2013

Nominal Regular Expressions for Languages over Infinite Alphabets

Alexander Kurz
Chapman University, akurz@chapman.edu

Tomoyuki Suzuki
Academy of Sciences of the Czech Republic

Emilio Tuosto
University of Leicester

Nominal Regular Expressions for Languages over Infinite Alphabets

Extended Abstract

Alexander Kurz[1], Tomoyuki Suzuki[2], and Emilio Tuosto[1]

[1] Department of Computer Science, University of Leicester, UK
[2] Institute of Computer Science, Academy of Sciences of the Czech Republic, Czech Republic

Abstract. We propose regular expressions to abstractly model and study properties of resource-aware computations. Inspired by nominal techniques – as those popular in process calculi – we extend classical regular expressions with names (to model computational resources) and suitable operators (for allocation, deallocation, scoping of, and freshness conditions on resources). We discuss classes of such nominal regular expressions, show how such expressions have natural interpretations in terms of languages over infinite alphabets, and give Kleene theorems to characterise their formal languages in terms of nominal automata.

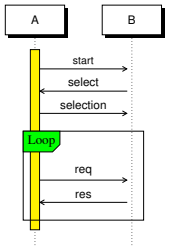
1 Introduction

We equip regular expressions with different types of name binders in order to define a theoretical framework to model and study computations involving resource-handling. In particular, we are interested in computations where resources can be freshly generated, used, and then deallocated. We use names to abstract away from the actual nature of resources; in fact, we adopt a very general notion of computational resources that encompass e.g., memory cells, communication ports, cryptographic keys, threads' identifiers etc. This allows us to use an infinite set $\mathcal{N}$ of *names* to denote resources while binders and freshness conditions formalise the life-cycle of resources. Freshness conditions are taken from the theory of *nominal sets* where $n\#X$ states that $n \in \mathcal{N}$ does not appear (free) in a structure X, which can be a set of names or, more generally, a term built from names and set-theoretic constructions [2]. To this end, $\mathcal{N}$ is equipped with the action of the finitely generated permutations, which then extends to words and languages. Moreover, in the spirit of nominal sets, all languages of interest to us will be closed under the action of permutations.

Together with the use of classical operators of regular languages we then define languages over infinite alphabets including $\mathcal{N}$. As we will see, there are different ways of extending regular expressions with binders or freshness conditions. We consider some natural definitions of nominal regular expressions and give Kleene theorems to characterise their languages in terms of automata.

Besides having interesting theoretical aspects, automata and languages over infinite alphabets can also be adopted to specify and verify properties of systems. This is very much in the spirit of HD-automata which were invented to check equivalence of π-calculus processes. We use a scenario based on distributed choreographies (as those

envisaged by W3C [9]) and show how to specify correct executions of realisations of choreographies, which can be described as message sequence charts. The figure below describes a protocol between two distributed components A and B. After starting the protocol, A waits for a list of services offered by B to select from (for simplicity, data is not represented). Upon request, B replies to A with a list of options to select from. Then A makes her selection and loops to send a number of requests (with a req message) to each of which B replies with a result (with the res message). We describe a few possible realisations of the above choreography.



As a first implementation, think of A and B as repeatedly executing the protocol. This can be conveniently captured using session types [5], where each run of the protocol is uniquely identified with *session names*. Languages over infinite alphabets can suitably specify such runs; for instance, consider

$$\mathcal{L}_{ses} = \left\{ a\,b\,r_0 \cdots r_k \;\middle|\; \forall k \in \mathbb{N}, \forall 0 \leq i \neq j \leq k . r_i \neq r_j \right\}$$

where a and b are two distinct letters representing the two components executing A and B (and $\mathbb{N}$ is the set of natural numbers). A word $a\,b\,r_0 \cdots r_k \in \mathcal{L}_{ses}$ corresponds to a trace where a and b engage in k runs of the protocol and r_i identifies the i-th run.

Another suitable implementation would be one where B is multi-threaded and for instance activates a new thread for each request in the loop. A simplification usually adopted in session-based frameworks is that a thread serving a request cannot be involved in other requests. Assuming that in each loop A makes two requests:

$$\mathcal{L}_{onet} = \left\{ a\,b\,r_0\,p_0\,p'_0 \cdots r_k\,p_k\,p'_k \;\middle|\; \begin{array}{r} \forall k \in \mathbb{N}, \forall 0 \leq i \leq k . p_i \# \{p'_i, r_i\} \wedge p'_i \# \{p_i, r_i\} \\ \wedge r_i \# \{r_0, \ldots, r_{i-1}, p_0, \ldots, p_{i-1}, p'_0, \ldots, p'_{i-1}\} \end{array} \right\}$$

where p_i and p'_i are the names of the two processes that serve the first and the second request from A in the i-th run. Note that in $\mathcal{L}_{onet}$, p_i is not required to be distinct from p_j (or from p'_j).

In yet another realisation, the threads of B would have to activate other threads to serve the requests from A. (In session-based frameworks this is known as *delegation*.) As a simplified model of these traces, let

$$\mathcal{L}_{thr}(r_i) = \bigcup_{h \in \mathbb{N}} \left\{ v_0 d \cdots v_h d \;\middle|\; \forall 0 \leq j \leq h, \exists p \neq p' \in \mathcal{N} \setminus \{r_i\} . v_j \in \{p, p'\}^* \right\}$$

(where d marks when a thread wants to delegate its computation) and consider:

$$\mathcal{L}_{ths} = \bigcup_{k \in \mathbb{N}} \left\{ a b r_0 w_0 r_0 \cdots r_k w_k r_k \;\middle|\; \forall 0 \leq i \leq k . w_i \in \mathcal{L}_{thr}(r_i) \wedge r_i \# r_0 w_0 \cdots r_{i-1} w_{i-1} \right\}$$

An original contribution of this paper is the introduction of *relative global freshness*, a notion of freshness that enables us to control how to *forget* (i.e. deallocate) names. For example, we will see that the languages $\mathcal{L}_{onet}$ and $\mathcal{L}_{ths}$ can be accepted by automata using relative global freshness. In fact, crucially, the freshness condition on the names

for threads allows names to be re-used once the run is finished. This is possible due to the peculiar ability of relative global freshness to "forget" names.

Related to this, we point out that relative global freshness is different from global freshness as defined in [14]. Indeed, the classes of languages we consider are all closed under concatenation, in contrast to [14].

2 Nominal regular expressions

We fix a finite set $\mathcal{S}$ of 'letters' and a countably infinite set $\mathcal{N}$ of 'names' and consider languages over infinite alphabets as sets of finite words over $\mathcal{S} \cup \mathcal{N}$.

We define *nominal regular expressions* (NREs, for short) by extending classical regular expressions with names $n \in \mathcal{N}$ and name binders. We use different types of angled brackets $\langle \_ \rangle$ (decorated with sub- and/or super-scripts) to denote binders. The brackets identify the scope of the binder and are indexed by the name they bind. The interpretation of NREs is defined formally in § 4, here we discuss the basic ideas.

Basic nominal regular expressions (b-NREs) are defined by

$$\mathsf{ne} ::= \mathit{1} \mid \mathit{0} \mid n \mid s \mid \mathsf{ne} + \mathsf{ne} \mid \mathsf{ne} \circ \mathsf{ne} \mid \mathsf{ne}^* \mid \langle_n \mathsf{ne} \rangle_n \tag{1}$$

where *1* and *0* are constants to denote the language consisting of the empty word ε only and the empty language, n and s range over $\mathcal{N}$ and $\mathcal{S}$, resp., the operators $+$, $\circ$, and $\_^*$ are familiar from regular expressions, and $\langle_n \mathsf{ne} \rangle_n$ is our notation for name binding: ne is the scope of the binder and the occurrences of n in ne are bound. For example, in $n \langle_n n \rangle_n n$, the first and the last occurrences of n are free, whilst the occurrence in the bracket is bound and is interpreted as a locally fresh name, that is a name distinct from the occurrences of n outside the scope of the binder. So the language of $n \langle_n n \rangle_n n$ is

$$\left\{ nmn \in \mathcal{N}^3 \;\middle|\; m \in \mathcal{N} \setminus \{n\} \right\}$$

The (de-)allocation mechanism featured by b-NREs is very simple: a fresh name is allocated when entering the scope of a binder and deallocated when leaving. The freshness conditions on the allocated name require that it is distinct from the other currently allocated names. The next class of NREs has a more sophisticated deallocation mechanism.

NREs with permutations (p-NREs) extend b-NREs by permutation actions:

$$\mathsf{ne} ::= \mathit{1} \mid \mathit{0} \mid n \mid s \mid \mathsf{ne} + \mathsf{ne} \mid \mathsf{ne} \circ \mathsf{ne} \mid \mathsf{ne}^* \mid \langle_n \mathsf{ne} \rangle_n^m \tag{2}$$

Novel with respect to to b-NREs is the notation $\rangle_n^m$ which evokes the name transposition $(m\,n)$ to be applied when leaving the scope of the binder. In other words, when $\rangle_n^m$ closes the scope of n, the name m is deallocated while we leak n by replacing all free occurrences of m after $\rangle_n^m$ with n. For example, in the p-NRE $\langle_m m \langle_n n \rangle_n^m m \rangle_m^m$, the occurrences of m after $\rangle_n^m$ actually mean n since m is replaced by n when the scope of n is closed. For example, $\langle_m (\langle_n n \rangle_n^m)^* \rangle_m^m$ is the language of all words where any two successive names must be different, see [10] for more details.

Remark 1. We consider b-NREs as special p-NREs, identifying $\langle_n \mathsf{ne} \rangle_n$ with $\langle_n \mathsf{ne} \rangle_n^n$.

The deallocation device of p-NREs requires some care; intuitively, a name m can be deallocated only *after* it has been allocated. This condition is formalised by requiring that in a p-NRE ne any subexpression $\langle_n \mathsf{ne}'\rangle_n^m$ with $n \neq m$ occurs within the scope of a binder $\langle_{m\,-}\rangle_m^{m'}$. We will consider only p-NREs satisfying this condition.

NREs with underlines (u-NREs) extend b-NREs by *relative global freshness*:

$$\mathsf{ne} ::= 1 \mid 0 \mid n \mid \underline{n} \mid s \mid \mathsf{ne}+\mathsf{ne} \mid \mathsf{ne}\circ\mathsf{ne} \mid \mathsf{ne}^* \mid \langle_n \mathsf{ne}\rangle_n$$

Novel with respect to to b-NREs is the notation $\underline{n}$, which denotes relative global freshness. It requires that the name represented by $\underline{n}$ is distinct from any name allocated *after* n. For instance, the language of the u-NRE $\langle_n n\langle_m mn\rangle_m \underline{n}\langle_m m\rangle_m\rangle_n$ is

$$\left\{nmnn'm' \in \mathcal{N}^5 \;\middle|\; n \neq m \text{ and } n' \neq m, n \text{ and } m' \neq n'\right\}$$

where n' corresponds to the name denoted by $\underline{n}$ in the u-NRE has to be different from both n and m even if the latter has been deallocated. Note the difference with the freshness condition on m' (corresponding to the second binder on m in the u-NRE) which is required to be different only from n'.

NREs with underlines and permutations (up-NREs) combine p-NREs and u-NREs:

$$\mathsf{ne} ::= 1 \mid 0 \mid n \mid \underline{n} \mid s \mid \mathsf{ne}+\mathsf{ne} \mid \mathsf{ne}\circ\mathsf{ne} \mid \mathsf{ne}^* \mid \langle_n \mathsf{ne}\rangle_n^m$$

where the conditions on p-NREs also hold for up-NREs.

Examples of NREs for the languages of § 1 are

Language	Corresponding NRE	Type of NRE
$\mathcal{L}_{ses}$	$ab\langle_n \underline{n}^*\rangle_n$	u-NRE
$\mathcal{L}_{onet}$	$ab\langle_n \left(\underline{n}\langle_m m\langle_l l\rangle_l\rangle_m\right)^*\rangle_n$	u-NRE
$\mathcal{L}_{ths}$	$ab\langle_n \left(\underline{n}\langle_m \left(\langle_l (m+l)^* d\rangle_l^m + \langle_l (m+l)^* d\rangle_l\right)^*\rangle_m\right)^*\rangle_n$	up-NRE

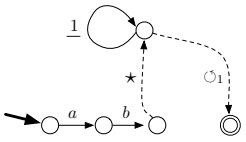
This correspondences are obtained by applying the method presented in § 4. The details of a more complex example are given in Appendix A.

3 Chronicle deallocating automata

The class of *chronicle deallocating automata* characterises languages over $\mathcal{S} \cup \mathcal{N}$. Ex. 1 below gives an intuition of our automata which are defined in Def. 1.

Example 1. The language $\mathcal{L}_{ses}$ in § 1 is accepted by the automaton in the figure below

that first consumes the two letters a and b, then allocates a locally fresh name in register 1 *with the* $\star$*-transition (without consuming any letter). By means of repeated* $\underline{1}$*-transitions, it can consume* $r_0,\ldots,r_k$ *guaranteeing their freshness with respect to the chronicle for the first register. Finally, the name in the first register is deallocated and the final state is reached.*



Given a natural number k, in the following, $\mathsf{reg}(k) \stackrel{\text{def}}{=} \{1,\ldots,k\}$ denotes a set of k registers. The empty word is denoted by ε.

Definition 1. *A* chronicle deallocating automaton (over $\mathcal{S}$) *is a finite-state automaton* $\langle Q, q_0, F, tr\rangle$ *with states* Q*, initial state* $q_0 \in Q$*, final states* $F \subseteq Q$ *and transition relation* tr *such that*

- Q *is equipped with a map* $\|\_\| : Q \to \mathbb{N}$ *such that* $\|q\| = 0$ *for all* $q \in F \cup \{q_0\}$*;*
- *writing* $\mathsf{reg}(q)$ *for* $\mathsf{reg}(\|q\|)$*, each* $q \in Q$ *has a set of possible labels*

$$\mathfrak{L}(q) \stackrel{\text{def}}{=} \mathcal{S} \cup \mathsf{reg}(q) \cup \{\star\} \cup \{\underline{i} \mid i \in \mathsf{reg}(q)\} \cup \{\circlearrowleft_i \mid i \in \mathsf{reg}(q)\}$$

- *for* $q \in Q$ *and* $\alpha \in \mathfrak{L}(q) \cup \{\varepsilon\}$*, the set* $tr(q,\alpha) \subseteq Q$ *contains the* α-successor states *of* q *satisfying the conditions below for all* $q' \in tr(q,\alpha)$*:*

$$\begin{array}{ll} \|q'\| = \|q\| + 1, & \text{if } \alpha = \star \\ \|q'\| = \|q\| - 1, & \text{if } \alpha = \circlearrowleft_i \text{ for } i \in \mathsf{reg}(q) \\ \|q'\| = \|q\|, & \text{if } \alpha = \varepsilon \text{ or } \alpha \in \mathcal{S} \cup \mathsf{reg}(q) \cup \{\underline{i} \mid i \in \mathsf{reg}(q)\} \end{array}$$

We let CDA$^\sharp$ *denote the class of chronicle deallocating automata. A* CDA$^\sharp$ *is (i)* deterministic *if, for each* $q \in Q$*,* $|tr(q,\varepsilon)| = 0$ *and* $|tr(q,\alpha)| = 1$*, if* $\alpha \in \mathfrak{L}(q)$*, (ii) a* chronicle *automaton (*CA$^\sharp$*) if for all* $q \in Q$ *there is no* $\circlearrowleft_i$*-transition for* $i \in \{1,\ldots,\|q\|-1\}$*, (iii) a* deallocating *automaton (*DA$^\sharp$*) if for all* $q \in Q$ *there is no* $\underline{i}$*-transition for* $i \in \{1,\ldots,\|q\|-1\}$*, and (iv) an automaton* with freshness *(*A$^\sharp$*) if it is both* CA$^\sharp$ *and* DA$^\sharp$*.*

Configurations of CDA$^\sharp$ are defined in terms of *chronicles* that keep track of names assigned to registers. The chronicle s_i of a register i is a non-empty word on $\mathcal{N}$ together with one of the names in s_i that pinpoints the *current value* of i, denoted as $cv(s_i)$. Let s and t be chronicles. The *extension* $s@t$ of s with t is the concatenation of the words s and t; while $s \setminus t$ is the word obtained by deleting from s the names in t.

We write $L = [e_1,\ldots,e_k]$ for a list of elements $e_1,\ldots,e_k$ (with $[]$ being the empty list) and define $L[\![i]\!] = e_i$. An *extant chronicle* is a (possibly empty) finite list $E = [s_1,\ldots,s_k]$ of chronicles; we define $cv(E) = [cv(s_1),\ldots,cv(s_k)]$, which is always a list of pairwise distinct names. We extend @ and $\setminus$ to extant chronicles element-wise: $E@s = [s_1@s,\ldots,s_k@s]$ and $E \setminus t = [s_1 \setminus t,\ldots,s_k \setminus t]$. We may identify a list with the underlying set of its elements (e.g. writing $e \in L$ when there is i such that $e = L[\![i]\!]$ and $n \in s$ when n occurs in the chronicle s). Also, for an extant chronicle E, $cv(E)[\![i]\!]$ and $E[\![i]\!]$ indicate the current value and the chronicle of a register i, respectively. Given two extant chronicles E and E', we let $E + E'$ be the list obtained by appending E' to E.

Definition 2. *A* configuration *of a CDA$^\sharp$ $\mathcal{H} = \langle Q, q_0, F, tr\rangle$ is a triple $\langle q, w, E\rangle$ where $q \in Q$, w is a word, and E is an extant chronicle. A configuration $\langle q, w, E\rangle$ is* initial *if $q = q_0$ and $E = []$, and it is* accepting *if $q \in F$, $w = \varepsilon$ and $E = []$. Given two configurations $t = \langle q, w, E\rangle$ and $t' = \langle q', w', E'\rangle$, $\mathcal{H}$* moves from t to t' *(written as $t \xrightarrow{\mathcal{H}} t'$) if there is $\alpha \in \mathfrak{L}(q) \cup \{\varepsilon\}$ such that $q' \in tr(q, \alpha)$ and*

$$\begin{cases} \alpha \in \mathsf{reg}(q)\,, \ \textit{if} \quad w = (cv(E)[\![\alpha]\!])w' \quad \textit{and} \quad E' = E \\ \alpha \in \mathcal{S} \cup \{\varepsilon\}, \ \textit{if} \quad w = \alpha w' \quad \textit{and} \quad E' = E \\ \alpha = \star, \qquad \textit{if} \quad w = w', \quad n \in \mathcal{N} \setminus cv(E)\,, \quad E' = (E@n) + [n], \\ \qquad\qquad cv(E')[\![\|q'\|]\!] = n \quad \textit{and} \quad \forall i \in \mathsf{reg}(q)\,.cv(E')[\![i]\!] = cv(E)[\![i]\!] \\ \alpha = \underline{i}, \qquad \textit{if} \quad w = nw', \quad n \in \mathcal{N} \setminus (cv(E) \cup E[\![i]\!])\,, \quad E' = E@n, \\ \qquad\qquad cv(E')[\![i]\!] = n \quad \textit{and} \quad \forall j \in \mathsf{reg}(q') \setminus \{i\}.cv(E')[\![j]\!] = cv(E)[\![j]\!] \\ \alpha = \circlearrowleft_i, \qquad \textit{if} \quad w = w', \quad E' + E[\![\mathsf{reg}(q)]\!] = E, \quad cv(E')[\![i]\!] = cv(E)[\![\|q\|]\!] \\ \qquad\qquad\qquad\qquad\qquad \textit{and} \quad \forall j \in \mathsf{reg}(q') \setminus \{i\}.cv(E')[\![j]\!] = cv(E)[\![j]\!] \end{cases}$$

The set $reach_{\mathcal{H}}(t)$ of states reached by $\mathcal{H}$ from the configuration t is given by

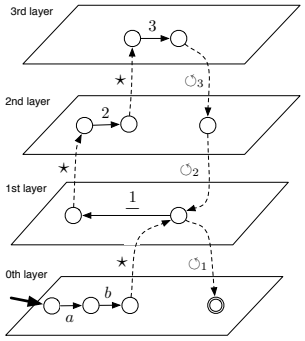
$$reach_{\mathcal{H}}(t) \stackrel{\text{def}}{=} \begin{cases} \{q\} & \textit{if } t = \langle q, \varepsilon, E\rangle \\ \bigcup_{t \xrightarrow{\mathcal{H}} t'} reach_{\mathcal{H}}(t') & \textit{otherwise} \end{cases}$$

A run of $\mathcal{H}$ on a word w *is a sequence of moves of $\mathcal{H}$ from $\langle q_0, w, []\rangle$.*

Intuitively, $\star$-transitions allocate a new register as the highest register of the target state (given $N \subseteq \mathcal{N}$ and $n \in \mathcal{N}$, we write $n\#N$, read 'n is fresh for N', when $n \notin N$). Such a register is initially assigned with a *locally fresh* name n, that is a name n fresh for the current content of the registers; accordingly the chronicle of the new register is created and initialised (together with the updates of the other chronicles) to record the use of n. Relative global freshness is implemented by $\underline{i}$-transitions that (as for local freshness) assign a name n fresh with respect to the current values of registers and (unlike in local freshness) also fresh with respect to the i-th register's chronicle; contextually, n is assigned to the i-th register and all the other chronicles are updated to record the use of the new name n. Transitions labelled by $\circlearrowleft_i$ permute the content, but not the chronicle, of the i-th register with the highest register of the current state q and dispose the chronicle of the $\mathsf{reg}(q)$-th register of q.

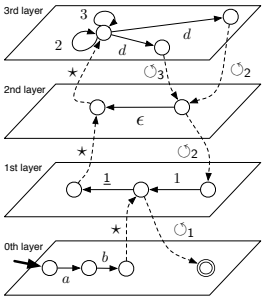
Definition 3. *Given a word w on $\mathcal{S} \cup \mathcal{N}$, a CDA$^\sharp$ $\mathcal{H}$* accepts *(or* recognises*) w when $F \cap reach_{\mathcal{H}}(\langle q_0, w, []\rangle) \neq \emptyset$. The set $\mathcal{L}_{\mathcal{H}}$ of words accepted by $\mathcal{H}$ is* the language of $\mathcal{H}$.

Example 2. The automaton for the language $\mathcal{L}_{onet}$ in § 1 is more complex than the one in Ex. 1. Initially the automaton behaves as the one in Ex. 1. After the $\underline{1}$-transition, it allocates the second and the third registers to consume p_0 and p_0'. Note that p_0 and p_0' are also recorded into the chronicle of the first register;



hence, when the automaton comes back to the first layer by the $\circlearrowleft_3$ *and* $\circlearrowleft_2$ *transitions, the next consumed name* r_1 *is guaranteed to be globally fresh (namely* r_1 *is different from* r_0*,* p_0 *and* p'_0*). On the other hand, the other registers cannot remember anything after they have been deallocated. So when the automaton goes up again to the second and the third layers by* $\star$*-transitions, it has two registers for* locally *fresh names for* p_1 *and* p'_1 *with respect to the current session name* r_1*. And, for instance, one of them can be* r_0*.*

Example 3. The automaton for the language $\mathcal{L}_{ths}$ *in § 1 is the most complex and is given below. The automaton allocates a new register to consume a name* r_i *for each run* i *by means of the* $\underline{1}$*-transition on the first layer. From there the automaton has to accept the* $w_i \in \mathcal{L}_{thr}(r_i)$*. Similarly to the previous cases, two* $\star$*-transitions allocates the registers to accept two locally fresh names* p *and* p'*. Now the automaton can loop in the left-most state of the third layer consuming the occurrences of* p *and* p'*. When* d *appears on the input, the automaton non-deterministically decides which process to deallocate with transitions* $\circlearrowleft_2$ *or* $\circlearrowleft_3$*. Now the* ε*-transition lets the automaton accept a new thread or deallocate the other register and start a new session after consuming the name of the current run stored in the first register. After delegating two threads, we finish the session* r_i *by the* 1*-transition on the first layer.*



By adding to the automaton of Ex. 3 an ε-transition from the final state to the initial state, we can repeat the whole process by reusing also the names of previous runs (since the first chronicle is released when the automaton moves back to the lowest layer).

4 Interpreting NREs

We formally define the language associated with a nominal regular expression. Technically, we adapt the *context* and the *language calculus* [10] for the new classes of NREs.

Schematic words. To assign languages over infinite alphabets to NREs, it is natural to introduce the notion of *schematic words*. We consider a countably infinite collection of *placeholders* $\maltese_i$ A *schematic word* $[\![\maltese_1 \cdots \maltese_k \mid \phi]\!]$ consists of a finite word of placeholders and a condition ϕ of the form

$$\phi ::= \maltese_i \neq \maltese_j \;\mid\; \phi, \phi \;\mid\; \phi \vee \phi$$

with the intention that $\maltese_i \neq \maltese_j$ means that $\maltese_i$ and $\maltese_j$ are not identical, ',' means 'and' and $\vee$ means 'or'. So, for example, a schematic word $[\![\maltese_1\maltese_2\maltese_3\maltese_1 \mid \maltese_1 \neq \maltese_3]\!]$ expresses the collection of words whose third letter is different from the first and the last letters (but the second letter can be any name): $\{abca \mid a,b,c \in \mathcal{N}, a \neq c\}$.

Languages over infinite alphabets recognised by "nominal automata" are typically closed under permutations, see e.g. Proposition 1. Schematic words describe such languages by means of inequations (freshness) of names. In Fig. 2, we shall use schematic words which are extended to contain names from $\mathcal{N}$ (as well as placeholders).

From maps to permutations. Given a function f with domain $dom(f)$, the update $f_{[a\mapsto b]}$ has domain $dom(f)\cup\{a\}$ with $f_{[a\mapsto b]}(a)=b$; $\perp$ is the empty map. We consider lists over $\mathcal{N}$ with no repeated elements (ranged over by N, M). Let $lth(N)$ be the length of N and, for $n\in\mathcal{N}$. The transposition of n and m, denoted by $(m\,n)$, is the bijection that swaps m and n and is the identity on any other names. Given two lists N and M of length k, let $N\triangleright M$ be the map from N to M such that

$$N\triangleright M : N[\![i]\!]\mapsto M[\![i]\!] \qquad \text{for each } i\in\{1,\dots,k\}$$

which we extend it to a permutation $\pi_{[N\triangleright M]}$ on $\mathcal{N}$ that restricts to a bijection on $N\cup M$ and to the identity on $\mathcal{N}\setminus(N\cup M)$, see also [10]. In Fig. 2, to transfer name (placeholder) information, we consider the above permutation for a list of current values $C=[n_1,\dots,n_k]$ and an extant chronicle $E=[s_1,\dots,s_k]$, with respect to the natural bijection: $n_i\mapsto cv(s_i)$ for each i (abusing notation, we write $\pi_{[C\triangleright E]}$ for $\pi_{[C\triangleright cv(E)]}$). Also, we may include some placeholders for $\pi_{[C\triangleright E]}$.

Permutations on expressions and extant chronicles. For an up-NRE ne and a bijection π on $\mathcal{N}$, the permutation action of π on ne, denoted as $\pi\cdot\mathsf{ne}$, is

1. $\pi\cdot 1=1$ $\qquad$ $\pi\cdot 0=0$ $\qquad$ $\pi\cdot n=\pi(n)$ $\qquad$ $\pi\cdot\underline{n}=\underline{\pi(n)}$ $\qquad$ $\pi\cdot s=s$
2. $\pi\cdot(\mathsf{ne}_1+\mathsf{ne}_2)=(\pi\cdot\mathsf{ne}_1)+(\pi\cdot\mathsf{ne}_2)$ $\qquad$ $\pi\cdot(\mathsf{ne}_1\circ\mathsf{ne}_2)=(\pi\cdot\mathsf{ne}_1)\circ(\pi\cdot\mathsf{ne}_2)$
3. $\pi\cdot(\mathsf{ne}^*)=(\pi\cdot\mathsf{ne})^*$ $\qquad$ $\pi\cdot(\langle_n\mathsf{ne}\rangle_n^m)=\langle_{\pi(n)}(\pi\cdot\mathsf{ne})\rangle_{\pi(n)}^{\pi(m)}$

The permutation action of π on a chronicle $s_i=n_{i_1}\dots n_{i_k}$, denoted as $\pi\cdot s_i$, is $t=\pi(n_{i_1})\dots\pi(n_{i_k})$ with $cv(t)=\pi\cdot cv(s_i)$. Finally, the permutation action of π on an extant chronicle $E=[s_1,\dots,s_k]$ is $\pi\cdot E=[\pi\cdot s_1,\dots,\pi\cdot s_k]$. Note that we may include placeholders in contexts in Fig. 2.

Contextualised expressions are triples $C\ddagger\mathsf{ne}\ddagger E$ where ne is a nominal regular expression, C is a finite list of pairwise distinct (including indices) names and placeholders, i.e. $C\in(\mathcal{N}\cup\{\maltese,\maltese_1,\dots\})^*$ (called *pre-context*) and E is an extant chronicle (*post-context*). To compute languages from NREs, we may include placeholders in (extant) chronicles. Placeholders appear in contexts in the language calculus Fig. 2 only to abstract some names. Intuitively, C is the list of the names and placeholders "used before" ne and E is the extant chronicle "established after" ne. More precisely, the post-context E possesses two important data: for CTXC and the construction of corresponding automata (on the inductive step for $\langle_n\mathsf{ne}\rangle_n^m$), it tells "which registers are permuted when brackets are closes" and, for LNGC, it reserves numbers of relative-global fresh names for each register, which is necessary to consider ($\check{\mathsf{o}}$) in Fig. 2. It is useful to explicitly express the current values of (extant) chronicles and write $n\natural s$ for the chronicle s with $cv(s)=n$ and $[n_1\natural s_1,\dots,n_k\natural s_k]$ for an extant chronicle $E=[s_1,\dots,s_k]$ with $cv(E)=[n_1,\dots,n_k]=[cv(s_1),\dots,cv(s_k)]$.

$$\frac{C \ddagger \mathsf{ne}_1 + \mathsf{ne}_2 \ddagger E}{C \ddagger \mathsf{ne}_1 \ddagger E}\,(\hat{+}_1) \qquad \frac{C \ddagger \mathsf{ne}_1 + \mathsf{ne}_2 \ddagger E}{C \ddagger \mathsf{ne}_2 \ddagger E}\,(\hat{+}_2)$$

$$\frac{C \ddagger \mathsf{ne}_1 \circ \mathsf{ne}_2 \ddagger E}{C \ddagger \mathsf{ne}_1 \ddagger \mathbb{C} \qquad C \ddagger \mathsf{ne}_2 \ddagger E}\,(\hat{\circ}) \qquad \frac{C \ddagger \mathsf{ne}^* \ddagger E}{C \ddagger \underbrace{\mathsf{ne} \circ \cdots \circ \mathsf{ne}}_{h \text{ times}} \ddagger E}\,(\hat{*})$$

$$\frac{C \ddagger \langle_n \mathsf{ne} \rangle_n^m \ddagger E \qquad m = n \qquad C' = C + [\star] \qquad E' = (E@\star) + [\star]}{C' \ddagger (n\,\star) \cdot \mathsf{ne} \ddagger cv(E') \natural E'}\,(\hat{\Diamond}_=)$$

$$\frac{C \ddagger \langle_n \mathsf{ne} \rangle_n^m \ddagger E \quad m \neq n \quad C' = C + [\star] \quad E' = (E@\star) + [\star m] \quad cv(E') = cv(E) + [\star]}{C' \ddagger (n\,\star) \cdot \mathsf{ne} \ddagger ((m\,\star) \cdot cv(E')) \natural E'}\,(\hat{\Diamond}_\neq)$$

Fig. 1. CTXC: Rules for contextualised expressions

The context calculus CTXC is defined in Fig. 1, where we assume that $\star \# C$, $C = [n_1, \ldots, n_k]$ is a list of pairwise distinct names, and $\mathbb{C} = [n_1 \natural s_1, \ldots, n_k \natural s_k]$ is the extant chronicle where for each $1 \leq i \leq k$, $s_i = n_i \cdots n_k$. As in [10], the rules in Fig. 1 propagate pre- and post-contexts to all subexpressions of ne with a top-down visit of the abstract syntax tree of ne. The rule $(\hat{*})$ unfolds the Kleene star an arbitrary but bound number of times h; later (c.f. LNGC in Fig. 2) we will take the union of the languages computed for each unfolding. By rules $(\hat{\Diamond}_=)$ and $(\hat{\Diamond}_\neq)$ it is clear that in pre-contexts it is necessary just to record the names already used from the root to the current node of the tree. Instead, in the post-context it is necessary to keep track of the "names created" in the current subexpression for relative global freshness. We will see that this is crucial for computing the local freshness and the concatenation of the languages of NREs. Notice that in rule $(\hat{\Diamond}_\neq)$, for an extant chronicle $E = [s_1, \ldots, s_k]$, the notation $cv(E) \natural E$ abbreviates $[s_1 \natural cv(s_1), \ldots, s_k \natural cv(s_k)]$.

Remark 2. For NREs without permutations, we do not need to consider the rule $(\hat{\Diamond}_\neq)$ in Fig. 1. For NREs without underlines, only current values of registers matter.

Note how $(\hat{\Diamond}_\neq)$ deals with permutations: the expression ne in the binder is contextualised by a local renaming of the (content corresponding to) n with $\star$ which "after" ne (namely in the post-context) is also replaced for m in the current values, while m is added to the chronicle corresponding to the new register.

Example 4. For the up-NRE $\langle_n \underline{n} \langle_m \langle_l m \rangle_l^m \rangle_m \underline{n} \rangle_n$, contexts are computed as

$$\dfrac{\dfrac{[] \ddagger \langle_n \underline{n} \langle_m \langle_l m \rangle_l^m \rangle_m \underline{n} \rangle_n \ddagger []}{[a] \ddagger \underline{a} \langle_m \langle_l m \rangle_l^m \rangle_m \underline{a} \ddagger [a \natural a]}\,(\hat{\Diamond}_\neq)}{[a] \ddagger \underline{a} \ddagger [a \natural a] \qquad \dfrac{[a] \ddagger \langle_m \langle_l m \rangle_l^m \rangle_m \underline{a} \ddagger [a \natural a]}{\dfrac{\dfrac{[a] \ddagger \langle_m \langle_l m \rangle_l^m \rangle_m \ddagger [a \natural a]}{[a,b] \ddagger \langle_l b \rangle_l^b \ddagger [a \natural ab, b \natural b]}\,(\hat{\Diamond}_=)}{[a,b,c] \ddagger b \ddagger [a \natural abc, c \natural bc, b \natural cb]}\,(\hat{\Diamond}_=) \qquad [a] \ddagger \underline{a} \ddagger [a \natural a]}\,(\hat{\circ})\,(\hat{\circ})$$

Note that the up-NRE is quite similar to a simple trace of $\mathcal{L}_{ths}$ in § 1.

$$\frac{C \ddagger 1 \ddagger E}{C \ddagger [\![\varepsilon \mid]\!] \ddagger E}\ (1) \qquad \frac{C \ddagger 0 \ddagger E}{C \ddagger [\![\emptyset \mid \bot]\!] \ddagger E}\ (0) \qquad \frac{C \ddagger s \ddagger E}{C \ddagger [\![s \mid]\!] \ddagger E}\ (s) \qquad \frac{C \ddagger n \ddagger E}{C \ddagger [\![n \mid]\!] \ddagger E}\ (n)$$

$$\frac{C \ddagger \underline{n} \ddagger E \qquad C[\![i]\!] = n}{C \ddagger [\![\maltese \mid \maltese \# C, \maltese \underline{\#}^i \mathbb{C}[\![i]\!]]\!] \ddagger ((n\,\maltese)\cdot cv(E)) \natural (E @ \maltese)}\ (\underline{n})$$

$$\frac{C \ddagger [\![\bullet_1^1 \cdots \bullet_{k_1}^1 \mid \phi_1]\!] \ddagger E_1 \qquad C \ddagger [\![\bullet_1^2 \cdots \bullet_{k_2}^2 \mid \phi_2]\!] \ddagger E_2}{C \ddagger [\![\bullet_1^1 \cdots \bullet_{k_1}^1 @ \pi_{[C \rhd E_1]} \cdot \left(\bullet_1^2 \cdots \bullet_{k_2}^2\right) \mid \phi_1, \left(\pi_{[C \rhd E_1]} \cdot \phi_2\right)]\!] \ddagger E_1 @ \left(\pi_{[C \rhd E_1]} \cdot E_2\right)}\ (\eth)$$

$$\frac{C + [n] \ddagger [\![\bullet_1 \cdots \bullet_k \mid \phi]\!] \ddagger E + [m \natural t]}{C \ddagger [\![(n\,\maltese)\cdot \bullet_1 \cdots \bullet_k \mid \maltese \# C, ((n\,\maltese)\cdot \phi)]\!] \ddagger (n\,\maltese)\cdot E}\ (\hat{\diamond})$$

Fig. 2. LNGC: Rules for computing schematic words

The language calculus (LNGC for short) is given in Fig. 2 (the new notations are explained below in the comment of the rules). Given an NRE ne, the rules in Fig. 2 are meant to be applied "bottom up" to the proof trees computed by the CTXC starting from the contextualised expressions $[] \ddagger \mathsf{ne} \ddagger []$. Also, in each instance of the rules $\hat{\diamond}$ and $\underline{n}$ a completely fresh placeholder has to be introduced. Therefore, in the LNGC, we do not have rules for $\_^*$ nor $+$. Instead, we take unions for $\_^*$ and $+$ after computing languages of each tree. On each application of the rules $(\underline{n})$ and $(\hat{\diamond})$, we must use a new $\maltese$, see eg the occurrence of $\maltese_1$ and $\maltese_4$ in Ex. 5. We use $\bullet$'s to range over $\mathcal{N} \cup \{\maltese, \maltese_1, \ldots\}$.

Remark 3. We can use the same pre-context C in both premises of rule $(\eth)$ in Fig. 2 because CTXC duplicates the pre-contexts when decomposing the expression while LNGC recovers the same pre-contexts when visiting the tree.

Rules (*1*), (*0*), (*n*) and (*s*) yield the natural interpretation for the corresponding elementary expressions. Note that $[\![\emptyset \mid \bot]\!]$ in rule (*0*) generates nothing by concatenations of schematic words. The remaining rules are more delicate since our NREs encompass both freshness "with respect to current values" (local freshness) and freshness "with respect to chronicles" (relative global freshness). Therefore, it is important to identify which names have to be locally and which relative globally fresh. Formally, this is done by noticing that each language obtained by LNGC from $+$-free NREs can be expressed as a finite conjunction of freshness conditions that we write as $\maltese_1 \# S_1 \text{ and } \cdots \text{ and } \maltese_r \underline{\#}^i S'_r$ where $S_j, S'_{j'}$ are lists of names in C or some placeholders $\maltese$ (note that $\#$ is not the $\#$ operation, it is just a syntactic device to mark the type of freshness required on placeholders; similarly, $\underline{\#}^i$ represent marks that relative global freshness with respect to the chronicles of the i-th regist is required).). This presentation of a language can be obtained by inspecting the corresponding NRE and noting the conditions on names that occur in the scopes of binders.

Rule $(\underline{n})$ is for relative global freshness; $\maltese \# C$ means that $\maltese$ is locally fresh for current values C and $\maltese \underline{\#}^i \mathbb{C}[\![i]\!]$ for $C[\![i]\!] = n$ does that $\maltese$ is fresh with respect to the i-th chronicle.

Rule ($\check{\circ}$) deals with the concatenation of two languages by attaching each schematic word v of second language to each schematic word w of the first; note that, since permutations and underlines may change the post-contexts, it is necessary to use a permutation $\pi_{[C\rhd E_1]}$ to rename everything and update global fresh information before concatenating schematic words. Given two schematic words $[\![\bullet_1^1\cdots\bullet_{k_1}^1 \mid \phi_1]\!]$ and $[\![\bullet_1^2\cdots\bullet_{k_2}^2 \mid \phi_2]\!]$, we append the first schematic word with the second schematic word permuted by $\pi_{[C\rhd E_1]}$. In addition, we also update the freshness condition ϕ_2. There are two types of "updates" in $\pi_{[C\rhd E_1]}\cdot\phi_2$: update for local freshness $\#$ and relative global freshness $\underline{\#}^i$: see rules ($\underline{n}$) and ($\check{\diamond}$). For the local freshness $\#$ in ϕ_2, say $\bullet_j\#\bullet_{j_1}\cdots\bullet_{j_l}$, we just replace this condition as $\pi_{[C\rhd E_1]}\cdot\bullet_j\#\pi_{[C\rhd E_1]}\cdot(\bullet_{j_1}\cdots\bullet_{j_l})$ in $\pi_{[C\rhd E_1]}\cdot\phi_2$. And for the global freshness with respect to the register i $\underline{\#}^i$ in ϕ_2, say $\bullet_j\underline{\#}^i\bullet_{j_1}\cdots\bullet_{j_l}$, we replace this condition "with appending" the register i's chronicle freshness information as $\pi_{[C\rhd E_1]}\cdot\bullet_j\underline{\#}^iE_1[\![i]\!]@\left(\pi_{[C\rhd E_1]}\cdot(\bullet_{j_1}\cdots\bullet_{j_l})\right)$. Therefore, for $\underline{\#}^i$, we not only permute the freshness conditions but also update the corresponding previous chronicle information in E_1 in $\pi_{[C\rhd E_1]}\cdot\phi_2$. We also note that, for $E_1@\left(\pi_{[C\rhd E_1]}\cdot E_2\right)$ in the same rule, we keep the current values of the whole extant chronicle $cv\left(\pi_{[C\rhd E_1]}\cdot E_2\right)$, i.e. $\pi_{[C\rhd E_1]}\cdot cv(E_2)$.

Rule ($\check{\diamond}$) deallocates the last name n in the pre-context and the last chronicle t in the post-context, i.e $m\natural t$. Accordingly, we abstract the name n to a placeholder $\diamondsuit$.

Ex. 5 below shows an application of the rules; the reader is referred to Appendix A for an example of a more complex language.

Example 5. To show the difference between local and global freshness we compute the language considered in Ex. 4:

$$\dfrac{\dfrac{[a]\ddagger\underline{a}\ddagger[a\natural a]}{[a]\ddagger[\diamondsuit_1\mid\diamondsuit_1\#^{\partial}a]\ddagger[\diamondsuit_1\natural a\diamondsuit_1]}(\underline{a})}{\mathfrak{L}_l}\qquad\dfrac{\dfrac{[a]\ddagger\underline{a}\ddagger[a\natural a]}{[a]\ddagger[\diamondsuit_4\mid\diamondsuit_4\#^{\partial}a]\ddagger[\diamondsuit_4\natural a\diamondsuit_4]}(\underline{a})}{\mathfrak{L}_r}$$

$$\dfrac{\dfrac{\mathfrak{L}_l\qquad\dfrac{\dfrac{\dfrac{\dfrac{[a,b,c]\ddagger b\ddagger[a\natural abc,c\natural bc,b\natural cb]}{[a,b,c]\ddagger[b\mid]\ddagger[a\natural abc,c\natural bc,b\natural cb]}(b)}{[a,b]\ddagger[b\mid\diamondsuit_3\#ab]\ddagger[a\natural ab\diamondsuit_3,\diamondsuit_3\natural b\diamondsuit_3]}(\check{\diamond})}{[a]\ddagger[\diamondsuit_2\mid\diamondsuit_2\#a,\diamondsuit_3\#a\diamondsuit_2]\ddagger[a\natural a\diamondsuit_2\diamondsuit_3]}(\check{\diamond})\qquad\mathfrak{L}_r}{[a]\ddagger[\diamondsuit_2\diamondsuit_4\mid\diamondsuit_2\#a,\diamondsuit_3\#a\diamondsuit_2,\diamondsuit_4\#^{\partial}a\diamondsuit_2\diamondsuit_3]\ddagger[\diamondsuit_4\natural a\diamondsuit_2\diamondsuit_3a\diamondsuit_4]}(\hat{\diamond})}{[a]\ddagger[\diamondsuit_1\diamondsuit_2\diamondsuit_4\mid\diamondsuit_1\#^{\partial}a,\diamondsuit_2\#\diamondsuit_1,\diamondsuit_3\#\diamondsuit_1\diamondsuit_2,\diamondsuit_4\#^{\partial}a\diamondsuit_1\diamondsuit_2\diamondsuit_3]\ddagger[\diamondsuit_4\natural a\diamondsuit_1\diamondsuit_1\diamondsuit_2\diamondsuit_3\diamondsuit_1\diamondsuit_4]}(\hat{\diamond})}{[]\ddagger[\diamondsuit_1\diamondsuit_2\diamondsuit_4\mid\diamondsuit_1\#^{\partial}\diamondsuit,\diamondsuit_2\#\diamondsuit_1,\diamondsuit_3\#\diamondsuit_1\diamondsuit_2,\diamondsuit_4\#^{\partial}\diamondsuit\diamondsuit_1\diamondsuit_2\diamondsuit_3]\ddagger[]}(\check{\diamond})$$

where, for compactness, conditions of the form $\diamondsuit\#S$ and $\diamondsuit\underline{\#}^1S'$ are abbreviated as $\diamondsuit\#^{\partial}S'$ provided that $S\subset S'$). Note how $\underline{\#}^1$ (and $\#^{\partial}$) differs from $\#$. Since the expression's depth is at most three, only by local freshness $\#$, we cannot encounter freshness conditions with respect to more than four other placeholders, as $\diamondsuit_4\#^{\partial}\diamondsuit\diamondsuit_1\diamondsuit_2\diamondsuit_3$.

The language of an up-NRE ne with no free names is obtained by three steps: 1. compute schematic words with LNGC on all the proof-trees generated by CTXC starting with $[]\ddagger\mathsf{ne}\ddagger[]$, 2. interpret all schematic words naturally into languages over infinite alphabets, and 3. take the union of all the languages. We denote the language obtained from ne by $\mathbf{L}(\mathsf{ne})$.

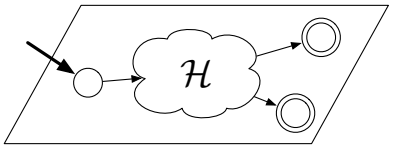


Fig. 3. The CDA$^\sharp$ $\mathcal{H}_{(\!|\mathsf{ne}|\!)}$.

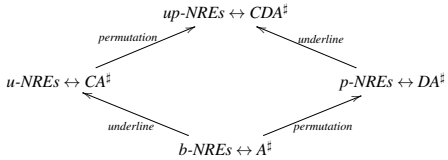
Definition 4. *A language over infinite alphabets is* nominal regular *if there is an up-NREs* ne *such that the language is* $\mathbf{L}(\mathsf{ne})$.

Proposition 1. *All nominal regular expressions are closed under* α*-equivalence.*

5 Kleene theorems

We now give our main results.

Theorem 1. *Nominal regular languages are accepted by CDA$^\sharp$ as follows:*



- *languages described by up-NREs are accepted by CDA$^\sharp$*
- *languages described by u-NREs are accepted by CA$^\sharp$*
- *languages described by p-NREs are accepted by DA$^\sharp$*
- *languages described by b-NREs are accepted by A$^\sharp$.*

For up-NREs, we inductively construct the corresponding CDA$^\sharp$. To do so, we extend the notion of CDA$^\sharp$ to CDA$^\sharp$ *in-contexts*, $C \ddagger \mathcal{H}_{(\!|\mathsf{ne}|\!)} \ddagger E$, languages to languages *in-contexts*, $C \ddagger \mathbf{L}(\mathsf{ne}) \ddagger E$, and up-NREs to up-NREs *in-contexts*, $C \ddagger \mathsf{ne} \ddagger E$. For automata-in-contexts $C \ddagger \mathcal{H}_{(\!|\mathsf{ne}|\!)} \ddagger E$ with $\mathcal{H}_{(\!|\mathsf{ne}|\!)} = \langle Q, q_0, tr, F\rangle$, we let

- $\|q\| \geq lth(C)$ for each $q \in Q$, especially, $\|q_0\| = lth(C)$ and $\|q\| = lth(C)$ for $q \in F$,
- the initial configuration is $\langle q_0, w, \mathbb{C}\rangle$ and final configurations $\langle q, \varepsilon, E'\rangle$ for some $q \in F$ and some extant chronicle E' ($lth(E') = lth(C)$),
- for each free name n in ne, there is a unique index $i \in \{1, \ldots, lth(C)\}$ with $C[\![i]\!] = n$.

5.1 From NRE to CDA$^\sharp$

We show the inductive construction of CDA$^\sharp$ for up-NREs; the construction is similar to the one in [11]. The inductive construction is informally depicted in Fig. 4 where the CDA$^\sharp$

$$\mathcal{H}_{(\!|\mathsf{ne}|\!)} = \langle Q, q_0, tr, F\rangle \qquad \text{and} \qquad \mathcal{H}_{(\!|\mathsf{ne}_h|\!)} = \langle Q_h, q_{(h,0)}, tr_h, F_h\rangle, \;\; h = 1,2 \qquad (3)$$

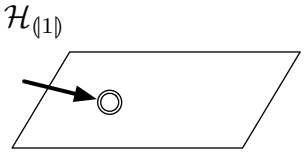
in the inductive steps are instances of the generic automaton of Fig. 3 (which, for readability, encompasses only two final states, but in general could have more) and respectively correspond to the following up-NREs in-contexts

$$C \ddagger \mathsf{ne} \ddagger E \qquad C \ddagger \mathsf{ne}_1 \ddagger E \qquad C \ddagger \mathsf{ne}_2 \ddagger E$$

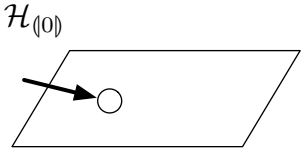
In the last case of Fig. 3 for the NRE $\langle_n \mathsf{ne}\rangle_n^m$, we let the contexts for ne be $C+[n]$ and $E+[m\natural t]$, i.e. $C+[n] \ \ddagger\ \mathcal{H}_{(\!|\mathsf{ne}|\!)} \ \ddagger\ E+[m\natural t]$. Note also that the automata in (3) used in the inductive cases may generate states at higher levels, see the last case of Fig. 4.

Notice that, for simplicity, we assume that NREs have no free names. Hence, n or $\underline{n}$ are local names which must be stored in a unique register i and pre-contexts C. In addition, post-contexts E do not coincide with the "real" post-contexts when each word is accepted. This is because, for example, if a NRE has a Kleene star, the real post-contexts may change from time to time depending on how many times we make loops to accept words.

Base cases. If $\mathsf{ne} = 1$, we let the corresponding CDA$^\sharp$ in-contexts $C \ \ddagger\ \mathcal{H}_{(\!|1|\!)} \ \ddagger\ E$ with $\mathcal{H}_{(\!|1|\!)} = \langle Q, q_0, tr, F\rangle$ where $Q = \{q_0\}$, $tr = \emptyset$ and $F = \{q_0\}$. Note that the number of registers in q_0 is determined by $lth(C)$, i.e. $\|q_0\| = lth(C)$.



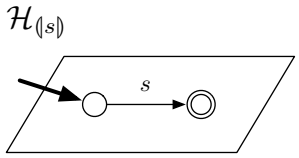
If $\mathsf{ne} = 0$, we let the corresponding CDA$^\sharp$ in-contexts $C \ \ddagger\ \mathcal{H}_{(\!|0|\!)} \ \ddagger\ E$ with $\mathcal{H}_{(\!|0|\!)} = \langle Q, q_0, tr, F\rangle$ where $Q = \{q_0\}$, $tr = \emptyset$ and $F = \emptyset$. Note that the number of registers in q_0 is determined by $lth(C)$.



If $\mathsf{ne} = s$, we let the corresponding CDA$^\sharp$ in-contexts $C \ \ddagger\ \mathcal{H}_{(\!|s|\!)} \ \ddagger\ E$ with $\mathcal{H}_{(\!|s|\!)} = \langle Q, q_0, tr, F\rangle$ where $Q = \{q_0, q_1\}$, $F = \{q_1\}$ and

$$\begin{cases} tr(q,\alpha) = \{q_1\} & \text{if } q = q_0 \text{ and } \alpha = s \\ tr(q,\alpha) = \emptyset & \text{otherwise} \end{cases}$$

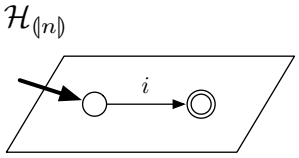
Note that the number of registers in q_0 and q_1 is the same as $lth(C)$, i.e. $\|q_0\| = \|q_1\| = lth(C)$.



If $\mathsf{ne} = n$, we let the corresponding CDA$^\sharp$ in-contexts $C \ddagger \mathcal{H}_{(\!|n|\!)} \ddagger E$ with $\mathcal{H}_{(\!|n|\!)} = \langle Q, q_0, tr, F\rangle$ where $Q = \{q_0, q_1\}$, $F = \{q_1\}$ and

$$\begin{cases} tr(q,\alpha) = \{q_1\} & \text{if } q = q_0 \text{ and } C[\![\alpha]\!] = n \\ tr(q,\alpha) = \emptyset & \text{otherwise} \end{cases}$$

Note that the number of registers in q_0 and q_1 is the same as $lth(C)$, i.e. $\|q_0\| = \|q_1\| = lth(C)$.

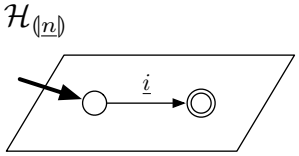


Remember that, since we are only considering closed up-NREs (no free name), this n must be local (stored in a unique register in q_0) and appears in the pre-context C. This unique register is identified by $C[\![\alpha]\!] = n$, and the above picture is assuming $C[\![i]\!] = n$.

If $\mathsf{ne} = \underline{n}$, we let the corresponding CDA$^\sharp$ in-contexts $C \ddagger \mathcal{H}_{(\!|\underline{n}|\!)} \ddagger E$ with $\mathcal{H}_{(\!|\underline{n}|\!)} = \langle Q, q_0, tr, F\rangle$ where $Q = \{q_0, q_1\}$, $F = \{q_1\}$ and

$$\begin{cases} tr(q,\alpha) = \{q_1\} & \text{if } q = q_0 \text{ and } C[\![\alpha]\!] = n \\ tr(q,\alpha) = \emptyset & \text{otherwise} \end{cases}$$

Note that the number of registers in q_0 and q_1 is the same as $lth(C)$, i.e. $\|q_0\| = \|q_1\| = lth(C)$.

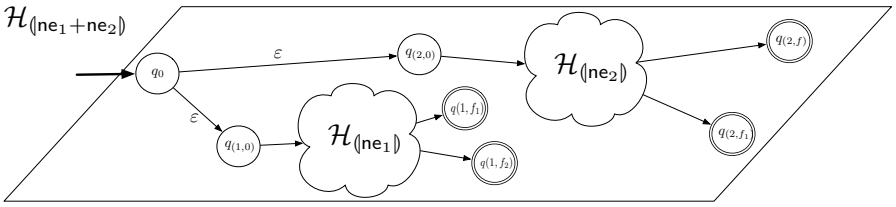


By our assumption, the underline $\underline{n}$ can be added only for local names. So, n is local (stored in a unique register in q_0) and appears in the pre-context C. The register number is identified $C[\![\alpha]\!] = n$ as above, and the picture is assuming $C[\![i]\!] = n$.

Inductive cases. For $\mathsf{ne}_1 + \mathsf{ne}_2$, the corresponding CDA$^\sharp$ in-contexts $C \ddagger \mathcal{H}_{(\!|\mathsf{ne}_1+\mathsf{ne}_2|\!)} \ddagger E$ is $\mathcal{H}_{(\!|\mathsf{ne}_1+\mathsf{ne}_2|\!)} = \langle Q^+, q_0, tr^+, F^+\rangle$ where

- q_0 is a new initial state with $\|q_0\| = lth(C)$
- $Q^+ = \{q_0\} \cup Q_1 \cup Q_2$
- $\begin{cases} tr^+(q,\alpha) = \{q_{(1,0)}, q_{(2,0)}\} & \text{if } q = q_0 \text{ and } \alpha = \varepsilon \\ tr^+(q,\alpha) = tr_1(q,\alpha) & \text{if } q \in Q_1 \\ tr^+(q,\alpha) = tr_2(q,\alpha) & \text{if } q \in Q_2 \end{cases}$
- $F^+ = F_1 \cup F_2$

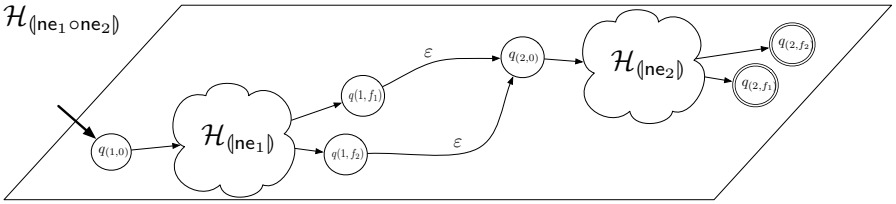
Notice that the previous initial states $q_{(1,0)}$ and $q_{(2,0)}$ have the same amount of registers as q_0, by the inductive hypothesis, i.e. $\left\|q_{(1,0)}\right\| = \left\|q_{(2,0)}\right\| = \|q_0\| = lth(C)$.



For $\mathsf{ne}_1 \circ \mathsf{ne}_2$, the corresponding CDA$^\sharp$ in-contexts $C \ddagger \mathcal{H}_{(\!|\mathsf{ne}_1 \circ \mathsf{ne}_2|\!)} \ddagger E$ is $\mathcal{H}_{(\!|\mathsf{ne}_1 \circ \mathsf{ne}_2|\!)} = \langle Q^\circ, q_{(1,0)}, tr^\circ, F_2 \rangle$ where

- $Q^\circ = Q_1 \cup Q_2$
- $\begin{cases} tr(q,\alpha) = tr_1(q,\alpha) \cup \{q_{(2,0)}\} & \text{if } q \in F_1 \text{ and } \alpha = \varepsilon \\ tr(q,\alpha) = tr_1(q,\alpha) & \text{if } q \in Q_1 \setminus F_1 \text{ or } \alpha \neq \varepsilon \\ tr(q,\alpha) = tr_2(q,\alpha) & \text{if } q \in Q_2 \end{cases}$

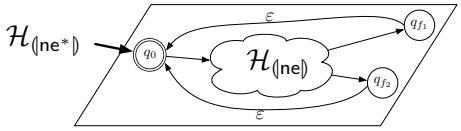
By our construction, we easily check that, in each automaton in-contexts, the initial state has the same number of registers as those of all final states. By the induction hypothesis, notice that the final states in $\mathcal{H}_{(\!|\mathsf{ne}_1|\!)}$ has the same amount of registers as that of the initial state $q_{(2,0)}$ in $\mathcal{H}_{(\!|\mathsf{ne}_2|\!)}$.



For ne^*, the corresponding CDA$^\sharp$ in-contexts $C \ddagger \mathcal{H}_{(\!|\mathsf{ne}^*|\!)} \ddagger E$ is $\mathcal{H}_{(\!|\mathsf{ne}^*|\!)} = \langle Q, q_0, tr^*, F^* \rangle$ where

- $\begin{cases} tr^*(q,\alpha) = tr(q,\alpha) \cup \{q_0\} & \text{if } q \in F \text{ and } \alpha = \varepsilon \\ tr^*(q,\alpha) = tr(q,\alpha) & \text{otherwise} \end{cases}$
- $F^* = \{q_0\}$

Notice that the initial state has the same number of registers as those of all the final states.

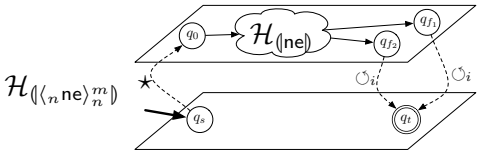


For $\langle_n \mathsf{ne} \rangle_n^m$, the corresponding CDA$^\sharp$ in-contexts $C \ddagger \mathcal{H}_{(\!|\langle_n \mathsf{ne} \rangle_n^m|\!)} \ddagger E$ is $\mathcal{H}_{(\!|\langle_n \mathsf{ne} \rangle_n^m|\!)} = \langle Q^\diamond, q_s, tr^\diamond, F^\diamond \rangle$ where

- q_s and q_t are new states with $\|q_s\| = \|q_t\| = lth(C)$ (remember $\mathcal{H}_{(\!|\mathsf{ne}|\!)}$ is in-contexts $C + [n]$ and $E + [m \sharp t]$)
- $Q^\diamond = \{q_s, q_t\} \cup Q$

- q_s is the initial state
- $\begin{cases} tr^{\Diamond}(q,\alpha) = \{q_0\} & \text{if } q = q_s \text{ and } \alpha = \star \\ tr^{\Diamond}(q,\alpha) = \emptyset & \text{if } (q = q_s \text{ and } \alpha \neq \star) \text{ or } q = q_t \\ tr^{\Diamond}(q,\alpha) = \{q_t\} & \text{if } q \in F \text{ and } \alpha = \circlearrowleft_i \text{ with } cv(E + [m\natural t])[\![i]\!] = n \\ tr^{\Diamond}(q,\alpha) = tr(q,\alpha) & \text{otherwise} \end{cases}$
- $F^{\Diamond} = \{q_t\}$

Remember that, for this case, we are assuming that $\mathcal{H}_{(\!|\mathsf{ne}|\!)}$ are in-contexts between $C+[n]$ and $E+[m\natural t]$. Namely, $\|q_s\| = \|q_t\| = lth(C)$ and $\|q_0\| = lth(C)+1$. Hence, q_s can take a $\star$-transition to q_0 and final states in $\mathcal{H}_{(\!|\mathsf{ne}|\!)}$ can take $\circlearrowleft_i$-transitions to q_t.



Moreover, notice that, by our assumption for the superscript names on closing brackets, m must be in a unique register in the pre-context C, which can be identified with $cv(E+[m\natural t])[\![i]\!] = n$ and $C[\![i]\!] = m$. Recall the rules ($\hat{\Diamond}_{=}$) and ($\hat{\Diamond}_{\neq}$) in Fig. 1.

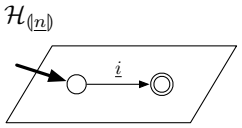
5.2 CDA$^\sharp$ accepts NREs

Proposition 2. *For up-NREs in-contexts $C \ddagger 1 \ddagger E$, $C \ddagger 0 \ddagger E$, $C \ddagger s \ddagger E$, $C \ddagger n \ddagger E$ and $C \ddagger \underline{n} \ddagger E$, the corresponding CDA$^\sharp$ in-contexts $C \ddagger \mathcal{H}_{(\!|1|\!)} \ddagger E$, $C \ddagger \mathcal{H}_{(\!|0|\!)} \ddagger E$, $C \ddagger \mathcal{H}_{(\!|s|\!)} \ddagger E$, $C \ddagger \mathcal{H}_{(\!|n|\!)} \ddagger E$ and $C \ddagger \mathcal{H}_{(\!|\underline{n}|\!)} \ddagger E$ accept the language in-contexts $C \ddagger \mathbf{L}(1) \ddagger E$, $C \ddagger \mathbf{L}(0) \ddagger E$, $C \ddagger \mathbf{L}(s) \ddagger E$, $C \ddagger \mathbf{L}(n) \ddagger E$ and $C \ddagger \mathbf{L}(\underline{n}) \ddagger E$, respectively.*

Proof. The only non-trivial case is $C \ddagger \underline{n} \ddagger E$. In this case, the language in-contexts is obtained by the language calculus as follows:

$$\left\{\star \in \mathcal{N} \;\mid\; \star \# C \text{ and } \star \,\underline{\#}^i\, \mathbb{C}[\![i]\!]\right\}$$

with the post context $((n\,\star)\cdot cv(E))\natural(E@\star)$, where i is the register number whose current value is n, i.e. $cv(C)[\![i]\!] = n$. The corresponding CDA$^\sharp$ is given as follows:



Hence, the initial configuration $\langle q_0, \star, \mathbb{C}\rangle$ can reach the final state q_1 if and only if $\star \# C$ and $\star \,\underline{\#}^i\, \mathbb{C}[\![i]\!]$. This is because $\star$ must be fresh for all the current names $\star \# C$ also for the chronicle i, i.e. $\mathbb{C}[\![i]\!]$. Notice that, later on, when we concatenate with other languages, we may change, by permuting names and appending chronicles, the local and global freshness conditions (recall rule ($\eth$) in Fig. 2 and see how it works in Appendix A). Also, the post-context must correspond to the post-context given by the language calculus, i.e. $((n\,\star)\cdot cv(E))\natural(E@\star)$, by the definition of the movement of CDA$^\sharp$. □

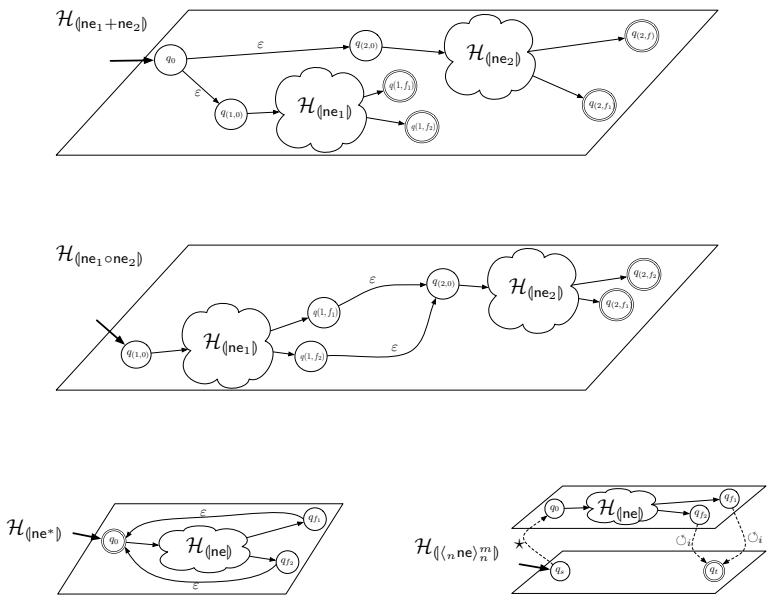


Fig. 4. The inductive constructions for $\mathcal{H}_{(\!|\mathsf{ne}_1+\mathsf{ne}_2|\!)}$, $\mathcal{H}_{(\!|\mathsf{ne}_1\circ\mathsf{ne}_2|\!)}$, $\mathcal{H}_{(\!|\mathsf{ne}^*|\!)}$ and $\mathcal{H}_{(\!|\langle_n \mathsf{ne}\rangle^m_n|\!)}$.

The construction from NREs to automata is summarised by the next two propositions.

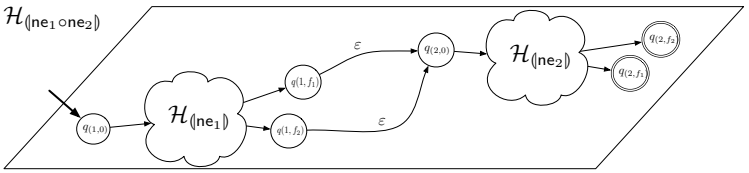
Proposition 3. *Given two NREs* ne_1 *and* ne_2*, a pre-context C, and a post-context E, the CDA*$^\sharp$ *in-contexts* $C \ddagger \mathcal{H}_{(\!|\mathsf{ne}_1+\mathsf{ne}_2|\!)} \ddagger E$ *recognises the language in-contexts* $C \ddagger \mathbf{L}(\mathsf{ne}_1+\mathsf{ne}_2) \ddagger E$ *while the CDA*$^\sharp$ *in-contexts* $C \ddagger \mathcal{H}_{(\!|\mathsf{ne}_1\circ\mathsf{ne}_2|\!)} \ddagger E$ *recognises the language in-contexts* $C \ddagger \mathbf{L}(\mathsf{ne}_1\circ\mathsf{ne}_2) \ddagger E$.

Proof. Because of the context calculus, for an NRE in-contexts $C \ddagger \mathsf{ne}_1\circ\mathsf{ne}_2 \ddagger E$, we assume, as the inductive hypothesis, the languages in-contexts $C \ddagger \mathbf{L}(\mathsf{ne}_1) \ddagger \mathbb{C}$ and $C \ddagger \mathbf{L}(\mathsf{ne}_2) \ddagger E$ obtained by $C \ddagger \mathsf{ne}_1 \ddagger \mathbb{C}$ and $C \ddagger \mathsf{ne}_2 \ddagger E$ are accepted by automata in-contexts $C \ddagger \mathcal{H}_{(\!|\mathsf{ne}_1|\!)} \ddagger \mathbb{C}$ and $C \ddagger \mathcal{H}_{(\!|\mathsf{ne}_2|\!)} \ddagger E$, respectively. Here we let the schematic words for $\mathbf{L}(\mathsf{ne}_1)$ and $\mathbf{L}(\mathsf{ne}_2)$ be $[\![\bullet^1_1 \dots \bullet^1_{k_1} \mid \phi_1]\!]$ and $[\![\bullet^2_1 \cdots \bullet^2_{k_2} \mid \phi_2]\!]$, respectively. Note that, the post-contexts in languages in-contexts are not necessarily reflecting the real extent chronicles in their final states. This is because it may have loops or unions in $\mathcal{H}_{(\!|\mathsf{ne}_1|\!)}$ and $\mathcal{H}_{(\!|\mathsf{ne}_2|\!)}$, then the extant chronicles may change depending on how many times each word makes loops until it is recognised, etc. However, when we consider each path (without unions and Kleene stars), i.e. schematic words in Fig. 2, each post-contexts reflects the 'real' extant chronicle in the final configuration in the corresponding CDA$^\sharp$: see Appendix A.

By the language calculus, we obtain the following schematic word for each pair of schematic words $\mathbf{L}_1$ and $\mathbf{L}_2$:

$$[\![\bullet^1_1 \cdots \bullet^1_{k_1} \circ \pi_{[C \rhd E_1]} \cdot \left(\bullet^2_1 \cdots \bullet^2_{k_2}\right) \mid \phi_1, \left(\pi_{[C \rhd E_1]} \cdot \phi_2\right)]\!]$$

Since ne_1 and ne_2 may contain $+$ or $\_^*$, the post-contexts $\mathbb{C}$ and E obtained in Fig. 1 may change to some other extant chronicles depending on which path we take or how many time we make loops etc during LNGC. Hence, we assume for the current schematic words in-contexts that they have E_1 and E_2 as their post-contexts. Notice that, the post-chronicles E_1 and E_2 reflect the extant chronicles in their final configurations. As the permutation action $\pi_{[C \rhd E_1]}$, by definition, permutes the current values to start $\mathcal{H}_{(\!|\mathsf{ne}_2|\!)}$ to the current values of E_1. Not only that, it appends the chronicles in E_1 to the initial configuration of the second CDA$^\sharp$. Accordingly, it updates the local freshness conditions and the relative global freshness conditions in ϕ_2 to the appropriate one: also see how it works in Appendix A. Hence, the construction of $C \ddagger \mathcal{H}_{(\!|\mathsf{ne}_1 \circ \mathsf{ne}_2|\!)} \ddagger E$ works. □

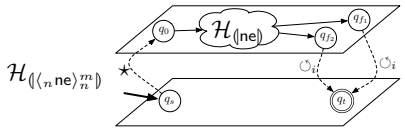


Proposition 4. *Given an NRE* ne*, a pre-context C, and a post-context E, the CDA*$^\sharp$ *in-contexts the CDA*$^\sharp$ *in-contexts* $C \ddagger \mathcal{H}_{(\!|\mathsf{ne}^*|\!)} \ddagger E$ *recognises the language in-contexts* $C \ddagger \mathbf{L}(\mathsf{ne}^*) \ddagger E$ *while the CDA*$^\sharp$ *in-contexts* $C \ddagger \mathcal{H}_{(\!|\langle_n \mathsf{ne} \rangle_n^m|\!)} \ddagger E$ *recognises the language in-contexts* $C \ddagger \mathbf{L}(\langle_n \mathsf{ne} \rangle_n^m) \ddagger E$.

Proof. Let a language in-contexts $C + [n] \ddagger \mathbf{L}(\mathsf{ne}) \ddagger E + [m \natural t]$ be recognised by the CDA$^\sharp$ in-contexts $C + [n] \ddagger \mathcal{H}_{(\!|\mathsf{ne}|\!)} \ddagger E + [m \natural t]$, with the schematic word for $\mathbf{L}(\mathsf{ne})$ being $[\![\bullet_1 \cdots \bullet_k \mid \phi]\!]$. By the rules of LNGC, a schematic word for $\mathbf{L}\langle_n \mathsf{ne} \rangle_n^m$ is

$$[\![(n \mathbin{\text{✡}}) \cdot \bullet_1 \cdots \bullet_k \mid \mathbin{\text{✡}} \# cv(C), (n \mathbin{\text{✡}}) \cdot \phi]\!] \tag{4}$$

The corresponding CDA$^\sharp$ in-contexts can store any name $\star$ locally fresh wrt $cv(C)$. By induction hypothesis, for each instance $(n \star) \cdot \bullet_1 \cdots \bullet_k$ of (4) such that $\star \# cv(C)$ and $(n \star) \circ \phi$ holds, it is the case that a state of $\mathcal{H}_{(\!|\langle_n \mathsf{ne} \rangle_n^m|\!)}$ corresponding to a final state of $\mathcal{H}_{(\!|\mathsf{ne}|\!)}$ is reached (now on the 1-st layer of $\mathcal{H}_{(\!|\langle_n \mathsf{ne} \rangle_n^m|\!)}$). To help the intuition, consider the following figure (where q_{f_1} and q_{f_2} are final states of $\mathcal{H}_{(\!|\mathsf{ne}|\!)}$).



Now, to remove an appropriate current value and the last chronicle from the final extant chronicle, we have to choose the corresponding $\circlearrowleft_i$ for some i. Thanks to the

rules of CTXC, we can choose i such that $m = C[\![i]\!]$ (recall rule $(\hat{\diamond}_{\neq})$ in Fig. 1 and note that the existence of i is guaranteed by our constrains: $\langle_n \mathsf{ne} \rangle_n^m$ must appear in a scope of m). Hence, the automaton stops with accounting of the corresponding permutations on $\rangle_n^m$.

The proof of the other cases is similar. □

5.3 Each CDA$^\sharp$ has an NRE

Th. 2 shows that each language accepted by a CDA$^\sharp$ can be described by an NRE.

Theorem 2. *Each language accepted by an CDA$^\sharp$, CA$^\sharp$, DA$^\sharp$ or A$^\sharp$ is nominal regular. That is, there exists an up-NRE, u-NRE, p-NRE or b-NRE which generates the same language.*

Proof. This is almost the same as the proof in [10]. The only difference is that we have $\underline{i}$ transitions. For those transitions, we just take the corresponding names with underlines.

Let $\mathcal{H}$ be a CDA$^\sharp$. Since each layer, if we ignore $\star$-transitions and $\circlearrowleft_i$-transitions, is a classical automaton. Hence, by the well known method (ε-closure and the powerset construction), we make each layer deterministic. For all $\star$-transitions and $\circlearrowleft_i$-transitions, we make another powerset construction to connect each layer as follows: for each state $Q_j = \left\{q_1^j, \ldots, q_k^j\right\}$ on the j-th layer, remember that each state is a subset of states because of the first powerset construction, we let

$$tr'(Q_j, \star) \stackrel{\text{def}}{=} \left\{q^{j+1} \mid \exists q^j \in Q_j.\ q^{j+1} \in tr(q^j, \star)\right\}$$
$$tr'(Q_j, \circlearrowleft_i) \stackrel{\text{def}}{=} \left\{q^{j-1} \mid \exists q^j \in Q_j.\ q^{j-1} \in tr(q^j, \circlearrowleft_i)\right\}$$

where tr and tr' are transitions after the first powerset construction and the second one, respectively. So the CDA$^\sharp$ is now deterministic.

For the obtained automaton, as in the case of the classical language theory, we calculate paths inductively. But, in our case, the inductive steps are also separated into two steps. Namely, the first step is on the highest layer of the automaton, which is almost the same as the classical method. The only difference is that in our automaton, names are labeled by natural numbers. After that, we make another induction on layers, see also [11]). Notice that, to bind names, we use a canonical naming, i.e. $[n_1, \ldots, n_h]$ (n_i is allocated to the label i). Hence the translation from accepted paths to expressions are straightforward.

Finally, the definition of subclasses of CDA$^\sharp$ tells their corresponding types of nominal regular expressions (CDA$^\sharp$, CA$^\sharp$, DA$^\sharp$ and A$^\sharp$ corresponding to up-NREs, u-NREs, p-NREs and NREs, respectively). □

Corollary 1. *Nominal regular languages are closed under union, concatenation and Kleene star.*

For the languages with explicit binders considered in [11] it is possible to define a notion of *resource-sensitive complementation* and prove that such languages are closed under resource-sensitive complementation. This is not possible when considering languages over infinite alphabets *without* explicit binders.

As a corollary of our theory, we describe how to define nominal regular expressions for fresh-register automata and register automata. Consider the following subclass of up-NREs (that we call *first-degree* up-NREs):

$$\mathsf{fne} ::= 1 \mid 0 \mid n_i \mid \underline{n_i} \mid s \mid \langle n_{h+1} n_{h+1} \rangle^{n_i}_{n_{h+1}} \mid \mathsf{fne} + \mathsf{fne} \mid \mathsf{fne} \circ \mathsf{fne} \mid \mathsf{fne}^*$$

where $n_1, \ldots, n_{h+1}$ are pairwise distinct names and $s \in \mathcal{S}$. Furthermore, an *h-prefixed first-degree* up-NREs is a first-degree up-NRE of the form $\langle n_1 \cdots \langle n_h \mathsf{fne} \rangle_{n_h} \cdots \rangle_{n_1}$ where fne is a binder-free first-degree up-NRE. Then we can prove the following result:

Theorem 3. *For every FRA (RA), there is an up-NRE (p-NRE) which generates the accepted language. More precisely, the up-NRE (p-NRE) is* h-prefixed first-degree. *Hence every FRA (RA) is expressible by an h-prefixed first-degree up-NRE (p-NRE)* $\langle n_1 \cdots \langle n_h \mathsf{fne} \rangle_{n_h} \cdots \rangle_{n_1}$.

6 Conclusion

We studied different types of automata and languages over infinite alphabets and gave Kleene type theorems characterising them by regular expressions. On the one hand, this extends the work on automata over infinite alphabets begun in [6], on the other hand the automata we propose are variations on the HD-automata of [12,13] (in particular, our transitions allocating fresh-names and permuting names are borrowed and adapted from HDA). As HDA are automata internal in the category of named sets, this also means, see [3], that our work can be seen in the context of nominal sets [2] and the more recent line of research on nominal automata [1].

Regular expressions for register automata were investigated in [7,8]. A difference is that the NREs of this paper have primitives for allocation and deallocation and permutations. Moreover, we also introduced NREs for relative global freshness.

The novel notion of relative global freshness is closely related to the recent [4,15]. Whereas we are interested in choreographies, [4] use register automata to monitor the execution of Java programs that generate a potentially unbounded number of names, albeit without using global freshness or histories. The history register automata (HRA) of [15] share with $\mathrm{CDA}^\sharp$ the ability to "forget" names since reset transitions can modify histories. We observe that [15] makes no attempt at finding a class of corresponding regular expressions. A detailed comparison as well as the definition of NREs for HRA have to be left as future work.

References

1. M. Bojanczyk, B. Klin, and S. Lasota. Automata with group actions. In *LICS'11*.
2. M. Gabbay and A. M. Pitts. A new approach to abstract syntax involving binders. In *LICS'99*.
3. F. Gadducci, M. Miculan, and U. Montanari. About permutation algebras, (pre)sheaves and named sets. *Higher-Order and Symbolic Computation*, 19(2-3), 2006.
4. R. Grigore, D. Distefano, R. L. Petersen, and N. Tzevelekos. Runtime verification based on register automata. In *TACAS'13*.

5. K. Honda, V. Vasconcelos, and M. Kubo. Language primitives and type discipline for structured communication-based programming. In *ESOP'13*.
6. M. Kaminski and N. Francez. Finite-memory automata. *Theoret. Comput. Sci.*, 134(2), 1994.
7. M. Kaminski and T. Tan. Regular expressions for languages over infinite alphabets. *Fundam. Inform.*, 69(3), 2006.
8. M. Kaminski and D. Zeitlin. Finite-memory automata with non-deterministic reassignment. *Int. J. Found. Comput. Sci.*, 21(5), 2010.
9. N. Kavantzas, D. Burdett, G. Ritzinger, T. Fletcher, and Y. Lafon. `http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217`. Working Draft 17 December 2004.
10. A. Kurz, T. Suzuki, and E. Tuosto. A characterisation of languages on infinite alphabets with nominal regular expressions. In *IFIP TCS'12*.
11. A. Kurz, T. Suzuki, and E. Tuosto. On nominal regular languages with binders. In *FoSSaCS'12*.
12. U. Montanari and M. Pistore. An introduction to history dependent automata. *Electr. Notes Theor. Comput. Sci.*, 10, 1997.
13. U. Montanari and M. Pistore. Structured coalgebras and minimal HD-automata for the *pi*-calculus. *Theor. Comput. Sci.*, 340(3), 2005.
14. N. Tzevelekos. Fresh-register automata. In *POPL'11*.
15. N. Tzevelekos and R. Grigore. History-register automata. In *FoSSaCS'13*.

A An example

As a more complex example, we consider the language and the automaton for the following up-NRE:

$$\langle_n n \langle_m m \langle_l l \rangle_l^m m \langle_l \underline{n} l \underline{m} \rangle_l \rangle_m \rangle_n$$

By CTXC in Fig. 1, we obtain the proof tree in Fig. 5. In the tree, it is not necessary that c is different from d. It is also not important ow to choose their names. The only thing we have to care is to keep pre-contexts *C pairwise distinct*. Note that, in Fig. 5, Fig. 6 and Fig. 7, capital alphabets A, B, C, D, E, F are added to elementary NREs in-contexts, and round-bracketed numbers (1) - (10) are added to point out NREs on each inductive step. Also, double-dashed lines are used for simplifications (in particular, to remove repeating names in chronicles).

For the derivation tree, by LNGC in Fig. 2, we compute the schematic word from the backward direction (i.e. from leaves to the root) as in Fig. 6 and Fig. 7. The schematic word we obtain is

$$[\![\hexagram\hexagram_5\hexagram_4\hexagram_4\hexagram_1\hexagram_3\hexagram_2 \mid \begin{pmatrix} \hexagram \neq \hexagram_1, \hexagram \neq \hexagram_3, \hexagram \neq \hexagram_4, \hexagram \neq \hexagram_5, \hexagram_1 \neq \hexagram_2, \\ \hexagram_1 \neq \hexagram_3, \hexagram_1 \neq \hexagram_4, \hexagram_1 \neq \hexagram_5, \hexagram_2 \neq \hexagram_3, \\ \hexagram_2 \neq \hexagram_4, \hexagram_2 \neq \hexagram_5, \hexagram_3 \neq \hexagram_4, \hexagram_4 \neq \hexagram_5 \end{pmatrix}]\!]$$

so the nominal regular language is

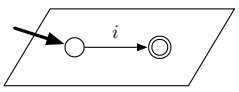
$$\{ abccdef \in \mathcal{N}^* \mid a \neq b, a \neq c, a \neq d, a \neq e, b \neq c, b \neq d, b \neq f, \\ c \neq d, c \neq e, c \neq f, d \neq e, d \neq f, e \neq f \}.$$

Notice that a and b can appear as f and e, respectively, in this language.

The languages in-contexts and automata in-contexts for base cases are considered as follows: Note that, as the example does not possess unions nor Kleene stars, we can denote the "real" extant chronicles in the post-contexts (hence we do so below, instead of showing the same post-contests as NREs in-contexts and automata in-contexts).

A: For the NRE in-contexts $[a] \ddagger a \ddagger [a\natural a]$, the language in-contexts and the CDA$^\sharp$ are

$$[a] \ddagger \{a\} \ddagger [a\natural a]$$



In this case, since the length of $[a]$ is 1, the automaton is on a first layer, hence each state has only one register. In the picture, i should be 1 and the initial assignment of the register is: $1 \mapsto a$ with the natural extant chronicle $[a]$. The run is: $\langle q_0, a, [a\natural a] \rangle \xrightarrow{1} \langle q_1, \varepsilon, [a\natural a] \rangle$. So, the CDA$^\sharp$ recognises the language in-contexts.

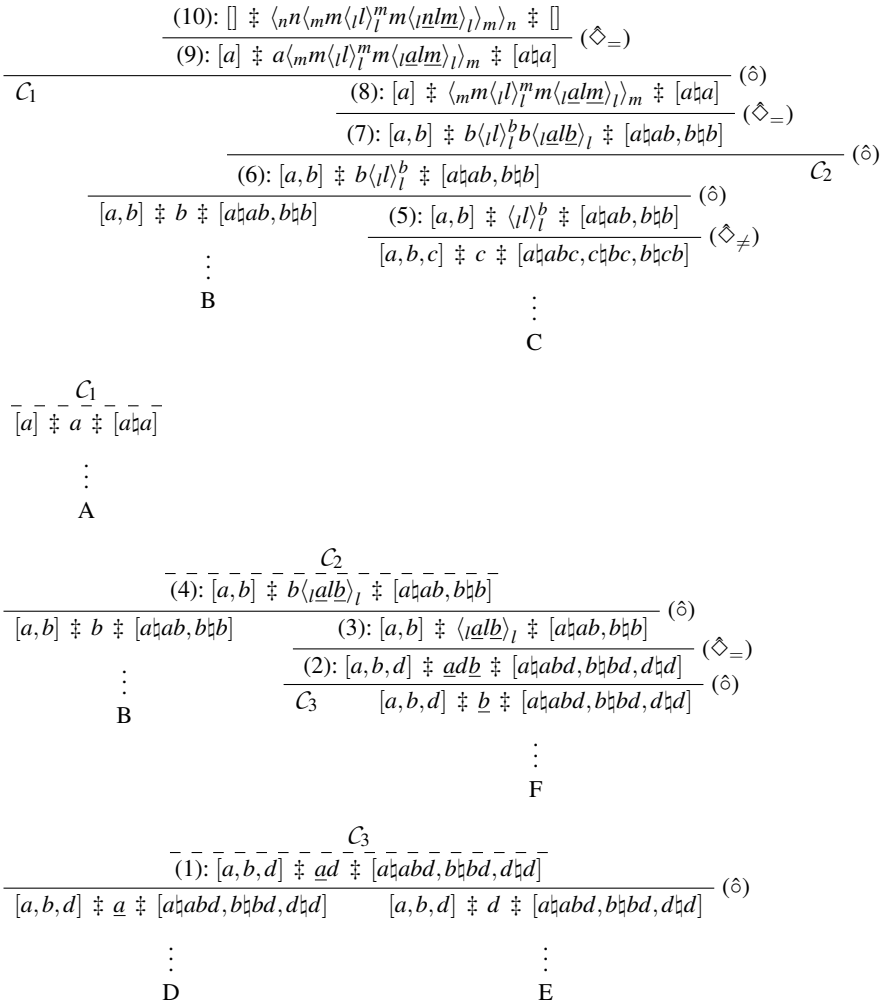


Fig. 5. CTXC for $\langle_n n\langle_m m\langle_l l\rangle_l^m m\langle_l n l m\rangle_l\rangle_m\rangle_n$

D

$$\begin{array}{c}\vdots\\ \dfrac{[a,b,d] \ddagger \underline{a} \ddagger [a\natural abd, b\natural bd, d\natural d]}{[a,b,d] \ddagger \{\davidsstar_1 \mid \davidsstar_1 \# abd, \davidsstar_1 \underline{\#}^1 abd\} \ddagger [\davidsstar_1 \natural abd\davidsstar_1, b\natural bd\davidsstar_1, d\natural d\davidsstar_1]}\,(\underline{a})\end{array}$$

E

$$\begin{array}{c}\vdots\\ \dfrac{[a,b,d] \ddagger d \ddagger [a\natural abd, b\natural bd, d\natural d]}{[a,b,d] \ddagger \{d \mid\} \ddagger [a\natural abd, b\natural bd, d\natural d]}\,(d)\end{array}$$

$$\dfrac{\cdots \qquad \cdots}{[a,b,d] \ddagger \{\davidsstar_1 d \mid \davidsstar_1 \# abd, \davidsstar_1 \underline{\#}^1 abd\} \ddagger [\davidsstar_1 \natural abd\davidsstar_1\davidsstar_1 bd, bd\davidsstar_1 bd\natural b, d\natural d\davidsstar_1 d]}\,(\check{\eth})$$

$$[a,b,d] \ddagger \{\davidsstar_1 d \mid \davidsstar_1 \# abd, \davidsstar_1 \underline{\#}^1 abd\} \ddagger [\davidsstar_1 \natural abd\davidsstar_1, b\natural bd\davidsstar_1, d\natural d\davidsstar_1]$$

$$\mathcal{L}_3$$

F

$$\begin{array}{c}\vdots\\ \dfrac{[a,b,d] \ddagger \underline{b} \ddagger [a\natural abd, b\natural bd, d\natural d]}{[a,b,d] \ddagger \{\davidsstar_2 \mid \davidsstar_2 \# abd, \davidsstar_2 \underline{\#}^2 bd\} \ddagger [a\natural abd\davidsstar_2, \davidsstar_2 \natural bd\davidsstar_2, d\natural d\davidsstar_2]}\,(\underline{b})\end{array}$$

$$\dfrac{\mathcal{L}_3 \qquad \cdots}{[a,b,d] \ddagger \{\davidsstar_1 d\davidsstar_2 \mid \davidsstar_1 \# abd, \davidsstar_1 \underline{\#}^1 abd, \davidsstar_2 \# \davidsstar_1 bd, \davidsstar_2 \underline{\#}^2 bd\davidsstar_1 bdbd\} \ddagger [\davidsstar_1 \natural abd\davidsstar_1\davidsstar_1 bd\davidsstar_2, \davidsstar_2 \natural bd\davidsstar_1 bd\davidsstar_2, d\natural d\davidsstar_1 d\davidsstar_2]}\,(\check{\eth})$$

$$\dfrac{[a,b,d] \ddagger \{\davidsstar_1 d\davidsstar_2 \mid \davidsstar_1 \# abd, \davidsstar_1 \underline{\#}^1 abd, \davidsstar_2 \# \davidsstar_1 bd, \davidsstar_2 \underline{\#}^2 bd\davidsstar_1\} \ddagger [\davidsstar_1 \natural abd\davidsstar_1\davidsstar_2, \davidsstar_2 \natural bd\davidsstar_1\davidsstar_2, d\natural d\davidsstar_1\davidsstar_2]}{[a,b] \ddagger \{\davidsstar_1\davidsstar_3\davidsstar_2 \mid \davidsstar_3 \# ab, \davidsstar_1 \# ab\davidsstar_3, \davidsstar_1 \underline{\#}^1 ab\davidsstar_3, \davidsstar_2 \# \davidsstar_1 b\davidsstar_3, \davidsstar_2 \underline{\#}^2 b\davidsstar_3\davidsstar_1\} \ddagger [\davidsstar_1 \natural ab\davidsstar_3\davidsstar_1\davidsstar_2, \davidsstar_2 \natural b\davidsstar_3\davidsstar_1\davidsstar_2]}\,(\check{\Diamond})$$

B

$$\begin{array}{c}\vdots\\ \dfrac{[a,b] \ddagger b \ddagger [a\natural ab, b\natural b]}{[a,b] \ddagger \{b \mid\} \ddagger [a\natural ab, b\natural b]}\,(b)\end{array}$$

$$\dfrac{\cdots \qquad \cdots}{[a,b] \ddagger \{b\davidsstar_1\davidsstar_3\davidsstar_2 \mid \davidsstar_3 \# ab, \davidsstar_1 \# ab\davidsstar_3, \davidsstar_1 \underline{\#}^1 abab\davidsstar_3, \davidsstar_2 \# \davidsstar_1 b\davidsstar_3, \davidsstar_2 \underline{\#}^2 bb\davidsstar_3\davidsstar_1\} \ddagger [\davidsstar_1 \natural abab\davidsstar_3\davidsstar_1\davidsstar_2, \davidsstar_2 \natural bb\davidsstar_3\davidsstar_1\davidsstar_2]}\,(\check{\eth})$$

$$[a,b] \ddagger \{b\davidsstar_1\davidsstar_3\davidsstar_2 \mid \davidsstar_3 \# ab, \davidsstar_1 \# ab\davidsstar_3, \davidsstar_1 \underline{\#}^1 ab\davidsstar_3, \davidsstar_2 \# \davidsstar_1 b\davidsstar_3, \davidsstar_2 \underline{\#}^2 b\davidsstar_3\davidsstar_1\} \ddagger [\davidsstar_1 \natural ab\davidsstar_3\davidsstar_1\davidsstar_2, \davidsstar_2 \natural b\davidsstar_3\davidsstar_1\davidsstar_2]$$

$$\mathcal{L}_2$$

A

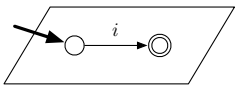
$$\begin{array}{c}\vdots\\ \dfrac{[a] \ddagger a \ddagger [a\natural a]}{[a] \ddagger \{a \mid\} \ddagger [a\natural a]}\,(a)\\ \mathcal{L}_1\end{array}$$

Fig. 6. First half of LNGC for $\langle_n n \langle_m m \langle_l l \rangle_l^m m \langle_l \underline{n} l \underline{m} \rangle_l \rangle_m \rangle_n$

Fig. 7. Second half of LNGC for $\langle_n n\langle_m m\langle_l l\rangle_l^m m\langle_l \underline{n} l \underline{m}\rangle_l\rangle_m\rangle_n$

B: For the NRE in-contexts $[a,b] \ddagger b \ddagger [a\natural ab, b\natural b]$, the language in-contexts and the CDA$^\sharp$ are

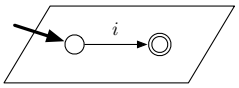
$$[a,b] \ddagger \{b\} \ddagger [a\natural ab, b\natural b]$$



In this case, since the length of $[a,b]$ is 2, the automaton is on a second layer, hence each state has two registers. In the picture, i should be 2 and the initial assignment of the registers is: $1 \mapsto a$ and $2 \mapsto b$ with the natural extant chronicle $[ab,b]$. The run is: $\langle q_0, b, [a\natural ab, b\natural b]\rangle \xrightarrow{2} \langle q_1, \varepsilon, [a\natural ab, b\natural b]\rangle$. So, the CDA$^\sharp$ recognises the language in-contexts.

C: For the NRE in-contexts $[a,b,c] \ddagger c \ddagger [a\natural abc, c\natural bc, b\natural cb]$, the language in-contexts and the CDA$^\sharp$ are

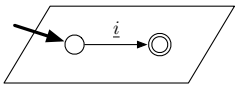
$$[a,b,c] \ddagger \{c\} \ddagger [a\natural abc, c\natural bc, b\natural cb]$$



In this case, since the length of $[a,b,c]$ is 3, the automaton is on a third layer, hence each state has three registers. In the picture, i should be 3 and the initial assignment of the registers is: $1 \mapsto a$, $2 \mapsto b$ and $3 \mapsto c$ with the natural extant chronicle $[abc,bc,c]$. The run is: $\langle q_0, c, [a\natural abc, b\natural bc, c\natural c]\rangle \xrightarrow{3} \langle q_1, \varepsilon, [a\natural abc, b\natural bc, c\natural c]\rangle$. So, the CDA$^\sharp$ recognises the language in-contexts.

D: For the NRE in-contexts $[a,b,d] \ddagger \underline{a} \ddagger [a\natural abd, b\natural bd, d\natural d]$, the language in-contexts and the CDA$^\sharp$ are

$$[a,b,d] \ddagger \left\{n \mid n\#abd, (n\underline{\#}^1 abd)\right\} \ddagger [n\natural abdn, b\natural bdn, d\natural dn]$$



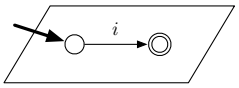
In this case, since the length of $[a,b,d]$ is 3, the automaton is on a third layer, hence each state has three registers. In the picture, i should be 1 and the initial assignment of the registers is: $1 \mapsto a$, $2 \mapsto b$ and $3 \mapsto d$ with the natural extant chronicle $[abd,bd,d]$. The run is:

$$\langle q_0, n, [a\natural abd, b\natural bd, d\natural d]\rangle \xrightarrow{\underline{1}} \langle q_1, \varepsilon, [n\natural abdn, b\natural bdn, d\natural dn]\rangle$$

where any name $n\#abd$. So, the CDA$^\sharp$ recognises the language in-contexts. One may feel that there seems no difference between $\#$ and $\underline{\#}^1$. However, the difference appear when we concatenate it with other languages: recall rule ($\eth$) in Fig. 2. That is, the natural chronicle is used as a bookmark for the later use here, for permutations and concatenations.

E: For the NRE in-contexts $[a,b,d] \ddagger d \ddagger [a\natural abd, bd\natural b, d\natural d]$, the language in-contexts and the CDA$^\sharp$ are

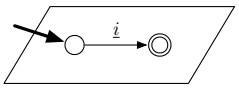
$$[a,b,d] \ddagger \{d\} \ddagger [a\natural abd, b\natural bd, d\natural d]$$



In this case, since the length of $[a,b,d]$ is 3, the automaton is on a third layer, hence each state has three registers. In this picture, i should be 3 and the initial assignment of the registers is: $1 \mapsto a$, $2 \mapsto b$ and $3 \mapsto d$ with the natural extant chronicle $[abd, bd, d]$. The run is: $\langle q_0, d, [a\natural abd, b\natural bd, d\natural d]\rangle \xrightarrow{3} \langle q_1, \varepsilon, [a\natural abd, b\natural bd, d\natural d]\rangle$. So, the CDA$^\sharp$ recognises the language in-contexts.

F: For the NRE in-contexts $[a,b,d] \ddagger \underline{b} \ddagger [a\natural abd, b\natural bd, d\natural d]$, the language in-contexts and the CDA$^\sharp$ are

$$[a,b,d] \ddagger \left\{n \mid n\#abd, (n\underline{\#}^2 bd)\right\} \ddagger [a\natural abdn, n\natural bdn, d\natural dn]$$

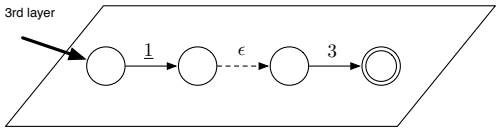


In this case, since the length of $[a,b,d]$ is 3, the automaton is on a third layer, hence each state has three registers. In this picture, i should be 2 and the initial assignment of the registers is: $1 \mapsto a$, $2 \mapsto b$ and $3 \mapsto d$ with the natural extant chronicle $[abd, bd, d]$. The run is: $\langle q_0, n, [a\natural abd, b\natural bd, d\natural d]\rangle \xrightarrow{\underline{2}} \langle q_1, \varepsilon, [a\natural abdn, n\natural bdn, d\natural dn]\rangle$, where any name $n\#abd$. Therefore, the CDA$^\sharp$ recognises the language in-contexts. As $\underline{b}$ is a relative global fresh transition with respect to the second chronicle, we take $\star_2 \underline{\#}^2 bd$ by means of the natural chronicle for the register 2 as a bookmark.

The languages in-contexts and the automata in-contests for inductive steps are as follows (note that we simplify chronicles or remove some ε-transitions):

(1) For the NRE $[a,b,d] \ddagger \underline{a}d \ddagger [a\natural abd, b\natural bd, d\natural d]$, the language in-contexts and the CDA$^\sharp$ are

$$C \ddagger \mathbf{L}(\mathsf{ne}) \ddagger E = [a,b,d] \ddagger \left\{nd \mid n\#abd, (n\underline{\#}^1 abd)\right\} \ddagger [n\natural abdn, b\natural bdn, d\natural dn]$$



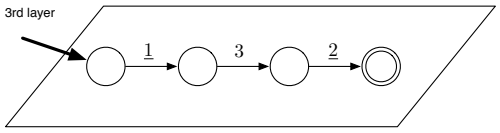
In the language, we have simplified chronicles. This step is a concatenation of cases D and E on a third layer. The initial assignment of the registers is: $1 \mapsto a$, $2 \mapsto b$ and $3 \mapsto d$ with the natural extant chronicle $[abd, bd, d]$. The run is:

$$\begin{aligned}\langle q_0, nd, [a\natural abd, b\natural bd, d\natural d]\rangle &\xrightarrow{\underline{1}} \langle q_1, d, [n\natural abdn, b\natural bdn, d\natural dn]\rangle \\ &\xrightarrow{3} \langle q_2, \varepsilon, [n\natural abdn, b\natural bdn, d\natural dn]\rangle\end{aligned}$$

where any name $n\#abd$. So, it is easy to see that the CDA$^\sharp$ accepts the language in-contexts.

(2) For the NRE $[a,b,d] \ddagger \underline{a}d\underline{b} \ddagger [a\natural abd, b\natural bd, d\natural d]$, the language in-contexts and the CDA$^\sharp$ are

$$\begin{aligned} C &: [a,b,d] \\ \mathbf{L}(\mathsf{ne}) &: \left\{ ndm \mid n\#abd, (n\underline{\#}^1 abd), m\#nbd, (m\underline{\#}^2 bdn) \right\} \\ E &: [n\natural abdnm, m\natural bdnm, d\natural dnm] \end{aligned}$$



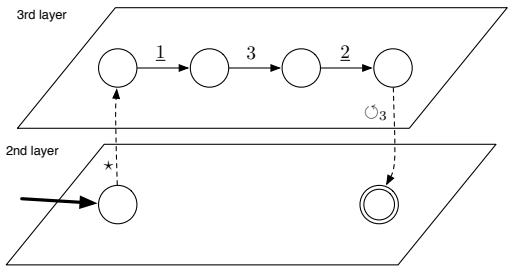
This step is the concatenation of cases (1) and F on a third layer. The initial assignment of the registers is: $1 \mapsto a$, $2 \mapsto b$ and $3 \mapsto d$ with the natural extant chronicle $[abd, bd, d]$. The run is:

$$\begin{aligned} \langle q_0, ndm, [a\natural abd, b\natural bd, d\natural d] \rangle &\xrightarrow{\underline{1}} \langle q_1, dm, [n\natural abdn, b\natural bdn, d\natural dn] \rangle \\ &\xrightarrow{3} \langle q_2, m, [n\natural abdn, b\natural bdn, d\natural dn] \rangle \\ &\xrightarrow{\underline{2}} \langle q_3, \varepsilon, [n\natural abdnm, m\natural bdnm, d\natural dnm] \rangle \end{aligned}$$

where any names $n\#abd$ and $m\#bdn$. Hence, the CDA$^\sharp$ accepts the language in-contexts. Notice that, when we concatenate the languages, the latter words are permute a with $\hexagram_1$ and updated chronicles (and relative-global freshness). One may find that m can be a, because of $m\#bdn$. The fact reflects the relative global freshness (with respect to the chronicle 2).

(3) For the NRE $[a,b] \ddagger \langle_l \underline{a} l \underline{b} \rangle_l \ddagger [a\natural ab, b\natural b]$, the language in-contexts and the CDA$^\sharp$ are

$$\begin{aligned} C &: [a,b] \\ \mathbf{L}(\mathsf{ne}) &: \left\{ nlm \mid l\#ab, n\#abl, (n\underline{\#}^1 abl), m\#nbl, (m\underline{\#}^2 bln) \right\} \\ E &: [n\natural ablnm, m\natural blnm] \end{aligned}$$



This step abstracts case (2). The initial assignment of the registers is: $1 \mapsto a$ and

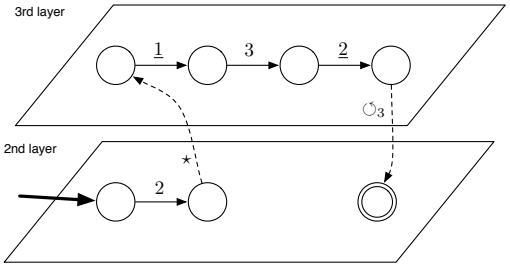
$2 \mapsto b$ with the natural extant chronicle $[ab, b]$. The run is:

$$\begin{aligned}\langle q_0, nlm, [a\natural ab, b\natural b]\rangle &\xrightarrow{\star} \langle q_1, nlm, [a\natural abl, b\natural bl, l\natural l]\rangle \\ &\xrightarrow{\underline{1}} \langle q_2, lm, [n\natural abln, b\natural bln, l\natural ln]\rangle \\ &\xrightarrow{3} \langle q_3, m, [n\natural abln, b\natural bln, l\natural ln]\rangle \\ &\xrightarrow{\underline{2}} \langle q_4, \varepsilon, [n\natural ablnm, m\natural blnm, l\natural lnm]\rangle \\ &\xrightarrow{\circlearrowleft 3} \langle q_5, \varepsilon, [n\natural ablnm, m\natural blnm]\rangle\end{aligned}$$

where any names $l\#ab$, $n\#abl$ and $m\#bln$. Hence, the CDA$^\sharp$ accepts the language in-contexts.

(4) For the NRE $[a,b] \ddagger b\langle_l \underline{a} l \underline{b}\rangle_l \ddagger [a\natural ab, b\natural b]$, the language in-contexts and the CDA$^\sharp$ are

$$\begin{aligned}C &: [a,b] \\ \mathbf{L}(\mathsf{ne}) &: \left\{bnlm \mid l\#ab, n\#abl, (n\underline{\#}^1 abl), m\#nbl, (m\underline{\#}^2 bln)\right\} \\ E &: [n\natural ablnm, m\natural blnm]\end{aligned}$$



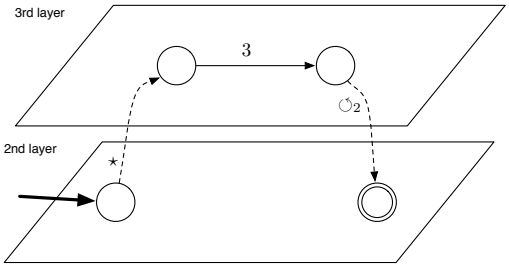
This step concatenates case B with case (3). The initial assignment of the registers is: $1 \mapsto a$ and $2 \mapsto b$ with the natural extant chronicle $[ab, b]$. The run is:

$$\begin{aligned}\langle q_0, bnlm, [a\natural ab, b\natural b]\rangle &\xrightarrow{\underline{2}} \langle q_1, nlm, [a\natural ab, b\natural b]\rangle \\ &\xrightarrow{\star} \langle q_2, nlm, [a\natural abl, b\natural bl, l\natural l]\rangle \\ &\xrightarrow{\underline{1}} \langle q_3, lm, [n\natural abln, b\natural bln, l\natural ln]\rangle \\ &\xrightarrow{3} \langle q_4, m, [n\natural abln, b\natural bln, l\natural ln]\rangle \\ &\xrightarrow{\underline{2}} \langle q_5, \varepsilon, [n\natural ablnm, m\natural blnm, l\natural lnm]\rangle \\ &\xrightarrow{\circlearrowleft_3} \langle q_6, \varepsilon, [n\natural ablnm, m\natural blnm]\rangle\end{aligned}$$

where any names $l\#ab$, $n\#abl$ and $m\#bln$. Hence, the CDA$^\sharp$ accepts the language in-contexts.

(5) For the NRE $[a,b] \ddagger \langle_l l\rangle_l^b \ddagger [a\natural ab, b\natural b]$, the language in-contexts and the CDA$^\sharp$ are

$$C \ddagger \mathbf{L}(\mathsf{ne}) \ddagger E = [a,b] \ddagger \{n \mid n\#ab\} \ddagger [a\natural abn, n\natural bn]$$



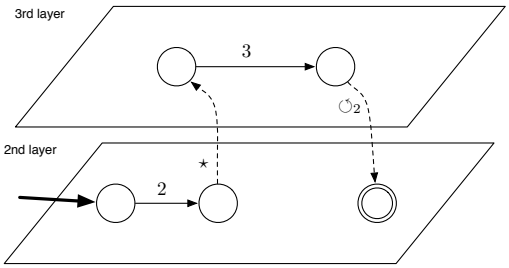
This step abstracts case C. Since the post extant chronicle of case C remembers the permutation, the transition to the final state labelled with $\circlearrowleft_2$ not with $\circlearrowleft_3$. Notice that the permutation effect is left in the post extant chronicle, see the current value of the register 2. The initial assignment of the registers is: $1 \mapsto a$ and $2 \mapsto b$ with the natural extant chronicle $[ab, b]$. The run is:

$$\begin{aligned}\langle q_0, n, [a\natural ab,]\natural a\rangle &\xrightarrow{\star} \langle q_1, n, a\natural abn, b\natural bn, n\natural n\rangle\\ &\xrightarrow{3} \langle q_2, \varepsilon, [a\natural abn, b\natural bn, n\natural n]\rangle\\ &\xrightarrow{\circlearrowleft_2} \langle q_3, \varepsilon, [a\natural abn, n\natural bn]\rangle\end{aligned}$$

where any name $n\#ab$. Therefore, the CDA$^\sharp$ accepts the language in-contexts with keeping the permutation action on the post-context. But, in the final state, the configuration of the registers turns to be on the level of a schematic word: $1 \mapsto a$ and $2 \mapsto \hexagram_4$ (not b) with the extant chronicle $[a\natural ab\hexagram_4, \hexagram_4\natural b\hexagram_4]$.

(6) For the NRE $[a,b] \ddagger b\langle_l l\rangle_l^b \ddagger [a\natural ab, b\natural b]$, the language in-contexts and the CDA$^\sharp$ are

$$C \ddagger \mathbf{L}(\mathsf{ne}) \ddagger E = [a,b] \ddagger \{bn \mid n\#ab\} \ddagger [a\natural abn, n\natural bn]$$



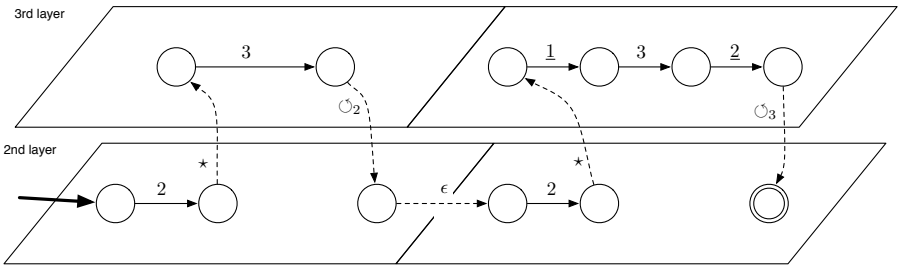
This step concatenates case B with case (5). The initial assignment of the registers is: $1 \mapsto a$ and $2 \mapsto b$ with the natural extant chronicle $[ab, b]$. The run is:

$$\begin{aligned}\langle q_0, bn, [a\natural ab, b\natural b]\rangle &\xrightarrow{2} \langle q_1, n, [a\natural ab, b\natural b]\rangle\\ &\xrightarrow{\star} \langle q_2, n, [a\natural abn, b\natural bn, n\natural n]\rangle\\ &\xrightarrow{3} \langle q_3, \varepsilon, [a\natural abn, b\natural bn, n\natural n]\rangle\\ &\xrightarrow{\circlearrowleft_2} \langle q_4, \varepsilon, [a\natural abn, n\natural bn]\rangle\end{aligned}$$

Hence, the CDA$^\sharp$ accepts the language in-contexts. Note that the permutation action in $\circlearrowleft_2$ is still preserved in safe in the post-context as we expect, i.e. the extant chronicle is $[a\natural ab\text{✡}_4, \text{✡}_4\natural b\text{✡}_4]$ on the level of a schematic word.

(7) For the NRE $[a,b] \ddagger b\langle_l l\rangle_l^b b\langle_l \underline{alb}\rangle_l \ddagger$, the language in-contexts and the CDA$^\sharp$ are

$$\begin{aligned} C &: [a,b] \\ \mathbf{L}(\mathsf{ne}) &: \left\{ bnnn'lm \mid n\#ab, l\#an, n'\#anl, n'\underline{\#}^1 abnl, m\#nn'l, m\underline{\#}^2 bnln' \right\} \\ E &: [n'\natural abnln'm, m\natural bnln'm] \end{aligned}$$



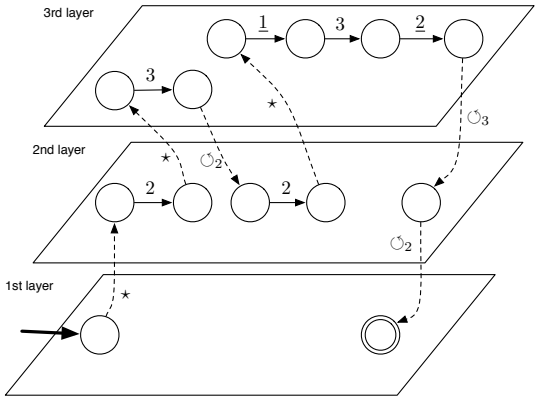
This step concatenates cases (4) and (6). The initial assignment of the registers is: $1 \mapsto a$ and $2 \mapsto b$ with the natural extant chronicle $[ab,b]$. The run is:

$$\begin{aligned} \langle q_0, bnnn'lm, [a\natural ab, b\natural b]\rangle &\xrightarrow{2} \langle q_1, nnn'lm, [a\natural ab, b\natural b]\rangle \\ &\xrightarrow{\star} \langle q_2, nnn'lm, [a\natural abn, b\natural bn, n\natural n]\rangle \\ &\xrightarrow{3} \langle q_3, nn'lm, [a\natural abn, b\natural bn, n\natural n]\rangle \\ &\xrightarrow{\circlearrowleft_2} \langle q_4, nn'lm, [a\natural abn, n\natural bn]\rangle \\ &\xrightarrow{2} \langle q_5, n'lm, [a\natural abn, n\natural bn]\rangle \\ &\xrightarrow{\star} \langle q_6, n'lm, [a\natural abnl, n\natural bnl, l\natural l]\rangle \\ &\xrightarrow{1} \langle q_7, lm, [n'\natural abnln', n\natural bnln', l\natural ln']\rangle \\ &\xrightarrow{3} \langle q_8, m, [n'\natural abnln', n\natural bnln', l\natural ln']\rangle \\ &\xrightarrow{2} \langle q_9, \varepsilon, [n'\natural abnln'm, m\natural bnln'm, l\natural ln'm]\rangle \\ &\xrightarrow{\circlearrowleft_3} \langle q_{10}, \varepsilon, [n'\natural abnln'm, m\natural bnln'm]\rangle \end{aligned}$$

where any names $n\#ab$, $l\#an$, $n'\#abnl$ and $m\#bnln'$. The important things are: 1. when we take the ε-transition, the name of the register 2 is n (✡_4 on the level of a schematic word). However, the concatenation of languages rule ($\check{\circ}$) reflects the fact as the permutation $\pi_{[C\rhd E_1]}$ and the relative global freshness by appending corresponding chronicles in E_1. Therefore, the CDA$^\sharp$ accepts the language in-contexts without any problem. Also, all the information are kept in the post-context in safe again.

(8) For the NRE $[a] \ddagger \langle_m m\langle_l l\rangle_l^m m\langle_l \underline{a} l \underline{m}\rangle_l\rangle_m \ddagger [a\natural a]$, the language in-contexts and the CDA$^\sharp$ are

$$
\begin{aligned}
C &: [a] \\
\mathbf{L}(\mathsf{ne}) &: \left\{ mnnn'lm' \mid m\#a, n\#am, l\#an, n'\#anl, n'\underline{\#}^1 amnl, m'\#n'nl, m'\underline{\#}^2 mnln' \right\} \\
E &: [n'\natural amnln'm']
\end{aligned}
$$



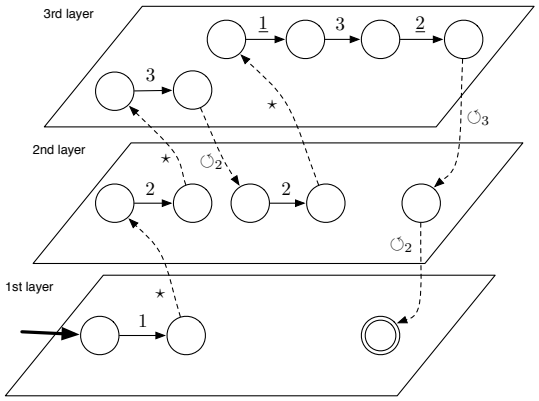
This step abstracts the second register of case (7). The initial assignment of the register is: $1 \mapsto a$ with the natural extant chronicle $[a]$. The run is:

$$
\begin{aligned}
\langle q_0, mnnn'lm', [a\natural a]\rangle &\xrightarrow{\star} \langle q_1, mnnn'lm', [a\natural am, m\natural m]\rangle \\
&\xrightarrow{2} \langle q_2, nnn'lm', [a\natural am, m\natural m]\rangle \\
&\xrightarrow{\star} \langle q_3, nnn'lm', [a\natural amn, m\natural mn, n\natural n]\rangle \\
&\xrightarrow{3} \langle q_4, nn'lm', [a\natural amn, m\natural mn, n\natural n]\rangle \\
&\xrightarrow{\circlearrowleft_2} \langle q_5, nn'lm', [a\natural amn, n\natural mn]\rangle \\
&\xrightarrow{2} \langle q_6, n'lm', [a\natural amn, n\natural mn]\rangle \\
&\xrightarrow{\star} \langle q_7, n'lm', [a\natural amnl, n\natural mnl, l\natural l]\rangle \\
&\xrightarrow{1} \langle q_8, lm', [n'\natural amnln', n\natural mnln', l\natural ln']\rangle \\
&\xrightarrow{3} \langle q_9, m', [n'\natural amnln', n\natural mnln', l\natural ln']\rangle \\
&\xrightarrow{2} \langle q_{10}, \varepsilon, [n'\natural amnln'm', m'\natural mnln'm', l\natural ln'm']\rangle \\
&\xrightarrow{\circlearrowleft_3} \langle q_{11}, \varepsilon, [n'\natural amnln'm', m'\natural mnln'm']\rangle \\
&\xrightarrow{\circlearrowleft_2} \langle q_{12}, \varepsilon, [n'\natural amnln'm']\rangle
\end{aligned}
$$

where any names $m\#a$, $n\#am$, $l\#an$, $n'\#amnl$ and $m'\#mnln'$. So, the abstracted m ($\hexagram_5$ on a schematic word) can be any name except a. Accordingly, we replace b by $\hexagram_5$. Hence, the CDA$^\sharp$ accepts the language in-contexts.

(9) For the NRE $[a] \ddagger a\langle_m m\langle_l l\rangle_l^m m\langle \underline{alm}\rangle_{\underline{alm}}\rangle_m \ddagger [a\natural a]$, the language in-contexts and the CDA$^\sharp$ are

$$C : [a]$$
$$\mathbf{L}(\mathsf{ne}) : \left\{ amnnn'lm' \mid m\#a, n\#am, l\#an, n'\#anl, n'\underline{\#}^1 amnl, m'\#n'nl, m'\underline{\#}^2 mnln' \right\}$$
$$E : [n'\natural amnln'm']$$



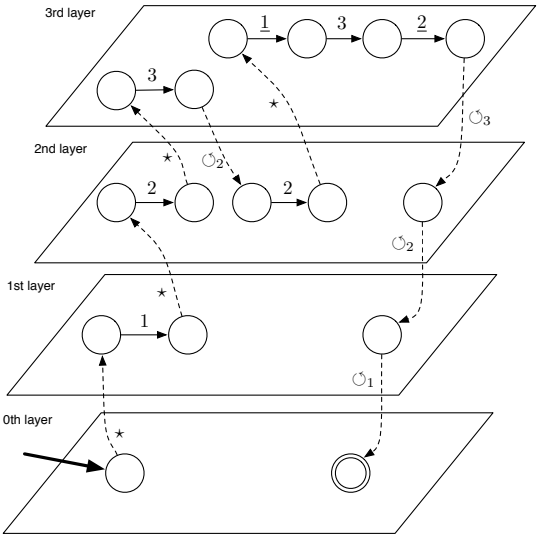
This step concatenates case A with case (8). The initial assignment of the register is: $1 \mapsto a$ with the natural extant chronicle $[a]$. The run is:

$$\begin{aligned}
\langle q_0, amnnn'lm', [a\natural a]\rangle &\xrightarrow{1} \langle q_1, mnnn'lm', [a\natural a]\rangle \\
&\xrightarrow{\star} \langle q_2, mnnn'lm', [a\natural am, m\natural m]\rangle \\
&\xrightarrow{2} \langle q_3, nnn'lm', [a\natural am, m\natural m]\rangle \\
&\xrightarrow{\star} \langle q_4, nnn'lm', [a\natural amn, m\natural mn, n\natural n]\rangle \\
&\xrightarrow{3} \langle q_5, nn'lm', [a\natural amn, m\natural mn, n\natural n]\rangle \\
&\xrightarrow{\circlearrowleft_2} \langle q_6, nn'lm', [a\natural amn, n\natural mn]\rangle \\
&\xrightarrow{2} \langle q_7, n'lm', [a\natural amn, n\natural mn]\rangle \\
&\xrightarrow{\star} \langle q_8, n'lm', [a\natural amnl, n\natural mnl, l\natural l]\rangle \\
&\xrightarrow{1} \langle q_9, lm', [n'\natural amnln', n\natural mnln', l\natural ln']\rangle \\
&\xrightarrow{3} \langle q_{10}, m', [n'\natural amnln', n\natural mnln', l\natural ln']\rangle \\
&\xrightarrow{2} \langle q_{11}, \varepsilon, [n'\natural amnln'm', m'\natural mnln'm', l\natural ln'm']\rangle \\
&\xrightarrow{\circlearrowleft_3} \langle q_{12}, \varepsilon, [n'\natural amnln'm', m'\natural mnln'm']\rangle \\
&\xrightarrow{\circlearrowleft_2} \langle q_{13}, \varepsilon, [n'\natural amnln'm']\rangle
\end{aligned}$$

where any names $m\#a$, $n\#am$, $l\#an$, $n'\#amnl$ and $m'\#mnln'$. Hence, the CDA$^\sharp$ accepts the language in-contexts.

(10) For the NRE $[] \ddagger \langle_n n \langle_m m \langle_l l \rangle_l^m m \langle_l \underline{n} l \underline{m} \rangle_l \rangle_m \rangle_n \ddagger []$, the language in-contexts and the CDA$^\sharp$ are

$$
\begin{aligned}
C &: [] \\
\mathbf{L}(\mathsf{ne}) &: \left\{ lmnnn'l'm' \mid m\#l, n\#lm, l'\#ln, n'\#lnl', n'\underline{\#}^1 lmnl', m'\#n'nl', m'\underline{\#}^2 mnl'n' \right\} \\
E &: []
\end{aligned}
$$



This step abstracts the first register in case (9). The initial assignment is empty. The

run is:

$$\begin{aligned}
\langle q_0, lmnnn'l'm', [] \rangle &\xrightarrow{\star} \langle q_1, lmnnn'l'm', [l \natural l] \rangle \\
&\xrightarrow{1} \langle q_2, mnnn'l'm', [l \natural l] \rangle \\
&\xrightarrow{\star} \langle q_3, mnnn'l'm', [l \natural lm, m \natural m] \rangle \\
&\xrightarrow{2} \langle q_4, nnn'l'm', [l \natural lm, m \natural m] \rangle \\
&\xrightarrow{\star} \langle q_5, nnn'l'm', [l \natural lmn, m \natural mn, n \natural n] \rangle \\
&\xrightarrow{3} \langle q_6, nn'l'm', [l \natural lmn, m \natural mn, n \natural n] \rangle \\
&\xrightarrow{\circlearrowleft_2} \langle q_7, nn'l'm', [l \natural lmn, n \natural mn] \rangle \\
&\xrightarrow{2} \langle q_8, n'l'm', [l \natural lmn, n \natural mn] \rangle \\
&\xrightarrow{\star} \langle q_9, n'l'm', [l \natural lmnl', n \natural mnl', l' \natural l'] \rangle \\
&\xrightarrow{1} \langle q_{10}, l'm', [n' \natural lmnl'n', n \natural mnl'n', l' \natural l'n'] \rangle \\
&\xrightarrow{3} \langle q_{11}, m', [n' \natural lmnl'n', n \natural mnl'n', l' \natural l'n'] \rangle \\
&\xrightarrow{2} \langle q_{12}, \varepsilon, [n' \natural lmnl'n'm', m' \natural mnl'n'm', l' \natural l'n'm'] \rangle \\
&\xrightarrow{\circlearrowleft_3} \langle q_{13}, \varepsilon, [n' \natural lmnl'n'm', m' \natural mnl'n'm'] \rangle \\
&\xrightarrow{\circlearrowleft_2} \langle q_{14}, \varepsilon, [n' \natural lmnl'n'm'] \rangle \\
&\xrightarrow{\circlearrowleft_1} \langle q_{15}, \varepsilon, [] \rangle
\end{aligned}$$

where any names l, $m\#l$, $n\#lm$, $l'\#ln$, $n'\#lmnl'$ and $m'\#mnl'n'$. Therefore, the CDA$^\sharp$ accepts the language for the NRE $\langle_n n \langle_m m \langle_l l \rangle_l^m m \langle_l \underline{nl} \underline{m} \rangle_l \rangle_m \rangle_n$.