

A Quasiorder-Based Perspective on Residual Automata

Pierre Ganty 

IMDEA Software Institute, Madrid, Spain
pierre.ganty@imdea.org

Elena Gutiérrez 

IMDEA Software Institute, Madrid, Spain
Universidad Politécnica de Madrid, Spain
elena.gutierrez@imdea.org

Pedro Valero 

IMDEA Software Institute, Madrid, Spain
Universidad Politécnica de Madrid, Spain
pedro.valero.mejia@gmail.com

Abstract

In this work, we define a framework of automata constructions based on quasiorders over words to provide new insights on the class of residual automata. We present a new residualization operation and a generalized double-reversal method for building the canonical residual automaton for a given language. Finally, we use our framework to offer a quasiorder-based perspective on NL^* , an online learning algorithm for residual automata. We conclude that quasiorders are fundamental to residual automata as congruences are to deterministic automata.

2012 ACM Subject Classification Theory of computation \rightarrow Formal languages and automata theory; Theory of computation \rightarrow Regular languages

Keywords and phrases Residual Automata, Quasiorders, Double-Reversal Method, Canonical RFA, Regular Languages

Digital Object Identifier 10.4230/LIPIcs.MFCS.2020.40

Related Version An extended version is available at <http://arxiv.org/abs/2007.00359>.

Funding All authors were partially supported by the Spanish project PGC2018-102210-B-I00 BOSCO
Pierre Ganty: Partially supported by the Madrid regional project S2018/TCS-4339 BLOQUES and the Ramón y Cajal fellowship RYC-2016-20281.

Elena Gutiérrez: Partially supported by BES-2016-077136 grant from the Spanish Ministry of Economy, Industry and Competitiveness.

1 Introduction

Residual automata (RFAs for short) are finite-state automata for which each state defines a *residual* of its language, where the residual of a language L by a word u is defined as the set of words w such that $uw \in L$. The class of RFAs lies between deterministic (DFAs) and nondeterministic automata (NFAs). They share with DFAs a significant property: the existence of a canonical minimal form for any regular language. On the other hand, they share with NFAs the existence of automata that are exponentially smaller (in the number of states) than the corresponding minimal DFA for the language. These properties make RFAs specially appealing in certain areas of computer science such as Grammatical Inference [10, 14].

RFAs were first introduced by Denis et al. [8, 9]. They defined an algorithm for *residualizing* an automaton, which is a variation of the well-known subset construction used for determinization, and showed that there exists a *unique canonical* RFA, which is minimal in



© Pierre Ganty, Elena Gutiérrez, and Pedro Valero;
licensed under Creative Commons License CC-BY

45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020).

Editors: Javier Esparza and Daniel Král'; Article No. 40; pp. 40:1–40:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the number of states, for every regular language. Moreover, they showed that the residual-equivalent of the double-reversal method [4] holds, i.e. residualizing an automaton \mathcal{N} whose reverse is residual yields the canonical RFA for the language accepted by \mathcal{N} .

Later, Tamm [16] generalized the double-reversal method for RFAs by giving a sufficient and necessary condition that guarantees that the residualization operation defined by Denis et al. [9] yields the canonical RFA. In fact, this generalization comes in the same lines as that of Brzozowski and Tamm [5] for the double-reversal method for building the minimal DFA.

These results evidence the existence of a relationship between RFAs and DFAs. In fact, a connection between these two classes of automata was already established by Myers et al. [1, 15] from a category-theoretical point of view. Concretely, they [1] use this perspective to address the residual-equivalent of the double-reversal method proposed by Denis et al. [9] to obtain the canonical RFA.

In this work we evidence this connection between RFAs and DFAs from the point of view of quasiorders over words. Specifically, we show that *quasiorders* are fundamental to RFAs as *congruences* are for DFAs.

Previously, we studied the problem of building DFAs using congruences, i.e., equivalence relations over words with good properties w.r.t. concatenation [11]. This way, we derived several well-known results about minimization of DFAs, including the double-reversal method and its generalization by Brzozowski and Tamm [5]. While the use of congruences over words suited for the construction of a subclass of residual automata, namely, *deterministic* automata, these are no longer useful to describe the more general class of *nondeterministic* residual automata. By moving from *congruences* over words to *quasiorders*, we are able to introduce nondeterminism in our automata constructions.

We consider quasiorders with good properties w.r.t. *right* and *left* concatenation. In particular, we define the so-called right *language-based* quasiorder, whose definition relies on a given regular language; and the right *automata-based* quasiorder, whose definition relies on a finite representation of the language, i.e., an automaton. We also give counterpart definitions for quasiorders that behave well with respect to *left* concatenation. Relying on quasiorders that *preserve* a given regular language, i.e., the closure of the language w.r.t. the quasiorder coincides with the language, we will provide a framework of finite-state automata constructions for the language.

When instantiating our automata constructions using the right language-based quasiorder, we obtain the canonical RFA for the given language; while using the right automata-based quasiorder yields an RFA for the language accepted by the automaton that has, at most, as many states as the RFA obtained by the residualization operation defined by Denis et al. [9]. Similarly, *left* automata-based and language-based quasiorders yield co-residual automata, i.e., automata whose reverse is residual.

Our quasiorder-based framework allows us to give a simple correctness proof of the double-reversal method for building the canonical RFA. Moreover, it allows us to generalize this method in the same fashion as Brzozowski and Tamm [5] generalized the double-reversal method for building the minimal DFA. Specifically, we give a characterization of the class of automata for which our automata-based quasiorder construction yields the canonical RFA.

We compare our characterization with the class of automata, defined by Tamm [16], for which the residualization operation of Denis et al. [9] yields the canonical RFA and show that her class of automata is strictly contained in the class we define. Furthermore, we highlight the connection between the generalization of Brzozowski and Tamm [5] and the one of Tamm [16] for the double-reversal methods for DFAs and RFAs, respectively.

Finally, we revisit the problem of learning residual automata from a quasiorder-based perspective. Specifically, we observe that the NL^* algorithm defined by Bollig et al. [3], inspired by the popular Angluin's L^* algorithm for learning DFAs [2], can be seen as an algorithm that starts from a quasiorder and refines it at each iteration. At the end of each iteration, the automaton built by NL^* coincides with our quasiorder-based automata construction applied to the refined quasiorder.

Structure of the paper. After preliminaries in Section 2, we introduce in Section 3 automata constructions based on quasiorders and establish the duality between these constructions when using right and left quasiorders. We instantiate these constructions in Section 4 with the language-based and automata-based quasiorders and study the relations between the resulting automata. As a consequence, we derive in Section 5 a generalization of the double-reversal method for building the canonical RFA for a language. In addition, we show a novel quasiorder-based perspective on the NL^* algorithm for learning residual automata in Section 6. For space reasons, a formal description of the NL^* algorithm as well as supplementary results, including the pseudocode of our quasiorder-based version of NL^* , and missing proofs are deferred to the extended version of this paper [12].

2 Preliminaries

Languages. Let Σ be a finite nonempty *alphabet* of symbols. Given a word $w \in \Sigma^*$, we will use $|w|$ to denote the *length* of w . We denote w^R the *reverse* of w . Given a language $L \subseteq \Sigma^*$, $L^R \stackrel{\text{def}}{=} \{w^R \mid w \in L\}$ denotes the *reverse language* of L and L^c , its *complement* language.

We denote the left (resp. right) quotient of L by a word u , also known as *residual*, as $u^{-1}L \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid uw \in L\}$ (resp. $Lu^{-1} \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid wu \in L\}$). Denis et al. [9] defined the notion of *composite* and *prime* residuals that we extend to right quotients as follows. A left (resp. right) quotient $u^{-1}L$ (resp. Lu^{-1}) is *composite* iff it is the union of all the left (resp. right) quotients that it strictly contains, i.e. $u^{-1}L = \bigcup_{x \in \Sigma^*, x^{-1}L \subsetneq u^{-1}L} x^{-1}L$ (resp. $Lu^{-1} = \bigcup_{x \in \Sigma^*, Lx^{-1} \subsetneq Lu^{-1}} Lx^{-1}$). Otherwise, we say the quotient is *prime*.

Automata. A (*nondeterministic*) *finite-state automaton* (NFA for short), or simply *automaton*, is a 5-tuple $\mathcal{N} = (Q, \Sigma, \delta, I, F)$, where Q is a finite set of *states*, Σ is an alphabet, $I \subseteq Q$ are the *initial* states, $F \subseteq Q$ are the *final* states, and $\delta : Q \times \Sigma \rightarrow \wp(Q)$ is the *transition* function, where $\wp(Q)$ denotes the powerset w.r.t. Q . We denote the *extended transition function* from Σ to Σ^* by $\hat{\delta}$, defined in the usual way, and, given $w \in \Sigma^*$ and $S \in \wp(Q)$, we define $\text{post}_w^{\mathcal{N}}(S) \stackrel{\text{def}}{=} \{q \in Q \mid \exists q' \in S, q \in \hat{\delta}(q', w)\}$ and $\text{pre}_w^{\mathcal{N}}(S) \stackrel{\text{def}}{=} \{q \in Q \mid \exists q' \in S, q' \in \hat{\delta}(q, w)\}$.

Given $S, T \subseteq Q$, $W_{S,T}^{\mathcal{N}} \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid \exists q \in S, q' \in T, q' \in \hat{\delta}(q, w)\}$. In particular, when $S = \{q\}$ and $T = F$, we say that $W_{q,F}^{\mathcal{N}}$ is the *right language* of state q . Likewise, when $S = I$ and $T = \{q\}$, we say that $W_{I,q}^{\mathcal{N}}$ is the *left language* of state q . In general, we omit the automaton \mathcal{N} from the superscript when it is clear from the context. We say that a state q is *unreachable* iff $W_{I,q}^{\mathcal{N}} = \emptyset$ and we say that q is *empty* iff $W_{q,F}^{\mathcal{N}} = \emptyset$. Finally, the language accepted by an automaton \mathcal{N} is $\mathcal{L}(\mathcal{N}) = \bigcup_{q \in I} W_{q,F}^{\mathcal{N}} = \bigcup_{q \in F} W_{I,q}^{\mathcal{N}} = W_{I,F}^{\mathcal{N}}$.

The NFA $\mathcal{N}' = (Q', \Sigma, \delta', I', F')$ is a *sub-automaton* of \mathcal{N} iff $Q' \subseteq Q$, $I' \subseteq I$, $F' \subseteq F$ and $q' \in \delta'(q, a) \Rightarrow q' \in \delta(q, a)$ with $q, q' \in Q$ and $a \in \Sigma$. The *reverse* of \mathcal{N} , denoted by \mathcal{N}^R , is defined as $\mathcal{N}^R = (Q, \Sigma, \delta_r, F, I)$ where $q \in \delta_r(q', a)$ iff $q' \in \delta(q, a)$. Clearly, $\mathcal{L}(\mathcal{N})^R = \mathcal{L}(\mathcal{N}^R)$.

Residual Automata. A *residual finite-state automaton* (RFA for short) is an NFA such that the right language of each state is a left quotient of the accepted language. We write RFA instead of RFSA [9] to be consistent with the abbreviations NFA and DFA. Formally, an RFA is an automaton $\mathcal{N} = (Q, \Sigma, \delta, I, F)$ such that $\forall q \in Q, \exists u \in \Sigma^*, W_{q,F}^{\mathcal{N}} = u^{-1}\mathcal{L}(\mathcal{N})$.

We say an automaton is *co-residual* (co-RFA for short) if its reverse is an RFA, i.e., $\forall q \in Q, \exists u \in \Sigma^*, W_{I,q}^{\mathcal{N}} = \mathcal{L}(\mathcal{N})u^{-1}$. We say $u \in \Sigma^*$ is a *characterizing word* for $q \in Q$ iff $W_{q,F}^{\mathcal{N}} = u^{-1}\mathcal{L}(\mathcal{N})$ and we say \mathcal{N} is *consistent* iff every state q is reachable by a characterizing word for q . Moreover, \mathcal{N} is *strongly consistent* iff every state q is reachable by every characterizing word of q .

Denis et al. [9] define a *residualization* operation that, given NFA \mathcal{N} , builds an RFA \mathcal{N}^{res} such that $\mathcal{L}(\mathcal{N}^{\text{res}}) = \mathcal{L}(\mathcal{N})$. Let $\mathcal{N} = (Q, \Sigma, \delta, I, F)$ be an NFA and $u \in \Sigma^*$, the set $\text{post}_u^{\mathcal{N}}(I)$ is *coverable* iff $\text{post}_u^{\mathcal{N}}(I) = \bigcup_{x \in \Sigma^*, \text{post}_x^{\mathcal{N}}(I) \subseteq \text{post}_u^{\mathcal{N}}(I)} \text{post}_x^{\mathcal{N}}(I)$. Define $\mathcal{N}^{\text{res}} \stackrel{\text{def}}{=} (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{I}, \tilde{F})$ as an RFA with $\tilde{Q} = \{\text{post}_u^{\mathcal{N}}(I) \mid u \in \Sigma^* \wedge \text{post}_u^{\mathcal{N}}(I) \text{ is not coverable}\}$, $\tilde{I} = \{S \in \tilde{Q} \mid S \subseteq I\}$, $\tilde{F} = \{S \in \tilde{Q} \mid S \cap F \neq \emptyset\}$ and $\tilde{\delta}(S, a) = \{S' \in \tilde{Q} \mid S' \subseteq \delta(S, a)\}$ for every $S \in \tilde{Q}$ and $a \in \Sigma$.

Finally, the *canonical* RFA for a regular language L is the RFA $\mathcal{C} \stackrel{\text{def}}{=} (Q, \Sigma, \delta, I, F)$ with $Q = \{u^{-1}L \mid u \in \Sigma^* \wedge u^{-1}L \text{ is prime}\}$, $I = \{u^{-1}L \in Q \mid u^{-1}L \subseteq L\}$, $F = \{u^{-1}L \in Q \mid \varepsilon \in u^{-1}L\}$ and $\delta(u^{-1}L, a) = \{v^{-1}L \in Q \mid v^{-1}L \subseteq a^{-1}(u^{-1}L)\}$ for every $u^{-1}L \in Q$ and $a \in \Sigma$. As shown by Denis et al. [9], the canonical RFA is a strongly consistent RFA and it is the minimal (in number of states) RFA such that $\mathcal{L}(\mathcal{N}) = L$. Moreover, the canonical RFA is maximal in the number of transitions.

Quasiorders. A *quasiorder* over Σ^* (qo for short) \preceq is a reflexive and transitive binary relation over Σ^* . A *symmetric* qo is called an *equivalence relation*. A quasiorder \preceq is a *right (resp. left) quasiorder* and we denote it \preceq^r (resp. \preceq^ℓ) iff for all $u, v \in \Sigma^*$, we have that $u \preceq v \Rightarrow ua \preceq va$ (resp. $u \preceq v \Rightarrow au \preceq av$), for all $a \in \Sigma$. For example, the quasiorder defined by $u \preceq_{\text{len}} v \stackrel{\text{def}}{\iff} |u| \leq |v|$, is a left and right qo but not an equivalence relation.

Given two qo's \preceq and \preceq' , we say that \preceq is *finer* than \preceq' (or \preceq' is *coarser* than \preceq) iff $\preceq \subseteq \preceq'$. For every qo \preceq , we define its *strict* version as: $u \prec v \stackrel{\text{def}}{\iff} u \preceq v \wedge v \not\preceq u$ and we define $(\preceq)^{-1}$ as: $u (\preceq)^{-1} v \stackrel{\text{def}}{\iff} v \preceq u$. Note that every qo \preceq induces an equivalence relation defined as $\sim \stackrel{\text{def}}{=} \preceq \cap (\preceq)^{-1}$.

We adopt the definition of *closure* of a subset of $S \subseteq \Sigma^*$ w.r.t. a qo \preceq introduced by de Luca and Varricchio [7]. Concretely, given a qo \preceq on Σ^* and a subset $S \subseteq \Sigma^*$, we define the *upper closure* (or simply *closure*) of S w.r.t. \preceq as $\text{cl}_{\preceq}(S) \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid \exists x \in S, x \preceq w\}$. We say that $\text{cl}_{\preceq}(S)$ is a *principal* iff $\text{cl}_{\preceq}(S) = \text{cl}_{\preceq}(\{u\})$, for some $u \in \Sigma^*$. In that case, we write $\text{cl}_{\preceq}(u)$ instead of $\text{cl}_{\preceq}(\{u\})$. Note that, $\text{cl}_{\preceq}(u) = \text{cl}_{\preceq}(v)$, for all $v \in \Sigma^*$ such that $u \sim v$. Finally, given a language $L \subseteq \Sigma^*$, we say that a qo \preceq is *L-preserving* iff $\text{cl}_{\preceq}(L) = L$.

3 Automata Constructions from Quasiorders

We will consider right and left quasiorders on Σ^* (and their corresponding closures) and we will use them to define RFAs constructions for regular languages. The following lemma gives a characterization of right and left quasiorders.

► **Lemma 1.** *The following properties hold:*

1. \preceq^r is a right quasiorder iff $\text{cl}_{\preceq^r}(u)v \subseteq \text{cl}_{\preceq^r}(uv)$, for all $u, v \in \Sigma^*$.
2. \preceq^ℓ is a left quasiorder iff $v \text{cl}_{\preceq^\ell}(u) \subseteq \text{cl}_{\preceq^\ell}(vu)$, for all $u, v \in \Sigma^*$.

Given a regular language L , we are interested in left and right quasiorders that are *L-preserving*. We will use the principals of these quasiorders as states of automata constructions that yield RFAs and co-RFAs accepting the language L . Therefore, in the sequel, we will only consider quasiorders that induce a finite number of principals, i.e., quasiorders \preceq such that the induced equivalence $\sim \stackrel{\text{def}}{=} \preceq \cap (\preceq)^{-1}$ has finite index.

Next, we introduce the notion of L -composite principals which, intuitively, correspond to states of our automata constructions that can be removed without altering the language accepted by the automata.

► **Definition 2** (L -Composite Principal). *Let L be a regular language and let \preceq^r (resp. \preceq^ℓ) be a right (resp. left) quasiorder on Σ^* . Given $u \in \Sigma^*$, the principal $\text{cl}_{\preceq^r}(u)$ (resp. $\text{cl}_{\preceq^\ell}(u)$) is L -composite iff*

$$u^{-1}L = \bigcup_{x \in \Sigma^*, x \prec^r u} x^{-1}L \quad (\text{resp. } Lu^{-1} = \bigcup_{x \in \Sigma^*, x \prec^\ell u} Lx^{-1})$$

If $\text{cl}_{\preceq^r}(u)$ (resp. $\text{cl}_{\preceq^\ell}(u)$) is not L -composite then it is L -prime.

We sometimes use the terms *composite* and *prime principal* when the language L is clear from the context. Observe that, if $\text{cl}_{\preceq^r}(u)$ is L -composite, for some $u \in \Sigma^*$, then so is $\text{cl}_{\preceq^r}(v)$, for every $v \in \Sigma^*$ such that $u \sim^r v$. The same holds for a left quasiorder \preceq^ℓ .

Given a regular language L and a right quasiorder \preceq^r that is L -preserving, the following automata construction yields an RFA that accepts exactly the language L .

► **Definition 3** (Automata construction $H^r(\preceq^r, L)$). *Let \preceq^r be a right quasiorder and let $L \subseteq \Sigma^*$ be a language. Define the automaton $H^r(\preceq^r, L) \stackrel{\text{def}}{=} (Q, \Sigma, \delta, I, F)$ where $Q = \{\text{cl}_{\preceq^r}(u) \mid u \in \Sigma^*, \text{cl}_{\preceq^r}(u) \text{ is } L\text{-prime}\}$, $I = \{\text{cl}_{\preceq^r}(u) \in Q \mid \varepsilon \in \text{cl}_{\preceq^r}(u)\}$, $F = \{\text{cl}_{\preceq^r}(u) \in Q \mid u \in L\}$ and $\delta(\text{cl}_{\preceq^r}(u), a) = \{\text{cl}_{\preceq^r}(v) \in Q \mid \text{cl}_{\preceq^r}(u)a \subseteq \text{cl}_{\preceq^r}(v)\}$ for all $\text{cl}_{\preceq^r}(u) \in Q, a \in \Sigma$.*

► **Lemma 4.** *Let $L \subseteq \Sigma^*$ be a regular language and let \preceq^r be a right L -preserving quasiorder. Then $H^r(\preceq^r, L)$ is an RFA such that $\mathcal{L}(H^r(\preceq^r, L)) = L$.*

Given a regular language L and a left L -preserving quasiorder \preceq^ℓ , we can give a similar automata construction of a co-RFA that recognizes exactly the language L .

► **Definition 5** (Automata construction $H^\ell(\preceq^\ell, L)$). *Let \preceq^ℓ be a left quasiorder and let $L \subseteq \Sigma^*$ be a language. Define the automaton $H^\ell(\preceq^\ell, L) = (Q, \Sigma, \delta, I, F)$ where $Q = \{\text{cl}_{\preceq^\ell}(u) \mid u \in \Sigma^*, \text{cl}_{\preceq^\ell}(u) \text{ is } L\text{-prime}\}$, $I = \{\text{cl}_{\preceq^\ell}(u) \in Q \mid u \in L\}$, $F = \{\text{cl}_{\preceq^\ell}(u) \in Q \mid \varepsilon \in \text{cl}_{\preceq^\ell}(u)\}$, and $\delta(\text{cl}_{\preceq^\ell}(u), a) = \{\text{cl}_{\preceq^\ell}(v) \in Q \mid a \text{cl}_{\preceq^\ell}(v) \subseteq \text{cl}_{\preceq^\ell}(u)\}$ for all $\text{cl}_{\preceq^\ell}(u) \in Q, a \in \Sigma$.*

► **Lemma 6.** *Let $L \subseteq \Sigma^*$ be a language and let \preceq^ℓ be a left L -preserving quasiorder. Then $H^\ell(\preceq^\ell, L)$ is a co-RFA such that $\mathcal{L}(H^\ell(\preceq^\ell, L)) = L$.*

Observe that the automaton $H^r = H^r(\preceq^r, L)$ (resp. $H^\ell = H^\ell(\preceq^\ell, L)$) is *finite*, since we assume \preceq^r (resp. \preceq^ℓ) induces a finite number of principals. Note also that H^r (resp. H^ℓ) possibly contains empty (resp. unreachable) states but no state is unreachable (resp. empty).

Moreover, notice that by keeping all principals of \preceq^r (resp. \preceq^ℓ) as states, instead of only the prime ones as in Definition 3 (resp. Definition 5), we would obtain an RFA (resp. a co-RFA) with (possibly) more states that also recognizes L .

The following lemma shows that H^r and H^ℓ inherit the left-right duality between \preceq^r and \preceq^ℓ through the reverse operation.

► **Lemma 7.** *Let \preceq^r and \preceq^ℓ be a right and a left quasiorder, respectively, and let $L \subseteq \Sigma^*$ be a language. If $u \preceq^r v \Leftrightarrow u^R \preceq^\ell v^R$ then $H^r(\preceq^r, L)$ is isomorphic to $(H^\ell(\preceq^\ell, L^R))^R$.*

Finally, it follows from the next theorem that given two right L -preserving quasiorders, \preceq_1^r and \preceq_2^r , if $\preceq_1^r \subseteq \preceq_2^r$ then the automaton $H^r(\preceq_1^r, L)$ has, at least, as many states as $H^r(\preceq_2^r, L)$. The same holds for left L -preserving quasiorders and H^ℓ . Observe that this is not obvious since only the L -prime principals correspond to states of the automata construction.

► **Theorem 8.** Let $L \subseteq \Sigma^*$ be a language and let \preceq_1 and \preceq_2 be two left or two right L -preserving quasiorders. If $\preceq_1 \subseteq \preceq_2$ then:

$$|\{\text{cl}_{\preceq_1}(u) \mid u \in \Sigma^* \wedge \text{cl}_{\preceq_1}(u) \text{ is } L\text{-prime}\}| \geq |\{\text{cl}_{\preceq_2}(u) \mid u \in \Sigma^* \wedge \text{cl}_{\preceq_2}(u) \text{ is } L\text{-prime}\}| .$$

4 Language-based Quasiorders and their Approximation using NFAs

In this section we instantiate our automata constructions using two classes of quasiorders, namely, the so-called *Nerode's* quasiorders [6], whose definition is based on a given regular language; and the *automata-based* quasiorders, whose definition relies on a given automaton.

► **Definition 9** (Language-based Quasiorders). Let $u, v \in \Sigma^*$ and let $L \subseteq \Sigma^*$ be a language. Define:

$$u \preceq_L^r v \stackrel{\text{def}}{\iff} u^{-1}L \subseteq v^{-1}L \quad \text{Right-language-based Quasiorder} \quad (1)$$

$$u \preceq_L^\ell v \stackrel{\text{def}}{\iff} Lu^{-1} \subseteq Lv^{-1} \quad \text{Left-language-based Quasiorder} \quad (2)$$

It is well-known that for every regular language L there exists a finite number of quotients $u^{-1}L$ [7]. Therefore, the language-based quasiorders defined above induce a finite number of principals since each principal set is determined by a quotient of L .

► **Definition 10** (Automata-based Quasiorders). Let $u, v \in \Sigma^*$ and let $\mathcal{N} = (Q, \Sigma, \delta, I, F)$ be an NFA. Define:

$$u \preceq_{\mathcal{N}}^r v \stackrel{\text{def}}{\iff} \text{post}_u^{\mathcal{N}}(I) \subseteq \text{post}_v^{\mathcal{N}}(I) \quad \text{Right-Automata-based Quasiorder} \quad (3)$$

$$u \preceq_{\mathcal{N}}^\ell v \stackrel{\text{def}}{\iff} \text{pre}_u^{\mathcal{N}}(F) \subseteq \text{pre}_v^{\mathcal{N}}(F) \quad \text{Left-Automata-based Quasiorder} \quad (4)$$

Clearly, the automata-based quasiorders induce a finite number of principals since each principal is represented by a subset of the states of \mathcal{N} .

► **Remark 11.** The pairs of quasiorders $\preceq_L^r - \preceq_L^\ell$ and $\preceq_{\mathcal{N}}^r - \preceq_{\mathcal{N}}^\ell$ from Definitions 9 and 10 are dual, i.e. $u \preceq_L^r v \iff u^R \preceq_L^\ell v^R$ and $u \preceq_{\mathcal{N}}^r v \iff u^R \preceq_{\mathcal{N}}^\ell v^R$.

The following result shows that the principals of $\preceq_{\mathcal{N}}^r$ and $\preceq_{\mathcal{N}}^\ell$ can be described, respectively, as intersections of left and right languages of the states of \mathcal{N} while the principals of \preceq_L^r and \preceq_L^ℓ correspond to intersections of quotients of L .

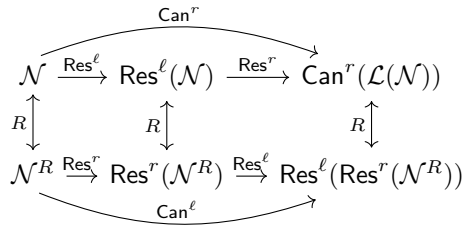
► **Lemma 12.** Let $\mathcal{N} = (Q, \Sigma, \delta, I, F)$ be an NFA with $\mathcal{L}(\mathcal{N}) = L$. Then, for every $u \in \Sigma^*$,

$$\begin{aligned} \text{cl}_{\preceq_{\mathcal{N}}^r}(u) &= \bigcap_{q \in \text{post}_u^{\mathcal{N}}(I)} W_{I,q}^{\mathcal{N}} & \text{cl}_{\preceq_L^r}(u) &= \bigcap_{w \in \Sigma^*, w \in u^{-1}L} Lw^{-1} \\ \text{cl}_{\preceq_{\mathcal{N}}^\ell}(u) &= \bigcap_{q \in \text{pre}_u^{\mathcal{N}}(F)} W_{q,F}^{\mathcal{N}} & \text{cl}_{\preceq_L^\ell}(u) &= \bigcap_{w \in \Sigma^*, w \in Lu^{-1}} w^{-1}L . \end{aligned}$$

As shown by Ganty et al. [13], given an NFA \mathcal{N} with $L = \mathcal{L}(\mathcal{N})$, the quasiorders \preceq_L^r and $\preceq_{\mathcal{N}}^r$ are right L -preserving quasiorders, while the quasiorders \preceq_L^ℓ and $\preceq_{\mathcal{N}}^\ell$ are left L -preserving quasiorders. Therefore, by Lemma 4 and 6, our automata constructions applied to these quasiorders yield automata for L .

Finally, as shown by de Luca and Varricchio [6], we have that $\preceq_{\mathcal{N}}^r$ is finer than \preceq_L^r , i.e., $\preceq_{\mathcal{N}}^r \subseteq \preceq_L^r$. In that sense we say that $\preceq_{\mathcal{N}}^r$ *approximates* \preceq_L^r . As the following lemma shows, the approximation is precise, i.e., $\preceq_{\mathcal{N}}^r = \preceq_L^r$, whenever \mathcal{N} is a co-RFA with no empty states.

► **Lemma 13.** Let \mathcal{N} be a co-RFA with no empty states such that $L = \mathcal{L}(\mathcal{N})$. Then $\preceq_L^r = \preceq_{\mathcal{N}}^r$. Similarly, if \mathcal{N} is an RFA with no unreachable states and $L = \mathcal{L}(\mathcal{N})$ then $\preceq_L^\ell = \preceq_{\mathcal{N}}^\ell$.



■ **Figure 1** Relations between the constructions $\text{Res}^\ell, \text{Res}^r, \text{Can}^\ell$ and Can^r . Note that constructions Can^r and Can^ℓ are applied to the language accepted by the automaton in the origin of the labeled arrow while constructions Res^r and Res^ℓ are applied directly to the automaton.

The upper part of the diagram follows from Theorem 15 (f), the squares follow from Theorem 15 (c) and the bottom curved arc follows from Theorem 15 (b). Incidentally, the diagram shows a new relation which is a consequence of the left-right dualities between \preceq_L^ℓ and \preceq_L^r , and $\preceq_{\mathcal{N}}^\ell$ and $\preceq_{\mathcal{N}}^r$: $\text{Can}^\ell(\mathcal{L}(\mathcal{N}^R))$ is isomorphic to $\text{Res}^\ell(\text{Res}^r(\mathcal{N}^R))$.

4.1 Automata Constructions

In what follows, we will use $\text{Can}^r, \text{Can}^\ell$ and $\text{Res}^r, \text{Res}^\ell$ to denote the constructions $\text{H}^r, \text{H}^\ell$ when applied, respectively, to the language-based quasiorders induced by a regular language and the automata-based quasiorders induced by an NFA.

► **Definition 14.** Let \mathcal{N} be an NFA accepting the language $L = \mathcal{L}(\mathcal{N})$. Define:

$$\begin{aligned} \text{Can}^r(L) &\stackrel{\text{def}}{=} \text{H}^r(\preceq_L^r, L) & \text{Res}^r(\mathcal{N}) &\stackrel{\text{def}}{=} \text{H}^r(\preceq_{\mathcal{N}}^r, L) \\ \text{Can}^\ell(L) &\stackrel{\text{def}}{=} \text{H}^\ell(\preceq_L^\ell, L) & \text{Res}^\ell(\mathcal{N}) &\stackrel{\text{def}}{=} \text{H}^\ell(\preceq_{\mathcal{N}}^\ell, L) . \end{aligned}$$

Given an NFA \mathcal{N} accepting the language $L = \mathcal{L}(\mathcal{N})$, all constructions in the above definition yield automata accepting L . However, while the constructions using the right quasiorders result in RFAs, those using left quasiorders result in co-RFAs. Furthermore, it follows from Remark 11 and Lemma 7 that $\text{Can}^\ell(L)$ is isomorphic to $(\text{Can}^r(L^R))^R$ and $\text{Res}^\ell(\mathcal{N})$ is isomorphic to $(\text{Res}^r(\mathcal{N}^R))^R$.

It follows from Theorem 8 that the automata $\text{Res}^r(\mathcal{N})$ and $\text{Res}^\ell(\mathcal{N})$ have more states than $\text{Can}^r(L)$ and $\text{Can}^\ell(L)$, respectively. Intuitively, $\text{Can}^r(L)$ is the minimal RFA for L , i.e. it is isomorphic to the canonical RFA for L , since \preceq_L^r is the coarsest right L -preserving quasiorder [6]. On the other hand, as we evidenced in Example 17, $\text{Res}^r(\mathcal{N})$ is a sub-automaton of \mathcal{N}^{res} [9] for every NFA \mathcal{N} .

Finally, it follows from Lemma 13 that residualizing (Res^r) a co-RFA with no empty states ($\text{Res}^\ell(\mathcal{N})$) results in the canonical RFA for $\mathcal{L}(\mathcal{N})$ ($\text{Can}^r(\mathcal{L}(\mathcal{N}))$).

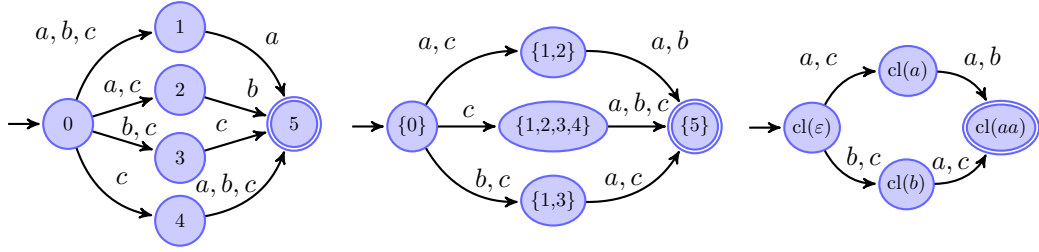
We formalize all these notions in Theorem 15. Figure 1 summarizes all these connections between the automata constructions given in Definition 14.

► **Theorem 15.** Let \mathcal{N} be an NFA with $L = \mathcal{L}(\mathcal{N})$. Then the following properties hold:

- $\mathcal{L}(\text{Can}^r(L)) = \mathcal{L}(\text{Can}^\ell(L)) = L = \mathcal{L}(\text{Res}^r(\mathcal{N})) = \mathcal{L}(\text{Res}^\ell(\mathcal{N}))$.
- $\text{Can}^\ell(L)$ is isomorphic to $(\text{Can}^r(L^R))^R$.
- $\text{Res}^\ell(\mathcal{N})$ is isomorphic to $(\text{Res}^r(\mathcal{N}^R))^R$.
- $\text{Can}^r(L)$ is isomorphic to the canonical RFA for L .
- $\text{Res}^r(\mathcal{N})$ is isomorphic to a sub-automaton of \mathcal{N}^{res} and $\mathcal{L}(\text{Res}^r(\mathcal{N})) = \mathcal{L}(\mathcal{N}^{\text{res}}) = L$.
- $\text{Res}^r(\text{Res}^\ell(\mathcal{N}))$ is isomorphic to $\text{Can}^r(L)$.

Let \mathcal{N} be an NFA with $L = \mathcal{L}(\mathcal{N})$. If $\preceq_L^r = \preceq_{\mathcal{N}}^r$ then the automata $\text{Can}^r(L)$ and $\text{Res}^r(\mathcal{N})$ are isomorphic. The following result shows that the reverse implication also holds.

► **Lemma 16.** Let \mathcal{N} be an NFA with $L = \mathcal{L}(\mathcal{N})$. Then $\preceq_L^r = \preceq_{\mathcal{N}}^r$ iff $\text{Res}^r(\mathcal{N})$ is isomorphic to $\text{Can}^r(\mathcal{L}(\mathcal{N}))$.



■ **Figure 2** An NFA \mathcal{N} and the RFAs \mathcal{N}^{res} and $\text{Res}^r(\mathcal{N})$. We omit the empty states for clarity.

The following example illustrates the differences between our residualization operation, $\text{Res}^r(\mathcal{N})$, and the one defined by Denis et al. [9], \mathcal{N}^{res} , on a given NFA \mathcal{N} : the automaton $\text{Res}^r(\mathcal{N})$ has, at most, as many states as \mathcal{N}^{res} . This follows from the fact that for every $u \in \Sigma^*$, if $\text{post}_u^{\mathcal{N}}(I)$ is coverable then $\text{cl}_{\preceq_{\mathcal{N}}^r}(u)$ is composite but *not* vice-versa.

► **Example 17.** Let $\mathcal{N} = (Q, \Sigma, \delta, I, F)$ be the automata on the left of Figure 2 and let $L = \mathcal{L}(\mathcal{N})$. To build \mathcal{N}^{res} we compute $\text{post}_u^{\mathcal{N}}(I)$, for all $u \in \Sigma^*$. Let $C \stackrel{\text{def}}{=} L^c \setminus \{\varepsilon, a, b, c\}$.

$$\begin{array}{lll} \text{post}_{\varepsilon}^{\mathcal{N}}(I) = \{0\} & \text{post}_a^{\mathcal{N}}(I) = \{1, 2\} & \forall w \in L, \text{post}_w^{\mathcal{N}}(I) = \{5\} \\ \text{post}_c^{\mathcal{N}}(I) = \{1, 2, 3, 4\} & \text{post}_b^{\mathcal{N}}(I) = \{1, 3\} & \forall w \in C, \text{post}_w^{\mathcal{N}}(I) = \emptyset \end{array}$$

Since none of these sets is coverable by the others, they are all states of \mathcal{N}^{res} . The resulting RFA \mathcal{N}^{res} is shown in the center of Figure 2. On the other hand, let us denote $\text{cl}_{\preceq_{\mathcal{N}}^r}$ simply by cl . In order to build $\text{Res}^r(\mathcal{N})$ we need to compute the principals $\text{cl}(u)$, for all $u \in \Sigma^*$. By definition of $\preceq_{\mathcal{N}}^r$, we have that $w \in \text{cl}(u) \Leftrightarrow \text{post}_u^{\mathcal{N}}(I) \subseteq \text{post}_w^{\mathcal{N}}(I)$. Therefore, we obtain:

$$\text{cl}(\varepsilon) = \{\varepsilon\} \quad \text{cl}(a) = \{a, c\} \quad \text{cl}(b) = \{b, c\} \quad \text{cl}(c) = \{c\} \quad \forall w \in L, \text{cl}(w) = L \quad \forall w \in C, \text{cl}(w) = \Sigma^* .$$

Since $a \prec_{\mathcal{N}}^r c$, $b \prec_{\mathcal{N}}^r c$ and $\forall w \in \Sigma^*$, $cw \subseteq L \Leftrightarrow (aw \subseteq L \vee bw \subseteq L)$, it follows that $\text{cl}(c)$ is L -composite. The resulting RFA $\text{Res}^r(\mathcal{N})$ is shown on the right of Figure 2. \square

5 Double-Reversal Method for Building the Canonical RFA

Denis et al. [9] show that their residualization operation satisfies the residual-equivalent of the double-reversal method for building the minimal DFA. More specifically, they prove that if an NFA \mathcal{N} is a co-RFA with no empty states then their residualization operation applied to \mathcal{N} results in the canonical RFA for $\mathcal{L}(\mathcal{N})$. As a consequence, $((\mathcal{N}^R)^{\text{res}})^R$ is the canonical RFA for $\mathcal{L}(\mathcal{N})$.

In this section we first show that the residual-equivalent of the double-reversal method holds within our framework, i.e. $\text{Res}^r((\text{Res}^r(\mathcal{N}^R))^R)$ is isomorphic to $\text{Can}^r(\mathcal{N})$. Then, we generalize this method along the lines of the generalization of the double-reversal method for building the minimal DFA given by Brzozowski and Tamm [5]. To this end, we extend our previous work [11] in which we provided a congruence-based perspective on the generalized double-reversal method for DFAs. By moving from congruences to quasiorders, we find a *necessary* and *sufficient* condition on an NFA \mathcal{N} so that $\text{Res}^r(\mathcal{N})$ yields the canonical RFA for $\mathcal{L}(\mathcal{N})$. Finally, we compare our generalization with the one given by Tamm [16].

5.1 Double-reversal Method

We give a simple proof of the double-reversal method for building the canonical RFA.

► **Theorem 18** (Double-Reversal). *Let \mathcal{N} be an NFA. Then $\text{Res}^r((\text{Res}^r(\mathcal{N}^R))^R)$ is isomorphic to the canonical RFA for $\mathcal{L}(\mathcal{N})$.*

Proof. It follows from Theorem 15 (c), (d) and (f). ◀

Note that Theorem 18 can be inferred from Figure 1 by following the path starting at \mathcal{N} , labeled with $R - \text{Res}^r - R - \text{Res}^r$ and ending in $\text{Can}^r(\mathcal{L}(\mathcal{N}))$.

5.2 Generalization of the Double-reversal Method

Next we show that residualizing an automaton yields the canonical RFA iff the left language of every state is closed w.r.t. the right Nerode quasiorder.

► **Theorem 19.** *Let $\mathcal{N} = (Q, \Sigma, \delta, I, F)$ be an NFA with $L = \mathcal{L}(\mathcal{N})$. Then $\text{Res}^r(\mathcal{N})$ is the canonical RFA for L iff $\forall q \in Q, \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$.*

Proof. We first show that $\forall q \in Q, \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$ is a *necessary* condition, i.e. if $\text{Res}^r(\mathcal{N})$ is the canonical RFA for L then $\forall q \in Q, \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$ holds. By Lemma 16 we have that if $\text{Res}^r(\mathcal{N})$ is the canonical RFA then $\preceq_L^r = \preceq_{\mathcal{N}}^r$. Moreover,

$$\begin{aligned} \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}) &= \text{[By definition of } \text{cl}_{\preceq_L^r}] \\ \{w \in \Sigma^* \mid \exists u \in W_{I,q}^{\mathcal{N}}, u^{-1}L \subseteq w^{-1}L\} &= \text{[Since } \preceq_L^r = \preceq_{\mathcal{N}}^r] \\ \{w \in \Sigma^* \mid \exists u \in W_{I,q}^{\mathcal{N}}, \text{post}_u^{\mathcal{N}}(I) \subseteq \text{post}_w^{\mathcal{N}}(I)\} &\subseteq \text{[Since } u \in W_{I,q}^{\mathcal{N}} \Leftrightarrow q \in \text{post}_u^{\mathcal{N}}(I)] \\ \{w \in \Sigma^* \mid q \in \text{post}_w^{\mathcal{N}}(I)\} &= \text{[By definition of } W_{I,q}^{\mathcal{N}}] \\ &= W_{I,q}^{\mathcal{N}}. \end{aligned}$$

By reflexivity of \preceq_L^r , we conclude that $\text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$.

Next, we show that $\forall q \in Q, \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$ is also a *sufficient* condition. By Lemma 12 and condition $\forall q \in Q, \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$, we have that

$$\text{cl}_{\preceq_{\mathcal{N}}^r}(u) = \bigcap_{q \in \text{post}_u^{\mathcal{N}}(I)} W_{I,q}^{\mathcal{N}} = \bigcap_{q \in \text{post}_u^{\mathcal{N}}(I)} \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}).$$

Since $u \in \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}})$ for all $q \in \text{post}_u^{\mathcal{N}}(I)$, it follows that $\text{cl}_{\preceq_L^r}(u) \subseteq \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}})$ for all $q \in \text{post}_u^{\mathcal{N}}(I)$ and, since $\text{cl}_{\preceq_{\mathcal{N}}^r}(u) = \bigcap_{q \in \text{post}_u^{\mathcal{N}}(I)} \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}})$, we have that $\text{cl}_{\preceq_L^r}(u) \subseteq \text{cl}_{\preceq_{\mathcal{N}}^r}(u)$ for every $u \in \Sigma^*$, i.e., $\preceq_L^r \subseteq \preceq_{\mathcal{N}}^r$.

On the other hand, as shown by de Luca and Varricchio [6], we have that $\preceq_{\mathcal{N}}^r \subseteq \preceq_L^r$. We conclude that $\preceq_{\mathcal{N}}^r = \preceq_L^r$, hence $\text{Res}^r(\mathcal{N}) = \text{Can}^r(L)$. ◀

It is worth to remark that Theorem 19 does not hold when considering the residualization operation \mathcal{N}^{res} of Denis et al. [9] instead of $\text{Res}^r(\mathcal{N})$. As a counterexample we have the automata \mathcal{N} in Figure 2 where $\text{Res}^r(\mathcal{N})$ is the canonical RFA for $\mathcal{L}(\mathcal{N})$, hence \mathcal{N} satisfies the condition of Theorem 19, while \mathcal{N}^{res} is not canonical.

Co-atoms and co-rests

The condition of Theorem 19 is analogue to the one we gave for building the minimal DFA [11], except that the later is formulated in terms of congruences instead of quasiorders. In that case we proved that determinizing a given NFA \mathcal{N} yields the minimal DFA iff $\text{cl}_{\sim_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$ for every state q of \mathcal{N} , where $\sim_L^r \stackrel{\text{def}}{=} \preceq_L^r \cap (\preceq_L^r)^{-1}$ is the right Nerode's congruence [7].

Moreover, we showed that the principals of \sim_L^r coincide with the so-called *co-atoms* [11], which are non-empty intersections of complemented and uncomplemented right quotients of the language. This allowed us to connect our result for DFAs [11] with the generalization

of the double-reversal method for building the minimal DFA proposed by Brzozowski and Tamm [5], who establish that determinizing an NFA \mathcal{N} yields the minimal DFA for $\mathcal{L}(\mathcal{N})$ iff the left languages of the states of \mathcal{N} are unions of co-atoms of $\mathcal{L}(\mathcal{N})$.

Next, we give a formulation of the condition from Theorem 19 along the lines of the one given by Brzozowski and Tamm [5] for their generalization of the double-reversal method for building the minimal DFA.

To do that, let us call the intersections used in Lemma 12 to describe the principals of \preceq_L^ℓ and \preceq_L^r as *rests* and *co-rests* of L , respectively. As shown by Theorem 19, residualizing an NFA \mathcal{N} yields the canonical RFA for $\mathcal{L}(\mathcal{N})$ iff the left language of every state of \mathcal{N} satisfies $\text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$. By definition, $\text{cl}_{\preceq_L^r}(S) = S$ iff S is a union of principals of \preceq_L^r which, by Lemma 12 are the co-rests of L .

Therefore we derive the following statement, equivalent to Theorem 19, that we consider as the residual-equivalent of the generalization of the double-reversal method for building the minimal DFA proposed by Brzozowski and Tamm [5].

► **Corollary 20.** *Let $\mathcal{N} = (Q, \Sigma, \delta, I, F)$ be an NFA with $L = \mathcal{L}(\mathcal{N})$. Then $\text{Res}^r(\mathcal{N})$ is the canonical RFA for L iff the left languages of \mathcal{N} are union of co-rests.*

Tamm's Generalization of the Double-reversal Method for RFAs

Tamm [16] generalized the double-reversal method of Denis et al. [9] by showing that \mathcal{N}^{res} is the canonical RFA for $\mathcal{L}(\mathcal{N})$ iff the left languages of \mathcal{N} are union of the left languages of the canonical RFA for $\mathcal{L}(\mathcal{N})$.

In this section, we compare the generalization of Tamm [16] with ours. The two approaches differ in the definition of the residualization operation they consider and, as the following lemma shows, the sufficient and necessary condition from Theorem 19 is more general than that of Tamm [16, Theorem 4]

► **Lemma 21.** *Let $\mathcal{N} = (Q, \Sigma, \delta, I, F)$ be an NFA and let $\mathcal{C} = \text{Can}^r(\preceq_L^r, L) = (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{I}, \tilde{F})$ be the canonical RFA for $L = \mathcal{L}(\mathcal{N})$. If $W_{I,q}^{\mathcal{N}} = \bigcup_{q \in \tilde{Q}} W_{I,q}^{\mathcal{C}}$ then $\text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$.*

Proof. Since the canonical RFA, \mathcal{C} , is strongly consistent then $\preceq_{\mathcal{C}}^r = \preceq_L^r$ [12], hence $\text{Res}^r(\mathcal{C})$ is isomorphic to $\text{Can}^r(L)$. It follows from Theorem 19 that $\text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{C}}) = W_{I,q}^{\mathcal{C}}$ for every $q \in \tilde{Q}$. Therefore,

$$\begin{aligned} \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}) &= [\text{Since } W_{I,q}^{\mathcal{N}} = \bigcup_{q \in \tilde{Q}} W_{I,q}^{\mathcal{C}} \text{ and } \text{cl}_{\preceq_L^r}(\cup S_i) = \cup \text{cl}_{\preceq_L^r}(S_i)] \\ \bigcup_{q \in \tilde{Q}} \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{C}}) &= [\text{Since } \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{C}}) = W_{I,q}^{\mathcal{C}} \text{ for every } q \in \tilde{Q}] \\ &\bigcup_{q \in \tilde{Q}} W_{I,q}^{\mathcal{C}}. \end{aligned} \quad \blacktriangleleft$$

Observe that, since the canonical RFA $\mathcal{C} = (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{I}, \tilde{F})$ for a language L is *strongly consistent*, the left language of each state is a principal of $\text{cl}_{\preceq_L^r}$. In particular, if the right language of a state is $u^{-1}L$ then its left language is the principal $\text{cl}_{\preceq_L^r}(u)$. Therefore, if $W_{I,q}^{\mathcal{N}} = \bigcup_{q \in \tilde{Q}} W_{I,q}^{\mathcal{C}}$ then $W_{I,q}^{\mathcal{N}}$ is a closed set in $\text{cl}_{\preceq_L^r}$. However, the reverse implication does not hold since only the *L-prime* principals are left languages of states of \mathcal{C} .

On the other hand, L -composite principals for \preceq_L^r can be described as intersections of L -prime principals [12]. As a consequence, $\text{Res}^r(\mathcal{N})$ is isomorphic to \mathcal{C} iff the left languages of states of \mathcal{N} are *union of non-empty intersections of left languages of \mathcal{C}* , while, as shown by Tamm [16], \mathcal{N}^{res} is isomorphic to \mathcal{C} iff the left languages of the states of \mathcal{N} are *union of left languages of \mathcal{C}* .

6 Learning Residual Automata

Bollig et al. [3] devised the NL* algorithm for learning the canonical RFA for a given regular language. The algorithm describes the behavior of a *Learner* that infers a language L by performing membership queries on L (which are answered by a *Teacher*) and equivalence queries between the language accepted by a candidate automaton and L (which are answered by an *Oracle*). The algorithm terminates when the *Learner* builds an RFA accepting the language L .

In this section we present a quasiorder-based perspective on the NL* algorithm in which the *Learner* iteratively refines a quasiorder \preceq on Σ^* by querying the *Teacher* and uses an adaption of the automata construction $H^r(\preceq, L)$ from Definition 3 to build an automaton that is used to query the *Oracle*. We capture this approach in the so-called *NL[≲] algorithm* whose pseudocode we defer to the extended version of this paper [12]. Here we give the definitions and general steps of the NL[≲] algorithm.

The *Learner* maintains a prefix-closed finite set $\mathcal{P} \subseteq \Sigma^*$ and a suffix-closed finite set $\mathcal{S} \subseteq \Sigma^*$. The set \mathcal{S} is used to *approximate* the principals in \preceq_L^r for the words in \mathcal{P} . In order to manipulate these approximations, we define the following two operators.

► **Definition 22.** *Let L be a language, $\mathcal{S} \subseteq \Sigma^*$ and $u, v \in \Sigma^*$. Then $u^{-1}L =_{\mathcal{S}} v^{-1}L \stackrel{\text{def}}{\iff} (u^{-1}L \cap \mathcal{S}) = (v^{-1}L \cap \mathcal{S})$. Similarly, $u^{-1}L \subseteq_{\mathcal{S}} v^{-1}L \stackrel{\text{def}}{\iff} (u^{-1}L \cap \mathcal{S}) \subseteq (v^{-1}L \cap \mathcal{S})$.*

These operators allow us to define a version of Nerode's quasiorder restricted to \mathcal{S} .

► **Definition 23** (Right-language-based quasiorder w.r.t. \mathcal{S}). *Let L be a language, $\mathcal{S} \subseteq \Sigma^*$ and $u, v \in \Sigma^*$. Define $u \preceq_{L\mathcal{S}}^r v \stackrel{\text{def}}{\iff} u^{-1}L \subseteq_{\mathcal{S}} v^{-1}L$.*

Recall that the *Learner* only manipulates the principals for the words in \mathcal{P} . Therefore, we need to adapt the notion of composite principal for $\preceq_{L\mathcal{S}}^r$.

► **Definition 24** ($L_{\mathcal{S}}$ -Composite Principal w.r.t. \mathcal{P}). *Let $\mathcal{P}, \mathcal{S} \subseteq \Sigma^*$ with $u \in \mathcal{P}$ and let $L \subseteq \Sigma^*$ be a language. We say that the principal $\text{cl}_{\preceq_{L\mathcal{S}}^r}(u)$ is $L_{\mathcal{S}}$ -composite w.r.t. \mathcal{P} iff $u^{-1}L =_{\mathcal{S}} \bigcup_{x \in \mathcal{P}, x \preceq_{L\mathcal{S}}^r u} x^{-1}L$. Otherwise, we say it is $L_{\mathcal{S}}$ -prime w.r.t. \mathcal{P} .*

The *Learner* uses the quasiorder $\preceq_{L\mathcal{S}}^r$ to build an automaton by adapting the construction from Definition 3 in order to use only the information that is available by means of the sets \mathcal{S} and \mathcal{P} . Building such an automaton requires the quasiorder to satisfy two conditions: it must be *closed* and *consistent* w.r.t. \mathcal{P} .

► **Definition 25** (Closedness and Consistency of $\preceq_{L\mathcal{S}}^r$ w.r.t. \mathcal{P}).

- (a) $\preceq_{L\mathcal{S}}^r$ is closed w.r.t. \mathcal{P} iff $\forall u \in \mathcal{P}, a \in \Sigma, \text{cl}_{\preceq_{L\mathcal{S}}^r}(ua)$ is $L_{\mathcal{S}}$ -prime w.r.t. $\mathcal{P} \Rightarrow \exists v \in \mathcal{P}, \text{cl}_{\preceq_{L\mathcal{S}}^r}(ua) = \text{cl}_{\preceq_{L\mathcal{S}}^r}(v)$.
- (b) $\preceq_{L\mathcal{S}}^r$ is consistent w.r.t. \mathcal{P} iff $\forall u, v \in \mathcal{P}, a \in \Sigma : u \preceq_{L\mathcal{S}}^r v \Rightarrow ua \preceq_{L\mathcal{S}}^r va$.

At each iteration, the *Learner* checks whether the quasiorder $\preceq_{L\mathcal{S}}^r$ is closed and consistent w.r.t. \mathcal{P} . If $\preceq_{L\mathcal{S}}^r$ is not closed w.r.t. \mathcal{P} , then it finds $\text{cl}_{\preceq_{L\mathcal{S}}^r}(ua)$ with $u \in \mathcal{P}, a \in \Sigma$ such that $\text{cl}_{\preceq_{L\mathcal{S}}^r}(ua)$ is $L_{\mathcal{S}}$ -prime w.r.t. \mathcal{P} and it is not equal to some $\text{cl}_{\preceq_{L\mathcal{S}}^r}(v)$ with $v \in \mathcal{P}$. Then the *Learner* adds ua to \mathcal{P} .

Similarly, if $\preceq_{L\mathcal{S}}^r$ is not consistent w.r.t. \mathcal{P} , the *Learner* finds $u, v \in \mathcal{P}, a \in \Sigma, x \in \mathcal{S}$ such that $u \preceq_{L\mathcal{S}}^r v$ but $uax \in L \wedge vax \notin L$. Then the *Learner* adds ax to \mathcal{S} . When the quasiorder $\preceq_{L\mathcal{S}}^r$ is closed and consistent w.r.t. \mathcal{P} , the *Learner* builds the automaton $R(\preceq_{L\mathcal{S}}^r, \mathcal{P})$.

Definition 26 is an adaptation of the automata construction H^r from Definition 3. Instead of considering all principals, it considers only those that correspond to words in \mathcal{P} . Moreover,

the notion of L -primality is replaced by L_S -primality w.r.t. \mathcal{P} , since the algorithm does not manipulate quotients of L by words in Σ^* but the approximation through \mathcal{S} of the quotients of L by words in \mathcal{P} (see Definition 22). Note that, if $\mathcal{S} = \mathcal{P} = \Sigma^*$ then $\text{Can}^r(L) = \mathcal{R}(\preceq_{L_S}^r, \mathcal{P})$.

► **Definition 26** (Automata construction $\mathcal{R}(\preceq_{L_S}^r, \mathcal{P})$). *Let $L \subseteq \Sigma^*$ be a language and let $\mathcal{P}, \mathcal{S} \subseteq \Sigma^*$. Define the automaton $\mathcal{R}(\preceq_{L_S}^r, \mathcal{P}) = (Q, \Sigma, \delta, I, F)$ with $Q = \{\text{cl}_{\preceq_{L_S}^r}(u) \mid u \in \mathcal{P}, \text{cl}_{\preceq_{L_S}^r}(u) \text{ is } L_S\text{-prime w.r.t. } \mathcal{P}\}$, $I = \{\text{cl}_{\preceq_{L_S}^r}(u) \in Q \mid \varepsilon \in \text{cl}_{\preceq_{L_S}^r}(u)\}$, $F = \{\text{cl}_{\preceq_{L_S}^r}(u) \in Q \mid u \in L\}$ and $\delta(\text{cl}_{\preceq_{L_S}^r}(u), a) = \{\text{cl}_{\preceq_{L_S}^r}(v) \in Q \mid \text{cl}_{\preceq_{L_S}^r}(u)a \subseteq \text{cl}_{\preceq_{L_S}^r}(v)\}$ for all $\text{cl}_{\preceq_{L_S}^r}(u) \in Q$ and $a \in \Sigma$.*

Finally, the *Learner* asks the *Oracle* whether $\mathcal{L}(\mathcal{R}(\preceq_{L_S}^r, \mathcal{P})) = L$. If the *Oracle* answers *yes* then the algorithm terminates. Otherwise, the *Oracle* returns a counterexample w for the language equivalence. Then, the *Learner* adds every suffix of w to \mathcal{S} and repeats the process.

Theorem 27 shows that the NL^{\preceq} algorithm exactly coincides with NL^* .

► **Theorem 27.** *NL^{\preceq} builds the same sets \mathcal{P} and \mathcal{S} , performs the same queries to the Oracle and the Teacher and returns the same RFA as NL^* , provided that both algorithms resolve nondeterminism the same way.*

It is worth to remark that, by replacing the right quasiorder $\preceq_{L_S}^r$ by the right congruence $\sim_{L_S} \stackrel{\text{def}}{=} \preceq_{L_S}^r \cap (\preceq_{L_S}^r)^{-1}$ in the above algorithm (precisely, in Definitions 25 and 26), the resulting algorithm corresponds to Angluin's L^* algorithm [2]. Note that, in that case, all principals $\text{cl}_{\sim_{L_S}}(u)$, with $u \in \Sigma^*$, are L_S -prime w.r.t. \mathcal{P} .

7 Related Work and Conclusions

Denis et al. [9] introduced the notion of RFA and canonical RFA for a language and devised a procedure, similar to the subset construction for DFAs, to build the RFA \mathcal{N}^{res} from a given automaton \mathcal{N} . Furthermore, they showed that \mathcal{N}^{res} is isomorphic to the canonical RFA \mathcal{C} for $\mathcal{L}(\mathcal{N})$ when \mathcal{N} is a co-RFA with no empty states. Later, Tamm [16] showed that \mathcal{N}^{res} is isomorphic to \mathcal{C} iff *the left language of every state of \mathcal{N} is a union of left languages of states of \mathcal{C}* . This result generalizes the double-reversal method for building the canonical RFA along the lines of the generalization by Brzozowski and Tamm [5] of the double-reversal method for DFAs, which claims that determinizing an automaton \mathcal{N} yields the minimal DFA iff *the left language of each state of \mathcal{N} is a union of co-atoms of $\mathcal{L}(\mathcal{N})$* . Although the two generalizations have a common foundation, the connection between the two results is not immediate.

Recently [11], we offered a congruence-based perspective of the generalized double-reversal method for DFAs and showed that determinizing an NFA, \mathcal{N} , yields the minimal DFA for $\mathcal{L}(\mathcal{N})$ iff $\text{cl}_{\sim_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$. In this paper we extend our previous work and devise quasiorder-based automata constructions that result in RFAs. One of these constructions, when instantiated with the automata-based quasiorder from Definition 10, defines a residualization operation that, given an NFA \mathcal{N} , produces the RFA $\text{Res}^r(\mathcal{N})$ with, at most, as many states as \mathcal{N}^{res} , the residualization operation defined by Denis et al. [9]. Observe that if \mathcal{N} is a co-RFA with no empty states then both \mathcal{N}^{res} and $\text{Res}^r(\mathcal{N})$ are isomorphic to \mathcal{C} .

On the other hand, Theorem 19 shows that $\text{Res}^r(\mathcal{N})$ is isomorphic to \mathcal{C} iff $\text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$. We believe that the similarity between the generalizations of the double-reversal methods for DFAs ($\text{cl}_{\sim_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$) and for RFAs ($\text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$) evidences that quasiorders are for RFAs as congruences are for DFAs. Indeed, determinizing an NFA \mathcal{N} with $L = \mathcal{L}(\mathcal{N})$ yields the minimal DFA for L iff $\sim_{\mathcal{N}}^r = \sim_L^r$ [11] and, similarly, when residualizing \mathcal{N} with our residualization operation we obtain the canonical RFA for L iff $\preceq_{\mathcal{N}}^r = \preceq_L^r$, as shown by Lemma 16.

<p>Brzozowski and Tamm [5]</p> $\mathcal{N}^D \equiv \mathcal{M}$ <p>iff</p> <p>$\forall q, W_{I,q}^{\mathcal{N}}$ is a union of co-atoms</p>	<p>Ganty et al. [11]</p> $\mathcal{N}^D \equiv \mathcal{M}$ <p>iff</p> <p>$\forall q, \text{cl}_{\sim_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$</p>	<p>In the diagram: \mathcal{N} is an NFA with $L = \mathcal{L}(\mathcal{N})$; \mathcal{N}^D is the result of determinizing \mathcal{N} with the standard subset construction; \mathcal{M} is the minimal DFA for L; $\mathcal{C} = \text{Can}^r(L)$ is the canonical RFA for L and $\mathcal{N}_1 \equiv \mathcal{N}_2$ denotes that automaton \mathcal{N}_1 is isomorphic to \mathcal{N}_2.</p>
<p>Tamm [16]</p> $\mathcal{N}^{\text{res}} \equiv \mathcal{C}$ <p>iff</p> <p>$\forall q, W_{I,q}^{\mathcal{N}}$ is a union of $W_{I,q'}^{\mathcal{C}}$</p>	<p>Theorem 19</p> $\text{Res}^r(\mathcal{N}) \equiv \mathcal{C}$ <p>iff</p> <p>$\forall q, \text{cl}_{\preceq_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$</p>	

■ **Figure 3** Summary of the existing results about the generalized double-reversal method for building the minimal DFA (first row) and the canonical RFA (second row) for a given language. The results on the first column are based on the notion of *atoms* of a language while the results on the second column are based on *quasiorders*.

It is worth to remark that the left languages of the minimal DFA for L are principals of \sim_L^r [11]. Therefore, the condition $\text{cl}_{\sim_L^r}(W_{I,q}^{\mathcal{N}}) = W_{I,q}^{\mathcal{N}}$, which guarantees that determinizing \mathcal{N} yields the minimal DFA, can be stated as: *the left language of each state of \mathcal{N} is a union of left languages of states of the minimal DFA*. Thus, this characterization is the DFA-equivalent of Tamm's condition [16] for RFAs.

Figure 3 summarizes the existing results about these double-reversal methods.

Moreover, we support the idea that quasiorders are natural to residual automata by observing that the NL^* algorithm can be interpreted as an algorithm that, at each iteration, refines an approximation of the Nerode's quasiorder and builds an RFA using our automata construction.

Finally, it is worth to mention that Myers et al. [15] describe different canonical non-determinism automata constructions for a given regular language and show how to obtain the canonical RFA. They do it by first constructing the minimal DFA for the language interpreted in a variety of join-semilattices and then applying a dual equivalence between this variety and the category of closure spaces. In some sense, this already establishes a connection between the class of DFAs and RFAs. Indeed, the same authors [1] use this category-theoretical perspective to address the residual-equivalent of the double-reversal method proposed by Denis et al. [9]. In contrast, this work revisits different methods to construct the canonical RFA relying on the simple notion of quasiorders on words, as a natural extension of our work on congruences for the study of minimization techniques for DFAs.

References

- 1 Jirí Adámek, Robert S. R. Myers, Henning Urbat, and Stefan Milius. On continuous non-determinism and state minimality. In *MFPS*, volume 308 of *Electronic Notes in Theoretical Computer Science*, pages 3–23. Elsevier, 2014.
- 2 Dana Angluin. Learning regular sets from queries and counterexamples. *Inf. Comput.*, 75(2):87–106, 1987.
- 3 Benedikt Bollig, Peter Habermehl, Carsten Kern, and Martin Leucker. Angluin-style learning of NFA. In *IJCAI*, pages 1004–1009, 2009.
- 4 Janusz A. Brzozowski. Canonical regular expressions and minimal state graphs for definite events. *Mathematical Theory of Automata*, 12(6):529–561, 1962.
- 5 Janusz A. Brzozowski and Hellis Tamm. Theory of átomata. *Theor. Comput. Sci.*, 539:13–27, 2014.

- 6 Aldo de Luca and Stefano Varricchio. Well quasi-orders and regular languages. *Acta Inf.*, 31(6):539–557, 1994.
- 7 Aldo de Luca and Stefano Varricchio. *Finiteness and Regularity in Semigroups and Formal Languages*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 1999.
- 8 François Denis, Aurélien Lemay, and Alain Terlutte. Learning regular languages using non deterministic finite automata. In *ICGI*, volume 1891 of *Lecture Notes in Computer Science*, pages 39–50. Springer, 2000.
- 9 François Denis, Aurélien Lemay, and Alain Terlutte. Residual finite state automata. *Fundam. Inform.*, 51(4):339–368, 2002.
- 10 François Denis, Aurélien Lemay, and Alain Terlutte. Learning regular languages using RFSAs. *Theor. Comput. Sci.*, 313(2):267–294, 2004.
- 11 Pierre Ganty, Elena Gutiérrez, and Pedro Valero. A congruence-based perspective on automata minimization algorithms. In *MFCs*, volume 138 of *LIPICs*, pages 77:1–77:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- 12 Pierre Ganty, Elena Gutiérrez, and Pedro Valero. A quasiorder-based perspective on residual automata, 2020. [arXiv:2007.00359](https://arxiv.org/abs/2007.00359).
- 13 Pierre Ganty, Francesco Ranzato, and Pedro Valero. Language inclusion algorithms as complete abstract interpretations. In *SAS*, volume 11822 of *Lecture Notes in Computer Science*, pages 140–161. Springer, 2019.
- 14 Anna Kasprzik. Inference of residual finite-state tree automata from membership queries and finite positive data. In *Developments in Language Theory*, volume 6795 of *Lecture Notes in Computer Science*, pages 476–477. Springer, 2011.
- 15 Robert S. R. Myers, Jirí Adámek, Stefan Milius, and Henning Urbat. Coalgebraic constructions of canonical nondeterministic automata. *Theor. Comput. Sci.*, 604:81–101, 2015.
- 16 Hellis Tamm. Generalization of the double-reversal method of finding a canonical residual finite state automaton. In *DCFS*, volume 9118 of *Lecture Notes in Computer Science*, pages 268–279. Springer, 2015.