Complexity of correctness for pomset logic proof nets

Lê Thành Dũng NGUYỄN
Laboratoire d’informatique de Paris Nord, Villetaneuse, France
nltd@nguyentito.eu

January 23, 2020

Retoré’s pomset logic [Ret97] is an extension of MLL+Mix with a binary connective denoted by ‘<’ whose particularity is to be non-commutative and self-dual. In this note, we apply the graph-theoretic methods of [Ngu18] to pomset logic and obtain a coNP-completeness result for proof net correctness.

The system of proof nets for pomset logic extends the MLL+Mix correctness criterion – “there is no (undirected) cycle using at most one premise edge of each ‘<’” – by considering directed cycles, which can only visit both premises of a ‘<’ if the left one comes before the right one in the cycle.

In [Ret97], Retoré presents pomset proof nets as “RB-graphs”, that is, as digraphs (i.e. directed graphs) equipped with perfect matchings (see also [Ret03] for the MLL+Mix case). The advantage is that the correctness criterion can then be stated as a combinatorial property in the vocabulary of mainstream graph theory: it is the absence of alternating circuits. We recall these notions below.

**Definition 1.** A digraph $G = (V, A)$ consists of a finite set of vertices $V$ and a set of arcs $A \subseteq V^2 \setminus \{(u, u) \mid u \in V\}$. An circuit of length $n$ is a $\mathbb{Z}/n\mathbb{Z}$-indexed sequence $u_0, \ldots, u_{n-1} \in V$ without repetitions such that for all $i \in \mathbb{Z}/n\mathbb{Z}$, $(u_i, u_{i+1}) \in A$.

A perfect matching $M$ of a digraph is a subset of arcs such that:

- any vertex $u \in V$ has exactly one outgoing arc in $M$ and exactly one incoming arc in $M$ (i.e. there is exactly one pair $(v, w) \in V^2$ such that $(u, v) \in M$ and $(w, u) \in M)$;
- for all $u, v \in V$, $(u, v) \in M \iff (v, u) \in M$ – morally, $M$ consists of undirected edges.

An alternating circuit is a circuit $u_0, \ldots, u_{n-1}$ such that for all $i \in \mathbb{Z}/n\mathbb{Z}$, exactly one of $(u_{i-1}, u_i)$ and $(u_i, u_{i+1})$ is in $M$ (so that the other one is in $A \setminus M$). Note that this forces the length $n$ of the circuit to be even.

We claim that there is a converse reduction to Retoré’s RB-graphs.

**Theorem 2.** The existence of an alternating circuit for a perfect matching in a digraph is equivalent by polynomial time reductions to the incorrectness of a pomset proof structure in linear time.

**Proof sketch.** One direction is Retoré’s RB-graphs; the other has been done for alternating cycles in undirected graphs and MLL+Mix proof structures in our previous work [Ngu18]. In short, the idea is to represent edges by $\otimes$-links if they are in the matching, and by $\text{ax}$-links if they are outside the matching. To extend that reduction (called “proofification” in [Ngu18]), we use a gadget with two axiom links and one ‘<’-link to encode directed arcs $(u, v)$ whose reverse arc is not in the digraph. See Figure 1 for an example which should be enough to infer the whole construction. □

The claim in the title then follows using our previous work in pure graph theory:

---

1In the paper [GLMM13], the definition of “circuit” includes this prohibition on vertex repetitions. This seems to be common in the graph theory literature. In the same way we shall use “path” to refer to elementary paths.
Figure 1: A digraph equipped with a perfect matching and its translation into a pomset proof structure. Note that \( e \) and \( g \) are directed arcs, while the other undirected edges in the figure represent pairs of directed arcs of the form \( \{(u, v), (v, u)\} \). Since the original digraph has no alternating circuit, the proof structure is a pomset proof net. Compare with \[Ngu18, Figure 6\].

**Theorem 3** ([Ngu20]). Detecting alternating circuits in digraphs is an NP-complete problem.

**Corollary 4.** Deciding the correctness of a proof structure in pomset logic is coNP-complete.

**Proof.** Immediate from the two previous theorems.

To make the present note more self-contained, we propose a direct proof of Theorem 3, by reduction from CNF-SAT. (The proof in [Ngu20] is very concise but depends on a specialized graph-theoretic result [GLMM13, Theorem 5] – whose proof method inspired the one we use below – which in turn depends on other papers.)

For the remainder of this note, we fix an instance of CNF-SAT, that is, a list of clauses \( \{C_1, \ldots, C_n\} \); each clause is a list of literals \( C_i = \{l_{i,1}, \ldots, l_{i,m(i)}\} \); finally, each literal is either \( x \) or \( \neg x \) for some variable \( x \in \{x_1, \ldots, x_p\} \).

Given this instance, we consider a set of vertices \( V_{occ} \) that contains one vertex for each literal occurrence – \( V_{occ} = \{v_{i,j} \mid (i, j) \in I\} \) – plus two auxiliary vertices \( s \) and \( t \).

First, let us build two directed graphs \( G_{cl} \) and \( G_{var} \) whose vertex sets are both \( V_{occ} \cup \{s, t\} \).

**Lemma 5.** From the given CNF-SAT instance, one can build in polynomial time a directed graph \( G_{cl} = (V_{occ} \cup \{s, t\}, A_{cl}) \) such that:

- \( G_{cl} \) is acyclic (i.e. contains no circuits), \( s \) has no incoming edges and \( t \) has no outgoing edges;
- each path from \( s \) to \( t \) in \( G_{cl} \) visits exactly the intermediate vertices \( \{v_{i,j[i]} \mid 1 \leq i \leq n\} \) for some \( j[1], \ldots, j[n] \) (with \( 1 \leq j[i] \leq m(i) \));
- conversely, every such choice of one literal per clause induces a (unique) path from \( s \) to \( t \).

**Proof.** We take:

\[
A_{cl} = \{(v_{i,j}, v_{i+1,j'}) \mid 1 \leq i \leq n - 1, (i, j) \in I, (i + 1, j') \in I\}
\cup \{(s, v_{1,j} \mid j \in \{1, \ldots, m(1)\}\} \cup \{(v_{n,j'}, t) \mid j' \in \{1, \ldots, m(n)\}\}
\]

It is straightforward to check that the required properties hold. For instance, the absence of circuits in \( G_{cl} \) is a consequence of the following fact: for all \( (v_{i,j}, v'_{i,j'}) \in A_{cl}, j' > i \).

**Lemma 6.** From the given CNF-SAT instance, one can build in polynomial time a directed graph \( G_{var} = (V_{occ} \cup \{s, t\}, A_{var}) \) such that:

- \( G_{var} \) is acyclic, \( t \) has no incoming edges and \( s \) has no outgoing edges (note that the roles of \( t \) and \( s \) are reversed compared to \( G_{cl} \)).
The non-matching edges are obtained from the original edges:

\[ \text{Proof.} \]

... that the alternating circuits for \( M \) there is a single outgoing arc, and it leads to the next occurrence. Finally, once the last occurrence induced by the clauses. Then as long as we are on an occurrence of \( x \) until the last occurrence of either \( l \) digraph. First, we have to choose... 

... that the vertices correspond to the literals set to \( Y \subseteq X \) as an assignment \( \chi_Y : X \rightarrow \{ \text{true, false} \} \), with \( Y = \chi_Y^{-1}(\{ \text{true} \}) \). As expected, we say that a literal \( l \) is set to \( \text{true} \) if \( l \in Y \) or \( l = \neg x \) for some \( x \in X \setminus Y \); otherwise we say that \( l \) is set to \( \text{false} \). So we consider that the vertices correspond to the literals set to \( \text{false} \).

\[ \text{Proof.} \]

... what the paths starting from \( t \) will look like once we have defined the digraph. First, we have to choose \( l_1 \in \{ x_1, \neg x_1 \} \) and go to its first occurrence (first for the order induced by the clauses). Then as long as we are on an occurrence of \( l_1 \) which is not the last one, there is a single outgoing arc, and it leads to the next occurrence. Finally, once the last occurrence of \( l_1 \) is reached, we may go to the first occurrence of \( l_2 \) for some choice \( l_2 \in \{ x_2, \neg x_2 \} \). And so on, until the last occurrence of either \( x_p \) or \( \neg x_p \) which finally allows us to arrive at \( s \).

To enforce this, we define \( A_{\text{var}} \) to consist of all the arcs:

- \( (v_{i,j}, v_{i',j'}) \) such that \( l_{i,j} = l_{i',j'} = l \) and the occurrence of \( l \) in \( C_l \) is the successor of its occurrence in \( C_l' \), i.e., \( i < i' \) and \( i < i'' < i' \implies l \notin C_{l'}; \)
- \( (v_{i,j}, v_{i',j'}) \) such that for some \( (l_{i,j}, l_{i',j'}) \in \bigcup_{1 \leq k \leq p-1} \{ \{ x_k, \neg x_k \} \times \{ x_{k+1}, \neg x_{k+1} \} \} \), \( C_l \) is the last clause containing a literal equal to \( l_{i,j} \) while \( C_{l'} \) is the first clause containing \( l_{i',j'}; \)
- \( (t, v_{i,j}) \) and \( (t, v_{i',j'}) \), where \( l_{i,j} = x_1, l_{i',j'} = \neg x_1 \) and \( C_{i'}, C_{i'} \) are the first clauses in which those literals appear respectively;
- \( (v_{i,j}, s) \) and \( (v_{i',j'}, s) \) for the last occurrences \( l_{i,j}, l_{i',j'} \) of \( x_p, \neg x_p. \)

The next step is to “superimpose” in some way these two graphs \( G_1 \) and \( G_{\text{var}} \) using the following generic construction. This is where we use perfect matchings.

**Lemma 7.** Let \( G_1 = (V \cup \{ s, t \}, A_1) \) and \( G_2 = (V \cup \{ s, t \}, A_2) \) \( s, t \notin V \) be two directed graphs with the same vertex set. Assume that \( G_1 \) and \( G_2 \) are acyclic, and \( s \) (resp. \( t \)) has no incoming edge in \( G_1 \) (resp. \( G_2 \)) and no outgoing edge in \( G_2 \) (resp. \( G_1 \)).

Then one can build in polynomial time a digraph \( G' \) equipped with a perfect matching \( M \) such that the alternating circuits for \( M \) in \( G' \) are in bijection with the pairs \( (P_1, P_2) \) where:

- \( P_1 \) is a path from \( s \) to \( t \) in \( G_1; \)
- \( P_2 \) is a path from \( t \) to \( s \) in \( G_2; \)
- \( P_1 \setminus \{ s, t \} \) and \( P_2 \setminus \{ s, t \} \) are vertex-disjoint.

**Proof.** Our construction for \( G' \) associates to each original vertex a matching edge:

\[ V' = \{ s_1, s_2, t_1, t_2 \} \cup \{ v^d \mid v \in V, d \in \{ \uparrow, \downarrow \} \} \]

\[ M = \{(s_1, s_2), (s_2, s_1), (t_1, t_2), (t_2, t_1)\} \cup \{(v^d, v^{d'}) \mid v \in V, (d, d') \in \{(\uparrow, \downarrow), (\downarrow, \uparrow)\}\} \]

The non-matching edges are obtained from the original edges: \( G' = (V', M \cup A'_1 \cup A'_2) \) where

\[ A'_1 = \{(s_1, v^\uparrow) \mid (s, v) \in A_1\} \cup \{(v^\downarrow, t_1) \mid (u, t) \in A_1\} \cup \{(u^\uparrow, v^\downarrow) \mid u, v \in V, (u, v) \in A_1\} \]

\[ A'_2 = \{(t_2, v^\downarrow) \mid (t, v) \in A_2\} \cup \{(u^\downarrow, s_2) \mid (u, s) \in A_2\} \cup \{(u^\uparrow, v^\downarrow) \mid u, v \in V, (u, v) \in A_2\} \]
so that the subsets of arcs $M$, $A'_1$ and $A'_2$ are disjoint.

Given a pair $(P_1 = s, u_1, \ldots, u_r, t; P_2 = t, v_1, \ldots, v_q, s)$ as specified in the lemma statement, we can build an alternating circuit for $M$ in $G'$:

$$s_1, u_1^+, u_2^+, \ldots, u_r^+, t_1, t_2, v_1^+, v_2^+, \ldots, v_q^+, s_2$$

Conversely, we want to extract a pair of paths $(P_1, P_2)$ from any alternating circuit in $G'$. First, observe that $G'_1 = (V', M \cup A'_1)$ is acyclic (hint: consider the transitive closure of $A_1$ – which, by acyclicity assumption on $G_1$, is a partial order – and take its lexicographic product with the order on $\{ \uparrow, \downarrow \}$ such that $\uparrow \leq \downarrow$) and that $G'_2 = (V', M \cup A'_2)$ is also acyclic for similar reasons.

Therefore, such a circuit cannot be entirely included in either $G_1$ or $G_2$. It must contain two arcs $e_1 \in A_1$ and $e_2 \in A_2$. Let $\pi_i$ be the directed subpath of the circuit starting with $e_i$ and ending with $e_{3-i}$. Then:

- $\pi_1$ contains a subpath $v_1, v_2, v_3, v_4$ with $(v_1, v_2) \in A'_1$, $(v_2, v_3) \in M$ and $(v_3, v_4) \in A'_2$. Since $v_4$ is the target of an arc in $A_1$, either $v_2 = t_2$ or $v_2 = v^i$ for some $v \in V$. In the latter case, we have $v_3 = v^i$, which is impossible for the source of an arc in $A_2$. Therefore $(v_2, v_3) = (t_1, t_2)$.

- $\pi_2$ contains a subpath $v'_1, v'_2, v'_3, v'_4$ with $(v'_1, v'_2) \in A_2$, $(v'_2, v'_3) \in M$ and $(v'_3, v'_4) \in A_1$.

Similarly to the previous case, we conclude that $(v'_2, v'_3) = (s_2, s_1)$.

To recapitulate the discussion: the circuit must switch at some point from arcs in $A_1$ to arcs in $A_2$, and it must also switch back at some point; it can only do the former by crossing $(t_1, t_2)$ and the latter by crossing $(s_2, s_1)$. Therefore, this alternating circuit decomposes into an alternating path $P'_1$ from $s_1$ to $t_1$ in $G'_1$ and an alternating path $P'_2$ from $t_2$ to $s_2$ in $G'_2$, glued together by $(t_1, t_2)$ and $(s_2, s_1)$. These paths are vertex-disjoint because they form a circuit together and a circuit has no vertex repetitions by definition; they can be lifted to yield the desired pair $(P_1, P_2)$.

\[ \square \]

**Remark 8.** This construction is strongly inspired by the proof of [GLMM13, Theorem 5]. For the reader familiar with graph theory: the latter morally proceeds by adding arc directions to a reduction (called Häggkvist’s transformation) from directed graphs to undirected 2-edge-colored graphs; we instead start from a well-known correspondence between directed graphs and perfect matchings in bipartite undirected graphs.

We can now combine these ingredients into a reduction from CNF-SAT to the alternating directed circuit problem.

**Direct proof of Theorem 3.** We apply the construction of the previous lemma to $G_{cl}$ and $G_{var}$ (for $V = V_{occ}$). An alternating circuit in the resulting digraph with perfect matching corresponds to a path $P_{cl}$ from $s$ to $t$ in $G_{cl}$ plus path $P_{var}$ from $t$ to $s$ in $G_{var}$, that are vertex-disjoint except at $s$ and $t$. We have to show that the existence of the latter is equivalent to that of an assignment that satisfies all the clauses $C_1, \ldots, C_n$.

Suppose that we are given such an assignment. First, there exists a unique path $P_{var}$ in $G_{var}$ that visits all literal occurrences set to \texttt{false} (Lemma 6). Since the assignment is satisfying, we may choose in each clause $C_i$ a literal $l_{i,j}$ to \texttt{true}. This corresponds by Lemma 5 to a path $P_{cl}$ in $G_{cl}$. If some vertex of $V_{occ}$ were to appear in both $P_{var}$ and $P_{cl}$, it would mean that the corresponding literal is set both to \texttt{false} and to \texttt{true}.

The converse direction proceeds by a similar reasoning.

\[ \square \]

**References**

[GLMM13] Laurent Gourvès, Adria Lyra, Carlos A. Martínhon, and Jérôme Monnot. Complexity of trails, paths and circuits in arc-colored digraphs. *Discrete Applied Mathematics*, 161(6):819–828, April 2013.
[Ngu18] Lê Thành Dũng Nguyên. Unique perfect matchings and proof nets. In Hélène Kirchner, editor, 3rd International Conference on Formal Structures for Computation and Deduction (FSCD 2018), volume 108 of Leibniz International Proceedings in Informatics (LIPIcs), pages 25:1–25:20, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Ngu20] Lê Thành Dũng Nguyên. Constrained path-finding and structure from acyclicity. CoRR, abs/1901.07028, 2020.

[Ret97] Christian Retoré. Pomset logic: A non-commutative extension of classical linear logic. In Gerhard Goos, Juris Hartmanis, Jan Leeuwen, Philippe Grootes, and J. Roger Hindley, editors, Typed Lambda Calculi and Applications, volume 1210, pages 300–318. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[Ret03] Christian Retoré. Handsome proof-nets: perfect matchings and cographs. Theoretical Computer Science, 294(3):473–488, February 2003.