Electronic Notes in Theoretical Computer Science 84 (2003)
URL: http://www.elsevier.nl/locate/entcs/volume84.html 12 pages

System NEL is Undecidable

Lutz Straßburger [1,2]

Technische Universität Dresden
Fakultät Informatik
01062 Dresden, Germany

Abstract

System NEL is a conservative extension of multiplicative exponential linear logic (extended by the rules mix and nullary mix) by a self-dual noncommutative connective called *seq* which has an intermediate position between the connectives *par* and *times*. In this paper, I will show that system NEL is undecidable by encoding two counter machines into NEL. Although the encoding is simple, the proof of the faithfulness is a little intricate because there is no sequent calculus and no phase semantics available for NEL.

1 Introduction

Since the beginning of linear logic [5], the complexity of the provability problem of its fragments has been studied. The multiplicative fragment is NP-complete [10], the multiplicative additive fragment is PSPACE-complete and full propositional linear logic is undecidable [16]. The decidability of the multiplicative exponential fragment (MELL) is still an open problem. But in a purely noncommutative setting, i.e. in the presence of two mutually dual noncommutative connectives, the multiplicatives and the exponentials are sufficient to get undecidability [16].

In this paper, I will address the decidability question for a mixed commutative and noncommutative system in which there is only one single self-dual noncommutative connective. I will show that also in this case, the multiplicatives and the exponentials alone are sufficient to get undecidability, as it has been conjectured in [8]. For showing this, Guglielmi proposes in [6] an encoding of Post's correspondence problem, which makes the noncommutativity correspond to sequential composition of words. Since I was not able to find a complete proof along these lines, I will here use an encoding of two counter machines, which also has the advantage of being simpler. If it turns out that

[1] Email: lutz.strassburger@inf.tu-dresden.de

[2] Supported by DFG Graduiertenkolleg 334

Associativity

$$\begin{aligned}[\boldsymbol{R},[\boldsymbol{T}],\boldsymbol{U}] &= [\boldsymbol{R},\boldsymbol{T},\boldsymbol{U}]\\ (\boldsymbol{R},(\boldsymbol{T}),\boldsymbol{U}) &= (\boldsymbol{R},\boldsymbol{T},\boldsymbol{U})\\ \langle\boldsymbol{R};\langle\boldsymbol{T}\rangle;\boldsymbol{U}\rangle &= \langle\boldsymbol{R};\boldsymbol{T};\boldsymbol{U}\rangle\end{aligned}$$

Commutativity

$$\begin{aligned}[\boldsymbol{R},\boldsymbol{T}] &= [\boldsymbol{T},\boldsymbol{R}]\\ (\boldsymbol{R},\boldsymbol{T}) &= (\boldsymbol{T},\boldsymbol{R})\end{aligned}$$

Unit

$$\begin{aligned}[\circ,\boldsymbol{R}] &= [\boldsymbol{R}]\\ (\circ,\boldsymbol{R}) &= (\boldsymbol{R})\\ \langle\circ;\boldsymbol{R}\rangle &= \langle\boldsymbol{R}\rangle\\ \langle\boldsymbol{R};\circ\rangle &= \langle\boldsymbol{R}\rangle\end{aligned}$$

Singleton

$$[R] = (R) = \langle R\rangle = R$$

Exponentials

$$\begin{aligned}?\circ &= \circ\\ !\circ &= \circ\\ ??R &= ?R\\ !!R &= !R\end{aligned}$$

Negation

$$\begin{aligned}\bar{\circ} &= \circ\\ \overline{[R_1,\dots,R_h]} &= (\bar{R}_1,\dots,\bar{R}_h)\\ \overline{(R_1,\dots,R_h)} &= [\bar{R}_1,\dots,\bar{R}_h]\\ \overline{\langle R_1;\dots;R_h\rangle} &= \langle\bar{R}_1;\dots;\bar{R}_h\rangle\\ \overline{?R} &= !\bar{R}\\ \overline{!R} &= ?\bar{R}\\ \bar{\bar{R}} &= R\end{aligned}$$

Contextual Closure

if $R = T$ then $S\{R\} = S\{T\}$

Fig. 1. Basic equations for the syntactic equivalence =

MELL is decidable (as many believe), then the border to undecidability is crossed by this new self-dual noncommutative connective. Such a connective did first occur in Retoré's pomset logic [19] and has then been rediscovered in Guglielmi's system BV [7]. I conjecture that the two logics are the same, but the proof of this is not yet complete. The new noncommutative connective is important for applications in linguistics as well as in concurrency: Because of the self-duality it corresponds quite well to the notion of sequentiality in many process algebras. For example, in [3] Bruscoli shows the correspondence to the prefixing operation of CCS [17].

In the following, I will first (in Section 2) introduce system NEL [9], which is a conservative extension of MELL plus *mix* [4] plus *nullary mix* [1] by a self-dual noncommutative connective called *seq* [7]. It has been shown by Tiu in [21] that a logic containing that connective cannot be presented in the sequent calculus because deep rewriting is crucial for reasoning with *seq*. For that reason, I will use here the *calculus of structures* [7,8,2], which is a generalisation of the one-sided sequent calculus. Rules do not work on sequents but on structures, which are intermediate expressions between formulae and sequents.

Then, in Section 3, I will introduce two counter machines [18,15] and show in Section 4 how they are encoded in system NEL. The encoding is pretty much inspired by [12], and the proof of its completeness is an easy exercise (done in Section 5).

However, the proof of the faithfulness of the encoding is quite different from what has been done so far. There are two reasons for this: First, the simple way of extracting the computation sequence of the machine from the proof of the encoding, as done in [16,11] for full linear logic, is not possible because the calculus of structures allows more freedom in applying and permuting rules than the sequent calculus does. And second, the use of phase spaces [5], as it has been done in [13,14,12] is not possible because (so far) there is no phase semantics available for NEL.

The method I will use instead is the following. The given proof in system NEL of an encoding of a two-counter machine is first transformed into a certain normal form, which allows to remove the exponentials. The resulting proof in the multiplicative fragment has as conclusion a structure which has the shape of what I call a *weak encoding*. From this proof, I will extract the first computation step of the machine and another proof (in the multiplicative fragment) which has as conclusion again a weak encoding. By an inductive argument it is then possible to obtain the whole computation sequence. For this, I will first discuss the multiplicative fragment (namely Guglielmi's system BV [7,8]) in Section 6, and then show the full proof in Section 7.

2 System NEL

In order to present a system in the calculus of structures, we first need to define a language of structures, in the same way as we need to define a language of formulae when presenting a system in the sequent calculus or natural deduction.

Definition 2.1 There are countably many *positive* and *negative atoms*. They, positive or negative, are denoted by a, b, c, d, p and q. *Structures* are denoted by S, P, Q, R, T, U, V, W, X and Z. The structures of the *language* NEL are generated by

$$S ::= a \mid \circ \mid [\,\underbrace{S,\dots,S}_{>0}\,] \mid (\,\underbrace{S,\dots,S}_{>0}\,) \mid \langle\underbrace{S;\dots;S}_{>0}\rangle \mid ?S \mid !S \mid \bar{S} \quad ,$$

where $\circ$, the *unit*, is not an atom; $[S_1,\dots,S_h]$ is a *par structure*, $(S_1,\dots,S_h)$ is a *times structure*, $\langle S_1;\dots;S_h\rangle$ is a *seq structure*, $?S$ is a *why-not structure* and $!S$ is an *of-course structure*; $\bar{S}$ is the *negation* of the structure S. Structures with a hole that does not appear in the scope of a negation are denoted by $S\{\ \}$. The structure R is a *substructure* of $S\{R\}$, and $S\{\ \}$ is its *context*. I will simplify the indication of context in cases where structural parentheses fill the hole exactly: for example, $S[R,T]$ stands for $S\{[R,T]\}$.

Structures come with equational theories establishing some basic, decidable algebraic laws by which structures are indistinguishable. These are analogous to the laws of associativity, commutativity, idempotency, and so on, usually imposed on sequents. The difference is that the notions of formula and sequent are merged and the equations are extended to formulae. The structures of the

$$\circ{\downarrow}\,\frac{}{\circ}\qquad \mathsf{ai}{\downarrow}\,\frac{S\{\circ\}}{S[a,\bar{a}]}\qquad \mathsf{s}\,\frac{S([R,U],T)}{S[(R,T),U]}\qquad \mathsf{q}{\downarrow}\,\frac{S\langle[R,U];[T,V]\rangle}{S[\langle R;T\rangle,\langle U;V\rangle]}$$

$$\mathsf{p}{\downarrow}\,\frac{S\{![R,T]\}}{S[!R,?T]}\qquad \mathsf{w}{\downarrow}\,\frac{S\{\circ\}}{S\{?R\}}\qquad \mathsf{b}{\downarrow}\,\frac{S[?R,R]}{S\{?R\}}$$

Fig. 2. System NEL

language NEL are equivalent modulo the relation =, defined in Fig. 1. There, $\boldsymbol{R}$, $\boldsymbol{T}$ and $\boldsymbol{U}$ stand for finite, nonempty sequences of structures (sequences may contain ',' or ';' separators as appropriate in the context).

There is a straightforward two-way correspondence between structures not involving seq and formulae of MELL. For example $![(?a,b),\bar{c},!\bar{d}]$ corresponds to $!((?a\otimes b)\mathbin{⅋} c^{\perp}\mathbin{⅋}!d^{\perp})$, and vice versa. Units are mapped into $\circ$, since $1\equiv\perp$, when the rules mix [4] and mix0 [1] are added to MELL.

The next step in defining a system is to show the inference rules. In the calculus of structures, an (*inference*) *rule* is a scheme $\rho\,\dfrac{T}{R}$, where ρ is the *name* of the rule, T is its *premise* and R is its *conclusion*. If a rule ρ has no premise, then it is called an *axiom*. Observe that contrary to the sequent calculus, all rules have at most one premise.

A (*formal*) *system*, denoted by $\mathcal{S}$, is a set of rules. A *derivation* in a system $\mathcal{S}$ is a finite sequence of instances of rules of $\mathcal{S}$, and is denoted by Δ; a derivation can consist of just one structure. The topmost structure in a derivation is called its *premise*; the bottommost structure is called *conclusion*. A derivation Δ whose premise is T, conclusion is R, and whose rules are in $\mathcal{S}$ is denoted by $\begin{array}{c}T\\ \Delta\Vert\mathcal{S}\\ R\end{array}$. Similarly, $\begin{array}{c}\Pi\Vert\mathcal{S}\\ R\end{array}$ denotes a *proof* Π, which is a derivation with no premise.

The rules of *system* NEL are shown in Fig. 2. Except for the axiom, all are of the kind $\rho\,\dfrac{S\{T\}}{S\{R\}}$. This rule scheme specifies that if a structure matches R, in a context $S\{\ \}$, it can be rewritten as specified by T, in the same context $S\{\ \}$ (or vice versa if one reasons top-down). Fig. 3 shows an example for a proof in system NEL.

For system NEL, the cut rule has the following shape: $\mathsf{i}{\uparrow}\,\dfrac{S(R,\bar{R})}{S\{\circ\}}$.

Theorem 2.2 (Cut Elimination) *The rule* i↑ *is admissible for system* NEL*, in other words, for every proof* $\begin{array}{c}\Pi\Vert\mathsf{NEL}\cup\{\mathsf{i}{\uparrow}\}\\ R\end{array}$*, there is a proof* $\begin{array}{c}\Pi'\Vert\mathsf{NEL}\\ R\end{array}$. □

For a proof of that result and a more detailed discussion on the proof

$$\mathsf{w}\!\downarrow\frac{\mathsf{q}\!\downarrow\frac{\mathsf{q}\!\downarrow\frac{\mathsf{ai}\!\downarrow\frac{\mathsf{ai}\!\downarrow\frac{\mathsf{ai}\!\downarrow\frac{\circ\!\downarrow\frac{}{\circ}}{[a,\bar{a}]}}{\langle[a,\bar{a}];[b,\bar{b}]\rangle}}{\langle[a,\bar{a}];[b,\bar{b}];[c,\bar{c}]\rangle}}{\langle[a,\bar{a}];[\langle b;c\rangle,\langle\bar{b};\bar{c}\rangle]\rangle}}{[\langle a;b;c\rangle,\langle\bar{a};\bar{b};\bar{c}\rangle]}}{[?(\langle c;d\rangle,\bar{c}),\langle a;b;c\rangle,\langle\bar{a};\bar{b};\bar{c}\rangle]}$$

Fig. 3. A proof in system NEL

theory of NEL, the reader is referred to [9]. Observe that NEL is a conservative extension of $\mathsf{MELL}+\mathsf{mix}+\mathsf{mix0}$. We have that $[R,T]\multimap\langle R;T\rangle\multimap(R,T)$. For the precise relation between NEL and linear logic, the reader should consult [20] and [7].

3 Two Counter Machines

Two counter machines have been introduced by Minsky in [18] as two tape non-writing Turing machines. He showed that any (usual) Turing machine can be simulated on a two counter machine. In [15], Lambek showed that any recursive function can be computed by an n counter machine, for some number $n \in \mathbf{N}$.

Definition 3.1 A *two counter machine* $\mathcal{M}$ is a tuple $(\mathcal{Q}, q_0, n_0, m_0, q_f, \mathcal{T})$, where $\mathcal{Q}$ is a finite set of *states*, $q_0 \in \mathcal{Q}$ is called the *initial state*, $q_f \in \mathcal{Q}$ is called the *final state*, the numbers $n_0, m_0 \in \mathbf{N}$ are the initial values of the two counters, and $\mathcal{T} \subseteq \mathcal{Q} \times \mathcal{I} \times \mathcal{Q}$ is a finite set of *transitions*, where $\mathcal{I} = \{\mathsf{inc1}, \mathsf{dec1}, \mathsf{zero1}, \mathsf{inc2}, \mathsf{dec2}, \mathsf{zero2}\}$ is the set of possible *instructions*. A *configuration* of $\mathcal{M}$ is given by a tuple (q, n, m), where $q \in \mathcal{Q}$ is a state and n and m are natural numbers. The configuration (q_0, n_0, m_0) is called *initial configuration*. A configuration (q', n', m') is *reachable in one step* from a configuration (q, n, m), written as $(q, n, m) \rightarrow (q', n', m')$, if one of the following six cases holds:

- $(q, \mathsf{inc1}, q') \in \mathcal{T}$ and $n' = n + 1$ and $m' = m$,
- $(q, \mathsf{dec1}, q') \in \mathcal{T}$ and $n > 0$ and $n' = n - 1$ and $m' = m$,
- $(q, \mathsf{zero1}, q') \in \mathcal{T}$ and $n' = n = 0$ and $m' = m$,
- $(q, \mathsf{inc2}, q') \in \mathcal{T}$ and $n' = n$ and $m' = m + 1$,
- $(q, \mathsf{dec2}, q') \in \mathcal{T}$ and $n' = n$ and $m > 0$ and $m' = m - 1$,
- $(q, \mathsf{zero2}, q') \in \mathcal{T}$ and $n' = n$ and $m' = m = 0$.

A configuration (q', n', m') is *reachable in r steps* from a configuration (q, n, m), written as $(q, n, m) \to^r (q', n', m')$, if

- $r = 0$ and $(q', n', m') = (q, n, m)$ or
- $r \geqslant 1$ and there is a configuration (q'', n'', m'') such that $(q, n, m) \to (q'', n'', m'')$ and $(q'', n'', m'') \to^{r-1} (q', n', m')$.

A configuration (q', n', m') is *reachable* from a configuration (q, n, m), written as $(q, n, m) \to^* (q', n', m')$, if there is an $r \in \mathbf{N}$ such that $(q, n, m) \to^r (q', n', m')$. In other words, the relation $\to^*$ is the reflexive and transitive closure of $\to$. A two counter machine $\mathcal{M} = (\mathcal{Q}, q_0, n_0, m_0, q_f, \mathcal{T})$ *accepts* a configuration (q, n, m), if $(q, n, m) \to^* (q_f, 0, 0)$.

Example 3.2 The running example in this paper will be the following

$$\begin{aligned} &\mathcal{M} = (\{q_0, q_1, q_2\}, q_0, 1, 0, q_1, \mathcal{T}) \quad , \quad \text{where} \\ &\mathcal{T} = \{(q_0, \mathsf{dec2}, q_2), (q_1, \mathsf{dec1}, q_1), (q_0, \mathsf{zero2}, q_1)\} \quad . \end{aligned}$$

The machine accepts for example the configuration $(q_0, 8, 0)$, because $(q_0, 8, 0) \to (q_1, 8, 0) \to^8 (q_1, 0, 0)$. More precisely, it accepts any configuration $(q_0, n, 0)$ for $n \geqslant 0$. In particular it also accepts its initial configuration $(q_0, 1, 0)$.

Theorem 3.3 *In general, it is undecidable whether a two counter machine accepts its initial configuration* [18,15]. □

4 Encoding Two Counter Machines in NEL

Let a be an atom and $n \in \mathbf{N}$. Then a^n denotes the structure $\langle a; a; \ldots; a\rangle$ with n copies of a. More precisely, $a^0 = \circ$ and $a^n = \langle a^{n-1}; a\rangle$, for $n \geqslant 1$.

Encoding 4.1 Let a two counter machine $\mathcal{M} = (\mathcal{Q}, q_0, n_0, m_0, q_f, \mathcal{T})$ be given. For each state $q \in \mathcal{Q}$, I will introduce a fresh atom, also denoted by q. Further, I will need four fresh atoms a, b, c and d. Without loss of generality, let $\mathcal{Q} = \{q_0, q_1, \ldots, q_z\}$ for some $z \geqslant 0$. Then $q_f = q_i$ for some $i \in \{0, \ldots, z\}$. A configuration (q, n, m) will be encoded by the structure $\langle b; a^n; q; c^m; d\rangle$. Since $\mathcal{T}$ is finite, we have $\mathcal{T} = \{t_1, t_2, \ldots, t_h\}$ for some $h \in \mathbf{N}$ (if $\mathcal{T} = \varnothing$, then $h = 0$). For each $k \in \{1, \ldots, h\}$, I will define the structure T_k, that encodes the transition t_k, as follows. For all $i, j \in \{0, \ldots, z\}$,

- if $t_k = (q_i, \mathsf{inc1}, q_j)$, then $T_k = (\bar{q}_i, \langle a; q_j\rangle)$,
- if $t_k = (q_i, \mathsf{dec1}, q_j)$, then $T_k = (\langle \bar{a}; \bar{q}_i\rangle, q_j)$,
- if $t_k = (q_i, \mathsf{zero1}, q_j)$, then $T_k = (\langle \bar{b}; \bar{q}_i\rangle, \langle b; q_j\rangle)$,
- if $t_k = (q_i, \mathsf{inc2}, q_j)$, then $T_k = (\bar{q}_i, \langle q_j; c\rangle)$,
- if $t_k = (q_i, \mathsf{dec2}, q_j)$, then $T_k = (\langle \bar{q}_i; \bar{c}\rangle, q_j)$,
- if $t_k = (q_i, \mathsf{zero2}, q_j)$, then $T_k = (\langle \bar{q}_i; \bar{d}\rangle, \langle q_j; d\rangle)$.

I will say that a structure T *encodes* a transition of $\mathcal{M}$, if $T = T_k$ for some

$k \in \{1,\dots,h\}$. The machine $\mathcal{M}$ is then encoded by the structure

$$\mathcal{M}_{\text{enc}} = [?T_1,\dots,?T_h,\langle b;a^{n_0};q_0;c^{m_0};d\rangle,\langle\bar{b};\bar{q}_f;\bar{d}\rangle] \quad ,$$

which is called the *encoding* of $\mathcal{M}$.

Example 4.2 The machine in Example 3.2 is encoded by the structure

$$\mathcal{M}_{\text{enc}} = [?(\langle\bar{q}_0;\bar{c}\rangle,q_2),?(\langle\bar{a};\bar{q}_1\rangle,q_1),?(\langle\bar{q}_0;\bar{d}\rangle,\langle q_1;d\rangle),\langle b;a;q_0;d\rangle,\langle\bar{b};\bar{q}_1;\bar{d}\rangle] \quad .$$

Theorem 4.3 *A two counter machine $\mathcal{M}$ accepts its initial configuration if and only if its encoding $\mathcal{M}_{\text{enc}}$ is provable in* NEL.

The remaining sections are devoted to the proof of this theorem. The main result of this paper is an immediate consequence:

Corollary 4.4 *Provability in system* NEL *is undecidable.*

5 Completeness of the Encoding

Lemma 5.1 *Given a two counter machine $\mathcal{M} = (\mathcal{Q},q_0,n_0,m_0,q_f,\mathcal{T})$.*

$$\textit{If}\quad (q_i,n,m)\to(q_j,n',m')\quad\textit{then}\qquad \begin{array}{c}[?T_1,\dots,?T_h,\langle b;a^{n'};q_j;c^{m'};d\rangle,\langle\bar{b};\bar{q}_f;\bar{d}\rangle]\\ \Vert{\scriptstyle\mathsf{NEL}}\\ [?T_1,\dots,?T_h,\langle b;a^{n};q_i;c^{m};d\rangle,\langle\bar{b};\bar{q}_f;\bar{d}\rangle]\end{array} \quad .$$

Proof. There are six possible cases how the machine $\mathcal{M}$ can go from (q_i,n,m) to (q_j,n',m'). I will show only the case where the first counter is decremented (the others are similar). We have $(q_i,\mathtt{dec1},q_j)\in\mathcal{T}$ and $n>0$ and $n'=n-1$ and $m'=m$. Therefore $T_k = (\langle\bar{a};\bar{q}_i\rangle,q_j)$ for some $k\in\{1,\dots,h\}$. Now use

$$\mathsf{b}\!\downarrow\dfrac{\mathsf{q}\!\downarrow\dfrac{\mathsf{q}\!\downarrow\dfrac{\mathsf{s}\dfrac{\mathsf{q}\!\downarrow\dfrac{\mathsf{ai}\!\downarrow\dfrac{\mathsf{ai}\!\downarrow\dfrac{[?T_1,\dots,?T_h,\langle b;a^{n-1};q_j;c^m;d\rangle,\langle\bar{b};\bar{q}_f;\bar{d}\rangle]}{[?T_1,\dots,?T_h,\langle b;a^{n-1};([\bar{q}_i,q_i],q_j);c^m;d\rangle,\langle\bar{b};\bar{q}_f;\bar{d}\rangle]}}{[?T_1,\dots,?T_h,\langle b;a^{n-1};(\langle[\bar{a},a];[\bar{q}_i,q_i]\rangle,q_j);c^m;d\rangle,\langle\bar{b};\bar{q}_f;\bar{d}\rangle]}}{[?T_1,\dots,?T_h,\langle b;a^{n-1};([\langle\bar{a};\bar{q}_i\rangle,\langle a;q_i\rangle],q_j);c^m;d\rangle,\langle\bar{b};\bar{q}_f;\bar{d}\rangle]}}{[?T_1,\dots,?T_h,\langle b;a^{n-1};[(\langle\bar{a};\bar{q}_i\rangle,q_j),\langle a;q_i\rangle];c^m;d\rangle,\langle\bar{b};\bar{q}_f;\bar{d}\rangle]}}{[?T_1,\dots,?T_h,\langle[(\langle\bar{a};\bar{q}_i\rangle,q_j),\langle b;a^n;q_i\rangle];c^m;d\rangle,\langle\bar{b};\bar{q}_f;\bar{d}\rangle]}}{[?T_1,\dots,?T_h,(\langle\bar{a};\bar{q}_i\rangle,q_j),\langle b;a^n;q_i;c^m;d\rangle,\langle\bar{b};\bar{q}_f;\bar{d}\rangle]}}{[?T_1,\dots,?T_h,\langle b;a^n;q_i;c^m;d\rangle,\langle\bar{b};\bar{q}_f;\bar{d}\rangle]} \quad .$$

□

Now we can prove the first direction of Theorem 4.3.

Proposition 5.2 *Given a two counter machine $\mathcal{M} = (\mathcal{Q},q_0,n_0,m_0,q_f,\mathcal{T})$.*

$$\textit{If}\quad (q_0,n_0,m_0)\to^*(q_f,0,0)\quad\textit{then}\qquad \begin{array}{c}\overline{}\!\!\Vert{\scriptstyle\mathsf{NEL}}\\ \mathcal{M}_{\text{enc}}\end{array} \quad .$$

$$\circ\!\downarrow\frac{}{\circ}\qquad \mathsf{ai}\!\downarrow\frac{S\{\circ\}}{S[a,\bar{a}]}\qquad \mathsf{s}\,\frac{S([R,U],T)}{S[(R,T),U]}\qquad \mathsf{q}\!\downarrow\frac{S\langle[R,U];[T,V]\rangle}{S[\langle R;T\rangle,\langle U;V\rangle]}$$

Fig. 4. System BV

Proof. Use

$$\begin{array}{c} \Pi\,\Vert\,\mathsf{NEL} \\ [?T_1,\ldots,?T_h,\langle b;q_f;d\rangle,\langle\bar{b};\bar{q}_f;\bar{d}\rangle] \\ \Delta\,\Vert\,\mathsf{NEL} \\ [?T_1,\ldots,?T_h,\langle b;a^{n_0};q_0;c^{m_0};d\rangle,\langle\bar{b};\bar{q}_f;\bar{d}\rangle] \end{array},$$

where Δ is obtained from Lemma 5.1 by an easy inductive argument and Π exists trivially (cf. Fig. 3). □

6 Some Facts about System BV

In order to show the other direction, I need first to establish some properties of the multiplicative fragment of system NEL. That fragment is called system BV [8,7] and is shown in Fig. 4.

If a structure R is provable in BV, then every atom a occurs as often in R as $\bar{a}$. This is easy to see: the only possibility, where an atom a can disappear is an instance of $\mathsf{ai}\!\downarrow$. But then at the same time an atom $\bar{a}$ does disappear.

A BV structure R is called a *non-par structure* if it it does not contain a par structure as substructure. Let R be a BV structure and let a be an atom occurring in R. I will say that the atom a is *unique* in R if it occurs exactly once. For example, in $[\langle c;c;d\rangle,\langle\bar{c};\bar{c};\bar{d}\rangle]$, the atoms d and $\bar{d}$ are unique, but not c and $\bar{c}$. A set $\mathcal{P}$ of atoms is called *clean* if for all atoms $a\in\mathcal{P}$, we have $\bar{a}\notin\mathcal{P}$. If $e:\mathcal{P}\to\mathcal{Q}$ be a mapping, where $\mathcal{P}$ and $\mathcal{Q}$ are two clean sets of atoms, and R is a BV structure, then R^e is the structure obtained from R where each atom $p\in\mathcal{P}$ is replaced by $q=e(p)\in\mathcal{Q}$. If for example $\mathcal{P}=\{a,b\}$ and $\mathcal{Q}=\{c\}$ and $e(a)=e(b)=c$, then we have $[\langle a;b;c;d\rangle,\langle\bar{a};\bar{d};\bar{b};\bar{a}\rangle]^e=[\langle c;c;c;d\rangle,\langle\bar{c};\bar{d};\bar{c};\bar{c}\rangle]$.

The following two lemmata will play a crucial role in the proof of the faithfulness of the encoding of two counter machines. Lemma 6.1 is almost a triviality and Lemma 6.2 can be established by using Guglielmi's *Splitting Lemma* [7].

Lemma 6.1 *Given two clean sets $\mathcal{P}$ and $\mathcal{Q}$ of atoms, a mapping $e:\mathcal{P}\to\mathcal{Q}$ and a structure R. If R is provable in BV, then R^e is also provable in BV.* □

Lemma 6.2 *Let $R=[Z,(\bar{V},T),\langle U;V;W\rangle]$ be a BV structure, such that $\bar{V}$ is a non-par structure and all atoms occurring in V are unique in R. If R is provable in BV, then $R'=[Z,\langle U;T;W\rangle]$ is also provable in BV.* □

Definition 6.3 A BV structure Q is called a *negation circuit* if there is a clean set of atoms $\mathcal{P}=\{a_1,a_2,\ldots,a_n\}$, such that $Q=[Z_1,\ldots,Z_n]$, where

- $Z_j=(a_j,\bar{a}_{j+1})$ or $Z_j=\langle a_j;\bar{a}_{j+1}\rangle$ for every $j=1,\ldots,n-1$, and

- $Z_n = (a_n, \bar{a}_1)$ or $Z_n = \langle a_n; \bar{a}_1 \rangle$.

I will say that a structure P *contains a negation circuit* if there is a structure Q, which is a negation circuit that can be obtained from P by replacing some atoms in P by $\circ$, and all atoms that occur in Q are unique in P.

Example 6.4 The structure $P = [(a, c, [\bar{d}, b]), \bar{c}, \langle \bar{b}; [\bar{a}, d] \rangle]$ contains the negation circuit $Q = [(a, b), \langle \bar{b}; \bar{a} \rangle] = [(a, \circ, [\circ, b]), \circ, \langle \bar{b}; [\bar{a}, \circ] \rangle]$.

Proposition 6.5 *Let P be a* BV *structure. If P contains a negation circuit, then P is not provable in* BV.

Proof. By induction on the number of atoms in the negation circuit. □

Remark 6.6 I believe that the converse of Proposition 6.5 does also hold. This would then immediately imply the equivalence between Guglielmi's BV and Retoré's pomset logic [19].

7 Faithfulness of the Encoding

The main ingredient of the proof of the second direction of Theorem 4.3 is the notion of weak encoding together with a crucial use of Proposition 6.5.

Definition 7.1 Let $\mathcal{M} = (\mathcal{Q}, q_0, n_0, m_0, q_f, \mathcal{T})$ be a two counter machine. Let $W = [U_1, \ldots, U_r, \langle b; a^n; q; c^m; d \rangle, \langle \bar{b}; \bar{q}_f; \bar{d} \rangle]$ be a BV structure for some $r, n, m \geqslant 0$ and $q \in \mathcal{Q}$. Then W is called a *weak encoding* of $\mathcal{M}$, if the structures $U_1, \ldots, U_r$ encode transitions of $\mathcal{M}$, i.e. for every $l \in \{1, \ldots, r\}$, we have that $U_l = T_k$ for some $k \in \{1, \ldots, h\}$.

Observe that in a weak encoding W of a machine $\mathcal{M}$, some transitions T_k might occur many times and some might not occur at all.

Lemma 7.2 *Given a two counter machine $\mathcal{M} = (\mathcal{Q}, q_0, n_0, m_0, q_f, \mathcal{T})$.*

If $\begin{array}{c} \|\!|\,\mathsf{NEL} \\ \mathcal{M}_{\mathrm{enc}} \end{array}$ *then there is a weak encoding W of $\mathcal{M}$, such that* $\begin{array}{c} \|\!|\,\mathsf{BV} \\ W \\ \|\,\{\mathsf{w}\!\downarrow, \mathsf{b}\!\downarrow\} \\ \mathcal{M}_{\mathrm{enc}} \end{array}$.

Proof. The rules $\mathsf{w}\!\downarrow$ and $\mathsf{b}\!\downarrow$ can be permuted under any other rule in NEL. Since $\mathcal{M}_{\mathrm{enc}}$ does not contain any !, there are no (nontrivial) instances of $\mathsf{p}\!\downarrow$. □

The following lemma is nothing but an act of bureaucracy. The idea is to rename the atoms $q_0, \ldots, q_z$ that encode the states of the machine in such a way that each new atom occurs only once. This will then simplify the extraction of the computation sequence from the proof.

Lemma 7.3 *Let $\mathcal{M} = (\mathcal{Q}, q_0, n_0, m_0, q_f, \mathcal{T})$ be a two counter machine and let $W = [U_1, \ldots, U_r, \langle b; a^n; q; c^m; d \rangle, \langle \bar{b}; \bar{q}_f; \bar{d} \rangle]$ be a weak encoding of $\mathcal{M}$. Further, let $\mathcal{P} = \{p_0, \ldots, p_r\}$ be a clean set of $r + 1$ fresh atoms. If W is provable in*

BV*, then there is a structure* $\tilde{W} = [\tilde{U}_1, \dots, \tilde{U}_r, \langle b; a^n; p_0; c^m; d\rangle, \langle \bar{b}; \bar{p}_r; \bar{d}\rangle]$ *and a mapping* $e : \mathcal{P} \to \mathcal{Q}$*, such that*

- $\tilde{W}$ *is provable in* BV*,*
- *all atoms* $p_0, \bar{p}_0, \dots, p_r, \bar{p}_r$ *occur exactly once in* $\tilde{W}$*,*
- *for every* $l \in \{1, \dots, r\}$*, the atoms* $\bar{p}_{l-1}$ *and* p_l *occur inside* $\tilde{U}_l$*,*
- $\tilde{W}^e = W$*, and*
- *for every* $l \in \{1, \dots, r\}$*, we have* $\tilde{U}_l^e = U_{l'}$ *for some* $l' \in \{1, \dots, r\}$*.* □

Example 7.4 We get $W = [(\langle \bar{a}; \bar{q}_1\rangle, q_1), (\langle \bar{q}_0; \bar{d}\rangle, \langle q_1; d\rangle), \langle b; a; q_0; d\rangle, \langle \bar{b}; \bar{q}_1; \bar{d}\rangle]$ as weak encoding for the encoding in Example 4.2. From this we can obtain $\tilde{W} = [\, (\langle \bar{p}_0; \bar{d}\rangle, \langle p_1; d\rangle)\, , (\langle \bar{a}; \bar{p}_1\rangle, p_2)\, , \langle b; a; p_0; d\rangle\, , \langle \bar{b}; \bar{p}_2; \bar{d}\rangle\,]$, with $e(p_0) = q_0$ and $e(p_1) = e(p_2) = q_1$.

The following lemma is at the heart of this paper.

Lemma 7.5 *Let* $\mathcal{M} = (\mathcal{Q}, q_0, n_0, m_0, q_f, \mathcal{T})$ *a two counter machine and let* $W = [U_1, \dots, U_r, \langle b; a^n; q; c^m; d\rangle, \langle \bar{b}; \bar{q}_f; \bar{d}\rangle]$ *be a weak encoding of* $\mathcal{M}$*. If* $\begin{array}{c}\Vert\,\mathsf{BV}\\ W\end{array}$*, then* $(q, n, m) \to^r (q_f, 0, 0)$*.*

Proof. By induction on r. If $r = 0$ then $W = [\langle b; a^n; q; c^m; d\rangle, \langle \bar{b}; \bar{q}_f; \bar{d}\rangle]$. This is only provable if $n = m = 0$ and $q = q_f$, i.e. if $W = [\langle b; q_f; d\rangle, \langle \bar{b}; \bar{q}_f; \bar{d}\rangle]$. We trivially have that $(q_f, 0, 0) \to^0 (q_f, 0, 0)$.

Let us now consider the case where $r > 0$. By Lemma 7.3, there is a set $\mathcal{P} = \{p_0, \dots, p_r\}$ of $r + 1$ fresh atoms, a mapping $e : \mathcal{P} \to \mathcal{Q}$ and a provable structure $\tilde{W} = [\tilde{U}_1, \dots, \tilde{U}_r, \langle b; a^n; p_0; c^m; d\rangle, \langle \bar{b}; \bar{p}_r; \bar{d}\rangle]$, with $\tilde{W}^e = W$, and such that the killer $\bar{p}_0$ of p_0 is inside $\tilde{U}_1$. Now we have six cases, how $\tilde{U}_1$ can look like. Let me here show only the case where $\tilde{U}_1 = (\langle \bar{a}; \bar{p}_0\rangle, p_1)$. It contains all the ideas and the other cases are very similar. In this case we have

$$\tilde{W} = [\ (\langle \bar{a}; \bar{p}_0\rangle, p_1), \tilde{U}_2, \dots, \tilde{U}_r, \langle b; a^n; p_0; c^m; d\rangle, \langle \bar{b}; \bar{p}_r; \bar{d}\rangle\ ] \quad .$$

Mark inside $\tilde{W}$ the atom $\bar{a}$ inside $\tilde{U}_1$ by $\bar{a}^\bullet$ and its killer by $a^\bullet$. By way of contradiction, assume now that $a^\bullet$ occurs inside $\tilde{U}_l = (\bar{p}_{l-1}, \langle a^\bullet; p_l\rangle)$ for some $l \in \{2, \dots, r\}$. This means that

$$\begin{aligned}\tilde{W} = [\ &(\langle \bar{a}^\bullet; \bar{p}_0\rangle, p_1), \tilde{U}_2, \dots, \tilde{U}_{l-1}, (\bar{p}_{l-1}, \langle a^\bullet; p_l\rangle), \tilde{U}_{l+1}, \dots, \tilde{U}_r,\\ &\langle b; a^n; p_0; c^m; d\rangle, \langle \bar{b}; \bar{p}_r; \bar{d}\rangle\ ] \quad .\end{aligned}$$

But then $\tilde{W}$ contains the negation circuit $[(\bar{a}^\bullet, p_1), (\bar{p}_1, p_2), \dots, (\bar{p}_{l-1}, a^\bullet)]$. This is by Proposition 6.5 a contradiction to the provability of $\tilde{W}$. Hence, the atom $a^\bullet$ must occur inside the encoding of the configuration, which means that $n > 0$. Furthermore, we have that

$$\tilde{W} = [\ (\langle \bar{a}^\bullet; \bar{p}_0\rangle, p_1), \tilde{U}_2, \dots, \tilde{U}_r, \langle b; a^{n'}; a^\bullet; a^{n''}; p_0; c^m; d\rangle, \langle \bar{b}; \bar{p}_r; \bar{d}\rangle\ ] \quad ,$$

for some n', n'' with $n = n' + 1 + n''$. I will now show that $n'' = 0$. For this, assume by way of contradiction, that $n'' > 0$. Mark the first atom a in $a^{n''}$ by a° and its killer by $\bar{a}^\circ$. Then $\bar{a}^\circ$ must occur inside $\tilde{U}_k = (\langle \bar{a}^\circ; \bar{p}_{k-1}\rangle, p_k)$ for some $k \in \{2, \dots, r\}$. Then we have that

$$\begin{aligned}\tilde{W} = [\ & (\langle \bar{a}^{\bullet}; \bar{p}_0 \rangle, p_1), \tilde{U}_2, \ldots, \tilde{U}_{k-1}, (\langle \bar{a}^{\circ}; \bar{p}_{k-1} \rangle, p_k), \tilde{U}_{k+1}, \ldots, \tilde{U}_r, \\ & \langle b; a^{n'}; a^{\bullet}; a^{\circ}; a^{n''-1}; p_0; c^m; d \rangle, \langle \bar{b}; \bar{p}_r; \bar{d} \rangle\ ] \quad , \quad \text{which}\end{aligned}$$

contains the negation circuit $[\langle a^{\bullet}; a^{\circ} \rangle, \langle \bar{a}^{\circ}; \bar{p}_{k-1} \rangle, (p_{k-1}, \bar{p}_{k-2}), \ldots, (p_2, \bar{p}_1), (p_1, \bar{a}^{\bullet})]$, which is (by Proposition 6.5) a contradiction to the provability of $\tilde{W}$. Hence, the atom a° cannot exist, which means that $n'' = 0$ and $n' = n - 1$. This means that

$$\tilde{W} = [\ \tilde{U}_2, \ldots, \tilde{U}_r, (\langle \bar{a}^{\bullet}; \bar{p}_0 \rangle, p_1), \langle b; a^{n-1}; a^{\bullet}; p_0; c^m; d \rangle, \langle \bar{b}; \bar{p}_r; \bar{d} \rangle\ ] \quad .$$

Since this is provable in BV, we have (by Lemma 6.2) that

$$\tilde{W}' = [\ \tilde{U}_2, \ldots, \tilde{U}_r, \langle b; a^{n-1}; p_1; c^m; d \rangle, \langle \bar{b}; \bar{p}_r; \bar{d} \rangle\ ]$$

is also provable. Let now $W' = \tilde{W}'^{e}$ and $e(p_1) = q'$. Then

$$W' = [\ U_1, \ldots, U_{l-1}, U_{l+1}, \ldots, U_r, \langle b; a^{n-1}; q'; c^m; d \rangle, \langle \bar{b}; \bar{q}_f; \bar{d} \rangle\ ] \quad ,$$

for some $l \in \{1, \ldots, r\}$. As before, W' is a weak encoding of $\mathcal{M}$ and (by Lemma 6.1) provable in BV. Hence, we can apply the induction hypothesis and get $(q', n-1, m) \to^{r-1} (q_f, 0, 0)$. Furthermore, we have that $U_l = \tilde{U}_1^{e} = (\langle \bar{a}; \bar{q} \rangle, q')$. Therefore $(q, \mathtt{dec1}, q') \in \mathcal{T}$. Since we also have $n > 0$, we have $(q, n, m) \to (q', n-1, m)$, which gives us $(q, n, m) \to^{r} (q_f, 0, 0)$. □

Proposition 7.6 *Given a two counter machine $\mathcal{M} = (\mathcal{Q}, q_0, n_0, m_0, q_f, \mathcal{T})$.*

$$\text{If} \quad \begin{array}{c} \top\!\!\Vert\,\mathsf{NEL} \\ \mathcal{M}_{\mathrm{enc}} \end{array} \quad \text{then} \quad (q_0, n_0, m_0) \to^{*} (q_f, 0, 0) \quad .$$

Proof. Immediate from Lemma 7.2 and Lemma 7.5. □

References

[1] Samson Abramsky and Radha Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59(2):543–574, 1994.

[2] Kai Brünnler and Alwen Fernanto Tiu. A local system for classical logic. In R. Nieuwenhuis and A. Voronkov, editors, *LPAR 2001*, volume 2250 of *Lecture Notes in Artificial Intelligence*, pages 347–361. Springer-Verlag, 2001.

[3] Paola Bruscoli. A purely logical account of sequentiality in proof search. In Peter J. Stuckey, editor, *Logic Programming, 18th International Conference*, volume 2401 of *Lecture Notes in Artificial Intelligence*, pages 302–316. Springer-Verlag, 2002.

[4] Arnaud Fleury and Christian Retoré. The mix rule. *Mathematical Structures in Computer Science*, 4(2):273–285, 1994.

[5] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[6] Alessio Guglielmi. Butterflies, 2002. On the web at: http://www.ki.inf.tu-dresden.de/˜guglielm/Research/Notes/AG7.pdf.

[7] Alessio Guglielmi. A system of interaction and structure. Technical Report WV-02-10, Technische Universität Dresden, 2002. Submitted. On the web at: http://www.ki.inf.tu-dresden.de/˜guglielm/Research/Gug/Gug.pdf.

[8] Alessio Guglielmi and Lutz Straßburger. Non-commutativity and MELL in the calculus of structures. In Laurent Fribourg, editor, *Computer Science Logic, CSL 2001*, volume 2142 of *LNCS*, pages 54–68. Springer-Verlag, 2001.

[9] Alessio Guglielmi and Lutz Straßburger. A non-commutative extension of MELL. In Matthias Baaz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, LPAR 2002*, volume 2514 of *LNAI*, pages 231–246. Springer-Verlag, 2002.

[10] Max I. Kanovich. The complexity of Horn fragments of linear logic. *Annals of Pure and Applied Logic*, 69(2–3):195–241, 1994.

[11] Max I. Kanovich. The direct simulation of Minsky machines in linear logic. In Jean-Yves Girard, Yves Lafont, and Laurent Regnier, editors, *Advances in Linear Logic*, pages 123–145. Cambridge University Press, 1995.

[12] Max I. Kanovich. Simulating computations in second order non-commutative linear logic. In Jean-Yves Girard, Mitsuhiro Okada, and Andre Scedrov, editors, *Electronic Notes in Theoretical Computer Science*, volume 3. Elsevier Science Publishers, 2000.

[13] Yves Lafont. The undecidability of second order linear logic without exponentials. *The Journal of Symbolic Logic*, 61(2):541–548, 1996.

[14] Yves Lafont and Andre Scedrov. The undecidability of second order multiplicative linear logic. *Information and Computation*, 125:46–51, 1996.

[15] Joachim Lambek. How to program an infinite abacus. *Canad. Math. Bull.*, 4(3):295–302, 1961.

[16] P. Lincoln, J. Mitchell, A. Scedrov, and N. Shankar. Decision problems for propositional linear logic. *Annals of Pure and Applied Logic*, 56(1–3):239–311, 1992.

[17] Robin Milner. *Communication and Concurrency.* International Series in Computer Science. Prentice Hall, 1989.

[18] Marvin L. Minsky. Recursive unsolvability of Post's problem of ”tag” and other topics in theory of Turing machines. *The Annals of Mathematics*, 74(3):437–455, 1961.

[19] Christian Retoré. Pomset logic: A non-commutative extension of classical linear logic. In Ph. de Groote and J. R. Hindley, editors, *Typed Lambda Calculus and Applications, TLCA'97*, volume 1210 of *Lecture Notes in Computer Science*, pages 300–318, 1997.

[20] Lutz Straßburger. MELL in the Calculus of Structures. Technical Report WV-01-03, Technische Universität Dresden, 2001. Accepted for publication in Theoretical Computer Science. On the web at: http://www.ki.inf.tu-dresden.de/˜lutz/els.pdf.

[21] Alwen Fernanto Tiu. Properties of a logical system in the calculus of structures. Technical Report WV-01-06, Technische Universität Dresden, 2001. On the web at: http://www.cse.psu.edu/˜tiu/thesisc.pdf.