

## Chapter 6

# Guarded Fragments with Counting

### 6.1 The guarded fragment

We observed in Ch. 5 that multimodal logic  $\mathcal{M}$  can be thought of as the fragment of first-order logic, over signatures of unary and (one or more) binary predicates, in which quantification is restricted to the patterns  $\forall v(r(u, v) \rightarrow \psi(v))$  and  $\exists v(r(u, v) \wedge \psi(v))$ . This idea can be generalized.

We again fix some purely relational signature  $\Sigma$ , but this time with predicates of any arity, and we again denote by  $\mathcal{L}$  the language of first-order logic over  $\Sigma$ . The *guarded fragment* of first-order logic,  $\mathcal{G}$ , is defined to be the smallest set of  $\mathcal{L}$ -formulas satisfying the following conditions:

1. every atomic formula is in  $\mathcal{G}$ ;
2.  $\mathcal{G}$  is closed under Boolean connectives;
3. if  $\psi \in \mathcal{G}$  with  $\text{Vars}(\psi) = \bar{x}, \bar{y}$ , and  $p$  is a predicate of the same arity as  $\bar{x}, \bar{y}$ , then  $\forall \bar{x}(p(\bar{x}, \bar{y}) \rightarrow \psi) \in \mathcal{G}$  and  $\exists \bar{x}(p(\bar{x}, \bar{y}) \wedge \psi) \in \mathcal{G}$ ;
4. if  $\psi \in \mathcal{G}$ , and  $\psi$  has at most one free variable, then  $\forall x\psi \in \mathcal{G}$  and  $\exists x\psi \in \mathcal{G}$ .

For  $k > 1$ , we denote by  $\mathcal{G}_{\approx}^k$  the subset of  $\mathcal{G}$  involving at most the variables  $x_1, \dots, x_k$ . Thus,  $\mathcal{G}^k = \mathcal{G} \cap \mathcal{L}_{\approx}^k$ , for all  $k > 0$ .  $\mathcal{L}_{\approx}^k$  not involving the equality predicate. We call  $\mathcal{G}^k$  the *k-variable guarded fragment*. Thus, multi-modal logic  $\mathcal{M}$  is a proper subset of the 2-variable guarded fragment  $\mathcal{G}^2$ .

**Lemma 6.1.** *The guarded fragment has the finite model property.*

Thus, the problems  $\mathcal{G}$ -Sat and  $\mathcal{G}$ -Fin-Sat coincide, as do  $\mathcal{G}^k$ -Sat and  $\mathcal{G}^k$ -Fin-Sat, for all  $k > 0$ .

**Theorem 6.1.** *The problem  $\mathcal{G}$ -Sat is 2-EXPTIME complete.*

**Theorem 6.2.** *For every  $k > 0$ , the problem  $\mathcal{G}^k$ -Sat is EXPTIME complete.*

## 6.2 The guarded fragment with counting

We have seen that adding counting quantifiers to the multi-modal logic,  $\mathcal{M}$ , to form graded multi-modal logic,  $\mathcal{GK}$ , has no effect on the complexity of satisfiability. The obvious next question, therefore, is what happens when counting quantifiers are added to the guarded fragment.

In the sequel, we restrict consideration to a purely relational signature of 0-ary, unary and binary predicates. If  $r$  is any binary predicate (including  $\approx$ ), we call an atomic formula having either of the forms  $r(x, y)$  or  $r(y, x)$  a *guard-atom*. The two-variable guarded fragment with counting quantifiers,  $\mathcal{GC}^2$ , can then be defined as the smallest set of formulas satisfying the following conditions:

1.  $\mathcal{GC}^2$  contains all atomic formulas and is closed under Boolean combinations;
2. if  $\varphi$  is a formula of  $\mathcal{GC}^2$  with at most one free variable, and  $u$  is a variable (i.e. either  $x$  or  $y$ ), then the formulas  $\forall u\varphi$  and  $\exists u\varphi$  are in  $\mathcal{GC}^2$ ;
3. if  $\varphi$  is a formula of  $\mathcal{GC}^2$ ,  $\gamma$  a guard-atom,  $u$  a variable, and  $Q$  any of the quantifiers  $\exists$ ,  $\exists_{\leq C}$ ,  $\exists_{\geq C}$ ,  $\exists_{=C}$  (for  $C > 0$ ), then the formulas  $\forall u(\gamma \rightarrow \varphi)$ ,  $Qu(\gamma \wedge \varphi)$  and  $Qu\gamma$  are in  $\mathcal{GC}^2$ .

According to the above syntax, the non-counting quantifiers  $\exists$  and  $\forall$  may apply without restriction to formulas with at most one free variable; however, they may apply to formulas with two free variables only in the presence of a guard-atom. By contrast, the counting quantifiers  $\exists_{\leq C}$ ,  $\exists_{\geq C}$ ,  $\exists_{=C}$  may only ever apply in the presence of a guard atom (which by definition has two free variables). Note in particular that the formula  $\exists_{=1}xp(x)$  is not in  $\mathcal{GC}^2$ . In fact, the next lemma shows that no formula of  $\mathcal{GC}^2$  can force a predicate  $p$  to be uniquely instantiated in its models.

**Lemma 6.2.** *Let  $\varphi$  be a formula of  $\mathcal{GC}^2$  with signature  $\Sigma$  (so that  $\Sigma$  has no individual constants),  $\mathfrak{A}$  a structure interpreting  $\Sigma$ , and  $I$  a nonempty set. For  $i \in I$ , let  $\mathfrak{A}_i$  be a copy of  $\mathfrak{A}$ , with the domains  $A_i$  pairwise disjoint. If  $\varphi$  is satisfied in  $\mathfrak{A}$ , then it is satisfied in the structure  $\mathfrak{A}'$  with domain  $A' = \bigcup_{i \in I} A_i$  and interpretations  $\sigma^{\mathfrak{A}'} = \bigcup_{i \in I} \sigma^{\mathfrak{A}_i}$  for every  $\sigma \in \Sigma$ .*

*Proof.* If  $\theta : \{x, y\} \rightarrow A$  is any variable assignment, and  $i \in I$ , let  $\theta_i$  be the variable assignment which maps  $x$  and  $y$  to the corresponding elements in  $A_i \subseteq A'$ . A routine structural induction on  $\varphi$  shows that  $\mathfrak{A} \models_{\theta} \varphi$  if and only if, for some (= for all)  $i \in I$ ,  $\mathfrak{A}' \models_{\theta_i} \varphi$ .  $\square$

It follows immediately that, if a formula of  $\mathcal{GC}^2$  has a finite model, then it has arbitrarily large finite models, and indeed infinite models.

**Lemma 6.3.** *Let  $\varphi$  be a  $\mathcal{GC}^2$ -formula. We can compute, in time bounded by a*

polynomial function of  $\|\varphi\|$ , a formula

$$\psi = \forall x \alpha \wedge \bigwedge_{1 \leq h \leq l} \forall x \forall y (e_h(x, y) \rightarrow (\beta_h \vee x \approx y)) \wedge \bigwedge_{1 \leq i \leq m} \forall x \exists =_{C_i} y (f_i(x, y) \wedge x \not\approx y) \quad (6.1)$$

such that: (i)  $\alpha$  is a quantifier-free formula not involving  $\approx$  with  $x$  as its only variable; (ii)  $l$  and  $m$  are positive integers; (iii) for all  $h$  ( $1 \leq h \leq l$ ),  $e_h$  is a binary predicate other than  $\approx$ , and  $\beta_h$  is a quantifier-free formula not involving  $\approx$  with  $x$  and  $y$  as its only variables; (iv) for all  $i$  ( $1 \leq i \leq m$ ),  $C_i$  is a positive integer, and  $f_i$  is a binary predicate other than  $\approx$ ; (v)  $\varphi$  is satisfiable if and only if  $\psi$  is satisfiable; (vi)  $\varphi$  is finitely satisfiable if and only if  $\psi$  is finitely satisfiable.

*Proof.* We proceed exactly as for Lemma 4.1. Note that, if  $\varphi$  is finitely satisfiable, then it has models over arbitrarily large finite domains. Therefore, we may without loss of generality assume that the size of any model of  $\varphi$  of interest is greater than any quantifier subscript mentioned in  $\varphi$ .

The only difficulty is to show that the conjuncts of the form  $\forall u \forall v \chi$  generated in Stage 1 of the procedure described there can be made guarded. To see that this is so, consider the treatment of a subformula  $\theta(u) = \exists \leq_D v \chi$  of  $\varphi_0$ . Since  $\varphi_0$  is guarded,  $\chi$  must be of the form  $\gamma \wedge \eta$ , where  $\gamma$  is an atomic formula  $g(u, v)$  or  $g(v, u)$ . Again, we define  $\varphi_1 := \varphi_0[p(u)/\theta(u)]$ , where  $p$  is a new unary predicate, and

$$\begin{aligned} \psi_1 := \forall u \exists =_D v r_1(u, v) \wedge \forall u \exists =_{D+1} v r_2(u, v) \wedge \\ \forall u \forall v (p(u) \rightarrow (\theta \rightarrow r_1(u, v))) \wedge \\ \forall u \forall v (\neg p(u) \rightarrow (r_2(u, v) \rightarrow \theta)), \end{aligned}$$

arguing, just as in Lemma 4.1 that  $\psi_0$  and  $\varphi_1 \wedge \psi_1$  are satisfiable over sufficiently large domains. Now consider in more detail  $\psi_1$ . Its latter two conjuncts are, as they stand, not guarded. However, noting that  $\theta$  is  $(\gamma \wedge \eta)$ , we see that these conjuncts are in fact logically equivalent to the guarded formula

$$\forall u \forall v (\gamma \rightarrow ((p(u) \wedge \eta) \rightarrow r_1(u, v))) \wedge \forall u \forall v (\neg r_2(u, v) \rightarrow (p(u) \rightarrow (\gamma \wedge \eta))).$$

The other cases are dealt with similarly, and we obtain the desired formula  $\psi$ .  $\square$

To show that the (finite) satisfiability problem for  $\mathcal{GC}^2$  is in EXPTIME, it therefore suffices to consider only formulas  $\varphi$  of the form (6.1). Furthermore, we may assume without loss of generality that no 0-ary predicates (proposition letters) occur in  $\varphi$ , since we can consider each of the (at most  $2^{\|\varphi\|}$ ) truth-value assignments to the 0-ary predicates of  $\varphi$  in turn, replacing each 0-ary predicate with  $\top$  or  $\perp$  according to its truth-value in the considered assignment.

Accordingly, fix  $\varphi$  to be some formula of the form (6.1) over a signature of unary and binary predicates. Set  $C = \max_{1 \leq i \leq m} C_i$ , and let  $\Sigma$  be the signature of  $\varphi$  together with  $\log((mC)^2 + 1)$  (rounded up) new unary predicates. Thus,  $|\Sigma|$  is bounded by a polynomial (actually, linear) function of  $\|\varphi\|$ . Since  $\Sigma$  is the only signature we shall be concerned with in the sequel, we generally suppress reference to it. Thus, ‘predicate’ henceforth means ‘predicate in  $\Sigma \cup \{\approx\}$ ’, ‘structure’ henceforth means ‘structure interpreting  $\Sigma$ ’, and so on. We keep the meanings of the symbols

$$\Sigma, \varphi, \alpha, l, e_1, \dots, e_l, \beta_1, \dots, \beta_l, m, C_1, \dots, C_m, C, f_1, \dots, f_m$$

fixed throughout this paper. The predicates  $f_1, \dots, f_m$  will play a key role in the ensuing argument; we refer to them as the *counting predicates*. There is no restriction on these predicates’ occurring in other parts of  $\varphi$ .

Again, we can regard the signature  $\Sigma$  as a classified signature ...

Let the 1-types (over  $\Sigma$ ) be enumerated in some order as the sequence

$$\Pi = \pi_0, \dots, \pi_{P-1}.$$

Evidently,  $P$  is a power of 2, so  $p = \log P$  is an integer. (Actually,  $p = |\Sigma|$ .) Now let  $s$  be any bit string ( $0 \leq |s| \leq p$ ), and denote the string of length 0 by  $\epsilon$ . We inductively define the sub-sequence  $\Pi_s$  of  $\Pi$  by setting  $\Pi_\epsilon$  to be the whole of  $\Pi$ , and setting  $\Pi_{s0}$  and  $\Pi_{s1}$  to be the left and right halves of  $\Pi_s$ , respectively. Formally:

$$\Pi_\epsilon = \pi_0, \dots, \pi_{P-1},$$

and if  $\Pi_s = \pi_j, \dots, \pi_{k-1}$ , with  $|s| < p$ ,

$$\Pi_{s0} = \pi_j, \dots, \pi_{\frac{k+j}{2}-1}$$

$$\Pi_{s1} = \pi_{\frac{k+j}{2}}, \dots, \pi_{k-1}.$$

Thus, if  $|s| = p$ , then  $\Pi_s$  is a one-element sequence  $\pi_j$ , where  $j$  is the integer ( $0 \leq j < P$ ) encoded by the bit-string  $s$  in the usual way. To avoid clumsy circumlocutions, we occasionally equivocate between bit-strings and the integers they encode, thus, for instance, writing  $\pi_s$  instead of  $\pi_j$  in this case. But we will only ever write  $\pi_s$  if  $|s| = p$ . In addition, we occasionally for convenience treat sequences as if they were sets, writing, for instance,  $\pi \in \Pi_s$ .

We now use the sequences  $\Pi_s$  to define sets of invertible message-types indexed by bit-strings as follows. Let  $\Lambda$  be the set of all invertible message-types. If  $\pi$  is any 1-type, and  $s$  is any bit-string such that  $|s| \leq p$ , let

$$\Lambda_{\pi,s} = \{\lambda \in \Lambda \mid \text{tp}_1(\lambda) = \pi \text{ and } \text{tp}_2(\lambda) \in \Pi_s\}.$$

Thus, the  $\Lambda_{\pi,s}$  are sets of invertible message-types identified purely by their terminal 1-types. Except in very special cases, these sets will contain more than one member, even when  $|s| = p$ . However, for chromatic structures, we have the following important fact.

**Lemma 6.4.** *Let  $\mathfrak{A}$  be a chromatic structure,  $a \in A$ ,  $\pi = \text{tp}^{\mathfrak{A}}[a]$ , and  $s$  a bit-string with  $|s| = p$ . Then there can be at most one element  $b \in A \setminus \{a\}$  such that  $\text{tp}^{\mathfrak{A}}[a, b] \in \Lambda_{\pi, s}$ .*

*Proof.* Any two such elements would be connected by a chain of two invertible message-types, and would both have 1-type  $\pi_s$ .  $\square$

Finally, we use bit strings to index sequences of 2-types that are not invertible message-types. Again, fix any 1-type  $\pi$ , and consider the set of non-invertible message-types  $\mu$  such that  $\text{tp}_1(\mu) = \pi$ . Let these be enumerated in some way as a sequence

$$\mu_{\pi, 0}, \dots, \mu_{\pi, R-1}.$$

Furthermore, consider the set of silent 2-types  $\mu$  such that  $\text{tp}_1(\mu) = \pi$ . Let these be enumerated in some way as a sequence

$$\mu_{\pi, R}, \dots, \mu_{\pi, Q-1}.$$

Thus, the sequence

$$M_{\pi} = \mu_{\pi, 0}, \dots, \mu_{\pi, Q-1}$$

is an enumeration of precisely those 2-types  $\tau$  such that  $\text{tp}_1(\tau) = \pi$  and  $\tau^{-1}$  is not a message-type. Evidently,  $R$  and  $Q$  are independent of the choice of  $\pi$ ; moreover,  $Q$  is a power of 2, so  $q = \log Q$  is an integer. (We remark that  $R$  is not a power of 2.) Let  $t$  be any bit string ( $0 \leq |t| \leq q$ ). We inductively define the sub-sequence  $M_{\pi, t}$  of  $M_{\pi}$  by setting  $M_{\pi, \epsilon}$  to be the whole of  $M_{\pi}$ , and setting  $M_{\pi, t0}$  and  $M_{\pi, t1}$  to be the left and right halves of  $M_{\pi, t}$ , respectively. Formally:

$$M_{\pi, \epsilon} = \mu_{\pi, 0}, \dots, \mu_{\pi, Q-1},$$

and if  $M_{\pi, t} = \mu_{\pi, j}, \dots, \mu_{\pi, k-1}$ , with  $|t| < q$ ,

$$\begin{aligned} M_{\pi, t0} &= \mu_{\pi, j}, \dots, \mu_{\pi, \frac{k+j}{2}-1} \\ M_{\pi, t1} &= \mu_{\pi, \frac{k+j}{2}}, \dots, \mu_{\pi, k-1}. \end{aligned}$$

Thus, if  $|t| = q$ , then  $M_{\pi, t}$  is a one-element sequence  $\mu_{\pi, j}$ , where  $j$  is the integer ( $0 \leq j < Q$ ) encoded by the bit-string  $t$  in the usual way. Again we may for convenience write  $\mu_{\pi, t}$  instead of  $\mu_{\pi, j}$  in this case, but here too we only ever write  $\mu_{\pi, t}$  if  $|t| = q$ .

## 6.3 Spectra and tallies

The approach taken here involves identifying various configurational properties of elements in finite structures. These we now proceed to define. We continue to use the symbols introduced in Section 6.2 with their advertised meanings. In particular,  $f_1, \dots, f_m$  are the counting predicates occurring in the formula  $\varphi$  given in (6.1), and the integers  $C_1, \dots, C_m$  are the corresponding numerical quantifier subscripts.

Let  $\mathfrak{A}$  be a finite structure,  $a \in A$ , and  $\pi = \text{tp}^{\mathfrak{A}}[a]$ , and suppose  $\mathfrak{A} \models \varphi$ . Evidently, for all  $i$  ( $1 \leq i \leq m$ ), there are exactly  $C_i$  elements  $b \in A \setminus \{a\}$  such that  $\mathfrak{A} \models f_i[a, b]$ :

$$C_i = |\{b \in A \setminus \{a\} : \mathfrak{A} \models f_i[a, b]\}|.$$

Now, for any bit-string  $s$  ( $0 \leq |s| \leq p$ ), define the  $s$ -spectrum of  $a$  in  $\mathfrak{A}$ , denoted  $\text{sp}_s^{\mathfrak{A}}[a]$ , to be the  $m$ -element vector whose  $i$ th component ( $1 \leq i \leq m$ ) is the number of elements  $b \in A \setminus \{a\}$  such that  $\mathfrak{A} \models f_i[a, b]$  and  $\text{tp}^{\mathfrak{A}}[a, b] \in \Lambda_{\pi, s}$ :

$$(\text{sp}_s^{\mathfrak{A}}[a])_i = |\{b \in A \setminus \{a\} : \mathfrak{A} \models f_i[a, b] \text{ and } \text{tp}^{\mathfrak{A}}[a, b] \in \Lambda_{\pi, s}\}|.$$

Similarly, for any bit-string  $t$  ( $0 \leq |t| \leq q$ ), define the  $t$ -tally of  $a$  in  $\mathfrak{A}$ , denoted  $\text{tl}_t^{\mathfrak{A}}[a]$ , to be the  $m$ -element vector whose  $i$ th component is the number of elements  $b \in A \setminus \{a\}$  such that  $\mathfrak{A} \models f_i[a, b]$  and  $\text{tp}^{\mathfrak{A}}[a, b] \in M_{\pi, t}$ :

$$(\text{tl}_t^{\mathfrak{A}}[a])_i = |\{b \in A \setminus \{a\} : \mathfrak{A} \models f_i[a, b] \text{ and } \text{tp}^{\mathfrak{A}}[a, b] \in M_{\pi, t}\}|.$$

Henceforth, by *vector*, we shall always mean “ $m$ -dimensional vector over  $\mathbb{N}$ ”, with indices running from 1 to  $m$ . We denote the vector  $(C_1, \dots, C_m)$  by  $\mathbf{C}$  and the  $m$ -dimensional zero vector  $(0, \dots, 0)$  by  $\mathbf{0}$ . If  $\mathbf{u}$  and  $\mathbf{v}$  are vectors, we write  $\mathbf{u} \leq \mathbf{v}$  if every component of  $\mathbf{u}$  is less than or equal to the corresponding component of  $\mathbf{v}$ ; we write  $\mathbf{u} < \mathbf{v}$  if  $\mathbf{u} \leq \mathbf{v}$  and  $\mathbf{u} \neq \mathbf{v}$ . Similarly for  $\geq$  and  $>$ . Recalling further that  $C = \max_{1 \leq i \leq m} C_i$ , the number of vectors  $\mathbf{u}$  such that  $\mathbf{u} \leq \mathbf{C}$  is evidently bounded by  $(C+1)^m$ , and hence by an exponential function of  $\|\varphi\|$ . Moreover, the  $s$ -spectrum and  $t$ -tally of  $a$  is always a vector  $\leq \mathbf{C}$ . Lastly, given any 2-type  $\tau$ , we write  $\mathbf{C}_\tau$  for the vector whose  $i$ th component is given by:

$$(\mathbf{C}_\tau)_i = \begin{cases} 1 & \text{if } f_i(x, y) \in \tau, \\ 0 & \text{otherwise.} \end{cases} \quad (6.2)$$

To better understand this apparatus, let  $\mathfrak{A}$ ,  $a$ ,  $\pi$  be as above, and consider first the case where  $s$  and  $t$  are the empty string  $\epsilon$ . Since  $\Lambda_{\pi, \epsilon}$  is the set of invertible message-types  $\lambda$  such that  $\text{tp}_1(\lambda) = \pi$ ,  $\text{sp}_\epsilon^{\mathfrak{A}}[a]$  is simply the vector whose  $i$ th component records the number of elements  $b$  to which  $a$  sends a message of *invertible* type containing the atom  $f_i(x, y)$ . Likewise,  $\text{tl}_\epsilon^{\mathfrak{A}}[a]$  is the vector whose  $i$ th component records the number of elements  $b$  to which  $a$  sends a message of *non-invertible* type containing the atom  $f_i(x, y)$ . If  $0 < |s| \leq p$ , then  $\text{sp}_s^{\mathfrak{A}}[a]$  is obtained in the same way as  $\text{sp}_\epsilon^{\mathfrak{A}}[a]$ , except that we discount all messages whose type is not a member of  $\Lambda_{\pi, s}$ . Likewise, if  $0 < |t| \leq q$ , then  $\text{tl}_t^{\mathfrak{A}}[a]$  is obtained in the same way as  $\text{tl}_\epsilon^{\mathfrak{A}}[a]$ , except that we discount all messages whose type is not a member of  $M_{\pi, t}$ .

**Lemma 6.5.** *Suppose  $\mathfrak{A}$  is a model of  $\varphi$ . Let  $a \in A$ ,  $\pi = \text{tp}^{\mathfrak{A}}[a]$ , and  $s, t$  be bit-strings such that  $|s| < p$  and  $|t| < q$ . Then*

$$\text{sp}_\epsilon^{\mathfrak{A}}[a] + \text{tl}_\epsilon^{\mathfrak{A}}[a] = \mathbf{C} \quad (6.3)$$

$$\text{sp}_{s_0}^{\mathfrak{A}}[a] + \text{sp}_{s_1}^{\mathfrak{A}}[a] = \text{sp}_s^{\mathfrak{A}}[a] \quad (6.4)$$

$$\text{tl}_{t_0}^{\mathfrak{A}}[a] + \text{tl}_{t_1}^{\mathfrak{A}}[a] = \text{tl}_t^{\mathfrak{A}}[a]. \quad (6.5)$$

*Proof.* Immediate.  $\square$

Thus, while  $\Lambda_{\pi,s}$  is the *union* of the sets  $\Lambda_{\pi,s_0}$  and  $\Lambda_{\pi,s_1}$ ,  $\text{sp}_s^{\mathfrak{A}}[a]$  is the *vector sum* of the spectra  $\text{sp}_{s_0}^{\mathfrak{A}}[a]$  and  $\text{sp}_{s_1}^{\mathfrak{A}}[a]$ ; similarly for tallies.

As the strings  $s$  and  $t$  get longer, the sets  $\Lambda_{\pi,s}$  and the sequences  $M_{\pi,t}$  get smaller, and so the vectors  $\text{sp}_s^{\mathfrak{A}}[a]$  and  $\text{tl}_t^{\mathfrak{A}}[a]$  become, as it were, more selective in the information they record. It is therefore instructive to consider what happens to  $s$ -spectra and  $t$ -tallies when  $s$  and  $t$  have maximal length. The latter case is the easier to understand, and so we consider it first. If  $|t| = q$ , then  $M_{\pi,t}$  by construction contains just one 2-type,  $\mu = \mu_{\pi,t}$ , which is either a non-invertible message-type, or else a silent 2-type. If  $\mu$  is a non-invertible message-type, then, since  $\mathfrak{A} \models \varphi$ , there can be only finitely many elements  $b \in A \setminus \{a\}$  such that  $\text{tp}^{\mathfrak{A}}[a, b] = \mu$ . Let this number be  $n$ . Evidently:

$$\text{tl}_t^{\mathfrak{A}}[a] = n\mathbf{C}_{\mu}. \quad (6.6)$$

On the other hand, if  $\mu$  is a silent 2-type, then  $\text{tl}_t^{\mathfrak{A}}[a] = \mathbf{C}_{\mu} = \mathbf{0}$ .

Turning now to  $s$ -spectra with  $|s| = p$ , recall that  $\Lambda_{\pi,s}$  in general has more than one element. However, Lemma 6.4 tells us that, in a *chromatic* model, no element may send more than one message whose type is in  $\Lambda_{\pi,s}$ . This enables us to establish the following simple result, which will be useful in the sequel:

**Lemma 6.6.** *Suppose that  $\mathfrak{A}$  is a chromatic model of  $\varphi$ . Let  $a \in A$ ,  $\pi$  be a 1-type, and  $s$  be a bit-string with  $|s| = p$ . If  $\text{tp}^{\mathfrak{A}}[a] = \pi$  and  $\text{sp}_s^{\mathfrak{A}}[a] > \mathbf{0}$ , then there exists  $\lambda \in \Lambda_{\pi,s}$  with  $\text{sp}_s^{\mathfrak{A}}[a] = \mathbf{C}_{\lambda}$  such that  $a$  sends a message of type  $\lambda$  to some  $b \in A \setminus \{a\}$ . Conversely, if there exists  $\lambda \in \Lambda_{\pi,s}$  such that  $a$  sends a message of type  $\lambda$  to some  $b \in A \setminus \{a\}$ , then  $\text{tp}^{\mathfrak{A}}[a] = \pi$  and  $\text{sp}_s^{\mathfrak{A}}[a] = \mathbf{C}_{\lambda}$ .*

*Proof.* Suppose  $\text{tp}^{\mathfrak{A}}[a] = \pi$  and  $\text{sp}_s^{\mathfrak{A}}[a] > \mathbf{0}$ . Then there exists  $b \in A \setminus \{a\}$  such that  $\text{tp}^{\mathfrak{A}}[a, b] \in \Lambda_{\pi,s}$ . By Lemma 6.4, this  $b$  is unique, so, letting  $\lambda = \text{tp}^{\mathfrak{A}}[a, b]$ , we have  $\text{sp}_s^{\mathfrak{A}}[a] = \mathbf{C}_{\lambda}$  as required. Conversely, suppose  $a$  sends a message of type  $\lambda \in \Lambda_{\pi,s}$  to some element  $b \in A \setminus \{a\}$ . Certainly, then,  $\text{tp}^{\mathfrak{A}}[a] = \pi$ , and again, by Lemma 6.4, this  $b$  is the only element in  $A \setminus \{a\}$  to which  $a$  sends a message having any type in  $\Lambda_{\pi,s}$ ; it follows that  $\text{sp}_s^{\mathfrak{A}}[a] = \mathbf{C}_{\lambda}$  as required.  $\square$

## 6.4 Transformation into a constraint satisfaction problem

In the sequel, we take  $\pi$  to vary over the set of 1-types,  $\lambda$  to vary over the set of invertible message-types,  $s$  to vary over the set of bit-strings of length at most  $p$ ,  $t$  to vary over the set of bit-strings of length at most  $q$ , and  $\mathbf{u}$ ,  $\mathbf{v}$  and  $\mathbf{w}$  to vary over the set of vectors  $\leq \mathbf{C}$ . (Similarly for their primed counterparts  $\pi'$ ,  $\lambda'$ ,  $s'$ ,  $t'$ ,  $\mathbf{u}'$ ,  $\mathbf{v}'$  and  $\mathbf{w}'$ .) We refer to these sets as the *standard ranges* of the respective letters; they are summarized in Table 6.1. Occasionally, additional

Symbol	Standard range
$\pi, \pi'$	all 1-types
$\lambda, \lambda'$	all invertible message-types
$s, s'$	all bit-strings of length at most $p$
$t, t'$	all bit-strings of length at most $q$
$\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{u}', \mathbf{v}', \mathbf{w}'$	all vectors $\leq \mathbf{C}$

Table 6.1: Standard ranges of symbols used as indices

restrictions will be imposed on these ranges.

Now let  $V$  be the set whose elements are the following (distinct) symbols, where the indices  $\lambda, \pi, s, t, \mathbf{u}, \mathbf{v}, \mathbf{w}$  vary over their standard ranges:

$$x_\lambda, \quad y_{\pi,s,\mathbf{u}}, \quad z_{\pi,t,\mathbf{u}}, \\ \hat{y}_{\pi,s,\mathbf{v},\mathbf{w}} \text{ whenever } |s| < p, \quad \hat{z}_{\pi,t,\mathbf{v},\mathbf{w}} \text{ whenever } |t| < q.$$

The symbols  $\hat{y}_{\pi,s,\mathbf{v},\mathbf{w}}$  and  $\hat{z}_{\pi,t,\mathbf{v},\mathbf{w}}$  are not defined when  $|s| = p$  and  $|t| = q$ . The cardinality of  $V$  is evidently bounded by an exponential function of  $\|\varphi\|$ . We impose some arbitrary order on  $V$ , and refer to its elements as *variables*. If  $U$  is a non-empty set of variables, enumerated, in the imposed order, as  $\{u_1, \dots, u_k\}$ , let  $\sum U$  denote the term  $u_1 + \dots + u_k$ ; if  $U$  is the empty set, let  $\sum U$  denote the (constant) term 0. In the sequel, we take a *constraint* to be an equation or inequality involving arithmetical terms over  $V$ , or a conditional statement formed from two such inequalities. A *solution* of a set of constraints over some numerical domain  $\mathbb{D}$  is simply an assignment  $\theta : V \rightarrow \mathbb{D}$  under which all the constraints in question evaluate (in the obvious way) to true. Using this apparatus, we proceed to construct, given the formula  $\varphi$  in (6.1), a set  $\mathcal{E}$  of constraints. We prove below that  $\mathcal{E}$  has a solution over  $\mathbb{N}$  if and only if  $\varphi$  is finitely satisfiable.

To motivate this construction, suppose, for the moment, that  $\mathfrak{A}$  is a finite model of  $\varphi$ . We may then define the assignment  $\theta : V \rightarrow \mathbb{N}$ , which we may think of as the ‘intended’ assignment relative to  $\mathfrak{A}$ , as follows. If  $\pi$  is a 1-type, write  $A_\pi$  to denote the set of elements of  $A$  having 1-type  $\pi$  in  $\mathfrak{A}$ :

$$A_\pi = \{a \in A \mid \text{tp}^{\mathfrak{A}}[a] = \pi\}.$$

Now, for any  $\lambda$  in its standard range, let  $\theta(x_\lambda)$  be the number of elements of  $A$  sending any message of type  $\lambda$  to some other element. For any  $\pi, s, t, \mathbf{u}$  in their standard ranges, let  $\theta(y_{\pi,s,\mathbf{u}})$  be the number of elements of  $A_\pi$  having  $s$ -spectrum  $\mathbf{u}$ , and let  $\theta(z_{\pi,t,\mathbf{u}})$  be the number of elements of  $A_\pi$  having  $t$ -tally  $\mathbf{u}$ . Finally, for any  $\pi, s, t, \mathbf{v}, \mathbf{w}$  in their standard ranges, with  $|s| < p$  and  $|t| < q$ , let  $\theta(\hat{y}_{\pi,s,\mathbf{v},\mathbf{w}})$  be the number of elements of  $A_\pi$  having  $s_0$ -spectrum  $\mathbf{v}$  and  $s_1$ -spectrum  $\mathbf{w}$ , and let  $\theta(\hat{z}_{\pi,t,\mathbf{v},\mathbf{w}})$  be the number of elements of  $A_\pi$  having  $t_0$ -tally  $\mathbf{v}$  and  $t_1$ -tally  $\mathbf{w}$ . This assignment is summarized in Table 6.2.

Variable	Value in $\mathbb{N}$ under $\theta$
$x_\lambda$	$ \{a \in A : \text{there exists } b \in A \setminus \{a\} \text{ such that } \text{tp}^{\mathfrak{A}}[a, b] = \lambda\} $
$y_{\pi, s, \mathbf{u}}$	$ \{a \in A_\pi : \text{sp}_s^{\mathfrak{A}}[a] = \mathbf{u}\} $
$z_{\pi, t, \mathbf{u}}$	$ \{a \in A_\pi : \text{tl}_t^{\mathfrak{A}}[a] = \mathbf{u}\} $
$\hat{y}_{\pi, s, \mathbf{v}, \mathbf{w}}$	$ \{a \in A_\pi : \text{sp}_{s0}^{\mathfrak{A}}[a] = \mathbf{v} \text{ and } \text{sp}_{s1}^{\mathfrak{A}}[a] = \mathbf{w}\} $ , whenever $ s  < p$
$\hat{z}_{\pi, t, \mathbf{v}, \mathbf{w}}$	$ \{a \in A_\pi : \text{tl}_{t0}^{\mathfrak{A}}[a] = \mathbf{v} \text{ and } \text{tl}_{t1}^{\mathfrak{A}}[a] = \mathbf{w}\} $ , whenever $ t  < q$

Table 6.2: The assignment  $\theta : V \rightarrow \mathbb{N}$ , assuming that  $\mathfrak{A}$  is a finite model of  $\varphi$ .

We construct  $\mathcal{E}$  in three stages. First, let  $\mathcal{E}_1$  be the following set of constraints involving the variables  $V$ , where  $\pi$ ,  $\mathbf{u}$ ,  $\mathbf{v}$ ,  $\mathbf{w}$  again vary over their standard ranges, and  $s, t$  vary over bit-strings such that  $|s| < p$  and  $|t| < q$ :

$$z_{\pi, \epsilon, \mathbf{u}} = y_{\pi, \epsilon, \mathbf{C} - \mathbf{u}} \quad (6.7)$$

$$y_{\pi, s, \mathbf{u}} = \sum \{\hat{y}_{\pi, s, \mathbf{v}', \mathbf{w}'} \mid \mathbf{v}' + \mathbf{w}' = \mathbf{u}\} \quad (6.8)$$

$$z_{\pi, t, \mathbf{u}} = \sum \{\hat{z}_{\pi, t, \mathbf{v}', \mathbf{w}'} \mid \mathbf{v}' + \mathbf{w}' = \mathbf{u}\} \quad (6.9)$$

$$y_{\pi, s0, \mathbf{v}} = \sum \{\hat{y}_{\pi, s, \mathbf{v}, \mathbf{w}'} \mid \mathbf{v} + \mathbf{w}' \leq \mathbf{C}\} \quad (6.10)$$

$$y_{\pi, s1, \mathbf{w}} = \sum \{\hat{y}_{\pi, s, \mathbf{v}', \mathbf{w}} \mid \mathbf{v}' + \mathbf{w} \leq \mathbf{C}\} \quad (6.11)$$

$$z_{\pi, t0, \mathbf{v}} = \sum \{\hat{z}_{\pi, t, \mathbf{v}, \mathbf{w}'} \mid \mathbf{v} + \mathbf{w}' \leq \mathbf{C}\} \quad (6.12)$$

$$z_{\pi, t1, \mathbf{w}} = \sum \{\hat{z}_{\pi, t, \mathbf{v}', \mathbf{w}} \mid \mathbf{v}' + \mathbf{w} \leq \mathbf{C}\} \quad (6.13)$$

$$1 \leq \sum \{y_{\pi', \epsilon, \mathbf{u}'} \mid \pi' \text{ a 1-type, } \mathbf{u}' \leq \mathbf{C}\}. \quad (6.14)$$

**Lemma 6.7.** *Suppose  $\mathfrak{A}$  is a finite model of  $\varphi$ , and let the variables in  $V$  take the values specified, relative to  $\mathfrak{A}$ , in Table 6.2. Then the constraints in  $\mathcal{E}_1$  are all satisfied.*

*Proof.* The constraints (6.7)–(6.9) follow easily from the respective equations in Lemma 6.5. The constraints (6.10)–(6.13) are immediate. In the (single) constraint (6.14), the sum on the right-hand side evaluates to the cardinality of  $A$ , since every  $a \in A$  has a unique 1-type and a unique  $\epsilon$ -spectrum. But  $A$  is nonempty by definition.  $\square$

To define the next set of constraints, we require some additional terminology. Let  $\tau$  be any 2-type. Since  $\tau$  is a finite set of formulas with free variables  $x$  and  $y$ , we may write  $\bigwedge \tau$  to denote their conjunction. Referring again to the formula (6.1), we say that  $\tau$  is *forbidden*, if the formula

$$\alpha(x) \wedge \alpha(y) \wedge \bigwedge_{1 \leq h \leq l} \left( (e_h(x, y) \rightarrow \beta_h(x, y)) \wedge (e_h(y, x) \rightarrow \beta_h(y, x)) \right) \wedge \bigwedge \tau \quad (6.15)$$

is unsatisfiable. Thus, if  $\mathfrak{A} \models \varphi$  and  $a, b$  are distinct elements of  $A$ , then  $\text{tp}^{\mathfrak{A}}[a, b]$  cannot be forbidden. Since (6.15) is purely Boolean, we can evidently identify the forbidden 2-types in time bounded by an exponential function of  $\|\varphi\|$ .

Now let  $\mathcal{E}_2$  consist of the following constraints, where  $\lambda, \pi$  vary over their standard ranges,  $s, t$  vary over bit-strings such that  $|s| = p, |t| = q$ , and  $\mathbf{u}$  varies over vectors such that  $\mathbf{0} < \mathbf{u} \leq \mathbf{C}$ :

$$y_{\pi, s, \mathbf{u}} = \sum \{x_{\lambda'} \mid \lambda' \in \Lambda_{\pi, s} \text{ and } \mathbf{C}_{\lambda'} = \mathbf{u}\} \quad (6.16)$$

$$x_{(\lambda^{-1})} = x_{\lambda} \quad (6.17)$$

$$x_{\lambda} = 0 \quad \text{whenever } \text{tp}_1(\lambda) = \text{tp}_2(\lambda) \quad (6.18)$$

$$x_{\lambda} = 0 \quad \text{whenever } \lambda \text{ is forbidden} \quad (6.19)$$

$$z_{\pi, t, \mathbf{u}} = 0 \quad \text{whenever } \mu_{\pi, t} \text{ is forbidden.} \quad (6.20)$$

$$z_{\pi, t, \mathbf{u}} = 0 \quad \text{whenever } \mathbf{u} \text{ is not a scalar multiple of } \mathbf{C}_{\mu}, \text{ for } \mu = \mu_{\pi, t} \quad (6.21)$$

**Lemma 6.8.** *Suppose  $\mathfrak{A}$  is a finite, chromatic model of  $\varphi$ , and let the variables in  $V$  take the values specified, relative to  $\mathfrak{A}$ , in Table 6.2. Then the constraints in  $\mathcal{E}_2$  are all satisfied.*

*Proof.* To see that the constraints (6.16) hold, fix  $\pi, s, \mathbf{u}$  (with  $|s| = p$  and  $\mathbf{u} > \mathbf{0}$ ), and write

$$A_{\pi, s, \mathbf{u}} = \{a \in A \mid \text{tp}^{\mathfrak{A}}[a] = \pi \text{ and } \text{sp}_s^{\mathfrak{A}}[a] = \mathbf{u}\}.$$

In addition, for any invertible message-type  $\lambda$ , write

$$A_{\lambda} = \{a \in A \mid \text{there exists } b \in A \setminus \{a\} \text{ such that } \text{tp}^{\mathfrak{A}}[a, b] = \lambda\}.$$

From Table 6.2, we have  $|A_{\pi, s, \mathbf{u}}| = \theta(y_{\pi, s, \mathbf{u}})$  and  $|A_{\lambda}| = \theta(x_{\lambda})$ . Moreover, by Lemma 6.4, the sets  $A_{\lambda'}$ , for  $\lambda'$  ranging over the elements of  $\Lambda_{\pi, s}$ , are pairwise disjoint. But Lemma 6.6 just states that

$$A_{\pi, s, \mathbf{u}} = \bigcup \{A_{\lambda'} \mid \lambda' \in \Lambda_{\pi, s} \text{ and } \mathbf{C}_{\lambda'} = \mathbf{u}\},$$

whence the relevant instance of the constraints (6.16) follows. To see that the constraints (6.17) hold, observe that, since  $\mathfrak{A}$  is chromatic,  $\theta(x_{\lambda})$  is actually the total number of messages of (invertible) type  $\lambda$  sent by elements of  $\mathfrak{A}$ , and similarly for  $\lambda^{-1}$ . But then  $\theta(x_{\lambda})$  and  $\theta(x_{(\lambda^{-1})})$  are obviously equal. The constraints (6.18) are immediate given that  $\mathfrak{A}$  is chromatic. The constraints (6.19) and (6.20) are immediate given that  $\mathfrak{A} \models \varphi$ . Lastly, Equation (6.6) states that, for  $|t| = q$ , no element with 1-type  $\pi$  can have a  $t$ -tally which is not a scalar multiple of  $\mathbf{C}_{\mu}$ , where  $\mu$  is the sole element in  $M_{\pi, t}$ . The constraints (6.21) then follow.  $\square$

Let  $\mathcal{E}_3$  consist of the following constraints, where  $\pi$  varies over all 1-types,  $t$  varies over bit-strings such that  $|t| = q$ , and  $\mathbf{u}$  varies over vectors such that  $\mathbf{0} < \mathbf{u} \leq \mathbf{C}$ :

$$z_{\pi, t, \mathbf{u}} > 0 \quad \Rightarrow \quad \sum \{y_{\pi', \epsilon, \mathbf{u}'} \mid \pi' = \text{tp}_2(\mu_{\pi, t}) \text{ and } \mathbf{u}' \leq \mathbf{C}\} > 0. \quad (6.22)$$

**Lemma 6.9.** *Suppose  $\mathfrak{A}$  is a finite model of  $\varphi$ , and let the variables in  $V$  take the values specified, relative to  $\mathfrak{A}$ , in Table 6.2. Then the constraints in  $\mathcal{E}_3$  are all satisfied.*

*Proof.* Fix  $\pi$ ,  $t$ ,  $\mathbf{u}$  (with  $|t| = q$  and  $\mathbf{u} > \mathbf{0}$ ), so that  $\mu_{\pi,t}$  is the sole element of  $M_{\pi,t}$ . If  $\theta(z_{\pi,t,\mathbf{u}}) > 0$ , some element has  $t$ -tally  $\mathbf{u}$ ; and since  $\mathbf{u} > \mathbf{0}$ ,  $\mu_{\pi,t}$  must be a non-invertible message-type (i.e. not a silent 2-type), and moreover at least one message of that type must be sent in  $\mathfrak{A}$ . Therefore,  $\mathfrak{A}$  contains at least one element whose 1-type is  $\text{tp}_2(\mu_{\pi,t})$ . But the number of elements in  $A$  whose 1-type is  $\text{tp}_2(\mu_{\pi,t})$  is  $\sum\{\theta(y_{\pi',\epsilon,\mathbf{u}'}) \mid \pi' = \text{tp}_2(\mu_{\pi,t}) \text{ and } \mathbf{u}' \leq \mathbf{C}\}$ .  $\square$

Let  $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3$ . Measuring the size  $\|\mathcal{E}\|$  of  $\mathcal{E}$  in the usual way, it is evident that  $\|\mathcal{E}\|$  is bounded above by an exponential function of  $\|\varphi\|$ . From the preceding lemmas:

**Lemma 6.10.** *Let  $\varphi$  and  $\mathcal{E}$  be as above. If  $\varphi$  is finitely satisfiable, then  $\mathcal{E}$  has a solution over  $\mathbb{N}$ .*

*Proof.* Suppose  $\varphi$  is finitely satisfiable. By Lemma 4.2, let  $\mathfrak{A}$  be a finite, chromatic model of  $\varphi$ . Assign to the variables in  $V$  the values given in Table 6.2. Then apply Lemmas 6.7–6.9.  $\square$

When  $\varphi$  is finitely satisfiable, we may think of the variables  $V$  as corresponding to configurational properties of elements in its finite, chromatic models. If the values of these variables are taken to record how often these configurational properties are realized in some such model, as prescribed in Table 6.2, then the constraints  $\mathcal{E}$  will be satisfied.

## 6.5 Main result

We proceed to establish a converse of Lemma 6.10: given a solution of  $\mathcal{E}$  over  $\mathbb{N}$ , there exists a finite model  $\mathfrak{A}$  of  $\varphi$  such that that solution records how often certain configurational properties are realized in  $\mathfrak{A}$ , as specified in Table 6.2.

To reduce notational clutter, we use the variable names  $x_\lambda$ ,  $y_{\pi,s,\mathbf{u}}$ ,  $z_{\pi,t,\mathbf{u}}$ ,  $\hat{y}_{\pi,s,\mathbf{v},\mathbf{w}}$ ,  $\hat{z}_{\pi,t,\mathbf{v},\mathbf{w}}$  to stand for the corresponding natural numbers in some solution of  $\mathcal{E}$  (and similarly for terms involving these variables). Fix some 1-type  $\pi$ , and let  $A_\pi$  be a set with exactly

$$\sum\{y_{\pi,\epsilon,\mathbf{u}'} \mid \mathbf{u}' \leq \mathbf{C}\}$$

elements (possibly zero). Think of  $A_\pi$  as a set of elements which ‘want’ to have 1-type  $\pi$  in some yet-to-be-built finite model  $\mathfrak{A}$  of  $\varphi$ .

Our first step will be to define, for any  $s$ ,  $t$  in their standard ranges, vector-valued functions  $\mathbf{f}_{\pi,s}$  and  $\mathbf{g}_{\pi,t}$  on  $A_\pi$ . For  $a \in A_\pi$ , think of  $\mathbf{f}_{\pi,s}(a)$  as the  $s$ -spectrum which  $a$  wants to have in  $\mathfrak{A}$  (when  $\mathfrak{A}$  is eventually built); and think of  $\mathbf{g}_{\pi,t}(a)$  as the  $t$ -tally which  $a$  wants to have in  $\mathfrak{A}$ . Formally, the definitions of these functions simply depend on our solution of  $\mathcal{E}$ ; informally, however, it helps

to keep in mind Table 6.2 when understanding the construction. In particular, if we want  $y_{\pi,s,\mathbf{u}}$  to represent the number of elements having 1-type  $\pi$  and  $s$ -spectrum  $\mathbf{u}$  in  $\mathfrak{A}$ , we will need to ensure that exactly this number of elements  $a \in A_\pi$  satisfy  $\mathbf{f}_{\pi,s}(a) = \mathbf{u}$ ; and similarly for  $t$ -tallies. That is, we will need to ensure that, for all  $\mathbf{u} \leq \mathbf{C}$ ,

$$|\mathbf{f}_{\pi,s}^{-1}(\mathbf{u})| = y_{\pi,s,\mathbf{u}} \quad (6.23)$$

$$|\mathbf{g}_{\pi,t}^{-1}(\mathbf{u})| = z_{\pi,t,\mathbf{u}}. \quad (6.24)$$

Furthermore, if  $|s| < p$  and  $|t| < q$ , then, recalling Lemma 6.5, we will also need to ensure that, for all  $a \in A_\pi$ ,

$$\mathbf{f}_{\pi,\epsilon}(a) + \mathbf{g}_{\pi,\epsilon}(a) = \mathbf{C} \quad (6.25)$$

$$\mathbf{f}_{\pi,s_0}(a) + \mathbf{f}_{\pi,s_1}(a) = \mathbf{f}_{\pi,s}(a) \quad (6.26)$$

$$\mathbf{g}_{\pi,t_0}(a) + \mathbf{g}_{\pi,t_1}(a) = \mathbf{g}_{\pi,t}(a). \quad (6.27)$$

The following rather technical lemma simply guarantees that these requirements can be satisfied.

**Lemma 6.11.** *Suppose  $x_\lambda, y_{\pi,s,\mathbf{u}}, z_{\pi,t,\mathbf{u}}, \hat{y}_{\pi,s,\mathbf{v},\mathbf{w}}, \hat{z}_{\pi,t,\mathbf{v},\mathbf{w}}$  (with indices having the appropriate ranges) are natural numbers satisfying the constraints  $\mathcal{E}$  given above. Fix any 1-type  $\pi$ , and let  $A_\pi$  be a set of cardinality  $\sum\{y_{\pi,\epsilon,\mathbf{u}'} \mid \mathbf{u}' \leq \mathbf{C}\}$ . Then there exists a system of functions on  $A_\pi$*

$$\mathbf{f}_{\pi,s} : A_\pi \rightarrow \{\mathbf{u} \mid \mathbf{u} \leq \mathbf{C}\} \quad \mathbf{g}_{\pi,t} : A_\pi \rightarrow \{\mathbf{u} \mid \mathbf{u} \leq \mathbf{C}\},$$

where the indices  $s$  and  $t$  vary over their standard ranges, satisfying the following conditions: (i) Equations (6.23) and (6.24) hold for all vectors  $\mathbf{u} \leq \mathbf{C}$ ; (ii) if  $|s| < p$  and  $|t| < q$ , then Equations (6.25)–(6.27) hold for all  $a \in A_\pi$ .

*Proof.* Decompose the set  $A_\pi$  into pairwise disjoint (possibly empty) sets  $A_{\mathbf{u}}$  such that  $|A_{\mathbf{u}}| = y_{\pi,\epsilon,\mathbf{u}}$ , where the index  $\mathbf{u}$  varies over all vectors  $\leq \mathbf{C}$ . This is possible by the cardinality of  $A_\pi$ . For all  $\mathbf{u} \leq \mathbf{C}$ , and all  $a \in A_{\mathbf{u}}$ , set

$$\mathbf{f}_{\pi,\epsilon}(a) = \mathbf{u} \quad \mathbf{g}_{\pi,\epsilon}(a) = \mathbf{C} - \mathbf{u}.$$

This assignment evidently satisfies (6.23) for  $s = \epsilon$ ; and by the constraints (6.7), it also satisfies (6.24) for  $t = \epsilon$ . Moreover, it is immediate that, for all  $a \in A_\pi$ , Equation (6.25) holds as required.

We now construct the functions  $\mathbf{f}_{\pi,s}$ , where  $0 < |s| \leq p$ , by induction on  $s$ . Assume that, for some  $s$  ( $0 \leq |s| < p$ ),  $\mathbf{f}_{\pi,s}$  has been defined and satisfies (6.23). For every vector  $\mathbf{u} \leq \mathbf{C}$ , decompose  $\mathbf{f}_{\pi,s}^{-1}(\mathbf{u})$  into pairwise disjoint (possibly empty) sets  $A_{\mathbf{v},\mathbf{w}}$  such that  $|A_{\mathbf{v},\mathbf{w}}| = \hat{y}_{\pi,s,\mathbf{v},\mathbf{w}}$ , where the indices  $\mathbf{v}, \mathbf{w}$  vary over all vectors satisfying  $\mathbf{v} + \mathbf{w} = \mathbf{u}$ . This is possible by the constraints (6.8) together with the assumption that  $\mathbf{f}_{\pi,s}$  satisfies (6.23). Having thus decomposed the sets  $\mathbf{f}_{\pi,s}^{-1}(\mathbf{u})$  (for all  $\mathbf{u} \leq \mathbf{C}$ ), we see that, for any  $a \in A_\pi$ , there is precisely one (ordered) pair of vectors  $\mathbf{v}, \mathbf{w}$  such that  $a \in A_{\mathbf{v},\mathbf{w}}$ ; hence we may set

$$\mathbf{f}_{\pi,s_0}(a) = \mathbf{v} \quad \mathbf{f}_{\pi,s_1}(a) = \mathbf{w}.$$

This defines the functions  $\mathbf{f}_{\pi,s_0}$  and  $\mathbf{f}_{\pi,s_1}$ . It is immediate that, for all  $a \in A_\pi$ , Equation (6.26) holds as required.

To see that  $\mathbf{f}_{\pi,s_0}$  and  $\mathbf{f}_{\pi,s_1}$  both satisfy Equation (6.23), note that  $\mathbf{f}_{\pi,s_0}(a) = \mathbf{v}$  if and only if, for some vector  $\mathbf{w}'$  such that  $\mathbf{v} + \mathbf{w}' \leq \mathbf{C}$ ,  $a \in A_{\mathbf{v},\mathbf{w}'}$ . Similarly,  $\mathbf{f}_{\pi,s_1}(a) = \mathbf{w}$  if and only if, for some vector  $\mathbf{v}'$  such that  $\mathbf{v}' + \mathbf{w} \leq \mathbf{C}$ ,  $a \in A_{\mathbf{v}',\mathbf{w}}$ . That is,

$$\begin{aligned}\mathbf{f}_{\pi,s_0}^{-1}(\mathbf{v}) &= \bigcup \{A_{\mathbf{v},\mathbf{w}'} \mid \mathbf{v} + \mathbf{w}' \leq \mathbf{C}\} \\ \mathbf{f}_{\pi,s_1}^{-1}(\mathbf{w}) &= \bigcup \{A_{\mathbf{v}',\mathbf{w}} \mid \mathbf{v}' + \mathbf{w} \leq \mathbf{C}\},\end{aligned}$$

with the collections of sets on the respective right-hand sides being pairwise disjoint. By the constraints (6.10)–(6.11), together with the fact that  $|A_{\mathbf{v},\mathbf{w}}| = \hat{y}_{\pi,s,\mathbf{v},\mathbf{w}}$  for all  $\mathbf{v}, \mathbf{w}$ , we have:

$$\begin{aligned}|\mathbf{f}_{\pi,s_0}^{-1}(\mathbf{v})| &= y_{\pi,s_0,\mathbf{v}} \\ |\mathbf{f}_{\pi,s_1}^{-1}(\mathbf{w})| &= y_{\pi,s_1,\mathbf{w}},\end{aligned}$$

which establishes (6.23) for the functions  $\mathbf{f}_{\pi,s_0}$  and  $\mathbf{f}_{\pi,s_1}$ . This completes the induction. The construction of the functions  $\mathbf{g}_{\pi,t}$  proceeds completely analogously, using the constraints (6.9), (6.12) and (6.13).  $\square$

**Lemma 6.12.** *Let the functions  $\mathbf{f}_{\pi,s}$  and  $\mathbf{g}_{\pi,t}$  be constructed as in Lemma 6.11. Then, for all  $a \in A_\pi$ , we have*

$$\sum \{\mathbf{f}_{\pi,s'}(a) : |s'| = p\} + \sum \{\mathbf{g}_{\pi,t'}(a) : |t'| = q\} = \mathbf{C}.$$

*Proof.* We prove the stronger result that, for all  $a \in A_\pi$ ,  $j$  ( $0 \leq j \leq p$ ) and  $k$  ( $0 \leq k \leq q$ ),

$$\sum \{\mathbf{f}_{\pi,s'}(a) : |s'| = j\} + \sum \{\mathbf{g}_{\pi,t'}(a) : |t'| = k\} = \mathbf{C}, \quad (6.28)$$

using a double induction on  $j$  and  $k$ . If  $j = k = 0$ , then the left-hand side of (6.28) is simply  $\mathbf{f}_{\pi,\epsilon}(a) + \mathbf{g}_{\pi,\epsilon}(a)$ , which is equal to  $\mathbf{C}$  by (6.25). Suppose now that the result holds for the pair  $j, k$ , with  $j < p$ . Then

$$\begin{aligned}& \sum \{\mathbf{f}_{\pi,s'}(a) : |s'| = (j+1)\} + \sum \{\mathbf{g}_{\pi,t'}(a) : |t'| = k\} \\ &= \sum \{\mathbf{f}_{\pi,s'0}(a) + \mathbf{f}_{\pi,s'1}(a) : |s'| = j\} + \sum \{\mathbf{g}_{\pi,t'}(a) : |t'| = k\} \\ &= \sum \{\mathbf{f}_{\pi,s'}(a) : |s'| = j\} + \sum \{\mathbf{g}_{\pi,t'}(a) : |t'| = k\} \quad \text{by (6.26)} \\ &= \mathbf{C} \quad \text{by inductive hypothesis.}\end{aligned}$$

This establishes the result for the pair  $j+1, k$ . An analogous argument using (6.27) applies when  $k < m$ , completing the induction.  $\square$

Before we come to the promised converse of Lemma 6.10, we remark on the (exponentially many) choices made during the construction of the various functions  $\mathbf{f}_{\pi,s}$  and  $\mathbf{g}_{\pi,t}$  in the proof of Lemma 6.11—specifically, in the decomposition of certain sets into collections of subsets. A given solution of  $\mathcal{E}$  ensures that a system of functions  $\mathbf{f}_{\pi,s}$  and  $\mathbf{g}_{\pi,t}$  exists, subject to the given conditions; but it by no means determines them.

**Lemma 6.13.** *Let  $\mathcal{E}$  be as above and  $k$  a positive integer. If  $\mathcal{E}$  has a solution over  $\mathbb{N}$ , then it has a solution over  $\mathbb{N}$  in which all positive values are at least  $k$ .*

*Proof.* Suppose  $\mathcal{E}$  has a solution  $\theta : V \rightarrow \mathbb{N}$ . Now define  $\theta' : V \rightarrow \mathbb{N}$  by  $\theta'(v) = k\theta(v)$ . By inspection,  $\theta'$  is a solution of  $\mathcal{E}$ .  $\square$

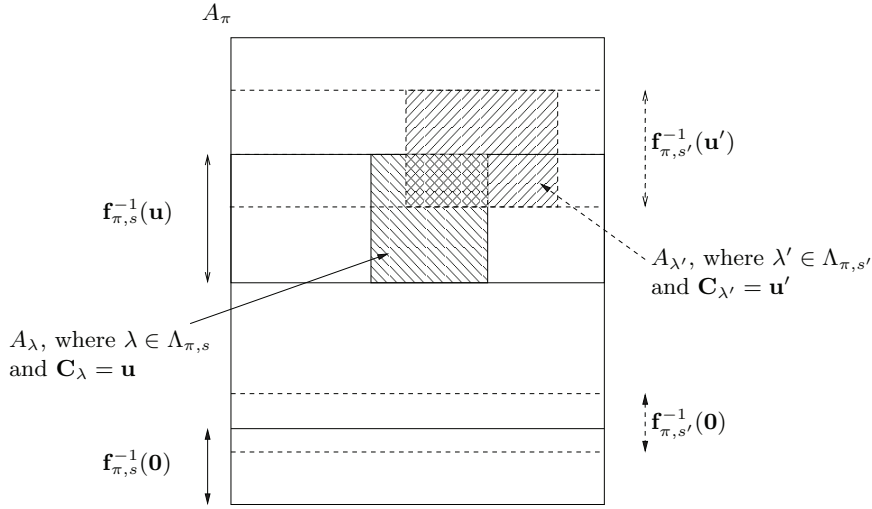
Model-theoretically, Lemma 6.13 is simply a reflection of Lemma 6.2.

**Lemma 6.14.** *Let  $\varphi$  and  $\mathcal{E}$  be as above. If  $\mathcal{E}$  has a solution over  $\mathbb{N}$ , then  $\varphi$  is finitely satisfiable.*

*Proof.* By Lemma 6.13, we may assume that  $\mathcal{E}$  has a solution in which all positive values are greater than or equal to  $3mC$ . Again, we use the variable names  $x_\lambda, y_{\pi,s,\mathbf{u}}, z_{\pi,t,\mathbf{u}}, \hat{y}_{\pi,s,\mathbf{v},\mathbf{w}}, \hat{z}_{\pi,t,\mathbf{v},\mathbf{w}}$  to stand for the corresponding values in this solution. Our task is to construct a model  $\mathfrak{A}$  of  $\varphi$ .

For each 1-type  $\pi$ , let  $A_\pi$  be a set of cardinality  $\sum\{y_{\pi,\epsilon,\mathbf{u}} \mid \mathbf{u} \leq \mathbf{C}\}$ , with the  $A_\pi$  pairwise disjoint; and let  $A = \bigcup\{A_\pi \mid \pi \text{ a 1-type}\}$ . Think of  $A_\pi$  as the set of elements of  $A$  which ‘want’ to have 1-type  $\pi$ . By the constraint (6.14),  $A \neq \emptyset$ . For every 1-type  $\pi$ , let the functions  $\mathbf{f}_{\pi,s}$  and  $\mathbf{g}_{\pi,t}$  on  $A_\pi$  be constructed as in Lemma 6.11; we are interested only in those  $\mathbf{f}_{\pi,s}$  and  $\mathbf{g}_{\pi,t}$  where  $|s| = p$ , and  $|t| = q$ . Think of  $\mathbf{f}_{\pi,s}(a)$  as the  $s$ -spectrum which  $a$  wants to have, and of  $\mathbf{g}_{\pi,t}(a)$  as the  $t$ -tally which  $a$  wants to have. Finally, consider any set  $\mathbf{f}_{\pi,s}^{-1}(\mathbf{u})$ , where  $\mathbf{0} < \mathbf{u} \leq \mathbf{C}$  and  $|s| = p$ . Using the constraints (6.16) and Equation (6.23), we can decompose  $\mathbf{f}_{\pi,s}^{-1}(\mathbf{u})$  into pairwise disjoint (possibly empty) sets  $A_\lambda$  with  $|A_\lambda| = x_\lambda$ , where  $\lambda$  varies over the set of invertible message-types such that  $\lambda \in \Lambda_{\pi,s}$  and  $\mathbf{C}_\lambda = \mathbf{u}$ . It follows that, if  $a \in A_\lambda$ , with  $\lambda \in \Lambda_{\pi,s}$ , then  $\mathbf{C}_\lambda = \mathbf{f}_{\pi,s}(a)$ . Think of  $A_\lambda$  as the set of elements of  $A_\pi$  which want to send a single message of (invertible) type  $\lambda$ .

Before proceeding, we pause to consider the construction just described in respect of any of the sets  $A_\pi$ . Fixing, for the moment, some bit-string  $s$  with  $|s| = p$ , we see that  $A_\pi$  is decomposed into the pairwise disjoint sets  $\mathbf{f}_{\pi,s}^{-1}(\mathbf{u})$  (as  $\mathbf{u}$  varies over vectors  $\leq \mathbf{C}$ ), and that each of the sets  $\mathbf{f}_{\pi,s}^{-1}(\mathbf{u})$ , where  $\mathbf{0} < \mathbf{u} \leq \mathbf{C}$ , is further decomposed into the pairwise disjoint subsets  $A_\lambda$  (as  $\lambda$  varies over the elements of  $\Lambda_{\pi,s}$  such that  $\mathbf{C}_\lambda = \mathbf{u}$ ). Note that the set  $\mathbf{f}_{\pi,s}^{-1}(\mathbf{0})$  is not subject to this further stage of decomposition. This process is performed for *every* bit string  $s$  with  $|s| = p$ , so that different values of  $s$  lead to independent—and possibly overlapping—decompositions, as illustrated in Fig. 6.1. Likewise, for every bit-string  $t$  with  $|t| = q$ ,  $A_\pi$  is decomposed into the pairwise disjoint sets  $\mathbf{g}_{\pi,t}^{-1}(\mathbf{u})$  (as  $\mathbf{u}$  varies over vectors  $\leq \mathbf{C}$ ). Again, decompositions corresponding to different values of  $t$  should be thought of as independent of each other.

Figure 6.1: The decompositions of  $A_\pi$  for the strings  $s$  and  $s'$ .

We now proceed to construct, for every  $a \in A$ , a ‘mosaic piece’; these pieces will be assembled into the desired structure  $\mathfrak{A}$ . Formally, a mosaic piece is a finite multiset of message-types (the reader is asked to excuse the mixed metaphor); informally, we may think of a mosaic piece as a finite collection of ‘messages’ sent by  $a$ , each of which is labelled with some (invertible or non-invertible) message-type (Fig. 6.2). Recall from Section 6.2 that, if  $\pi$  is any 1-type, then  $\mu_{\pi,0}, \dots, \mu_{\pi,R-1}$  is an enumeration of the non-invertible message-types  $\mu$  such that  $\text{tp}_1(\mu) = \pi$ . Fix  $a \in A$ , and let  $\pi$  be the unique 1-type such that  $a \in A_\pi$ . The messages in the mosaic piece corresponding to  $a$  shall be as follows. (i) For every bit-string  $s$  such that  $|s| = p$ , if  $\mathbf{f}_{\pi,s}(a) > \mathbf{0}$ , let  $\lambda_{a,s}$  be the invertible message-type  $\lambda \in \Lambda_{\pi,s}$  such that  $a \in A_\lambda$  (hence  $\mathbf{C}_\lambda = \mathbf{f}_{\pi,s}(a)$ ), and let the mosaic piece corresponding to  $a$  contain a single message labelled  $\lambda_{a,s}$ . Note that, if  $\mathbf{f}_{\pi,s}(a) > \mathbf{0}$ , then  $\lambda_{a,s}$  exists and is unique by the construction of the sets  $A_\lambda$ . (ii) For every bit string  $t$  such that  $|t| = q$ , if  $\mu = \mu_{\pi,t}$  is a non-invertible message-type and  $\mathbf{C}_\mu > \mathbf{0}$ , let  $n_{a,t}$  be the unique natural number  $n$  such that  $\mathbf{g}_{\pi,t}(a) = n\mathbf{C}_\mu$ , and let the mosaic piece corresponding to  $a$  contain  $n_{a,t}$  distinct messages labelled  $\mu_{\pi,t}$ . Note that, if  $\mathbf{g}_{\pi,t}(a) = \mathbf{0}$ , then  $n_{a,t} = 0$ ; on the other hand, if  $\mathbf{g}_{\pi,t}(a) > \mathbf{0}$ , then  $n_{a,t}$  exists by the constraints (6.21) and Equation (6.24). The resulting mosaic piece is depicted in Fig. 6.2, where, for readability, we have replaced any bit-strings by the integers they conventionally denote.

For all  $a \in A$  and all  $i$  ( $1 \leq i \leq m$ ), let  $C_{a,i}$  ( $1 \leq i \leq m$ ) be the number of messages in the mosaic piece for  $a$  (as just constructed) having any label  $\nu$  for which  $f_i(x, y) \in \nu$ , and furthermore let  $\mathbf{C}_a$  be the vector  $(C_{a,1}, \dots, C_{a,m})$ . By

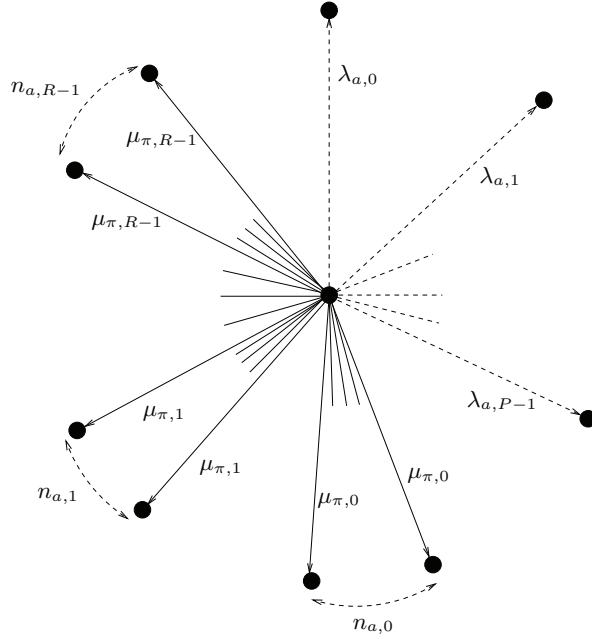


Figure 6.2: The mosaic piece corresponding to  $a \in A_\pi$ . For each  $j$  ( $0 \leq j < P$ ),  $a$  may or may not send a message labelled  $\lambda_{a,j}$  (hence the dotted lines); if it does, then  $\lambda_{a,j} \in \Lambda_{\pi,j}$ . For each  $k$  ( $0 \leq k < R$ ),  $a$  sends  $n_{a,k}$  messages labelled  $\mu_{\pi,k}$ ; but the numbers  $n_{a,k}$  can be zero.

inspection of Fig. 6.2,

$$\mathbf{C}_a = \sum \{\mathbf{f}_{\pi,s'}(a) : |s'| = p\} + \sum \{\mathbf{g}_{\pi,t'}(a) : |t'| = q\},$$

and so, by Lemma 6.12,

$$\mathbf{C}_a = \mathbf{C}. \quad (6.29)$$

If the mosaic piece corresponding to  $a$  contains a message labelled with some message-type  $\nu$ , we will say that  $a$  sends a message of type  $\nu$ . Equation (6.29) is evidently a necessary condition for mosaic pieces that are going to be assembled into a model  $\mathfrak{A}$  of  $\varphi$ . We build  $\mathfrak{A}$  in four steps as follows.

**Step 1** (Fixing the 1-types): For all 1-types  $\pi$  and all  $a \in A_\pi$ , set  $\text{tp}^{\mathfrak{A}}[a] = \pi$ . Since the  $A_\pi$  are pairwise disjoint, no clashes arise.

**Step 2** (Fixing the invertible message-types): Let  $\lambda$  be any invertible message-type. By construction, exactly  $|A_\lambda| = x_\lambda$  elements of  $A$  send some message labelled with  $\lambda$ , and each of those elements sends exactly one such message. Hence, the number of messages labelled with  $\lambda$  (over all  $a \in A$ ) is  $x_\lambda$ ; likewise, the number of messages labelled with  $\lambda^{-1}$  is  $x_{\lambda^{-1}}$ . By the constraints (6.17),

we may put the  $\lambda$ -labelled messages and the  $\lambda^{-1}$ -labelled messages in 1–1 correspondence. If  $a \in A$  sends a  $\lambda$ -labelled message, let  $b \in A$  send the corresponding  $\lambda^{-1}$ -labelled message, and set  $\text{tp}^{\mathfrak{A}}[a, b] = \lambda$ . For this assignment to make sense, we need to check that  $a$  and  $b$  are distinct. But, by construction, we must have  $x_\lambda > 0$ , whence, by the constraints (6.18),  $\text{tp}_1(\lambda) \neq \text{tp}_2(\lambda)$ , so that  $A_{\text{tp}_1(\lambda)}$  and  $A_{\text{tp}_2(\lambda)}$  are disjoint sets containing  $a$  and  $b$ , respectively. Thus, the assignment  $\text{tp}^{\mathfrak{A}}[a, b] = \lambda$  makes sense, and does not clash with the 1-type assignments in Step 1. We can think of the element  $b$  as ‘receiving’ the message sent by  $a$  (and vice versa). Moreover, by construction, for every 1-type  $\pi'$ ,  $a$  sends at most one message labelled with an invertible message-type  $\lambda'$  such that  $\text{tp}_2(\lambda') = \pi'$ . Therefore, there is no chance that these assignments clash with each other. Note, incidentally, that this method of avoiding clashes means that  $\mathfrak{A}$  will turn out to be chromatic.

**Step 3** (Fixing the non-invertible message-types): As a preliminary, for every 1-type  $\pi$ , we decompose  $A_\pi$  into three pairwise disjoint (possibly empty) sets  $A_{\pi,0}$ ,  $A_{\pi,1}$  and  $A_{\pi,2}$  satisfying the condition that, if  $|A_\pi| \geq 3mC$ , then  $|A_{\pi,j}| \geq mC$  for all  $j$  ( $0 \leq j \leq 2$ ). Now let  $\mu$  be any non-invertible message-type, let  $\pi = \text{tp}_1(\mu)$ , and let  $\rho = \text{tp}_2(\mu)$ . (Note that  $\pi$  and  $\rho$  may be identical.) Let  $t$  be the bit-string of length  $q$  such that  $\mu = \mu_{\pi,t}$ , and suppose some element  $a$  sends  $n_{a,t} > 0$  messages labelled  $\mu$ . It follows that  $a \in A_\pi$ , and also that there is a vector  $\mathbf{u} > \mathbf{0}$  such that  $\mathbf{g}_{\pi,t}(a) = \mathbf{u}$ , and hence such that  $\mathbf{g}_{\pi,t}^{-1}(\mathbf{u})$  is non-empty. By Equation (6.24),  $z_{\pi,t,\mathbf{u}}$  is positive, whence, by the constraints (6.22),  $\sum\{y_{\rho,\epsilon,\mathbf{u}'} \mid \mathbf{u}' \leq \mathbf{C}\}$  is also positive, and therefore, by our choice of solution, greater than or equal to  $3mC$ . But recall that, since  $\rho$  is a 1-type,  $|A_\rho| = \sum\{y_{\rho,\epsilon,\mathbf{u}'} \mid \mathbf{u}' \leq \mathbf{C}\}$ , so that each of the sets  $A_{\rho,0}$ ,  $A_{\rho,1}$  and  $A_{\rho,2}$  contains at least  $mC$  elements. Since  $a \in A_\pi$ , let  $j$  ( $0 \leq j \leq 2$ ) be such that  $a \in A_{\pi,j}$ , let  $k = j + 1 \pmod{3}$ , and select  $n_{a,t}$  elements  $b$  from  $A_{\rho,k}$  which have not yet been chosen to receive any other messages (invertible or non-invertible) sent by  $a$ . Since the total number of messages sent by  $a$  is certainly at most  $mC$ , we never run out of choices. For each of these elements  $b$ , set  $\text{tp}^{\mathfrak{A}}[a, b] = \mu$ . Since  $\pi = \text{tp}_1(\mu)$  and  $\rho = \text{tp}_2(\mu)$ , these assignments cannot clash with those made in Step 1, and by construction, they cannot clash with assignments corresponding to other messages sent by  $a$ . We need only check that they cannot clash with assignments corresponding to messages sent by  $b$ . Specifically, we must ensure that, if  $\text{tp}^{\mathfrak{A}}[a, b] = \mu$  is assigned as just described, it is not possible for  $a$  to be chosen to receive a  $\mu'$ -labelled message sent by  $b$ , where  $\mu'$  is some non-invertible message-type. But any  $\mu'$ -labelled message sent by  $b \in A_{\rho,k}$ , with  $\text{tp}_2(\mu') = \pi$ , could only be sent to an element in  $A_{\pi,j'}$ , where  $j' = k + 1 \pmod{3}$ ; and by assumption,  $A_{\pi,j}$  and  $A_{\pi,j'}$  are disjoint, (Fig. 6.3). Observe that this conclusion follows even if  $\pi = \rho$ .

**Step 4** (Fixing the remaining 2-types): Recall that a guard-atom is any atom  $p(x, y)$  or  $p(y, x)$ , where  $p$  is a binary predicate. If  $\text{tp}^{\mathfrak{A}}[a, b]$  has not been defined, set it to be the 2-type

$$\pi \cup \rho[y/x] \cup \{\neg\gamma \mid \gamma \text{ is a guard-atom not involving } \approx\},$$

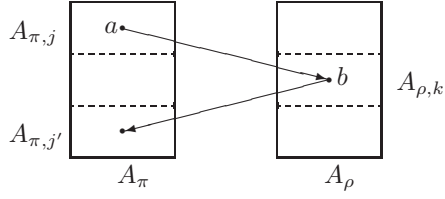


Figure 6.3: Fixing the non-invertible message-types.

where  $\pi = \text{tp}^{\mathfrak{A}}[a]$ ,  $\rho = \text{tp}^{\mathfrak{A}}[b]$ , and  $\rho[y/x]$  is the result of replacing  $x$  by  $y$  in  $\rho$ . Note that neither this 2-type nor its inverse is a message-type. Note also that, since the integers  $C_1, \dots, C_m$  and  $m$  are by assumption all positive,  $a$  and  $b$  certainly send some messages, so that the constraints (6.19) and (6.20) ensure that both  $\alpha \wedge \bigwedge \pi$  and  $\alpha \wedge \bigwedge \rho$  are satisfiable.

This completes the definition of  $\mathfrak{A}$ ; it remains to show that  $\mathfrak{A} \models \varphi$ . Referring to (6.1), we consider first the conjuncts:

$$\forall x \alpha \wedge \bigwedge_{1 \leq h \leq l} \forall x \forall y (e_h(x, y) \rightarrow (\beta_h \vee x \approx y)).$$

We see from the constraints (6.19) and (6.20) that no 2-type assignment in Steps 2 and 3 violates these conjuncts. And it is obvious that no assignment in Step 4 does so. (This is where we use the guardedness of  $\varphi$ , of course.) Finally, we consider the conjuncts

$$\bigwedge_{1 \leq i \leq m} \forall x \exists_{=C_i} y (f_i(x, y) \wedge x \not\approx y).$$

To see that these conjuncts are all satisfied, it suffices to note Equation (6.29) and the fact that none of the 2-types assigned in Step 4 is a message-type.  $\square$

The constraints  $\mathcal{E}$  all have the forms

$$\begin{aligned} x_1 + \dots + x_n &= x \\ x_1 + \dots + x_n &\geq 1 \\ x &= 0 \\ x > 0 &\Rightarrow x_1 + \dots + x_n > 0, \end{aligned} \tag{6.30}$$

where  $n > 0$ ,  $x, x_1, \dots, x_n$  are variables. We now investigate the problem of determining whether  $\mathcal{E}$  has a solution. The following lemma essentially repeats Lutz, Sattler and Tendera [18], Proposition 11. (Those authors in turn credit Calvanese [3].) We give a proof for convenience. An *integer programming problem* is a system of linear equalities and inequalities interpreted over  $\mathbb{Z}$ . A *linear programming problem* is a system of linear equalities and inequalities interpreted over  $\mathbb{Q}$ .

**Lemma 6.15.** *Let  $\varphi$  and  $\mathcal{E}$  be as above. An algorithm exists to determine whether  $\mathcal{E}$  has a solution over  $\mathbb{N}$  in time bounded by a polynomial function of  $\|\mathcal{E}\|$ , and hence by an exponential function of  $\|\varphi\|$ .*

*Proof.* Evidently,  $\mathcal{E}$  can be regarded as a very large disjunction of integer programming problems, each one of which has size bounded by  $\|\mathcal{E}\|$ . By a well-known theorem (see Papadimitriou [23]), there is a monotonic function  $h$ , computable in polynomial time, such that, if an integer programming problem of size  $n$  has a solution, then it has a solution in which every value is bounded by  $h(n) > 0$ . Hence,  $\mathcal{E}$  has a solution over  $\mathbb{N}$  if and only if it has a solution over  $\mathbb{N}$  in which every value is bounded by  $H = h(\|\mathcal{E}\|)$ .

Now consider the integer programming problem  $\mathcal{E}_H$  defined by replacing every constraint of the form  $x > 0 \Rightarrow x_1 + \dots + x_n > 0$  in  $\mathcal{E}$  by the corresponding inequalities

$$\begin{aligned} Hy &\geq x \\ x_1 + \dots + x_n &\geq y, \end{aligned}$$

where  $y$  is a new variable. Every solution of  $\mathcal{E}_H$  over  $\mathbb{N}$  is clearly a solution of  $\mathcal{E}$ . Conversely, suppose  $\theta : V \rightarrow \mathbb{N}$  is a solution of  $\mathcal{E}$  in which all values are bounded by  $H$ . Let  $y$  be any of the new variables of  $\mathcal{E}_H$ , introduced to eliminate the constraint  $x > 0 \Rightarrow x_1 + \dots + x_n > 0$ ; and extend  $\theta$  to give a value to  $y$  as follows:

$$\theta(y) = \begin{cases} 0 & \text{if } \theta(x) = 0 \\ 1 & \text{otherwise.} \end{cases}$$

It is routine to check that extending  $\theta$  in this way for all the new variables  $y$  in  $\mathcal{E}_H$  yields a solution of  $\mathcal{E}_H$ . Hence  $\mathcal{E}$  can be transformed, in time bounded by a polynomial function of  $\|\mathcal{E}\|$ , into a constraint set  $\mathcal{E}_H$ , in which all constraints are of the forms

$$\begin{aligned} x_1 + \dots + x_n &= x & x &= 0 \\ x_1 + \dots + x_n &\geq 1 & Hx_1 &\geq x_2 \\ x_1 + \dots + x_n &\geq x, \end{aligned}$$

such that  $\mathcal{E}$  has a solution (over  $\mathbb{N}$ ) if and only if  $\mathcal{E}_H$  has. It is obvious that, if  $\mathcal{E}_H$  has a solution over the non-negative rationals, then it has a solution over  $\mathbb{N}$  as well. (Simply multiply by the product of all the denominators.) Hence, we can equivalently regard  $\mathcal{E}_H$  as a linear programming problem. But linear programming is in PTIME, by Khachiyan's theorem [15].  $\square$

**Theorem 6.3.** *The finite satisfiability problem for  $\mathcal{GC}^2$  is in EXPTIME.*

*Proof.* Lemmas 6.3, 6.10, 6.14 and 6.15.  $\square$

## 6.6 The Satisfiability Problem

The above technique also provides a simple proof of a result derived in Kazakov [14], namely, that the satisfiability problem for  $\mathcal{GC}^2$  is in EXPTIME.

**Notation 2.** Let  $\mathbb{N}^*$  denote the set  $\mathbb{N} \cup \{\aleph_0\}$ . We extend the ordering  $>$  and the arithmetic operations  $+$  and  $\cdot$  from  $\mathbb{N}$  to  $\mathbb{N}^*$  in the obvious way. Specifically, we define  $\aleph_0 > n$  for all  $n \in \mathbb{N}$ ; we define  $\aleph_0 + \aleph_0 = \aleph_0 \cdot \aleph_0 = \aleph_0$  and  $0 \cdot \aleph_0 = \aleph_0 \cdot 0 = 0$ ; we define  $n + \aleph_0 = \aleph_0 + n = \aleph_0$  for all  $n \in \mathbb{N}$ ; and we define  $n \cdot \aleph_0 = \aleph_0 \cdot n = \aleph_0$  for all  $n \in \mathbb{N}$  such that  $n > 0$ . Under this extension,  $>$  remains a total order, and  $+$ ,  $\cdot$  remain associative and commutative.

Using the arithmetic in Notation 2, consider again the constraints  $\mathcal{E}$  given in (6.7)–(6.14), (6.16)–(6.21) and (6.22), but now with the variables ranging over the whole of  $\mathbb{N}^*$ .

**Lemma 6.16.** *Let  $\varphi$  and  $\mathcal{E}$  be as above. Then  $\varphi$  is satisfiable if and only if  $\mathcal{E}$  has a solution over  $\mathbb{N}^*$ .*

*Proof.* If  $\varphi$  is satisfiable, then it has a model  $\mathfrak{A}$  which is finite or countably infinite. Now assign to the variables in  $V$  values in  $\mathbb{N}^*$  as directed by Table 6.2. The reasoning of Lemmas 6.7–6.10 then goes through, with the obvious changes of formulation, exactly as in the finite case. For the converse, proceed exactly as for Lemmas 6.11–6.14.  $\square$

**Lemma 6.17.** *The set of constraints  $\mathcal{E}$  has a solution over  $\mathbb{N}^*$  if and only if it has a solution over  $\{0, \aleph_0\}$ .*

*Proof.* Suppose  $\mathcal{E}$  has a solution  $\theta : V \rightarrow \mathbb{N}^*$ . Now define  $\theta' : V \rightarrow \{0, \aleph_0\}$  by  $\theta'(v) = \aleph_0 \theta(v)$ . By inspection,  $\theta'$  is a solution of  $\mathcal{E}$ .  $\square$

Model-theoretically, Lemma 6.17 is simply a reflection of Lemma 6.2.

Since the domain  $\{0, \aleph_0\}$  has only 2-elements, variables interpreted over it are essentially Boolean. If  $x \in V$ , let us write  $X$  for the corresponding statement  $x = 0$ , so that the constraints  $\mathcal{E}$  are viewed as formulas of propositional logic. For example, a constraint of the form

$$x_1 + \cdots + x_n = x$$

becomes the set of propositional logic formulas

$$\{X_1 \wedge \cdots \wedge X_n \rightarrow X\} \cup \{X \rightarrow X_i \mid 1 \leq i \leq n\};$$

a constraint of the form

$$x_1 + \cdots + x_n \geq 1$$

becomes the propositional logic formula

$$X_1 \wedge \cdots \wedge X_n \rightarrow \perp;$$

and a constraint of the form

$$x > 0 \Rightarrow x_1 + \cdots + x_n > 0$$

becomes the propositional logic formula

$$X_1 \wedge \cdots \wedge X_n \rightarrow X.$$

A quick check reveals that all of the resulting formulas are Horn-clauses. This immediately yields:

**Theorem 6.4** (Kazakov). *The satisfiability problem for  $\mathcal{GC}^2$  is in EXPTIME.*

The proof in Kazakov [14] proceeds by showing that satisfiability in  $\mathcal{GC}^2$  can be reduced in polynomial time to satisfiability in the 3-variable guarded fragment; Theorem 6.4 then follows by the complexity bound for the latter established in Grädel [9]. The approach taken in the present paper is thus somewhat more direct. Moreover, Kazakov's reduction is not conservative, and, as mentioned, yields no complexity bound for the corresponding finite satisfiability problem.