclusion

We have defined the notion of finitely generated linear constraint systems and we have shown that there is a complete correspondence between these systems and the class of T-out-sequential, quasi-elementary high-level nets. This interpretation allows to study the parallelism implicit in the notion of entailment, and its complexity: by using known results of Petri-net theory, we have proved that the entailment problem is NP-complete.

It is worth noting that this result depends critically on the nature of the generating entailment relation, i.e. on the property of inferring one single constraint at the time. Another natural way of defining the basic entailment relation would be as a relation from multiset to multisets, thus obtaining something similar to the concept of multiset rewriting. However, in this case the translation into T-out-sequential nets would not have been possible, and NP-completeness (probably) would not hold.

Acknowledgments. We are indebted to Raymond Devillers and four anonymous reviewers for comments.

References

1. B. Baumgarten: *Petri-Netze. Grundlagen und Anwendungen.* BI-Verlag, 1990.
2. E. Best, H. Fleischhack, W. Frączak, R.P. Hopkins, H. Klaudel and E. Pelz: A Class of Composable High Level Petri Nets. Proc. of 16th ICPN, LNCS Vol.935, Springer Verlag, 103–120, 1995.
3. F.S. de Boer and C. Palamidessi: Embedding as a Tool for Language Comparison. *Information and Computation* 108(1):128–157, 1994.
4. F.S. de Boer, C. Palamidessi and E. Best: Concurrent Constraint Programming with Information Removal. Proc. First Intl. Workshop on Concurrent Constraint Programming, Venice, 1–13, May 1995.
5. C. Brown and D. Gurr: A Categorical Linear Framework for Petri Nets. Proc. Fifth Annual IEEE Symposium on Logic in Computer Science, Philadelphia, IEEE Comput. Soc. Press, 208–218, 1990.
6. N. Carriero and D. Gelernter: Linda in Context. *Comm. ACM* 32(4):445–458, 1989.
7. S. Christensen, Y. Hirshfeld and F. Moller: Bisimulation Equivalence is Decidable for Basic Parallel Processes. Proc. CONCUR'93, LNCS Vol.715, Springer Verlag, 143-175, 1993.
8. U. Engberg and G. Winskel: Petri Nets as Models of Linear Logic. Proc. 15th Colloquium on Trees in Algebra and Programming, LNCS Vol.431, Springer-Verlag, 147–161, 1990.
9. J. Esparza: Petri Nets, Commutative Context-free Grammars and Basic Parallel Processes. Proc. FCT'95, LNCS Vol.965, Springer Verlag, 221–232, 1995.
10. H.J. Genrich: Predicate-transition Nets. In *Petri Nets: Central Models and their Properties*, LNCS Vol.254, Springer-Verlag, 207-247, 1987.
11. K. Jensen: *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use.* Volume 1. EATCS Monographs on Theoretical Computer Science, 1992.
12. D. Gelernter: Generative Communication in Linda. *ACM TOPLAS* 7(1):80–112, 1985.
13. U. Montanari and F. Rossi: Concurrent Semantics for Concurrent Constraint Programming via Contextual Nets. In V. Saraswat and P. Van Entenryck (eds) *Constraint Programming.* The MIT Press, 1995.
14. W. Reisig: *Petri Nets. An Introduction.* EATCS Monographs on Theoretical Computer Science Vol. 4, Springer-Verlag (1985).
15. W. Reisig: Petri Nets and Algebraic Specifications. *Theoretical Computer Science* 80, 1-34, 1991.
16. V.A. Saraswat and M. Rinard: Concurrent Constraint Programming. Proc. seventeenth ACM Symposium on Principles of Programming Languages, 232–245, ACM, New York, 1990.
17. V.A. Saraswat, M. Rinard, and P. Panangaden: Semantics Foundations of Concurrent Constraint Programming. Proc. eighteenth ACM Symposium on Principles of Programming Languages, ACM, New York, 1991.
18. D. Scott: Domains for Denotational Semantics. Proc. ICALP, Springer Verlag, 1982.
19. J. Vautherin: Parallel Systems Specification with Colored Petri Nets and Algebraic Specification. LNCS Vol. 266, Springer-Verlag, 1987.

A Multisets

In this appendix we recall some basic notions about multisets. Given a set X, a multiset of elements of X is a mapping $\mu\colon X \to \mathbb{N}$ where $\mathbb{N}$ is the set of natural numbers. We sometimes use extensional notation: for instance, the multiset μ such that $\mu(a) = 2$, $\mu(b) = 1$ and $\mu(c) = 0$ for $c \notin \{a, b\}$, will be represented by $\mu = \{a, a, b\}$. We denote by $\mathcal{M}(X)$ the set of the multisets on X, and by $\mathcal{M}_{\mathrm{f}}(X)$ the set of the *finite* multisets on X, i.e. those multisets μ for which the set $\{a \in X \mid \mu(a{\neq}0)\}$ is finite. If μ is finite then the number $|\mu| = \sum_{a \in X} \mu(z)$ is well-defined; it is called the *cardinality* of μ. A useful notion is *multiset inclusion*, denoted by $\subseteq$, and defined as $\mu' \subseteq \mu$ iff $\forall a \in X.\ \mu'(a) \leq \mu(a)$.
The *multiset sum* is a (total) function $\oplus\colon \mathcal{M}(X) \times \mathcal{M}(X) \to \mathcal{M}(X)$ defined as

$$(\mu \oplus \mu')(a) = \mu(a) + \mu'(a).$$

The *multiset difference* is a (partial) function $\ominus : \mathcal{M}(X) \times \mathcal{M}(X) \to \mathcal{M}(X)$. If $\mu' \subseteq \mu$ then $\mu \ominus \mu'$ is defined by putting, for each $a \in X$,

$$(\mu \ominus \mu')(a) = \mu(a) - \mu'(a).$$

Let $b \in X$ and μ a multiset. The *restriction* $\mu|_b$ is defined as the multiset

$$\mu|_b(a) = \begin{cases} \mu(a) & \text{if } a = b \\ 0 & \text{if } a \neq b. \end{cases}$$