# Complete Abstractions for Checking Language Inclusion

Pierre Ganty*, Francesco Ranzato† and Pedro Valero§*
*IMDEA Software Institute, Madrid, Spain. Email: pierre.ganty@imdea.org
†University of Padova, Italy. Email: ranzato@math.unipd.it
§Universidad Politécnica de Madrid, Spain. Email: pedro.valero@imdea.org



***Abstract*—We study the language inclusion problem $L_1 \subseteq L_2$ where $L_1$ is regular or context-free. Our approach relies on abstract interpretation and checks whether an overapproximating abstraction of $L_1$, obtained by successively overapproximating the Kleene iterates of its least fixpoint characterization, is included in $L_2$. We show that a language inclusion problem is decidable whenever this overapproximating abstraction satisfies a completeness condition (i.e. its loss of precision causes no false alarm) and prevents infinite ascending chains (i.e. it guarantees termination of least fixpoint computations). Such overapproximating abstraction function on languages can be defined using quasiorder relations on words where the abstraction gives the language of all words "greater than or equal to" a given input word for that quasiorder. We put forward a range of quasiorders that allow us to systematically design decision procedures for different language inclusion problems such as context-free languages into regular languages and regular languages into trace sets of one-counter nets. We also provide quasiorders for which the induced inclusion checking procedure corresponds to well-known state-of-the-art algorithms like the so-called antichain algorithms. Finally, we provide an equivalent greatest fixpoint language inclusion check which relies on quotients of languages and, to the best of our knowledge, was not previously known.**

## 1. Introduction

Language inclusion is a fundamental and classical problem which consists in deciding, given two languages $L_1$ and $L_2$, whether $L_1 \subseteq L_2$ holds. We consider languages of finite words over a finite alphabet $\Sigma$.

The basic idea of our approach for solving a language inclusion problem $L_1 \subseteq L_2$ is to leverage Cousot and Cousot's abstract interpretation [9], [10] for checking the inclusion of an overapproximation (i.e. a superset) of $L_1$ into $L_2$. Assuming that $L_1$ is specified as least fixpoint of an equation system on $\wp(\Sigma^*)$, an approximation of $L_1$ is obtained by applying an overapproximating language abstraction function $\rho : \wp(\Sigma^*) \to \wp(\Sigma^*)$ at each step of the Kleene iterates converging to the least fixpoint. This $\rho$ is an upper closure operator which is used in standard abstract interpretation for approximating an input language by adding several words (possibly none) to it. This abstract interpretation-based approach provides an abstract inclusion check $\rho(L_1) \subseteq L_2$ which is always sound by construction. We then give conditions on $\rho$ which ensure a *complete* abstract inclusion check, namely the answer to $\rho(L_1) \subseteq L_2$ is always exact (no false alarms in abstract interpretation terminology): (i) $\rho(L_2) = L_2$; (ii) $\rho$ is a complete abstraction for symbol concatenation $aX$, for all $a \in \Sigma$, according to the standard notion of completeness in abstract interpretation [9], [18]. This approach leads us to design in Section 4 a generic algorithmic framework for language inclusion problems which is parameterized by an underlying language abstraction (cf. Theorem 4.5).

We then focus on overapproximating abstractions $\rho$ which are induced by a quasiorder relation $\leqslant$ on words in $\Sigma^*$. Here, a language $L$ is overapproximated by adding all the words which are "greater than or equal to" some word of $L$ for $\leqslant$. This allows us to instantiate the above conditions (i) and (ii) for having a complete abstract inclusion check in terms of the quasiorder $\leqslant$. Termination, which corresponds to having finitely many Kleene iterates in the fixpoint computations, is guaranteed by requiring that the relation $\leqslant$ is a well-quasiorder.

We define quasiorders satisfying the above conditions which are directly derived from the standard Myhill and Nerode equivalence relations on words. These quasiorders have been first investigated by Ehrenfeucht et al. [15] and have been later generalized and extended by de Luca and Varricchio [11], [12]. In particular, drawing from a result by de Luca and Varricchio [11], we show that the language abstractions induced by these Myhill and Nerode quasiorders are the most general ones which fit in our algorithmic framework for checking language inclusion. While Myhill and Nerode quasiorder abstractions do not depend on some language representation (e.g., some class of automata or grammars), we provide quasiorders which instead exploit an underlying language representation given by a finite automaton. In particular, by selecting suitable well-quasiorders for the class of language inclusion problems at hand we are able to systematically derive decision procedures for a number of different inclusion problems $L_1 \subseteq L_2$: (i) both $L_1$ and $L_2$ and regular; (ii) $L_1$ is context-free and $L_2$ is regular; (iii) $L_1$ is regular and $L_2$ is the trace language of a one-counter net.

These decision procedures that we systematically derive here by instantiating our framework are then related to existing language inclusion checking algorithms. We study in detail the case where both languages $L_1$ and $L_2$ are regular

and represented by finite state automata. When our decision procedure for $L_1 \subseteq L_2$ is derived from a well-quasiorder on $\Sigma^*$ exploiting the automaton-based representation of $L_2$ it turns out that we obtain the well-known antichain algorithm by De Wulf et al. [13]. Also, it turns out that by including a simulation relation in the definition of the well-quasiorder we derive a decision procedure that partially matches the inclusion algorithm by Abdulla et al. [2], hence also that by Bonchi and Pous [5]. Moreover, we systematically derive an antichain algorithm for the case where $L_1$ is represented by a context-free grammar and $L_2$ is represented by a finite state automaton. In this case, the resulting decision procedure closely resembles the antichain algorithm by Holík and Meyer [22]. A similar phenomenon happens for the inclusion problem of a regular language into the set of traces of a one-counter net: in this case the decision procedure that we systematically derive matches the algorithm by Hofman and Chen [20].

Finally, we leverage a standard duality result in abstract fixpoint checking [8] and put forward a greatest fixpoint approach (instead of the above least fixpoint approach) for the case where $L_1$ is represented by a *linear* context-free grammar and $L_2$ is regular. In this case, we exploit the properties of the overapproximating abstraction induced by the quasiorder in order to show that the Kleene iterates of this greatest fixpoint computation are finitely many. Interestingly, the Kleene iterates of the greatest fixpoint are finitely many whether you apply the overapproximating abstraction or not, a known phenomenon happening for so-called forward complete abstract interpretations [17].

## 2. Background

**Order Theory Basics.** $\langle D, \leqslant \rangle$ is a *quasiordered set* (qoset) when $\leqslant$ is a quasiorder relation on $D$, that is, reflexive and transitive. A qoset $\langle D, \leqslant \rangle$ satisfies the *ascending* (resp. *descending*) *chain condition* (ACC, resp. DCC) if there is no countably infinite sequence of distinct elements $\{x_i\}_{i \in \mathbb{N}}$ such that, for all $i \in \mathbb{N}$, $x_i \leqslant x_{i+1}$ (resp. $x_{i+1} \leqslant x_i$). A qoset is called ACC (DCC) when it satisfies the ACC (DCC).

A qoset $\langle D, \leqslant \rangle$ is a *partially ordered set* (poset) when $\leqslant$ is antisymmetric. A subset of a poset is *directed* if it is nonempty and every pair of elements has an upper bound in it. A poset $\langle D, \leqslant \rangle$ is a *directed-complete partial order* (CPO) if it has the least upper bound (lub) of all its directed subsets. A poset if a *join-semilattice* if it has the lub of all its nonempty finite subsets (so that binary lubs are enough). A poset is a *complete lattice* if it has the lub of all its arbitrary (possibly empty) subsets (so that it also has the greatest lower bound (glb) of all its arbitrary subsets).

A qoset $\langle D, \leqslant \rangle$ is a *well-quasiordered set* (wqoset) when for every countably infinite sequence of elements $\{x_i\}_{i \in \mathbb{N}}$ there exist $i, j \in \mathbb{N}$ such that $i < j$ and $x_i \leqslant x_j$. For every qoset $\langle D, \leqslant \rangle$ we define the following relation between subsets $X, Y \subseteq D$:

$$X \sqsubseteq Y \overset{\triangle}{\Longleftrightarrow} \forall x \in X, \exists y \in Y,\ y \leqslant x.$$

A *minor* of a set $X \subseteq D$, denoted by $\lfloor X \rfloor$, is a subset of $X$ satisfying: (i) $X \sqsubseteq \lfloor X \rfloor$ and (ii) $\lfloor X \rfloor$ is an *antichain*, that is, $x_1 \leqslant x_2$ for no $x_1, x_2 \in \lfloor X \rfloor$. Let us recall that every subset of a wqoset $\langle D, \leqslant \rangle$ has at least one minor set, all minor sets are finite and if $\langle D, \leqslant \rangle$ is additionally a poset then there exists exactly one minor set. We denote the set of antichains of $\langle D, \leqslant \rangle$ by $\mathrm{AC}_{\langle D, \leqslant \rangle} \triangleq \{X \subseteq D \mid X \text{ is an antichain}\}$. It turns out that $\langle \mathrm{AC}_{\langle D, \leqslant \rangle}, \sqsubseteq \rangle$ is a qoset, it is ACC if $\langle D, \leqslant \rangle$ is a wqoset and it is a poset if $\langle D, \leqslant \rangle$ is a poset.

**Kleene Iterates.** Let $\langle X, \leqslant \rangle$ be a qoset, $f : X \to X$ be a function and $b \in X$. Then, the trace of values of the variable $x \in X$ computed by the following iterative procedure:

$$\texttt{Kleene}(f, b) \triangleq \begin{cases} x := b; \\ \textbf{while } f(x) \leqslant x \textbf{ do } x := f(x); \\ \textbf{return } x; \end{cases}$$

provides the possibly infinite sequence of so-called Kleene iterates of the function $f$ starting from the basis $b$. When $\langle X, \leqslant \rangle$ is a ACC CPO, $b \leqslant f(b)$ and $f$ is monotonic then $\texttt{Kleene}(f, b)$ terminates and returns the least fixpoint of the function $f$ which is greater than or equal to $b$.

Let us also recall that given a monotonic function $f \colon C \to C$ on a complete lattice $C$, its least and greatest fixpoints always exist, and we denote them, resp., by $\mathrm{lfp}(f)$ and $\mathrm{gfp}(f)$.

For the sake of clarity, we overload the notation and use the same symbol for an operator/relation and its componentwise (i.e. pointwise) extension on product domains. A vector $\vec{\boldsymbol{Y}}$ in some product domain $D^{|S|}$ might be also denoted by $\langle Y_i \rangle_{i \in S}$. In such case, $\vec{\boldsymbol{Y}}_q$ denotes its component $Y_q$.

**Language Theory Basics.** Let $\Sigma$ be an *alphabet* (that is, a finite nonempty set of symbols). Concatenation in $\Sigma^*$ is simply denoted by juxtaposition, both for concatenating words $uv$, languages $L_1 L_2$ and words with languages, e.g. $uL$ and $uLv$. We sometimes use the symbol $\cdot$ to refer explicitly to the concatenation operation.

A *finite automaton* (FA) is a tuple $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$ where $\Sigma$ is the *alphabet*, $Q$ is the finite set of *states*, $I \subseteq Q$ are the *initial states*, $F \subseteq Q$ are the *final states*, and $\delta \colon Q \times \Sigma \to \wp(Q)$ is the *transition relation*, where $q \xrightarrow{a} q'$ denotes that $q' \in \delta(q, a)$. If $u \in \Sigma^*$ and $q, q' \in Q$ then $q \overset{u}{\rightsquigarrow} q'$ means that the state $q'$ is reachable from $q$ by following the string $u$. Therefore, $q \overset{\epsilon}{\rightsquigarrow} q'$ holds iff $q = q'$. The *language generated by an FA* $\mathcal{A}$, denoted $\mathcal{L}(\mathcal{A})$, is given by $\mathcal{L}(\mathcal{A}) \triangleq \{u \in \Sigma^* \mid \exists q_i \in I, \exists q_f \in F,\ q_i \overset{u}{\rightsquigarrow} q_f\}$.

## 3. Inclusion Check by Complete Abstractions

The *language inclusion problem* consists in checking whether $L_1 \subseteq L_2$ holds where $L_1$ and $L_2$ are two languages over an alphabet $\Sigma$. In this section, we show how backward complete abstractions $\rho$ can be used to compute $\rho(L_1)$, an overapproximation of $L_1$ such that $\rho(L_1) \subseteq L_2 \Leftrightarrow L_1 \subseteq L_2$.

Let $\mathrm{uco}(C)$ denote the set of upper closure operators (or simply closure operators) on a poset $\langle C, \leq_C \rangle$, that is,

the set of monotonic, idempotent (i.e., $\rho(x) = \rho(\rho(x))$) and increasing (i.e., $x \leq_C \rho(x)$) functions in $C \to C$. We often write $c \in \rho(C)$ (or simply $c \in \rho$ when $C$ is clear from the context) to denote that there exists $c' \in C$ with $c = \rho(c')$. Recall that this happens iff $\rho(c) = c$. More details about closure operators can be found on Appendix A.

Closure-based abstract interpretation [10] can be applied to solve a generic inclusion checking problem stated through least fixpoints as follows. Let $\rho \in \mathrm{uco}(C)$ and $c_2 \in C$ such that $c_2 \in \rho$. Then, for all $c_1 \in C$, it turns out that

$$c_1 \leq_C c_2 \Leftrightarrow \rho(c_1) \leq_C \rho(c_2) \Leftrightarrow \rho(c_1) \leq_C c_2 \qquad (1)$$

We apply here the standard notion of backward completeness in abstract interpretation [9], [10], [18]. In abstract interpretation a closure operator $\rho \in \mathrm{uco}(C)$ on a concrete domain $C$ plays the role of abstraction function for objects of $C$. A closure $\rho \in \mathrm{uco}(C)$ is called backward complete for a concrete monotonic function $f : C \to C$ when $\rho f = \rho f \rho$ holds. The intuition is that backward completeness models an ideal situation where no loss of precision is accumulated in the computations of $\rho f$ when its concrete input objects are approximated by $\rho$. It is well known that in this case backward completeness implies completeness of least fixpoints, namely, $\rho(\mathrm{lfp}(f)) = \mathrm{lfp}(\rho f) = \mathrm{lfp}(\rho f \rho)$ holds by assuming that the these least fixpoints exist (this is the case, e.g., when $C$ is a CPO). Theorem 3.1 shows that in order to check an inclusion $c_1 \leq_C c_2$ for some $c_1 = \mathrm{lfp}(f)$ and $c_2 \in \rho$, it is enough to perform an inclusion check $\mathrm{lfp}(\rho f) \leq_C \rho(c_2)$ which works on the abstraction $\rho(C)$.

**Theorem 3.1.** *If $C$ is a CPO, $f : C \to C$ is monotonic, $\rho$ is backward complete for $f$ and $c_2 \in \rho$, then $\mathrm{lfp}(f) \leq_C c_2 \Leftrightarrow \mathrm{lfp}(\rho f) \leq_C c_2$. In particular, if $\langle \rho, \leq_C \rangle$ is ACC then the Kleene iterates of $\mathrm{lfp}(\rho f)$ are finitely many.*

In the following sections we apply this general abstraction technique for a number of different language inclusion problems, by designing decision algorithms which rely on specific backward complete abstractions of $\wp(\Sigma^*)$.

## 4. An Algorithmic Framework for Language Inclusion

### 4.1. Languages as Fixed Points

Let $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$ be a FA. Given $S, T \subseteq Q$, define

$$W^{\mathcal{A}}_{S,T} \triangleq \{u \in \Sigma^* \mid \exists q \in S,\ \exists q' \in T \colon q \overset{u}{\rightsquigarrow} q'\} \ .$$

When $S = \{q\}$ or $T = \{q'\}$ we abuse the notation and write $W^{\mathcal{A}}_{q,T}$, $W^{\mathcal{A}}_{S,q'}$, or $W^{\mathcal{A}}_{q,q'}$. Also, we omit the automaton $\mathcal{A}$ in superscripts when this is clear from the context. The language accepted by $\mathcal{A}$ is $\mathcal{L}(\mathcal{A}) \triangleq W^{\mathcal{A}}_{I,F}$. Observe that[1]

$$\mathcal{L}(\mathcal{A}) = \bigcup_{q \in I} W^{\mathcal{A}}_{q,F} = \bigcup_{q \in F} W^{\mathcal{A}}_{I,q} \ . \qquad (2)$$

Let us recall how to define the language accepted by an automaton as a solution of a set of equations (see, e.g., [27, Section I.2.4.3]). Given a Boolean predicate $p(x)$ (typically a membership predicate) and two sets $T$ and $F$, let us define

$$\delta^T_F(p(x)) \triangleq \begin{cases} T & \text{if } p(x) \text{ holds} \\ F & \text{otherwise} \end{cases} \ .$$

The FA $\mathcal{A}$ induces the following set of equations:

$$\mathrm{Eqn}(\mathcal{A}) \triangleq \{X_q = \delta^{\{\epsilon\}}_{\varnothing}(q \in F) \cup \textstyle\bigcup_{a \in \Sigma, q \xrightarrow{a} q'} a X_{q'} \mid q \in Q\}$$

where $X_q \in \wp(\Sigma^*)$, so that the functions in the right-hand sides of $\mathrm{Eqn}(\mathcal{A})$ have type $\wp(\Sigma^*)^{|Q|} \to \wp(\Sigma^*)$. Since $\langle \wp(\Sigma^*)^{|Q|}, \subseteq \rangle$ is a (product) complete lattice (as $\langle \wp(\Sigma^*), \subseteq \rangle$ is a complete lattice) and all the functions in $\mathrm{Eqn}(\mathcal{A})$ are monotonic, the least solution $\langle Y_q \rangle_{q \in Q}$ of $\mathrm{Eqn}(\mathcal{A})$ does exist. It is easy to check that $Y_q = W^{\mathcal{A}}_{q,F}$ for every $q \in Q$.

Note that, by concatenating on the right, one can define an equivalent set of equations whose least solution coincides with $W^{\mathcal{A}}_{I,q}$ instead of $W^{\mathcal{A}}_{q,F}$ (see Appendix B).

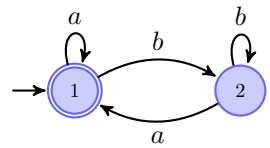


Figure 1. Finite automaton $\mathcal{A}$ over alphabet $\Sigma = \{a, b\}$ with states $\{1, 2\}$ such that $\mathcal{L}(\mathcal{A}) = (a + (b^+ a))^*$.

**Example 4.1.** Let us consider the automaton $\mathcal{A}$ in Figure 1. The set of equations induced by $\mathcal{A}$ are as follows:

$$\mathrm{Eqn}(\mathcal{A}) = \begin{cases} X_1 = \{\epsilon\} \cup a X_1 \cup b X_2 \\ X_2 = \varnothing \cup a X_1 \cup b X_2 \end{cases} \qquad \Diamond$$

We define the vector $\vec{\boldsymbol{\epsilon}}^F \in \wp(\Sigma^*)^{|Q|}$ and the function $\mathrm{Pre}_{\mathcal{A}} \colon \wp(\Sigma^*)^{|Q|} \to \wp(\Sigma^*)^{|Q|}$, which are used to formalize the equations in $\mathrm{Eqn}(\mathcal{A})$:

$$\vec{\boldsymbol{\epsilon}}^F \triangleq \langle \delta^{\{\epsilon\}}_{\varnothing}(q \in F) \rangle_{q \in Q}$$
$$\mathrm{Pre}_{\mathcal{A}}(\langle X_q \rangle_{q \in Q}) \triangleq \langle \textstyle\bigcup_{a \in \Sigma, q \xrightarrow{a} q'} a X_{q'} \rangle_{q \in Q} \ .$$

Since $\epsilon \in W^{\mathcal{A}}_{q,F}$ for all $q \in F$, we initialize the fixpoint computation with $\vec{\boldsymbol{\epsilon}}^F$. Thus, it turns out that

$$\langle W^{\mathcal{A}}_{q,F} \rangle_{q \in Q} = \mathrm{lfp}(\lambda \vec{\boldsymbol{X}}.\ \vec{\boldsymbol{\epsilon}}^F \cup \mathrm{Pre}_{\mathcal{A}}(\vec{\boldsymbol{X}})) \ . \qquad (3)$$

Together with Equation (2), it follows that $\mathcal{L}(\mathcal{A})$ equals the union of the component languages of the vector $\mathrm{lfp}(\lambda \vec{\boldsymbol{X}}.\ \vec{\boldsymbol{\epsilon}}^F \cup \mathrm{Pre}_{\mathcal{A}}(\vec{\boldsymbol{X}}))$ indexed by initial states.

**Example 4.2** (Continuation of Example 4.1)**.** The fixpoint characterization of $\langle W^{\mathcal{A}}_{q,F} \rangle_{q \in Q}$ is:

$$\begin{pmatrix} W^{\mathcal{A}}_{q_1,q_1} \\ W^{\mathcal{A}}_{q_2,q_1} \end{pmatrix} = \mathrm{lfp}\left(\lambda \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}. \begin{pmatrix} \{\epsilon\} \cup a X_1 \cup b X_2 \\ \varnothing \cup a X_1 \cup b X_2 \end{pmatrix}\right) \ .$$

The fixpoint is $\begin{pmatrix} W^{\mathcal{A}}_{q_1,q_1} \\ W^{\mathcal{A}}_{q_2,q_1} \end{pmatrix} = \begin{pmatrix} (a + (b^+ a))^* \\ (a + b)^* a \end{pmatrix}$. $\Diamond$

1. Note that, as usual, $\bigcup \varnothing = \varnothing$.

**Fixpoint-based Inclusion Check.** Consider the language inclusion problem $L_1 \subseteq L_2$ where $L_1 = \mathcal{L}(\mathcal{A})$ for some FA $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$. The language $L_2$ can be formalized as a vector in $\wp(\Sigma^*)^{|Q|}$ as follows:

$$\overrightarrow{\boldsymbol{L_2}}^I \triangleq \langle \delta^{L_2}_{\Sigma^*}(q \in I) \rangle_{q \in Q}. \tag{4}$$

Using (2), (3) and (4), it is routine to prove that

$$\mathcal{L}(\mathcal{A}) \subseteq L_2 \Leftrightarrow \mathrm{lfp}(\lambda \overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{\epsilon}}^F \cup \mathrm{Pre}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}})) \subseteq \overrightarrow{\boldsymbol{L_2}}^I \ . \tag{5}$$

## 4.2. Abstract Inclusion Check: Closures

In what follows we use Theorem 3.1 for solving the language inclusion problem. In this context, we have that $C = \langle \wp(\Sigma^*)^{|Q|}, \subseteq \rangle$, $f = \lambda \overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{\epsilon}}^F \cup \mathrm{Pre}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}})$ and $\rho : \wp(\Sigma^*)^{|Q|} \to \wp(\Sigma^*)^{|Q|}$ is an upper closure operator.

**Theorem 4.3.** *If $\rho \in \mathrm{uco}(\wp(\Sigma^*))$ is backward complete for $\lambda X.\, aX$ for all $a \in \Sigma$, then, for all FAs $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$, $\rho$ is backward complete for $\mathrm{Pre}_{\mathcal{A}}$ and $\lambda \overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{\epsilon}}^F \cup \mathrm{Pre}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}})$.*

**Corollary 4.4.** *If $\rho \in \mathrm{uco}(\wp(\Sigma^*))$ is backward complete for $\lambda X.\, aX$ for all $a \in \Sigma$ then*

$$\rho(\mathrm{lfp}(\lambda \overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{\epsilon}}^F \cup \mathrm{Pre}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}}))) = \mathrm{lfp}(\lambda \overrightarrow{\boldsymbol{X}}.\, \rho(\overrightarrow{\boldsymbol{\epsilon}}^F \cup \mathrm{Pre}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}}))).$$

Note that if $\rho$ is backward complete for $\lambda X.aX$ for all $a \in \Sigma$ and $L_2 \in \rho$ then, as a straightforward consequence of Theorem 3.1 and Corollary 4.3, Equation 5 becomes

$$\mathcal{L}(\mathcal{A}) \subseteq L_2 \Leftrightarrow \mathrm{lfp}(\lambda \overrightarrow{\boldsymbol{X}}.\, \rho(\overrightarrow{\boldsymbol{\epsilon}}^F \cup \mathrm{Pre}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}}))) \subseteq \overrightarrow{\boldsymbol{L_2}}^I \ . \tag{6}$$

## 4.3. Abstract Inclusion Check: Galois Connections

To solve a language inclusion problem $\mathcal{L}(\mathcal{A}) \subseteq L_2$ using Equation (6) we must compute the corresponding least fixpoint and then decide its inclusion in $\overrightarrow{\boldsymbol{L_2}}^I$. Since closure operators are fully isomorphic to Galois connections [10, Section 6], they allow us to conveniently define and reason on abstract domains independently of their representation. Recall that a *Galois Connection* (GC) between two posets $\langle C, \leq_C \rangle$ (called concrete domain) and $\langle A, \leq_A \rangle$ (called abstract domain) consists of two functions $\alpha \colon C \to A$ and $\gamma \colon A \to C$ such that $\alpha(c) \leq_A a \Leftrightarrow c \leq_C \gamma(a)$ always holds. A GC is denoted by $\langle C, \leq_C \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \leq_A \rangle$.

The next result shows that there exists an algorithm that solves the language inclusion problem $\mathcal{L}(\mathcal{A}) \subseteq L_2$ on an abstraction $D$ of the concrete domain of languages $\wp(\Sigma^*)$ whenever $D$ satisfies a list of requirements related to backward completeness and computability.

**Theorem 4.5.** *Let $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$ be a FA and let $L_2$ be a language over $\Sigma$. Let $\langle \wp(\Sigma^*), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle D, \sqsubseteq \rangle$ be a GC where $\langle D, \sqsubseteq \rangle$ is a poset. Assume that the following properties hold:*

*(i) $L_2 \in \gamma(D)$ and for every $a \in \Sigma$, $X \in \wp(\Sigma^*)$ we have $\alpha(aX) = \alpha(a\gamma\alpha(X))$.*

*(ii) $(D, \sqsubseteq, \sqcup)$ is an effective domain, meaning that: $(D, \sqsubseteq)$ is ACC, every element of $D$ has a finite representation, $\sqsubseteq$ is decidable and $\sqcup$ is a computable binary lub.*

*(iii) There is an algorithm, say $\mathrm{Pre}^\sharp(\overrightarrow{\boldsymbol{X}})$, which computes $\alpha(\mathrm{Pre}_{\mathcal{A}}(\gamma(\overrightarrow{\boldsymbol{X}})))$, for all $\overrightarrow{\boldsymbol{X}} \in \wp(\Sigma^*)^{|Q|}$.*

*(iv) There is an algorithm, say $\epsilon^\sharp$, computing $\alpha(\overrightarrow{\boldsymbol{\epsilon}}^F)$.*

*(v) There is an algorithm, say $\mathrm{Incl}^\sharp(\overrightarrow{\boldsymbol{X}})$, deciding the abstract inclusion $\overrightarrow{\boldsymbol{X}} \sqsubseteq \alpha(\overrightarrow{\boldsymbol{L_2}}^I)$, for every vector $\overrightarrow{\boldsymbol{X}} \in \alpha(\wp(\Sigma^*)^{|Q|})$.*

*Then, the following algorithm decides whether $\mathcal{L}(\mathcal{A}) \subseteq L_2$:*

$\langle Y_q \rangle_{q \in Q} := \mathtt{Kleene}(\lambda \overrightarrow{\boldsymbol{X}}.\, \epsilon^\sharp \sqcup \mathrm{Pre}^\sharp(\overrightarrow{\boldsymbol{X}}), \overrightarrow{\boldsymbol{\varnothing}})$;
**return** $\mathrm{Incl}^\sharp(\langle Y_q \rangle_{q \in Q})$;

**Quasiorder Galois Connections.** It turns out that Theorem 4.5 still holds for abstract domains which are mere qosets rather than posets.

**Definition 4.6** (**Quasiorder GC**). A *quasiorder GC* (QGC) $\langle C, \leq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle D, \sqsubseteq \rangle$ consists of: (a) two qosets $\langle C, \leq \rangle$ and $\langle D, \sqsubseteq \rangle$ such that one of them is a poset; (b) two functions $\alpha \colon C \to D$ and $\gamma \colon D \to C$ such that $\alpha(c) \sqsubseteq d \Leftrightarrow c \leq \gamma(d)$ holds for all $c \in C$ and $d \in D$. ■

Analogously to GCs, it is easily seen that in QGCs both $\alpha$ and $\gamma$ are monotonic as well as $c \leq \gamma(\alpha(c))$ and $\alpha(\gamma(d)) \sqsubseteq d$ always hold. Observe that if $C$ is a poset and $d \sqsubseteq d' \sqsubseteq d$ with $d \neq d'$ then $\gamma(d) = \gamma(d')$, because $\gamma$ is monotonic, and conversely, if $D$ is a poset and $c \leq c' \leq c$ with $c \neq c'$ then $\alpha(c) = \alpha(c')$ holds. Also, similarly to GCs, if $C$ is a poset then $\gamma \circ \alpha \in \mathrm{uco}(\langle C, \leq \rangle)$ holds for QGCs as well.

In the following, we apply all the standard order-theoretic notions used for posets also to the qosets $\langle C, \leq \rangle$ and $\langle D, \sqsubseteq \rangle$ by implicitly referring to the quotient posets $\langle C_{/\cong_C}, \leq_{/\cong_C} \rangle$ and $\langle D_{/\cong_D}, \sqsubseteq_{/\cong_D} \rangle$ where $\cong_C \triangleq \leq \cap \leq^{-1}$ and $\cong_D \triangleq \sqsubseteq \cap \sqsubseteq^{-1}$. For example:

- $\langle D, \sqsubseteq \rangle$ is ACC (CPO) means that the poset $\langle D_{/\cong_D}, \sqsubseteq_{/\cong_D} \rangle$ is ACC (CPO).
- $\langle D, \sqsubseteq \rangle$ is a join-semilattice means that $\langle D_{/\cong_D}, \sqsubseteq_{/\cong_D} \rangle$ is a join-semilattice; a binary lub for $D$ (one could have several binary lubs) is a function $\lambda \langle d, d' \rangle . d \sqcup d'$ such that $\lambda \langle [d]_{\cong_D}, [d']_{\cong_D} \rangle . [d \sqcup d']_{\cong_D}$ is the lub in the poset $\langle D_{/\cong_D}, \sqsubseteq_{/\cong_D} \rangle$.

**Corollary 4.7.** *Theorem 4.5 still holds for a QGC $\langle \wp(\Sigma^*), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle D, \sqsubseteq \rangle$ where $\langle D, \sqsubseteq \rangle$ is a qoset.*

# 5. Instantiating the Framework

In this section we focus on a particular class of closures: those induced by quasiorders. Then, we provide a list of conditions on quasiorders such that the induced closures fit our framework. In addition, we study some instances of such quasiorders and compare them.

## 5.1. Word-based Abstractions

Let $\leqslant$ be a quasiorder on words in $\Sigma^*$. A corresponding closure operator $\rho_\leqslant \in \mathrm{uco}(\wp(\Sigma^*))$ is defined as follows:

$$\rho_\leqslant(X) \triangleq \{ v \in \Sigma^* \mid \exists u \in X,\ u \leqslant v \} \ . \tag{7}$$

Thus, $\rho_{\leqslant}(X)$ is the $\leqslant$-upward closure of $X$ and it is easy to check that $\rho_{\leqslant}$ is indeed a closure on $\langle \wp(\Sigma^*), \subseteq \rangle$.

A quasiorder $\leqslant$ on $\Sigma^*$ is *left-monotonic* (*right-monotonic*) if $\forall y, x_1, x_2 \in \Sigma^*,\ x_1 \leqslant x_2 \Rightarrow yx_1 \leqslant yx_2$ ($x_1 y \leqslant x_2 y$). Also, $\leqslant$ is called monotonic if it is both left- and right-monotonic.

**Definition 5.1** (**$L$-Consistent Quasiorder**). Let $L \in \wp(\Sigma^*)$, a quasiorder $\leqslant_L$ on $\Sigma^*$ is called *left* (resp. *right*) *$L$-consistent* when (a) $\leqslant_L \cap (L \times \neg L) = \varnothing$ and (b) $\leqslant_L$ is left- (resp. right-) monotonic. Also, $\leqslant_L$ is called *$L$-consistent* when it is both left and right $L$-consistent. ■

It turns out that a $L$-consistent quasiorder induces a closure which includes $L$ and is backward complete.

**Lemma 5.2.** *Let $L$ be a language over $\Sigma$ and $\leqslant_L$ be a left (resp. right) $L$-consistent quasiorder on $\Sigma^*$. Then,*
(*a*) $\rho_{\leqslant_L}(L) = L$.
(*b*) $\rho_{\leqslant_L}$ *is backward complete for $\lambda X.\, aX$ (resp. $\lambda X.\, Xa$) for all $a \in \Sigma$.*

Moreover, we show that the $\leqslant$-upward closure $\rho_{\leqslant}$ defined in (7) can be equivalently defined through the qoset of antichains. In fact, the qoset of antichains $\langle \mathrm{AC}_{\langle \Sigma^*, \leqslant \rangle}, \sqsubseteq \rangle$ can be viewed as a language abstraction through the minor abstraction map. More precisely, let $\alpha_{\leqslant} : \wp(\Sigma^*) \to \mathrm{AC}_{\langle \Sigma^*, \leqslant \rangle}$ and $\gamma_{\leqslant} : \mathrm{AC}_{\langle \Sigma^*, \leqslant \rangle} \to \wp(\Sigma^*)$ be defined as follows:

$$\alpha_{\leqslant}(X) \triangleq \lfloor X \rfloor \qquad \gamma_{\leqslant}(Y) \triangleq \rho_{\leqslant}(Y) \ . \tag{8}$$

**Theorem 5.3.** *Let $\langle \Sigma^*, \leqslant \rangle$ be a qoset.*
(a) $\langle \wp(\Sigma^*), \subseteq \rangle \underset{\alpha_{\leqslant}}{\overset{\gamma_{\leqslant}}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \Sigma^*, \leqslant \rangle}, \sqsubseteq \rangle$ *is a QGC.*
(b) $\gamma_{\leqslant} \circ \alpha_{\leqslant} = \rho_{\leqslant}$.

The QGC $\langle \wp(\Sigma^*), \subseteq \rangle \underset{\alpha_{\leqslant}}{\overset{\gamma_{\leqslant}}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \Sigma^*, \leqslant \rangle}, \sqsubseteq \rangle$ allows us to represent and manipulate $\leqslant$-upward closed sets in $\wp(\Sigma^*)$ using finite subsets, as already shown by Abdulla et al. [1].

We are now in position to show that, given a language $L_2$ with decidable membership problem, for every decidable $L_2$-consistent wqo $\leqslant_{L_2}$, the QGC $\langle \wp(\Sigma^*), \subseteq \rangle \underset{\alpha_{\leqslant_{L_2}}}{\overset{\gamma_{\leqslant_{L_2}}}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \Sigma^*, \leqslant_{L_2} \rangle}, \sqsubseteq \rangle$ of Theorem 5.3 (a) yields an algorithm for deciding the inclusion $\mathcal{L}(\mathcal{A}) \subseteq L_2$ where $\mathcal{A}$ is a FA. In particular, for a left $L_2$-consistent wqo $\leqslant^l_{L_2}$, the algorithm `FAIncW` solves the inclusion problem. `FAIncW` is called "word-based" because $\langle Y_q \rangle_{q \in Q}$ consists of finite sets of words.

**Theorem 5.4.** *Let $\mathcal{A}$ be a FA and let $L_2$ be a language such that* (*i*) *membership in $L_2$ is decidable; and* (*ii*) *there exists a decidable left $L_2$-consistent wqo on $\Sigma^*$. Then,* `FAIncW` *decides the inclusion $\mathcal{L}(\mathcal{A}) \subseteq L_2$.*

*Proof:* Let $\leqslant^l_{L_2}$ be a decidable left $L_2$-consistent wqo on $\Sigma^*$. Let us check that hypotheses (i)-(v) of Theorem 4.5 are satisfied for $\langle D, \sqsubseteq \rangle = \langle \mathrm{AC}_{\langle \Sigma^*, \leqslant^l_{L_2} \rangle}, \sqsubseteq \rangle$, $\alpha = \lfloor \cdot \rfloor$, $\gamma = \rho_{\leqslant^l_{L_2}}$. Indeed we apply Corollary 4.7 because $\langle \mathrm{AC}_{\langle \Sigma^*, \leqslant^l_{L_2} \rangle}, \sqsubseteq \rangle$ is a qoset so that we deal with a QGC.

`FAIncW`**:** Word-based algorithm for $\mathcal{L}(\mathcal{A}) \subseteq L_2$

**Data:** FA $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$
**Data:** Decision procedure for membership in $L_2$
**Data:** Decidable left $L_2$-consistent wqo $\leqslant^l_{L_2}$
**Result:** Whether $\mathcal{L}(\mathcal{A}) \subseteq L_2$ holds

1 $\langle Y_q \rangle_{q \in Q} := \mathtt{Kleene}(\lambda \vec{\boldsymbol{X}}.\, \lfloor \vec{\boldsymbol{\epsilon}}^F \rfloor \sqcup \lfloor \mathrm{Pre}_{\mathcal{A}}(\vec{\boldsymbol{X}}) \rfloor, \vec{\boldsymbol{\varnothing}})$;
2 **forall** $q \in I$ **do**
3 **forall** $u \in Y_q$ **do**
4 **if** $u \notin L_2$ **then return** *false*;
5 **return** *true*;

(*i*) It follows from Theorem 5.3 (b) and Lemma 5.2 (a) that $L_2 \in \gamma(D)$. Moreover:

$$\begin{aligned} \alpha(aX) &= [\alpha = \alpha\gamma\alpha] \\ \alpha(\gamma(\alpha(aX))) &= [\text{L. 5.2 (b) with } \rho_{\leqslant^l_{L_2}} = \gamma\alpha] \\ \alpha(\gamma(\alpha(a\gamma(\alpha(X))))) &= [\alpha = \alpha\gamma\alpha] \\ \alpha(a\gamma\alpha(X)) \ . & \end{aligned}$$

(*ii*) It turns out that $\langle \mathrm{AC}_{\langle \Sigma^*, \leqslant^l_{L_2} \rangle}, \sqsubseteq \rangle$ is ACC because $\leqslant^l_{L_2}$ is a wqo. Moreover, the decidability of the binary relation $\leqslant^l_{L_2}$ entails that $W_1 \sqcup W_2 \triangleq \lfloor W_1 \cup W_2 \rfloor$ is a computable binary lub in $\langle \mathrm{AC}_{\langle \Sigma^*, \leqslant^l_{L_2} \rangle}, \sqsubseteq \rangle$. Hence, $\langle \mathrm{AC}_{\langle \Sigma^*, \leqslant^l_{L_2} \rangle}, \sqsubseteq, \sqcup \rangle$ is an effective domain.

(*iii*) Let us first observe that by Lemma 5.2 (b) and Theorem 4.3, $\rho_{\leqslant^l_{L_2}}$ is backward complete for $\mathrm{Pre}_{\mathcal{A}}$. Then, it turns out that $\alpha(\mathrm{Pre}_{\mathcal{A}}(\gamma(\vec{\boldsymbol{X}}))) = \lfloor \mathrm{Pre}_{\mathcal{A}}(\vec{\boldsymbol{X}}) \rfloor$ since

$$\begin{aligned} \alpha(\mathrm{Pre}_{\mathcal{A}}(\gamma(\vec{\boldsymbol{X}}))) &= [\alpha = \alpha\gamma\alpha] \\ \alpha\gamma\alpha(\mathrm{Pre}_{\mathcal{A}}(\gamma(\vec{\boldsymbol{X}}))) &= [\text{def. } \alpha \text{ and } \gamma] \\ \lfloor \rho_{\leqslant^l_{L_2}}(\mathrm{Pre}_{\mathcal{A}}(\rho_{\leqslant^l_{L_2}}(\vec{\boldsymbol{X}}))) \rfloor &= [\text{bw. completeness}] \\ \lfloor \rho_{\leqslant^l_{L_2}}(\mathrm{Pre}_{\mathcal{A}}(\vec{\boldsymbol{X}})) \rfloor &= [\alpha = \alpha\gamma\alpha] \\ \lfloor \mathrm{Pre}_{\mathcal{A}}(\vec{\boldsymbol{X}}) \rfloor \ . & \end{aligned}$$

This entails that $\alpha(\mathrm{Pre}_{\mathcal{A}}(\gamma(\vec{\boldsymbol{X}})))$ is computable.

(*iv*) $\alpha(\{\epsilon\}) = \{\epsilon\}$ and $\alpha(\varnothing) = \varnothing$, hence $\alpha(\vec{\boldsymbol{\epsilon}}^F) = \lfloor \vec{\boldsymbol{\epsilon}}^F \rfloor$ is trivial to compute.

(*v*) Since $\alpha(\vec{\boldsymbol{L_2}}^I) = \alpha(\langle \delta^{L_2}_{\Sigma^*}(q \in I) \rangle_{q \in Q}$, the relation $\langle Y_q \rangle_{q \in Q} \sqsubseteq \alpha(\vec{\boldsymbol{L_2}}^I)$ trivially holds for $Y_q$ with $q \notin I$. Therefore it suffices to check that $\forall q \in I$, $Y_q \sqsubseteq \alpha(L_2)$ is decidable. We have that:

$$\begin{aligned} Y_q \sqsubseteq \alpha(L_2) &\Leftrightarrow [\text{def. } \sqsubseteq \text{ and } \alpha] \\ \forall y \in Y_q, \exists x \in \lfloor L_2 \rfloor, x \leqslant^l_{L_2} y &\Leftrightarrow [\lfloor X \rfloor \text{ is a minor set}] \\ \forall y \in Y_q, \exists x \in L_2, x \leqslant^l_{L_2} y &\Leftrightarrow [\text{def. } \rho_{\leqslant^l_{L_2}}(L_2)] \\ \forall y \in Y_q, y \in \rho_{\leqslant^l_{L_2}}(L_2) &\Leftrightarrow [\rho_{\leqslant^l_{L_2}}(L_2) = L_2] \\ \forall y \in Y_q, y \in L_2 \ . & \end{aligned}$$

This latter condition coincides with the check performed by lines 2-5 of algorithm `FAIncW` and is therefore decidable.

Summing up, by Corollary 4.7, algorithm FAIncW solves $\mathcal{L}(\mathcal{A}) \subseteq L_2$. □

A symmetric version of algorithm FAIncW (and of Theorem 5.4) for *right* $L_2$-consistent wqos, which relies on equations concatenating to the right (instead of to the left as in $\mathrm{Eqn}(\mathcal{A})$), is given in Appendix B.

In what follows, we consider different quasiorders and show that they fulfill the requirements of Theorem 5.4 (or its symmetric for right quasiorders), hence, they yield algorithms for solving the language inclusion problem.

## 5.2. Nerode Quasiorders

Given $w \in \Sigma^*$ and $X \in \wp(\Sigma^*)$, left and right quotients are defined as usual: $w^{-1}X \triangleq \{u \in \Sigma^* \mid wu \in X\}$ and $Xw^{-1} \triangleq \{u \in \Sigma^* \mid uw \in X\}$. Given a language $L \subseteq \Sigma^*$, let us define the following quasiorders on $\Sigma^*$:

$$u \leqq^l_L v \overset{\triangle}{\Longleftrightarrow} Lu^{-1} \subseteq Lv^{-1}$$
$$u \leqq^r_L v \overset{\triangle}{\Longleftrightarrow} u^{-1}L \subseteq v^{-1}L \ .$$

De Luca and Varricchio [11] call them, resp., the *left* ($\leqq^l_L$) and *right* ($\leqq^r_L$) *Nerode quasiorders relative to* $L$. The following result shows that Nerode quasiorders are the most general (greatest for set inclusion) $L_2$-consistent quasiorders for which the above algorithm FAIncW can be used to decide tha language inclusion $\mathcal{L}(\mathcal{A}) \subseteq L_2$.

**Lemma 5.5.** *Let $L \subseteq \Sigma^*$ be a language.*
(a) *$\leqq^l_L$ and $\leqq^r_L$ are, resp., left and right $L$-consistent qos. If $L$ is regular then $\leqq^l_L$ and $\leqq^r_L$ are, additionally, decidable wqos.*
(b) *Let $\leqslant$ be a quasiorder on $\Sigma^*$. If $\leqslant$ is left (resp. right) $L$-consistent then $\rho_{\leqq^l_L} \subseteq \rho_{\leqslant}$ (resp. $\rho_{\leqq^r_L} \subseteq \rho_{\leqslant}$).*

*Proof:* De Luca and Varricchio [11, Theorem 2.4] show that $\leqq^l_L$ and $\leqq^r_L$ are left and right monotonic, respectively. Moreover, if $L$ is regular then they are wqos. Observe that given $u \in L$ and $v \notin L$ we have that $\epsilon \in Lu^{-1}$ and $\epsilon \in u^{-1}L$ while $\epsilon \notin Lv^{-1}$ and $\epsilon \notin v^{-1}L$. Hence, $\leqq^l_L$ ($\leqq^r_L$) is a left (right) $L$-consistent quasiorder. Finally, if $L$ is regular then both relations are clearly decidable.

Let us now show point (b). We consider the left case (the right case is symmetric). De Luca and Varricchio [11, Section 2, point 4] observe that $\leqq^l_L$ is maximum in the set of all left $L$-consistent quasiorders, i.e. every left $L$-consistent quasiorder $\leqslant$ is such that $x \leqslant y \Rightarrow x \leqq^l_L y$. As a consequence, $\rho_{\leqslant}(U) \subseteq \rho_{\leqq^l_L}(U)$ holds for all $U \in \wp(\Sigma^*)$:

$$\begin{aligned} \rho_{\leqslant}(U) = & \quad [\text{def. } \rho_{\leqslant}] \\ \{x \in \Sigma^* \mid \exists u \in U,\ u \leqslant x\} \subseteq & \quad [x \leqslant y \Rightarrow x \leqq^l_L y] \\ \{x \in \Sigma^* \mid \exists u \in U,\ u \leqq^l_L x\} = & \quad [\text{def. } \rho_{\leqq^l_L}] \\ \rho_{\leqq^l_L}(U) \ . & \end{aligned}$$

Therefore, $\rho_{\leqq^l_L} \subseteq \rho_{\leqslant}$. □

Let us now consider a first application of Theorem 5.4 for deciding $\mathcal{L}(\mathcal{A}) \subseteq L_2$ Because membership is decidable for regular languages, Lemma 5.5 (a) for $\leqq^l_{L_2}$ shows that the hypotheses (i) and (ii) of Theorem 5.4 hold, hence algorithm FAIncW decides the inclusion $\mathcal{L}(\mathcal{A}) \subseteq L_2$. Under these hypotheses, as a consequence of Lemma 5.5 (b) we have that $\leqq^l_{L_2}$ is the most general (i.e., greatest for set inclusion) left $L_2$-consistent quasiorder for which algorithm FAIncW can be used to decide $\mathcal{L}(\mathcal{A}) \subseteq L_2$.

We conclude with some useful remarks on the complexity of Nerode quasiorder relations. For the inclusion problem between languages generated by automata, deciding the (left or right) Nerode quasiorder can be easily shown[2] to be as hard as the language inclusion problem (which is PSPACE-hard). For the inclusion problem of a language generated by an automaton within the trace set of a one-counter net (cf. Section 5.3.2) the right Nerode quasiorder is a right language-consistent well-quasiorder but it turns out to be undecidable (cf. Lemma 5.11).

## 5.3. State-based Quasiorders

Consider the inclusion problem $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$ where $\mathcal{A}_1$ and $\mathcal{A}_2$ are FAs. In the following, we study a class of well-quasiorders based on $\mathcal{A}_2$. This is a strict subclass of Nerode quasiorders defined in Section 5.2 and sidesteps the untractability or undecidability of Nerode quasiorders yet allowing to define an algorithm solving the language inclusion problem.

**5.3.1. Inclusion in Regular Languages.** We define the quasiorders $\leq^l_{\mathcal{A}}$ and $\leq^r_{\mathcal{A}}$ on $\Sigma^*$ induced by a FA $\mathcal{A} = \langle Q, \delta, I, F, \Sigma\rangle$ as follows:

$$\begin{aligned} u \leq^l_{\mathcal{A}} v &\overset{\triangle}{\Longleftrightarrow} \mathrm{pre}^{\mathcal{A}}_u(F) \subseteq \mathrm{pre}^{\mathcal{A}}_v(F) \\ u \leq^r_{\mathcal{A}} v &\overset{\triangle}{\Longleftrightarrow} \mathrm{post}^{\mathcal{A}}_u(I) \subseteq \mathrm{post}^{\mathcal{A}}_v(I) \end{aligned} \tag{9}$$

where, for any $X \subseteq Q$ and $u \in \Sigma^*$, $\mathrm{pre}^{\mathcal{A}}_u(X) \triangleq \{q \in Q \mid u \in W^{\mathcal{A}}_{q,X}\}$ and $\mathrm{post}^{\mathcal{A}}_u(X) \triangleq \{q' \in Q \mid u \in W^{\mathcal{A}}_{X,q'}\}$.

**Lemma 5.6.** *Let $\mathcal{A}$ be an FA. Then $\leq^l_{\mathcal{A}}$ and $\leq^r_{\mathcal{A}}$ are, resp., decidable left and right $\mathcal{L}(\mathcal{A})$-consistent wqos.*

It follows from Lemma 5.6 that Theorem 5.4 applies to $\leq^l_{\mathcal{A}_2}$ (and $\leq^r_{\mathcal{A}_2}$), so that one can instantiate the algorithm FAIncW with the wqo $\leq^l_{\mathcal{A}_2}$ for deciding $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$. Turning back to the left Nerode wqo $\leqq^l_{\mathcal{L}(\mathcal{A}_2)}$ we find that:

$$\begin{aligned} u \leqq^l_{\mathcal{L}(\mathcal{A}_2)} v &\Leftrightarrow \mathcal{L}(\mathcal{A}_2)u^{-1} \subseteq \mathcal{L}(\mathcal{A}_2)v^{-1} \\ &\Leftrightarrow W_{I,\mathrm{pre}^{\mathcal{A}_2}_u(F)} \subseteq W_{I,\mathrm{pre}^{\mathcal{A}_2}_u(F)} \ . \end{aligned}$$

Since $\mathrm{pre}^{\mathcal{A}_2}_u(F) \subseteq \mathrm{pre}^{\mathcal{A}_2}_v(F) \Rightarrow W_{I,\mathrm{pre}^{\mathcal{A}_2}_u(F)} \subseteq W_{I,\mathrm{pre}^{\mathcal{A}_2}_u(F)}$, it follows that $u \leq^l_{\mathcal{A}_2} v \Rightarrow u \leqq^l_{\mathcal{L}(\mathcal{A}_2)} v$. Moreover, by Lemmas 5.5 (b) and 5.6, we also have that $\rho_{\leq^l_{\mathcal{L}(\mathcal{A}_2)}} \subseteq \rho_{\leq^l_{\mathcal{A}_2}}$.

**Simulation-based Quasiorders.** Let us recall that, given a FA $\mathcal{A} = \langle Q, \delta, I, F, \Sigma\rangle$, a *simulation* on $\mathcal{A}$ is a relation

2. Hint: given $\mathcal{A}_1$ and $\mathcal{A}_2$, group them into $\mathcal{A}_3$ and add transitions $q \overset{a}{\to} q'$ and $q \overset{b}{\to} q''$ for all $q \in I_3, q' \in I_1, q'' \in I_2$ to $\delta_3$. Then $a \leqq^r_{\mathcal{L}(\mathcal{A}_3)} b \Leftrightarrow a^{-1}\mathcal{L}(\mathcal{A}_3) \subseteq b^{-1}\mathcal{L}(\mathcal{A}_3) \Leftrightarrow \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$.

$\preceq \subseteq Q \times Q$ such that if $p \preceq q$ then: (*i*) $p \in F$ implies $q \in F$ and (*ii*) for every transition $p \xrightarrow{a} p'$, there exists a transition $q \xrightarrow{a} q'$ such that $p' \preceq q'$. It is well known that simulation implies language inclusion, i.e., if $\preceq$ is a simulation on $\mathcal{A}$ then

$$q \preceq q' \Rightarrow W^{\mathcal{A}}_{q,F} \subseteq W^{\mathcal{A}}_{q',F} \ .$$

We lift a qo $\preceq$ on $Q$ to a qo $\preceq^{\forall\exists}$ on $\wp(Q)$ as follows:

$$X \preceq^{\forall\exists} Y \iff \forall x \in X, \exists y \in Y,\ x \preceq y$$

so that $X \preceq^{\forall\exists} Y \Rightarrow W^{\mathcal{A}}_{X,F} \subseteq W^{\mathcal{A}}_{Y,F}$ holds. Therefore, we define the *right simulation-based quasiorder* $\preceq^r_{\mathcal{A}}$ on $\Sigma^*$ as:

$$u \preceq^r_{\mathcal{A}} v \iff \mathrm{post}^{\mathcal{A}}_u(I) \preceq^{\forall\exists} \mathrm{post}^{\mathcal{A}}_v(I) \qquad (10)$$

**Lemma 5.7.** *Given a simulation relation $\preceq$ on $\mathcal{A}$, the right simulation-based qo $\preceq^r_{\mathcal{A}}$ is a decidable right $\mathcal{L}(\mathcal{A})$-consistent wqo.*

Thus, once again, Theorem 5.4 applies to $\preceq^r_{\mathcal{A}_2}$ and this allows us to instantiate the algorithm `FAIncW` to $\preceq^r_{\mathcal{A}_2}$ for deciding $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$.

Observe that $u \preceq^r_{\mathcal{A}_2} v$ implies $W_{\mathrm{post}^{\mathcal{A}_2}_u(I),F} \subseteq W_{\mathrm{post}^{\mathcal{A}_2}_v(I),F}$ which is equivalent to the right Nerode quasiorder $u \leqq^r_{\mathcal{L}(\mathcal{A}_2)} v$, so that $u \preceq^r_{\mathcal{A}_2} v \Rightarrow u \leqq^r_{\mathcal{L}(\mathcal{A}_2)} v$. Moreover, $u \leq^r_{\mathcal{A}_2} v \Rightarrow u \preceq^r_{\mathcal{A}_2} v$ trivially holds. Summing up, the following containments relate (the right versions of) state-based, simulation-based and Nerode quasiorders:

$$\leq^r_{\mathcal{A}_2} \subseteq \preceq^r_{\mathcal{A}_2} \subseteq \leqq^r_{\mathcal{L}(\mathcal{A}_2)}$$

All these quasiorders are decidable $\mathcal{L}(\mathcal{A}_2)$-consistent wqos so that the algorithm `FAIncW` can be instantiated to each of them for deciding $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$.

**5.3.2. Inclusion in Traces of One-Counter Nets.** In this section show that our framework can be used to systematically derive an algorithm for deciding the inclusion $\mathcal{L}(\mathcal{A}) \subseteq L_2$ when $L_2$ is the trace set of a one-counter net. We proceed by showing that there exists a decidable $L_2$-consistent quasiorder so that we can apply Theorem 5.4.

Intuitively, a one-counter net is a FA equipped with a nonnegative integer counter. Formally, a One-Counter Net (OCN) [23] is a tuple $\mathcal{O} = \langle Q, \Sigma, \delta \rangle$ where $Q$ is the finite set of *states*, $\Sigma$ is the *alphabet* and $\delta \subseteq Q \times \Sigma \times \{-1, 0, 1\} \times Q$ is the set of *transitions*. A *configuration* of $\mathcal{O}$ is a pair $qn$ consisting of a state $q \in Q$ and a value $n \in \mathbb{N}$ for the counter. Given two configurations $qn$ and $q'n'$ we write $qn \xrightarrow{a} q'n'$ and call it a *a-step* (or simply *step*) if there exists a transition $(q, a, d, q') \in \delta$ such that $n' = n + d$. Given $qn \in Q \times \mathbb{N}$, the *trace set of an OCN*, $T(qn) \subseteq \Sigma^*$, is defined as follows:

$$T(qn) \triangleq \{u \in \Sigma^* \mid Z^{qn}_u \neq \varnothing\} \text{ with}$$
$$Z^{qn}_u \triangleq \{q_k n_k \mid qn \xrightarrow{a_1} q_1 n_1 \cdots \xrightarrow{a_k} q_k n_k \wedge a_1 \cdots a_k = u\}$$

Observe that $Z^{qn}_\epsilon = \{qn\}$ and $Z^{qn}_u$ is finite for all words $u \in \Sigma^*$. For a set $S \subseteq Q \times \mathbb{N}$, $T(S) \triangleq \bigcup_{qn \in S} T(qn)$.

Let $\mathbb{N}_\perp \triangleq \mathbb{N} \cup \{\perp\}$ where $\perp \leq_{\mathbb{N}_\perp} n$ holds for all values $n \in \mathbb{N}_\perp$. For a finite set of states $S \subseteq Q \times \mathbb{N}$ define the so-called macro state $M_S \colon Q \to \mathbb{N}_\perp$ as

$$M_S(q) \triangleq \max\{n \in \mathbb{N} \mid qn \in S\}$$

where $\max \varnothing \triangleq \perp$. Define the following quasiorder on $\Sigma^*$:

$$u \leq^r_{qn} v \iff \forall q' \in Q,\ M_{Z^{qn}_u}(q') \leq_{\mathbb{N}_\perp} M_{Z^{qn}_v}(q')$$

**Lemma 5.8.** *Given a OCN $\mathcal{O}$ together with a configuration $qn$, $\leq^r_{qn}$ is a right $T(qn)$-consistent decidable wqo.*

Thus, as a consequence of Theorem 5.4, Lemma 5.8 and the decidability of membership in $T(qn)$, we derive the following known decidability result ([24, Theorem 3.2]) by resorting to our framework.

**Theorem 5.9.** *Given a FA $\mathcal{A}$ and a OCN $\mathcal{O}$ together with a configuration $qn$, the problem $\mathcal{L}(\mathcal{A}) \subseteq T(qn)$ is decidable.*

Moreover, the following result closes a conjecture made by De Luca and Varricchio [11, Section 6].

**Lemma 5.10.** *The right Nerode quasiorder $\leqq^r_{T(qn)}$ relative to $T(qn)$ is a well-quasiorder.*

*Proof:* De Luca and Varricchio [11] show that $\leqq^r_{T(qn)}$ is maximum in the set of all right $T(qn)$-consistent quasiorders, that is, $u \leq^r_{qn} v$ implies $u \leqq^r_{T(qn)} v$ for all $u, v \in \Sigma^*$. Since $\leq^r_{qn}$ is a wqo then $\leqq^r_{T(qn)}$ is a wqo. □

It is worth remarking that, by Lemma 5.5 (a), the left and right Nerode quasiorders relative to $T(qn)$ are $T(qn)$-consistent. However, the left Nerode quasiorder does not need to be a wqo for otherwise $T(qn)$ would be regular.

**Lemma 5.11.** *The right Nerode quasiorder for the trace set of OCN is undecidable.*

*Proof:* Let $\leqq^r_{T(qn)}$ denote the right Nerode quasiorder for a $T(qn)$. Undecidability for $\leqq^r_{T(qn)}$ follows from the undecidability of the trace inclusion problem for nondeterministic OCNs [20, Theorem 20] by an argument similar to the automata case. □

We conjecture that, using our framework, Theorem 5.9 can be extended to traces of Petri Nets, which is already known to be true [24].

## 6. A Novel Perspective on the Antichain Algorithm

Consider two FAs $\mathcal{A}_1 = \langle Q_1, \delta_1, I_1, F_1, \Sigma \rangle$ and $\mathcal{A}_2 = \langle Q_2, \delta_2, I_2, F_2, \Sigma \rangle$. and consider the left $\mathcal{L}(\mathcal{A}_2)$-consistent wqo $\leq^l_{\mathcal{A}_2}$ defined in (9). Theorem 5.4 shows that the algorithm `FAIncW` solves the inclusion problem $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$ by working on the qoset abstraction $\langle \mathrm{AC}_{\langle \Sigma^*, \leq^l_{\mathcal{A}_2} \rangle}, \sqsubseteq \rangle$ of antichains of $\langle \Sigma^*, \leq^l_{\mathcal{A}_2} \rangle$.

Note that since $u \leq^l_{\mathcal{A}_2} v \Leftrightarrow \mathrm{pre}^{\mathcal{A}_2}_u(F_2) \subseteq \mathrm{pre}^{\mathcal{A}_2}_v(F_2)$, it suffices to keep the sets of states $\mathrm{pre}^{\mathcal{A}_2}_u(F_2)$ for each word $u$ instead of the words themselves. Thus, we could design

an algorithm analogous to FAIncW but working on the poset abstraction $\langle \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle}, \sqsubseteq\rangle$ of antichains of sets of states of $\langle \wp(Q_2), \subseteq\rangle$. In order to do this, $\langle \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle}, \sqsubseteq\rangle$ can be viewed as a further abstraction of the antichain qoset $\langle \mathrm{AC}_{\langle \Sigma^*,\leq^l_{\mathcal{A}_2}\rangle}, \sqsubseteq'\rangle$ (where $\sqsubseteq'$ is used for distinguishing the two ordering relations on antichains) through the abstraction and concretization maps $\alpha_{\mathcal{A}_2} : \mathrm{AC}_{\langle \Sigma^*,\leq^l_{\mathcal{A}_2}\rangle} \to \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle}$ and $\gamma_{\mathcal{A}_2} : \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle} \to \mathrm{AC}_{\langle \Sigma^*,\leq^l_{\mathcal{A}_2}\rangle}$ defined as follows:

$$\alpha_{\mathcal{A}_2}(X) \triangleq \{\mathrm{pre}^{\mathcal{A}_2}_u(F_2) \mid u \in X\}$$
$$\gamma_{\mathcal{A}_2}(Y) \triangleq \lfloor \{u \in \Sigma^* \mid \mathrm{pre}^{\mathcal{A}_2}_u(F_2) \in Y\}\rfloor$$

**Lemma 6.1.** $\langle \mathrm{AC}_{\langle \Sigma^*,\leq^l_{\mathcal{A}_2}\rangle}, \sqsubseteq'\rangle \underset{\alpha_{\mathcal{A}_2}}{\overset{\gamma_{\mathcal{A}_2}}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle}, \sqsubseteq\rangle$ *is a QGC.*

Combining the word-based algorithm FAIncW with these functions $\alpha_{\mathcal{A}_2}$ and $\gamma_{\mathcal{A}_2}$ we are able to systematically derive a novel algorithm solving the inclusion $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$ using the abstract domain $\langle \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle}, \sqsubseteq\rangle$ by composing the two QGCs:

$$\langle \wp(\Sigma^*), \subseteq\rangle \underset{\alpha_{\leq^l_{\mathcal{A}_2}}}{\overset{\gamma_{\leq^l_{\mathcal{A}_2}}}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \Sigma^*,\leq^l_{\mathcal{A}_2}\rangle}, \sqsubseteq'\rangle \quad \text{[Theorem 5.3 (a)]}$$

$$\langle \mathrm{AC}_{\langle \Sigma^*,\leq^l_{\mathcal{A}_2}\rangle}, \sqsubseteq'\rangle \underset{\alpha_{\mathcal{A}_2}}{\overset{\gamma_{\mathcal{A}_2}}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle}, \sqsubseteq\rangle \quad \text{[Lemma 6.1]}$$

Let $\alpha\colon \wp(\Sigma^*) \to \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle}$, $\gamma\colon \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle} \to \wp(\Sigma^*)$ and $\mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\langle X_q\rangle_{q\in Q_1}) : \wp(Q_2)^{|Q_1|} \to \wp(Q_2)^{|Q_1|}$ be defined as follows:

$$\alpha(X) \triangleq \lfloor \{\mathrm{pre}^{\mathcal{A}_2}_u(F_2) \mid u \in X\}\rfloor$$
$$\gamma(Y) \triangleq \{u \in \Sigma^* \mid \exists y \in Y, y \subseteq \mathrm{pre}^{\mathcal{A}_2}_u(F_2)\}$$
$$\mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\langle X_q\rangle_{q\in Q_1}) \triangleq$$
$$\langle \lfloor \{\, \mathrm{pre}^{\mathcal{A}_2}_a(s) \mid \exists a \in \Sigma, q' \in Q_1,\, q \xrightarrow{a}_{\mathcal{A}_1} q' \wedge s \in X_{q'}\}\rfloor\rangle_{q\in Q_1}$$

**Lemma 6.2.** *The following hold:*
(a) $\alpha = \alpha_{\mathcal{A}_2} \circ \alpha_{\leq^l_{\mathcal{A}_2}}$
(b) $\gamma = \gamma_{\leq^l_{\mathcal{A}_2}} \circ \gamma_{\mathcal{A}_2}$
(c) $\langle \wp(\Sigma^*), \subseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle}, \sqsubseteq\rangle$ *is a GC.*
(d) $\gamma \circ \alpha = \rho_{\leq^l_{\mathcal{A}_2}}$
(e) $\mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\vec{\boldsymbol{X}}) = \alpha_{\mathcal{A}_2} \circ \alpha_{\leq^l_{\mathcal{A}_2}} \circ \mathrm{Pre}_{\mathcal{A}_1} \circ \gamma_{\leq^l_{\mathcal{A}_2}} \circ \gamma_{\mathcal{A}_2}(\vec{\boldsymbol{X}})$ *for all* $\vec{\boldsymbol{X}} \in \alpha(\wp(\Sigma^*)^{|Q_1|})$

It follows from Lemma 6.2 that the Galois Connection $\langle \wp(\Sigma^*), \subseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle}, \sqsubseteq\rangle$ together with the abstract function $\mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}$ satisfy the requirements (i)-(iv) of Theorem 4.5. In order to obtain an algorithm solving the inclusion $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$ it remains to show that requirement (v) of Theorem 4.5 holds, i.e., there is an algorithm to decide whether $\vec{\boldsymbol{Y}} \sqsubseteq \alpha(\vec{\boldsymbol{L}}^{I_2}_{\boldsymbol{2}})$ for every $\vec{\boldsymbol{Y}} \in \alpha(\wp(\Sigma^*))^{|Q_1|}$.

Let us notice that the Kleene's iterates of the abstract function $\lambda\vec{\boldsymbol{X}}.\, \alpha(\vec{\boldsymbol{\epsilon}}^{F_1}) \sqcup \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\vec{\boldsymbol{X}})$ of Theorem 4.5 are vectors in $\langle \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle}, \sqsubseteq\rangle$ where each component $q \in Q_1$ represents (through its minor set) a set of sets of states that are predecessors of $F_2$ in $\mathcal{A}_2$ by a word generated by $\mathcal{A}_1$ from state $q$ (i.e. $\mathrm{pre}^{\mathcal{A}_2}_u(F_2)$ with $u \in W^{\mathcal{A}_1}_{q,F_1}$). Since $\epsilon \in W^{\mathcal{A}_1}_{q,F_1}$ for all $q \in F_1$ and $\mathrm{pre}^{\mathcal{A}_2}_\epsilon(F_2) = F_2$ the iterations of the procedure Kleene start with $\alpha(\vec{\boldsymbol{\epsilon}}^{F_1}) = \langle \delta^{F_2}_{\varnothing}(q \in F_1)\rangle_{q\in Q_1}$. By taking the minor of each vector component, we are considering smaller sets which still preserve the relation $\sqsubseteq$ (because $A \sqsubseteq B \Leftrightarrow \lfloor A\rfloor \sqsubseteq B \Leftrightarrow A \sqsubseteq \lfloor B\rfloor \Leftrightarrow \lfloor A\rfloor \sqsubseteq \lfloor B\rfloor$). Let $\vec{\boldsymbol{Y}}$ be the fixpoint computed by the Kleene procedure. We have that, for each $q \in Q_1$, $\vec{\boldsymbol{Y}}_q = \lfloor \{\mathrm{pre}^{\mathcal{A}_2}_u(F_2) \mid u \in W^{\mathcal{A}_1}_{q,F_1}\}\rfloor$. Whenever $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$ holds, all the sets of states in $\vec{\boldsymbol{Y}}_q$ for $q \in I_1$ are predecessors of $F_2$ in $\mathcal{A}_2$ by words in $\mathcal{L}(\mathcal{A}_2)$, so that they all contain at least one initial state in $I_2$. As a result, we obtain the "state-based" algorithm FAIncS.

**FAIncS:** State-based algorithm for $\mathcal{L}(\mathcal{A}_1)\subseteq\mathcal{L}(\mathcal{A}_2)$

**Data:** FA $\mathcal{A}_1 = \langle Q_1, \delta_1, I_1, F_1, \Sigma\rangle$
**Data:** FA $\mathcal{A}_2 = \langle Q_2, \delta_2, I_2, F_2, \Sigma\rangle$
**Result:** Whether $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$ holds

```
⟨Y_q⟩_{q∈Q_1} := Kleene(λX⃗. α(ε⃗^{F_1}) ⊔ Pre^{A_2}_{A_1}(X⃗), ∅⃗);
forall q ∈ I_1 do
    forall s ∈ Y_q do
        if s ∩ I_2 = ∅ then return false;
return true;
```

**Theorem 6.3.** *Let* $\mathcal{A}_1, \mathcal{A}_2$ *be two FAs. The algorithm* FAIncS *decides* $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$.

In what follows we show that FAIncS precisely coincides with the well-known antichain algorithm put forward by Wulf et al. [13]. To this end, let us consider the following poset of antichains $\langle \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle}, \widehat{\sqsubseteq}\rangle$ where

$$X \,\widehat{\sqsubseteq}\, Y \overset{\triangle}{\Longleftrightarrow} \forall x \in X, \exists y \in Y,\ x \subseteq y$$

Thus, we have that $\langle \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle}, \widehat{\sqsubseteq}\rangle = \langle \mathrm{AC}_{\langle \wp(Q_2),\supseteq\rangle}, \sqsubseteq\rangle$ and, as observed in [14], $\langle \mathrm{AC}_{\langle \wp(Q_2),\subseteq\rangle}, \widehat{\sqsubseteq}\rangle$ is a finite lattice, where $\widehat{\sqcap}$ and $\widehat{\sqcup}$ denote, resp., glb and lub of antichains.

Let $S^c$ denote the complement of a generic subset $S$. The antichain algorithm described by Wulf et. al [13] for checking $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$ can be stated as follows.

**Theorem 6.4 ([13, Theorem 6]).** *Let*

$$\overrightarrow{\boldsymbol{\mathcal{FP}}} \triangleq \widehat{\bigsqcap}(\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(S) \ \widehat{\sqcup}\ \langle \delta^{\{F_2^c\}}_{\varnothing}(q \in F_1)\rangle_{q\in Q_1})$$

*where* $\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1} : \wp(Q_2)^{|Q_1|} \to \wp(Q_2)^{|Q_1|}$ *is defined by:*

$$\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\langle X_q\rangle_{q\in Q_1}) \triangleq \langle \lceil \{Y \mid \exists a \in \Sigma, q' \in Q_1, X \subseteq Q_2,$$
$$q \xrightarrow{a}_{\mathcal{A}_1} q' \wedge X \in X_{q'} \wedge \mathrm{post}^{\mathcal{A}_2}_a(Y) \subseteq X\}\rceil\rangle_{q\in Q_1}$$

*Then,* $\mathcal{L}(\mathcal{A}_1) \nsubseteq \mathcal{L}(\mathcal{A}_2)$ *iff* $\exists q \in I_2, \{I_2\} \,\widehat{\sqsubseteq}\, \overrightarrow{\boldsymbol{\mathcal{FP}}}_q$.

The intuition behind the antichain algorithm is to compute for each state $q \in Q_1$ the set of states that are not predecessors of $F_2$ in $\mathcal{A}_2$ by any word generated by $\mathcal{A}_1$ from $q$ (i.e. $(\mathrm{pre}^{\mathcal{A}_2}_u(F_2))^c$ with $u \in W^{\mathcal{A}_1}_{q,F_1}$). Since $\epsilon \in W^{\mathcal{A}_1}_{q,F_1}$ for all $q \in F_1$ and $\mathrm{pre}^{\mathcal{A}_2}_\epsilon(F_2) = F_2$, the iteration begins with $\langle \delta^{\{F_2^c\}}_{\varnothing}(q \in F_1)\rangle_{q\in Q_1}$, which is

the complement of $\alpha(\overrightarrow{\boldsymbol{\epsilon}}^{F_1})$. By using the major operator $\lceil\cdot\rceil$ (dual of the minor operator) the antichain algorithm processes smaller sets while preserving the relation $\sqsubseteq'$ (as $A \sqsubseteq' B \Leftrightarrow \lceil A\rceil \sqsubseteq' B \Leftrightarrow A \sqsubseteq' \lceil B\rceil \Leftrightarrow \lceil A\rceil \sqsubseteq' \lceil B\rceil$). As a consequence, $\overrightarrow{\overline{\boldsymbol{\mathcal{FP}}}}_q = \lceil\{(\mathrm{pre}_u(F_2))^c \mid u \in W^{\mathcal{A}_1}_{q,F_1}\}\rceil$ and if $\mathcal{L}(\mathcal{A}_1) \nsubseteq \mathcal{L}(\mathcal{A}_2)$ then there exists a word $u \in \mathcal{L}(\mathcal{A}_1)$ such that $u \notin \mathcal{L}(\mathcal{A}_2)$, so that $I_2 \cap \mathrm{pre}^{\mathcal{A}_2}_u(F_2) = \varnothing$, namely, $I_2 \subseteq (\mathrm{pre}^{\mathcal{A}_2}_u(F_2))^c$.

To summarize, while our algorithm FAIncS considers upper closed sets in $\wp(Q_2)$ represented by their minimal elements, the antichain algorithm considers dual downward closed sets in $\wp(Q_2)$ represented by their maximal elements. Equivalently, FAIncS works on the abstraction $\langle \mathrm{AC}_{\langle\wp(Q_2),\subseteq\rangle}, \sqsubseteq\rangle$ while the antichain algorithm works on its dual lattice $\langle \mathrm{AC}_{\langle\wp(Q_2),\supseteq\rangle}, \sqsubseteq\rangle$. Theorem 6.5 precisely formalizes this duality between these two algorithms.

**Theorem 6.5.** *At each step of the least fixpoint computations for $\overrightarrow{\overline{\boldsymbol{\mathcal{FP}}}}$ of Theorem 6.4 and $\overrightarrow{\boldsymbol{Y}}$ of* FAIncS*, the following invariant holds:*

$$\forall S \subseteq Q_2, q \in Q_1,\ \{S\} \mathrel{\widehat{\sqsubseteq}} \overrightarrow{\overline{\boldsymbol{\mathcal{FP}}}}_q \Leftrightarrow \{S^c\} \sqsubseteq \overrightarrow{\boldsymbol{Y}}_q$$

The forward antichain algorithm (previously we considered the backward version) can be shown to be equivalent to the algorithm systematically derived within our framework when considering the quasiorder $u \leqslant^r_{\mathcal{A}_2} v$ as defined in (9).

Abdulla et al. [2] and subsequently Bonchi and Pous [5] improved the original antichain algorithm by exploiting a precomputed simulation quasiorder relation on the states of the input automata. Note that $\preceq^r_{\mathcal{A}}$, by definition, does not consider pairs of states in the simulation relation outside $Q_2 \times Q_2$, while the works mentioned above do so.

## 7. Inclusion for Context Free Languages

A *context-free grammar* (CFG) is a tuple $\mathcal{G} = \langle \mathcal{V}, \Sigma, P\rangle$ where $\mathcal{V} = \{X_0, \ldots, X_n\}$ is the finite set of *variables* including the *start symbol* $X_0$, $\Sigma$ is the finite alphabet of *terminals*, and $P$ is the set of *productions* $X_i \to \beta$ where $\beta \in (\mathcal{V} \cup \Sigma)^*$. We assume, for simplicity and without loss of generality, that CFGs are in Chomsky Normal Form (CNF), that is, every production $X_i \to \beta \in P$ is such that $\beta \in (\mathcal{V} \times \mathcal{V}) \cup \Sigma \cup \{\epsilon\}$ and if $\beta = \epsilon$ then $i = 0$ [6]. We also assume that for all $X_i \in \mathcal{V}$ there exists a production $X_i \to \beta \in P$, otherwise $X_i$ can be safely removed from $\mathcal{V}$.

### 7.1. Extending the Framework to CFGs

Similarly to the case of automata discussed in Sections 4 and 5, a context-free grammar $\mathcal{G} = (\mathcal{V}, \Sigma, P)$ in CNF induces the following set of equations:

$$\mathrm{Eqn}(\mathcal{G}) \triangleq \{X_i = \textstyle\bigcup_{X_i \to \beta_j \in P} \beta_j \mid i \in [0, n]\}\ .$$

We define the vector $\overrightarrow{\boldsymbol{b}} \in \wp(\Sigma^*)^{|\mathcal{V}|}$ and the function $\mathrm{Fn}_{\mathcal{G}} : \wp(\Sigma^*)^{|\mathcal{V}|} \to \wp(\Sigma^*)^{|\mathcal{V}|}$ which are used to formalize the fixpoint equations in $\mathrm{Eqn}(\mathcal{G})$ as follows:

- $\overrightarrow{\boldsymbol{b}} \triangleq \langle b_i\rangle_{i\in[0,n]} \in \wp(\Sigma^*)^{|\mathcal{V}|}$ where is component is of the form $b_i \triangleq \{\beta \mid X_i \to \beta \in P,\ \beta \in \Sigma \cup \{\epsilon\}\}$.
- $\mathrm{Fn}_{\mathcal{G}}(\langle X_i\rangle_{i\in[0,n]}) \triangleq \langle \beta^{(i)}_1 \cup \ldots \cup \beta^{(i)}_{k_i}\rangle_{i\in[0,n]}$ where each $\beta^{(i)}_j$ is such that $\beta^{(i)}_j \in \mathcal{V}^2$ and $X_i \to \beta^{(i)}_j \in P$.

Notice that $\lambda\overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}})$ is a well-defined monotonic function in $\wp(\Sigma^*)^{|\mathcal{V}|} \to \wp(\Sigma^*)^{|\mathcal{V}|}$. Given the fixpoint $\langle Y_i\rangle_{i\in[0,n]} = \mathrm{lfp}(\lambda\overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}}))$, it is known from Ginsburg and Rice [19] that the language $\mathcal{L}(\mathcal{G})$ accepted by $\mathcal{G}$ is such that $\mathcal{L}(\mathcal{G}) = Y_0$.

**Example 7.1.** Consider a CFG $\mathcal{G}$ in CNF with $\Sigma = \{a, b\}$ and productions $\{X_0 \to X_0X_1 \mid X_1X_0 \mid b,\ X_1 \to a\}$. The corresponding equation system is $\mathrm{Eqn}(\mathcal{G}) = \{X_0 = X_0X_1 \cup X_1X_0 \cup \{b\},\ X_1 = \{a\}\}$. Also, we have that $\overrightarrow{\boldsymbol{b}} = \langle\{b\}, \{a\}\rangle \in \wp(\Sigma^*)^2$ and $\mathrm{Fn}_{\mathcal{G}} : \wp(\Sigma^*)^2 \to \wp(\Sigma^*)^2$ is given by $\mathrm{Fn}_{\mathcal{G}}(\langle X_0, X_1\rangle) = \langle X_0X_1 \cup X_1X_0, \varnothing\rangle$. The infinite sequence of Kleene iterates of the least fixpoint computation of $\lambda\overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}})$ goes as follows:

$$\langle\varnothing, \varnothing\rangle \Rightarrow \langle\{b\}, \{a\}\rangle \Rightarrow \langle\{ab, ba, b\}, \{a\}\rangle \Rightarrow \langle\{ab, ba, b, aab, aba, baa\}, \{a\}\rangle \Rightarrow \cdots \quad \Diamond$$

Hence, by Ginsburg and Rice [19] we have that

$$\mathcal{L}(\mathcal{G}) \subseteq L_2 \ \Leftrightarrow\ \mathrm{lfp}(\lambda\overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}})) \subseteq \overrightarrow{\boldsymbol{L_2}}^{X_0}$$

where $\overrightarrow{\boldsymbol{L_2}}^{X_0} \triangleq \langle\delta^{L_2}_{\Sigma^*}(i = 0)\rangle_{i\in[0,n]}$.

**Theorem 7.2.** *Let $\mathcal{G} = (\mathcal{V}, \Sigma, P)$ be a CFG in CNF. If $\rho \in \mathrm{uco}(\wp(\Sigma^*))$ is backward complete for both $\lambda X.Xa$ and $\lambda X.aX$, for all $a \in \Sigma$, then $\rho$ is backward complete for $\mathrm{Fn}_{\mathcal{G}}$ and $\lambda\overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}})$.*

As a consequence, by backward completeness of $\rho$,

$$\rho(\mathrm{lfp}(\lambda\overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}}))) = \mathrm{lfp}(\lambda\overrightarrow{\boldsymbol{X}}.\, \rho(\overrightarrow{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}})))\ .$$

Note that if $\rho$ is backward complete for left and right concatenation and $\rho(L_2) = L_2$ then, as a straightforward consequence of Theorems 3.1 and 7.2, we have that:

$$\mathcal{L}(\mathcal{G}) \subseteq L_2 \Leftrightarrow \mathrm{lfp}(\lambda\overrightarrow{\boldsymbol{X}}.\, \rho(\overrightarrow{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}}))) \subseteq \overrightarrow{\boldsymbol{L_2}}^{X_0}\ . \quad (11)$$

The following results are the equivalent of Theorem 4.5 and Corollary 4.7 for context-free languages.

**Theorem 7.3.** *Let $\mathcal{G} = \langle\mathcal{V}, \Sigma, P\rangle$ be a CFG in CNF and let $L_2$ be a language over $\Sigma$. Let $\langle\wp(\Sigma^*), \subseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle D, \sqsubseteq\rangle$ be a GC where $\langle D, \sqsubseteq\rangle$ is a poset. Assume that the following properties hold:*

(*i*) *$L_2 \in \gamma(D)$ and for every $a \in \Sigma$, $X \in \wp(\Sigma^*)$ we have $\alpha(aX) = \alpha(a\gamma\alpha(X))$ and $\alpha(Xa) = \alpha(\gamma\alpha(X)a)$.*

(*ii*) *$(D, \sqsubseteq, \sqcup)$ is an effective domain, meaning that: $(D, \sqsubseteq)$ is ACC, every element of $D$ has a finite representation, $\sqsubseteq$ is decidable and $\sqcup$ is a computable binary lub.*

(*iii*) *There is an algorithm, say $\mathrm{Fn}^\sharp(\overrightarrow{\boldsymbol{X}})$, computing $\alpha(\mathrm{Fn}_{\mathcal{G}}(\gamma(\overrightarrow{\boldsymbol{X}})))$, for all $\overrightarrow{\boldsymbol{X}} \in \wp(\Sigma^*)^{|\mathcal{V}|}$.*

(*iv*) *There is an algorithm, say $\mathrm{b}^\sharp$, computing $\alpha(\overrightarrow{\boldsymbol{b}})$.*

(*v*) *There is an algorithm, say $\mathrm{Incl}^\sharp(\overrightarrow{\boldsymbol{X}})$, deciding the abstract inclusion $\overrightarrow{\boldsymbol{X}} \sqsubseteq \alpha(\overrightarrow{\boldsymbol{L_2}}^{X_0})$, for every vector $\overrightarrow{\boldsymbol{X}} \in \alpha(\wp(\Sigma^*)^{|\mathcal{V}|})$.*

*Then, the following algorithm decides whether $\mathcal{L}(\mathcal{G}) \subseteq L_2$:*

$\langle Y_i \rangle_{i\in[0,n]} := \texttt{Kleene}(\lambda \vec{\boldsymbol{X}}.\, \mathrm{b}^\sharp \sqcup \mathrm{Fn}^\sharp(\vec{\boldsymbol{X}}), \vec{\boldsymbol{\varnothing}})$;
**return** $\mathrm{Incl}^\sharp(\langle Y_i \rangle_{i\in[0,n]})$;

**Corollary 7.4.** *Theorem 7.3 still holds for a QGC $\langle \wp(\Sigma^*), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle D, \sqsubseteq \rangle$ where $\langle D, \sqsubseteq \rangle$ is a qoset.*

## 7.2. Instantiating the Framework

As we did in Section 5 for the language inclusion problem $\mathcal{L}(\mathcal{A}) \subseteq L_2$, we next show how to systematically derive an algorithm solving $\mathcal{L}(\mathcal{G}) \subseteq L_2$ and we discuss and compare some quasiorders for which our framework applies.

**Word-based Abstractions** By Lemma 5.2, it turns out that a (left and right) $L$-consistent quasiorder $\leqslant_L$ on $\Sigma^*$ induces the $\leqslant_L$-upward closure $\rho_{\leqslant_L} \in \mathrm{uco}(\wp(\Sigma^*))$ defined in (7) such that: (a) $\rho_{\leqslant_L}(L) = L$, and (b) $\rho_{\leqslant_L}$ is backward complete for $\lambda X.\, aX$ and $\lambda X.\, Xa$, for all $a \in \Sigma$. Moreover, the closure $\rho_{\leqslant_L}$ can be defined through the qoset of antichains. As shown by Theorem 5.3, the maps $\alpha_{\leqslant_L}$ and $\gamma_{\leqslant_L}$ define a QGC $\langle \wp(\Sigma^*), \subseteq \rangle \underset{\alpha_{\leqslant_L}}{\overset{\gamma_{\leqslant_L}}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \Sigma^*, \leqslant_L \rangle}, \sqsubseteq \rangle$ with $\rho_{\leqslant_L} = \gamma_{\leqslant_L} \circ \alpha_{\leqslant_L}$.

Theorem 5.4 shows that every decidable left $L_2$-consistent wqo $\leqslant_{L_2}$ yields an algorithm for deciding $\mathcal{L}(\mathcal{A}) \subseteq L_2$. Next, we show a similar result for $\mathcal{L}(\mathcal{G}) \subseteq L_2$.

**`CFGIncW`: Word-based algorithm for $\mathcal{L}(\mathcal{G}) \subseteq L_2$**

**Data:** CFG $\mathcal{G} = \langle \mathcal{V}, \Sigma, P \rangle$
**Data:** Decision procedure for membership in $L_2$
**Data:** Decidable $L_2$-consistent wqo $\leqslant_{L_2}$
**Result:** Whether $\mathcal{L}(\mathcal{G}) \subseteq L_2$ holds

$\langle Y_i \rangle_{i\in[0,n]} := \texttt{Kleene}(\lambda \vec{\boldsymbol{X}}.\, \lfloor \vec{\boldsymbol{b}} \rfloor \sqcup \lfloor \mathrm{Fn}_{\mathcal{G}}(\vec{\boldsymbol{X}}) \rfloor, \vec{\boldsymbol{\varnothing}})$;
**forall** $u \in Y_0$ **do**
  **if** $u \notin L_2$ **then return** *false*;
**return** *true*;

**Theorem 7.5.** *Let $\mathcal{G}$ be a CFG and let $L_2$ be a language such that there exists a decidable $L_2$-consistent wqo on $\Sigma^*$. Then,* `CFGIncW` *decides the inclusion $\mathcal{L}(\mathcal{G}) \subseteq L_2$.*

Let us compare Theorem 5.4 with Theorem 7.5. The former had the additional hypothesis (i) that the membership problem in $L_2$ is decidable. Such condition is de facto true for Theorem 7.5 since a quasiorder is a $L_2$-consistent wqo iff $L_2$ is regular (as proved by De Luca and Varricchio [11, Theorem 2.1]).

**Myhill Quasiorder** Given a language $L$ over $\Sigma$, define the following quasiorder on $\Sigma^*$:

$$u \leqq_L v \iff r^L(u) \subseteq r^L(v)$$

where

$$r^L(u) \triangleq \{(x, y) \in \Sigma^* \times \Sigma^* \mid xuy \in L\} \ .$$

De Luca and Varricchio [11] call $\leqq_L$ the *Myhill quasiorder relative to $L$*.

**Lemma 7.6.** *Let $L \subseteq \Sigma^*$ be a language.*
(*a*) *$\leqq_L$ is a (left and right) $L$-consistent quasiorder. Moreover, $\leqq_L$ is well-quasiorder iff $L$ is regular. Also, if $L$ is regular then $\leqq_L$ is decidable.*
(*b*) *Let $\leq$ be a quasiorder. If $\leq$ is (left and right) $L$-consistent then $\rho_{\leqq_L} \subseteq \rho_{\leq}$.*

As a consequence, $\leqq_{L_2}$ is the most general (greatest for set inclusion) $L_2$-consistent quasiorder for which the above algorithm `CFGIncW` can be used to decide the language inclusion $\mathcal{L}(\mathcal{G}) \subseteq L_2$. However, deciding the Myhill quasiorder $\leqq_{L_2}$ can be easily shown to be as hard as the language inclusion problem (which is PSPACE-hard). In the following, we restrict ourselves to the problem $\mathcal{L}(\mathcal{G}) \subseteq \mathcal{L}(\mathcal{A})$ and consider a wqo based on $\mathcal{A}$ which yields an effective algorithm for deciding the inclusion.

**State-based Quasiorder** We define the quasiorder $\leq_{\mathcal{A}}$ on $\Sigma^*$ induced by a FA $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$ as follows:

$$u \leq_{\mathcal{A}} v \iff \mathrm{ctx}^{\mathcal{A}}(u) \subseteq \mathrm{ctx}^{\mathcal{A}}(v) \ . \tag{12}$$

where $\mathrm{ctx}^{\mathcal{A}}(u) \triangleq \{(q, q') \in Q^2 \mid u \in W_{q,q'}\}$.

**Lemma 7.7.** *Let $\mathcal{A}$ be an FA. Then $\leq_{\mathcal{A}}$ is a decidable $L(\mathcal{A})$-consistent well-quasiorder.*

Observe that for the Myhill quasiorder $\leq_{\mathcal{L}(\mathcal{A})}$ we have

$$u \leq_{\mathcal{L}(\mathcal{A})} v \quad \Leftrightarrow \quad \begin{array}{c} \bigcup\{W_{I,q} \times W_{q',F} \mid u \in W_{q,q'}\} \\ \rotatebox{90}{$\subseteq$} \\ \bigcup\{W_{I,q} \times W_{q',F} \mid v \in W_{q,q'}\} \end{array}$$

Note that $W_{I,\pi_1(A)} \times W_{\pi_2(A),F} \subseteq W_{I,\pi_1(B)} \times W_{\pi_2(B),F}$[3] holds for all $A \subseteq B \subseteq Q \times Q$. Thus, $u \leq_{\mathcal{A}} v \Rightarrow u \leq_{\mathcal{L}(\mathcal{A})} v$, hence $\rho_{\leq_{\mathcal{L}(\mathcal{A})}} \subseteq \rho_{\leq_{\mathcal{A}}}$, as stated by Lemma 7.6.

## 7.3. A Systematic Approach to the Antichain Algorithm

Consider a CFG $\mathcal{G} = \langle \mathcal{V}, \Sigma, P \rangle$ and a FA $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$ and let $\leq_{\mathcal{A}}$ be the $\mathcal{L}(\mathcal{A})$-consistent wqo defined in (12). Theorem 7.3 shows that the algorithm `CFGIncW` solves the inclusion problem $\mathcal{L}(\mathcal{G}) \subseteq \mathcal{L}(\mathcal{A})$ by working on the antichain abstraction $\langle \mathrm{AC}_{\langle \Sigma^*, \leq_{\mathcal{A}} \rangle}, \sqsubseteq \rangle$.

Similarly to the case of the quasiorder $\leq^l_{\mathcal{A}}$ (Section 6) it suffices to keep the sets $\mathrm{ctx}^{\mathcal{A}}(u)$ of pairs of states of $Q$ for each word $u$ instead of the words themselves. Therefore, we can systematically derive an algorithm analogous to `CFGIncW` but working on the antichain poset $\langle \mathrm{AC}_{\langle \wp(Q\times Q), \subseteq \rangle}, \sqsubseteq \rangle$ viewed as an abstraction of $\langle \mathrm{AC}_{\langle \Sigma^*, \leq_{\mathcal{A}} \rangle}, \sqsubseteq' \rangle$ (where $\sqsubseteq'$ is used for distinguishing the two orderings). Here, the abstraction map $\alpha_{\mathcal{A}} : \mathrm{AC}_{\langle \Sigma^*, \leq_{\mathcal{A}} \rangle} \to \mathrm{AC}_{\langle \wp(Q\times Q), \subseteq \rangle}$ and concretization map $\gamma_{\mathcal{A}} : \mathrm{AC}_{\langle \wp(Q\times Q), \subseteq \rangle} \to \mathrm{AC}_{\langle \Sigma^*, \leq_{\mathcal{A}} \rangle}$ are defined as follows:

$$\alpha_{\mathcal{A}}(X) \triangleq \{\mathrm{ctx}^{\mathcal{A}}(u) \mid u \in X\}$$
$$\gamma_{\mathcal{A}}(Y) \triangleq \lfloor \{u \in \Sigma^* \mid \mathrm{ctx}^{\mathcal{A}}(u) \in Y\} \rfloor$$

3. $\pi_1$ is the projection on the first component, $\pi_2$ on the second.

**Lemma 7.8.** $\langle \mathrm{AC}_{\langle \Sigma^*, \leq_{\mathcal{A}} \rangle}, \sqsubseteq' \rangle \underset{\alpha_{\mathcal{A}}}{\overset{\gamma_{\mathcal{A}}}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \wp(Q \times Q), \subseteq \rangle}, \sqsubseteq \rangle$ *is a QGC.*

As done in Section 6 we combine the word-based algorithm `CFGIncW` with the functions $\alpha_{\mathcal{A}}$ and $\gamma_{\mathcal{A}}$ in order to obtain a "state-based" algorithm deciding $\mathcal{L}(\mathcal{G}) \subseteq \mathcal{L}(\mathcal{A})$. Let us define the functions $\alpha \colon \wp(\Sigma^*) \to \mathrm{AC}_{\langle \wp(Q \times Q), \subseteq \rangle}$, $\gamma \colon \mathrm{AC}_{\langle \wp(Q \times Q), \subseteq \rangle} \to \wp(\Sigma^*)$ and $\mathrm{Fn}^{\mathcal{A}}_{\mathcal{G}}(\langle X_i \rangle_{i \in [0,n]}) : \wp(Q \times Q)^{|\mathcal{V}|} \to \wp(Q \times Q)^{|\mathcal{V}|}$ as follows:

$$\alpha(X) \triangleq \lfloor \{ \mathrm{ctx}^{\mathcal{A}}(u) \mid u \in X \} \rfloor$$
$$\gamma(Y) \triangleq \{ u \in \Sigma^* \mid \exists y \in Y, y \subseteq \mathrm{ctx}^{\mathcal{A}}(u) \}$$
$$\mathrm{Fn}^{\mathcal{A}}_{\mathcal{G}}(\langle X_i \rangle_{i \in [0,n]}) \triangleq \langle \lfloor \{ X_j \circ X_k \mid X_i \to X_j X_k \in P \} \rfloor \rangle_{i \in [0,n]}$$

where $X \circ Y \triangleq \{ (q, q') \mid (q, q'') \in X \wedge (q'', q') \in Y \}$ is standard composition of two relations $X, Y \subseteq Q \times Q$.

**Lemma 7.9.** *The following hold:*
(a) $\alpha = \alpha_{\mathcal{A}} \circ \alpha_{\leq_{\mathcal{A}}}$
(b) $\gamma = \gamma_{\leq_{\mathcal{A}}} \circ \gamma_{\mathcal{A}}$
(c) $\langle \wp(\Sigma^*), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \wp(Q \times Q), \subseteq \rangle}, \sqsubseteq \rangle$ *is a GC.*
(d) $\gamma \circ \alpha = \rho_{\leq_{\mathcal{A}}}$
(e) $\mathrm{Fn}^{\mathcal{A}}_{\mathcal{A}_1}(\vec{\boldsymbol{X}}) = \alpha_{\mathcal{A}} \circ \alpha_{\leq_{\mathcal{A}}} \circ \mathrm{Fn}_{\mathcal{A}_1} \circ \gamma_{\leq_{\mathcal{A}}} \circ \gamma_{\mathcal{A}}(\vec{\boldsymbol{X}})$ *for all* $\vec{\boldsymbol{X}} \in \alpha(\wp(\Sigma^*)^{|\mathcal{V}|})$

`CFGIncS`: State-based algorithm for $L(\mathcal{G}) \subseteq L(\mathcal{A})$

**Data:** CFG $\mathcal{G} = \langle \mathcal{V}, \Sigma, P \rangle$
**Data:** FA $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$
**Result:** Whether $L_1 \subseteq L_2$ holds

```
⟨Y_i⟩_{i∈[0,n]} := Kleene(λX⃗. ⌊b⃗⌋ ⊔ Fn^A_G(X⃗), ∅⃗);
forall y ∈ Y_0 do
    if y ∩ (I × F) = ∅ then return false;
return true;
```

**Theorem 7.10.** *Let $\mathcal{G}$ be a CFG and $\mathcal{A}$ be a FA. The algorithm* `CFGIncS` *decides $L(\mathcal{G}) \subseteq L(\mathcal{A})$.*

The resulting algorithm `CFGIncS` shares some features with two previous works. On the one hand, it is related to the work of Hofmann and Chen [21] which defines an abstract interpretation-based language inclusion decision procedure similar to ours. Even though Hofmann and Chen's algorithm and ours both manipulate sets of pairs of states of an automaton, their abstraction is based on equivalence relations and not quasiorders. Since quasiorders are strictly more general than equivalences our framework can be instantiated to a larger class of abstractions, most importantly coarser ones. Finally, it is worth pointing out that Hofmann and Chen's [21] approach aims at including languages of finite and also infinite words.

A second related work is that of Holík and Meyer [22] who define an antichain like algorithm manipulating sets of pairs of states. Holík and Meyer [22] start from the standard antichain algorithm for the automata case and rely on their expert knowledge about it to design an ad-hoc antichain algorithm for checking the inclusion of grammar languages into automata languages. By contrast, our approach is not ad-hoc but systematic, since we derive `CFGIncS` starting from the known Myhill quasiorder. The study of a precise relationship between Holík and Meyer's algorithm and `CFGIncS` is left as future work.

## 8. Equivalent Greatest Fixpoint Check

Let us recall [8, Theorem 4] that if $g \colon C \to C$ is a monotonic function on a complete lattice $\langle C, \leq, \vee, \wedge \rangle$ and $\widetilde{g} \colon C \to C$ is the right-adjoint function of $g$ then the following equivalence holds: for any $c, c' \in C$,

$$\mathrm{lfp}(\lambda x.\, c \vee g(x)) \leq c' \;\Leftrightarrow\; c \leq \mathrm{gfp}(\lambda y.\, c' \wedge \widetilde{g}(y)) \qquad (13)$$

This property has been exploited to derive equivalent invariance proof methods for programs [8]. In the following, we use it to derive an equivalent algorithm for deciding the inclusion $\mathcal{L}(\mathcal{G}) \subseteq \mathcal{L}(\mathcal{A})$ for a *linear* CFG $\mathcal{G}$ and a FA $\mathcal{A}$ which relies on the computation of a greatest fixpoint rather than a least fixpoint.

Given two languages $X, Y \in \wp(\Sigma^*)$, we define

$$XY \triangleq \{ xy \in \Sigma^* \mid \forall x \in X,\, y \in Y \},$$
$$X^{-1}Y \triangleq \{ u \in \Sigma^* \mid \forall x \in X,\, xu \in Y \},$$
$$XY^{-1} \triangleq \{ u \in \Sigma^* \mid \forall y \in Y,\, uy \in X \}\ .$$

Thus, $X^{-1}Y$ and $XY^{-1}$ are a universal generalization to languages of, resp., left and right quotients of words (recalled in Section 5.2). It turns out that concatenation and quotients give rise to the following equivalences.

**Lemma 8.1.** *For all $X, Y, Z \subseteq \Sigma^*$ and $w \in \Sigma^*$:*
(a) $X \subseteq ZY^{-1} \Leftrightarrow XY \subseteq Z \Leftrightarrow Y \subseteq X^{-1}Z$.
(b) $wY \subseteq Z \Leftrightarrow Y \subseteq w^{-1}Z$ *and* $Xw \subseteq Z \Leftrightarrow X \subseteq Zw^{-1}$.

For this greatest fixpoint based language inclusion check, we restrict ourselves to linear context-free languages. Without loss of generality [19], we assume that these languages are represented by *linear context-free grammars* where each production $X_i \to \beta$ is such that $\beta \in \Sigma\mathcal{V} \cup \mathcal{V}\Sigma \cup \Sigma \cup \{\epsilon\}$. For instance the grammar $\mathcal{G}$ on $\Sigma = \{a, b, c\}$ with rules $\{X_0 \to c,\ X_0 \to X_1 b,\ X_1 \to a X_0\}$ is a linear CFG specifying the language $\{a^n c b^n \mid n \geq 0\}$. Let us also recall that the set of linear context-free languages properly contains all regular languages.

Given a linear CFG $\mathcal{G} = \langle \mathcal{V}, \Sigma, P \rangle$, we define the function $\widetilde{\mathrm{Fn}}_{\mathcal{G}} : \wp(\Sigma^*)^{|\mathcal{V}|} \to \wp(\Sigma^*)^{|\mathcal{V}|}$ as follows:

$$\widetilde{\mathrm{Fn}}_{\mathcal{G}}(\langle X_i \rangle_{i \in [0,n]}) \triangleq \left\langle \begin{array}{l} \bigcap_{X_j \to a X_i \in P} a^{-1} X_j \ \cap \\ \bigcap_{X_k \to X_i b \in P} X_k b^{-1} \end{array} \right\rangle_{i \in [0,n]}$$

where, as usual, $\bigcap \varnothing = \Sigma^*$. It turns out that $\widetilde{\mathrm{Fn}}_{\mathcal{G}}$ is the adjoint of $\mathrm{Fn}_{\mathcal{G}}$.

**Lemma 8.2.** *If $\mathcal{G}$ is a linear CFG then for all $\vec{\boldsymbol{X}}, \vec{\boldsymbol{Y}} \in \wp(\Sigma^*)^{|\mathcal{V}|}$, $\mathrm{Fn}_{\mathcal{G}}(\vec{\boldsymbol{X}}) \subseteq \vec{\boldsymbol{Y}} \Leftrightarrow \vec{\boldsymbol{Y}} \subseteq \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\vec{\boldsymbol{X}})$.*

Hence, by applying the adjunction (13), it turns out that:

$$\begin{aligned} \mathcal{L}(\mathcal{G}) \subseteq L_2 \Leftrightarrow \\ \mathrm{lfp}(\lambda \overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}})) \subseteq \overrightarrow{\boldsymbol{L_2}}^{X_0} \Leftrightarrow \\ \overrightarrow{\boldsymbol{b}} \subseteq \mathrm{gfp}(\overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}})) \end{aligned} \tag{14}$$

Assuming for now that the Kleene iterates of the greatest fixpoint computation $\mathrm{gfp}(\overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}}))$ are finitely many, we define the following algorithm for the inclusion $\mathcal{L}(\mathcal{G}) \subseteq L_2$:

(1) compute the Kleene iterates of $\mathrm{gfp}(\overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}}))$;
(2) check whether the output vector $\langle Y_i \rangle_{i \in [0,n]} \in \wp(\Sigma^*)^{|\mathcal{V}|}$ is such that for all $i$, $b_i \subseteq Y_i$, where $\langle b_i \rangle_{i \in [0,n]} = \overrightarrow{\boldsymbol{b}}$.

The regularity of $L_2$ together with the basic property of regular languages of being closed under intersections and quotients shows that each Kleene iterate is a regular language and computable. Also, since, by definition, each $b_i$ is a finite set of words, the final check can be simply implemented by resorting to membership queries in $Y_i$ where $\langle Y_i \rangle_{i \in [0,n]}$ is the greatest fixpoint. To the best of our knowledge, the above algorithm has never been described in the literature before.

Next, we discharge the fundamental assumption on which the previous algorithm depends on: the Kleene iterates of $\mathrm{gfp}(\overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}}))$ are finitely many. To show this, we proceed as follows. First, we consider an abstract version of the greatest fixpoint computation for a closure operator such that the Kleene iterates thereof are finitely many. This closure operator will be $\rho_{\leq_{\mathcal{A}}}$ where $L_2 = \mathcal{L}(\mathcal{A})$ and we will show that $\rho_{\leq_{\mathcal{A}}}$ is *forward complete* for $\lambda \overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}})$. Forward completeness of abstract interpretations [17] is different from backward completeness already used in the previous sections. In particular, as a consequence of having a forward complete abstraction, it turns out that the Kleene iterates of the concrete and abstract greatest fixpoint computations coincide. The intuition here is that this forward complete closure $\rho_{\leq_{\mathcal{A}}}$ allows us to disclose the property that every Kleene iterate of $\mathrm{gfp}(\overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}}))$ belongs to the image of the closure $\rho_{\leq_{\mathcal{A}}}$, i.e., every Kleene iterate is a language which is $\leq_{\mathcal{A}}$-upward closed. A similar phenomenon occurs in well-structured transition systems [1], [16].

Let us now describe in detail this abstraction. A closure $\rho \in \mathrm{uco}(C)$ on a concrete domain $C$ is forward complete for a monotonic function $f : C \to C$ if $\rho f \rho = f \rho$. The intuition here is that forward completeness means that no loss of precision is accumulated when the output of the computations of $f\rho$ is approximated by $\rho$. Dually to the case of backward completeness, forward completeness implies that $\mathrm{gfp}(f) = \mathrm{gfp}(f\rho) = \mathrm{gfp}(\rho f \rho)$ holds, when these greatest fixpoints exist (this is the case, e.g., when $C$ is a complete lattice). It turns out that forward and backward completeness are linked by a duality on the function $f$.

**Lemma 8.3 ([17, Corollary 1]).** *Let $\langle C, \leq_C \rangle$ be a complete lattice and assume that $f \colon C \to C$ admits a right-adjoint $\widetilde{f} \colon C \to C$, i.e., $f(c) \leq_C c' \Leftrightarrow c \leq_C \widetilde{f}(c')$ always holds. Then, $\rho$ is backward complete for $f$ iff $\rho$ is forward complete for $\widetilde{f}$.*

Thus, by Lemma 8.3, in the following result instead of assuming the hypotheses implying that a closure $\rho$ is forward complete for $\widetilde{\mathrm{Fn}}_{\mathcal{G}}$ we assume the hypotheses which guarantee that $\rho$ is backward complete for its adjoint $\mathrm{Fn}_{\mathcal{G}}$.

**Theorem 8.4.** *Let $\mathcal{G} = \langle \mathcal{V}, \Sigma, P \rangle$ be a linear CFG and let $\mathcal{A}$ be an FA with $L_2 = \mathcal{L}(\mathcal{A})$. If $\rho \in \mathrm{uco}(\wp(\Sigma^*))$ satisfies:*
*(1) $\rho(L_2) = L_2$;*
*(2) $\rho$ is backward complete for $\lambda X.\, aX$ and $\lambda X.\, Xa$ for all $a \in \Sigma$*
*then $\mathcal{L}(\mathcal{G}) \subseteq L_2$ iff $\overrightarrow{\boldsymbol{b}} \subseteq \mathrm{gfp}(\lambda \overrightarrow{\boldsymbol{X}}.\, \rho(\overrightarrow{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}})))$. Moreover, the Kleene iterates coincide in lockstep with those of $\mathrm{gfp}(\lambda \overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}}))$.*

As announced, we can now establish that the Kleene iterates of $\mathrm{gfp}(\lambda \overrightarrow{\boldsymbol{Y}}.\, \overrightarrow{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\overrightarrow{\boldsymbol{Y}}))$ are finitely many. Let $L_2 = \mathcal{L}(\mathcal{A})$, for some FA $\mathcal{A}$, and consider the corresponding state-based quasiorder $\leq_{\mathcal{A}}$ on $\Sigma^*$ as defined in (12). Lemma 7.7 tells us that $\leq_{\mathcal{A}}$ is a $L_2$-consistent wqo. Furthermore, since $Q$ is finite we have that both $\leq_{\mathcal{A}}$ and $(\leq_{\mathcal{A}})^{-1}$ are wqos, so that, in turn, $\langle \rho_{\leq_{\mathcal{A}}}, \subseteq \rangle$ is both ACC and DCC. The definition of $\leq_{\mathcal{A}}$ shows that every chain in $\langle \rho_{\leq_{\mathcal{A}}}, \subseteq \rangle$ has at most $2^{|Q|^2}$ elements. This means that if we compute $2^{|Q|^2}$ Kleene iterates then we have necessarily computed the greatest fixpoint. It follows from the DCC property that the iterates of $\mathrm{gfp}(\lambda \overrightarrow{\boldsymbol{X}}.\, \rho_{\leq_{\mathcal{A}_2}}(\overrightarrow{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\overrightarrow{\boldsymbol{X}})))$ are finitely many, hence so are the iterates of $\mathrm{gfp}(\lambda \overrightarrow{\boldsymbol{Y}}.\, \overrightarrow{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\overrightarrow{\boldsymbol{Y}}))$ because they go in lockstep as proved in Theorem 8.4.

## 9. Conclusion

We believe we have only scratched the surface of the use of well-quasiorders on words for solving language inclusion problems. Future directions include leveraging well-quasiorders for infinite words [3], [26] to shed new light on the inclusion problem between $\omega$-regular languages. Our results could also be extended to inclusion of tree languages by relying on the extensions of Myhill-Nerode theorems for tree languages [25].

Another interesting topic for future work is the enhancement of quasiorders using simulation relations. Even though we already showed in this paper that simulations can be used to refine our language inclusion algorithms, we are not on par with the thoughtful use of simulation relations made by Abdulla et al. [2] and Bonchi and Pous [5].

Finally, let us mention that the correspondence between least and greatest fixpoint based inclusion checks assuming complete abstractions was studied by Bonchi et al. [4] with the aim of formally connecting sound up-to techniques and complete abstract interpretations. Possible developments include the study of our abstract interpretation-based algorithms for language inclusion from the point of view of sound up-to techniques.

## References


[1] P. A. Abdulla, K. Cerans, B. Jonsson, and Y.-K. Tsay, "General decidability theorems for infinite-state systems," in *LICS '96: Proc. 11th Annual IEEE Symp. on Logic in Computer Science*. IEEE Computer Society, 1996, pp. 313–321.

[2] P. A. Abdulla, Y.-F. Chen, L. Holík, R. Mayr, and T. Vojnar, "When simulation meets antichains," in *Proceedings of the 16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'10)*. Springer Berlin Heidelberg, 2010, pp. 158–174. [Online]. Available: https://doi.org/10.1007/978-3-642-12002-2_14

[3] A. Arnold, "A syntactic congruence for rational $\omega$-languages," *Theoretical Computer Science*, vol. 39, pp. 333–335, Jan. 1985. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0304397585901483

[4] F. Bonchi, P. Ganty, R. Giacobazzi, and D. Pavlovic, "Sound up-to techniques and complete abstract domains," in *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science - LICS '18*. ACM Press, 2018. [Online]. Available: https://doi.org/10.1145/3209108.3209169

[5] F. Bonchi and D. Pous, "Checking nfa equivalence with bisimulations up to congruence," in *Proceedings of the 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, ser. POPL'13. New York, NY, USA: ACM, 2013, pp. 457–468. [Online]. Available: http://doi.acm.org/10.1145/2429069.2429124

[6] N. Chomsky, "On certain formal properties of grammars," *Information and Control*, vol. 2, no. 2, pp. 137–167, 1959.

[7] P. Cousot, "Méthodes itératives de construction et d'approximation de points fixes d'opérateurs monotones sur un treillis, analyse sémantique de programmes (in French)," Thèse d'État ès sciences mathématiques, Université Joseph Fourier, Grenoble, France, 21 March 1978.

[8] ——, "Partial completeness of abstract fixpoint checking," in *Proceedings of the 4th International Symposium on Abstraction, Reformulation, and Approximation*, ser. SARA'02. London, UK, UK: Springer-Verlag, 2000, pp. 1–25. [Online]. Available: https://doi.org/10.1007/3-540-44914-0_1

[9] P. Cousot and R. Cousot, "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints," in *Proceedings of the 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL'77)*. New York, NY, USA: ACM, 1977, pp. 238–252. [Online]. Available: http://doi.acm.org/10.1145/512950.512973

[10] ——, "Systematic design of program analysis frameworks," in *Proceedings of the 6th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL'79)*. New York, NY, USA: ACM, 1979, pp. 269–282. [Online]. Available: http://doi.acm.org/10.1145/567752.567778

[11] A. de Luca and S. Varricchio, "Well quasi-orders and regular languages," *Acta Informatica*, vol. 31, no. 6, pp. 539–557, Jun 1994. [Online]. Available: https://doi.org/10.1007/BF01213206

[12] ——, *Finiteness and Regularity in Semigroups and Formal Languages*, 1st ed. Springer Publishing Company, Incorporated, 2011.

[13] M. De Wulf, L. Doyen, T. A. Henzinger, and J. F. Raskin, "Antichains: A new algorithm for checking universality of finite automata," in *Proceedings of the 18th International Conference on Computer Aided Verification*, ser. CAV'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 17–30. [Online]. Available: http://dx.doi.org/10.1007/11817963_5

[14] M. De Wulf, "From timed models to timed implementations," Ph.D. dissertation, Université libre de Bruxelles, Faculté des Sciences – Informatique, Bruxelles, Dec. 2006. [Online]. Available: http://hdl.handle.net/2013/ULB-DIPOT:oai:dipot.ulb.ac.be:2013/210797

[15] A. Ehrenfeucht, D. Haussler, and G. Rozenberg, "On regularity of context-free languages," *Theoretical Computer Science*, vol. 27, no. 3, pp. 311–332, Jan. 1983.

[16] A. Finkel and P. Schnoebelen, "Well-structured transition systems everywhere!" *Theoretical Computer Science*, vol. 256, no. 1-2, pp. 63–92, apr 2001. [Online]. Available: https://doi.org/10.1016/s0304-3975(00)00102-x

[17] R. Giacobazzi and E. Quintarelli, "Incompleteness, counterexamples, and refinements in abstract model-checking," in *Proceedings of the 8th Static Analysis Symposium (SAS'01), LNCS vol. 2126*, P. Cousot, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 356–373. [Online]. Available: https://doi.org/10.1007/3-540-47764-0_20

[18] R. Giacobazzi, F. Ranzato, and F. Scozzari, "Making abstract interpretations complete," *J. ACM*, vol. 47, no. 2, pp. 361–416, Mar. 2000. [Online]. Available: http://doi.acm.org/10.1145/333979.333989

[19] S. Ginsburg and H. G. Rice, "Two families of languages related to algol," *J. ACM*, vol. 9, no. 3, pp. 350–371, Jul. 1962. [Online]. Available: http://doi.acm.org/10.1145/321127.321132

[20] P. Hofman, R. Mayr, and P. Totzke, "Decidability of weak simulation on one-counter nets," in *Proceedings of the 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, ser. LICS '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 203–212. [Online]. Available: http://dx.doi.org/10.1109/LICS.2013.26

[21] M. Hofmann and W. Chen, "Abstract interpretation from büchi automata," in *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL'14) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'14)*. ACM Press, 2014. [Online]. Available: https://doi.org/10.1145/2603088.2603127

[22] L. Holík and R. Meyer, "Antichains for the verification of recursive programs," in *Networked Systems*. Springer International Publishing, 2015, pp. 322–336. [Online]. Available: https://doi.org/10.1007/978-3-319-26850-7_22

[23] J. E. Hopcroft and J. D. Ullman, *Formal Languages and Their Relation to Automata*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1969.

[24] P. Janar, J. Esparza, and F. Moller, "Petri nets and regular processes," *Journal of Computer and System Sciences*, vol. 59, no. 3, pp. 476 – 503, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022000099916434

[25] D. Kozen, "On the myhill-nerode theorem for trees," *Bulletin of the EATCS*, vol. 47, pp. 170–173, 1992.

[26] M. Ogawa, "Well-quasi-orders and regular $\omega$-languages," *Theoretical Computer Science*, vol. 324, no. 1, pp. 55–60, Sep. 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397504002075

[27] J. Sakarovitch, *Elements of Automata Theory*. Cambridge University Press, 2009.

## Appendix A.
## Closures and Galois Connections

Let us recall some basic notions on closure operators and Galois connections commonly used in abstract interpretation (see, e.g., [7], [10], [18]). Let $\langle C, \leq_C, \vee, \wedge\rangle$ be a complete lattice. An *upper closure operator*, or simply *closure*, on $\langle C, \leq_C\rangle$ is a function $\rho : C \to C$ which is:

- *Monotone*: $x \leq_C y \Rightarrow \rho(x) \leq_C \rho(y)$ for all $x, y \in C$;
- *Idempotent*: $\rho(\rho(x)) = \rho(x)$ for all $x \in C$;
- *Extensive*: $x \leq_C \rho(x)$ for all $x \in C$

The set of all upper closed operators on $C$ is denoted by $\mathrm{uco}(C)$. One useful property of closures states that for all $X \subseteq C$,

$$\rho(\vee X) = \rho(\vee \rho(X)) \quad \text{and} \quad \wedge\rho(X) = \rho(\wedge\rho(X)) \ .$$

Given two closures $\rho, \rho' \in \mathrm{uco}(C)$, $\rho$ is a *coarser* abstraction than $\rho'$ iff the image of $\rho$ is a subset of the image of $\rho'$, i.e. $\rho \subseteq \rho'$, and this happens iff for any $x \in C$, $\rho'(x) \leq_C \rho(x)$.

**Lemma A.1.** *Let $\langle C, \leq_C\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \leq_A\rangle$ be a GC between a poset and a qoset (or viceversa). Then the following properties hold:*
(a) *$x \leq_C \gamma \circ \alpha(x)$ and $\alpha \circ \gamma(y) \leq_A y$.*
(b) *$\alpha$ and $\gamma$ are monotonic functions.*
(c) *If $A$ is a poset then $\alpha = \alpha \circ \gamma \circ \alpha$. If $C$ is a poset then $\gamma = \gamma \circ \alpha \circ \gamma$.*

*Proof:* Assume that $\langle A, \leq_A\rangle$ is a poset (the case where $\langle C, \leq_C\rangle$ is a poset is dual).
(a): $\alpha(x) \leq_A \alpha(x) \Leftrightarrow x \leq_C \gamma(\alpha(x))$ and $\alpha(\gamma(y)) \leq_A y \Leftrightarrow \gamma(y) \leq_C \gamma(y)$.
(b): Assume that $c \leq_C c'$. Thus, $c \leq_C c' \leq_C \gamma(\alpha(c'))$, hence $c \leq_C \gamma(\alpha(c')) \Rightarrow \alpha(c) \leq_A \alpha(c')$.
(c): $c \leq_C \gamma(\alpha(c)) \Rightarrow \alpha(c) \leq_A \alpha(\gamma(\alpha(c)))$. Also, $\alpha(\gamma(\alpha(c))) \leq_A \alpha(c) \Leftrightarrow \gamma(\alpha(c)) \leq_C \gamma(\alpha(c))$. Thus, since $\langle A, \leq_A\rangle$ is a poset, $\alpha(\gamma(\alpha(c))) = \alpha(c)$. □

**Lemma A.2.** *Let $\langle C, \leq_C\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \leq_A\rangle$ be a GC between complete lattices. If $f : C \to C$ is a monotonic function then $\gamma(\mathrm{lfp}(\alpha \circ f \circ \gamma)) = \mathrm{lfp}(\gamma \circ \alpha \circ f)$.*

*Proof:* First we show that $\gamma(\mathrm{lfp}(\alpha f \gamma)) \geq_C \mathrm{lfp}(\gamma\alpha f)$.

$$\begin{aligned} \mathrm{lfp}(\gamma\alpha f) \leq_C \gamma(\mathrm{lfp}(\alpha f\gamma)) &\Leftarrow \\ \gamma\alpha f(\gamma(\mathrm{lfp}(\alpha f\gamma))) \leq_C \gamma(\mathrm{lfp}(\alpha f\gamma)) &\Leftrightarrow \\ \gamma(\mathrm{lfp}(\alpha f\gamma)) \leq_C \gamma(\mathrm{lfp}(\alpha f\gamma)) &\ . \end{aligned}$$

Then, we prove that $\gamma(\mathrm{lfp}(\alpha f\gamma)) \leq_C \mathrm{lfp}(\gamma\alpha f)$.

$$\begin{aligned} \gamma(\mathrm{lfp}(\alpha f\gamma)) \leq_C \mathrm{lfp}(\gamma\alpha f) &\Leftrightarrow \\ \gamma(\mathrm{lfp}(\alpha f\gamma)) \leq_C \gamma\alpha(\mathrm{lfp}(\gamma\alpha f)) &\Leftarrow \\ \mathrm{lfp}(\alpha f\gamma) \leq_A \alpha(\mathrm{lfp}(\gamma\alpha f)) &\Leftrightarrow \\ \alpha f\gamma(\alpha\,\mathrm{lfp}(\gamma\alpha f)) \leq_A \alpha(\mathrm{lfp}(\gamma\alpha f)) &\Leftrightarrow \\ \alpha f(\mathrm{lfp}(\gamma\alpha f)) \leq_A \alpha(\mathrm{lfp}(\gamma\alpha f)) &\Leftrightarrow \\ \gamma\alpha f(\mathrm{lfp}(\gamma\alpha f)) \leq_C \gamma\alpha(\mathrm{lfp}(\gamma\alpha f)) &\Leftrightarrow \\ \mathrm{lfp}(\gamma\alpha f) \leq_C \gamma\alpha(\mathrm{lfp}(\gamma\alpha f)) &\ . \end{aligned}$$

□

## Appendix B.
## Right Monotonicity

In this section we show results equivalent to the ones from Section 4 but requiring right (instead of left) monotonicity. Let $\mathcal{A} = \langle Q, \delta, I, F, \Sigma\rangle$ be an FA with $L = \mathcal{L}(\mathcal{A})$ and recall $W^{\mathcal{A}}_{I,q} \triangleq \{w \in \Sigma^* \mid \exists q_i \in I, q_i \overset{w}{\rightsquigarrow} q\}$. It is easy to observe that $W_{I,q} = \delta^{\{\epsilon\}}_{\varnothing}(q \in I) \cup \bigcup_{a\in\Sigma, a\in W_{q',q}} W_{I,q'}a$ which induces the following sets of fixpoint equations on $\wp(\Sigma^*)$

$$\mathrm{Eqn}(\mathcal{A}) \triangleq \{X_q = \delta^{\{\epsilon\}}_{\varnothing}(q \in I) \cup \textstyle\bigcup_{a\in\Sigma, q' \overset{a}{\to} q} X_{q'}a \mid q \in Q\} \ .$$

We have that $\langle \wp(\Sigma^*)^{|Q|}, \subseteq, \cup, \cap\rangle$ is a (product) complete lattice and all functions in $\mathrm{Eqn}(\mathcal{A})$ (of type $\wp(\Sigma^*)^{|Q|} \to \wp(\Sigma^*)$) are monotonic so that the least fixpoint solution $\vec{\boldsymbol{Y}} = \langle Y_q\rangle_{q\in Q}$ of $\mathrm{Eqn}(\mathcal{A})$ do exist. It is easy to show that $Y_q = W^{\mathcal{A}}_{I,q}$ for every $q \in Q$, hence $\mathcal{L}(\mathcal{A}) = \bigcup_{q_i\in I} Y_{q_i}$ following (2).

**Example B.1.** Let us consider the automaton $\mathcal{A}$ of Figure 1. The set of equations induced by $\mathcal{A}$ are as follows:

$$\mathrm{Eqn}(\mathcal{A}) = \begin{cases} X_1 = \{\epsilon\} \cup X_1 a \cup X_2 a \\ X_2 = \varnothing \cup X_1 b \cup X_2 b \end{cases} . \qquad \Diamond$$

Let us introduce the vector $\overrightarrow{\boldsymbol{\epsilon}}^I \in \wp(\Sigma^*)^{|Q|}$ and the function $\mathrm{Post}_{\mathcal{A}} \colon \wp(\Sigma^*)^{|Q|} \to \wp(\Sigma^*)^{|Q|}$, used to formalize the fixpoint equations in $\mathrm{Eqn}(\mathcal{A})$:

$$\overrightarrow{\boldsymbol{\epsilon}}^I \triangleq \langle \delta_{\varnothing}^{\{\epsilon\}}(q \in I) \rangle_{q \in Q} \qquad\qquad \mathrm{Post}_{\mathcal{A}}(\langle X_q \rangle_{q \in Q}) \triangleq \langle \textstyle\bigcup_{a \in \Sigma, q' \xrightarrow{a} q} X_{q'} a \rangle_{q \in Q} \ .$$

Thus, we have

$$\langle W_{I,q}^{\mathcal{A}} \rangle_{q \in Q} = \mathrm{lfp}(\lambda \overrightarrow{\boldsymbol{X}}.\ \overrightarrow{\boldsymbol{\epsilon}}^I \cup \mathrm{Post}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}})) \tag{15}$$

By equality (2), $\mathcal{L}(\mathcal{A})$ is the union of the languages of $\mathrm{lfp}(\lambda \overrightarrow{\boldsymbol{X}}.\ \overrightarrow{\boldsymbol{\epsilon}}^I \cup \mathrm{Post}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}}))$ for the components associated to $F$.

**Example B.2.** Consider again the automaton $\mathcal{A}$ from Figure 1. The fixpoint equations induced by $\mathcal{A}$ are as follows:

$$\begin{pmatrix} W_{q_1,q_1}^{\mathcal{A}} \\ W_{q_1,q_2}^{\mathcal{A}} \end{pmatrix} = \mathrm{lfp}\left( \lambda \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} . \begin{pmatrix} \{\epsilon\} \cup X_1 a \cup X_2 a \\ \varnothing \cup X_1 b \cup X_2 b \end{pmatrix} \right)$$

The fixpoint is $\begin{pmatrix} W_{q_1,q_1}^{\mathcal{A}} \\ W_{q_1,q_2}^{\mathcal{A}} \end{pmatrix} = \begin{pmatrix} (a + (b^+ a))^* \\ a^* b (b + a^+ b)^* \end{pmatrix}$, hence $\mathcal{L}(\mathcal{A}) = (a + (b^+ a))^*$. $\Diamond$

Let us go back to the language inclusion problem $\mathcal{L}(\mathcal{A}) \subseteq L_2$ where $\mathcal{A}$ is an FA $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$. We formalize the language $L_2$ as the following vector in $\wp(\Sigma^*)^{|Q|}$: $\overrightarrow{\boldsymbol{L_2}}^F \triangleq \langle \delta_{\Sigma^*}^{L_2}(q \in F) \rangle_{q \in Q}$. Thus

$$\begin{aligned} \mathcal{L}(\mathcal{A}) \subseteq L_2 \Leftrightarrow &\quad \text{[by (2)]} \\ \forall q \in F, W_{I,q}^{\mathcal{A}} \subseteq L_2 \Leftrightarrow &\quad \text{[definition of } \overrightarrow{\boldsymbol{L_2}}^F\text{]} \\ \langle W_{I,q}^{\mathcal{A}} \rangle_{q \in Q} \subseteq \overrightarrow{\boldsymbol{L_2}}^F \Leftrightarrow &\quad \text{[by (15)]} \\ \mathrm{lfp}(\lambda \overrightarrow{\boldsymbol{X}}.\ \overrightarrow{\boldsymbol{\epsilon}}^I \cup \mathrm{Post}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}})) \subseteq \overrightarrow{\boldsymbol{L_2}}^F & \end{aligned}$$

The following result is the equivalent of Theorem 4.3 for right concatenation and it can be proved in a similar manner.

**Theorem B.3.** *If $\rho \in \mathrm{uco}(\wp(\Sigma^*))$ and $\rho$ is backward complete for $\lambda X.\, Xa$ for all $a \in \Sigma$, then $\rho$ is backward complete for $\mathrm{Post}_{\mathcal{A}}$, and also for $\lambda \overrightarrow{\boldsymbol{X}}.\ \overrightarrow{\boldsymbol{\epsilon}}^I \cup \mathrm{Post}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}})$ for all FA $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$.*

Similarly, Theorem 4.5 can be adapted to the new equation system.

**Theorem B.4.** *Let $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$ be an FA and let $L_2$ be a language over $\Sigma$. Let $\langle \wp(\Sigma^*), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle D, \sqsubseteq \rangle$ be a GC where $\langle D, \sqsubseteq \rangle$ is a poset. If all of the following properties hold:*

*(i) $L_2 \in \gamma(D)$ and for every $a \in \Sigma$, $X \in \wp(\Sigma^*)$ we have $\alpha(Xa) = \alpha(\gamma\alpha(X)a)$.*
*(ii) $(D, \sqsubseteq, \sqcup)$ is an effective domain, meaning that: $(D, \sqsubseteq)$ is ACC, every element of $D$ has a finite representation, $\sqsubseteq$ is decidable and $\sqcup$ is a computable binary lub.*
*(iii) There is an algorithm, say $\mathrm{Post}^\sharp(\overrightarrow{\boldsymbol{X}})$, computing $\alpha(\mathrm{Post}_{\mathcal{A}}(\gamma(\overrightarrow{\boldsymbol{X}})))$.*
*(iv) There is an algorithm, say $\epsilon^\sharp$, computing $\alpha(\overrightarrow{\boldsymbol{\epsilon}}^I)$.*
*(v) There is an algorithm, say $\mathrm{Incl}^{\sharp\sharp}(\overrightarrow{\boldsymbol{X}})$, deciding the abstract inclusion $\overrightarrow{\boldsymbol{X}} \sqsubseteq \alpha(\overrightarrow{\boldsymbol{L_2}}^F)$ for every vector $\overrightarrow{\boldsymbol{X}} \in \alpha(\wp(\Sigma^*)^{|Q|})$.*

*Then, the following algorithm decides whether $\mathcal{L}(\mathcal{A}) \subseteq L_2$:*

$\langle Y_q \rangle_{q \in Q} := \texttt{Kleene}(\lambda \overrightarrow{\boldsymbol{X}}.\, \epsilon^\sharp \sqcup \mathrm{Post}^\sharp(\overrightarrow{\boldsymbol{X}}), \overrightarrow{\boldsymbol{\varnothing}})$;
**return** $\mathrm{Incl}^\sharp(\langle Y_q \rangle_{q \in Q})$;

**Corollary B.5.** *Theorem B.4 remains true when we have a QGC.*

We have shown in Lemma 5.2 that for every right $L$-consistent quasiorder there exists a backward complete $L$-closed closure operator $\rho_{\leq_L^r}$. Theorem 5.4 gives an algorithm for solving the language inclusion problem $\mathcal{L}(\mathcal{A}) \subseteq L_2$ whenever membership in $L_2$ is decidable and there exists a decidable left $L_2$-consistent wqo $\leqslant_{L_2}$. Next we show that Theorem 5.4 also holds when provided a decidable *right* $L_2$-consistent quasiorder.

`FAIncWr`: Word-based algorithm for $\mathcal{L}(\mathcal{A}) \subseteq L_2$

**Data:** FA $\mathcal{A} \triangleq \langle Q, \delta, I, F, \Sigma \rangle$ s.t. $L = \mathcal{L}(\mathcal{A})$
**Data:** Decision procedure for membership in $L_2$
**Data:** Right $L_2$-consistent decidable wqo $\leqslant^r_{L_2}$
**Result:** Whether $\mathcal{L}(\mathcal{A}) \subseteq L_2$ holds

1 $\langle Y_q \rangle_{q \in Q} := \texttt{Kleene}(\lambda \overrightarrow{\boldsymbol{X}}.\, \lfloor \overrightarrow{\boldsymbol{\epsilon}}^I \rfloor \sqcup \lfloor \mathrm{Post}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}}) \rfloor, \overrightarrow{\boldsymbol{\varnothing}})$;
2 **forall** $q \in F$ **do**
3     **forall** $u \in Y_q$ **do**
4         **if** $u \notin L_2$ **then return** *false*;
5 **return** *true*;

**Theorem B.6.** *Let $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$ be a FA and let $L_2$ be a language such that (i) membership $u \in L_2$ is decidable; and (ii) there exists a decidable right $L_2$-consistent wqo on $\Sigma^*$. Then,* `FAIncW` *decides the inclusion $\mathcal{L}(\mathcal{A}) \subseteq L_2$.*

*Proof:* We proceed by showing that all the premises of Theorem B.4 are satisfied for $\langle D, \sqsubseteq \rangle = \langle \mathrm{AC}_{\langle \Sigma^*, \leqslant^r_{L_2} \rangle}, \sqsubseteq \rangle$, $\alpha = \lfloor \cdot \rfloor$ and $\gamma = \rho_{\leqslant^r_{L_2}}$. Indeed we apply Corollary B.5 because $\langle \mathrm{AC}_{\langle \Sigma^*, \leqslant^r_{L_2} \rangle}, \sqsubseteq \rangle$ is a qoset so that we deal with a QGC rather than a GC.

(*i*) $\gamma(L_2) = L_2$ by Lemma 5.2 since $\gamma = \rho_{\leqslant^r_{L_2}}$. Furthermore

$$\begin{aligned} \alpha(Xa) = & \quad [\text{QGC}] \\ \alpha(\gamma(\alpha(Xa))) = & \quad [\text{L. 5.2 with } \rho_{\leqslant^r_{L_2}} = \gamma\alpha] \\ \alpha(\gamma(\alpha(\gamma(\alpha(X))a))) = & \quad [\text{QGC}] \\ \alpha(\gamma\alpha(X)a) & \ . \end{aligned}$$

(*ii*) It turns out that $\langle \mathrm{AC}_{\langle \Sigma^*, \leqslant^r_{L_2} \rangle}, \sqsubseteq \rangle$ is ACC because $\leqslant^r_{L_2}$ is a wqo. Moreover, the decidability of $\leqslant^r_{L_2}$ entails that $W_1 \sqcup W_2 \triangleq \lfloor W_1 \cup W_2 \rfloor$ is a computable binary lub, therefore $\langle \mathrm{AC}_{\langle \Sigma^*, \leqslant^r_{L_2} \rangle}, \sqsubseteq, \sqcup \rangle$ is an effective domain.

(*iii*) Let us first observe that by Lemma 5.2 (b) and Theorem 4.3, $\rho_{\leqslant^r_{L_2}}$ is backward complete for $\mathrm{Post}_{\mathcal{A}_1}$. Moreover, $\alpha(\mathrm{Post}_{\mathcal{A}}(\gamma(\overrightarrow{\boldsymbol{X}}))) = \lfloor \mathrm{Post}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}}) \rfloor$ since

$$\begin{aligned} \alpha(\mathrm{Post}_{\mathcal{A}}(\gamma(\overrightarrow{\boldsymbol{X}}))) = & \quad [\text{GC}] \\ \alpha\gamma\alpha(\mathrm{Post}_{\mathcal{A}}(\gamma(\overrightarrow{\boldsymbol{X}}))) = & \quad [\text{def. } \alpha \text{ and } \gamma] \\ \lfloor \rho_{\leqslant^r_{L_2}}(\mathrm{Post}_{\mathcal{A}}(\rho_{\leqslant^r_{L_2}}(\overrightarrow{\boldsymbol{X}}))) \rfloor = & \quad [\text{bw. completeness}] \\ \lfloor \rho_{\leqslant^r_{L_2}}(\mathrm{Post}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}})) \rfloor = & \quad [\text{GC}] \\ \lfloor \mathrm{Post}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}}) \rfloor & \ . \end{aligned}$$

This entails that $\alpha(\mathrm{Post}_{\mathcal{A}}(\gamma(\overrightarrow{\boldsymbol{X}})))$ is computable.

(*iv*) $\alpha(\{\epsilon\}) = \{\epsilon\}$ and $\alpha(\varnothing) = \varnothing$, hence $\alpha(\overrightarrow{\boldsymbol{\epsilon}}^I) = \lfloor \overrightarrow{\boldsymbol{\epsilon}}^I \rfloor$ is trivial to compute.

(*v*) Since $\alpha(\overrightarrow{\boldsymbol{L_2}}^F) = \alpha(\langle \delta^{L_2}_{\Sigma^*}(q \in F) \rangle_{q \in Q}$, the relation $\langle Y_q \rangle_{q \in Q} \sqsubseteq \alpha(\overrightarrow{\boldsymbol{L_2}}^F)$ trivially holds for all components with $q \notin F$. Therefore it suffices to check that $\forall q \in F$, $Y_q \sqsubseteq \alpha(L_2)$ is decidable. We have that:

$$\begin{aligned} Y_q \sqsubseteq \alpha(L_2) \Leftrightarrow & \quad [\text{defs. } \sqsubseteq \text{ and } \alpha] \\ \forall y \in Y_q, \exists x \in \lfloor L_2 \rfloor, x \leqslant^r_{L_2} y \Leftrightarrow & \quad [\lfloor X \rfloor \text{ is a minor set}] \\ \forall y \in Y_q, \exists x \in L_2, x \leqslant^r_{L_2} y \Leftrightarrow & \quad [\text{Def. } \rho_{\leqslant^r_{L_2}}(L_2)] \\ \forall y \in Y_q, y \in \rho_{\leqslant^r_{L_2}}(L_2) \Leftrightarrow & \quad [\rho_{\leqslant^r_{L_2}}(L_2) = L_2] \\ \forall y \in Y_q, y \in L_2 & \ . \end{aligned}$$

This latter condition coincides with the check performed by lines 2-5 of algorithm `FAIncWr` and is therefore decidable.

By Corollary B.5, algorithm `FAIncWr` solves $L_1 \subseteq L_2$. □

# Appendix C. Deferred Proofs

**Theorem 4.3.** *If $\rho \in \mathrm{uco}(\wp(\Sigma^*))$ is backward complete for $\lambda X.\, aX$ for all $a \in \Sigma$, then, for all FAs $\mathcal{A} = \langle Q, \delta, I, F, \Sigma \rangle$, $\rho$ is backward complete for $\mathrm{Pre}_{\mathcal{A}}$ and $\lambda \overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{\epsilon}}^F \cup \mathrm{Pre}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}})$.*

*Proof:* By definition we have that $\mathrm{Pre}_{\mathcal{A}}(\langle X_q\rangle_{q\in Q}) = \langle \bigcup_{a\in\Sigma, q\xrightarrow{a}q'} aX_{q'}\rangle_{q\in Q}$. Hence

$$\begin{aligned}
\rho(\textstyle\bigcup_{a\in\Sigma, q\xrightarrow{a}q'} aX_{q'}) &= \quad [\rho(\cup Y) = \rho(\cup\rho(Y))] \\
\rho(\textstyle\bigcup_{a\in\Sigma, q\xrightarrow{a}q'} \rho(aX_{q'})) &= \quad [\text{backward completeness of } \lambda X.\, aX] \\
\rho(\textstyle\bigcup_{a\in\Sigma, q\xrightarrow{a}q'} \rho(a\rho(X_{q'}))) &= \quad [\rho(\cup Y) = \rho(\cup\rho(Y))] \\
\rho(\textstyle\bigcup_{a\in\Sigma, q\xrightarrow{a}q'} a\rho(X_{q'}))\ &.
\end{aligned}$$

Thus, by a straightforward componentwise application on vectors in $\wp(\Sigma^*)^{|Q|}$, we obtain that $\rho$ is backward complete for $\mathrm{Pre}_{\mathcal{A}}$. Next, we turn to backward completeness of $\rho$ for $\lambda\overrightarrow{\boldsymbol{X}}.\, \overrightarrow{\boldsymbol{\epsilon}}^F \cup \mathrm{Pre}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}})$:

$$\begin{aligned}
\rho(\overrightarrow{\boldsymbol{\epsilon}}^F \cup \mathrm{Pre}_{\mathcal{A}}(\rho(\overrightarrow{\boldsymbol{X}}))) &= \quad [\rho(\cup Y) = \rho(\cup\rho(Y))] \\
\rho(\rho(\overrightarrow{\boldsymbol{\epsilon}}^F) \cup \rho(\mathrm{Pre}_{\mathcal{A}}(\rho(\overrightarrow{\boldsymbol{X}})))) &= \quad [\text{backward completeness of } \mathrm{Pre}_{\mathcal{A}}] \\
\rho(\rho(\overrightarrow{\boldsymbol{\epsilon}}^F) \cup \rho(\mathrm{Pre}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}}))) &= \quad [\rho(\cup Y) = \rho(\cup\rho(Y))] \\
\rho(\overrightarrow{\boldsymbol{\epsilon}}^F \cup\ \mathrm{Pre}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}}))\ &.
\end{aligned}$$

□

**Theorem 4.5.** *Let $\mathcal{A} = \langle Q, \delta, I, F, \Sigma\rangle$ be a FA and let $L_2$ be a language over $\Sigma$. Let $\langle \wp(\Sigma^*), \subseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle D, \sqsubseteq\rangle$ be a GC where $\langle D, \sqsubseteq\rangle$ is a poset. Assume that the following properties hold:*

*(i) $L_2 \in \gamma(D)$ and for every $a \in \Sigma$, $X \in \wp(\Sigma^*)$ we have $\alpha(aX) = \alpha(a\gamma\alpha(X))$.*
*(ii) $(D, \sqsubseteq, \sqcup)$ is an effective domain, meaning that: $(D, \sqsubseteq)$ is ACC, every element of $D$ has a finite representation, $\sqsubseteq$ is decidable and $\sqcup$ is a computable binary lub.*
*(iii) There is an algorithm, say $\mathrm{Pre}^\sharp(\overrightarrow{\boldsymbol{X}})$, which computes $\alpha(\mathrm{Pre}_{\mathcal{A}}(\gamma(\overrightarrow{\boldsymbol{X}})))$, for all $\overrightarrow{\boldsymbol{X}} \in \wp(\Sigma^*)^{|Q|}$.*
*(iv) There is an algorithm, say $\epsilon^\sharp$, computing $\alpha(\overrightarrow{\boldsymbol{\epsilon}}^F)$.*
*(v) There is an algorithm, say $\mathrm{Incl}^\sharp(\overrightarrow{\boldsymbol{X}})$, deciding the abstract inclusion $\overrightarrow{\boldsymbol{X}} \sqsubseteq \alpha(\overrightarrow{\boldsymbol{L_2}}^I)$, for every vector $\overrightarrow{\boldsymbol{X}} \in \alpha(\wp(\Sigma^*)^{|Q|})$.*

*Then, the following algorithm decides whether $\mathcal{L}(\mathcal{A}) \subseteq L_2$:*

$\langle Y_q\rangle_{q\in Q} := \texttt{Kleene}(\lambda\overrightarrow{\boldsymbol{X}}.\, \epsilon^\sharp \sqcup \mathrm{Pre}^\sharp(\overrightarrow{\boldsymbol{X}}), \overrightarrow{\boldsymbol{\varnothing}})$;
**return** $\mathrm{Incl}^\sharp(\langle Y_q\rangle_{q\in Q})$;

*Proof:* Let $\rho = \gamma \circ \alpha \in \mathrm{uco}(\wp(\Sigma^*))$. Then, it follows from property (i) that $L_2 \in \rho$ and $\rho(aX) = \rho(a\rho(X))$. Therefore

$$\begin{aligned}
\mathcal{L}(\mathcal{A}) \subseteq L_2 &\Leftrightarrow [\text{by (6)}] \\
\mathrm{lfp}(\lambda\overrightarrow{\boldsymbol{X}}.\, \rho(\overrightarrow{\boldsymbol{\epsilon}}^F \cup \mathrm{Pre}_{\mathcal{A}}(\overrightarrow{\boldsymbol{X}}))) \subseteq \overrightarrow{\boldsymbol{L_2}}^I &\Leftrightarrow [\text{L. A.2}] \\
\gamma(\mathrm{lfp}(\lambda\overrightarrow{\boldsymbol{X}}.\, \alpha(\overrightarrow{\boldsymbol{\epsilon}}^F) \sqcup \alpha(\mathrm{Pre}_{\mathcal{A}}(\gamma(\overrightarrow{\boldsymbol{X}}))))) \subseteq \overrightarrow{\boldsymbol{L_2}}^I &\Leftrightarrow [\text{GC}] \\
\mathrm{lfp}(\lambda\overrightarrow{\boldsymbol{X}}.\, \alpha(\overrightarrow{\boldsymbol{\epsilon}}^F) \sqcup \alpha(\mathrm{Pre}_{\mathcal{A}}(\gamma(\overrightarrow{\boldsymbol{X}})))) \sqsubseteq \alpha(\overrightarrow{\boldsymbol{L_2}}^I)\ &.
\end{aligned}$$

Since $(D, \sqsubseteq)$ is ACC, `Kleene` is an algorithm computing the least fixpoint. Properties (ii), (iii) and (iv) ensure that the `Kleene` iterates of $\lambda\overrightarrow{\boldsymbol{X}}.\, \alpha(\overrightarrow{\boldsymbol{\epsilon}}^F) \sqcup \alpha(\mathrm{Pre}_{\mathcal{A}}(\gamma(\overrightarrow{\boldsymbol{X}})))$ are computable and it is possible to check whether the iterates have reach a fixpoint. Property (v) ensures decidability of the required $\sqsubseteq$-check since all `Kleene` iterates are in $\alpha(\wp(\Sigma^*))^{|Q|}$. □

**Lemma 5.2.** *Let $L$ be a language over $\Sigma$ and $\leqslant_L$ be a left (resp. right) $L$-consistent quasiorder on $\Sigma^*$. Then,*
*(a) $\rho_{\leqslant_L}(L) = L$.*
*(b) $\rho_{\leqslant_L}$ is backward complete for $\lambda X.\, aX$ (resp. $\lambda X.\, Xa$) for all $a \in \Sigma$.*

*Proof:* We consider the left case, the right case is symmetric.

(*a*) The inclusion $L \subseteq \rho_{\leqslant_L}(L)$ holds because $\rho_{\leqslant_L}$ is an upper closure. Property (a) of Definition 5.1 entails $\rho_{\leqslant_L}(L) \subseteq L$.
(*b*) We prove that $\rho_{\leqslant_L}(aX) = \rho_{\leqslant_L}(a\rho_{\leqslant_L}(X))$ for every $a \in \Sigma$. Monotonicity of concatenation together with monotonicity and extensivity of $\rho_{\leqslant_L}$ imply that $\rho_{\leqslant_L}(aX) \subseteq \rho_{\leqslant_L}(a\rho_{\leqslant_L}(X))$ holds. For the reverse inclusion

$$\begin{aligned}
\rho_{\leqslant_L}(a\rho_{\leqslant_L}(X)) &= \quad [\text{definition of } \rho_{\leqslant_L}] \\
\rho_{\leqslant_L}(\{ay \mid \exists x \in X, x \leqslant_L y\}) &= \quad [\text{definition of } \rho_{\leqslant_L}] \\
\{z \mid \exists y, ay \leqslant_L z \wedge \exists x \in X, x \leqslant_L y\} &\subseteq \quad [\text{left monotonicity}] \\
\{z \mid \exists y, ay \leqslant_L z \wedge \exists x \in X, ax \leqslant_L ay\} &= \quad [\text{transitivity of } \leqslant_L] \\
\{z \mid \exists x \in X, ax \leqslant_L z\} &= \quad [\text{definition of } \rho_{\leqslant_L}] \\
\rho_{\leqslant_L}(aX)\ &.
\end{aligned}$$

□

**Theorem 5.3.** *Let $\langle \Sigma^*, \leqslant \rangle$ be a qoset.*
(a) $\langle \wp(\Sigma^*), \subseteq \rangle \underset{\alpha_\leqslant}{\overset{\gamma_\leqslant}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \Sigma^*, \leqslant \rangle}, \sqsubseteq \rangle$ *is a QGC.*
(b) $\gamma_\leqslant \circ \alpha_\leqslant = \rho_\leqslant$.

*Proof:* Since $\langle \wp(\Sigma^*), \subseteq \rangle$ is a poset and $\langle \mathrm{AC}_{\langle \wp(\Sigma^*), \leqslant \rangle}, \sqsubseteq \rangle$ is a qoset, property (a) holds iff the relation $\alpha_\leqslant(X) \sqsubseteq Y \Leftrightarrow X \subseteq \gamma_\leqslant(Y)$ holds.

$$\begin{aligned}
\alpha_\leqslant(X) \sqsubseteq Y \Leftrightarrow & \quad [\text{definition of } \sqsubseteq] \\
\forall z \in \alpha_\leqslant(X), \exists y \in Y,\ y \leqslant z \Leftrightarrow & \quad [\text{definitions of } \alpha_\leqslant \text{ and } \lfloor \cdot \rfloor] \\
\forall x \in X, \exists y \in Y\ y \leqslant x \Leftrightarrow & \quad [\text{definition of } \gamma_\leqslant] \\
\forall x \in X,\ x \in \gamma_\leqslant(Y) \Leftrightarrow & \quad [\text{definition of } \subseteq] \\
X \subseteq \gamma_\leqslant(Y)\ & .
\end{aligned}$$

Property (b) trivially holds by definition of $\alpha_\leqslant$ and $\gamma_\leqslant$. □

**Lemma 5.6.** *Let $\mathcal{A}$ be an FA. Then $\leq^l_{\mathcal{A}}$ and $\leq^r_{\mathcal{A}}$ are, resp., decidable left and right $\mathcal{L}(\mathcal{A})$-consistent wqos.*

*Proof:* Since, for every $u \in \Sigma^*$, $\mathrm{pre}^{\mathcal{A}}_u(F)$ is a finite and computable set, it turns out that $\leq^l_{\mathcal{A}}$ is a decidable well-quasiorder. Let us check that $\leq^l_{\mathcal{A}}$ is left $\mathcal{L}(A)$-consistent according to Definition 5.1 (a)-(b).

(a) By picking $x \in \mathcal{L}(\mathcal{A})$ and $y \notin \mathcal{L}(\mathcal{A})$ we have that $\mathrm{pre}^{\mathcal{A}}_x(F)$ contains some initial state while $\mathrm{pre}^{\mathcal{A}}_y(F)$ does not, hence $x \not\leq^l_{\mathcal{A}} y$.

(b) Let us check that $\leq^l_{\mathcal{A}}$ is left monotonic. Observe that $\mathrm{pre}^{\mathcal{A}}_x$ is a monotonic function and that

$$\mathrm{pre}^{\mathcal{A}}_{uv} = \mathrm{pre}^{\mathcal{A}}_u \circ \mathrm{pre}^{\mathcal{A}}_v \quad . \tag{16}$$

Therefore:

$$\begin{aligned}
x_1 \leq^l_{\mathcal{A}} x_2 \Rightarrow & \quad [\text{definition of } \leq^l_{\mathcal{A}}] \\
\mathrm{pre}^{\mathcal{A}}_{x_1}(F) \subseteq \mathrm{pre}^{\mathcal{A}}_{x_2}(F) \Rightarrow & \quad [\mathrm{pre}^{\mathcal{A}}_a \text{ is monotonic}] \\
\mathrm{pre}^{\mathcal{A}}_a(\mathrm{pre}^{\mathcal{A}}_{x_1}(F)) \subseteq \mathrm{pre}^{\mathcal{A}}_a(\mathrm{pre}^{\mathcal{A}}_{x_2}(F)) \Leftrightarrow & \quad [\text{by (16)}] \\
\mathrm{pre}^{\mathcal{A}}_{ax_1}(F) \subseteq \mathrm{pre}^{\mathcal{A}}_{ax_2}(F) \Leftrightarrow & \quad [\text{definition of } \leq^l_{\mathcal{A}}] \\
ax_1 \leq^l_{\mathcal{A}} ax_2\ & .
\end{aligned}$$

□

It can be similarly proved that $\leq^r_{\mathcal{A}}$ is a decidable right $\mathcal{L}(\mathcal{A})$-consistent quasiorder by relying on the fact that function $\mathrm{post}^{\mathcal{A}}_x$ is monotonic and $\mathrm{post}^{\mathcal{A}}_{uv} = \mathrm{post}^{\mathcal{A}}_v \circ \mathrm{post}^{\mathcal{A}}_u$.

**Lemma 5.7.** *Given a simulation relation $\preceq$ on $\mathcal{A}$, the right simulation-based qo $\preceq^r_{\mathcal{A}}$ is a decidable right $\mathcal{L}(\mathcal{A})$-consistent wqo.*

Recall the following definitions:

$$q \preceq q' \Leftrightarrow \Big( (q \in F \Rightarrow q' \in F) \wedge \Big( q \xrightarrow{a} q_1 \Rightarrow \Big( q' \xrightarrow{a} q_2 \wedge\ q_1 \preceq q_2 \Big) \Big) \Big)$$

$$X \preceq^{\forall\exists} Y \Leftrightarrow \forall x \in X, \exists y \in Y : x \preceq y, \qquad u \preceq^r_{\mathcal{A}} v \Leftrightarrow \mathrm{post}_u(I) \preceq^{\forall\exists} \mathrm{post}_v(I)$$

*Proof:*

Let $u \in \mathcal{L}(\mathcal{A})$ and $v \notin \mathcal{L}(\mathcal{A})$. Then $(F \cap \mathrm{post}_u(I)) \neq \varnothing$ while $(F \cap \mathrm{post}_v(I)) = \varnothing$ hence there exists $q \in (\mathrm{post}_u(F) \cap F)$ such that $q \preceq^r_{\mathcal{A}} q'$ for no $q' \in \mathrm{post}_v(F)$ since, by definition of *simulation* it would imply $q' \in (\mathrm{post}_v(F) \cap F)$, which contradicts the fact that $(F \cap \mathrm{post}_v(I)) = \varnothing$. Therefore $u \not\preceq^r_{\mathcal{A}} v$.

Next we show that $\preceq^r_{\mathcal{A}}$ is right monotonic, i.e. $u \preceq^r_{\mathcal{A}} v \Rightarrow ua \preceq^r_{\mathcal{A}} va$.

$$\begin{aligned}
u \preceq^r_{\mathcal{A}} v \Leftrightarrow & \quad [\text{definition of } \preceq^r_{\mathcal{A}}] \\
\mathrm{post}_u(I) \preceq^{\forall\exists} \mathrm{post}_v(I) \Leftrightarrow & \quad [\text{definition of } \preceq^{\forall\exists}] \\
\forall x \in \mathrm{post}_u(I), \exists y \in \mathrm{post}_v(I), x \preceq y \Rightarrow & \quad [\text{definition of } \preceq] \\
\forall x \xrightarrow{a} x' \text{ with } x \in \mathrm{post}_u(I),\ \exists y \xrightarrow{a} y' \text{ with } y \in \mathrm{post}_v(u), x' \preceq y' \Leftrightarrow & \quad [x \xrightarrow{a} x' \wedge x \in \mathrm{post}_u(I) \Leftrightarrow x' \in \mathrm{post}_{ua}(I)] \\
\forall x' \in \mathrm{post}_{ua}(I), \exists y' \in \mathrm{post}_{va}(I),\ x' \preceq y' \Leftrightarrow & \quad [\text{definition of } \preceq^{\forall\exists}] \\
\mathrm{post}_{ua}(I) \preceq^{\forall\exists} \mathrm{post}_{va}(I) \Leftrightarrow & \quad [\text{definition of } \preceq^r_{\mathcal{A}}] \\
ua \preceq^r_{\mathcal{A}} va\ & .
\end{aligned}$$

Therefore $\preceq^r_{\mathcal{A}}$ is a right $\mathcal{L}(\mathcal{A})$-consistent quasiorder.

Finally, since $\wp(Q)$ is finite, it follows that $\preceq^r_{\mathcal{A}}$ is a well-quasiorder and, since $\mathrm{post}_u(I)$ is finite and computable for every $u$, it follows that $\preceq^r_{\mathcal{A}}$ is decidable. $\square$

**Lemma 5.8.** *Given a OCN $\mathcal{O}$ together with a configuration $qn$, $\leq^r_{qn}$ is a right $T(qn)$-consistent decidable wqo.*

*Proof:* Well quasiordering follows from Dickson's Lemma [27, Section II.7.1.2].

Since the configuraton $qn$ if fixed, in what follows we omit the superscript $qn$ from sets $Z^{qn}_u$ for clarity. We conclude from the finiteness of $Z_u$ and $Z_v$ that $M_{Z_u}$ and $M_{Z_v}$ are computable, hence the ordering $\leq^r_{\mathcal{O}}$ is decidable. Let $u \in T(qn)$ and $v \notin T(qn)$ then $u \leq^r_{\mathcal{O}} v$ does not hold since $M_{Z_u}(q) \neq \bot$ for some $q \in Q$ but $M_{Z_v}(q) = \bot$ for all $q \in Q$ because $Z_v = \varnothing$. It remains to show that $u \leq^r_{\mathcal{O}} v$ implies $ua \leq^r_{\mathcal{O}} va$ for all $a \in \Sigma$. We proceed by contradiction. Assume $u \leq^r_{\mathcal{O}} v$ and $\exists q \in Q, M_{Z_{ua}}(q) > M_{Z_{ua}}(q)$. Then $m_1 \triangleq \max\{n \mid qn \in Z_{ua}\} > m_2 \triangleq \max\{n \mid qn \in Z_{va}\}$ and, therefore, $\forall (q', a, d, q) \in \delta$ we have $q'(m_1 - d) \in Z_u$ and $q'(m_2 - d) \in Z_v$. Since $m_1 - d > m_2 - d$ we have that $\max\{n \mid q'n \in Z_u\} > \max\{n \mid q'n \in Z_v\}$, which contradicts $u \leqslant^r_{\mathcal{O}}$. $\square$

**Lemma 6.1.** $\langle \mathrm{AC}_{\langle \Sigma^*, \leq^l_{\mathcal{A}_2} \rangle}, \sqsubseteq' \rangle \underset{\alpha_{\mathcal{A}_2}}{\overset{\gamma_{\mathcal{A}_2}}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \wp(Q_2), \subseteq \rangle}, \sqsubseteq \rangle$ *is a QGC.*

*Proof:*

$$
\begin{aligned}
\alpha_{\mathcal{A}_2}(X) \sqsubseteq Y \Leftrightarrow & \quad [\text{definition of } \sqsubseteq] \\
\forall z \in \alpha_{\mathcal{A}_2}(X), \exists y \in Y,\ y \subseteq z \Leftrightarrow & \quad [\text{definition of } \alpha_{\mathcal{A}_2}] \\
\forall v \in X, \exists y \in Y,\ y \subseteq \mathrm{pre}^{\mathcal{A}_2}_v(F_2) \Leftrightarrow & \quad [\text{definitions of } \gamma_{\mathcal{A}_2} \text{ and } \lfloor \cdot \rfloor] \\
\forall v \in X, \exists u \in \gamma_{\mathcal{A}_2}(Y),\ \mathrm{pre}^{\mathcal{A}_2}_u(F_2) \subseteq \mathrm{pre}^{\mathcal{A}_2}_v(F_2) \Leftrightarrow & \quad [\text{definition of } \leq^l_{\mathcal{A}_2}] \\
\forall v \in X, \exists u \in \gamma_{\mathcal{A}_2}(Y),\ u \leq^l_{\mathcal{A}_2} v \Leftrightarrow & \quad [\text{definition of } \sqsubseteq'] \\
X \sqsubseteq' \gamma_{\mathcal{A}_2}(Y)\ . &
\end{aligned}
$$

$\square$

**Lemma 6.2.** *The following hold:*

(a) $\alpha = \alpha_{\mathcal{A}_2} \circ \alpha_{\leq^l_{\mathcal{A}_2}}$
(b) $\gamma = \gamma_{\leq^l_{\mathcal{A}_2}} \circ \gamma_{\mathcal{A}_2}$
(c) $\langle \wp(\Sigma^*), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \wp(Q_2), \subseteq \rangle}, \sqsubseteq \rangle$ *is a GC.*
(d) $\gamma \circ \alpha = \rho_{\leq^l_{\mathcal{A}_2}}$
(e) $\mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\vec{\boldsymbol{X}}) = \alpha_{\mathcal{A}_2} \circ \alpha_{\leq^l_{\mathcal{A}_2}} \circ \mathrm{Pre}_{\mathcal{A}_1} \circ \gamma_{\leq^l_{\mathcal{A}_2}} \circ \gamma_{\mathcal{A}_2}(\vec{\boldsymbol{X}})$ *for all* $\vec{\boldsymbol{X}} \in \alpha(\wp(\Sigma^*)^{|Q_1|})$

Recall the following definitions

$$
\begin{aligned}
\alpha_{\leq^l_{\mathcal{A}_2}}(X) &\triangleq \lfloor X \rfloor & \gamma_{\leq^l_{\mathcal{A}_2}}(Y) &\triangleq \rho_{\leq^l_{\mathcal{A}_2}}(Y) \\
\alpha_{\mathcal{A}_2}(X) &\triangleq \{\mathrm{pre}^{\mathcal{A}_2}_u(F_2) \mid u \in X\} & \gamma_{\mathcal{A}_2}(Y) &\triangleq \lfloor \{u \in \Sigma^* \mid \mathrm{pre}^{\mathcal{A}_2}_u(F_2) \in Y\} \rfloor \\
\alpha(X) &\triangleq \lfloor \{\mathrm{pre}^{\mathcal{A}_2}_u(F_2) \mid u \in X\} \rfloor & \gamma(Y) &\triangleq \{u \in \Sigma^* \mid \exists y \in Y, y \subseteq \mathrm{pre}^{\mathcal{A}_2}_u(F_2)\}
\end{aligned}
$$

*Proof:*

(*a*)

$$
\begin{aligned}
\alpha_{\mathcal{A}_2}(\alpha_{\leq^l_{\mathcal{A}_2}}(X)) = & \quad [\text{definitions of } \alpha_{\leq^l_{\mathcal{A}_2}} \text{ and } \alpha_{\mathcal{A}_2}] \\
\{\mathrm{pre}^{\mathcal{A}_2}_u(F_2) \mid u \in \lfloor X \rfloor\} = & \quad [\text{definition of } \lfloor \cdot \rfloor] \\
\{\mathrm{pre}^{\mathcal{A}_2}_u(F_2) \mid u \in X \wedge \forall x \in X, x \not\leq^l_{\mathcal{A}_2} u\} = & \quad [\text{definition of } \leq^l_{\mathcal{A}_2}] \\
\{\mathrm{pre}^{\mathcal{A}_2}_u(F_2) \mid u \in X \wedge \forall x \in X, \mathrm{pre}^{\mathcal{A}_2}_x(F_2) \nsubseteq \mathrm{pre}^{\mathcal{A}_2}_u(F_2)\} = & \quad [\text{definition of } \lfloor \cdot \rfloor] \\
\lfloor \{\mathrm{pre}^{\mathcal{A}_2}_u(F_2) \mid u \in X\} \rfloor = & \quad [\text{definition of } \alpha] \\
\alpha(X)\ . &
\end{aligned}
$$

(*b*)

$$\begin{aligned}
\gamma_{\leq^l_{\mathcal{A}_2}}(\gamma_{\mathcal{A}_2}(Y)) = & \quad [\text{definition of } \gamma_{\mathcal{A}_2} \text{ and } \gamma_{\leq^l_{\mathcal{A}_2}}] \\
\rho_{\leq^l_{\mathcal{A}_2}}(\lfloor\{u \in \Sigma^* \mid \mathrm{pre}^{\mathcal{A}_2}_u(F_2) \in Y\}\rfloor) = & \quad [\text{definition of } \rho_{\leq^l_{\mathcal{A}_2}}] \\
\{x \in \Sigma^* \mid \exists y \in \lfloor\{u \in \Sigma^* \mid \mathrm{pre}^{\mathcal{A}_2}_u(F_2) \in Y\}\rfloor,\ y \leq^l_{\mathcal{A}_2} x\} = & \quad [\text{definition of } \lfloor\cdot\rfloor] \\
\{x \in \Sigma^* \mid \exists y \in \{u \in \Sigma^* \mid \mathrm{pre}^{\mathcal{A}_2}_u(F_2) \in Y\},\ y \leq^l_{\mathcal{A}_2} x\} = & \quad [\text{definition of } \leq^l_{\mathcal{A}_2}] \\
\{x \in \Sigma^* \mid \exists y \in \{u \in \Sigma^* \mid \mathrm{pre}^{\mathcal{A}_2}_u(F_2) \in Y\},\ \mathrm{pre}^{\mathcal{A}_2}_y(F_2) \subseteq \mathrm{pre}^{\mathcal{A}_2}_x(F_2)\} = & \\
\{x \in \Sigma^* \mid \exists y \in Y,\ y \subseteq \mathrm{pre}^{\mathcal{A}_2}_x(F_2)\} = & \quad [\text{definition of } \gamma] \\
\gamma(Y)\ . &
\end{aligned}$$

(*c*) This follows by composition of QGCs and by the fact that concrete and abstract domains are posets.
(*d*)

$$\begin{aligned}
\gamma(\alpha(X)) = & \quad [\text{definition of } \gamma] \\
\{u \in \Sigma^* \mid \exists y \in \alpha(X), y \subseteq \mathrm{pre}^{\mathcal{A}_2}_u(F_2)\} = & \quad [\text{definitions of } \alpha \text{ and } \lfloor\cdot\rfloor] \\
\{u \in \Sigma^* \mid \exists x \in X, \mathrm{pre}^{\mathcal{A}_2}_x(F_2) \subseteq \mathrm{pre}^{\mathcal{A}_2}_u(F_2)\} = & \quad [\text{definition of } \leq^l_{\mathcal{A}_2}] \\
\{u \in \Sigma^* \mid \exists x \in X, x \leq^l_{\mathcal{A}_2} u\} = & \quad [\text{definition of } \rho_{\leq^l_{\mathcal{A}_2}}] \\
\rho_{\leq^l_{\mathcal{A}_2}}(X)\ . &
\end{aligned}$$

(*e*) Due to properties (a) and (b) it suffices to show that $\alpha(\mathrm{Pre}_{\mathcal{A}_1}(\gamma(\overrightarrow{\boldsymbol{X}}))) = \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\overrightarrow{\boldsymbol{X}})$ for all $X$ in the image of $\alpha$:

$$\begin{aligned}
\alpha(\mathrm{Pre}_{\mathcal{A}_1}(\gamma(\overrightarrow{\boldsymbol{X}}))) = & \quad [\text{definition of } \mathrm{Pre}_{\mathcal{A}_1}] \\
\langle\alpha(\textstyle\bigcup_{a\in\Sigma, q\xrightarrow{a}_{\mathcal{A}_1} q'} a\gamma(\overrightarrow{\boldsymbol{X}}_{q'}))\rangle_{q\in Q_1} = & \quad [\text{definition of } \alpha] \\
\langle\lfloor\{\mathrm{pre}^{\mathcal{A}_2}_u(F_2) \mid u \in \textstyle\bigcup_{a\in\Sigma, q\xrightarrow{a}_{\mathcal{A}_1} q'} a\gamma(\overrightarrow{\boldsymbol{X}}_{q'})\rfloor\rangle_{q\in Q_1} = & \quad [\mathrm{pre}^{\mathcal{A}_2}_{av} = \mathrm{pre}^{\mathcal{A}_2}_a \circ \mathrm{pre}^{\mathcal{A}_2}_v] \\
\langle\lfloor\{\mathrm{pre}^{\mathcal{A}_2}_a(\{\mathrm{pre}^{\mathcal{A}_2}_u(F_2) \mid u \in \textstyle\bigcup_{q\xrightarrow{a}_{\mathcal{A}_1} q'} \gamma(\overrightarrow{\boldsymbol{X}}_{q'})\}) \mid a \in \Sigma\rfloor\rangle_{q\in Q_1} = & \\
\langle\lfloor\{\mathrm{pre}^{\mathcal{A}_2}_a(s) \mid a \in \Sigma, q \xrightarrow{a}_{\mathcal{A}_1} q', s \in \{\mathrm{pre}^{\mathcal{A}_2}_u(F_2) \mid u \in \gamma(\overrightarrow{\boldsymbol{X}}_{q'})\}\}\rfloor\rangle_{q\in Q_1} = & \quad [\lfloor\mathrm{pre}^{\mathcal{A}_2}_a(X)\rfloor = \lfloor\mathrm{pre}^{\mathcal{A}_2}_a(\lfloor X\rfloor)\rfloor] \\
\langle\lfloor\{\mathrm{pre}^{\mathcal{A}_2}_a(s) \mid a \in \Sigma, q \xrightarrow{a}_{\mathcal{A}_1} q', s \in \lfloor\{\mathrm{pre}^{\mathcal{A}_2}_u(F_2) \mid u \in \gamma(\overrightarrow{\boldsymbol{X}}_{q'})\}\rfloor\}\rfloor\rangle_{q\in Q_1} = & \quad [\text{definition of } \alpha] \\
\langle\lfloor\{\mathrm{pre}^{\mathcal{A}_2}_a(s) \mid a \in \Sigma, q \xrightarrow{a}_{\mathcal{A}_1} q', s \in \alpha(\gamma(\overrightarrow{\boldsymbol{X}}_{q'}))\rfloor\rangle_{q\in Q_1} = & \quad [\text{since } \overrightarrow{\boldsymbol{X}} \in \alpha,\ \alpha(\gamma(\overrightarrow{\boldsymbol{X}}_{q'})) = \overrightarrow{\boldsymbol{X}}_{q'}] \\
\langle\lfloor\{\mathrm{pre}^{\mathcal{A}_2}_a(s) \mid s \in \overrightarrow{\boldsymbol{X}}_{q'}, a \in \Sigma, q \xrightarrow{a}_{\mathcal{A}_1} q'\}\rfloor\rangle_{q\in Q_1} = & \\
\mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\overrightarrow{\boldsymbol{X}})\ . &
\end{aligned}$$

□

**Theorem 6.3.** *Let $\mathcal{A}_1, \mathcal{A}_2$ be two FAs. The algorithm* `FAIncS` *decides $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$.*

For clarity, we first recall some of the notation used in this paper:

$$\begin{aligned}
\alpha(X) &\triangleq \lfloor\{\mathrm{pre}^{\mathcal{A}_2}_x(F) \mid x \in X\}\rfloor \\
\gamma(Y) &\triangleq \{u \in \Sigma^* \mid \exists y \in Y, y \subseteq \mathrm{pre}^{\mathcal{A}_2}_u(F)\} \\
\mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\langle X_q\rangle_{q\in Q_1}) &\triangleq \langle\lfloor\{\,\mathrm{pre}^{\mathcal{A}_2}_a(s) \mid s \in X_{q'}, \exists a \in \Sigma,\ q \xrightarrow{a}_{\mathcal{A}_1} q'\}\rfloor\rangle_{q\in Q_1}
\end{aligned}$$

*Proof:* We show that the all the premises of Theorem 4.5 are satisfied for $\langle D, \sqsubseteq\rangle = \langle \mathrm{AC}_{\langle\wp(Q_2),\subseteq\rangle}, \sqsubseteq\rangle$ and the maps $\alpha$ and $\gamma$ defined before.

(*i*) Since $\rho_{\leq^l_{\mathcal{A}_2}}(X) = \gamma(\alpha(X))$, it follows from Lemmas 5.6 and 5.2 that $\gamma(\alpha(L_2)) = L_2$. Furthermore, for all $a \in \Sigma$, $X \in \wp(\Sigma^*)$ we next show that $\alpha(aX = \alpha(a\gamma\alpha(X))$

$$\begin{aligned}
\alpha(aX) = & \quad [\text{Galois Connection}] \\
\alpha(\gamma(\alpha(aX))) = & \quad [\text{Lemma 6.2 (d)}] \\
\alpha(\rho_{\leq^l_{\mathcal{A}_2}}(aX)) = & \quad [\text{Lemma 5.2 (b)}] \\
\alpha(a\rho_{\leq^l_{\mathcal{A}_2}}(X)) = & \quad [\text{Lemma 6.2 (d)}] \\
\alpha(a\gamma\alpha(X))\ . &
\end{aligned}$$

(*ii*) $(\mathrm{AC}_{\langle\wp(Q_2),\subseteq\rangle}, \sqsubseteq)$ is effective.

(*iii*) $\alpha(\{\epsilon\}) = \{F_2\}$ and $\alpha(\varnothing) = \varnothing$, hence $\lfloor \alpha(\overrightarrow{\boldsymbol{\epsilon}}^{F_2}) \rfloor$ is trivial to compute.

(*iv*) By Lemma 6.2 (e) we have $\alpha(\mathrm{Pre}_{\mathcal{A}_1}(\gamma(\overrightarrow{\boldsymbol{X}}))) = \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\overrightarrow{\boldsymbol{X}})$ for all $\overrightarrow{\boldsymbol{X}}$ in the image of $\alpha$.

(*v*) Since $\alpha(\overrightarrow{\boldsymbol{L_2}}^{I_1}) = \langle \alpha(\delta^{L_2}_{\Sigma^*}(q \in I_1)) \rangle_{q \in Q_1}$, the relation $\overrightarrow{\boldsymbol{Y}} \sqsubseteq \alpha(\overrightarrow{\boldsymbol{L_2}}^{I_1}))$ trivially holds for all components $q \notin I_1$. For the remaining components, it suffices to show that for all $Y$ in the image of $\alpha$ we have $Y \sqsubseteq \alpha(L_2) \Leftrightarrow \forall y \in Y,\ I_2 \cap y \neq \varnothing$, which coincides with the check performed by lines 2-5 of algorithm `FAIncS`.

$$
\begin{aligned}
Y \sqsubseteq \alpha(L_2) \Leftrightarrow &\quad [Y = \alpha(U) \text{ for some } U \in \wp(\Sigma^*)] \\
\alpha(U) \sqsubseteq \alpha(L_2) \Leftrightarrow &\quad [\text{Galois Connection}] \\
U \subseteq \gamma(\alpha(L_2)) \Leftrightarrow &\quad [\text{Lemmas 5.2, 5.6 and 6.2}] \\
U \subseteq L_2 \Leftrightarrow &\quad [\text{definition of } \mathrm{pre}^{\mathcal{A}_2}_u] \\
\forall u \in U, \mathrm{pre}_u(F_2) \cap I_2 \neq \varnothing \Leftrightarrow &\quad [\text{definition of } \alpha] \\
\forall y \in Y, y \cap I_2 \neq \varnothing \ . &
\end{aligned}
$$

It follows from these properties and Theorem 4.5 that algorithm `FAIncS` solves the inclusion problem $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$. □

**Theorem 6.4 ([13, Theorem 6]).** *Let*

$$\overrightarrow{\boldsymbol{\mathcal{FP}}} \triangleq \widehat{\bigsqcap}(\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(S) \ \widehat{\sqcup}\ \langle \delta^{\{F_2^c\}}_{\varnothing}(q \in F_1) \rangle_{q \in Q_1})$$

*where* $\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1} : \wp(Q_2)^{|Q_1|} \to \wp(Q_2)^{|Q_1|}$ *is defined by:*

$$
\begin{aligned}
\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\langle X_q \rangle_{q \in Q_1}) \triangleq \langle \lceil \{ Y \mid \exists a \in \Sigma, q' \in Q_1, X \subseteq Q_2, \\
q \xrightarrow{a}_{\mathcal{A}_1} q' \wedge X \in X_{q'} \wedge \mathrm{post}^{\mathcal{A}_2}_a(Y) \subseteq X \} \rceil \rangle_{q \in Q_1}
\end{aligned}
$$

*Then,* $\mathcal{L}(\mathcal{A}_1) \nsubseteq \mathcal{L}(\mathcal{A}_2)$ *iff* $\exists q \in I_2, \{I_2\} \widehat{\sqsubseteq} \overrightarrow{\boldsymbol{\mathcal{FP}}}_q$.

*Proof:* Let us introduce some notation necessary to understand the antichain algorithm of Wulf et. al [13] for deciding $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$. Let $(\mathrm{AC}_{\langle \wp(Q_1 \times \wp(Q_2)), \subseteq_\times \rangle}, \widehat{\sqsubseteq}_\times)$ be the complete lattice of antichains over $\wp(Q_1 \times \wp(Q_2))$ with

$$
\begin{aligned}
&X \widehat{\sqcup}_\times Y \triangleq \lceil \{(q,z) \mid (q,z) \in X \cup Y\} \rceil && X \widehat{\sqcap}_\times Y \triangleq \lceil \{(q, x \cap y) \mid (q,x) \in X \wedge (q,y) \in Y\} \\
&\lfloor X \rfloor_\times \triangleq \{(q,x) \in X \mid \forall (q',x') \in X, q = q' \Rightarrow x' \nsubseteq x\} && \lceil X \rceil_\times \triangleq \{(q,x) \in X \mid \forall (q',x') \in X, q = q' \Rightarrow x \nsubseteq x'\} \\
&X \widehat{\sqsubseteq}_\times Y \Leftrightarrow \forall (q,S) \in X, \exists (q,S') \in Y, S \subseteq S'
\end{aligned}
$$

and let $\mathrm{CPre}_q(S) \triangleq \lceil \{(q,X) \mid \exists a \in \Sigma, \exists (q',X') \in S, q \xrightarrow{a}_{\mathcal{A}_1} q' \wedge \mathrm{post}^{\mathcal{A}_2}_a(X) \subseteq X'\} \rceil$.

The result of Wulf et al. [13, Theorem 6] states that $\mathcal{L}(\mathcal{A}_1) \nsubseteq \mathcal{L}(\mathcal{A}_2)$ iff there is $q \in I_1$ such that $\{(q, I_2)\} \widehat{\sqsubseteq} \mathcal{FP}^q$ where

$$\mathcal{FP}^q = \widehat{\bigsqcap}_\times \{s \mid s \in \mathrm{CPre}_q(s) \ \widehat{\sqcup}_\times \ (F_1 \times \{F_2^c\})\}$$

It is easy to observe that the notation $(q, X) \in (Q_1, \wp(Q_2))$ used by Wulf to denote elements in $\mathrm{AC}_{\langle \wp(Q_1 \times \wp(Q_2)), \subseteq_\times \rangle}$ simply associates states of $\mathcal{A}_1$ with sets of states of $\mathcal{A}_2$. Therefore, we can modify the notation to work with vectors $\overrightarrow{\boldsymbol{X}} = \langle \{X \mid (q,X) \in S\} \rangle_{q \in Q_1}$ of $|Q_1|$ components in $(\mathrm{AC}_{\langle \wp(Q_2), \subseteq \rangle}, \widehat{\sqsubseteq})$, where

$$X \widehat{\sqsubseteq} Y \Leftrightarrow \forall x \in X, \exists y \in Y, x \subseteq y \qquad X \widehat{\sqcap} Y \triangleq \lceil \{x \cap y \mid x \in X, y \in Y\} \rceil \qquad X \widehat{\sqcup} Y \triangleq \lceil \{z \mid z \in X \cup Y\} \rceil$$

Then, we can replace $\mathrm{CPre}_q(S)$ by its vector-equivalent $\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\overrightarrow{\boldsymbol{X}})$:

$$\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\overrightarrow{\boldsymbol{X}}) \triangleq \langle \lceil \{Y \mid \exists a \in \Sigma, X \in \overrightarrow{\boldsymbol{X}}'_q, q \xrightarrow{a}_{\mathcal{A}_1} q' \wedge \mathrm{post}^{\mathcal{A}_2}_a(Y) \subseteq X\} \rceil \rangle_{q \in Q_1} \ .$$

and replace the list of $\mathcal{FP}^q$ by vector $\overrightarrow{\boldsymbol{\mathcal{FP}}}$ where

$$\overrightarrow{\boldsymbol{\mathcal{FP}}} \triangleq \widehat{\bigsqcap}(\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(S) \ \widehat{\sqcup}\ \langle \delta^{\{F_2^c\}}_{\varnothing}(q \in F_1) \rangle_{q \in Q_1})$$

Finally, the condition $\exists q \in I_1, \{(q, I_2)\} \widehat{\sqsubseteq}_\times \mathcal{FP}_q$ translates into $\exists q \in I_2, \{I_2\} \widehat{\sqsubseteq} \overrightarrow{\boldsymbol{\mathcal{FP}}}_q$. □

**Theorem 6.5.** *At each step of the least fixpoint computations for* $\overrightarrow{\boldsymbol{\mathcal{FP}}}$ *of Theorem 6.4 and* $\overrightarrow{\boldsymbol{Y}}$ *of* `FAIncS`*, the following invariant holds:*

$$\forall S \subseteq Q_2, q \in Q_1,\ \{S\} \widehat{\sqsubseteq} \overrightarrow{\boldsymbol{\mathcal{FP}}}_q \Leftrightarrow \{S^c\} \sqsubseteq \overrightarrow{\boldsymbol{Y}}_q$$

Let us recall some definitions

$$\begin{aligned}
\mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\langle X_q\rangle_{q\in Q_1}) &\triangleq \langle\lfloor\{\,\mathrm{pre}^{\mathcal{A}_2}_a(s) \mid s\in X_q, q\xrightarrow{a}_{\mathcal{A}_1} q'\}\rfloor\rangle_{q\in Q_1}\\
\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\langle X_q\rangle_{q\in Q_1}) &\triangleq \langle\lceil\{Y \mid \exists a\in\Sigma, X\in X_{q'}, q\xrightarrow{a}_{\mathcal{A}_1} q' \wedge \mathrm{post}^{\mathcal{A}_2}_a(Y)\subseteq X\}\rceil\rangle_{q\in Q_1}\\
\mathrm{post}^{\mathcal{A}_2}_u(X) &\triangleq \{q\in Q_2 \mid u\in W^{\mathcal{A}_2}_{X,q}\} \qquad \mathrm{pre}^{\mathcal{A}}_u(X) \triangleq \{q\in Q \mid u\in W^{\mathcal{A}}_{q,X}\}\\
X\sqsubseteq Y &\Leftrightarrow \forall x\in X, \exists y\in Y, y\subseteq x \qquad X\mathrel{\widehat{\sqsubseteq}} Y \Leftrightarrow \forall x\in X, \exists y\in Y, x\subseteq y
\end{aligned}$$

*Proof:* Define the complement of a set $R\subseteq\wp(Q_2)$ as $\complement(R)=\{S^c \mid S\in R\}$. As with other operators, we use the same symbol to denote the componentwise extension of the complement to vectors. Next we prove that

$$\forall S\in\wp(Q_2), \forall\vec{\boldsymbol{X}}\in\wp(Q_2)^{|Q_1|}, \forall q\in Q_1, \{S\}\mathrel{\widehat{\sqsubseteq}}\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\vec{\boldsymbol{X}})_q \Leftrightarrow \{S^c\}\sqsubseteq \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\complement(\vec{\boldsymbol{X}}))_q$$

We proceed by proving the two sides of the implication

$$\begin{aligned}
\{S\}\mathrel{\widehat{\sqsubseteq}}\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\vec{\boldsymbol{X}})_q \Leftrightarrow &\quad [\text{definition of } \widehat{\sqsubseteq}]\\
\exists Y\in\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\vec{\boldsymbol{X}})_q,\ S\subseteq Y \Leftrightarrow &\quad [\text{definition of } \mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}]\\
\exists Y\in\wp(Q), a\in\Sigma, X\in\vec{\boldsymbol{X}}_{q'}, q\xrightarrow{a} q',\ \mathrm{post}^{\mathcal{A}_2}_a(Y)\subseteq X\wedge S\subseteq Y \Rightarrow &\quad [S\subseteq Y\Rightarrow \mathrm{post}^{\mathcal{A}_2}_a(S)\subseteq \mathrm{post}^{\mathcal{A}_2}_a(Y)]\\
\exists a\in\Sigma, X\in\vec{\boldsymbol{X}}_{q'}, q\xrightarrow{a} q',\ \mathrm{post}^{\mathcal{A}_2}_a(S)\subseteq X \Leftrightarrow &\quad [\text{P. Cousot [8]: } \mathrm{post}^{\mathcal{A}_2}_a(S)\subseteq X\Leftrightarrow S\subseteq(\mathrm{pre}^{\mathcal{A}_2}_a(X^c))^c]\\
\exists a\in\Sigma, X\in\vec{\boldsymbol{X}}_{q'}, q\xrightarrow{a} q',\ S\subseteq(\mathrm{pre}^{\mathcal{A}_2}_a(X^c))^c \Leftrightarrow &\quad [A\subseteq B\Leftrightarrow B^c\subseteq A^c]\\
\exists a\in\Sigma, X\in\vec{\boldsymbol{X}}_{q'}, q\xrightarrow{a} q',\ \mathrm{pre}^{\mathcal{A}_2}_a(X^c)\subseteq S^c \Leftrightarrow &\quad [\text{definition of } \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}]\\
\exists Y\in\mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\complement(\vec{\boldsymbol{X}}))_q, Y\subseteq S^c \Leftrightarrow &\quad [\text{definition of } \sqsubseteq]\\
\{S^c\}\sqsubseteq \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\complement(\vec{\boldsymbol{X}}))_q\ . &
\end{aligned}\tag{17}$$

On the other hand

$$\begin{aligned}
\{S^c\}\sqsubseteq \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\complement(\vec{\boldsymbol{X}}))_q \Leftrightarrow &\quad [\text{definition of } \sqsubseteq]\\
\exists Y\in\mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\complement(\vec{\boldsymbol{X}}))_q,\ Y\subseteq S^c \Leftrightarrow &\quad [\text{definition of } \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}]\\
\exists Y\in\wp(Q), a\in\Sigma, X\in\complement(\vec{\boldsymbol{X}})_{q'}, q\xrightarrow{a} q',\ \mathrm{pre}^{\mathcal{A}_2}_a(X)\subseteq Y\wedge Y\subseteq S^c \Rightarrow &\quad [\text{transitivity of } \subseteq]\\
\exists a\in\Sigma, X\in\complement(\vec{\boldsymbol{X}})_{q'}, q\xrightarrow{a} q',\ \mathrm{pre}^{\mathcal{A}_2}_a(X)\subseteq S^c \Leftrightarrow &\quad [A\subseteq B\Leftrightarrow B^c\subseteq A^c]\\
\exists a\in\Sigma, X\in\complement(\vec{\boldsymbol{X}})_{q'}, q\xrightarrow{a} q',\ S\subseteq(\mathrm{pre}^{\mathcal{A}_2}_a(X))^c \Leftrightarrow &\quad [\text{P. Cousot [8]: } \mathrm{post}^{\mathcal{A}_2}_a(S)\subseteq X\Leftrightarrow S\subseteq(\mathrm{pre}^{\mathcal{A}_2}_a(X^c))^c]\\
\exists a\in\Sigma, X\in\complement(\vec{\boldsymbol{X}})_{q'}, q\xrightarrow{a} q',\ \mathrm{post}^{\mathcal{A}_2}_a(S)\subseteq X^c \Leftrightarrow &\quad [\text{definition of } \mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}]\\
\exists Y\in\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(X), S\subseteq Y \Leftrightarrow &\quad [\text{definition of } \widehat{\sqsubseteq}]\\
\{S\}\mathrel{\widehat{\sqsubseteq}}\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(X)\ . &
\end{aligned}\tag{18}$$

It follows from (17) and (18) that:

$$\{S\}\mathrel{\widehat{\sqsubseteq}}\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\vec{\boldsymbol{X}})_q \Leftrightarrow \{S^c\}\sqsubseteq \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\complement(\vec{\boldsymbol{X}}))_q\ . \tag{19}$$

On the other hand, observe that

$$\begin{aligned}
\complement(\lceil X\rceil) = \{s^c \mid s\in\lceil X\rceil\} = \{s^c \mid s\in\lceil X\rceil\wedge\forall x\in\lceil X\rceil, s\not\subseteq x\} = \{s \mid s^c\in\lceil X\rceil\wedge\forall x\in\lceil X\rceil, s^c\not\subseteq x\} =\\
\{s \mid s\in\complement(\lceil X\rceil)\wedge\forall x\in\lceil X\rceil, x^c\not\subseteq s\} = \{s \mid s\in\complement(\lceil X\rceil)\wedge\forall x\in\complement(\lceil X\rceil), x\not\subseteq s\} = \lfloor\complement(\lceil X\rceil)\rfloor\ .
\end{aligned}\tag{20}$$

Therefore $\forall q\in Q_1, \forall S\in\wp(Q_2), \forall\vec{\boldsymbol{X}}\in\wp(Q_2)^{|Q_1|}$ we have

$$\begin{aligned}
\{S\}\mathrel{\widehat{\sqsubseteq}}\mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\lceil\vec{\boldsymbol{X}}\rceil)_q \Leftrightarrow &\quad [\text{Equation 19}]\\
\{S^c\}\sqsubseteq \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\complement(\lceil\vec{\boldsymbol{X}}\rceil))_q \Leftrightarrow &\quad [\text{Equation 20}]\\
\{S^c\}\sqsubseteq \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\lfloor\complement(\lceil\vec{\boldsymbol{X}}\rceil)\rfloor)_q\ . &
\end{aligned}\tag{21}$$

Now, we show by induction in the steps of the fixpoint computations that

$$\forall S\in\wp(Q_2), \forall q\in Q_1,\ \{S\}\mathrel{\widehat{\sqsubseteq}}\overrightarrow{\boldsymbol{\mathcal{FP}}}_q \Leftrightarrow \{S^c\}\sqsubseteq\vec{\boldsymbol{Y}}_q\ . \tag{22}$$

where $\vec{\boldsymbol{Y}} = \mathtt{Kleene}(\lambda\vec{\boldsymbol{X}}.\,\lfloor\alpha(\vec{\boldsymbol{\epsilon}}^F)\rfloor\sqcup\lfloor\mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\vec{\boldsymbol{X}})\rfloor, \vec{\boldsymbol{\varnothing}})$.

- *Base case.* The vectors $\overrightarrow{\mathcal{FP}}$ and $\vec{\boldsymbol{Y}}$ are initialized as: $\overrightarrow{\mathcal{FP}}^0 = \langle \delta_{\varnothing}^{\{F_2^c\}}(q \in F_2) \rangle_{q \in Q_1}$ and $\vec{\boldsymbol{Y}}^0 = \langle \delta_{\varnothing}^{\{F_2\}}(q \in F_2) \rangle_{q \in Q_1}$. Clearly, $\forall S \in \wp(Q_2), \forall q \in Q_1$ the relation $\{S\} \mathrel{\widehat{\sqsubseteq}} \overrightarrow{\mathcal{FP}}_q \Leftrightarrow \{S^c\} \sqsubseteq \vec{\boldsymbol{Y}}_q$ holds since $S \subseteq F_2^c \Leftrightarrow F_2 \subseteq S^c$.
- *Inductive step.* Assume that (22) holds up to the $n$-th step of the fixpoint computation, i.e. $\forall S \in \wp(Q_2), \forall q \in Q_1$ we have

$$\{S\} \mathrel{\widehat{\sqsubseteq}} \overrightarrow{\mathcal{FP}}^n_q \Leftrightarrow \{S^c\} \sqsubseteq \vec{\boldsymbol{Y}}^n_q \tag{23}$$

Then $\forall S \in \wp(Q_2), \forall q \in Q_1$ we prove that $\{S\} \mathrel{\widehat{\sqsubseteq}} \overrightarrow{\mathcal{FP}}^n_q \Leftrightarrow \{S^c\} \sqsubseteq \complement(\overrightarrow{\mathcal{FP}}^n_q)$

$$\begin{aligned}
\{S\} \mathrel{\widehat{\sqsubseteq}} \overrightarrow{\mathcal{FP}}^n_q \Leftrightarrow & \quad \text{[definition of } \widehat{\sqsubseteq}] \\
\exists Y \in \overrightarrow{\mathcal{FP}}^n_q,\ S \subseteq Y \Leftrightarrow & \quad [A \subseteq B \Leftrightarrow B^c \subseteq A^c] \\
\exists Y \in \overrightarrow{\mathcal{FP}}^n_q,\ Y^c \subseteq S^c \Leftrightarrow & \quad \text{[definition of } \complement] \\
\exists Y' \in \complement(\overrightarrow{\mathcal{FP}}^n_q),\ Y' \subseteq S^c \Leftrightarrow & \quad \text{[definition of } \sqsubseteq] \\
\{S^c\} \sqsubseteq \complement(\overrightarrow{\mathcal{FP}}^n_q)\ . &
\end{aligned} \tag{24}$$

It follows from (23) and (24) that $\forall S \in \wp(Q_2), \forall q \in Q_1, \{S\} \sqsubseteq \vec{\boldsymbol{Y}}^n_q \Leftrightarrow \{S\} \sqsubseteq \complement(\overrightarrow{\mathcal{FP}}^n_q)$ hence (note quantifier $\forall S$)

$$\lfloor \vec{\boldsymbol{Y}}^n_q \rfloor = \lfloor \complement(\overrightarrow{\mathcal{FP}}^n_q) \rfloor\ . \tag{25}$$

On the other hand

$$\overrightarrow{\mathcal{FP}}^{n+1} = \langle \delta_{\varnothing}^{\{F_2^c\}}(q \in F_2) \rangle_{q \in Q_1} \mathbin{\widehat{\sqcup}} \mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\overrightarrow{\mathcal{FP}}^n) \quad \text{and} \quad \vec{\boldsymbol{Y}}^{n+1} = \langle \delta_{\varnothing}^{\{F_2\}}(q \in F_2) \rangle_{q \in Q_1} \sqcup \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\vec{\boldsymbol{Y}}^n)\ .$$

Therefore, $\forall S \in \wp(Q), q \in Q_1$ whenever $S \subseteq F_2^c$, we know (*base case*) that (22) holds. When $S \nsubseteq F_2^c$ then

$$\begin{aligned}
\{S\} \mathrel{\widehat{\sqsubseteq}} \overrightarrow{\mathcal{FP}}^{n+1}_q \Leftrightarrow & \quad \text{[definition of } \overrightarrow{\mathcal{FP}}^{n+1},\ S \nsubseteq F_2^c] \\
\{S\} \mathrel{\widehat{\sqsubseteq}} \mathrm{CPre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\overrightarrow{\mathcal{FP}}^n)_q \Leftrightarrow & \quad \text{[Equation (21)]} \\
\{S^c\} \sqsubseteq \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\lfloor \complement(\overrightarrow{\mathcal{FP}}^n) \rfloor)_q \Leftrightarrow & \quad \text{[Equation (25)]} \\
\{S^c\} \sqsubseteq \mathrm{Pre}^{\mathcal{A}_2}_{\mathcal{A}_1}(\lfloor \vec{\boldsymbol{Y}}^n \rfloor)_q \Leftrightarrow & \quad \text{[definition of } \vec{\boldsymbol{Y}}^{n+1}_q,\ S \nsubseteq F_2^c] \\
\{S^c\} \sqsubseteq \vec{\boldsymbol{Y}}^{n+1}_q\ . &
\end{aligned}$$

which concludes the inductive step. □

**Theorem 7.2.** *Let $\mathcal{G} = (\mathcal{V}, \Sigma, P)$ be a CFG in CNF. If $\rho \in \mathrm{uco}(\wp(\Sigma^*))$ is backward complete for both $\lambda X.Xa$ and $\lambda X.aX$, for all $a \in \Sigma$, then $\rho$ is backward complete for $\mathrm{Fn}_{\mathcal{G}}$ and $\lambda \vec{\boldsymbol{X}}.\ \vec{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\vec{\boldsymbol{X}})$.*

*Proof:* First we show that backward completeness for left and right concatenation can be extended from letter to words. We give the proof for the concatenation to the left, the case of the concatenation to the right has a similar proof. The formal statement to prove is $\rho(wX) = \rho(w\rho(X))$ for every $w \in \Sigma^*$. We proceed by induction on $|w|$. The base case is trivial to prove using the fact that $\rho$ is idempotent. For the inductive case ($|w| > 0$) let $u \in \Sigma^*$ and $a \in \Sigma$ be such that $w = au$.

$$\begin{aligned}
\rho(auX) = & \quad \text{[backward completeness for } \lambda X.\, aX] \\
\rho(a\rho(uX)) = & \quad \text{[Induction hypothesis]} \\
\rho(a\rho(u\rho(X))) = & \quad \text{[backward completeness for } \lambda X.\, aX] \\
\rho(au\rho(X))\ . &
\end{aligned}$$

Next we turn to the binary concatenation case, where the formal statement to prove is $\rho(YZ) = \rho(\rho(Y)\rho(Z))$ for all $Y, Z \in \wp(\Sigma^*)$

$$\begin{aligned}
\rho(\rho(Y)\rho(Z)) = &\quad \text{[definition of concatenation]}\\
\rho(\textstyle\bigcup_{u\in\rho(Y)} u\rho(Z)) = &\quad [\rho(\cup Z) = \rho(\cup\rho(Z))]\\
\rho(\textstyle\bigcup_{u\in\rho(Y)} \rho(u\rho(Z))) = &\quad \text{[backward completeness of } \lambda X.\, wX]\\
\rho(\textstyle\bigcup_{u\in\rho(Y)} \rho(uZ)) = &\quad [\rho(\cup Z) = \rho(\cup\rho(Z))]\\
\rho(\textstyle\bigcup_{u\in\rho(Y)} uZ) = &\quad \text{[definition of concatenation]}\\
\rho(\rho(Y)Z) = &\quad \text{[definition of concatenation]}\\
\rho(\textstyle\bigcup_{v\in Z} \rho(Y)v) = &\quad [\rho(\cup Z) = \rho(\cup\rho(Z))]\\
\rho(\textstyle\bigcup_{v\in Z} \rho(\rho(Y)v)) = &\quad \text{[backward completeness of } \lambda X.\, Xw]\\
\rho(\textstyle\bigcup_{v\in Z} \rho(Yv)) = &\quad [\rho(\cup Z) = \rho(\cup\rho(Z))]\\
\rho(\textstyle\bigcup_{v\in Z} Yv) = &\quad \text{[definition of concatenation]}\\
\rho(YZ)\ \ . &
\end{aligned}$$

Finally, the proof follows the same lines of the proof of Theorem 4.3. Indeed, it follows from the definition of $\mathrm{Fn}_{\mathcal{G}}(\langle X_i\rangle_{i\in[0,n]})$

$$\begin{aligned}
\rho(\textstyle\bigcup_{j=1}^{k_i} \beta_j^{(i)}) = &\quad \text{[definition of } \beta_j^{(i)}]\\
\rho(\textstyle\bigcup_{j=1}^{k_i} X_j^{(i)} Y_j^{(i)}) = &\quad [\rho(\cup Y) = \rho(\cup\rho(Y))]\\
\rho(\textstyle\bigcup_{j=1}^{k_i} \rho(X_j^{(i)} Y_j^{(i)})) = &\quad \text{[backward completeness of binary concatenation]}\\
\rho(\textstyle\bigcup_{j=1}^{k_i} \rho(\rho(X_j^{(i)})\rho(Y_j^{(i)}))) = &\quad [\rho(\cup Y) = \rho(\cup\rho(Y))]\\
\rho(\textstyle\bigcup_{j=1}^{k_i} \rho(X_j^{(i)})\rho(Y_j^{(i)}))\ \ . &
\end{aligned}$$

Hence, by a straightforward componentwise application on vectors in $\wp(\Sigma^*)^{|\mathcal{V}|}$, we obtain that $\rho$ is backward complete for $\mathrm{Fn}_{\mathcal{G}}$. In turn, $\rho$ is backward complete for $\lambda\vec{\boldsymbol{X}}.\,(\vec{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\vec{\boldsymbol{X}}))$, because:

$$\begin{aligned}
\rho(\vec{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\rho(\vec{\boldsymbol{X}}))) = &\quad [\rho(\cup Y) = \rho(\cup\rho(Y))]\\
\rho(\rho(\vec{\boldsymbol{b}}) \cup \rho(\mathrm{Fn}_{\mathcal{G}}(\rho(\vec{\boldsymbol{X}})))) = &\quad \text{[backward completeness of } \mathrm{Fn}_{\mathcal{G}}]\\
\rho(\rho(\vec{\boldsymbol{b}}) \cup \rho(\mathrm{Fn}_{\mathcal{G}}(\vec{\boldsymbol{X}}))) = &\quad [\rho(\cup Y) = \rho(\cup\rho(Y))]\\
\rho(\vec{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\vec{\boldsymbol{X}}))\ \ . &
\end{aligned}$$

□

**Theorem 7.3.** *Let $\mathcal{G} = \langle \mathcal{V}, \Sigma, P\rangle$ be a CFG in CNF and let $L_2$ be a language over $\Sigma$. Let $\langle \wp(\Sigma^*), \subseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle D, \sqsubseteq\rangle$ be a GC where $\langle D, \sqsubseteq\rangle$ is a poset. Assume that the following properties hold:*

*(i) $L_2 \in \gamma(D)$ and for every $a \in \Sigma$, $X \in \wp(\Sigma^*)$ we have $\alpha(aX) = \alpha(a\gamma\alpha(X))$ and $\alpha(Xa) = \alpha(\gamma\alpha(X)a)$.*

*(ii) $(D, \sqsubseteq, \sqcup)$ is an effective domain, meaning that: $(D, \sqsubseteq)$ is ACC, every element of $D$ has a finite representation, $\sqsubseteq$ is decidable and $\sqcup$ is a computable binary lub.*

*(iii) There is an algorithm, say $\mathrm{Fn}^\sharp(\vec{\boldsymbol{X}})$, computing $\alpha(\mathrm{Fn}_{\mathcal{G}}(\gamma(\vec{\boldsymbol{X}})))$, for all $\vec{\boldsymbol{X}} \in \wp(\Sigma^*)^{|\mathcal{V}|}$.*

*(iv) There is an algorithm, say $\mathrm{b}^\sharp$, computing $\alpha(\vec{\boldsymbol{b}})$.*

*(v) There is an algorithm, say $\mathrm{Incl}^\sharp(\vec{\boldsymbol{X}})$, deciding the abstract inclusion $\vec{\boldsymbol{X}} \sqsubseteq \alpha(\vec{\boldsymbol{L_2}}^{X_0})$, for every vector $\vec{\boldsymbol{X}} \in \alpha(\wp(\Sigma^*)^{|\mathcal{V}|})$.*

*Then, the following algorithm decides whether $\mathcal{L}(\mathcal{G}) \subseteq L_2$:*

$\langle Y_i\rangle_{i\in[0,n]} := \texttt{Kleene}(\lambda\vec{\boldsymbol{X}}.\,\mathrm{b}^\sharp \sqcup \mathrm{Fn}^\sharp(\vec{\boldsymbol{X}}), \vec{\boldsymbol{\varnothing}})$;
**return** $\mathrm{Incl}^\sharp(\langle Y_i\rangle_{i\in[0,n]})$;

*Proof:* Let $\rho = \gamma \circ \alpha \in \mathrm{uco}(\wp(\Sigma^*))$. Then, it follows from property (i) that $L_2 \in \rho$, $\rho(aX) = \rho(a\rho(X))$ and $\rho(Xa) = \rho(\rho(X)a)$. Therefore

$$\begin{aligned}
\mathcal{L}(\mathcal{A}) \subseteq L_2 \Leftrightarrow &\quad \text{[by (11)]}\\
\mathrm{lfp}(\lambda\vec{\boldsymbol{X}}.\,\rho(\vec{\boldsymbol{b}} \cup \mathrm{Fn}_{\mathcal{G}}(\vec{\boldsymbol{X}}))) \subseteq \vec{\boldsymbol{L_2}}^{X_0} \Leftrightarrow &\quad \text{[Lemma A.2]}\\
\gamma(\mathrm{lfp}(\lambda\vec{\boldsymbol{X}}.\,\alpha(\vec{\boldsymbol{b}}) \sqcup \alpha(\mathrm{Fn}_{\mathcal{G}}(\gamma(\vec{\boldsymbol{X}}))))) \subseteq \vec{\boldsymbol{L_2}}^{X_0} \Leftrightarrow &\quad \text{[Galois Connection]}\\
\mathrm{lfp}(\lambda\vec{\boldsymbol{X}}.\,\alpha(\vec{\boldsymbol{b}}) \sqcup \alpha(\mathrm{Fn}_{\mathcal{G}}(\gamma(\vec{\boldsymbol{X}})))) \sqsubseteq \alpha(\vec{\boldsymbol{L_2}}^{X_0})\ \ . &
\end{aligned}$$

Since $(D, \sqsubseteq)$ is ACC, `Kleene` is an algorithm computing the least fixpoint. Properties (ii), (iii) and (iv) ensure that the `Kleene` iterates of $\mathrm{lfp}(\lambda \vec{\boldsymbol{X}}.\, \alpha(\vec{\boldsymbol{b}}) \sqcup \alpha(\mathrm{Fn}_{\mathcal{G}}(\gamma(\vec{\boldsymbol{X}}))))$ are computable. Property (v) ensures decidability of the required $\sqsubseteq$-check since all `Kleene` iterates are in $\alpha(\wp(\Sigma^*))^{|\mathcal{V}|}$. □

**Theorem 7.5.** *Let $\mathcal{G}$ be a CFG and let $L_2$ be a language such that there exists a decidable $L_2$-consistent wqo on $\Sigma^*$. Then,* `CFGIncW` *decides the inclusion $\mathcal{L}(\mathcal{G}) \subseteq L_2$.*

Before proving Theorem 7.5 we show a result equivalent to Lemma 5.2 for left and right monotonic $L$-quasiorders.

**Lemma C.1.** *Let $L$ be a language over $\Sigma^*$ and let $\leqslant_L$ be a $L$-consistent quasiorder Then,*
*(a)* $\rho_{\leqslant_L}(L) = L$.
*(b)* $\rho_{\leqslant}$ *is backward complete for $\lambda X.\, aX$ and for $\lambda X.\, Xa$, with $a \in \Sigma$*

*Proof:* We consider the left case, the right case is symmetric.
(*a*) The inclusion $L \subseteq \rho_{\leqslant_L}(L)$ holds because $\rho_{\leqslant_L}$ is an upper closure. Property (a) of Definition 5.1 entails $\rho_{\leqslant_L}(L) \subseteq L$.
(*b*) We prove that $\rho_{\leqslant_L}(aX) = \rho_{\leqslant_L}(a\rho_{\leqslant_L}(X))$ for every $a \in \Sigma$. Monotonicity of concatenation together with monotonicity and extensivity of $\rho_{\leqslant_L}$ imply that $\rho_{\leqslant_L}(aX) \subseteq \rho_{\leqslant_L}(a\rho_{\leqslant_L}(X))$ holds. For the reverse inclusion

$$\begin{aligned}
\rho_{\leqslant_L}(a\rho_{\leqslant_L}(X)) = &\quad \text{[definition of } \rho_{\leqslant_L}\text{]}\\
\rho_{\leqslant_L}(\{ay \mid \exists x \in X, x \leqslant_L y\}) = &\quad \text{[definition of } \rho_{\leqslant_L}\text{]}\\
\{z \mid \exists y, ay \leqslant_L z \wedge \exists x \in X, x \leqslant_L y\} \subseteq &\quad \text{[left monotonicity of } \leqslant_L\text{]}\\
\{z \mid \exists y, ay \leqslant_L z \wedge \exists x \in X, ax \leqslant_L ay\} = &\quad \text{[transitivity of } \leqslant_L\text{]}\\
\{z \mid \exists x \in X, ax \leqslant_L z\} = &\quad \text{[definition of } \rho_{\leqslant_L}\text{]}\\
\rho_{\leqslant_L}(aX) \ . &
\end{aligned}$$

Backward completeness for right concatenation is proven similarly by relying on the right monotonicity of $\leqslant$.

□

*Proof of Theorem 7.5:* Let $\leqslant_{L_2}$ a decidable $L_2$-consistent wqo on $\Sigma^*$. Next we show that all the premises of Theorem 7.3 are satisfied for $\langle D, \sqsubseteq\rangle = \langle \mathrm{AC}_{\langle \Sigma^*, \leqslant_{L_2}\rangle}, \sqsubseteq\rangle$, $\alpha = \lfloor\cdot\rfloor$ and $\gamma = \rho_{\leqslant}$. Indeed, we apply Corollary 7.4 because $\langle \mathrm{AC}_{\langle \Sigma^*, \leqslant_{L_2}\rangle}, \sqsubseteq\rangle$ is a qoset so that we deal with a QGC rather than a GC.

(*i*) Since $L_2 = \rho_{\leqslant_{L_2}}(L_2) = \gamma(\alpha(L_2))$, it follows by by Lemma C.1 (a) that $L_2 \in \gamma(D)$. Moreover

$$\begin{aligned}
\alpha(aX) = &\quad \text{[Galois Connection]}\\
\alpha(\gamma(\alpha(aX))) = &\quad \text{[L. C.1 with } \rho_{\leqslant_{L_2}} = \gamma \circ \alpha\text{]}\\
\alpha(\gamma(\alpha(a\gamma(\alpha(X))))) = &\quad \text{[Galois Connection]}\\
\alpha(a\gamma\alpha(X)) \ . &
\end{aligned}$$

Similarly, $\alpha(Xa) = \alpha(\gamma\alpha(X)a)$.

(*ii*) It turns out that $\langle \mathrm{AC}_{\langle \Sigma^*, \leqslant_{L_2}\rangle}, \sqsubseteq\rangle$ is ACC because $\leqslant_{L_2}$ is a wqo. Moreover, the decidability of $\leqslant_{L_2}$ entails that $W_1 \sqcup W_2 \triangleq \lfloor W_1 \cup W_2 \rfloor$ is a computable binary lub in $\langle \mathrm{AC}_{\langle \Sigma^*, \leqslant_{L_2}\rangle}, \sqsubseteq\rangle$, therefore $\langle \mathrm{AC}_{\langle \Sigma^*, \leqslant_{L_2}\rangle}, \sqsubseteq, \sqcup\rangle$ is an effective domain.

(*iii*) We have $\alpha(\mathrm{Fn}_{\mathcal{G}}(\gamma(\vec{\boldsymbol{X}}))) = \lfloor \mathrm{Fn}_{\mathcal{G}}(\vec{\boldsymbol{X}})\rfloor$ since

$$\begin{aligned}
\alpha(\mathrm{Fn}_{\mathcal{G}}(\gamma(\vec{\boldsymbol{X}}))) = &\quad \text{[Galois Connection]}\\
\alpha\gamma\alpha(\mathrm{Fn}_{\mathcal{G}}(\gamma(\vec{\boldsymbol{X}}))) = &\quad \text{[definition of } \alpha \text{ and } \gamma\text{]}\\
\lfloor \rho_{\leqslant_{L_2}} \mathrm{Fn}_{\mathcal{G}}(\rho_{\leqslant_{L_2}}(\vec{\boldsymbol{X}}))\rfloor = &\quad \text{[Lemma C.1 (b)]}\\
\lfloor \rho_{\leqslant_{L_2}}(\mathrm{Fn}_{\mathcal{G}}(\vec{\boldsymbol{X}}))\rfloor = &\quad \text{[Galois Connection]}\\
\lfloor \mathrm{Fn}_{\mathcal{G}}(\vec{\boldsymbol{X}})\rfloor \ . &
\end{aligned}$$

(*iv*) $\alpha(\vec{\boldsymbol{b}})$ is computable since $\vec{\boldsymbol{b}}_i$ is finite and $\leqslant_{L_2}$ is decidable.

(*v*) Since $\alpha(\overrightarrow{\boldsymbol{L_2}}^{X_0}) = \alpha(\langle \delta^{L_2}_{\Sigma^*}(i = 0)\rangle_{i=[0,n]}$, the relation $\vec{\boldsymbol{Y}} \sqsubseteq \alpha(\overrightarrow{\boldsymbol{L_2}}^{X_0})$ trivially holds for all components $i > 0$. Therefore it suffices to check that $Y_0 \sqsubseteq \alpha(L_2)$

$$\begin{aligned}
Y_0 \sqsubseteq \alpha(L_2) \Leftrightarrow &\quad \text{[defs. } \sqsubseteq, \alpha\text{]}\\
\forall y \in Y_0, \exists x \in \lfloor L_2 \rfloor, x \leqslant_{L_2} y \Leftrightarrow &\quad [\lfloor X \rfloor \subseteq X]\\
\forall y \in Y_0, \exists x \in L_2, x \leqslant_{L_2} y \Leftrightarrow &\quad \text{[def. } \rho_{\leqslant_{L_2}}(L_2)\text{]}\\
\forall y \in Y_0, y \in \rho_{\leqslant_{L_2}}(L_2) \Leftrightarrow &\quad [\rho_{\leqslant_{L_2}}(L_2) = L_2]\\
\forall y \in Y_0, y \in L_2 \ . &
\end{aligned}$$

This latter condition coincides with the check performed by lines 2-5 of algorithm `CFGIncW` and is therefore decidable.

□

**Lemma 7.6.** *Let $L \subseteq \Sigma^*$ be a language.*
(*a*) *$\leqq_L$ is a (left and right) $L$-consistent quasiorder. Moreover, $\leqq_L$ is well-quasiorder iff $L$ is regular. Also, if $L$ is regular then $\leqq_L$ is decidable.*
(*b*) *Let $\leq$ be a quasiorder. If $\leq$ is (left and right) $L$-consistent then $\rho_{\leqq_L} \subseteq \rho_{\leq}$.*

*Proof:* We first show that $\leqq_L$ is left and right monotonic, i.e. $\forall a, b \in (\Sigma \cup \{\epsilon\}), u \leqq_L v \Rightarrow aub \leqq_L avb$

$$\begin{aligned}
u \leqq_L v \Leftrightarrow & \quad \text{[definition of } \leqq_L\text{]} \\
r^L(u) \subseteq r^L(v) \Leftrightarrow & \quad \text{[definitions of } r^L, \subseteq\text{]} \\
\forall xuy \in L,\ xvy \in L \Rightarrow & \quad [x = x'b,\ y = ay'] \\
\forall x'auby' \in L,\ x'avby' \in L \Leftrightarrow & \quad \text{[definitions of } r^L, \subseteq\text{]} \\
r^L(aub) \subseteq r^L(avb) \Leftrightarrow & \quad \text{[definition of } \leqq_L\text{]} \\
aub \leqq_L avb\ . &
\end{aligned}$$

Let $u \in L$ and $v \notin L$. By definition, $(\epsilon, \epsilon) \in r^L(u)$ but $(\epsilon, \epsilon) \notin r^L(v)$, hence $u \not\leqq_L v$. Therefore, $\leqq_L$ is a $L$-consistent quasiorder. It follows from De Luca and Varricchio [11, Theorem 2.1] that $\leqq_L$ is a wqo iff $L$ is regular.

It remains to show that $\leqq_L$ is decidable, which we do by using transducers. Define a *sequential transducer* as the 5-tuple $T = (S, \Sigma, \Delta, H, s_0)$ where $S$ is the finite set of states including the initial state $s_0$, $\Sigma$ is the input alphabet, $\Delta$ is the output alphabet and $H \subseteq S \times \Sigma \times \Delta \times S$ is the finite set of transitions.

For every $u \in \Sigma^*$, let $T_u = (\{q, q'\}, \Sigma, \Sigma \cup \{\sharp\}, H, q)$, with $H = \{q, u, \sharp, q'\} \cup \{(q, a, a, q), (q', a, a, q') \mid a \in \Sigma\}$. Observe that $T_u(L) = \{x\sharp y \mid (x, y) \in r^L(u)\}$ for every language $L$, hence $r^L(u) \subseteq r^L(v) \Leftrightarrow T_u(L) \subseteq T_v(L)$. When $L$ is regular, we know $T_u(L)$ is regular. It is straightforward to see that $T_u(L)$ is also computable, hence $\leqq_L$ is decidable.

Let us know show point (b). De Luca and Varricchio [11, Section 2, point 4] observe that $\leqq_L$ is maximum in the set of all $L$-consistent quasiorders, i.e. every $L$-consistent quasiorder $\leq$ on $\Sigma^*$ is such that $x \leq y \Rightarrow x \leqq_L y$. As a consequence, $\rho_{\leq}(U) \subseteq \rho_{\leqq_L}(U)$ holds for all $U \in \wp(\Sigma^*)$:

$$\begin{aligned}
\rho_{\leq}(U) = & \quad \text{[definition of } \rho_{\leq}\text{]} \\
\{x \in \Sigma^* \mid \exists u \in U, u \leq x\} \subseteq & \quad [x \leq y \Rightarrow x \leqq_L y] \\
\{x \in \Sigma^* \mid \exists u \in U, u \leqq_L x\} = & \quad \text{[definition of } \rho_{\leqq_L}\text{]} \\
\rho_{\leqq_L}(U)\ . &
\end{aligned}$$

In particular, $\rho_{\leqq_L}(\wp(\Sigma^*)) \subseteq \rho_{\leq}(\wp(\Sigma^*))$ holds. □

**Lemma 7.7.** *Let $\mathcal{A}$ be an FA. Then $\leq_{\mathcal{A}}$ is a decidable $L(\mathcal{A})$-consistent well-quasiorder.*

*Proof:* Let $u \in \mathcal{L}(\mathcal{A})$ and $v \notin \mathcal{L}(\mathcal{A})$. Then the set $\mathrm{ctx}^{\mathcal{A}}(u)$ must contain a pair in $I \times F$ while $\mathrm{ctx}^{\mathcal{A}}(v)$ does not, hence $u \not\leq_{\mathcal{A}} v$. Next we show $\leq_{\mathcal{A}}$ is left monotonic, i.e. $\forall a \in \Sigma, u \leq_{\mathcal{A}} v \Rightarrow au \leq_{\mathcal{A}} av$. Right monotonicity is proven similarly. Observe that for all $a \in \Sigma$:

$$au \in W_{q_1,q_2} \Leftrightarrow \exists q' \in Q,\ a \in W_{q_1,q'} \wedge u \in W_{q',q_2}\ . \tag{26}$$

Therefore

$$\begin{aligned}
u \leq_{\mathcal{A}} v \Leftrightarrow & \quad \text{[definitions of } \leq_{\mathcal{A}}, \subseteq\text{]} \\
\forall q_1, q_2,\ u \in W_{q_1,q_2} \Rightarrow v \in W_{q_1,q_2} \Rightarrow & \quad \text{[by (26)]} \\
\forall q', q''\ au \in W_{q',q''} \Rightarrow av \in W_{q',q''} \Leftrightarrow & \quad \text{[definition of } \leq_{\mathcal{A}}, \subseteq\text{]} \\
au \leq_{\mathcal{A}} av\ . &
\end{aligned}$$

Since $\wp(Q \times Q)$ is finite, it follows that $\leq$ is a wqo. Finally, decidability follows from the fact that $Q \times Q$ is finite and the sets $W_{q,q'}$ are regular and computable. □

**Lemma 7.8.** $\langle \mathrm{AC}_{\langle \Sigma^*, \leq_{\mathcal{A}} \rangle}, \sqsubseteq' \rangle \underset{\alpha_{\mathcal{A}}}{\overset{\gamma_{\mathcal{A}}}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \wp(Q \times Q), \subseteq \rangle}, \sqsubseteq \rangle$ *is a QGC.*

*Proof:*

$$\begin{aligned}
\alpha_{\mathcal{A}_2}(X) \sqsubseteq Y \Leftrightarrow & \quad \text{[definition of } \sqsubseteq\text{]} \\
\forall z \in \alpha_{\mathcal{A}}(X), \exists y \in Y,\ y \subseteq z \Leftrightarrow & \quad \text{[definition of } \alpha_{\mathcal{A}}\text{]} \\
\forall v \in X, \exists y \in Y,\ y \subseteq \mathrm{ctx}^{\mathcal{A}}(v) \Leftrightarrow & \quad \text{[definition of } \gamma_{\mathcal{A}}\text{]} \\
\forall v \in X, \exists u \in \gamma_{\mathcal{A}}(Y),\ \mathrm{ctx}^{\mathcal{A}}(u) \subseteq \mathrm{ctx}^{\mathcal{A}}(v) \Leftrightarrow & \quad \text{[definition of } \leq^l_{\mathcal{A}}\text{]} \\
\forall v \in X, \exists u \in \gamma_{\mathcal{A}}(Y),\ u \leq_{\mathcal{A}} v \Leftrightarrow & \quad \text{[definition of } \sqsubseteq'\text{]} \\
X \sqsubseteq' \gamma_{\mathcal{A}}(Y)\ . &
\end{aligned}$$

□

**Lemma 7.9.** *The following hold:*
(a) $\alpha = \alpha_{\mathcal{A}} \circ \alpha_{\leq_{\mathcal{A}}}$
(b) $\gamma = \gamma_{\leq_{\mathcal{A}}} \circ \gamma_{\mathcal{A}}$
(c) $\langle \wp(\Sigma^*), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{AC}_{\langle \wp(Q\times Q), \subseteq \rangle}, \sqsubseteq \rangle$ *is a GC.*
(d) $\gamma \circ \alpha = \rho_{\leq_{\mathcal{A}}}$
(e) $\mathrm{Fn}^{\mathcal{A}}_{\mathcal{A}_1}(\vec{\boldsymbol{X}}) = \alpha_{\mathcal{A}} \circ \alpha_{\leq_{\mathcal{A}}} \circ \mathrm{Fn}_{\mathcal{A}_1} \circ \gamma_{\leq_{\mathcal{A}}} \circ \gamma_{\mathcal{A}}(\vec{\boldsymbol{X}})$ *for all* $\vec{\boldsymbol{X}} \in \alpha(\wp(\Sigma^*)^{|\mathcal{V}|})$

Recall the following definitions

$$\begin{aligned} \alpha_{\leq_{\mathcal{A}}}(X) &\triangleq \lfloor X \rfloor & \gamma_{\leq_{\mathcal{A}}}(Y) &\triangleq \rho_{\leq_{\mathcal{A}}}(Y) \\ \alpha_{\mathcal{A}}(X) &\triangleq \{\mathrm{ctx}^{\mathcal{A}}(u) \mid u \in X\} & \gamma_{\mathcal{A}}(Y) &\triangleq \lfloor \{u \in \Sigma^* \mid \mathrm{ctx}^{\mathcal{A}}(u) \in Y\} \rfloor \\ \alpha(X) &\triangleq \lfloor \{\mathrm{ctx}^{\mathcal{A}}(u) \mid u \in X\} \rfloor & \gamma(Y) &\triangleq \{u \in \Sigma^* \mid \exists y \in Y, y \subseteq \mathrm{ctx}^{\mathcal{A}}(u)\} \end{aligned}$$

*Proof:*
(*a*)

$$\begin{aligned} \alpha_{\mathcal{A}}(\alpha_{\leq_{\mathcal{A}}}(X)) = & \quad [\text{definitions of } \alpha_{\leq_{\mathcal{A}}} \text{ and } \alpha_{\mathcal{A}}] \\ \{\mathrm{ctx}^{\mathcal{A}}(u) \mid u \in \lfloor X \rfloor\} = & \quad [\text{definition of } \lfloor\cdot\rfloor] \\ \{\mathrm{ctx}^{\mathcal{A}}(u) \mid u \in X \wedge \forall x \in X, x \not\leq_{\mathcal{A}} u\} = & \quad [\text{definition of } \leq_{\mathcal{A}}] \\ \{\mathrm{ctx}^{\mathcal{A}}(u) \mid u \in X \wedge \forall x \in X, \mathrm{ctx}^{\mathcal{A}}(x) \nsubseteq \mathrm{ctx}^{\mathcal{A}}(u)\} = & \quad [\text{definition of } \lfloor\cdot\rfloor] \\ \lfloor \{\mathrm{ctx}^{\mathcal{A}}(u) \mid u \in X\} \rfloor = & \quad [\text{definition of } \alpha] \\ \alpha(X) \ . & \end{aligned}$$

(*b*)

$$\begin{aligned} \gamma_{\leq_{\mathcal{A}}}(\gamma_{\mathcal{A}}(Y)) = & \quad [\text{definitions of } \gamma_{\mathcal{A}} \text{ and } \gamma_{\leq_{\mathcal{A}}}] \\ \rho_{\leq_{\mathcal{A}}}(\lfloor \{u \in \Sigma^* \mid \mathrm{ctx}^{\mathcal{A}}(u) \in Y\} \rfloor) = & \quad [\text{definition of } \rho_{\leq_{\mathcal{A}}}] \\ \{x \in \Sigma^* \mid \exists y \in \lfloor \{u \in \Sigma^* \mid \mathrm{ctx}^{\mathcal{A}}(u) \in Y\} \rfloor,\ y \leq_{\mathcal{A}} x\} = & \quad [\text{definition of } \lfloor\rfloor] \\ \{x \in \Sigma^* \mid \exists y \in \{u \in \Sigma^* \mid \mathrm{ctx}^{\mathcal{A}}(u) \in Y\},\ y \leq_{\mathcal{A}} x\} = & \quad [\text{definition of } \leq_{\mathcal{A}}] \\ \{x \in \Sigma^* \mid \exists y \in \{u \in \Sigma^* \mid \mathrm{ctx}^{\mathcal{A}}(u) \in Y\},\ \mathrm{ctx}^{\mathcal{A}}(y) \subseteq \mathrm{ctx}^{\mathcal{A}}(x)\} = & \\ \{x \in \Sigma^* \mid \exists y \in Y,\ y \subseteq \mathrm{ctx}^{\mathcal{A}}(x)\} = & \quad [\text{definition of } \gamma] \\ \gamma(Y) \ . & \end{aligned}$$

(*c*) This follows by composition of QGCs and by the fact that concrete and abstract domains are posets.
(*d*)

$$\begin{aligned} \gamma(\alpha(X)) = & \quad [\text{definition of } \gamma] \\ \{u \in \Sigma^* \mid \exists y \in \alpha(X),\ y \subseteq \mathrm{ctx}^{\mathcal{A}}(u)\} = & \quad [\text{definition of } \alpha] \\ \{u \in \Sigma^* \mid \exists x \in X,\ \mathrm{ctx}^{\mathcal{A}}(x) \subseteq \mathrm{ctx}^{\mathcal{A}}(u)\} = & \quad [\text{definition of } \rho_{\leq_{\mathcal{A}}}] \\ \rho_{\leq_{\mathcal{A}}}(X) \ . & \end{aligned}$$

(*e*) Due to properties a and b, it suffices to show that $\alpha(\mathrm{Fn}_{\mathcal{G}}(\gamma(\vec{\boldsymbol{X}}))) = \mathrm{Fn}^{\mathcal{A}}_{\mathcal{G}}(\vec{\boldsymbol{X}})$ for all $\vec{\boldsymbol{X}}$ in the image of $\alpha$. First, observe that

$$\begin{aligned} \mathrm{ctx}^{\mathcal{A}}(uv) = & \quad [\text{definitions of } \mathrm{ctx}^{\mathcal{A}} \text{ and } W_{q,q'}] \\ \{(q,q') \in Q^2 \mid q \overset{uv}{\rightsquigarrow} q'\} = & \quad [q \overset{uv}{\rightsquigarrow} q' \Leftrightarrow \exists q'' \in Q,\ q \overset{u}{\rightsquigarrow} q'' \wedge q'' \overset{v}{\rightsquigarrow} q'] \\ \{(q,q') \in Q^2 \mid \exists q'' \in Q, q \overset{u}{\rightsquigarrow} q'' \wedge q'' \overset{v}{\rightsquigarrow} q'\} = & \quad [\text{definition of } \circ \text{ for binary relations}] \\ \{(q,q'') \in Q^2 \mid q \overset{u}{\rightsquigarrow} q''\} \circ \{(q'',q') \in Q^2 \mid q'' \overset{v}{\rightsquigarrow} q'\} = & \quad [\text{definitions of } W_{q,q'} \text{ and } \mathrm{ctx}^{\mathcal{A}}] \\ \mathrm{ctx}^{\mathcal{A}}(u) \circ \mathrm{ctx}^{\mathcal{A}}(v) \ . & \end{aligned} \tag{27}$$

Next we show that for all $X$ in the image of $\alpha$ we have $\alpha(\mathrm{Fn}_{\mathcal{G}}(\gamma(\vec{\boldsymbol{X}}))) = \mathrm{Fn}^{\mathcal{A}}_{\mathcal{G}}(\vec{\boldsymbol{X}})$:

$$
\begin{aligned}
\alpha(\mathrm{Fn}_{\mathcal{G}}(\gamma(\vec{\boldsymbol{X}}))) = & \quad [\text{definition of } \mathrm{Fn}_{\mathcal{G}}] \\
\langle \alpha(\textstyle\bigcup_{X_i \to X_j X_k \in P} \gamma(\vec{\boldsymbol{X}}_j)\gamma(\vec{\boldsymbol{X}}_k))\rangle_{i\in[0,n]} = & \quad [\text{definition of } \alpha] \\
\langle \lfloor \{\mathrm{ctx}^{\mathcal{A}}(w) \mid w \in \textstyle\bigcup_{X_i \to X_j X_k \in P} \gamma(\vec{\boldsymbol{X}}_j)\gamma(\vec{\boldsymbol{X}}_k)\}\rfloor\rangle_{i\in[0,n]} = & \\
\langle \lfloor \{\mathrm{ctx}^{\mathcal{A}}(w) \mid \exists X_i \to X_j X_k \in P,\ w \in \gamma(\vec{\boldsymbol{X}}_j)\gamma(\vec{\boldsymbol{X}}_k)\}\rfloor\rangle_{i\in[0,n]} = & \quad [\text{definition of concatenation}] \\
\langle \lfloor \{\mathrm{ctx}^{\mathcal{A}}(uv) \mid \exists X_i \to X_j X_k \in P,\ u \in \gamma(\vec{\boldsymbol{X}}_j) \wedge v \in \gamma(\vec{\boldsymbol{X}}_k)\}\rfloor\rangle_{i\in[0,n]} = & \quad [\text{by 27}] \\
\langle \lfloor \{\mathrm{ctx}^{\mathcal{A}}(u) \circ \mathrm{ctx}^{\mathcal{A}}(v) \mid \exists X_i \to X_j X_k \in P,\ u \in \gamma(\vec{\boldsymbol{X}}_j) \wedge v \in \gamma(\vec{\boldsymbol{X}}_k)\}\rfloor\rangle_{i\in[0,n]} = & \quad [\text{definition of } X \circ Y] \\
\langle \lfloor \{\mathrm{ctx}^{\mathcal{A}}(u) \mid u \in \gamma(\vec{\boldsymbol{X}}_j), X_i \to X_j X_k\} \circ \{\mathrm{ctx}^{\mathcal{A}}(v) \mid v \in \gamma(\vec{\boldsymbol{X}}_k), X_i \to X_j X_k\}\rfloor\rangle_{i\in[0,n]} = & \quad [\lfloor X \circ Y\rfloor = \lfloor \lfloor X\rfloor \circ \lfloor Y\rfloor\rfloor] \\
\langle \lfloor \lfloor \{\mathrm{ctx}^{\mathcal{A}}(u) \mid u \in \gamma(\vec{\boldsymbol{X}}_j), X_i \to X_j X_k\}\rfloor \circ \lfloor \{\mathrm{ctx}^{\mathcal{A}}(v) \mid v \in \gamma(\vec{\boldsymbol{X}}_k), X_i \to X_j X_k\}\rfloor\rfloor\rangle_{i\in[0,n]} = & \quad [\alpha(\gamma(X)) = \lfloor X\rfloor] \\
\langle \lfloor \lfloor \{\vec{\boldsymbol{X}}_j \mid X_i \to X_j X_k\}\rfloor \circ \lfloor \{\vec{\boldsymbol{X}}_k \mid X_i \to X_j X_k\}\rfloor\rfloor\rangle_{i\in[0,n]} = & \quad [\lfloor X \circ Y\rfloor = \lfloor \lfloor X\rfloor \circ \lfloor Y\rfloor\rfloor] \\
\langle \lfloor \{\vec{\boldsymbol{X}}_j \mid X_i \to X_j X_k\} \circ \{\vec{\boldsymbol{X}}_k \mid X_i \to X_j X_k\}\rfloor\rangle_{i\in[0,n]} = & \quad [\text{definition of } \circ] \\
\langle \lfloor \{\vec{\boldsymbol{X}}_j \circ \vec{\boldsymbol{X}}_k \mid X_i \to X_j X_k\}\rfloor\rangle_{i\in[0,n]} = & \quad [\text{definition of } \mathrm{Fn}^{\mathcal{A}}_{\mathcal{G}}] \\
\mathrm{Fn}^{\mathcal{A}}_{\mathcal{G}}(\vec{\boldsymbol{X}}) \ . & \qquad \square
\end{aligned}
$$

**Theorem 7.10.** *Let $\mathcal{G}$ be a CFG and $\mathcal{A}$ be a FA. The algorithm* `CFGIncS` *decides $L(\mathcal{G}) \subseteq L(\mathcal{A})$.*

For clarity, we first recall some of the notation used in this paper:

$$
\begin{aligned}
\mathrm{ctx}(u) &\triangleq \{(q,q') \mid u \in W_{q,q'}\} \\
\alpha(X) &\triangleq \lfloor \{\mathrm{ctx}^{\mathcal{A}}(x) \mid x \in X\}\rfloor \\
\gamma(Y) &\triangleq \{u \in \Sigma^* \mid \exists y \in Y, y \subseteq \mathrm{ctx}^{\mathcal{A}}(u)\} \\
\mathrm{Fn}^{\mathcal{A}}_{\mathcal{G}}(\langle X_i\rangle_{i\in[0,n]}) &\triangleq \langle \lfloor \{X_j \circ X_k \mid X_i \to X_j X_k\rfloor\}\rangle_{i\in[0,|\mathcal{V}|]}
\end{aligned}
$$

*Proof:* We show that all the premises of Theorem 7.3 are satisfied for $\langle D, \sqsubseteq\rangle = \langle \mathrm{AC}_{\langle \wp(Q\times Q),\subseteq\rangle}, \sqsubseteq\rangle$ and the maps $\alpha$ and $\gamma$ define before.

(*i*) Since $\rho_{\leq_{\mathcal{A}}}(X) = \gamma(\alpha(X))$, it follows from Lemmas 7.7 and C.1 that $\gamma(\alpha(L_2)) = L_2$. Furthermore, for all $a \in \Sigma$, $X \in \wp(\Sigma^*)$ we next show that $\alpha(aX = \alpha(a\gamma\alpha(X))$ and $\alpha(X = \alpha(\gamma\alpha(X)a)$.

$$
\begin{aligned}
\alpha(aX) = & \quad [\text{Galois Connection}] \\
\alpha(\gamma(\alpha(aX))) = & \quad [\text{Lemma 7.9}] \\
\alpha(\rho_{\leq_{\mathcal{A}}}(aX)) = & \quad [\text{Lemma C.1}] \\
\alpha(a\rho_{\leq_{\mathcal{A}}}(X)) = & \quad [\text{Lemma 7.9}] \\
\alpha(a\gamma\alpha(X)) &
\end{aligned}
\qquad
\begin{aligned}
\alpha(Xa) = & \quad [\text{Galois Connection}] \\
\alpha(\gamma(\alpha(Xa))) = & \quad [\text{Lemma 7.9}] \\
\alpha(\rho_{\leq_{\mathcal{A}}}(Xa)) = & \quad [\text{Lemma C.1}] \\
\alpha(\rho_{\leq_{\mathcal{A}}}(X)a) = & \quad [\text{Lemma 7.9}] \\
\alpha(\gamma\alpha(X)a) &
\end{aligned}
$$

(*ii*) It turns out that $\langle \mathrm{AC}_{\langle \wp(Q\times Q),\subseteq\rangle}, \sqsubseteq\rangle$ is ACC because $\leqslant_{\mathcal{A}_2}$ is a wqo. Moreover, the decidability of $\leqslant_{\mathcal{A}_2}$ entails that $W_1 \sqcup W_2 \triangleq \lfloor W_1 \cup W_2\rfloor$ is a computable binary lub in $\langle \mathrm{AC}_{\langle \wp(Q\times Q),\subseteq\rangle}, \sqsubseteq\rangle$, therefore $\langle \mathrm{AC}_{\langle \wp(Q\times Q),\subseteq\rangle}, \sqsubseteq\rangle$ is an effective domain.

(*iii*) $\alpha(\{b\}) = \{(q,q') \mid q \xrightarrow{b} q'\}$ and $\alpha(\varnothing) = \varnothing$, hence $\lfloor \alpha(\vec{\boldsymbol{b}})\rfloor$ is trivial to compute.

(*iv*) It follows from Lemma 7.9 (e).

(*v*) Next, we prove that for all $Y$ in the image of $\alpha$ we have $Y \sqsubseteq \alpha(L_2) \Leftrightarrow \forall y \in Y,\ (I \times F) \cap y \neq \varnothing$, which coincides with the check performed by lines 2-4 of algorithm `CFGIncS`.

$$
\begin{aligned}
Y \sqsubseteq \alpha(L_2) \Leftrightarrow & \quad [Y = \alpha(U) \text{ for some } U \in \wp(\Sigma^*)] \\
\alpha(U) \sqsubseteq \alpha(L_2) \Leftrightarrow & \quad [\text{Galois Connection}] \\
U \subseteq \gamma(\alpha(L_2)) \Leftrightarrow & \quad [\text{Lemmas C.1, 7.7 and 7.9}] \\
U \subseteq L_2 \Leftrightarrow & \quad [\text{definition of } \mathrm{ctx}_{\mathcal{A}}(u)] \\
\forall u \in U, \mathrm{ctx}^{\mathcal{A}}(u) \cap (I \times F) \neq \varnothing \Leftrightarrow & \quad [\text{definition of } \alpha] \\
\forall y \in Y, y \cap (I \times F) \neq \varnothing &
\end{aligned}
$$

It follows from these properties and Theorem 7.3 that algorithm `CFGIncS` solves the inclusion problem $L_1 \subseteq L_2$. $\square$

**Lemma 8.1.** *For all $X, Y, Z \subseteq \Sigma^*$ and $w \in \Sigma^*$:*
(a) $X \subseteq ZY^{-1} \Leftrightarrow XY \subseteq Z \Leftrightarrow Y \subseteq X^{-1}Z$.
(b) $wY \subseteq Z \Leftrightarrow Y \subseteq w^{-1}Z$ *and* $Xw \subseteq Z \Leftrightarrow X \subseteq Zw^{-1}$.

*Proof:*
(a) By definition, for any $u \in \Sigma^*$, $u \in ZY^{-1}$ iff $uY \subseteq Z$. Hence, $X \subseteq ZY^{-1} \Leftrightarrow \forall u \in X,\ uY \subseteq Z \Leftrightarrow XY \subseteq Z$. Symmetrically, $XY \subseteq Z \Leftrightarrow Y \subseteq X^{-1}Z$ holds.
(b) It is a particular case of (a). □

**Lemma 8.2.** *If $\mathcal{G}$ is a linear CFG then for all $\vec{\boldsymbol{X}}, \vec{\boldsymbol{Y}} \in \wp(\Sigma^*)^{|\mathcal{V}|}$, $\mathrm{Fn}_{\mathcal{G}}(\vec{\boldsymbol{X}}) \subseteq \vec{\boldsymbol{Y}} \Leftrightarrow \vec{\boldsymbol{Y}} \subseteq \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\vec{\boldsymbol{X}})$.*

*Proof:* For clarity, we only consider rules in $P$ of the form $X_i \to aX_j$. It is routine to include the other case ($X_i \to X_j b$) in the proof using the equivalence $X_j b \subseteq X_i$ iff $X_j \subseteq X_i b^{-1}$.

$$\begin{aligned}
\mathrm{Fn}_{\mathcal{G}}(\langle X_i \rangle_{i\in[0,n]}) \subseteq \langle Y_i \rangle_{i\in[0,n]} &\Leftrightarrow \\
\forall i \in [0,n], \textstyle\bigcup_{X_i \to aX_j \in P} aX_j \subseteq Y_i &\Leftrightarrow \\
\forall i,j \in [0,n], X_i \to aX_j \in P \Rightarrow aX_j \subseteq Y_i &\Leftrightarrow \quad \text{[Lemma 8.1]} \\
\forall i,j \in [0,n], X_i \to aX_j \in P \Rightarrow X_j \subseteq a^{-1}Y_i &\Leftrightarrow \\
\forall j \in [0,n], X_j \subseteq \textstyle\bigcap_{X_i \to aX_j \in P} a^{-1}Y_i &\Leftrightarrow \\
\langle X_i \rangle_{i\in[0,n]} \subseteq \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\langle Y_i \rangle_{i\in[0,n]}) &
\end{aligned}$$

□

**Theorem 8.4.** *Let $\mathcal{G} = \langle \mathcal{V}, \Sigma, P \rangle$ be a linear CFG and let $\mathcal{A}$ be an FA with $L_2 = \mathcal{L}(\mathcal{A})$. If $\rho \in \mathrm{uco}(\wp(\Sigma^*))$ satisfies:*
(1) $\rho(L_2) = L_2$;
(2) *$\rho$ is backward complete for $\lambda X.\, aX$ and $\lambda X.\, Xa$ for all $a \in \Sigma$*
*then $\mathcal{L}(\mathcal{G}) \subseteq L_2$ iff $\vec{\boldsymbol{b}} \subseteq \mathrm{gfp}(\lambda \vec{\boldsymbol{X}}.\, \rho(\vec{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\vec{\boldsymbol{X}})))$. Moreover, the Kleene iterates coincide in lockstep with those of $\mathrm{gfp}(\lambda \vec{\boldsymbol{X}}.\, \vec{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\vec{\boldsymbol{X}}))$.*

*Proof:* Theorem 7.2 shows that if $\rho$ is backward complete for $\lambda X.\, aX$ and $\lambda X.\, Xa$ for all $a \in \Sigma$ then it is backward complete for $\mathrm{Fn}_{\mathcal{G}}$. Thus, by Lemma 8.3, $\rho$ is forward complete for $\widetilde{\mathrm{Fn}}_{\mathcal{G}}$. Hence $\rho$ is forward complete for $\lambda \vec{\boldsymbol{Y}}.\, \vec{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\vec{\boldsymbol{Y}})$, because:

$$\begin{aligned}
\rho(\vec{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\rho(\vec{\boldsymbol{Y}}))) = &\quad \text{[by forward completeness, } \rho(L_2) = L_2] \\
\rho(\rho(\vec{\boldsymbol{L_2}}^{X_0}) \cap \rho(\widetilde{\mathrm{Fn}}_{\mathcal{G}}(\rho(\vec{\boldsymbol{Y}})))) = &\quad [\rho(\cap \rho(Y)) = \cap \rho(Y)] \\
\rho(\vec{\boldsymbol{L_2}}^{X_0}) \cap \rho(\widetilde{\mathrm{Fn}}_{\mathcal{G}}(\rho(\vec{\boldsymbol{Y}}))) = &\quad \text{[by forward completeness, } \rho(L_2) = L_2] \\
\vec{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\rho(\vec{\boldsymbol{Y}})) &
\end{aligned}$$

Due to Equation (14), it follows that $\mathcal{L}(\mathcal{G}) \subseteq L_2$ iff $\vec{\boldsymbol{b}} \subseteq \mathrm{gfp}(\lambda \vec{\boldsymbol{Y}}.\, \vec{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\vec{\boldsymbol{Y}}))$. Finally, observe that the Kleene iterates computing $\mathrm{gfp}(\lambda \vec{\boldsymbol{Y}}.\, \vec{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\vec{\boldsymbol{Y}}))$ and those computing $\mathrm{gfp}(\lambda \vec{\boldsymbol{Y}}.\, \vec{\boldsymbol{L_2}}^{X_0} \cap \widetilde{\mathrm{Fn}}_{\mathcal{G}}(\rho(\vec{\boldsymbol{Y}})))$ coincide in lockstep since $\rho f \rho = f \rho$ and $\rho(L_2) = L_2$. □