On XPath with transitive axes and data tests

Diego Figueira

On XPath with Transitive Axes and Data Tests*

Diego Figueira
University of Edinburgh
Edinburgh, UK

ABSTRACT

We study the satisfiability problem for XPath with data equality tests. XPath is a node selecting language for XML documents whose satisfiability problem is known to be undecidable, even for very simple fragments. However, we show that the satisfiability for XPath with the rightward, leftward and downward reflexive-transitive axes (namely following-sibling-or-self, preceding-sibling-or-self, descendant-or-self) is decidable. Our algorithm yields a complexity of 3ExpSpace, and we also identify an expressive-equivalent normal form for the logic for which the satisfiability problem is in 2ExpSpace. These results are in contrast with the undecidability of the satisfiability problem as soon as we replace the reflexive-transitive axes with just transitive (non-reflexive) ones.

Categories and Subject Descriptors

I.7.2 [**Document Preparation**]: Markup Languages; H.2.3 [**Database Management**]: Languages; H.2.3 [**Languages**]: Query Languages

General Terms

Algorithms, Languages

Keywords

XML, XPath, unranked unordered tree, reflexive transitive axes, data-tree, infinite alphabet, data values

*We acknowledge the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the FET-Open grant agreement FOX, number FP7-ICT-233599.

1. INTRODUCTION

The simplest way of abstracting an XML document is by seeing it as a tree over a finite alphabet of *tags* or *labels*. However, this abstraction ignores all actual data stored in the document attributes. This is why there has been an increasing interest in data trees: trees that also carry data from an infinite domain. Here, we consider an XML modeled as an unranked ordered finite tree whose every node contains a label, and a vector of data values, one for each attribute. Labels belong to some finite alphabet, and data values to some infinite domain. We call these models *multi-attribute data trees* (see Figure 1). We study logics on these models, that can express data properties, namely equality of attributes' data values.

Here, we show decidability of the satisfiability problem for XPath where navigation can be done going downwards, rightwards or leftwards in the XML document, that is, where navigation is done using the reflexive-transitive XPath axes descendant-or-self, following-sibling-or-self, and preceding-sibling-or-self.

Formalisms for trees with data values

Several formalisms have been studied lately in relation to static analysis on trees with data values.

First-order logic.

One such formalisms is $\mathrm{FO}^2(<_h, succ_h, <_v, succ_v, \sim)$, first order logic with two variables, and binary relations to navigate the tree: the descendant $<_v$, child $succ_v$, next sibling $succ_h$ and following sibling $<_h$ (*i.e.*, the transitive closure of $succ_h$); and an equivalence relation $\sim$ to express that two nodes of the trees have the same data value. Although the decidability status for the satisfiability problem of this logic is unknown, it is known to be as hard as the reachability problem for BVASS (Branching Vector Addition System with States) [4]. If the signature has only the child and next sibling relation—$\mathrm{FO}^2(succ_h, succ_v, \sim)$—the logic is decidable in 3NExpTime as shown in [4].

Automata.

There have also been works on automata models for trees with data. Tree automata with registers to store and compare data values were studied in [20] as an extension to a similar model on words [19, 22]. A decidable alternating version of these automata called ATRA was studied in [18], and it was extended in [9, 12] to show decidability of the satisfiability problem for forward-XPath. The work [3] introduces a simple yet powerful automata model called *Class Automata*

on data trees that can capture $\mathrm{FO}^2(<_h, succ_h, <_v, succ_v, \sim)$, XPath, ATRA, and other models. Although its emptiness problem is undecidable, classes of data trees for which it is decidable are studied in [1]. Other formalisms include tree automata with set and linear constraints on cardinalities of sets of data values [6, 23].

XPath.

Here we concentrate on XPath, which is incomparable in terms of expressiveness with all the previously mentioned formalisms (except for Class Automata).

XPath is arguably the most widely used XML query language. It is implemented in XSLT and XQuery and it is used as a constituent part of several specification and update languages. XPath is fundamentally a general purpose language for addressing, searching, and matching pieces of an XML document. It is an open standard and constitutes a World Wide Web Consortium (W3C) Recommendation [5].

Query containment and query equivalence are important static analysis problems, which are useful to, for example, query optimization tasks. In logics closed under boolean operators—as the one treated here—, these problems reduce to checking for *satisfiability*: Is there a document on which a given query has a non-empty result? By answering this question we can decide at compile time whether the query contains a contradiction and thus the computation of the query (or subquery) on the document can be avoided. Or, by answering the query equivalence problem, one can test if a query can be safely replaced by another one which is more optimized in some sense (*e.g.*, in the use of some resource). Moreover, the satisfiability problem is crucial for applications on security [7], type checking transformations [21], and consistency of XML specifications.

Core-XPath (term coined in [17]) is the fragment of XPath 1.0 that captures all the navigational behavior of XPath. It has been well studied and its satisfiability problem is known to be decidable even in the presence of DTDs. The extension of this language with the possibility to make equality and inequality tests between attributes of elements in the XML document is named Core-Data-XPath in [4].

In an nutshell, the important formulas of Core-Data-XPath (henceforth *XPath*) are of the form

$$\langle \alpha @_i = \beta @_j \rangle,$$

where α, β are *path expressions*, that navigate the tree using *axes*: descendant, child, ancestor, next-sibling, etc. and can make tests in intermediary nodes. Such a formula is true at a node x of a multi-attribute data tree if there are two nodes y, z in the tree that can be reached with the relations denoted by α, β respectively, so that the ith attribute of y carries the *same* datum as the jth attribute of z.

Unfortunately, the satisfiability problem for XPath is undecidable [16]. How can we regain decidability for satisfiability of XPath then? We can restrict the models, or restrict the logic. The first possibility is to restrict the classes of documents in which one is interested, which is the approach taken in [1]. Another, more studied, approach is to restrict the syntax, which is the one taken here. One way to regain decidability is to syntactically restrict the amount of nodes that the XPath properties can talk about. In this vein, there have been studies on fragments without negation or without transitive axes [2, 16]. These fragments enjoy a small model property and are decidable. However, they cannot state global properties, involving all the nodes in an XML document. Ideally, we seek fragments with the following desirable features

- closed under boolean operators,
- having as much freedom as possible to navigate the tree in many directions: up, down, left, right,
- having the possibility to reach any node of the tree, with transitive axes, like descendant, following sibling (the transitive closure of the next sibling axis), etc.

However, decidability results are scarce, and most fragments with the characteristics just described are undecidable. There are, however, some exceptions ([10]).

- The *downward* fragment of XPath, containing the child and descendant axes, is decidable, ExpTime-complete [8, 13].
- The *forward* fragment of XPath, extending the downward fragment with the next sibling and the following sibling axes, is decidable with non-primitive recursive complexity [9, 12].
- The *vertical* fragment of XPath, extending the downward fragment with the parent and ancestor axes, is decidable with non-primitive recursive complexity [15].
- A last example is the present work: XPath with the reflexive transitive closure of the child, next-sibling and previous-sibling relations is decidable.

All the non-primitive recursive (NPR) upper bounds of the forward and vertical fragments are also matched with NPR lower bounds. That is, there is no primitive recursive function that bounds the time or space needed by any algorithm that computes the satisfiability for any of these two logics. Moreover, it is known that any fragment of XPath containing a transitive rightward, leftward or upward axis has a satisfiability problem which is either undecidable or decidable with a NPR lower bound [14].[1] Further, as soon as we have both the rightward and leftward transitive axes, the satisfiability becomes undecidable [14]. (Indeed, the downward fragment of XPath seemed to be the only one with elementary complexity up to now.)

The aforementioned hardness results make use of non-reflexive transitive relations. Surprisingly, the reductions do not seem to work when the relations are also reflexive. What is then the decidability status of the fragments of XPath with reflexive-transitive relations? This was a question raised in [14].

A partial answer to this question was given in [11]. There, it was shown that XPath restricted to data words is decidable even when we have both a reflexive-transitive future and past relations. (One can think of data words as XML documents of height 1, with only one attribute per node.) This result may seem surprising taking into account that if one of these relations is non-reflexive it is no longer decidable; and if we have only one non-reflexive transitive relation it is decidable with non-primitive recursive complexity. In [11] it was shown that the satisfiability problem is in 2ExpSpace (or ExpSpace if we adopt a certain normal form of

[1] These are the axes that are called preceding-sibling, following-sibling and ancestor in the XPath jargon.

the formulas). This was a first step in our study of the computational behavior of XPath with reflexive-transitive axes. The present work corresponds to the second part, in which we study XPath on XML documents (*i.e.*, trees) instead of words.

Contribution

We show decidability of the satisfiability for XPath with data equality tests between attributes, where navigation can be done going downwards, rightwards or leftwards in the XML document. The navigation can only be done by reflexive-transitive relations. These correspond to the XPath axes: preceding-sibling-or-self, following-sibling-or-self, and descendant-or-self axes.[2] Here we denote these axes with ${}^*\!\leftarrow$, $\rightarrow^*$ and $\downarrow_*$ respectively. As already mentioned, the fact that the relations are reflexive-transitive (as opposed to just transitive) is an essential feature to achieve decidability. Given the known complexity results on XPath, this fragment seems to be in balance between navigation and complexity. This work then argues in favor of studying XPath-like logics for trees with data with reflexive-transitive relations, since they may behave computationally much better than the non-reflexive counterpart, as evidenced here.

The extension of the prior work [11] on data words to work with trees with a descendant axis is highly non-trivial, requiring an altogether different formalism and algorithm strategy. Whereas in [11] the main object of study is a transition system—which comes naturally when working with words—this does not adapt well to working with trees. Instead, here we work with an algebra operating on abstractions of *forests* of multi-attribute data trees. Over this algebra, we prove some monotonicity properties, which are necessarily more involved than those used in [11] to account for the interplay between horizontal and vertical navigation of the logic.

Our algorithm yields a 3ExpSpace upper bound for the satisfiability problem of this XPath fragment. We also show that this can be lowered to 2ExpSpace if we work with an expressive-equivalent normal form, called *direct normal form*. Since XPath with just one reflexive-transitive relation is already ExpSpace-hard (even when the formula is in direct normal form) by [11], we cannot aim for much better complexities.

Due to space limitations, all the proofs are included in the Appendix, although the main intuitions, general strategy, and decomposition of the proof in simpler problems are presented in the body of the paper.

2. PRELIMINARIES

Basic notation.

Let $\mathbb{N}_0 \stackrel{def}{=} \{0, 1, 2, \dots\}$, $\mathbb{N} \stackrel{def}{=} \{1, 2, 3, \dots\}$, and let $[n] \stackrel{def}{=} \{1, \dots, n\}$ for any $n \in \mathbb{N}$. We fix once and for all $\mathbb{D}$ to be any infinite domain of data values; for simplicity in our examples we will consider $\mathbb{D} = \mathbb{N}_0$. In general we use the symbols $\mathbb{A}$, $\mathbb{B}$ for finite alphabets, and the symbols $\mathbb{E}$ and $\mathbb{F}$ for any kind of alphabet. By $\mathbb{E}^*$ we denote the set of finite sequences over $\mathbb{E}$, by $\mathbb{E}^+$ the set of finite sequences with at least one element over $\mathbb{E}$. We write 'ϵ' for the empty sequence and '$\cdot$' as the concatenation operator between sequences. By $|S|$ we denote the length of S (if S is a finite sequence), or its cardinality (if S is a set). We use $(a_i)_{i\in\{j,\dots,j+n\}}$ as short for $a_j a_{j+1} \cdots a_{j+n}$.

[2]Strictly speaking, these axes do not exist in XPath [5]. They must be interpreted as the reflexive version of the preceding-sibling, following-sibling and descendant axes respectively.

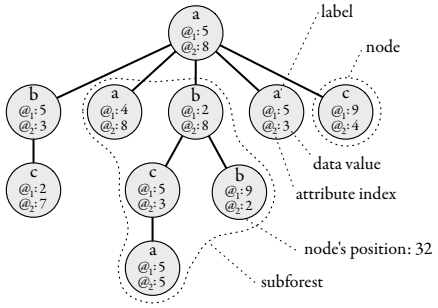


Figure 1: A multi-attribute data tree, where $\mathbb{A} = \{a, b, c\}$ and $\mathbf{k} = 2$.

Unranked finite trees with data.

By *Trees*($\mathbb{E}$) we denote the set of finite ordered and unranked trees over an alphabet $\mathbb{E}$. We view each **position** in a tree as an element of $\mathbb{N}^*$. Formally, we define $\mathsf{POS} \subseteq 2^{\mathbb{N}^*}$ as the set of sets of finite tree positions, such that: $X \in \mathsf{POS}$ iff (a) $X \subseteq \mathbb{N}^*, |X| < \infty$; (b) X is prefix-closed; and (c) if $n{\cdot}(i+1) \in X$ for $i \in \mathbb{N}$, then $n{\cdot}i \in X$. A tree is then a mapping from a set of positions to labels of the alphabet $\mathit{Trees}(\mathbb{E}) \stackrel{def}{=} \{\mathbf{t} : P \to \mathbb{E} \mid P \in \mathsf{POS}\}$. The root's position is the empty string ϵ. The position of any other node in the tree is the concatenation of the position of its parent and the node's index in the ordered list of siblings.

Given a tree $\mathbf{t} \in \mathit{Trees}(\mathbb{E})$, $\mathit{pos}(\mathbf{t})$ denotes the domain of $\mathbf{t}$, which consists of the set of positions of the tree, and $\mathsf{alph}(\mathbf{t}) = \mathbb{E}$ denotes the alphabet of the tree. From now on, we informally refer by 'node' to a position x together with the value $\mathbf{t}(x)$.

Given two trees $\mathbf{t}_1 \in \mathit{Trees}(\mathbb{E})$, $\mathbf{t}_2 \in \mathit{Trees}(\mathbb{F})$ such that $\mathit{pos}(\mathbf{t}_1) = \mathit{pos}(\mathbf{t}_2) = P$, we define $\mathbf{t}_1 \otimes \mathbf{t}_2 : P \to (\mathbb{E}{\times}\mathbb{F})$ as $(\mathbf{t}_1 \otimes \mathbf{t}_2)(x) \stackrel{def}{=} (\mathbf{t}_1(x), \mathbf{t}_2(x))$.

The set of **multi-attribute data trees** over a finite alphabet $\mathbb{A}$ of labels, $\mathbf{k}$ different attributes, and an infinite domain $\mathbb{D}$ is defined as $\mathit{Trees}(\mathbb{A}{\times}\mathbb{D}^{\mathbf{k}})$. Note that every tree $\mathbf{t} \in \mathit{Trees}(\mathbb{A}{\times}\mathbb{D}^{\mathbf{k}})$ can be decomposed into two trees $\mathbf{a} \in \mathit{Trees}(\mathbb{A})$ and $\mathbf{d} \in \mathit{Trees}(\mathbb{D}^{\mathbf{k}})$ such that $\mathbf{t} = \mathbf{a} \otimes \mathbf{d}$. Figure 1 shows an example of a multi-attribute data tree. The notation for the set of data values used in a data tree is $\mathit{data}(\mathbf{a} \otimes \mathbf{d}) \stackrel{def}{=} \{\mathbf{d}(x)(i) \mid x \in \mathit{pos}(\mathbf{d}), i \in [\mathbf{k}]\}$. With an abuse of notation we write $\mathit{data}(X)$ to denote all the elements of $\mathbb{D}$ contained in X, for whatever object X may be.

A **forest** is a sequence of trees, and the set of **multi-attribute data forests** over $\mathbb{A}$ and $\mathbf{k}$ is $(\mathit{Trees}(\mathbb{A}{\times}\mathbb{D}^{\mathbf{k}}))^*$. We will normally use the symbol $\bar{\mathbf{t}}$ for a forest of multi-attribute data trees. That is, $\bar{\mathbf{t}} \in (\mathit{Trees}(\mathbb{A}{\times}\mathbb{D}^{\mathbf{k}}))^*$. (Note that in particular $\bar{\mathbf{t}}$ can be an *empty* forest.) For any $(a, \bar{d}) \in \mathbb{A} \times \mathbb{D}^{\mathbf{k}}$, let us write $(a, \bar{d})\bar{\mathbf{t}}$ for the multi-attribute data tree that results from adding $(a, \bar{d})$ as a root of $\bar{\mathbf{t}}$. We call this operation **rooting**. We will usually write $\mathbf{t}$ (resp. $\bar{\mathbf{t}}$) to denote multi-attribute data trees (resp. forests) and t (resp. $\bar{t}$) to denote trees (resp. forests) over a finite alphabet.

XPath.

Next we define transitive XPath, the fragment of XPath where all axes are reflexive and transitive.

Transitive XPath is a two-sorted language, with *path* expressions (that we write $\alpha, \beta, \gamma, \delta$) and *node* expressions (that we write φ, ψ, η). Path expressions denote binary relations, resulting from composing the descendant, ancestor, preceding sibling and following sibling relations (which are denoted respectively by $\downarrow_*$, $\uparrow^*$, ${}^*\!\leftarrow$, $\rightarrow^*$ respectively), and node expressions. Node expressions are boolean formulas that test a property of a node like, for example, that is has a certain label, or that it has a descendant labeled a with the same data value in attribute i as the attribute j of an ancestor labeled b, which is expressed by $\langle\downarrow_*[a]@_i = \uparrow^*[b]@_j\rangle$. We write XPath$(\downarrow_*, \uparrow^*, \rightarrow^*, {}^*\!\leftarrow, =)$ to denote this logic, and we write XPath$(\mathcal{O}, =)$ for some $\mathcal{O} \subseteq \{\downarrow_*, \uparrow^*, \rightarrow^*, {}^*\!\leftarrow\}$, to denote the logic containing only the axes in $\mathcal{O}$. A *formula* of XPath$(\downarrow_*, \uparrow^*, \rightarrow^*, {}^*\!\leftarrow, =)$ is either a node expression or a path expression of the logic. Its syntax and semantics are defined in Figure 2. As another example, we can select the nodes that have a sibling labeled a to the left whose first attribute is the same as the second attribute of some descendant of a right sibling by the formula $\varphi = \langle{}^*\!\leftarrow[a]@_1 = \rightarrow^*\downarrow_*@_2\rangle$. Given a tree $\mathbf{t}$ as in Figure 1, we have $[\![\varphi]\!]^{\mathbf{t}} = \{\epsilon, 2, 3, 4, 5, 311\}$.

We write $\mathbf{t}, x \models \varphi$ (resp. $\mathbf{t}, (x, y) \models \alpha$) for $x \in pos(\mathbf{t})$ (resp. $x, y \in pos(\mathbf{t})$) as short for $x \in [\![\varphi]\!]^{\mathbf{t}}$ (resp. $(x, y) \in [\![\alpha]\!]^{\mathbf{t}}$). We write $\mathbf{t} \models \varphi$ as short for $\epsilon \in [\![\varphi]\!]^{\mathbf{t}}$.

In the case of XPath$({}^*\!\leftarrow, \downarrow_*, \rightarrow^*, =)$, we also extend the evaluation to multi-attribute data forests. Let $(a, \bar{d})$ be an arbitrary fix element of $\mathbb{A} \times \mathbb{D}^{\mathbf{k}}$. Given a forest $\bar{\mathbf{t}}$ and $x, y \in pos((a, \bar{d})\bar{\mathbf{t}})$, $x, y \neq \epsilon$, we define the satisfaction relation $\models$, as $\bar{\mathbf{t}}, x \models \varphi$ (resp. $\bar{\mathbf{t}}, (x, y) \models \alpha$) if $(a, \bar{d})\bar{\mathbf{t}}, x \models \varphi$ (resp. $(a, \bar{d})\bar{\mathbf{t}}, (x, y) \models \alpha$). (Note that since XPath$({}^*\!\leftarrow, \downarrow_*, \rightarrow^*, =)$ has no ascending axes, whether $\bar{\mathbf{t}}, x \models \varphi$ or not does not depend on $(a, \bar{d})$, we use it as a simple way of defining its semantics.)

The **satisfiability problem** for XPath$(\mathcal{O}, =)$ (henceforth noted SAT-XPath$(\mathcal{O}, =)$) is the problem of, given a formula φ of XPath$(\mathcal{O}, =)$, wether there exists a multi-attribute data tree $\mathbf{t}$ such that $\mathbf{t} \models \varphi$.

3. PROOF SKETCH

The main contribution of this paper is the following.

THEOREM 3.1. *SAT-*XPath$({}^*\!\leftarrow, \downarrow_*, \rightarrow^*, =)$ *is decidable in* 3EXPSPACE.

We reduce the problem of whether a formula φ of our logic XPath$({}^*\!\leftarrow, \downarrow_*, \rightarrow^*, =)$ is satisfiable, to the problem of whether one can obtain an element with a certain property by repeated applications of operations in some algebra, starting from a basic set of elements. We call it the *derivation problem*. First we introduce the algebra (Section 4), we then solve the derivation problem (Section 5) and finally we show the reduction from the logic into the derivation problem (Section 6).

We first introduce *forest profiles* in Section 4, which constitute the algebra domain. A forest profile is an abstraction of a multi-attribute data forest inside a context, where the context consists of the two (possibly empty) forests that are to the left and to the right. Figure 3 depicts one such possible forest, together with the left and right context forests. A forest profile contains, for each data value d and each path expression, the information of whether d can be reached by the path expression, and *where* it can be reached, either

- inside the main forest, starting from the leftmost root (the node $\overrightarrow{x}^1$ in Figure 3),
- inside the main forest but starting with the rightmost root (the node $\overleftarrow{x}^1$ in Figure 3),
- in the left context forest (starting from the node $\overleftarrow{x}^{\mathrm{E}}$ in Figure 3), or
- in the right context forest (starting from the node $\overrightarrow{x}^{\mathrm{E}}$ in Figure 3).

In this setting, path expressions are called *patterns* and their navigation is greatly simplified. Patterns can go first to the left, and then down, or first to the right and then down, or only down. They correspond to path expressions like, for example, $\rightarrow^*[a]\rightarrow^*[b]\downarrow_*[c]\downarrow_*[a \vee b]@_1$, or ${}^*\!\leftarrow[\neg a]\downarrow_*[a]@_2$. Further, node expressions contained in patterns are simple boolean combinations of tests for labels.

A forest profile also keeps track of a set of important data values called the *rigid values*. These are data values that play a determined function in the forest containing the abstracted forest (*i.e.*, in the concatenation of the left, main and right forests). Intuitively, a data value is rigid in a forest if it can be pinpointed by a path expression, in the sense that it is the *only* data value that can be reached with some path expression $\alpha@_i$. At this level of detail, we just mention that some special care must be taken for these rigid data values.

We equip the set of forest profiles with two operations, one that corresponds to concatenating two of the forests being abstracted, and another operation that corresponds to adding a root to the forest, converting it into a tree. This algebra is introduced in Section 4.2. In particular, the root operation is restricted to work only with forest profiles that are from certain set of *consistent* profiles. Consistent profiles will play an important role in the reduction from the logic to the algebra. The idea is that they are those profiles that are not in contradiction with the formula φ to test for satisfiability, that is, that could abstract subforests of a model of φ.

A *root profile*, is a profile that comes from the application of the root operation with a certain label of a certain alphabet $\mathbb{A}_{root}$ of root labels. An *empty profile* is the profile corresponding to the empty forest with an empty context. In Section 5 we define the *derivation problem* for forest profiles as the problem of whether there is a way of obtaining a root profile from the empty profile by repeated applications of the algebra operations.

We show that the derivation problem is decidable in 2EXPSPACE in Section 5. We first define a partial ordering on profiles in Section 5.2, this ordering will be of chief importance in our decidability result. We show a series of monotonicity properties that show that the set of derivable profiles is upward-closed. The purpose of the partial ordering is to reduce the derivation problem on the infinite set of forest profiles into a problem on a *finite* set of minimal profiles. The fact that the derivable profiles is upward-closed is indeed a key ingredient for this reduction to work.

$$\alpha,\beta \;::=\; o \mid \alpha[\varphi] \mid [\varphi]\alpha \mid \alpha\beta \qquad o \in \{\varepsilon, \downarrow_*, \uparrow^*, \rightarrow^*, {}^*\!\leftarrow\}\ ,$$
$$\varphi,\psi \;::=\; a \mid \neg\varphi \mid \varphi\wedge\psi \mid \varphi\vee\psi \mid \langle\alpha\rangle \mid \langle\alpha@_i = \beta@_j\rangle \mid \langle\alpha@_i \neq \beta@_j\rangle \qquad a \in \mathbb{A}, i,j \in [\mathbf{k}]\ .$$

$$[\![\downarrow_*]\!]^{\mathbf{t}} = \{(x, x{\cdot}i) \mid x{\cdot}i \in pos(\mathbf{t})\}^* \qquad [\![\uparrow^*]\!]^{\mathbf{t}} = \{(x{\cdot}i, x) \mid x{\cdot}i \in pos(\mathbf{t})\}^*$$
$$[\![\rightarrow^*]\!]^{\mathbf{t}} = \{(x{\cdot}i, x{\cdot}(i+1)) \mid x{\cdot}i, x{\cdot}(i+1) \in pos(\mathbf{t})\}^* \qquad [\![{}^*\!\leftarrow]\!]^{\mathbf{t}} = \{(x{\cdot}(i+1), x{\cdot}i) \mid x{\cdot}i, x{\cdot}(i+1) \in pos(\mathbf{t})\}^*$$
$$[\![\varepsilon]\!]^{\mathbf{t}} = \{(x,x) \mid x \in pos(\mathbf{t})\} \qquad [\![\alpha\beta]\!]^{\mathbf{t}} = \{(x,z) \mid \text{ there exists } y \text{ such that } (x,y) \in [\![\alpha]\!]^{\mathbf{t}}, (y,z) \in [\![\beta]\!]^{\mathbf{t}}\}$$
$$[\![[\varphi]]\!]^{\mathbf{t}} = \{(x,x) \in \mid x \in pos(\mathbf{t}), x \in [\![\varphi]\!]^{\mathbf{t}}\} \qquad [\![\langle\alpha\rangle]\!]^{\mathbf{t}} = \{x \in pos(\mathbf{t}) \mid \exists y.(x,y) \in [\![\alpha]\!]^{\mathbf{t}}\}$$
$$[\![a]\!]^{\mathbf{t}} = \{x \in pos(\mathbf{t}) \mid \mathbf{a}(x) = a\} \qquad [\![\varphi\wedge\psi]\!]^{\mathbf{t}} = [\![\varphi]\!]^{\mathbf{t}} \cap [\![\psi]\!]^{\mathbf{t}}$$
$$[\![\neg\varphi]\!]^{\mathbf{t}} = pos(\mathbf{t}) \setminus [\![\varphi]\!]^{\mathbf{t}}$$
$$[\![\langle\alpha@_i{=}\beta@_j\rangle]\!]^{\mathbf{t}} = \{x \in pos(\mathbf{t}) \mid \exists y,z\ (x,y) \in [\![\alpha]\!]^{\mathbf{t}}, (x,z) \in [\![\beta]\!]^{\mathbf{t}}, \mathbf{d}(y)(i) = \mathbf{d}(z)(j)\}$$
$$[\![\langle\alpha@_i{\neq}\beta@_j\rangle]\!]^{\mathbf{t}} = \{x \in pos(\mathbf{t}) \mid \exists y,z\ (x,y) \in [\![\alpha]\!]^{\mathbf{t}}, (x,z) \in [\![\beta]\!]^{\mathbf{t}}, \mathbf{d}(y)(i) \neq \mathbf{d}(z)(j)\}$$

Figure 2: The syntax of transitive XPath; and its semantics for a multi-attribute data tree $\mathbf{t} = \mathbf{a} \otimes \mathbf{d}$.

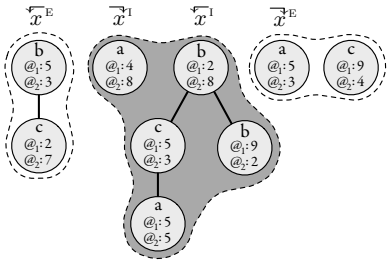


Figure 3: A multi-attribute data forest, with its left and right forests.

However, one problem we need to face is that the ordering has infinite antichains: every two profiles with different set of rigid values are incomparable. We tackle this in Section 5.4, where we show that we can bound the set of rigid values, obtaining an equivalent derivation problem on profiles with a small set of rigid values. Once we obtain this bound, the set of minimal profiles becomes finite, doubly exponential. Next, in Section 5.5 we show that, thanks to the monotonicity properties enjoyed by the algebra, we can work only with minimal elements. Finally, in Section 5.6 we give the concrete saturation-style algorithm that solves the derivation problem using doubly exponential space.

In Section 6 we show that the satisfiability problem for XPath(${}^*\!\leftarrow, \downarrow_*, \rightarrow^*, =$) can be reduced to the derivation problem in ExpSpace. In Section 6.1 we show a normal form, called *direct unnested normal form*, where direct unnested path expressions correspond, precisely, to the pattern expressions used in the forest profiles (basically all path expressions are of the form already described). We then show in Section 6.2 that one can reduce, in ExpSpace, the satisfiability problem for formulas in this normal form into the derivation problem, obtaining a 3ExpSpace decidability procedure for SAT-XPath(${}^*\!\leftarrow, \downarrow_*, \rightarrow^*, =$), obtaining Theorem 3.1.

4. FOREST PROFILES

We define abstractions of forests of multi-attribute data trees. These are called *forest profiles*. They are the main construct in our solution. One must think of a forest profile as the description, for every data value $d \in \mathbb{D}$, of all the possible ways of reaching the data value d via path expressions of XPath(${}^*\!\leftarrow, \downarrow_*, \rightarrow^*, =$). Some ways of reaching the data value may lie inside the forest being abstracted, and some outside the forest. Take for instance the forest in the middle of Figure 3. For every forest there we identify 4 nodes: the leftmost root, the rightmost root, the node to the left of the leftmost root (if any), and the node to the right of the rightmost root (if any). These are the nodes identified by $\overleftarrow{x}^{\mathrm{I}}, \overrightarrow{x}^{\mathrm{I}}, \overleftarrow{x}^{\mathrm{E}}, \overrightarrow{x}^{\mathrm{E}}$ respectively in the figure. The profile of this forest is represented by all the paths that can reach the data value 4, all those that can reach 2, etc. Take as an example the data value 5; this data value can be reached by

(i) $\rightarrow^*[a]@_1$ from $\overrightarrow{x}^{\mathrm{E}}$,

(ii) ${}^*\!\leftarrow[b]\downarrow_*[a]@_1$ from $\overleftarrow{x}^{\mathrm{I}}$,

(iii) $\rightarrow^*[a]\rightarrow^*[b]\downarrow_*[c]@_1$ from $\overrightarrow{x}^{\mathrm{I}}$, etc.

Remember that expressions are evaluated in a forest and, for example, an expression starting with $\rightarrow^*$ denotes the possibility to move forward in the sequence of tree roots of the forest. The idea is that we limit ourselves that whenever there are paths departing from $\overrightarrow{x}^{\mathrm{I}}$ or $\overleftarrow{x}^{\mathrm{I}}$ they must be *internal* to the forest (*i.e.*, internal to the gray forest in Figure 3), whenever there are paths from $\overrightarrow{x}^{\mathrm{E}}$ or $\overleftarrow{x}^{\mathrm{E}}$ they must be *external* to the forest (*i.e.*, either in the forest depicted to the left or to the right of the gray forest in Figure 3).

Let $\mathbb{A}$ be a finite alphabet of **labels**, let $\mathbb{A}_{root} \subseteq \mathbb{A}$ be the set of **root labels**, and let $\mathbb{D}$ be an infinite domain of **data values**. The set $\mathcal{B}(\mathbb{A})$ is the boolean closure of tests for labels from $\mathbb{A}$. For any $a \in \mathbb{A}$ and $\psi \in \mathcal{B}(\mathbb{A})$, we write $a \models \psi$ if the interpretation assigning *true* to a, and *false* to every other $b \in \mathbb{A}$, satisfies ψ. Let $\mathbf{k} \in \mathbb{N}$ be a fixed natural number, corresponding to the number of **attributes** at each node. We say that $i \in [\mathbf{k}]$ is an **attribute index**. We define the set of **patterns**, as any finite, subword-closed, subset of $(\mathcal{B}(\mathbb{A}))^*$, and we denote it by $\mathcal{P}$. We generally use the symbols $\alpha, \beta, \gamma, \delta \in \mathcal{P}$ to denote patterns. For every label $a \in \mathbb{A}$ we define the following set of patterns $\boldsymbol{\sigma}_a \subseteq \mathcal{P}$

$$\boldsymbol{\sigma}_a \stackrel{def}{=} \{\psi_1 \cdots \psi_k \in \mathcal{P} \mid a \models \psi_1 \wedge \cdots \wedge \psi_k\}.$$

Note that $\epsilon \in \boldsymbol{\sigma}_a$. The set of **composed patterns** is

$$\Pi \stackrel{def}{=} (\mathcal{P} \setminus \{\epsilon\}) \times \mathcal{P} \times [\mathbf{k}].$$

The intended meaning is that the first component operates on the siblings, the second on a downward path, and the third retrieves a data value from an attribute index. We will sometimes use the symbol $\bar{\alpha}$ to represent elements from Π, or (α, β, i) if we need to explicit the components of the composed pattern.

A **forest profile** $\mathfrak{f}$ is a tuple

$$\mathfrak{f} \;=\; (\overleftarrow{\chi}^{\mathrm{E}}, \overrightarrow{\chi}^{\mathrm{I}}, \overleftarrow{\chi}^{\mathrm{I}}, \overrightarrow{\chi}^{\mathrm{E}}, R)$$

where $R \subseteq \mathbb{D}$, and we call it the set of **rigid values** of $\mathfrak{f}$, and $\overleftarrow{\chi}^{\mathrm{E}}, \overrightarrow{\chi}^{\mathrm{I}}, \overleftarrow{\chi}^{\mathrm{I}}, \overrightarrow{\chi}^{\mathrm{E}} \subseteq \mathbb{D} \times \Pi$, and we call them the set of left/right external/internal **descriptions** respectively. In the example before, one shall interpret (i) as $(5, a, \epsilon, 1) \in \overrightarrow{\chi}^{\mathrm{E}}$, (ii) as $(5, b, a, 1) \in \overleftarrow{\chi}^{\mathrm{I}}$ and (iii) as $(5, a{\cdot}b, c, 1) \in \overrightarrow{\chi}^{\mathrm{I}}$. We use χ to denote a subset of $\mathbb{D} \times \Pi$; and we write $\bar{\chi}$ (resp. $\bar{\chi}_i$) to denote the 4-uple $(\overleftarrow{\chi}^{\mathrm{E}}, \overrightarrow{\chi}^{\mathrm{I}}, \overleftarrow{\chi}^{\mathrm{I}}, \overrightarrow{\chi}^{\mathrm{E}})$ (resp. $(\overleftarrow{\chi_i}^{\mathrm{E}}, \overrightarrow{\chi_i}^{\mathrm{I}}, \overleftarrow{\chi_i}^{\mathrm{I}}, \overrightarrow{\chi_i}^{\mathrm{E}})$). Likewise, we use $\mathfrak{f}$ (resp. $\mathfrak{f}_i$) to denote $(\bar{\chi}, R)$ (resp. $(\bar{\chi}_i, R_i)$).

We define, for every $\chi \subseteq \mathbb{D} \times \Pi$,

$$\begin{aligned}
\chi(d) &\stackrel{def}{=} \{(\alpha, \beta, i) \in \Pi \mid (d, \alpha, \beta, i) \in \chi\},\\
\chi(\alpha, \beta, i) &\stackrel{def}{=} \{d \in \mathbb{D} \mid (d, \alpha, \beta, i) \in \chi\}, \text{ and}\\
\bar{\chi}(d) &\stackrel{def}{=} (\overleftarrow{\chi}^{\mathrm{E}}(d), \overrightarrow{\chi}^{\mathrm{I}}(d), \overleftarrow{\chi}^{\mathrm{I}}(d), \overrightarrow{\chi}^{\mathrm{E}}(d)).
\end{aligned}$$

We define $data(\mathfrak{f}) \stackrel{def}{=} R \cup \{d \in \mathbb{D} \mid \bar{\chi}(d) \neq (\emptyset, \emptyset, \emptyset, \emptyset)\}$. We call $data(\mathfrak{f}) \setminus R$ the set of **flexible values** of $\mathfrak{f}$. We use the symbol $\bar{\pi}$ to denote $(\overleftarrow{\pi}^{\mathrm{E}}, \overrightarrow{\pi}^{\mathrm{I}}, \overleftarrow{\pi}^{\mathrm{I}}, \overrightarrow{\pi}^{\mathrm{E}})$ where $\overleftarrow{\pi}^{\mathrm{E}}, \overrightarrow{\pi}^{\mathrm{I}}, \overleftarrow{\pi}^{\mathrm{I}}, \overrightarrow{\pi}^{\mathrm{E}} \subseteq \Pi$. We further say that $\bar{\pi}$ is **the description of** $d \in \mathbb{D}$ **in** $\mathfrak{f}$ if $\bar{\chi}(d) = \bar{\pi}$.

4.1 Rigid and flexible values

In a forest satisfying some XPath formula, different data values have different roles. We distinguish here two categories of data values: rigid and flexible. Rigid data values are important for the satisfaction of the formula and special care is needed to treat these, whereas flexible values are not crucial, and they can be sometimes removed from the tree. Let us give some more precise intuition. We use the logic XPath to make this intuition clear, but we will then state the definitions in terms of forest profiles.

Given a multi-attribute data forest $\bar{\mathbf{t}}$ where $\bar{\mathbf{t}}, i \models \varphi$, suppose there is a data value d such that: there is some position $1 \leq j \leq |\bar{\mathbf{t}}|$ and some path expression α of φ of the form $\alpha = {\rightarrow}^* \beta @_k$ or $\alpha = {}^*{\leftarrow} \beta @_k$ so that d is the *only* data value that can be reached through α from j. When there is such a d we call it a **rigid value** for j, since the logic can identify it and pinpoint it from the rest of the data values. If d is rigid for at least one position $j \in \{1, \ldots, |\bar{\mathbf{t}}|\}$ we say that d is rigid for $\bar{\mathbf{t}}$. All the remaining data values of $\bar{\mathbf{t}}$ (which are the **flexible values**) play the role of assuring that "there are at least two data values reachable through α from position j" for some α and j. As such, its importance is only relative. In particular, if $\bar{\mathbf{t}}$ is a forest satisfying φ and containing d as a flexible value, consider $\bar{\mathbf{t}}'$ as the result of replacing, for some fresh data value d', every tree $\mathbf{t}''$ of $\bar{\mathbf{t}}$ with the forest $\mathbf{t}''{\cdot}(\mathbf{t}''[d \mapsto d'])$, where $\mathbf{t}''[d \mapsto d']$ is the result of replacing the data value d with d' in $\mathbf{t}''$, and leaving all the structures and labels as they were. Indeed, $\bar{\mathbf{t}}'$ will continue to satisfy φ; but this is not necessarily true if d was a rigid value. The same notions hold for our algebra on forest profiles. This is a key property that we need to exploit and hence the need to make explicit the set of rigid values of any given profile. We formalize this by defining an ordering on profiles corresponding to the operation just described, so that the forest profile abstracting $\bar{\mathbf{t}}'$ is bigger than the profile abstracting $\bar{\mathbf{t}}$. We make explicit (in Lemma 5.1) the aforementioned argument as a monotonicity property of the algebra.

We say that a forest profile $\mathfrak{f} = (\bar{\chi}, R)$ is **valid** if every $d \in \mathbb{D}$ so that $\overrightarrow{\chi}^{\mathrm{E}}(\alpha, \beta, i) = \{d\}$ or $\overleftarrow{\chi}^{\mathrm{E}}(\alpha, \beta, i) = \{d\}$ for some $(\alpha, \beta, i) \in \Pi$, is in R. We define $\mathfrak{F}$ as the set of all valid profiles.

4.2 Algebra

We equip $\mathfrak{F}$ with two operations. The idea is that these operations correspond to the *concatenation* of two forests, and to the addition of a root to a forest (called *rooting*), turning it into a tree.

Preliminaries

The set of **root patterns** of a forest profile $\mathfrak{f}$, denoted by $[\mathfrak{f}\rangle, \langle[\mathfrak{f}] \subseteq \mathcal{P}$ is defined as follows

$$\begin{aligned}
[\mathfrak{f}\rangle &\stackrel{def}{=} \{\alpha \mid (d, \alpha, \beta, i) \in \overrightarrow{\chi}^{\mathrm{I}} \text{ for some } d, \beta, i\},\\
\langle[\mathfrak{f}] &\stackrel{def}{=} \{\alpha \mid (d, \alpha, \beta, i) \in \overleftarrow{\chi}^{\mathrm{I}} \text{ for some } d, \beta, i\}.
\end{aligned}$$

Given $P \subseteq \mathcal{P}$ and $\chi \subseteq \mathbb{D} \times \Pi$, we define the **extension of** χ **by** P, denoted by $P{\cdot}\chi$, as the set

$$\begin{aligned}
P{\cdot}\chi \stackrel{def}{=} \chi \cup \{(d, \alpha' \cdot \alpha, \beta, i) \in \mathbb{D} \times \Pi \mid (d, \alpha, \beta, i) \in \chi,\\
\alpha' \in P\}.
\end{aligned}$$

It is easy to see that the extension operation distributes over union (*i.e.*, $P{\cdot}(\chi \cup \chi') = P{\cdot}\chi \cup P{\cdot}\chi'$ and $(P \cup P'){\cdot}\chi = P{\cdot}\chi \cup P'{\cdot}\chi$).

Fingerprints

We now define the *fingerprint* of a forest profile. It contains a summary information, sufficient to decide whether the tree abstracted by the profile satisfies a formula of XPath—as we show in Section 6.

Let $\mathcal{A} = \{\overleftarrow{\circ}, \overrightarrow{\circ}, \circ^{\downarrow}, \circ\}$. Given a profile $\mathfrak{f} \in \mathfrak{F}$ and $a \in \mathcal{A}$, we define the set $\mathfrak{f}.\chi_a$ as

- $\overleftarrow{\chi}^{\mathrm{I}} \cup \langle[\mathfrak{f}]{\cdot}\overleftarrow{\chi}^{\mathrm{E}}$ if $a = \overleftarrow{\circ}$,
- $\overrightarrow{\chi}^{\mathrm{I}} \cup [\mathfrak{f}\rangle{\cdot}\overrightarrow{\chi}^{\mathrm{E}}$ if $a = \overrightarrow{\circ}$,
- $\{(d, \alpha', \beta, i) \in \mathbb{D} \times \Pi \mid \exists \alpha.(d, \alpha, \beta, i) \in \overleftarrow{\chi}^{\mathrm{I}} \cup \overrightarrow{\chi}^{\mathrm{I}}\}$ if $a = \circ^{\downarrow}$, or
- $\{(d, \alpha, \beta, i) \in \overrightarrow{\chi}^{\mathrm{I}} \cup \overleftarrow{\chi}^{\mathrm{I}} \mid \beta = \epsilon\}$ if $a = \circ$.

Note that $\mathfrak{f}.\chi_a(\alpha, \beta, i)$ is independent of α when $a = \circ^{\downarrow}$, but it takes an element of Π as argument for the sake of uniformity of notation. The **fingerprint** of a profile $\mathfrak{f}$, noted $\xi(\mathfrak{f})$, is an element of

$$\begin{aligned}
\mathcal{F} \;\stackrel{def}{=}\; & \Pi \times \mathcal{A} \to \{0, 1, 2{+}\} \quad \cup\\
& \Pi \times \mathcal{A} \times \Pi \times \mathcal{A} \to \{0, 1{+}\},
\end{aligned}$$

where for $\bar{\alpha}, \bar{\alpha}' \in \Pi$, $a, a' \in \mathcal{A}$, we define $\xi(\mathfrak{f})(\bar{\alpha}, a, \bar{\alpha}', a')$ as 0 or 1+ depending on whether $|\mathfrak{f}.\chi_a(\bar{\alpha}) \cap \mathfrak{f}.\chi_{a'}(\bar{\alpha}')| = 0$ or not; and we define $\xi(\mathfrak{f})(\bar{\alpha}, a)$ as 0, 1, or 2+ depending on $|\mathfrak{f}.\chi_a(\bar{\alpha})|$ being 0, 1 or greater than 1 respectively.

We fix the set of **consistent fingerprints**, as a set of fingerprints $\Gamma \subseteq \mathcal{F}$. The usefulness of this set will become apparent in the reduction from XPath to the derivation problem of forest profiles in Section 6.2, but we can anticipate

that this set will represent all the profiles abstracting multi-attribute data trees that do not contradict the formula we are trying to satisfy. For the moment, however, the reader may simply consider Γ as a given arbitrary set of fingerprints.

Concatenation

For every two $\mathfrak{f}_1, \mathfrak{f}_2 \in \mathfrak{F}$ so that

(a) $R_1 = R_2$,

(b) $\overrightarrow{\chi}_1^{\mathrm{E}} = \overrightarrow{\chi}_2^{\mathrm{I}} \cup [\mathfrak{f}_2\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}$, and

(c) $\overleftarrow{\chi}_2^{\mathrm{E}} = \overleftarrow{\chi}_1^{\mathrm{I}} \cup \langle\mathfrak{f}_1] \cdot \overleftarrow{\chi}_1^{\mathrm{E}}$;

we define the **concatenation** of $\mathfrak{f}_1$ and $\mathfrak{f}_2$, denoted as $\mathfrak{f}_1 + \mathfrak{f}_2$ as $\mathfrak{f}_3$, where

$$R_3 = R_1 = R_2 \quad (+1)$$
$$\overrightarrow{\chi}_3^{\mathrm{E}} = \overrightarrow{\chi}_2^{\mathrm{E}} \quad (+2)$$
$$\overleftarrow{\chi}_3^{\mathrm{E}} = \overleftarrow{\chi}_1^{\mathrm{E}} \quad (+3)$$
$$\overrightarrow{\chi}_3^{\mathrm{I}} = \overrightarrow{\chi}_1^{\mathrm{I}} \cup [\mathfrak{f}_1\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{I}} \quad (+4)$$
$$\overleftarrow{\chi}_3^{\mathrm{I}} = \overrightarrow{\chi}_2^{\mathrm{I}} \cup \langle\mathfrak{f}_2] \cdot \overleftarrow{\chi}_1^{\mathrm{I}}. \quad (+5)$$

Notice that

- the concatenation is associative ($(\mathfrak{f}_1 + \mathfrak{f}_2) + \mathfrak{f}_3 = \mathfrak{f}_1 + (\mathfrak{f}_2 + \mathfrak{f}_3)$),
- the extension operation $\cdot$ distributes over the concatenation operation $+$ ($[\mathfrak{f}_1 + \mathfrak{f}_2\rangle \cdot \chi = [\mathfrak{f}_1\rangle \cdot ([\mathfrak{f}_2\rangle \cdot \chi)$),
- if $\mathfrak{f}_1 + \mathfrak{f}_2 = \mathfrak{f}_3$ and $\mathfrak{f}_1, \mathfrak{f}_2 \in \mathfrak{F}$, then $\mathfrak{f}_3 \in \mathfrak{F}$.

Rooting

Given $a \in \mathbb{A}$, and $\bar{d} \in \mathbb{D}^{\mathbf{k}}$, we define $(a, \bar{d})\mathfrak{f}_1 \subseteq \mathfrak{F}$, where $\mathfrak{f}_2 \in (a, \bar{d})\mathfrak{f}_1$ if

(a) $\xi(\mathfrak{f}_2) \in \Gamma$,

(b) $\overrightarrow{\chi}_1^{\mathrm{E}} = \overleftarrow{\chi}_1^{\mathrm{E}} = \emptyset$,

(c) $\overrightarrow{\chi}_2^{\mathrm{I}} = \overleftarrow{\chi}_2^{\mathrm{I}} = \{(d, \alpha, \beta\cdot\gamma, i) \in \mathbb{D} \times \Pi \mid \exists \alpha'.(d, \alpha', \gamma, i) \in \overrightarrow{\chi}_1^{\mathrm{I}} \cup \overleftarrow{\chi}_1^{\mathrm{I}}, \alpha, \beta \in \boldsymbol{\sigma}_a\} \cup \bigcup_{i \in [\mathbf{k}]} (\{\bar{d}(i)\} \times (\boldsymbol{\sigma}_a \setminus \{\epsilon\}) \times \boldsymbol{\sigma}_a \times \{i\})$

We say that $\mathfrak{f}_2$ is a **rooting of** $\mathfrak{f}_1$ **with** $(a, \bar{d})$.

Notice that since the root pattern of any pair of profiles $\mathfrak{f}_1, \mathfrak{f}_2 \in (a, \bar{d})\mathfrak{f}_3$ is the same, it is idempotent and absorbing ($[\mathfrak{f}_1\rangle \cdot [\mathfrak{f}_2\rangle \cdot \chi = [\mathfrak{f}_1\rangle \cdot \chi = [\mathfrak{f}_2\rangle \cdot \chi$, $[\mathfrak{f}_1\rangle \cdot \overrightarrow{\chi}_1^{\mathrm{I}} = \overrightarrow{\chi}_1^{\mathrm{I}}$).

4.3 The derivation problem

We define the **empty profile** as $\mathfrak{f}_\emptyset \stackrel{def}{=} (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$. Note that $\mathfrak{f}_\emptyset \in \mathfrak{F}$. The set of profiles that can be obtained from empty profiles by applying the rooting and concatenation operations is called the set of **derivable profiles**, and noted $\mathfrak{D}$. We say that $\mathfrak{f}$ is a **derivable root profile** if $\overrightarrow{\chi}^{\mathrm{E}} = \overleftarrow{\chi}^{\mathrm{E}} = \emptyset$ and $\mathfrak{f} \in (a, \bar{d})\mathfrak{f}'$ for some $\mathfrak{f}' \in \mathfrak{D}$, $a \in \mathbb{A}_{root}$ and $\bar{d} \in \mathbb{D}^{\mathbf{k}}$. Let a **derivation tree** for $\mathfrak{f}$ be a tree t whose every node is labeled by a forest profile and an element from $\mathbb{A} \times \mathbb{D}^{\mathbf{k}}$, except the leaves that are labeled only by the forest profile $\mathfrak{f}_\emptyset$ and

- the root is labeled with $\mathfrak{f}$,
- every internal node x of t labeled with a forest profile $\mathfrak{f}'$ and $(a, \bar{d})$ is so that $\mathfrak{f}' \in (a, \bar{d})(\mathfrak{f}_1 + \cdots + \mathfrak{f}_n)$, where $\mathfrak{f}_1, \ldots, \mathfrak{f}_n$ are the labels of the children of x.

Similarly, a **derivation forest** $\bar{t}$ for $\mathfrak{f}$ is a forest of derivation trees $\bar{t} = t_1 \cdots t_n$ for some profiles $\mathfrak{f}_1, \ldots, \mathfrak{f}_n$ so that $\mathfrak{f} = \mathfrak{f}_1 + \cdots + \mathfrak{f}_n$. Therefore, a profile $\mathfrak{f}$ is derivable if, and only if, there is a derivation forest for $\mathfrak{f}$.

We can now state the *derivation problem*, that is, whether there exists a derivable root profile, given $\mathbb{A}$, $\mathbb{A}_{root}$, $\mathcal{P}$ and Γ.

PROBLEM:	**The derivation problem**
INPUT:	A finite alphabet $\mathbb{A}$, $\mathbb{A}_{root} \subseteq \mathbb{A}$, a set of patterns $\mathcal{P}$, a set of fingerprints $\Gamma \subseteq \mathcal{F}$.
QUESTION:	Is there a derivable root profile?

In the next section we show that this problem is decidable. Later, in Section 6, we show that this problem is reducible from SAT-XPath($^*\!\leftarrow, \downarrow_*, \rightarrow^*, =$).

5. COMPUTING DERIVABLE PROFILES

In this section we solve the derivation problem, showing that it is decidable in 2EXPSPACE. To show this problem we work with some partial ordering on forest profiles (Section 5.2) that has some good monotonicity closure properties with our forest profile algebra (Section 5.3). This allows us to reduce the problem to a restricted derivation problem in which solutions can be found by only inspecting profiles with a bounded number of rigid values (Section 5.4), that are minimal elements of the ordering (Section 5.5). These are bounded and computable, allowing us to produce an algorithm solving the problem (Section 5.6).

5.1 Preliminaries

Given $\mathfrak{f}_1, \mathfrak{f}_2 \in \mathfrak{F}$ we define that $\mathfrak{f}_1$ and $\mathfrak{f}_2$ are **equivalent**, and we note it $\mathfrak{f}_1 \sim \mathfrak{f}_2$, if there is some bijection $g : \mathbb{D} \to \mathbb{D}$ so that $\mathfrak{f}_2$ is the result of replacing d by $g(d)$ in $\mathfrak{f}_1$; in this case we write $g(\mathfrak{f}_1) = \mathfrak{f}_2$. For a set $C \subseteq \mathfrak{F}$, we write $\mathfrak{f} \lessapprox C$ if there is $\mathfrak{f}' \sim \mathfrak{f}$ so that $\mathfrak{f}' \in C$. Given a forest profile $\mathfrak{f}$ and two data values $d \in data(\mathfrak{f})$, $d' \notin data(\mathfrak{f})$, we define $\mathfrak{f}[d \mapsto d']$ as the result of replacing d by d' in $\mathfrak{f}$. Note that $\mathfrak{f}[d \mapsto d'] \sim \mathfrak{f}$. Given two data values d, d' we write $\mathfrak{f}[d \mapsto d, d']$ to denote $\mathfrak{f}'$ where $R' = R$, $\bar{\chi}'(d') = \bar{\chi}(d)$ and $\bar{\chi}'(e) = \bar{\chi}(e)$ for every other $e \neq d'$. Note that if $d \in data(\mathfrak{f}) \setminus R$ and $d' \notin data(\mathfrak{f})$, we have that if $\mathfrak{f} \in \mathfrak{F}$ then $\mathfrak{f}[d \mapsto d, d'] \in \mathfrak{F}$.

We say that a data value $d \in \mathbb{D}$ is an **external data value** of $\mathfrak{f}$ if $\overleftarrow{\chi}^{\mathrm{E}}(d) \cup \overrightarrow{\chi}^{\mathrm{E}}(d) \neq \emptyset$. If further $\overrightarrow{\chi}^{\mathrm{I}}(d) \cap \overleftarrow{\chi}^{\mathrm{I}}(d) = \emptyset$, we say that d is a **strict external data value** of $\mathfrak{f}$. If $d \in data(\mathfrak{f})$ is not a strict external data value, it is then an **internal data value**, and if it is not en external data value, it is then a **strict internal data value**.

5.2 Ordering on profiles

We define a partial order $\preceq$ on forest profiles, that follows from our discussion of Section 4 on the role of flexible and rigid data values. It is the order in which we can make a profile bigger by adding a fresh data value to it, with the same description as that of a flexible data value already contained in it.

Given $\mathfrak{f}_1, \mathfrak{f}_2 \in \mathfrak{F}$, we define $\mathfrak{f}_1 \preceq \mathfrak{f}_2$ if either $\mathfrak{f}_1 = \mathfrak{f}_2$, or there is a flexible datum d of $\mathfrak{f}_1$ so that $\mathfrak{f}_1[d \mapsto d, d'] \preceq \mathfrak{f}_2$ for some $d' \notin data(\mathfrak{f}_1)$. Note that $\preceq$ is recursive, reflexive and transitive, and it is hence a partial order.

Note that if $\mathfrak{f}_1 \preceq \mathfrak{f}_2$ then $\langle\mathfrak{f}_1] = \langle\mathfrak{f}_2]$ and $[\mathfrak{f}_1\rangle = [\mathfrak{f}_2\rangle$. Note also that if $\mathfrak{f} \preceq \mathfrak{f}'$ then $\xi(\mathfrak{f}) = \xi(\mathfrak{f}')$.

We write $\mathfrak{f} \precapprox \mathfrak{f}'$ if $\mathfrak{f} \preceq \mathfrak{f}''$ for some $\mathfrak{f}'' \sim \mathfrak{f}'$. We say that a set of forest profiles $G \subseteq \mathfrak{F}$ is **upward closed** (resp. **downward**

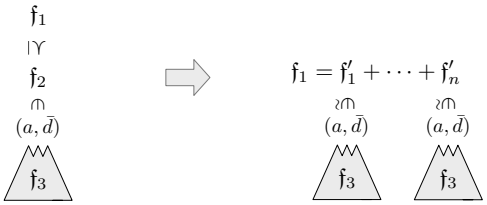


Figure 4: Statement of Lemma 5.1.

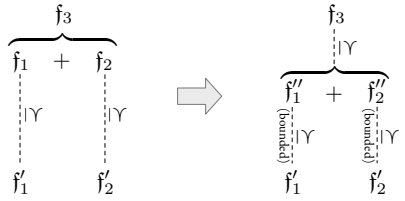


Figure 5: Statement of Lemma 5.3.

closed) with respect to $\precsim$, if for every $\mathfrak{f} \in G$ and $\mathfrak{f}' \succsim \mathfrak{f}$ (resp. $\mathfrak{f} \succsim \mathfrak{f}'$), we have $\mathfrak{f}' \in G$. We write

$$\uparrow G \stackrel{def}{=} \{\mathfrak{f} \in \mathfrak{F} \mid \mathfrak{f} \succsim \mathfrak{f}' \text{ for some } \mathfrak{f}' \in G\}$$

$$\downarrow G \stackrel{def}{=} \{\mathfrak{f} \in \mathfrak{F} \mid \mathfrak{f}' \succsim \mathfrak{f} \text{ for some } \mathfrak{f}' \in G\}$$

for the upward and downward closure of G with respect to $\precsim$. We say that G is $\uparrow\downarrow$-closed, if it is both upward and downward closed, that is, $G = \uparrow\downarrow G$.

5.3 Monotonicity properties

In order to devise an algorithm that tests the existence of a derivable root profile, we will need some monotonicity lemmas evidencing the relationship between $\preceq$ and the rooting and concatenation operations on profiles. The ultimate goal of these lemmas is to restrict the derivation problem to profiles that are *minimal* with respect to $\precsim$.

The next Lemma 5.1 states that for any two profiles $\mathfrak{f}_1 \succeq \mathfrak{f}_2$, $\mathfrak{f}_1$ can be seen as a concatenation of profiles that share the same descriptions of internal values as $\mathfrak{f}_1$, under certain restrictions, as it is shown next. This is a crucial property that follows from our discussion in Section 4.1.

LEMMA 5.1 (FIGURE 4). *For every $\mathfrak{f}_1 \succeq \mathfrak{f}_2 \in (a,\bar{d})\mathfrak{f}_3$, there is $n \in \mathbb{N}$, and $\mathfrak{f}'_i \Subset (a,\bar{d})\mathfrak{f}_3$ for every $i \in [n]$ so that*

$$\mathfrak{f}_1 = \mathfrak{f}'_1 + \cdots + \mathfrak{f}'_n.$$

The lemma above implies that the set of derivable profiles is upward closed. (Details in the Appendix.)

LEMMA 5.2. $\mathfrak{D} = \uparrow\mathfrak{D}$.

We finally state two other monotonicity properties that will be required to reduce the derivation problem into a similar problem that works only with minimal profiles in Section 5.5.

We say that a profile $\mathfrak{f}'$ is a **bounded extension** of a profile $\mathfrak{f}$ if $\mathfrak{f} \preceq \mathfrak{f}'$ and $|data(\mathfrak{f}')| \leq |data(\mathfrak{f})| + 3|\Pi|^4$. The following lemma tells us that for any $G \subseteq \mathfrak{F}$ and any profiles $\mathfrak{f}_1, \mathfrak{f}_2 \in \uparrow G$, there are bounded extensions $\mathfrak{f}''_1, \mathfrak{f}''_2$ of profiles of G so that $\mathfrak{f}''_1 + \mathfrak{f}''_2 \precsim \mathfrak{f}_1 + \mathfrak{f}_2$, as in Figure 5.

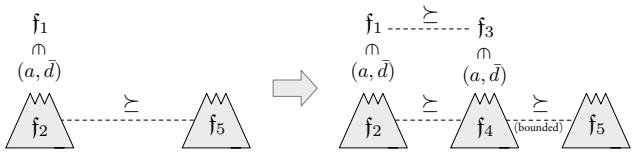


Figure 6: Statement of Lemma 5.4.

LEMMA 5.3 (FIGURE 5). *If $\mathfrak{f}_1 + \mathfrak{f}_2 = \mathfrak{f}_3$ and $\mathfrak{f}'_1 \preceq \mathfrak{f}_1$, $\mathfrak{f}'_2 \preceq \mathfrak{f}_2$, then $\mathfrak{f}''_1 + \mathfrak{f}''_2 \preceq \mathfrak{f}_3$, for some $\mathfrak{f}''_1, \mathfrak{f}''_2 \in \mathfrak{F}$ so that $\mathfrak{f}''_i$ is a bounded extension of $\mathfrak{f}'_i$, for all $i \in \{1,2\}$.*

A similar lemma holds for the rooting operation.

LEMMA 5.4 (FIGURE 6). *For every $\mathfrak{f}_1 \in (a,\bar{d})\mathfrak{f}_2$ and $\mathfrak{f}_2 \succeq \mathfrak{f}_5$, there is $\mathfrak{f}_4 \succeq \mathfrak{f}_5$ and $\mathfrak{f}_3 \Subset (a,\bar{d})\mathfrak{f}_4$ so that $|data(\mathfrak{f}_4)| \leq |data(\mathfrak{f}_5)| + |\Pi|^4 + |R_1|$ and $\mathfrak{f}_3 \preceq \mathfrak{f}_1$, $\mathfrak{f}_4 \preceq \mathfrak{f}_2$.*

5.4 Bounding the rigid values

In this section we show that we can reduce the derivation problem into a similar problem where all the profiles have boundedly many rigid values. This will be combined with the result of the next sections, stating that the derivation problem restricted to profiles with boundedly many rigid values is decidable in 2EXPSPACE, to solve the derivation problem.

LEMMA 5.5. *If there is a derivable root profile, then there is a derivation tree for a root profile so that all the profiles in the forest have no more than $2|\Pi|$ rigid values.*

Let $\mathfrak{F}_b$ be the set of all $\mathfrak{f} \in \mathfrak{F}$ that have no more than $2|\Pi|$ rigid values. Let $\mathfrak{D}_b$ be the set of derivable profiles restricted to $\mathfrak{F}_b$.

REMARK 5.6. *By Lemma 5.5 and in light of the definition of bounded extension, it follows that Lemma 5.4, when applied to profiles of $\mathfrak{F}_b$, yields a profile $\mathfrak{f}_4$ that is a* bounded extension *of $\mathfrak{f}_5$.*

By the Lemma just shown, we have the following

LEMMA 5.7. *There is a derivable root profile in $\mathfrak{D}$ if and only if there is a derivable root profile in $\mathfrak{D}_b$.*

We have then reduced the derivation problem into a simpler problem, the **bounded derivation problem**: testing whether there is a derivable root profile in $\mathfrak{D}_b$.

REMARK 5.8. *We have that $\mathfrak{D}_b$ is upward closed since $\mathfrak{D}$ is upward closed. That is, $\mathfrak{D}_b = \uparrow\mathfrak{D}_b$.*

Note that $\mathfrak{F}_b$ has boundedly many $\precsim$-minimal elements. In the next section we show how to restrict the problem to a problem that uses only these $\precsim$-minimal profiles. We will show how this yields a 2EXPSPACE algorithm in Section 5.6.

5.5 Restricting to minimal elements

Thanks to the result from the previous section stating that $\mathfrak{D}_b$ is upward closed, we can now show that we can work only with the minimal elements of $\mathfrak{F}_b$. The main necessary property concerns all those profiles $\mathfrak{f}' \in \mathfrak{F}_b$ that are '$\precsim$-related' to a profile $\mathfrak{f}'' \in \mathfrak{D}_b$, in the sense that $\mathfrak{f}' \succsim \mathfrak{f} \precsim \mathfrak{f}'' \in \mathfrak{D}_b$ for

some $\mathfrak{f}$. (Note that this set of profiles is precisely $\uparrow\downarrow\mathfrak{D}_b$.) The property states that the forest profiles algebra preserves the $\precsim$-relatedness.

Given $G \subseteq \mathfrak{F}_b$, let

$$\boldsymbol{R}_{up}^{(a,\bar{d})}(G) \stackrel{def}{=} \{\mathfrak{f} \in \mathfrak{F}_b \mid \mathfrak{f} \in (a,\bar{d})\mathfrak{f}', \mathfrak{f}' \in G\} \quad \text{for } (a,\bar{d}) \in \mathbb{A} \times \mathbb{D}^{\mathbf{k}},$$

$$\boldsymbol{R}_{up}(G) \stackrel{def}{=} \bigcup_{(a,\bar{d})\in\mathbb{A}\times\mathbb{D}^{\mathbf{k}}} \boldsymbol{R}_{up}^{(a,\bar{d})}(G),$$

$$\boldsymbol{R}_{+}(G) \stackrel{def}{=} \{\mathfrak{f} \in \mathfrak{F}_b \mid \mathfrak{f} = \mathfrak{f}_1 + \mathfrak{f}_2 \text{ where } \mathfrak{f}_1, \mathfrak{f}_2 \in G\},$$

$$\boldsymbol{R}(G) \stackrel{def}{=} \boldsymbol{R}_{up}(G) \cup \boldsymbol{R}_{+}(G).$$

LEMMA 5.9. $\boldsymbol{R}(\uparrow\downarrow\mathfrak{D}_b) \subseteq \downarrow\mathfrak{D}_b$.

5.6 The algorithm

In this section we show how to compute, in 2EXPSPACE, whether there exists a derivable root profile in $\mathfrak{D}_b$, solving thus the derivation problem.

For $G \subseteq \mathfrak{F}_b$, we define $G^{\sim} \stackrel{def}{=} \{\mathfrak{f} \mid \mathfrak{f} \sim \mathfrak{f}' \text{ for some } \mathfrak{f}' \in G\}$. We define $G/\!\sim$ as the set containing one representative profile of G for each $\sim$-equivalence class. We define $\text{MIN}(G)$ as the set of $\precsim$-minimal elements of G,

$$\text{MIN}(G) \stackrel{def}{=} \{\mathfrak{f} \in G \mid \text{for all } \mathfrak{f}' \in G \text{ so that } \mathfrak{f}' \precsim \mathfrak{f} \text{ we have } \mathfrak{f} \sim \mathfrak{f}'\}.$$

For any $\mathfrak{f} \in \mathfrak{F}$, we write $|\mathfrak{f}|$—the **size** of $\mathfrak{f}$—, as the size needed to write $\mathfrak{f}$. Note that for all $\mathfrak{f} \in \text{MIN}(\mathfrak{F}_b)$, $|\mathfrak{f}|$ is at most exponential in $|\mathcal{P}|$. For any $G \subseteq \mathfrak{F}$, we write $|G|$ to denote $\sum_{\mathfrak{f}\in G} |\mathfrak{f}|$.

Let us define $\boldsymbol{C}_i$ for every $i \in \mathbb{N}_0$ as

$$\boldsymbol{C}_0 \stackrel{def}{=} \{\mathfrak{f}_\emptyset\},$$
$$\boldsymbol{C}_{i+1} \stackrel{def}{=} \boldsymbol{C}_i \cup \text{MIN}\big(\downarrow\boldsymbol{R}(\uparrow\downarrow\boldsymbol{C}_i)\big)/\!\sim.$$

Let $k_0 \in \mathbb{N}_0$ be the first index so that $\boldsymbol{C}_{k_0}^{\sim} = \boldsymbol{C}_{k_0+1}^{\sim}$.

REMARK 5.10. *For every $i \in \mathbb{N}_0$, $\boldsymbol{C}_i \subseteq \text{MIN}(\mathfrak{F}_b)$.*

As a consequence of the property of the preceding section, we have that this algorithm computes $\text{MIN}(\downarrow\mathfrak{D}_b)$.

LEMMA 5.11. $\boldsymbol{C}_{k_0}^{\sim} = \text{MIN}(\downarrow\mathfrak{D}_b)$.

We further have that this computation is in 2EXPSPACE since $|\text{MIN}(\mathfrak{F}_b)/\!\sim|$ is doubly exponential in $|\mathcal{P}|$, hence we obtain the following. (Details in the Appendix.)

PROPOSITION 5.12. *The derivation problem is decidable in 2*EXPSPACE.

6. FROM XPATH TO FOREST PROFILES

In this section we reduce the satisfiability problem for XPath($^*\!\leftarrow, \downarrow_*, \rightarrow^*, =$) into the derivation problem for forest profiles.

In Section 6.1 we define a normal form for XPath($^*\!\leftarrow, \downarrow_*, \rightarrow^*, =$), called *direct unnested normal form*, and in Section 6.2 we show the reduction from the satisfiability problem of direct unnested XPath($^*\!\leftarrow, \downarrow_*, \rightarrow^*, =$) formulas into the derivation problem for forest profiles.

6.1 Normal forms

We will assume a certain normal form of the formula $\varphi \in \text{XPath}(^*\!\leftarrow, \downarrow_*, \rightarrow^*, =)$ to test for satisfiability. This will simplify the reduction into the derivation problem for forest profiles.

The normal form has two main properties. Firstly, it contains only path expressions that are *direct*, in the sense that the navigation consists in going left and then down, or going right and then down. And secondly, path expressions do not contain data tests as node expressions, in other words the formula is *unnested*. Next, we explain in detail these properties.

Preliminaries

Let $\alpha = a_1 \cdots a_n$ with $n > 0$ be a XPath($^*\!\leftarrow, \downarrow_*, \rightarrow^*, =$) path expression, where for every i, $a_i = [\psi]$ for some node expression ψ, or $a_i \in \{\varepsilon, {}^*\!\leftarrow, \downarrow_*, \rightarrow^*\}$. We say that α is in **alternating path normal form** if either $\alpha = \varepsilon$, or n is even and for all $1 \le i \le n$

- if i is even, $a_i = [\psi]$ for some node expression ψ,
- if i is odd, $a_i \in \{{}^*\!\leftarrow, \downarrow_*, \rightarrow^*\}$.

In other words, the path alternates between axes and tests for node expressions. We say that a formula is in alternating path normal form if all its path expressions are in alternating path normal form. Note that one can turn any formula $\varphi \in \text{XPath}(^*\!\leftarrow, \downarrow_*, \rightarrow^*, =)$ into an equivalent formula φ' in alternating path normal form in polynomial time, using the equivalences

$$\begin{array}{l} \langle[\psi]\alpha@_i \odot \beta@_j\rangle \equiv \psi \wedge \langle\alpha@_i \odot \beta@_j\rangle \text{ for } \odot \in \{=, \neq\}, \\ \langle\alpha@_i \odot [\psi]\beta@_j\rangle \equiv \psi \wedge \langle\alpha@_i \odot \beta@_j\rangle \text{ for } \odot \in \{=, \neq\}, \\ \qquad \alpha[\psi_1][\psi_2]\beta \equiv \alpha[\psi_1 \wedge \psi_2]\beta, \text{ and,} \\ \text{if } \alpha\beta \neq \epsilon,\ \ \alpha\beta \equiv \alpha[\top]\beta \ \text{ and } \ \alpha\varepsilon\beta \equiv \alpha\beta. \end{array} \qquad (\triangle)$$

For simplicity and without any loss of generality we can further assume that all our formulas do not contain formulas of the type $\langle\alpha\rangle$, since it is equivalent to $\langle\alpha@_1 = \alpha@_1\rangle$. We will henceforth assume that all the formulas we work with are in this form.

We say that a path expression in alternating path normal form is a **rightward path expression**, if it starts with $\rightarrow^*$ and all the axes in it are $\rightarrow^*$ (similarly with **leftward**, **downward** and $^*\!\leftarrow$, $\downarrow_*$). Notice that, for example, a leftward expression may contain node tests using rightward or downward axes. For example, $^*\!\leftarrow[\langle\downarrow_*[a]\rangle]^*\!\leftarrow[b]$ is a leftward expression while $^*\!\leftarrow[a]\downarrow_*[\langle^*\!\leftarrow[a]\rangle]$ is not.

Direct normal form

The object of the direct normal form is to avoid having unnecessary mixed directions in path formulas, that use perhaps $\rightarrow^*$ and $^*\!\leftarrow$ in the same expression, or that contain a $^*\!\leftarrow$ (or $\rightarrow^*$) axis after a $\downarrow_*$ axis. That is, we avoid having formulas like

$$\langle\ \rightarrow^*[a]{}^*\!\leftarrow@_1 = \downarrow_*[b]\rightarrow^*@_2\ \rangle$$

in favor of equivalent formulas with a more *direct* navigation, like

$$\begin{array}{l} \langle\ \rightarrow^*[\langle\rightarrow^*[a]\rangle]@_1 = \downarrow_*[\langle^*\!\leftarrow[b]\rangle]@_2\ \rangle\ \vee \\ \langle\ [\langle\rightarrow^*[a]\rangle]^*\!\leftarrow@_1 = \downarrow_*[\langle^*\!\leftarrow[b]\rangle]@_2\ \rangle. \end{array} \qquad (\ddagger)$$

In the formula above we factor the loops that may be in the navigation of the path expression to obtain a simple navigation that goes in only one horizontal direction.

We say that a formula $\varphi \in \mathrm{XPath}({}^{*}\!\leftarrow, \downarrow_*, \rightarrow^*, =)$ is in **direct normal form**, if every path expression is ε, or of the form $\alpha{\cdot}\beta$, where $\alpha{\cdot}\beta \neq \epsilon$ (*i.e.*, it is not the empty string), α is leftward, rightward or empty, and β is downward or empty. Note that, strictly speaking, the formula (‡) is not in direct normal form since its second disjunct is not in alternating path normal form, but the equivalent alternating path expression—using (△)—is in direct normal form.

LEMMA 6.1 (DIRECT NORMAL FORM). *There exists an exponential time translation that for every node expression $\varphi \in \mathrm{XPath}({}^{*}\!\leftarrow, \downarrow_*, \rightarrow^*, =)$ returns an equivalent node expression ψ in direct normal form.*

Unnested normal form

The second normal form consists in having formulas without nesting of data tests. That is, we avoid treating formulas like, for example

$$\langle\ \downarrow_*[\underbrace{\langle\ {}^{*}\!\leftarrow[a]@_1 = \rightarrow^*[b]@_1\ \rangle}_{\text{nested data test}}]@_1\ =\ \rightarrow^*[c]@_2\ \rangle\ .$$

If a formula is such that all its path expressions α contain only (boolean combinations of) tests for labels we call it a **non-recursive** formula.

We say that φ is in unnested normal form if $\varphi = \varphi_1 \wedge \varphi_2$ where $\varphi_1 \in \mathcal{B}(\mathbb{A})$ and φ_2 is a conjunction of tests of the form "if a node has some of the labels $\{a_1, \ldots, a_n\}$ then it satisfies ψ" for some non-recursive formula ψ and labels $a_1, \ldots, a_n \in \mathbb{A}$. Formally, φ_2 contains a conjunction of tests of the form

$$\neg\langle\ \downarrow_*[\tau \wedge \neg\psi]\ \rangle$$

for τ a disjunction of labels and ψ a non-recursive formula. Given $\varphi = \varphi_1 \wedge \varphi_2$ in unnested normal form, we write $\gamma_\varphi(a)$ for $a \in \mathbb{A}$ to denote the function where $\gamma_\varphi(a)$ is the conjunction of all the formulas ψ such that φ_2 contains $\neg\langle\rightarrow^*[\tau\wedge\neg\psi]\rangle$ as a subformula, for some disjunctive formula τ containing the label a.

Then, we obtain the following.

LEMMA 6.2 (UNNESTED NORMAL FORM). *There exists an exponential time translation that for every formula $\eta \in \mathrm{XPath}({}^{*}\!\leftarrow, \downarrow_*, \rightarrow^*, =)$ returns a formula φ in unnested normal form such that η is satisfiable iff φ is satisfiable. Further, the translation of a formula in direct normal form is in direct normal form.*

COROLLARY 6.3. *About the translation of Lemma 6.2:*

1. *The set of path subformulas resulting from the translation has cardinality polynomial in η.*
2. *Every path subformula resulting from the translation can be written using polynomial space.*

6.2 Reduction to the derivation problem

In this section we show how we can reduce the satisfiability problem of direct unnested $\mathrm{XPath}({}^{*}\!\leftarrow, \downarrow_*, \rightarrow^*, =)$ formulas into the derivation problem for forest profiles.

Let us fix $\phi = \phi_1 \wedge \phi_2$ in direct unnested normal form, where $\mathbb{A}$ as the finite alphabet, $\mathbf{k}$ as the number of attributes, $\mathbb{D}$ as any infinite domain, and $\mathbb{A}_{root}$ is the set of all $a \in \mathbb{A}$ that make ϕ_1 true.

Given a pattern $\alpha = \psi_1 \cdots \psi_k \in \mathcal{P}$, and an axis $o \in \{{}^{*}\!\leftarrow, \downarrow_*, \rightarrow^*\}$, we can convert α into a path expression as follows:

$$\mathrm{P}_o(\epsilon) \stackrel{def}{=} \varepsilon \qquad \text{if } k = 0,$$
$$\mathrm{P}_o(\psi_1 \cdots \psi_k) \stackrel{def}{=} o[\psi_1]\, o \cdots o[\psi_k] \qquad \text{if } k > 0.$$

Note that P_o is injective.

Let us define $\mathcal{P}_\phi$ as the set of patterns consisting of

- the constant $\top$ and the empty string ϵ,
- ψ, for every $\psi \in \mathcal{B}(\mathbb{A})$ that is a subformula of ϕ,
- every $\alpha \in (\mathcal{B}(\mathbb{A}))^*$ so that $\mathrm{P}_{\rightarrow^*}(\alpha)$, $\mathrm{P}_{{}^{*}\!\leftarrow}(\alpha)$, or $\mathrm{P}_{\downarrow_*}(\alpha)$ is a substring of ϕ.

It follows that $\mathcal{P}_\phi$ is finite and subword-closed.

For any direct non-recursive formula ψ that is a boolean combination of subformulas of ϕ and forest profile $\mathfrak{f}$, we define $\mathfrak{f} \vdash \psi$ as follows. If $\psi \in \mathbb{A}$, then $\mathfrak{f} \vdash \psi$ if and only if there is some $d \in \mathbb{D}$ and $i \in [\mathbf{k}]$ so that $(\psi, \epsilon, i) \in \overrightarrow{\chi}^{\mathrm{I}}(d)$. For all the boolean cases $\vdash$ is homomorphic. Suppose now that $\psi = \langle\alpha{\cdot}\beta@_i \neq \gamma{\cdot}\delta@_j\rangle$ where α is leftward, ε or empty, γ is rightward, ε or empty, and β, δ are downward or empty. We define $\mathfrak{f} \vdash \psi$ if there are some $d, d' \in \mathbb{D}$ so that $d \neq d'$ and

- if $\alpha = \epsilon$ or $\alpha = \varepsilon$, $(\top, \mathrm{P}^{-1}_{\downarrow_*}(\beta), i) \in \overleftarrow{\chi}^{\mathrm{I}}(d)$,
- if $\alpha \neq \epsilon$, $\alpha \neq \varepsilon$, $(\mathrm{P}^{-1}_{{}^{*}\!\leftarrow}(\alpha), \mathrm{P}^{-1}_{\downarrow_*}(\beta), i) \in (\langle\mathfrak{f}\rangle{\cdot}\overleftarrow{\chi}^{\mathrm{E}} \cup \overleftarrow{\chi}^{\mathrm{I}})(d)$,
- if $\gamma = \epsilon$ or $\gamma = \varepsilon$, $(\top, \mathrm{P}^{-1}_{\downarrow_*}(\delta), j) \in \overrightarrow{\chi}^{\mathrm{I}}(d')$,
- if $\gamma \neq \epsilon$, $\gamma \neq \varepsilon$, $(\mathrm{P}^{-1}_{\rightarrow^*}(\gamma), \mathrm{P}^{-1}_{\downarrow_*}(\delta), j) \in (\langle\mathfrak{f}\rangle{\cdot}\overrightarrow{\chi}^{\mathrm{E}} \cup \overrightarrow{\chi}^{\mathrm{I}})(d')$.

Note that if $\alpha = \varepsilon$ then $\beta = \epsilon$ (resp. with γ and δ). If both α and γ are rightwards or leftwards it is defined in an analogous way. The case for $=$ is also analogous, where $d = d'$. The idea is that $\mathfrak{f} \vdash \psi$ makes only sense when the derivation forest for $\mathfrak{f}$ is a tree, and the multi-attribute data tree $\mathbf{t}$ associated to the derivation tree is so that $\mathbf{t} \models \psi$.

For example, testing ψ is the same as testing if there is some pattern $(\psi, \_, \_)$ in $\overleftarrow{\chi}^{\mathrm{I}}$ or $\overrightarrow{\chi}^{\mathrm{I}}$. In a similar way, checking a formula like

$$\langle\rightarrow^*[a]\downarrow_*[b]@_1 = \downarrow_*[c]@_2\rangle$$

reduces to checking if there is a data value $d \in \mathbb{D}$ that can be reached with $(\top, c, 2)$ in the main forest (*i.e.*, in $\overleftarrow{\chi}^{\mathrm{I}}$ or $\overrightarrow{\chi}^{\mathrm{I}}$), and either

- d can be reached by $(a, b, 1)$ in the main forest, that is, $(a, b, 1) \in \overrightarrow{\chi}^{\mathrm{I}}$ (or equivalently $\overleftarrow{\chi}^{\mathrm{I}}$), or
- d can be reached in the right forest by $(a, b, 1)$, where a could be tested in the main forest (*i.e.*, $a \in [\mathfrak{f}\rangle$), that is, $(a, b, 1) \in [\mathfrak{f}\rangle{\cdot}\overrightarrow{\chi}^{\mathrm{E}}$.

Note that checking $\mathfrak{f} \vdash \psi$ takes polynomial time in the size of $\mathfrak{f}$ and ψ. Also, whether $\mathfrak{f} \vdash \bigwedge_{a\in\mathbb{A}}(a \Rightarrow \gamma_\varphi(a))$ holds or not depends only on $\xi(\mathfrak{f})$.

LEMMA 6.4. *Given a direct non-recursive formula ψ that is a boolean combination of subformulas of ϕ, and two forest profiles $\mathfrak{f}, \mathfrak{f}' \in \mathfrak{F}$ so that $\xi(\mathfrak{f}) = \xi(\mathfrak{f}')$ then $\mathfrak{f} \vdash \psi$ if, and only if, $\mathfrak{f}' \vdash \psi$.*

We can then write $\xi \models \psi$ for $\xi \in \mathcal{F}$ instead of $\mathfrak{f} \models \psi$ for any $\mathfrak{f}$ so that $\xi(\mathfrak{f}) = \xi$. We define the set of consistent profiles Γ_ϕ as all $\xi \in \mathcal{F}$ so that $\xi \vdash \bigwedge_{a\in\mathbb{A}}(a \Rightarrow \gamma_\varphi(a))$. The following lemma follows straight from the above definition of $\vdash$.

LEMMA 6.5. $\mathfrak{f} \vdash \bigwedge_{a\in\mathbb{A}}(a \Rightarrow \gamma_\varphi(a))$ *iff* $\xi(\mathfrak{f}) \in \Gamma_\phi$.

Abstractions.

Given multi-attribute data forests $\bar{\mathbf{t}}_l, \bar{\mathbf{t}}, \bar{\mathbf{t}}_r$, we define

$$abs(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}, \bar{\mathbf{t}}_r)$$

as the forest profile that abstracts the forest $\bar{\mathbf{t}}$ in the context of the forests $\bar{\mathbf{t}}_l$ to the left and $\bar{\mathbf{t}}_r$ to the right. We have already discussed the idea of this abstraction in Section 4. For example, for the forest of Figure 3, assuming $\mathcal{P} = \{\top, b{\cdot}c, b, c, \epsilon\}$, we would obtain an abstraction where

$$\overleftarrow{\chi}^{\mathrm{E}} = \{(5, b, b, 1), (5, b, \epsilon, 1), (3, b, \epsilon, 2), (2, b, c, 1), \dots\}.$$

We have that *abs* is basically an algebra morphism between multi-attribute data forests with rooting and concatenation and forest profiles with profile rooting and profile concatenation. Further, the profile $abs(\epsilon, \mathbf{t}, \epsilon)$ is a derivable root profile whenever $\mathbf{t} \models \phi$; and every derivable root profile is the abstraction of some tree $\mathbf{t}$ so that $\mathbf{t} \models \phi$. (The formal definition of *abs* and the properties are described in the Appendix.) As a corollary from these properties, we have the following.

COROLLARY 6.6. *There is a derivable root forest profile if, and only if, ϕ is satisfiable.*

By the above Corollary 6.6 and Proposition 5.12, we can check in 2EXPSPACE if there is a derivable root profile. This is 2EXPSPACE in the size of $\mathcal{P}_\phi$. Although bringing a formula φ into direct unnested normal form may result in a doubly exponential formula, by Corollary 6.3 it can be stored in exponential space, and $\mathcal{P}_\phi$ is then singly exponential. Hence, the procedure is 3EXPSPACE in the original formula φ. Thus, the decision procedure is in 3EXPSPACE and Theorem 3.1 follows.

Note that if the input formula is in direct normal form then we save one exponential in the reduction and we hence obtain a 2EXPSPACE decision procedure.

THEOREM 6.7. *The satisfiability problem for formulas of* XPath($^*\!\leftarrow, \downarrow_*, \rightarrow^*, =$) *in direct normal form is decidable in* 2EXPSPACE.

7. DISCUSSION

We have shown that XPath with downward, rightward and leftward reflexive-transitive axes is decidable. To show this, we devised an algebra with good monotonicity properties. This seems to be the right kind of approach to work with transitive relations, and it generalizes and simplifies, in some aspects, the work of [11].

Upward axes

One natural question that stems from the result presented here is whether it can be extended to work with an *upward* axis as well. However, we claim (without a proof) that already SAT-XPath($\uparrow^*, \rightarrow^*, =$) has a non-primitive recursive lower bound. Indeed, this can be proved by reusing the results on lower bounds of [14]. The cited work shows that XPath with one non-reflexive transitive axis is enough to prove non-primitive recursiveness provided that the axis is functional (*i.e.*, the transitive closure of an axis like $\rightarrow, \leftarrow, \uparrow$ but unlike $\downarrow$). Here, however, we feature *reflexive*-transitive axes instead of only transitive. Therefore, in principle we cannot use this result. However, one can somehow code $\uparrow^+$ with $\rightarrow^*[a]\uparrow^*[\neg a]$ for some label a. We leave the proof of this claim for the journal version of the present work.

By the previous claim, although it could be that full transitive XPath is decidable, it would have a non-primitive recursive lower bound. We can then answer negatively to the conjecture proposed in [11, Conjecture 2], stating that XPath($^*\!\leftarrow, \downarrow_*, \uparrow^*, \rightarrow^*, =$) be decidable in elementary time.

Future work

- The present work can be seen as a step forward in answering [11, Conjecture 1], suggesting that the extension of XPath($^*\!\leftarrow, \downarrow_*, \rightarrow^*, =$) with the *child* axis is decidable with elementary complexity. Our approach may perhaps be extended to handle the child relation.

- We suspect that XPath($^*\!\leftarrow, \downarrow_*, \rightarrow^*, =$) is in fact hard for 2EXPSPACE, even when the formulas are in direct normal form, and hence that SAT-direct-XPath($^*\!\leftarrow, \downarrow_*, \rightarrow^*, =$) is 2EXPSPACE-complete.

- We would also like to investigate further the approach taken in this paper to attempt to generalize it to work with the class of reflexive-transitive closures of regular languages.

8. REFERENCES

[1] Vince Bárány, Mikołaj Bojańczyk, Diego Figueira, and Paweł Parys. Decidable classes of documents for XPath. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'12)*, Leibniz International Proceedings in Informatics (LIPIcs), Hyderabad, India, 2012. Leibniz-Zentrum für Informatik.

[2] Michael Benedikt, Wenfei Fan, and Floris Geerts. XPath satisfiability in the presence of DTDs. *Journal of the ACM*, 55(2):1–79, 2008.

[3] Mikołaj Bojańczyk and Sławomir Lasota. An extension of data automata that captures XPath. In *Annual IEEE Symposium on Logic in Computer Science (LICS '10)*, 2010.

[4] Mikołaj Bojańczyk, Anca Muscholl, Thomas Schwentick, and Luc Segoufin. Two-variable logic on data trees and XML reasoning. *Journal of the ACM*, 56(3):1–48, 2009.

[5] James Clark and Steve DeRose. XML path language (XPath). Website, 1999. W3C Recommendation. `http://www.w3.org/TR/xpath`.

[6] Claire David, Leonid Libkin, and Tony Tan. Efficient reasoning about data trees via integer linear programming. *ACM Transactions on Database Systems*, 37(3):19, 2012.

[7] Wenfei Fan, Chee Yong Chan, and Minos N. Garofalakis. Secure XML querying with security views. In *ACM SIGACT-SIGMOD-SIGART*

International Conference on Management of Data (SIGMOD'04), pages 587–598. ACM Press, 2004.

[8] Diego Figueira. Satisfiability of downward XPath with data equality tests. In *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'09)*, pages 197–206. ACM Press, 2009.

[9] Diego Figueira. Forward-XPath and extended register automata on data-trees. In *International Conference on Database Theory (ICDT'10)*. ACM Press, 2010.

[10] Diego Figueira. *Reasoning on Words and Trees with Data*. Phd thesis, Laboratoire Spécification et Vérification, ENS Cachan, France, December 2010.

[11] Diego Figueira. A decidable two-way logic on data words. In *Annual IEEE Symposium on Logic in Computer Science (LICS'11)*, pages 365–374, Toronto, Canada, 2011. IEEE Computer Society Press.

[12] Diego Figueira. Alternating register automata on finite data words and trees. *Logical Methods in Computer Science*, 8(1), 2012.

[13] Diego Figueira. Decidability of downward XPath. *ACM Trans. Comput. Log.*, 13(4), 2012.

[14] Diego Figueira and Luc Segoufin. Future-looking logics on data words and trees. In *Int. Symp. on Mathematical Foundations of Comp. Sci. (MFCS'09)*, volume 5734 of *LNCS*, pages 331–343. Springer, 2009.

[15] Diego Figueira and Luc Segoufin. Bottom-up automata on data trees and vertical XPath. In *International Symposium on Theoretical Aspects of Computer Science (STACS'11)*, Leibniz International Proceedings in Informatics (LIPIcs). Leibniz-Zentrum für Informatik, 2011.

[16] Floris Geerts and Wenfei Fan. Satisfiability of XPath queries with sibling axes. In *International Symposium on Database Programming Languages (DBPL'05)*, volume 3774 of *Lecture Notes in Computer Science*, pages 122–137. Springer, 2005.

[17] Georg Gottlob, Christoph Koch, and Reinhard Pichler. Efficient algorithms for processing XPath queries. *ACM Transactions on Database Systems*, 30(2):444–491, 2005.

[18] Marcin Jurdziński and Ranko Lazić. Alternating automata on data trees and xpath satisfiability. *ACM Trans. Comput. Log.*, 12(3):19, 2011.

[19] Michael Kaminski and Nissim Francez. Finite-memory automata. *Theoretical Computer Science*, 134(2):329–363, 1994.

[20] Michael Kaminski and Tony Tan. Tree automata over infinite alphabets. In *Pillars of Computer Science*, volume 4800 of *Lecture Notes in Computer Science*, pages 386–423. Springer, 2008.

[21] Wim Martens and Frank Neven. Frontiers of tractability for typechecking simple xml transformations. *J. Comput. Syst. Sci.*, 73(3):362–390, 2007.

[22] Frank Neven, Thomas Schwentick, and Victor Vianu. Finite state machines for strings over infinite alphabets. *ACM Trans. Comput. Log.*, 5(3):403–435, 2004.

[23] Tony Tan. An automata model for trees with ordered data values. In *Annual IEEE Symposium on Logic in Computer Science (LICS'12)*, pages 586–595. IEEE Computer Society Press, 2012.

APPENDIX

We organize this appendix with the missing proofs, grouped by sections on the main body. In each case we repeat the statements for the reader's convenience.

A. APPENDIX TO SECTION 4

LEMMA A.1 (DISTRIBUTIVITY OF $(\cdot,\cup)$). *For every $P, P' \subseteq \mathcal{P}$ and $\chi, \chi' \subseteq \mathbb{D} \times \Pi$*

1. $P{\cdot}(\chi \cup \chi') = P{\cdot}\chi \cup P{\cdot}\chi'$,
2. $(P \cup P'){\cdot}\chi = P{\cdot}\chi \cup P'{\cdot}\chi$.

PROOF. We have that

$$\begin{aligned}
&P{\cdot}(\chi \cup \chi') \\
&= \chi \cup \chi' \cup \{(d, \alpha'{\cdot}\alpha, \beta, i) \mid (d, \alpha, \beta, i) \in \chi \cup \chi', \alpha' \in P\} && \text{(by definition of `$\cdot$')} \\
&= \chi \cup \{(d, \alpha'{\cdot}\alpha, \beta, i) \mid (d, \alpha, \beta, i) \in \chi, \alpha' \in P\} \cup \\
&\quad \chi' \cup \{(d, \alpha'{\cdot}\alpha, \beta, i) \mid (d, \alpha, \beta, i) \in \chi', \alpha' \in P\} \\
&= P{\cdot}\chi \cup P{\cdot}\chi'. && \text{(by definition of `$\cdot$')}
\end{aligned}$$

which proves the first statement.

On the other hand,

$$\begin{aligned}
&(P \cup P'){\cdot}\chi \\
&= \chi \cup \{(d, \alpha'{\cdot}\alpha, \beta, i) \mid (d, \alpha, \beta, i) \in \chi, \alpha' \in P \cup P'\} \\
&= \chi \cup \{(d, \alpha'{\cdot}\alpha, \beta, i) \mid (d, \alpha, \beta, i) \in \chi, \alpha' \in P\} \cup \\
&\quad \chi \cup \{(d, \alpha'{\cdot}\alpha, \beta, i) \mid (d, \alpha, \beta, i) \in \chi, \alpha' \in P'\} \\
&= P{\cdot}\chi \cup P'{\cdot}\chi
\end{aligned}$$

which proves the second statement. □

LEMMA A.2. *If $\mathfrak{f}_1 + \mathfrak{f}_2 = \mathfrak{f}_3$ and $\mathfrak{f}_1, \mathfrak{f}_2 \in \mathfrak{F}$, then $\mathfrak{f}_3 \in \mathfrak{F}$.*

PROOF. If $\overleftarrow{\chi}^{\mathrm{E}}_3(\alpha, \beta, i) = \{d\}$, then $\overleftarrow{\chi}^{\mathrm{E}}_1(\alpha, \beta, i) = \{d\}$, and hence $d \in R_1$ and therefore $d \in R_3 = R_1$. A symmetrical argument applies for $\overrightarrow{\chi}^{\mathrm{E}}_3$ and $\overrightarrow{\chi}^{\mathrm{E}}_2$. □

LEMMA A.3 (DISTRIBUTIVITY OF $(\cdot,+)$). *For every $\mathfrak{f}_1, \mathfrak{f}_2 \in \mathfrak{F}$ and $\chi \subseteq \mathbb{D} \times \Pi$*

- $[\![\mathfrak{f}_1 + \mathfrak{f}_2]\!]{\cdot}\chi = [\![\mathfrak{f}_1]\!]{\cdot}([\![\mathfrak{f}_2]\!]{\cdot}\chi)$,
- $\langle\!\langle \mathfrak{f}_1 + \mathfrak{f}_2 \rangle\!\rangle{\cdot}\chi = \langle\!\langle \mathfrak{f}_2 \rangle\!\rangle{\cdot}(\langle\!\langle \mathfrak{f}_1 \rangle\!\rangle{\cdot}\chi)$.

PROOF. Let $\mathfrak{f}_+ = \mathfrak{f}_1 + \mathfrak{f}_2$.

$$\begin{aligned}
&[\![\mathfrak{f}_1 + \mathfrak{f}_2]\!]{\cdot}\chi \\
&= \{\alpha \mid (d, \alpha, \beta, i) \in \overrightarrow{\chi}^{\mathfrak{r}_1}_+\}{\cdot}\chi && \text{(by definition of } [\![\ ]\!]) \\
&= \{\alpha \mid (d, \alpha, \beta, i) \in \overrightarrow{\chi}^{\mathfrak{r}_1}_1 \cup [\![\mathfrak{f}_1]\!]{\cdot}\overrightarrow{\chi}^{\mathfrak{r}_1}_2\}{\cdot}\chi && \text{(by definition of } \mathfrak{f}_1 + \mathfrak{f}_2) \\
&= \big(\{\alpha \mid (d, \alpha, \beta, i) \in \overrightarrow{\chi}^{\mathfrak{r}_1}_1\} \cup \{\alpha \mid (d, \alpha, \beta, i) \in [\![\mathfrak{f}_1]\!]{\cdot}\overrightarrow{\chi}^{\mathfrak{r}_1}_2\}\big){\cdot}\chi \\
&= \big([\![\mathfrak{f}_1]\!] \cup \{\alpha \mid (d, \alpha, \beta, i) \in [\![\mathfrak{f}_1]\!]{\cdot}\overrightarrow{\chi}^{\mathfrak{r}_1}_2\}\big){\cdot}\chi && \text{(by } (\cdot,\cup) \text{ distributivity)} \\
&= [\![\mathfrak{f}_1]\!]{\cdot}\{\alpha \mid (d, \alpha, \beta, i) \in [\![\mathfrak{f}_1]\!]{\cdot}\overrightarrow{\chi}^{\mathfrak{r}_1}_2\}{\cdot}\chi \\
&= [\![\mathfrak{f}_1]\!]{\cdot}\{(d, \alpha'{\cdot}\alpha, \beta, i) \mid (d, \alpha, \beta, i) \in \chi, \\
&\quad \alpha' \in \{\alpha \mid (d, \alpha, \beta, i) \in [\![\mathfrak{f}_1]\!]{\cdot}\overrightarrow{\chi}^{\mathfrak{r}_1}_2\}\} \\
&= [\![\mathfrak{f}_1]\!]{\cdot}\chi \cup \{(d, \alpha'{\cdot}\alpha, \beta, i) \mid (d, \alpha, \beta, i) \in \chi, \\
&\quad \alpha' \in \{\alpha \mid (d, \alpha, \beta, i) \in \{\alpha \mid (d, \alpha, \beta, i) \in \overrightarrow{\chi}^{\mathfrak{r}_1}_1\}{\cdot}\overrightarrow{\chi}^{\mathfrak{r}_1}_2\}\} \\
&= [\![\mathfrak{f}_1]\!]{\cdot}\chi \cup \{(d, \alpha'{\cdot}\alpha, \beta, i) \mid (d, \alpha, \beta, i) \in \chi, \\
&\quad \alpha' \in \{\alpha{\cdot}\alpha' \mid (d, \alpha, \beta, i) \in \overrightarrow{\chi}^{\mathfrak{r}_1}_1 \wedge \\
&\quad ((d', \alpha', \beta', i') \in \overrightarrow{\chi}^{\mathfrak{r}_1}_2 \vee \alpha' = \epsilon)\}\} \\
&= [\![\mathfrak{f}_1]\!]{\cdot}\chi \cup \{(d, \alpha'{\cdot}\alpha, \beta, i) \mid (d, \alpha, \beta, i) \in \chi, \\
&\quad \alpha' \in \{\alpha{\cdot}\alpha' \mid \alpha \in [\![\mathfrak{f}_1]\!], \alpha' \in [\![\mathfrak{f}_2]\!] \cup \{\epsilon\}\}\} \\
&= [\![\mathfrak{f}_1]\!]{\cdot}\chi \cup \{(d, \alpha'{\cdot}\alpha''{\cdot}\alpha, \beta, i) \mid (d, \alpha, \beta, i) \in \chi, \\
&\quad \alpha' \in [\![\mathfrak{f}_1]\!], \alpha'' \in [\![\mathfrak{f}_2]\!] \cup \{\epsilon\}\} \\
&= [\![\mathfrak{f}_1]\!]{\cdot}\chi \cup
\end{aligned}$$

$$
\begin{aligned}
&\{(d,\alpha'\cdot\alpha,\beta,i) \mid (d,\alpha,\beta,i)\in\chi, \alpha'\in[\mathfrak{f}_1\rangle\}\cup\\
&\{(d,\alpha'\cdot\alpha''\cdot\alpha,\beta,i) \mid (d,\alpha,\beta,i)\in\chi, \alpha'\in[\mathfrak{f}_1\rangle, \alpha''\in[\mathfrak{f}_2\rangle\}\\
=\ &[\mathfrak{f}_1\rangle\cdot\chi\cup[\mathfrak{f}_1\rangle\cdot\chi\cup\\
&\{(d,\alpha'\cdot\alpha''\cdot\alpha,\beta,i) \mid (d,\alpha,\beta,i)\in\chi, \alpha'\in[\mathfrak{f}_1\rangle, \alpha''\in[\mathfrak{f}_2\rangle\}\\
=\ &[\mathfrak{f}_1\rangle\cdot\chi\cup\\
&[\mathfrak{f}_1\rangle\cdot\{(d,\alpha''\cdot\alpha,\beta,i) \mid (d,\alpha,\beta,i)\in\chi, \alpha''\in[\mathfrak{f}_2\rangle\}\\
=\ &[\mathfrak{f}_1\rangle\cdot\chi\cup[\mathfrak{f}_1\rangle\cdot([\mathfrak{f}_2\rangle\cdot\chi)\\
=\ &[\mathfrak{f}_1\rangle\cdot([\mathfrak{f}_2\rangle\cdot\chi)
\end{aligned}
$$

An identical reasoning can be used to prove that $\langle[\mathfrak{f}_1+\mathfrak{f}_2]\cdot\chi = \langle[\mathfrak{f}_2]\cdot(\langle[\mathfrak{f}_1]\cdot\chi)$. □

LEMMA A.4 (+ ASSOCIATIVITY). *For every* $\mathfrak{f}_1,\mathfrak{f}_2,\mathfrak{f}_3\in\mathfrak{F}$, $(\mathfrak{f}_1+\mathfrak{f}_2)+\mathfrak{f}_3 = \mathfrak{f}_1+(\mathfrak{f}_2+\mathfrak{f}_3)$.

PROOF. Let $\mathfrak{f}_{12}=\mathfrak{f}_1+\mathfrak{f}_2$, $\mathfrak{f}_{23}=\mathfrak{f}_2+\mathfrak{f}_3$.

We first show that if $(\mathfrak{f}_1+\mathfrak{f}_2)+\mathfrak{f}_3$ can be applied, then $\mathfrak{f}_1+(\mathfrak{f}_2+\mathfrak{f}_3)$ can also be applied. Condition (a) is immediate, since $R_1=R_2=R_3$. Now we show that conditions (b) and (c) hold for $\mathfrak{f}_2+\mathfrak{f}_3$. Since $\overrightarrow{\chi}_2^{\mathrm{E}} = \overrightarrow{\chi}_{12}^{\mathrm{E}}$ and since the conditions between $\mathfrak{f}_{12}+\mathfrak{f}_3$ hold, we have that $\overrightarrow{\chi}_2^{\mathrm{E}} = \overrightarrow{\chi}_3^{\mathrm{I}}\cup[\mathfrak{f}_3\rangle\cdot\overrightarrow{\chi}_3^{\mathrm{E}}$ and hence that condition (b) holds for $\mathfrak{f}_2+\mathfrak{f}_3$.

To show that condition (c) holds, note that

$$
\begin{aligned}
\overleftarrow{\chi}_3^{\mathrm{E}} &= \overleftarrow{\chi}_{12}^{\mathrm{I}}\cup\langle[\mathfrak{f}_{12}]\cdot\overleftarrow{\chi}_{12}^{\mathrm{E}} && \text{(by condition (c) of } \mathfrak{f}_{12}+\mathfrak{f}_3)\\
&= \overleftarrow{\chi}_2^{\mathrm{I}}\cup\langle[\mathfrak{f}_2]\cdot\overleftarrow{\chi}_1^{\mathrm{I}}\cup\langle[\mathfrak{f}_{12}]\cdot\overleftarrow{\chi}_1^{\mathrm{E}} && \text{(by definition of } \mathfrak{f}_1+\mathfrak{f}_2)\\
&= \overleftarrow{\chi}_2^{\mathrm{I}}\cup\langle[\mathfrak{f}_2]\cdot\overleftarrow{\chi}_1^{\mathrm{I}}\cup\langle[\mathfrak{f}_2]\cdot(\langle[\mathfrak{f}_1]\cdot\overleftarrow{\chi}_1^{\mathrm{E}}) && \text{(by } (\cdot,+) \text{ distributivity)}\\
&= \overleftarrow{\chi}_2^{\mathrm{I}}\cup\langle[\mathfrak{f}_2]\cdot(\overleftarrow{\chi}_1^{\mathrm{I}}\cup\langle[\mathfrak{f}_1]\cdot\overleftarrow{\chi}_1^{\mathrm{E}}) && \text{(by } (\cdot,\cup) \text{ distributivity)}\\
&= \overleftarrow{\chi}_2^{\mathrm{I}}\cup\langle[\mathfrak{f}_2]\cdot\overleftarrow{\chi}_2^{\mathrm{E}}. && \text{(by condition (c) of } \mathfrak{f}_1+\mathfrak{f}_2)
\end{aligned}
$$

Hence, we obtain that $\overleftarrow{\chi}_3^{\mathrm{E}} = \overleftarrow{\chi}_2^{\mathrm{I}}\cup\langle[\mathfrak{f}_2]\cdot\overleftarrow{\chi}_2^{\mathrm{E}}$, and thus condition (c) holds for $\mathfrak{f}_2+\mathfrak{f}_3$.

On the other hand, assuming that $\mathfrak{f}_1+(\mathfrak{f}_2+\mathfrak{f}_3)$, we can show that the conditions to apply $(\mathfrak{f}_1+\mathfrak{f}_2)+\mathfrak{f}_3$ hold, in the same way as before, since all the definitions are symmetrical, where $\overrightarrow{\chi}_1^{\mathrm{E}} = \overrightarrow{\chi}_2^{\mathrm{I}}\cup[\mathfrak{f}_2\rangle\cdot\overrightarrow{\chi}_2^{\mathrm{E}}$ and $\overleftarrow{\chi}_2^{\mathrm{E}} = \overleftarrow{\chi}_1^{\mathrm{I}}\cup\langle[\mathfrak{f}_1]\cdot\overleftarrow{\chi}_1^{\mathrm{E}}$.

We now show that $(\mathfrak{f}_1+\mathfrak{f}_2)+\mathfrak{f}_3 = \mathfrak{f}_1+(\mathfrak{f}_2+\mathfrak{f}_3)$. Let $\mathfrak{f}_{1(23)}=\mathfrak{f}_1+(\mathfrak{f}_2+\mathfrak{f}_3)$ and $\mathfrak{f}_{(12)3}=(\mathfrak{f}_1+\mathfrak{f}_2)+\mathfrak{f}_3$.

We have that $\overrightarrow{\chi}_{1(23)}^{\mathrm{E}} = \overrightarrow{\chi}_{23}^{\mathrm{E}} = \overrightarrow{\chi}_3^{\mathrm{E}} = \overrightarrow{\chi}_{(12)3}^{\mathrm{E}}$ by item condition (+2) of +. In a similar way, by condition (+3) we have that $\overleftarrow{\chi}_{1(23)}^{\mathrm{E}} = \overleftarrow{\chi}_{(12)3}^{\mathrm{E}}$. On the other hand, we have

$$
\begin{aligned}
\overrightarrow{\chi}_{1(23)}^{\mathrm{I}} &= \overrightarrow{\chi}_1^{\mathrm{I}}\cup[\mathfrak{f}_1\rangle\cdot\overrightarrow{\chi}_{23}^{\mathrm{I}} && \text{(by condition (+4))}\\
&= \overrightarrow{\chi}_1^{\mathrm{I}}\cup[\mathfrak{f}_1\rangle\cdot(\overrightarrow{\chi}_2^{\mathrm{I}}\cup[\mathfrak{f}_2\rangle\cdot\overrightarrow{\chi}_3^{\mathrm{I}}) && \text{(by condition (+4))}\\
&= \overrightarrow{\chi}_1^{\mathrm{I}}\cup[\mathfrak{f}_1\rangle\cdot\overrightarrow{\chi}_2^{\mathrm{I}}\cup[\mathfrak{f}_1\rangle\cdot([\mathfrak{f}_2\rangle\cdot\overrightarrow{\chi}_3^{\mathrm{I}}) && \text{(by } (\cdot,\cup) \text{ distributivity)}\\
&= \overrightarrow{\chi}_1^{\mathrm{I}}\cup[\mathfrak{f}_1\rangle\cdot\overrightarrow{\chi}_2^{\mathrm{I}}\cup[\mathfrak{f}_{12}\rangle\cdot\overrightarrow{\chi}_3^{\mathrm{I}} && \text{(by } (\cdot,+) \text{ distributivity)}\\
&= \overrightarrow{\chi}_{12}^{\mathrm{I}}\cup[\mathfrak{f}_{12}\rangle\cdot\overrightarrow{\chi}_3^{\mathrm{I}} && \text{(by condition (+4))}\\
&= \overrightarrow{\chi}_{(12)3}^{\mathrm{I}} && \text{(by condition (+4))}
\end{aligned}
$$

and by a similar reasoning using condition (+5) we obtain that $\overleftarrow{\chi}_{1(23)}^{\mathrm{I}} = \overleftarrow{\chi}_{(12)3}^{\mathrm{I}}$. Hence, $(\mathfrak{f}_1+\mathfrak{f}_2)+\mathfrak{f}_3 = \mathfrak{f}_1+(\mathfrak{f}_2+\mathfrak{f}_3)$. □

LEMMA A.5. *For every* $\mathfrak{f}_1,\mathfrak{f}_2\in(a,\bar{d})\mathfrak{f}_3$ *and every* $\chi\subseteq\mathbb{D}\times\Pi$,

$$
\begin{aligned}
&[\mathfrak{f}_1\rangle\cdot[\mathfrak{f}_2\rangle\cdot\chi = [\mathfrak{f}_1\rangle\cdot\chi = [\mathfrak{f}_2\rangle\cdot\chi =\\
&\langle[\mathfrak{f}_1]\cdot\langle[\mathfrak{f}_2]\cdot\chi = \langle[\mathfrak{f}_1]\cdot\chi = \langle[\mathfrak{f}_2]\cdot\chi.
\end{aligned}
$$

PROOF. By condition (c) of rooting and definition of $[\ \rangle$, $\langle[\ ]$, we have that $\langle[\mathfrak{f}_1] = [\mathfrak{f}_1\rangle = \langle[\mathfrak{f}_2] = [\mathfrak{f}_2\rangle = \boldsymbol{\sigma}_a$. By definition of $\boldsymbol{\sigma}_a$, it is immediate that $\boldsymbol{\sigma}_a\cdot\boldsymbol{\sigma}_a\cdot\chi = \boldsymbol{\sigma}_a\cdot\chi$. Hence, the statement follows. □

LEMMA A.6. *For every* $\mathfrak{f}_1\in(a,\bar{d})\mathfrak{f}_2$,

$$
\overrightarrow{\chi}_1^{\mathrm{I}} = [\mathfrak{f}_1\rangle\cdot\overrightarrow{\chi}_1^{\mathrm{I}} \qquad \overleftarrow{\chi}_1^{\mathrm{I}} = [\mathfrak{f}_1\rangle\cdot\overleftarrow{\chi}_1^{\mathrm{I}}.
$$

PROOF. By condition (c) of rooting, $[\mathfrak{f}_1\rangle = \boldsymbol{\sigma}_a$ and since $\overrightarrow{\chi}_1^{\mathrm{I}} = \{(d,\alpha,\beta\cdot\gamma,i) \mid \exists\alpha'.(d,\alpha',\gamma,i)\in\overrightarrow{\chi}_2^{\mathrm{I}}\cup\overleftarrow{\chi}_2^{\mathrm{I}}, \alpha,\beta\in\boldsymbol{\sigma}_a\}\cup\bigcup_{i\in[\mathbf{k}]}(\{d(i)\}\times(\boldsymbol{\sigma}_a\setminus\{\epsilon\})\times\boldsymbol{\sigma}_a\times\{i\})$ it follows that $\boldsymbol{\sigma}_a\cdot\overrightarrow{\chi}_1^{\mathrm{I}} = \overrightarrow{\chi}_1^{\mathrm{I}}$, obtaining that $\overrightarrow{\chi}_1^{\mathrm{I}} = [\mathfrak{f}_1\rangle\cdot\overrightarrow{\chi}_1^{\mathrm{I}}$. Since $\overleftarrow{\chi}_1^{\mathrm{I}} = \overrightarrow{\chi}_1^{\mathrm{I}}$ by condition (c), we also obtain that $\overleftarrow{\chi}_1^{\mathrm{I}} = [\mathfrak{f}_1\rangle\cdot\overleftarrow{\chi}_1^{\mathrm{I}}$. □

B. APPENDIX TO SECTION 5

LEMMA B.1. $\preceq$ *and* $\precsim$ *are partial orders over* $\mathfrak{F}$.

PROOF. If $\mathfrak{f} \preceq \mathfrak{f}' \preceq \mathfrak{f}''$, suppose $\mathfrak{f}' = \mathfrak{f}[d_1 \mapsto d_1, d_1'] \cdots \mathfrak{f}[d_n \mapsto d_n, d_n']$ and $\mathfrak{f}'' = \mathfrak{f}'[e_1 \mapsto e_1, e_1'] \cdots \mathfrak{f}[e_m \mapsto e_n, e_m']$. Therefore, $\mathfrak{f}'' = \mathfrak{f}[d_1 \mapsto d_1, d_1'] \cdots \mathfrak{f}[d_n \mapsto d_n, d_n'][e_1 \mapsto e_1, e_1'] \cdots \mathfrak{f}[e_m \mapsto e_n, e_m']$ and hence $\mathfrak{f} \preceq \mathfrak{f}''$. On the other hand, it is plain that $\sim$ preserves transitivity and reflexivity, and that $\mathfrak{f} \preceq \mathfrak{f}$. □

LEMMA B.2. *If $\mathfrak{f}_1 \preceq \mathfrak{f}_2$ then $\langle\!|\mathfrak{f}_1|\!\rangle = \langle\!|\mathfrak{f}_2|\!\rangle$ and $[\!|\mathfrak{f}_1|\!] = [\!|\mathfrak{f}_2|\!]$.*

PROOF. This is immediate from the definition of $\langle\!|\ |\!\rangle$, $[\!|\ |\!]$, and $\preceq$. Note that, since $\mathfrak{f}_1 \preceq \mathfrak{f}_2$ we have $\{(\alpha, \beta, i) \mid \exists d \in \mathbb{D}.(d, \alpha, \beta, i) \in \mathfrak{f}_1\} = \{(\alpha, \beta, i) \mid \exists d \in \mathbb{D}.(d, \alpha, \beta, i) \in \mathfrak{f}_2\}$. Hence, we obtain that $\alpha \in [\!|\mathfrak{f}_1|\!]$ if and only if there is some $(d, \alpha, \beta, i) \in \overrightarrow{\chi}_1^{\mathrm{I}}$ if and only if there is some $(d', \alpha, \beta, i) \in \overrightarrow{\chi}_2^{\mathrm{I}}$ if and only if $\alpha \in [\!|\mathfrak{f}_2|\!]$. A similar argument shows that $\langle\!|\mathfrak{f}_1|\!\rangle = \langle\!|\mathfrak{f}_2|\!\rangle$. □

LEMMA B.3. *If $\mathfrak{f} \preceq \mathfrak{f}'$ then $\xi(\mathfrak{f}) = \xi(\mathfrak{f}')$.*

PROOF. Suppose, without any loss of generality, that $\mathfrak{f}' = \mathfrak{f}[d \mapsto d, d']$, where $d \in \mathit{data}(\mathfrak{f}) \setminus R$ and $d' \notin \mathit{data}(\mathfrak{f})$. Since d' behaves just as d, we have that, for any $\bar{\alpha}, \bar{\alpha}' \in \Pi$, $a, a' \in \{\overleftarrow{\circ}, \overrightarrow{\circ}, \circ^{\downarrow}, \circ\}$, $d' \in \mathfrak{f}'.\chi_a(\bar{\alpha}) \cap \chi_{a'}(\bar{\alpha}')$ iff $d \in \mathfrak{f}.\chi_a(\bar{\alpha}) \cap \mathfrak{f}.\chi_{a'}(\bar{\alpha}')$. Hence, $\mathfrak{f}'.\chi_a(\bar{\alpha}) \cap \mathfrak{f}'.\chi_{a'}(\bar{\alpha}') \neq \emptyset$ iff $\mathfrak{f}.\chi_a(\bar{\alpha}) \cap \mathfrak{f}.\chi_{a'}(\bar{\alpha}') \neq \emptyset$.

Further, since $d \notin R$ we have that if $d \in \chi_a(\bar{\alpha})$ in $\mathfrak{f}'$ (or in $\mathfrak{f}$) then $|\chi_a(\bar{\alpha})| > 1$. Hence $\xi(\mathfrak{f})(\bar{\alpha}, a) = \xi(\mathfrak{f}')(\bar{\alpha}, a) = 2+$ whenever $d \in \mathfrak{f}.\chi_a(\bar{\alpha})$. Further, since $\mathfrak{f}'$ preserves all the data values of $\mathfrak{f}$ with their descriptions, $\xi(\mathfrak{f})(\bar{\alpha}, a) = \xi(\mathfrak{f}')(\bar{\alpha}, a)$ whenever $d \notin \mathfrak{f}.\chi_a(\bar{\alpha}) \cap \mathfrak{f}.\chi_{a'}(\bar{\alpha}')$. □

LEMMA B.4. *If $d \in \mathit{data}(\mathfrak{f}) \setminus R$ and $d' \notin \mathit{data}(\mathfrak{f})$, we have that if $\mathfrak{f} \in \mathfrak{F}$ then $\mathfrak{f}[d \mapsto d, d'] \in \mathfrak{F}$.*

PROOF. This is immediate from the definition of validity. Since $d' \notin \mathit{data}(\mathfrak{f})$, then $\mathfrak{f}' = \mathfrak{f}[d \mapsto d, d']$ is the result of adding some fresh data value $d' \notin R$ with the same description as d'. Since $\mathfrak{f}' \notin \mathfrak{F}$, there is some $\{d''\} = \overrightarrow{\chi}^{\mathrm{E}'}(\bar{\alpha})$ for $\bar{\alpha} \in \Pi$ and $d'' \notin R$. By definition of $\mathfrak{f}'$, this means that $d'' \neq d$ and $d'' \neq d$, and hence that $\{d''\} = \overrightarrow{\chi}^{\mathrm{E}}(\bar{\alpha})$, and thus $\mathfrak{f} \notin \mathfrak{F}$, which is a contradiction. A symmetric argument can also be applied to $\overleftarrow{\chi}^{\mathrm{E}}$. □

Appendix to Section 5.3

LEMMA 5.1 (FIGURE 4). *For every $\mathfrak{f}_1 \succeq \mathfrak{f}_2 \in (a, \bar{d})\mathfrak{f}_3$, there is $n \in \mathbb{N}$, and $\mathfrak{f}_i' \in (a, \bar{d})\mathfrak{f}_3$ for every $i \in [n]$ so that*

$$\mathfrak{f}_1 = \mathfrak{f}_1' + \cdots + \mathfrak{f}_n'.$$

PROOF OF LEMMA 5.1. Since $\mathfrak{f}_1 \succeq \mathfrak{f}_2$, suppose

$$\mathfrak{f}_1 = \mathfrak{f}_2[d_1 \mapsto d_1, d_1'] \cdots [d_t \mapsto d_t, d_t'] \tag{$\dagger$}$$

where $\{d_1, \ldots, d_t\} \subseteq \mathit{data}(\mathfrak{f}_2) \setminus R_2$; $\{d_1', \ldots, d_t'\} \cap \mathit{data}(\mathfrak{f}_2) = \emptyset$; and $d_i' \neq d_j'$ for all $i \neq j$. We define $n = t + 2$.

The idea is that we will define $\mathfrak{f}_0', \mathfrak{f}_1', \ldots, \mathfrak{f}_{t+1}'$ so that $\mathfrak{f}_i'$ is, modulo some renaming of data values, the same as $\mathfrak{f}_2$ for all strictly internal values, and with the same descriptions. However, $\mathfrak{f}_i'$ has more *external* data values than $\mathfrak{f}_2$—namely $d_1', \ldots, d_t'$—that have basically the same descriptions as $d_1, \ldots, d_t$ in $\mathfrak{f}_2$. Although technically we could make use of less profiles, we prefer to define $t+1$ different profiles, to preserve some symmetries in the definitions, and thus to simplify proofs. We define $\mathfrak{f}_0', \ldots, \mathfrak{f}_{t+1}'$ in Figure **??**.

We shall now show that for every $\mathfrak{f}_i'$ there is some $\mathfrak{f}_i''$ so that $\mathfrak{f}_i' \sim \mathfrak{f}_i'' \in (a, \bar{d})\mathfrak{f}_3$.

CLAIM B.4.1. *For every $i \in [n]$, $\mathfrak{f}_i' \in (a, \bar{d})\mathfrak{f}_3$.*

PROOF. First note that $\mathfrak{f}_0', \mathfrak{f}_{t+1}' \in (a, \bar{d})\mathfrak{f}_3$, since all the internal descriptions of the internal data values are preserved, and $\xi(\mathfrak{f}_0') = \xi(\mathfrak{f}_{t+1}') = \xi(\mathfrak{f}_2)$. The fact that they all have the same fingerprints is because:

- For every $a \in \mathcal{A}$, $\bar{\alpha} \in \Pi$, and every data value $e \in \mathit{data}(\mathfrak{f}_2)$, we have that $e \in \chi_a(\bar{\alpha})$ in $\mathfrak{f}_2$ iff $e \in \chi_a(\bar{\alpha})$ in $\mathfrak{f}_0'$ iff $e \in \chi_a(\bar{\alpha})$ in $\mathfrak{f}_{t+1}'$. This is because, since $\overleftarrow{\chi}_{t+1}^{\mathrm{I}}{}'(e) = \overleftarrow{\chi}_0^{\mathrm{I}}{}'(e) = \overleftarrow{\chi}_2^{\mathrm{I}}(e)$, $\overrightarrow{\chi}_{t+1}^{\mathrm{I}}{}'(e) = \overrightarrow{\chi}_0^{\mathrm{I}}{}'(e) = \overrightarrow{\chi}_2^{\mathrm{I}}(e)$, $\overleftarrow{\chi}_0^{\mathrm{E}}{}'(e) = \overleftarrow{\chi}_2^{\mathrm{E}}(e)$, $\overrightarrow{\chi}_{t+1}^{\mathrm{E}}{}'(e) = \overrightarrow{\chi}_2^{\mathrm{E}}(e)$, we have that: for all $a \in \{\circ, \circ^{\downarrow}, \overleftarrow{\circ}\}$, $\chi_a(e)$ in $\mathfrak{f}_0'$ is equal to $\chi_a(e)$ in $\mathfrak{f}_2$; and for all $a \in \{\circ, \circ^{\downarrow}, \overrightarrow{\circ}\}$, $\chi_a(e)$ in $\mathfrak{f}_{t+1}'$ is equal to $\chi_a(e)$ in $\mathfrak{f}_2$. And further,
  - if $a = \overrightarrow{\circ}$,

$$\begin{aligned}
\mathfrak{f}_0'.\chi_a(e) &= \overrightarrow{\chi}_0^{\mathrm{I}}{}'(e) \cup [\!|\mathfrak{f}_0'|\!\rangle \cdot \overrightarrow{\chi}_0^{\mathrm{E}}{}'(e) \\
&= \overrightarrow{\chi}_2^{\mathrm{I}}(e) \cup [\!|\mathfrak{f}_2|\!\rangle \cdot (\overrightarrow{\chi}_2^{\mathrm{I}}(e) \cup [\!|\mathfrak{f}_2|\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(e)) \\
&= \overrightarrow{\chi}_2^{\mathrm{I}}(e) \cup [\!|\mathfrak{f}_2|\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(e) && \text{(by Lemmas ?? and ??)} \\
&= \mathfrak{f}_2.\chi_a(e)
\end{aligned}$$

  - if $a = \overleftarrow{\circ}$,

$$\mathfrak{f}_{t+1}'.\chi_a(e) = \overleftarrow{\chi}_0^{\mathrm{I}}{}'(e) \cup \langle\!|\mathfrak{f}_{t+1}'|\!] \cdot \overleftarrow{\chi}_{t+1}^{\mathrm{E}}{}'(e)$$

$$\mathfrak{f}'_0 : \begin{cases} R'_0 = R_2, \\ \bar{\chi}'_0(e) = (\overleftarrow{\chi}^{\mathrm{E}}_2(e), \overrightarrow{\chi}^{\mathrm{I}}_2(e), \overleftarrow{\chi}^{\mathrm{I}}_2(e), \overrightarrow{\chi}^{\mathrm{I}}_2(e) \cup [\mathfrak{f}_2\rangle \cdot \overrightarrow{\chi}^{\mathrm{E}}_2(e)) \\ \qquad\qquad \text{for every } e \in \mathit{data}(\mathfrak{f}_2), \\ \bar{\chi}'_0(d'_j) = (\overleftarrow{\chi}^{\mathrm{E}}_2(d_j), \emptyset, \emptyset, \overrightarrow{\chi}^{\mathrm{I}}_2(d_j) \cup [\mathfrak{f}_2\rangle \cdot \overrightarrow{\chi}^{\mathrm{E}}_2(d_j)) \\ \qquad\qquad \text{for every } j \in [t]. \end{cases}$$

For every $i \in \{1, \ldots, t\}$ we define $\mathfrak{f}'_i$ as

$$\mathfrak{f}'_i : \begin{cases} R'_i = R_2, \\ \bar{\chi}'_i(e) = (\overleftarrow{\chi}^{\mathrm{I}}_2(e) \cup \langle\mathfrak{f}_2] \cdot \overleftarrow{\chi}^{\mathrm{E}}_2(e), \overrightarrow{\chi}^{\mathrm{I}}_2(e), \\ \qquad \overleftarrow{\chi}^{\mathrm{I}}_2(e), \overrightarrow{\chi}^{\mathrm{I}}_2(e) \cup [\mathfrak{f}_2\rangle \cdot \overrightarrow{\chi}^{\mathrm{E}}_2(e)) \\ \qquad\qquad \text{for every } e \in \mathit{data}(\mathfrak{f}_2) \setminus \{d_i\}, \\ \bar{\chi}'_i(d_i) = (\overleftarrow{\chi}^{\mathrm{I}}_2(d_i) \cup \langle\mathfrak{f}_2] \cdot \overleftarrow{\chi}^{\mathrm{E}}_2(d_i), \emptyset, \\ \qquad \emptyset, \overrightarrow{\chi}^{\mathrm{I}}_2(d_i) \cup [\mathfrak{f}_2\rangle \cdot \overrightarrow{\chi}^{\mathrm{E}}_2(d_i)), \\ \bar{\chi}'_i(d'_i) = (\langle\mathfrak{f}_2] \cdot \overleftarrow{\chi}^{\mathrm{E}}_2(d_i), \overrightarrow{\chi}^{\mathrm{I}}_2(d_i), \overleftarrow{\chi}^{\mathrm{I}}_2(d_i), [\mathfrak{f}_2\rangle \cdot \overrightarrow{\chi}^{\mathrm{E}}_2(d_i)), \\ \bar{\chi}'_i(d'_j) = (\overleftarrow{\chi}^{\mathrm{I}}_2(d_j) \cup \langle\mathfrak{f}_2] \cdot \overleftarrow{\chi}^{\mathrm{E}}_2(d_j), \emptyset, \emptyset, [\mathfrak{f}_2\rangle \cdot \overrightarrow{\chi}^{\mathrm{E}}_2(d_j)) \\ \qquad\qquad \text{for every } j < i, \\ \bar{\chi}'_i(d'_j) = (\langle\mathfrak{f}_2] \cdot \overleftarrow{\chi}^{\mathrm{E}}_2(d_j), \emptyset, \emptyset, \overrightarrow{\chi}^{\mathrm{I}}_2(d_j) \cup [\mathfrak{f}_2\rangle \cdot \overrightarrow{\chi}^{\mathrm{E}}_2(d_j)) \\ \qquad\qquad \text{for every } j > i. \end{cases}$$

$$\mathfrak{f}'_{t+1} : \begin{cases} R'_{t+1} = R_2, \\ \bar{\chi}'_{t+1}(e) = (\overleftarrow{\chi}^{\mathrm{I}}_2(e) \cup \langle\mathfrak{f}_2] \cdot \overleftarrow{\chi}^{\mathrm{E}}_2(e), \overrightarrow{\chi}^{\mathrm{I}}_2(e), \overleftarrow{\chi}^{\mathrm{I}}_2(e), \overrightarrow{\chi}^{\mathrm{E}}_2(e)) \\ \qquad\qquad \text{for every } e \in \mathit{data}(\mathfrak{f}_2), \\ \bar{\chi}'_{t+1}(d'_j) = (\overleftarrow{\chi}^{\mathrm{I}}_2(d_j) \cup \langle\mathfrak{f}_2] \cdot \overleftarrow{\chi}^{\mathrm{E}}_2(d_j), \emptyset, \emptyset, \overrightarrow{\chi}^{\mathrm{E}}_2(d_j)) \\ \qquad\qquad \text{for every } j \in [t]. \end{cases}$$

Figure 7: Definition of $\mathfrak{f}'_0$, $\mathfrak{f}'_{t+1}$ and $\mathfrak{f}'_i$ for all $i \in [t]$.

$$\begin{aligned} &= \overleftarrow{\chi}^{\mathrm{I}}_2(e) \cup \langle\mathfrak{f}_2] \cdot (\overleftarrow{\chi}^{\mathrm{I}}_2(e) \cup \langle\mathfrak{f}_2] \cdot \overleftarrow{\chi}^{\mathrm{E}}_2(e)) \\ &= \overleftarrow{\chi}^{\mathrm{I}}_2(e) \cup \langle\mathfrak{f}_2] \cdot \overleftarrow{\chi}^{\mathrm{E}}_2(e) && \text{(by Lemmas ?? and ??)} \\ &= \mathfrak{f}_2.\chi_a(e). \end{aligned}$$

- The data values d'_j do not change the fingerprint $\xi(\mathfrak{f}_2)$, because d'_j is added to all those $\chi_a(\bar{\alpha})$ where $d_j \in \mathfrak{f}_2.\chi_a(\bar{\alpha})$, and $\mathfrak{f}_2.\chi_a(\bar{\alpha})$ has already two data values (otherwise d_j would be rigid).
  Therefore, for any $a \in \mathcal{A}$, $\bar{\alpha} \in \Pi$, if some $d'_j \in \mathfrak{f}'_0.\chi_a(\bar{\alpha})$, then $|\mathfrak{f}_2.\chi_a(\bar{\alpha})| \geq 2$, and $\mathfrak{f}_2.\chi_a(\bar{\alpha}) \subseteq \mathfrak{f}'_0.\chi_a(\bar{\alpha})$ (thus, $|\mathfrak{f}'_0.\chi_a(\bar{\alpha})| \geq 2$). Otherwise, if $d'_j \notin \mathfrak{f}'_0.\chi_a(\bar{\alpha})$ for every $j \in [t]$, then $\mathfrak{f}'_0.\chi_a(\bar{\alpha}) = \mathfrak{f}_2.\chi_a(\bar{\alpha})$.
  Moreover, for any $a, a' \in \mathcal{A}$, $\bar{\alpha}, \bar{\alpha}' \in \Pi$, if d'_j is in some $\mathfrak{f}'_0.\chi_a(\bar{\alpha}) \cap \mathfrak{f}'_0.\chi_{a'}(\bar{\alpha}')$ it means that $d_j \in \mathfrak{f}_2.\chi_a(\bar{\alpha}) \cap \mathfrak{f}_2.\chi_{a'}(\bar{\alpha}')$. Hence, $|\mathfrak{f}'_0.\chi_a(\bar{\alpha}) \cap \mathfrak{f}'_0.\chi_{a'}(\bar{\alpha}')| \geq 1$ iff $|\mathfrak{f}_2.\chi_a(\bar{\alpha}) \cap \mathfrak{f}_2.\chi_{a'}(\bar{\alpha}')| \geq 1$.
  As a result the fingerprint of $\mathfrak{f}_2$ and $\mathfrak{f}'_0$ are equal, and hence $\xi(\mathfrak{f}'_0) = \xi(\mathfrak{f}_2) \in \Gamma$. The same reasoning applies to $\mathfrak{f}'_{t+1}$.

Thus, $\mathfrak{f}'_0, \mathfrak{f}'_{t+1} \lessapprox (a, \bar{d})\mathfrak{f}_3$.

We now show that for every $i \in [t]$, there is $\mathfrak{f}''_i$ so that $\mathfrak{f}'_i \sim \mathfrak{f}''_i \in (a, \bar{d})\mathfrak{f}_3$. We define $\mathfrak{f}''_i$ as $\mathfrak{f}'_i[d_i \mapsto d'_i]$ for every $i \in [t]$. Observe that $\mathfrak{f}''_i \sim \mathfrak{f}'_i$ for every $i \in [t]$. Since every d_i is not rigid, $\mathfrak{f}'_i \in \mathfrak{F}$ and hence $\mathfrak{f}''_i \in \mathfrak{F}$. Further, note that $\mathfrak{f}''_i$ and $\mathfrak{f}_2$ share the same set of internal data values, and that the internal descriptions of these internal data values are the same in $\mathfrak{f}''_i$ and $\mathfrak{f}_2$. Also in this case we have that $\xi(\mathfrak{f}''_i) = \xi(\mathfrak{f}_2)$ for the same reason as before:

- For all $e \in \mathit{data}(\mathfrak{f}_2)$, $a \in \mathcal{A}$ and $\bar{\alpha} \in \Pi$ we have that $e \in \chi_a(\bar{\alpha})$ in $\mathfrak{f}_2$ iff $e \in \chi_a(\bar{\alpha})$ in $\mathfrak{f}''_i$.
- The fresh data values d'_j only add data values to sets $\chi_a(\bar{\alpha})$ that already have at least 2 data values. And $d'_j \in \mathfrak{f}'_i.\chi_a(\bar{\alpha}) \cap \mathfrak{f}'_i.\chi_{a'}(\bar{\alpha}')$ iff $d_j \in \mathfrak{f}_2.\chi_a(\bar{\alpha}) \cap \mathfrak{f}_2.\chi_{a'}(\bar{\alpha}')$.

This shows that $\xi(\mathfrak{f}'_i) = \xi(\mathfrak{f}_2) \in \Gamma$.

Thus, $\mathfrak{f}''_i \in (a, \bar{d})\mathfrak{f}_3$ since $\mathfrak{f}_2 \in (a, \bar{d})\mathfrak{f}_3$. □

We now check that $\mathfrak{f}_1 = \mathfrak{f}'_0 + \cdots + \mathfrak{f}'_{t+1}$.

Claim B.4.2. $\mathfrak{f}_1 = \mathfrak{f}'_0 + \cdots + \mathfrak{f}'_{t+1}$.

Proof. For each data value d, we are going to show that conditions (b) and (c) of concatenation hold, and that $\bar{\chi}_1(d)$ equals to the profile of d in $\mathfrak{f}'_1, + \cdots + \mathfrak{f}'_n$.

• Let us take any data value $d \in \mathit{data}(\mathfrak{f}_2) \setminus \{d_1, \ldots, d_t\}$. We first check that condition (b) holds for d, in other words, that for every $i \in \{0, \ldots, t\}$,

$$\overrightarrow{\chi}^{\mathrm{I}}_{i+1}(d) \cup [\mathfrak{f}_{i+1}\rangle \cdot \overrightarrow{\chi}^{\mathrm{E}}_{i+1}(d) = \overrightarrow{\chi}^{\mathrm{E}}_i(d).$$

We have

$$\begin{aligned}
&\overrightarrow{\chi}^{\mathrm{I}}_{i+1}(d) \cup [\mathfrak{f}_{i+1}]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{i+1}(d) \\
&= \overrightarrow{\chi}^{\mathrm{I}}_{i+1}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{i+1}(d) && \text{(by Lemma ??)} \\
&= \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{i+1}(d) && \text{(by definition)} \\
&= \begin{cases} \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2}(d) \text{ if } i = t, \text{ or} \\ \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot (\overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2}(d)) \\ \qquad\qquad\qquad\qquad \text{otherwise.} \end{cases}
\end{aligned}$$

If $i = t$, we further have that $\overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2}(d) = \overrightarrow{\chi}^{\mathrm{E}}_{i}(d)$, verifying (b). Otherwise, if $i \neq t$,

$$\begin{aligned}
\overrightarrow{\chi}^{\mathrm{I}}_{i+1}(d) \cup [\mathfrak{f}_{i+1}]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{i+1}(d) &= \cdots = \\
&= \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot (\overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2}(d)) \\
&= \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2}(d) && \text{(by } (\cdot, \cup) \text{ distributivity)} \\
&= \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2}(d) && \text{(by Lemma ??)} \\
&= [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2}(d) && \text{(since } \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \subseteq [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{I}}_{2}(d)) \\
&= \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2}(d) && \text{(by Lemma ??)} \\
&= \overrightarrow{\chi}^{\mathrm{E}}_{i}(d). && \text{(by definition)}
\end{aligned}$$

In any case, condition (b) holds between $\mathfrak{f}'_i$ and $\mathfrak{f}'_{i+1}$ for the data value d. Note that since all definitions are symmetrical, it also follows that condition (c) holds for d.

We now check that, if we call $\mathfrak{f}_+$ to $\mathfrak{f}'_1 + \cdots + \mathfrak{f}'_n$, then $\bar{\chi}_+(d) = \bar{\chi}_1(d)$. Here we only deal with $\overrightarrow{\chi}^{\mathrm{E}}_{+}$ and $\overrightarrow{\chi}^{\mathrm{I}}_{+}$ because the cases for $\overleftarrow{\chi}^{\mathrm{E}}_{+}$ and $\overleftarrow{\chi}^{\mathrm{I}}_{+}$ are symmetrical. By definition of $+$, we have that

$$\begin{aligned}
\overrightarrow{\chi}^{\mathrm{E}}_{+}(d) &= \overrightarrow{\chi}^{\mathrm{E}}_{2}(d) && \text{(by (+2))} \\
&= \overrightarrow{\chi}^{\mathrm{E}}_{1}(d), && \text{(by (??), since } d \in \mathit{data}(\mathfrak{f}_2)) \\
\overrightarrow{\chi}^{\mathrm{I}}_{+}(d) &= \big(\overrightarrow{\chi}^{\mathrm{I}}_{2} \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{I}}_{2} \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{I}}_{2} \cup \cdots \cup \\
&\qquad \underbrace{[\mathfrak{f}_2]\!\!\rightarrow\!\cdots [\mathfrak{f}_2]\!\!\rightarrow}_{n-1 \text{ times}} \cdot \overrightarrow{\chi}^{\mathrm{I}}_{2}\big)(d) && \text{(by (+4))} \\
&= ([\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{I}}_{2})(d) && \text{(by Lemma ??)} \\
&= \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) && \text{(by Lemma ??)} \\
&= \overrightarrow{\chi}^{\mathrm{I}}_{1}(d). && \text{(by (??), since } d \in \mathit{data}(\mathfrak{f}_2))
\end{aligned}$$

• Suppose now that we have $d = d_j$ for some $j \in [t]$. We check that condition (b) holds for d, in other words that for every $i \in \{0, \ldots, t\}$,

$$\overrightarrow{\chi}^{\mathrm{I}}_{i+1}(d) \cup [\mathfrak{f}_{i+1}]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{i+1}(d) = \overrightarrow{\chi}^{\mathrm{E}}_{i}(d).$$

We have

$$\overrightarrow{\chi}^{\mathrm{I}}_{i+1}(d) \cup [\mathfrak{f}_{i+1}]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{i+1}(d) = \overrightarrow{\chi}^{\mathrm{I}}_{i+1}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{i+1}(d) \qquad \text{(by Lemma ??)}$$

If $i = t$,

$$\begin{aligned}
\overrightarrow{\chi}^{\mathrm{I}}_{i+1}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{i+1}(d) &= \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2}(d) && \text{(by definition)} \\
&= \overrightarrow{\chi}^{\mathrm{E}}_{i}(d). && \text{(by definition)}
\end{aligned}$$

If $i \neq t$ and $j = i + 1$,

$$\begin{aligned}
\overrightarrow{\chi}^{\mathrm{I}}_{i+1}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{i+1}(d) &= \emptyset \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot (\overrightarrow{\chi}^{\mathrm{I}}_{2} \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2})(d) && \text{(by definition)} \\
&= [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2}(d) && \text{(by } (\cdot, \cup) \text{ distributivity)} \\
&= [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2}(d) && \text{(by Lemma ??)} \\
&= \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2}(d) && \text{(by Lemma ??)} \\
&= \overrightarrow{\chi}^{\mathrm{E}}_{i}(d). && \text{(by definition)}
\end{aligned}$$

If $i \neq t$ and $j \neq i + 1$,

$$\overrightarrow{\chi}^{\mathrm{I}}_{i+1}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{i+1}(d) = \overrightarrow{\chi}^{\mathrm{I}}_{2}(d) \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot (\overrightarrow{\chi}^{\mathrm{I}}_{2} \cup [\mathfrak{f}_2]\!\!\rightarrow\!\cdot \overrightarrow{\chi}^{\mathrm{E}}_{2})(d) \qquad \text{(by definition)}$$

$$
\begin{aligned}
&= \overrightarrow{\chi}_2^{\mathrm{I}}(d) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{I}}(d) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(d) && \text{(by } (\cdot, \cup) \text{ distributivity)} \\
&= \overrightarrow{\chi}_2^{\mathrm{I}}(d) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{I}}(d) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(d) && \text{(by Lemma ??)} \\
&= \overrightarrow{\chi}_2^{\mathrm{I}}(d) \cup \overrightarrow{\chi}_2^{\mathrm{I}}(d) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(d) && \text{(by Lemma ??)} \\
&= \overrightarrow{\chi}_2^{\mathrm{I}}(d) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(d) \\
&= \overrightarrow{\chi}_i^{\mathrm{E}}(d). && \text{(by definition)}
\end{aligned}
$$

Condition (c) follows by symmetry.

We check that $\overrightarrow{\chi}_+^{\mathrm{E}}(d) = \overrightarrow{\chi}_1^{\mathrm{E}}(d)$ and $\overrightarrow{\chi}_+^{\mathrm{I}}(d) = \overrightarrow{\chi}_1^{\mathrm{I}}(d)$. By definition of $+$, we have that

$$
\begin{aligned}
\overrightarrow{\chi}_+^{\mathrm{E}}(d) &= \overrightarrow{\chi}_2^{\mathrm{E}}(d) && \text{(by (+2))} \\
&= \overrightarrow{\chi}_1^{\mathrm{E}}(d), && \text{(by (??), since } d \in \mathit{data}(\mathfrak{f}_2)) \\
\overrightarrow{\chi}_+^{\mathrm{I}}(d) &= \big(\overrightarrow{\chi}_2^{\mathrm{I}} \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \emptyset \cup \cdots \cup \underbrace{[\mathfrak{f}_2\rangle\!\rangle \cdots [\mathfrak{f}_2\rangle\!\rangle}_{n-2 \text{ times}} \cdot \emptyset \cup \\
&\qquad \underbrace{[\mathfrak{f}_2\rangle\!\rangle \cdots [\mathfrak{f}_2\rangle\!\rangle}_{n-1 \text{ times}} \cdot \overrightarrow{\chi}_2^{\mathrm{I}}\big)(d) && \text{(by (+4))} \\
&= (\overrightarrow{\chi}_2^{\mathrm{I}} \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{I}})(d) && \text{(by Lemma ??)} \\
&= ([\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{I}})(d) && \text{(since } \overrightarrow{\chi}_2^{\mathrm{I}} \subseteq [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{I}}) \\
&= \overrightarrow{\chi}_2^{\mathrm{I}}(d) && \text{(by Lemma ??)} \\
&= \overrightarrow{\chi}_1^{\mathrm{I}}(d). && \text{(by (??), since } d \in \mathit{data}(\mathfrak{f}_2))
\end{aligned}
$$

• Finally, suppose $d = d'_j$ for some $j \in [t]$. We check that condition (b) holds for d, in other words that for every $i \in \{0, \ldots, t\}$,

$$
\overrightarrow{\chi}_{i+1}^{\mathrm{I}}(d'_j) \cup [\mathfrak{f}_{i+1}\rangle\!\rangle \cdot \overrightarrow{\chi}_{i+1}^{\mathrm{E}}(d'_j) = \overrightarrow{\chi}_i^{\mathrm{E}}(d'_j).
$$

By definition, we have

$$
\overrightarrow{\chi}_{i+1}^{\mathrm{I}}(d'_j) \cup [\mathfrak{f}_{i+1}\rangle\!\rangle \cdot \overrightarrow{\chi}_{i+1}^{\mathrm{E}}(d'_j) = \overrightarrow{\chi}_{i+1}^{\mathrm{I}}(d'_j) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_{i+1}^{\mathrm{E}}(d'_j) \qquad \text{(by Lemma ??)}
$$

If $i = t$,

$$
\begin{aligned}
\overrightarrow{\chi}_{i+1}^{\mathrm{I}}(d'_j) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_{i+1}^{\mathrm{E}}(d'_j) &= \emptyset \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(d_j) && \text{(by definition)} \\
&= \overrightarrow{\chi}_i^{\mathrm{E}}(d'_j). && \text{(by definition)}
\end{aligned}
$$

If $i \neq t$ and $j = i + 1$,

$$
\begin{aligned}
\overrightarrow{\chi}_{i+1}^{\mathrm{I}}(d'_j) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_{i+1}^{\mathrm{E}}(d'_j) &= \overrightarrow{\chi}_2^{\mathrm{I}}(d_j) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(d_j) && \text{(by definition)} \\
&= \overrightarrow{\chi}_2^{\mathrm{I}}(d_j) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(d_j) && \text{(by Lemma ??)} \\
&= \overrightarrow{\chi}_i^{\mathrm{E}}(d'_j). && \text{(by definition)}
\end{aligned}
$$

If $i \neq t$ and $j > i + 1$,

$$
\begin{aligned}
\overrightarrow{\chi}_{i+1}^{\mathrm{I}}(d'_j) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_{i+1}^{\mathrm{E}}(d'_j) &= \emptyset \cup [\mathfrak{f}_2\rangle\!\rangle \cdot (\overrightarrow{\chi}_2^{\mathrm{I}}(d_j) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(d_j)) && \text{(by definition)} \\
&= [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{I}}(d_j) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(d_j) && \text{(by } (\cdot, \cup) \text{ distributivity)} \\
&= [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{I}}(d_j) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(d_j) && \text{(by Lemma ??)} \\
&= \overrightarrow{\chi}_2^{\mathrm{I}}(d_j) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(d_j) && \text{(by Lemma ??)} \\
&= \overrightarrow{\chi}_i^{\mathrm{E}}(d'_j). && \text{(by definition)}
\end{aligned}
$$

If $i \neq t$ and $j < i + 1$,

$$
\begin{aligned}
\overrightarrow{\chi}_{i+1}^{\mathrm{I}}(d'_j) \cup [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_{i+1}^{\mathrm{E}}(d'_j) &= \emptyset \cup [\mathfrak{f}_2\rangle\!\rangle \cdot [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(d_j) && \text{(by definition)} \\
&= [\mathfrak{f}_2\rangle\!\rangle \cdot \overrightarrow{\chi}_2^{\mathrm{E}}(d_j) && \text{(by Lemma ??)} \\
&= \overrightarrow{\chi}_i^{\mathrm{E}}(d'_j). && \text{(by definition)}
\end{aligned}
$$

Condition (c) follows by symmetry.

We check that $\overrightarrow{\chi}_+^{\mathrm{E}}(d'_j) = \overrightarrow{\chi}_1^{\mathrm{E}}(d'_j)$ and $\overrightarrow{\chi}_+^{\mathrm{I}}(d'_j) = \overrightarrow{\chi}_1^{\mathrm{I}}(d'_j)$. By definition of $+$, we have that

$$
\begin{aligned}
\overrightarrow{\chi}_+^{\mathrm{E}}(d'_j) &= \overrightarrow{\chi}_2^{\mathrm{E}}(d_j) && \text{(by (+2))} \\
&= \overrightarrow{\chi}_1^{\mathrm{E}}(d'_j), && \text{(since } \bar{\chi}_1(d'_j) = \bar{\chi}_2(d_j) \text{ by (??))}
\end{aligned}
$$

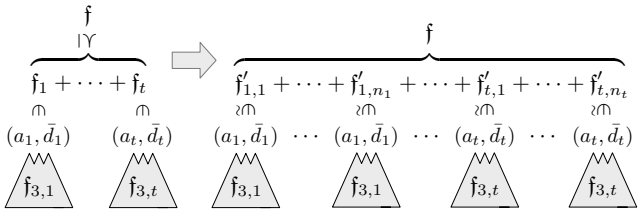


Figure 8: Statement of Lemma ??.

$$\begin{aligned}
\overrightarrow{\chi}^{\mathfrak{f}_1}_{+}(d'_j) = \big(\emptyset \cup [\mathfrak{f}_2\rangle\cdot\emptyset \cup \cdots \cup \underbrace{[\mathfrak{f}_2\rangle\cdots[\mathfrak{f}_2\rangle}_{j \text{ times}}\cdot\emptyset \ \cup \\
\underbrace{[\mathfrak{f}_2\rangle\cdots[\mathfrak{f}_2\rangle}_{j+1 \text{ times}}\cdot\overrightarrow{\chi}^{\mathfrak{f}_1}_{2} \ \cup \\
\underbrace{[\mathfrak{f}_2\rangle\cdots[\mathfrak{f}_2\rangle}_{j+2 \text{ times}}\cdot\emptyset \cup \cdots \cup \underbrace{[\mathfrak{f}_2\rangle\cdots[\mathfrak{f}_2\rangle}_{n-1 \text{ times}}\cdot\emptyset\big)(d_j) && \text{(by (+4))} \\
= ([\mathfrak{f}_2\rangle\cdot\overrightarrow{\chi}^{\mathfrak{f}_1}_{2})(d_j) && \text{(by Lemma ??)} \\
= \overrightarrow{\chi}^{\mathfrak{f}_1}_{2}(d_j) && \text{(by Lemma ??)} \\
= \overrightarrow{\chi}^{\mathfrak{f}_1}_{1}(d'_j). && \text{(since } \bar{\chi}_1(d'_j) = \bar{\chi}_2(d_j) \text{ by (??))}
\end{aligned}$$

Therefore, we have that condition (a) as well as conditions (b) and (c) hold true for every data value, and that $\mathfrak{f}'_0 + \cdots + \mathfrak{f}'_{t+1}$ and $\mathfrak{f}_1$ coincide. Thus, the Claim follows. □

This concludes our proof.

As a corollary of the previous lemma, we obtain the following.

Lemma B.5 (Figure ??). *For every $\mathfrak{f} \succeq \mathfrak{f}_1 + \cdots + \mathfrak{f}_t$ with $\mathfrak{f}_i \in (a_i, \bar{d}_i)\mathfrak{f}_{3,i}$, there are $\mathfrak{f}'_{i,j} \Subset (a_i, \bar{d}_i)\mathfrak{f}_{3,i}$ for every $i \in [t]$, $j \in [n_i]$ so that*

$$\mathfrak{f} = \mathfrak{f}'_{1,1} + \cdots + \mathfrak{f}'_{1,n_1} + \cdots + \mathfrak{f}'_{t,1} + \cdots + \mathfrak{f}'_{t,n_t}.$$

Proof of Lemma ??. This is a direct consequence of Lemma 5.1. Since $\mathfrak{f} \succeq \mathfrak{f}_1 + \cdots + \mathfrak{f}_t$, then

$$\mathfrak{f} = (\mathfrak{f}_1 + \cdots + \mathfrak{f}_t)[d_1 \mapsto d_1, d'_1] \cdots [d_n \mapsto d_n, d'_n]$$

for some $d_1, \ldots, d_n, d'_1, \ldots, d'_n \in \mathbb{D}$ so that $\{d_1, \ldots, d_n\} \cap R = \emptyset$ and $\{d'_1, \ldots, d'_n\} \cap \mathit{data}(\mathfrak{f}_1 + \cdots + \mathfrak{f}_t) = \emptyset$. It follows that

$$\begin{aligned}
\mathfrak{f} &= (\mathfrak{f}_1 + \cdots + \mathfrak{f}_t)[d_1 \mapsto d_1, d'_1] \cdots [d_n \mapsto d_n, d'_n] \\
&= \mathfrak{f}_1[d_1 \mapsto d_1, d'_1] \cdots [d_n \mapsto d_n, d'_n] + \cdots + \\
&\quad \mathfrak{f}_t[d_1 \mapsto d_1, d'_1] \cdots [d_n \mapsto d_n, d'_n].
\end{aligned}$$

Note that $\mathfrak{f}_i[d_1 \mapsto d_1, d'_1] \cdots [d_n \mapsto d_n, d'_n] \succeq \mathfrak{f}_i \in (a_i, \bar{d}_i)\mathfrak{f}_{3,i}$ for every $i \in [t]$. We can hence apply Lemma 5.1, and we obtain, for some $n_i \in \mathbb{N}$, $\mathfrak{f}'_{i,1}, \ldots, \mathfrak{f}'_{i,n_i} \Subset (a_i, \bar{d}_i)\mathfrak{f}_{3,i}$ so that $\mathfrak{f}'_{i,1} + \cdots + \mathfrak{f}'_{i,n_i} = \mathfrak{f}_i[d_1 \mapsto d_1, d'_1] \cdots [d_n \mapsto d_n, d'_n]$. Therefore,

$$\begin{aligned}
\mathfrak{f} &= \mathfrak{f}_1[d_1 \mapsto d_1, d'_1] \cdots [d_n \mapsto d_n, d'_n] + \cdots + \\
&\quad \mathfrak{f}_t[d_1 \mapsto d_1, d'_1] \cdots [d_n \mapsto d_n, d'_n] \\
&= \mathfrak{f}'_{1,1} + \cdots + \mathfrak{f}'_{1,n_1} + \cdots + \mathfrak{f}'_{t,1} + \cdots + \mathfrak{f}'_{t,n_t},
\end{aligned}$$

which concludes the proof. □

Lemma 5.2. $\mathfrak{D} = \uparrow\mathfrak{D}$.

Proof of Lemma 5.2. This is a direct consequence of Lemma ??. □

Lemma 5.3 (Figure 5). *If $\mathfrak{f}_1 + \mathfrak{f}_2 = \mathfrak{f}_3$ and $\mathfrak{f}'_1 \preceq \mathfrak{f}_1$, $\mathfrak{f}'_2 \preceq \mathfrak{f}_2$, then $\mathfrak{f}''_1 + \mathfrak{f}''_2 \preceq \mathfrak{f}_3$, for some $\mathfrak{f}''_1, \mathfrak{f}''_2 \in \mathfrak{F}$ so that $\mathfrak{f}''_i$ is a bounded extension of $\mathfrak{f}'_i$, for all $i \in \{1, 2\}$.*

Proof of Lemma 5.3. For every description $\bar{\pi}$ so that $\bar{\chi}_3(\bar{\pi}) \setminus R_3 \neq \emptyset$, let $d_{\bar{\pi}} \in \mathbb{D}$ be a data value so that $d_{\bar{\pi}} \in \bar{\chi}_3(\bar{\pi}) \setminus R_3$. Let D be the set of all these data values. Note that $|D| \leq |\Pi|^4$.

We define $\mathfrak{f}''_i = (R_3, \bar{\chi}''_i)$, where

$$\bar{\chi}''_i(d) = \begin{cases} \bar{\chi}'_i(d) & \text{if } d \notin D \\ \bar{\chi}_i(d) & \text{if } d \in D. \end{cases}$$

Note that $data(\mathfrak{f}_i'') \leq data(\mathfrak{f}_i') + |D| \leq data(\mathfrak{f}_i') + |\Pi|^4$.

CLAIM B.5.1. *$\mathfrak{f}_i' \preceq \mathfrak{f}_i'' \preceq \mathfrak{f}_i$ for all $i \in \{1,2\}$.*

PROOF. By definition of $\preceq$, for every $d \in D$ there is some $d' \in data(\mathfrak{f}_i') \setminus R_i$ so that $\bar{\chi}_i(d) = \bar{\chi}_i'(d')$. Therefore, $\mathfrak{f}_i'' = \mathfrak{f}_i'[d_1' \mapsto d_1', d_1] \cdots [d_1' \mapsto d_n', d_n]$ where $\{d_1, \ldots, d_n\} = D \setminus data(\mathfrak{f}_i')$ and for every $j \in [n]$, $d_j' \notin R_i$, $d_j \notin data(\mathfrak{f}_i')$, and $\bar{\chi}_i'(d_j') = \bar{\chi}_i(d_j)$. In other words, we have that $\mathfrak{f}_i' \preceq \mathfrak{f}_i'' \preceq \mathfrak{f}_i$ for all $i \in \{1,2\}$. $\square$

Therefore, $\mathfrak{f}_i''$ is a bounded extension of $\mathfrak{f}_i'$ for $i \in \{1,2\}$.

CLAIM B.5.2. *$\mathfrak{f}_1'' + \mathfrak{f}_2'' \preceq \mathfrak{f}_3$.*

PROOF. First, we have that $\mathfrak{f}_1'', \mathfrak{f}_2'' \in \mathfrak{F}$ because otherwise $\mathfrak{f}_1$ or $\mathfrak{f}_2$ would not be in $\mathfrak{F}$.

Since $\mathfrak{f}_i'' \preceq \mathfrak{f}_i$, by Lemma **??** it follows that

$$\langle\!|[\mathfrak{f}_i''] = \langle\!|[\mathfrak{f}_i], \quad [\mathfrak{f}_i''|\!\rangle = [\mathfrak{f}_i|\!\rangle. \tag{$\dagger$}$$

This means that, since $\mathfrak{f}_i''$ is the result of adding data values d with profile $\bar{\chi}_i(d)$, if conditions (a), (b), (c) hold for $\mathfrak{f}_1 + \mathfrak{f}_2$ they must also hold for $\mathfrak{f}_1'' + \mathfrak{f}_2''$. Therefore, all the preconditions to apply $\mathfrak{f}_1'' + \mathfrak{f}_2''$ hold. Let $\mathfrak{f}_3'' = \mathfrak{f}_1'' + \mathfrak{f}_2''$.

Note that all the profiles in question share the same set of rigid values, $R_3'' = R_1'' = R_2'' = R_1 = R_2 = R_3$ by definition of $\preceq$ and $+$. Every data value $d \in data(\mathfrak{f}_3'')$ is so that $\bar{\chi}_i''(d) = \bar{\chi}_i(d)$, and with (**??**) this means that $\bar{\chi}_3(d) = \bar{\chi}_3''(d)$, by definition of $\mathfrak{f}_1'' + \mathfrak{f}_2''$. For every other data value $d \in data(\mathfrak{f}_3) \setminus data(\mathfrak{f}_3'')$, it must be that $d \notin R_3$ and there must be some $\hat{d} \in D$ so that $\bar{\chi}_3(\hat{d}) = \bar{\chi}_3(d)$ by definition of D. Then, if $data(\mathfrak{f}_3) \setminus data(\mathfrak{f}_3'') = \{d_1, \ldots, d_n\}$ we have that

$$\mathfrak{f}_3 \;=\; \mathfrak{f}_3''[\hat{d}_1 \mapsto \hat{d}_1, d_1] \cdots [\hat{d}_n \mapsto \hat{d}_n, d_n],$$

and hence, $\mathfrak{f}_3'' \preceq \mathfrak{f}_3$. $\square$

By Claims **??** and **??**, and since $data(\mathfrak{f}_i'') \leq data(\mathfrak{f}_i') + |\Pi|^4$, the lemma follows. $\square$

LEMMA 5.4 (FIGURE 6). *For every $\mathfrak{f}_1 \in (a, \bar{d})\mathfrak{f}_2$ and $\mathfrak{f}_2 \succeq \mathfrak{f}_5$, there is $\mathfrak{f}_4 \succeq \mathfrak{f}_5$ and $\mathfrak{f}_3 \Subset (a, \bar{d})\mathfrak{f}_4$ so that $|data(\mathfrak{f}_4)| \leq |data(\mathfrak{f}_5)| + |\Pi|^4 + |R_1|$ and $\mathfrak{f}_3 \preceq \mathfrak{f}_1$, $\mathfrak{f}_4 \preceq \mathfrak{f}_2$.*

PROOF OF LEMMA 5.4. We are given $\mathfrak{f}_1, \mathfrak{f}_2, \mathfrak{f}_5$ and $(a, \bar{d})$. We show that there must be some $\mathfrak{f}_3$ and $\mathfrak{f}_4$ so that $\mathfrak{f}_1 \succeq \mathfrak{f}_3 \in (a, \bar{d})\mathfrak{f}_4$ with $\mathfrak{f}_4 \succeq \mathfrak{f}_5$. Figure 6 contains a graphical representation of the profiles we work with in the proof.

We first define $\mathfrak{f}_4$. For every internal description $\bar{\pi}$ so that $\overrightarrow{\pi}_1^{\mathrm{I}} \cup \overleftarrow{\pi}_1^{\mathrm{I}} \neq \emptyset$ and $\bar{\chi}_1(\bar{\pi}) \setminus R_1 \neq \emptyset$, let $d_{\bar{\pi}} \in \mathbb{D}$ be a data value of $\bar{\chi}_1(\bar{\pi}) \setminus R_1$. Let D be the set of all such data values. Note that $|D| \leq |\Pi|^4$.

$$\mathfrak{f}_4 : \begin{cases} R_4 = R_5 \\ \text{for every } d \in D \cup R_1,\ \bar{\chi}_4(d) = \bar{\chi}_2(d) \\ \text{for every } d \notin D \cup R_1,\ \bar{\chi}_4(d) = \bar{\chi}_5(d) \end{cases}$$

Note that $|data(\mathfrak{f}_4)| \leq |data(\mathfrak{f}_5)| + |\Pi|^4 + |R_1|$. Since $\mathfrak{f}_5 \preceq \mathfrak{f}_2$, we have that $\mathfrak{f}_5 \preceq \mathfrak{f}_4 \preceq \mathfrak{f}_2$.

CLAIM B.5.3. *$\mathfrak{f}_5 \preceq \mathfrak{f}_4 \preceq \mathfrak{f}_2$.*

PROOF. • We first show that $\mathfrak{f}_5 \preceq \mathfrak{f}_4$.

First notice that $R_5 = R_2$ since $\mathfrak{f}_5 \preceq \mathfrak{f}_2$, and that $R_4 = R_5$ by definition of R_4. We then have that for every $d \in R_5 = R_4$, $\bar{\chi}_4(d) = \bar{\chi}_2(d) = \bar{\chi}_5(d)$.

We show that for every $d \in data(\mathfrak{f}_5)$, $\bar{\chi}_4(d) = \bar{\chi}_5(d)$. If $d \in data(\mathfrak{f}_5) \setminus (D \cup R_1)$, we have that $\bar{\chi}_4(d) = \bar{\chi}_5(d)$. If on the other hand $d \in D \cup R_1$, we have $\bar{\chi}_4(d) = \bar{\chi}_2(d)$. Then $d \in data(\mathfrak{f}_2)$, and since $\mathfrak{f}_5 \preceq \mathfrak{f}_2$ we have that either $d \notin data(\mathfrak{f}_5)$ or $\bar{\chi}_2(d) = \bar{\chi}_5(d)$. Hence, for every $d \in data(\mathfrak{f}_5)$, $\bar{\chi}_4(d) = \bar{\chi}_5(d)$.

We finish the proof of $\mathfrak{f}_4 \preceq \mathfrak{f}_5$ by showing that for every $d \in data(\mathfrak{f}_4) \setminus R_4$ there is some $d' \in data(\mathfrak{f}_5) \setminus R_4$ so that $\bar{\chi}_4(d) = \bar{\chi}_5(d')$. If $d \notin D \cup R_1$, then of course we can take $d' = d$ and $\bar{\chi}_4(d) = \bar{\chi}_5(d')$. If $d \in D \cup R_1$, we have that $\bar{\chi}_4(d) = \bar{\chi}_2(d)$ by definition. Since $\mathfrak{f}_5 \preceq \mathfrak{f}_2$, there must be some $d' \in data(\mathfrak{f}_5) \setminus R_5$ so that $\bar{\chi}_5(d') = \bar{\chi}_2(d) = \bar{\chi}_5(d)$, and we are done. This finishes the proof that $\mathfrak{f}_5 \preceq \mathfrak{f}_4$.

• We now show that $\mathfrak{f}_4 \preceq \mathfrak{f}_2$.

Again, notice that $R_4 = R_2 = R_5$, and since for every $d \in D$, $\bar{\chi}_4(d)$ is either $\bar{\chi}_2(d)$ or $\bar{\chi}_5(d)$, it follows that for every $d \in R_4 = R_5$, $\bar{\chi}_4(d) = \bar{\chi}_2(d)$.

We show that for every $d \in data(\mathfrak{f}_4)$, $\bar{\chi}_4(d) = \bar{\chi}_2(d)$. If $d \in D \cup R_1$, then $\bar{\chi}_4(d) = \bar{\chi}_2(d)$ by definition. If $d \in R_4$, then $\bar{\chi}_4(d) = \bar{\chi}_2(d)$ as shown before. If $d \notin D \cup R_1 \cup R_4$, $\bar{\chi}_4(d) = \bar{\chi}_5(d)$. Notice that $\bar{\chi}_5(d) \neq (\emptyset, \emptyset, \emptyset, \emptyset)$, since otherwise $d \notin data(\mathfrak{f}_4)$. Then, $d \in data(\mathfrak{f}_5)$ and since $\mathfrak{f}_5 \preceq \mathfrak{f}_2$, $\bar{\chi}_5(d) = \bar{\chi}_2(d) = \bar{\chi}_4(d)$.

We finish the proof of $\mathfrak{f}_4 \preceq \mathfrak{f}_2$ by showing that for every $d \in data(\mathfrak{f}_2) \setminus R_2$ there is some $d' \in data(\mathfrak{f}_4) \setminus R_2$ so that $\bar{\chi}_2(d) = \bar{\chi}_4(d')$. If $d \in D \cup R_1$, then we can just simply take $d' = d$ and $\bar{\chi}_2(d) = \bar{\chi}_4(d')$. If $d \notin D \cup R_1$, then $\bar{\chi}_4(d) = \bar{\chi}_5(d)$, notice that $\bar{\chi}_5(d) \neq (\emptyset, \emptyset, \emptyset, \emptyset)$ because otherwise $d \notin data(\mathfrak{f}_4)$ (since $d \notin R_4$). Hence, $d \in data(\mathfrak{f}_5)$ and since $\mathfrak{f}_5 \preceq \mathfrak{f}_2$, we have that $\bar{\chi}_2(d) = \bar{\chi}_5(d)$, hence $\bar{\chi}_2(d) = \bar{\chi}_4(d)$. We just showed that $\mathfrak{f}_4 \preceq \mathfrak{f}_2$. $\square$

We define $\mathfrak{f}_3$ as follows

$$\mathfrak{f}_3 : \begin{cases} R_3 = R_1 \\ \text{for every } d \in R_1,\ \bar{\chi}_3(d) = \bar{\chi}_1(d) \\ \text{for every } d \in \mathit{data}(\mathfrak{f}_4),\ \bar{\chi}_3(d) = \bar{\chi}_1(d) \\ \text{for every } d \in \bar{d},\ \bar{\chi}_3(d) = \bar{\chi}_1(d) \\ \text{for every strict external value } d \text{ of } \mathfrak{f}_1,\ \bar{\chi}_3(d) = \bar{\chi}_1(d). \end{cases}$$

Claim B.5.4. $\mathfrak{f}_3 \preceq \mathfrak{f}_1$.

Proof. By definition we have that $R_1 = R_3$, and that $\bar{\chi}_3(d) = \bar{\chi}_1(d)$ for all $d \in R_1$. Also by definition, for every $d \in \mathit{data}(\mathfrak{f}_3)$ we have $\bar{\chi}_3(d) = \bar{\chi}_1(d)$.

We must show that for every $d \in \mathit{data}(\mathfrak{f}_1) \setminus (R_1 \cup \mathit{data}(\mathfrak{f}_3))$ there is some $d' \in \mathit{data}(\mathfrak{f}_3) \setminus R_1$ so that $\bar{\chi}_1(d) = \bar{\chi}_3(d')$. Take any such d. Note that d must be necessarily an internal data value, since $\mathfrak{f}_3$ contains any strict external data value of $\mathfrak{f}_1$. Hence, let $\bar{\pi} = \bar{\chi}_1(d)$, where $\overrightarrow{\pi}_1^{\mathrm{I}} \cup \overleftarrow{\pi}_1^{\mathrm{I}} \neq \emptyset$. By definition of D, there must be some $d_{\bar{\pi}} \in D$ so that $\bar{\chi}_1(d_{\bar{\pi}}) = \bar{\pi}$ and $d_{\bar{\pi}} \notin R_1$. Since $\bar{\pi}$ is internal and $\mathfrak{f}_1 \in (a, \bar{d})\mathfrak{f}_2$, $d_{\bar{\pi}} \in \bar{d} \cup \mathit{data}(\mathfrak{f}_2)$. If $d_{\bar{\pi}} \in \bar{d}$, we have, by definition of $\mathfrak{f}_3$, that $\bar{\chi}_3(d_{\bar{\pi}}) = \bar{\chi}_1(d_{\bar{\pi}}) = \bar{\pi} = \bar{\chi}_1(d)$. Hence, there is such $d' \in \mathit{data}(\mathfrak{f}_3) \setminus R_1$ so that $\bar{\chi}_1(d) = \bar{\chi}_3(d')$. Otherwise, suppose $d_{\bar{\pi}} \in \mathit{data}(\mathfrak{f}_2)$, which means, since $\overrightarrow{\pi}_1^{\mathrm{I}} \cup \overleftarrow{\pi}_1^{\mathrm{I}} \neq \emptyset$, that $\bar{\chi}_2(d_{\bar{\pi}}) \neq (\emptyset, \emptyset, \emptyset, \emptyset)$. By definition of $\mathfrak{f}_4$, $\bar{\chi}_4(d_{\bar{\pi}}) = \bar{\chi}_2(d_{\bar{\pi}}) \neq (\emptyset, \emptyset, \emptyset, \emptyset)$, thus $d_{\bar{\pi}} \in \mathit{data}(\mathfrak{f}_4)$. Then, by definition of $\mathfrak{f}_3$ we have that $\bar{\chi}_3(d_{\bar{\pi}}) = \bar{\chi}_1(d_{\bar{\pi}}) = \bar{\pi} = \bar{\chi}_1(d)$. Hence, in this case we also have that there is such $d' \in \mathit{data}(\mathfrak{f}_3) \setminus R_1$ so that $\bar{\chi}_1(d) = \bar{\chi}_3(d')$. □

Claim B.5.5. $\mathfrak{f}_3 \in (a, \bar{d})\mathfrak{f}_4$.

Proof. Since $\mathfrak{f}_4 \preceq \mathfrak{f}_2$ and $\mathfrak{f}_1 \in (a, \bar{d})\mathfrak{f}_2$, we have that

- $\xi(\mathfrak{f}_4) \in \Gamma$ since $\xi(\mathfrak{f}_2) \in \Gamma$, by Lemma **??**, and
- $\overrightarrow{\chi}_4^{\mathrm{E}} = \overleftarrow{\chi}_4^{\mathrm{E}} = \emptyset$ since $\overrightarrow{\chi}_2^{\mathrm{E}} = \overleftarrow{\chi}_2^{\mathrm{E}} = \emptyset$.

Therefore, conditions (a) and (b) hold for $\mathfrak{f}_4$. We must show condition (c), that is, $\overrightarrow{\chi}_3^{\mathrm{I}} = \overleftarrow{\chi}_3^{\mathrm{I}} = \{(d, \alpha, \beta\gamma, i) \in \mathbb{D} \times \Pi \mid \exists \alpha'.(d, \alpha', \gamma, i) \in \overrightarrow{\chi}_4^{\mathrm{I}} \cup \overleftarrow{\chi}_4^{\mathrm{I}}, \alpha, \beta \in \boldsymbol{\sigma}_a\} \cup \bigcup_{i \in [\mathbf{k}]} (\{\bar{d}(i)\} \times (\boldsymbol{\sigma}_a \setminus \{\epsilon\}) \times \boldsymbol{\sigma}_a \times \{i\})$

Take any data value of $d \in \bar{d} \cup \mathit{data}(\mathfrak{f}_4)$. We have that $\overrightarrow{\chi}_3^{\mathrm{I}}(d) = \overrightarrow{\chi}_1^{\mathrm{I}}(d)$ by definition. We also have that $\overleftarrow{\chi}_1^{\mathrm{I}}(d) = \overrightarrow{\chi}_1^{\mathrm{I}}(d) = \{(\alpha, \beta\gamma, i) \in \Pi \mid \exists \alpha'.(d, \alpha', \gamma, i) \in \overrightarrow{\chi}_2^{\mathrm{I}} \cup \overleftarrow{\chi}_2^{\mathrm{I}}, \alpha, \beta \in \boldsymbol{\sigma}_a\} \cup \bigcup_{d = \bar{d}(i), i \in [\mathbf{k}]} ((\boldsymbol{\sigma}_a \setminus \{\epsilon\}) \times \boldsymbol{\sigma}_a \times \{i\})$. Since $\mathfrak{f}_2$ and $\mathfrak{f}_4$ coincide in d, we also have that the set above is equal to $\{(\alpha, \beta\gamma, i) \in \Pi \mid \exists \alpha'.(d, \alpha', \gamma, i) \in \overrightarrow{\chi}_4^{\mathrm{I}} \cup \overleftarrow{\chi}_4^{\mathrm{I}}, \alpha, \beta \in \boldsymbol{\sigma}_a\} \cup \bigcup_{d = \bar{d}(i), i \in [\mathbf{k}]} ((\boldsymbol{\sigma}_a \setminus \{\epsilon\}) \times \boldsymbol{\sigma}_a \times \{i\})$. This, together with the fact that all internal values of $\mathfrak{f}_3$ are in $\mathit{data}(\mathfrak{f}_4) \cup \bar{d}$, implies that $\overrightarrow{\chi}_3^{\mathrm{I}} = \overleftarrow{\chi}_3^{\mathrm{I}} = \{(d, \alpha, \beta\gamma, i) \in \mathbb{D} \times \Pi \mid \exists \alpha'.(d, \alpha', \gamma, i) \in \overrightarrow{\chi}_4^{\mathrm{I}} \cup \overleftarrow{\chi}_4^{\mathrm{I}}, \alpha, \beta \in \boldsymbol{\sigma}_a\} \cup \bigcup_{i \in [\mathbf{k}]} (\{\bar{d}(i)\} \times (\boldsymbol{\sigma}_a \setminus \{\epsilon\}) \times \boldsymbol{\sigma}_a \times \{i\})$ as desired. Hence, condition (c) holds, and we have that $\mathfrak{f}_3 \in (a, \bar{d})\mathfrak{f}_4$. □

This concludes the proof. □

Appendix to Section 5.4

Lemma 5.5. *If there is a derivable root profile, then there is a derivation tree for a root profile so that all the profiles in the forest have no more than* $2|\Pi|$ *rigid values.*

Proof of Lemma 5.5. Suppose we have $\mathfrak{f} = \mathfrak{f}_1 + \cdots + \mathfrak{f}_n$. Let us first show that for every $(\alpha, \beta, i) \in \Pi$ there can be at most one data value $d \in \mathbb{D}$ so that there is some $j \in [n]$ with $\overleftarrow{\chi}_j^{\mathrm{E}}(\alpha, \beta, i) = \{d\}$. By means of contradiction, if there were two distinct data values d, d' then there would be two $\mathfrak{f}_j, \mathfrak{f}_k$ with $j \neq k$ so that $\overleftarrow{\chi}_j^{\mathrm{E}}(\alpha, \beta, i) = \{d\}$ and $\overleftarrow{\chi}_k^{\mathrm{E}}(\alpha, \beta, i) = \{d'\}$. Suppose without any loss of generality that $j < k$. Then, by definition of $+$, we must have that—since $d \in \overleftarrow{\chi}_j^{\mathrm{E}}(\alpha, \beta, i)$ and $\mathfrak{f}_j + \cdots + \mathfrak{f}_k$ is defined— $d \in \overleftarrow{\chi}_k^{\mathrm{E}}(\alpha, \beta, i)$. This is in contradiction with the fact that $\overleftarrow{\chi}_k^{\mathrm{E}}(\alpha, \beta, i) = \{d'\}$. The same happens by symmetry with $\overrightarrow{\chi}^{\mathrm{E}}$.

Let us define

$$\begin{aligned} R^l_{\bar{\chi}_1, \ldots, \bar{\chi}_n} &\stackrel{def}{=} \{d \in \mathbb{D} \mid \overleftarrow{\chi}_j^{\mathrm{E}}(\alpha, \beta, i) = \{d\} \\ &\qquad \text{for some } (\alpha, \beta, i) \in \Pi \text{ and } j \in [n]\}, \\ R^r_{\bar{\chi}_1, \ldots, \bar{\chi}_n} &\stackrel{def}{=} \{d \in \mathbb{D} \mid \overrightarrow{\chi}_j^{\mathrm{E}}(\alpha, \beta, i) = \{d\} \\ &\qquad \text{for some } (\alpha, \beta, i) \in \Pi \text{ and } j \in [n]\}, \\ R_{\bar{\chi}_1, \ldots, \bar{\chi}_n} &\stackrel{def}{=} R^l_{\bar{\chi}_1, \ldots, \bar{\chi}_n} \cup R^r_{\bar{\chi}_1, \ldots, \bar{\chi}_n}. \end{aligned}$$

By the discussion before, it follows that $|R_{\bar{\chi}_1, \ldots, \bar{\chi}_n}| \leq 2|\Pi|$. Consider the profiles $\mathfrak{f}', \mathfrak{f}'_1, \ldots, \mathfrak{f}'_n$ to be as $\mathfrak{f}, \mathfrak{f}_1, \ldots, \mathfrak{f}_n$ but with $R_{\bar{\chi}_1, \ldots, \bar{\chi}_n}$ as the set of rigid values. By construction of $R_{\bar{\chi}_1, \ldots, \bar{\chi}_n}$ we have that $\mathfrak{f}' = \mathfrak{f}'_1 + \cdots + \mathfrak{f}'_n$. In other words we have the following.

Claim B.5.6. *For every* $(R, \bar{\chi}) = (R, \bar{\chi}_1) + \cdots + (R, \bar{\chi}_n)$ *we have that*

$$(R_{\bar{\chi}_1, \ldots, \bar{\chi}_n}, \bar{\chi}) = (R_{\bar{\chi}_1, \ldots, \bar{\chi}_n}, \bar{\chi}_1) + \cdots + (R_{\bar{\chi}_1, \ldots, \bar{\chi}_n}, \bar{\chi}_n).$$

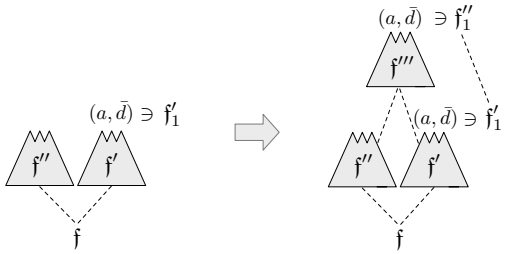


Figure 9: Statement of Lemma ??.

Using the above statement we can now prove the lemma. Let t be a derivation tree for a root profile $(R, \mathfrak{f})$. Let t' be the derivation tree that results from replacing in t the profile labels $(R, \bar{\chi}_1), \ldots, (R, \bar{\chi}_n)$ of any maximal sequence of siblings with the labels

$$(R_{\bar{\chi}_1,\ldots,\bar{\chi}_n}, \bar{\chi}_1), \ldots, (R_{\bar{\chi}_1,\ldots,\bar{\chi}_n}, \bar{\chi}_n).$$

By the claim above, t' is a derivation tree for $(R_{\bar{\chi}}, \mathfrak{f})$. (Note that this is true because in the rooting operation there is no restriction on which should be the set of rigid values of the parent profile.) It is immediate that $(R, \mathfrak{f})$ is a root profile if and only if $(R_{\bar{\chi}}, \mathfrak{f})$ is a root profile. Thus, the lemma follows. □

Appendix to Section 5.5

Lemma 5.9 is the consequence of the following properties.

LEMMA B.6 (FIGURE **??**). *For every* $\mathfrak{f}, \mathfrak{f}', \mathfrak{f}'', \mathfrak{f}_1' \in \mathfrak{F}_b$ *and* $(a, \bar{d}) \in \mathbb{A} \times \mathbb{D}^{\mathbf{k}}$ *so that* $\mathfrak{f} \precsim \mathfrak{f}''$, $\mathfrak{f} \precsim \mathfrak{f}'$ *and* $\mathfrak{f}_1' \in (a, \bar{d})\mathfrak{f}'$, *there are* $\mathfrak{f}''', \mathfrak{f}_1'' \in \mathfrak{F}_b$ *so that* $\mathfrak{f}'' \precsim \mathfrak{f}'''$, $\mathfrak{f}' \precsim \mathfrak{f}'''$, $\mathfrak{f}_1'' \in (a, \bar{d})\mathfrak{f}'''$ *and* $\mathfrak{f}_1' \precsim \mathfrak{f}_1''$.

PROOF OF LEMMA **??**. Without any loss of generality, let us assume that $\mathfrak{f} \preceq \mathfrak{f}'$ and $\mathfrak{f} \preceq \mathfrak{f}''$ so that $data(\mathfrak{f}'') \cap data(\mathfrak{f}') = data(\mathfrak{f}'') \cap data(\mathfrak{f}_1') = data(\mathfrak{f})$. We define $\mathfrak{f}'''$ so that $R''' = R$ and

$$\bar{\chi}'''(d) = \begin{cases} \bar{\chi}'(d) & \text{if } d \in data(\mathfrak{f}'), \\ \bar{\chi}''(d) & \text{else, if } d \in data(\mathfrak{f}''). \end{cases}$$

The following statement follows straight from this definition.

CLAIM B.6.1. $\mathfrak{f}'' \preceq \mathfrak{f}'''$, $\mathfrak{f}' \preceq \mathfrak{f}'''$.

PROOF. By definition of $\mathfrak{f} \preceq \mathfrak{f}''$ and $\mathfrak{f} \preceq \mathfrak{f}'$ for every data value $d \in data(\mathfrak{f}') \cup data(\mathfrak{f}'')$ there is a data value $f(d) \in data(\mathfrak{f})$ so that $\bar{\chi}(d) = \bar{\chi}'(d) = \bar{\chi}''(d)$ and $d \in R$ iff $f(d) \in R$. Then,

- $\mathfrak{f}''' = \mathfrak{f}'[f(d_1) \mapsto f(d_1), d_1] \cdots [f(d_n) \mapsto f(d_n), d_n]$ for $\{d_1, \ldots, d_n\} = data(\mathfrak{f}'') \setminus data(\mathfrak{f}')$, and
- $\mathfrak{f}''' = \mathfrak{f}''[f(d_1) \mapsto f(d_1), d_1] \cdots [f(d_n) \mapsto f(d_n), d_n]$ for $\{d_1, \ldots, d_n\} = data(\mathfrak{f}') \setminus data(\mathfrak{f}'')$.

Thus, $\mathfrak{f}'' \preceq \mathfrak{f}'''$, $\mathfrak{f}' \preceq \mathfrak{f}'''$. □

Remember that by definition of $\mathfrak{f}' \preceq \mathfrak{f}'''$, for every data value $d \in data(\mathfrak{f}''')$ there must be some data value $f(d) \in data(\mathfrak{f}')$ so that $\bar{\chi}'''(d) = \bar{\chi}'(d)$. We can further assume that for every $d \in data(\mathfrak{f}''')$, $d \in R_1'$ iff $f(d) \in R_1'$ (*i.e.*, we can simply define $f(d) = d$ for all $d \in R_1' \cap data(\mathfrak{f}'''))$. We define $\mathfrak{f}_1''$ as $R_1'' = R_1'$ and

$$\bar{\chi}_1''(d) = \begin{cases} \bar{\chi}_1'(d) & \text{if } d \in data(\mathfrak{f}_1') \\ \bar{\chi}_1'(f(d)) & \text{else, if } d \in data(\mathfrak{f}'''). \end{cases}$$

CLAIM B.6.2. $\mathfrak{f}_1'' \in (a, \bar{d})\mathfrak{f}'''$.

PROOF. For every data value of $d \in data(\mathfrak{f}_1')$ we have that $\bar{\chi}_1''(d)$ and $\bar{\chi}'''(d)$ are equal to $\bar{\chi}_1'(d)$ and $\bar{\chi}'(d)$, and it therefore verifies the conditions imposed by the rooting operation. Further, any other data value of $data(\mathfrak{f}_1'')$ (namely any data value from $data(\mathfrak{f}''') \setminus data(\mathfrak{f}_1'))$ behaves as some data value $f(d)$ in $\mathfrak{f}_1'$. That is $\bar{\chi}_1''(d)$ and $\bar{\chi}'''(d)$ are equal to $\bar{\chi}_1'(f(d))$ and $\bar{\chi}'(f(d))$. It then follows that that the data value must verify the conditions imposed by the rooting operation. Finally, by Lemma **??**, $\xi(\mathfrak{f}_1') \in \Gamma$ and $\mathfrak{f}_1' \precsim \mathfrak{f}_1''$, we have that $\xi(\mathfrak{f}_1'') \in \Gamma$. Hence, $\mathfrak{f}_1'' \in (a, \bar{d})\mathfrak{f}'''$. □

By definition of $\mathfrak{f}_1''$, it also follows that $\mathfrak{f}_1' \preceq \mathfrak{f}_1''$.

CLAIM B.6.3. $\mathfrak{f}_1' \preceq \mathfrak{f}_1''$.

PROOF. It is immediate from the definition that

$$\mathfrak{f}_1'' = \mathfrak{f}_1'[f(\hat{d}_1) \mapsto f(\hat{d}_1), \hat{d}_1, \ldots, f(\hat{d}_s) \mapsto f(\hat{d}_s), \hat{d}_s],$$

where $\{\hat{d}_1, \ldots, \hat{d}_s\} = data(\mathfrak{f}''') \setminus data(\mathfrak{f}_1')$. By definition of f, $f(\hat{d}_1), \ldots, f(\hat{d}_s) \in data(\mathfrak{f}') \setminus R_1'$, and hence $\mathfrak{f}_1' \preceq \mathfrak{f}_1''$. □

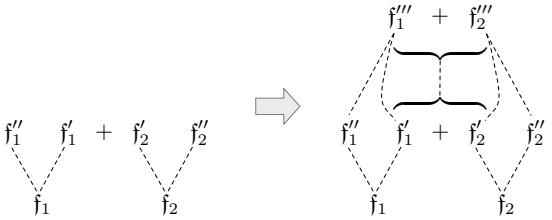


Figure 10: Statement of Lemma ??.

This concludes the proof. □

LEMMA B.7. $\boldsymbol{R}_{up}(\uparrow\downarrow\mathfrak{D}_b) \subseteq \downarrow\mathfrak{D}_b$.

PROOF OF LEMMA **??**. Let $\mathfrak{f}_1' \in \boldsymbol{R}_{up}(\uparrow\downarrow\mathfrak{D}_b)$. This means that there is some $\mathfrak{f}'' \in \mathfrak{D}_b$ and $\mathfrak{f}$ so that $\mathfrak{f} \preceq \mathfrak{f}''$, $\mathfrak{f} \preceq \mathfrak{f}'$, and $\mathfrak{f}_1' \in (a, \bar{d})\mathfrak{f}'$ for some $(a, \bar{d}) \in \mathbb{A} \times \mathbb{D}^{\mathbf{k}}$. We can then apply Lemma **??**, obtaining that there is some $\mathfrak{f}'''$ and $\mathfrak{f}_1''$ so that $\mathfrak{f}'' \preceq \mathfrak{f}'''$, $\mathfrak{f}' \preceq \mathfrak{f}'''$, $\mathfrak{f}_1' \preceq \mathfrak{f}_1''$, and $\mathfrak{f}_1'' \in (a, \bar{d})\mathfrak{f}'''$. Since $\mathfrak{D}_b$ is upward-closed by Remark 5.8, $\mathfrak{f}''' \in \mathfrak{D}_b$, and therefore $\mathfrak{f}_1'' \in \mathfrak{D}_b$ as well since $\mathfrak{f}_1'' \in (a, \bar{d})\mathfrak{f}'''$. Thus, as $\mathfrak{f}_1' \preceq \mathfrak{f}_1''$, we obtain $\mathfrak{f}_1' \in \downarrow\mathfrak{D}_b$. □

LEMMA B.8 (FIGURE **??**). *For every $i \in \{1,2\}$ and $\mathfrak{f}_i, \mathfrak{f}_i'\mathfrak{f}_i'' \in \mathfrak{F}_b$ so that $\mathfrak{f}_i \preceq \mathfrak{f}_i'$, $\mathfrak{f}_i \preceq \mathfrak{f}_i''$ and $\mathfrak{f}_1' + \mathfrak{f}_2'$ is defined, there are $\mathfrak{f}_1''', \mathfrak{f}_2''' \in \mathfrak{F}_b$ so that $\mathfrak{f}_1' + \mathfrak{f}_2' \preceq \mathfrak{f}_1''' + \mathfrak{f}_2'''$ and $\mathfrak{f}_i'' \preceq \mathfrak{f}_i'''$, $\mathfrak{f}_i' \preceq \mathfrak{f}_i'''$ for every $i \in \{1,2\}$.*

PROOF OF LEMMA **??**. Without any loss of generality, assume that $\mathfrak{f}_i \preceq \mathfrak{f}_i'$, $\mathfrak{f}_i \preceq \mathfrak{f}_i''$ and $data(\mathfrak{f}_i'') \cap data(\mathfrak{f}_i') = data(\mathfrak{f}_i)$ for every $i \in \{1,2\}$. Remember that by definition of $\mathfrak{f}_i \preceq \mathfrak{f}_i''$, for every data value $d \in data(\mathfrak{f}_i'')$ there must be some data value $f(d) \in data(\mathfrak{f}_i)$ so that $\bar{\chi}_i(d) = \bar{\chi}_i''(d) = \bar{\chi}_i'(d)$. We then define $\mathfrak{f}_i'''$ for every $i \in \{1,2\}$ so that $R_i''' = R_i$ and

$$\bar{\chi}_i'''(d) = \begin{cases} \bar{\chi}_i'(d) & \text{if } d \in data(\mathfrak{f}_i'), \\ \bar{\chi}_i''(d) & \text{else, if } d \in data(\mathfrak{f}_i''), \\ \bar{\chi}_i'(f(d)) & \text{else, if } d \in data(\mathfrak{f}_{3-i}''), \\ (\emptyset,\emptyset,\emptyset,\emptyset) & \text{otherwise.} \end{cases}$$

In the definition above, notice that for every $d \in data(\mathfrak{f}_{3-i}'')$ we have that $f(d) \in data(\mathfrak{f}_{3-i})$ and hence $f(d) \in data(\mathfrak{f}_{3-i}')$ which, by definition of $\mathfrak{f}_1' + \mathfrak{f}_2'$, means that $f(d) \in data(\mathfrak{f}_i')$.

The following claim follows straight from the definition just given.

CLAIM B.8.1. *$\mathfrak{f}_i' \preceq \mathfrak{f}_i'''$, $\mathfrak{f}_i'' \preceq \mathfrak{f}_i'''$ for all $i \in \{1,2\}$.*

PROOF. Since $\mathfrak{f}_i \preceq \mathfrak{f}_i''$ and $\mathfrak{f}_i \preceq \mathfrak{f}_i'$, there is some $\hat{f}$ so that for every $d \in data(\mathfrak{f}_i')$, $\hat{f}(d) \in data(\mathfrak{f}_i)$ so that $\bar{\chi}_i''(d) = \bar{\chi}_i(\hat{f}(d)) = \bar{\chi}_i''(\hat{f}(d))$ and $d \in R_i$ iff $\hat{f}(d) \in R_i$.

For every $i \in \{1,2\}$, it is immediate from the definition of $\mathfrak{f}_i'''$ that

$$\mathfrak{f}_i''' = \mathfrak{f}_i'[f(\hat{d}_1) \mapsto f(\hat{d}_1), \hat{d}_1] \cdots [f(\hat{d}_n) \mapsto f(\hat{d}_n), \hat{d}_n]\ [f(d_1) \mapsto f(d_1), d_1] \cdots [f(d_m) \mapsto f(d_m), d_m]$$

where $\{d_1, \ldots, d_m\} = data(\mathfrak{f}_{3-i}'')$ and $\{\hat{d}_1, \ldots, \hat{d}_n\} = data(\mathfrak{f}_i'') \setminus data(\mathfrak{f}_i')$.

Since $\mathfrak{f}_i \preceq \mathfrak{f}_i''$ and $\mathfrak{f}_i \preceq \mathfrak{f}_i'$, there is some $\hat{f}$ so that for every $d \in data(\mathfrak{f}_i')$, $\hat{f}(d) \in data(\mathfrak{f}_i)$ so that $\bar{\chi}_i'(d) = \bar{\chi}_i(\hat{f}(d)) = \bar{\chi}_i'(\hat{f}(d)) = \bar{\chi}_i''(\hat{f}(d))$ and $d \in R_i$ iff $\hat{f}(d) \in R_i$. For every $i \in \{1,2\}$, it is immediate from the definition of $\mathfrak{f}_i'''$ that

$$\mathfrak{f}_i''' = \mathfrak{f}_i''[\hat{f}(\hat{d}_1) \mapsto \hat{f}(\hat{d}_1), d_1] \cdots [\hat{f}(\hat{d}_n) \mapsto \hat{f}(\hat{d}_n), d_n]\ [f(d_1) \mapsto f(d_1), d_1] \cdots [f(d_m) \mapsto f(d_m), d_m]$$

where each d_i and $\hat{d}_i$ is so that $\{d_1, \ldots, d_m\} = data(\mathfrak{f}_{3-i}'')$ and $\{\hat{d}_1, \ldots, \hat{d}_n\} = data(\mathfrak{f}_i')$. □

We are then left with the following easy claim.

CLAIM B.8.2. *$\mathfrak{f}_1' + \mathfrak{f}_2' \preceq \mathfrak{f}_1''' + \mathfrak{f}_2'''$.*

PROOF. For every data value $d \in data(\mathfrak{f}_1') \cup data(\mathfrak{f}_2')$ it is easy to see that the conditions of $+$ apply for $\mathfrak{f}_1'''$ and $\mathfrak{f}_2'''$ since they have the same description for d. For any other data value $d \in data(\mathfrak{f}_1'') \cup data(\mathfrak{f}_2'')$ we have that $\mathfrak{f}_1'''$ and $\mathfrak{f}_2'''$ behave just as $\mathfrak{f}_1'$ and $\mathfrak{f}_2'$ for the data value $f(d)$. Therefore, the conditions of $+$ hold, and $\mathfrak{f}_1''' + \mathfrak{f}_2'''$ is well defined. Moreover, $\mathfrak{f}_1' + \mathfrak{f}_2' \preceq \mathfrak{f}_1''' + \mathfrak{f}_2'''$ since, by definition of $\mathfrak{f}_1''', \mathfrak{f}_2'''$ we have

$$\mathfrak{f}_1''' + \mathfrak{f}_2''' = (\mathfrak{f}_1' + \mathfrak{f}_2')[f(d_1) \mapsto f(d_1), d_1] \cdots [f(d_n) \mapsto f(d_n), d_n]$$

for $\{d_1, \ldots, d_n\} = data(\mathfrak{f}_1''' + \mathfrak{f}_2''') \setminus data(\mathfrak{f}_1' + \mathfrak{f}_2')$. □

This concludes the proof of the lemma. □

LEMMA B.9. $\boldsymbol{R}_{+}(\uparrow\downarrow\mathfrak{D}_b) \subseteq \downarrow\mathfrak{D}_b$.

Proof of Lemma **??**. Let $\mathfrak{f}'_3 \in \boldsymbol{R}_+(\uparrow\downarrow\mathfrak{D}_b)$. Then, there must be some $\mathfrak{f}'_1, \mathfrak{f}'_2 \in \mathfrak{F}_b$ so that $\mathfrak{f}'_1 + \mathfrak{f}'_2 = \mathfrak{f}'_3$ so that there exist $\mathfrak{f}_1, \mathfrak{f}_2, \mathfrak{f}''_1, \mathfrak{f}''_2$ where $\mathfrak{f}''_i \in \mathfrak{D}_b$, $\mathfrak{f}_i \preceq \mathfrak{f}'_i$, $\mathfrak{f}_i \precsim \mathfrak{f}''_i$ for all $i \in \{1,2\}$. Therefore, by Lemma **??**, there must be some $\mathfrak{f}'''_1, \mathfrak{f}'''_2 \in \mathfrak{F}_b$ so that $\mathfrak{f}'_3 = \mathfrak{f}'_1 + \mathfrak{f}'_2 \preceq \mathfrak{f}'''_1 + \mathfrak{f}'''_2$ and $\mathfrak{f}''_i \preceq \mathfrak{f}'''_i$, $\mathfrak{f}'_i \precsim \mathfrak{f}'''_i$ for all $i \in \{1,2\}$. Since $\mathfrak{f}''_1, \mathfrak{f}''_2 \in \mathfrak{D}_b$, and since $\mathfrak{D}_b$ is upward-closed by Remark 5.8, we have that $\mathfrak{f}'''_1, \mathfrak{f}'''_2 \in \mathfrak{D}_b$ and hence $\mathfrak{f}'''_1 + \mathfrak{f}'''_2 \in \mathfrak{D}_b$. Hence, since $\mathfrak{f}'_3 \precsim \mathfrak{f}'''_1 + \mathfrak{f}'''_2$ it follows that $\mathfrak{f}'_2 \in \downarrow\mathfrak{D}_b$. □

Proof of Lemma 5.9. Immediate from Lemmas **??** and **??**. □

Appendix to Section 5.6

Thanks to the property $\boldsymbol{R}(\uparrow\downarrow\mathfrak{D}_b) \subseteq \downarrow\mathfrak{D}_b$ one can show that $\boldsymbol{C}^{\sim}_{k_0} = \textsc{min}(\downarrow\mathfrak{D}_b)$ and that $\boldsymbol{C}_{i+1}$ can be computed from $\boldsymbol{C}_i$ in 2ExpSpace in $|\boldsymbol{C}_i|$ and $|\mathcal{P}|$. It is possible to test from the set $\textsc{min}(\downarrow\mathfrak{D}_b)$ whether there is a root derivable profile in $\mathfrak{D}_b$, hence obtaining that the derivation problem is decidable.

Lemma 5.11. $\boldsymbol{C}^{\sim}_{k_0} = \textsc{min}(\downarrow\mathfrak{D}_b)$.

Proof of Lemma 5.11. By Lemma 5.9 we know that $\boldsymbol{R}(\uparrow\downarrow\mathfrak{D}_b) \subseteq \downarrow\mathfrak{D}_b$. Therefore, since $\boldsymbol{C}_0 \subseteq \textsc{min}(\downarrow\mathfrak{D}_b)$, we have that for every i, $\boldsymbol{C}_i \subseteq \textsc{min}(\downarrow\mathfrak{D}_b)$. We then need to show that $\textsc{min}(\downarrow\mathfrak{D}_b) \subseteq \boldsymbol{C}^{\sim}_{k_0}$. Suppose, by means of contradiction, that there is some $\mathfrak{f} \in \mathfrak{F}_b$ so that $\mathfrak{f} \in \textsc{min}(\downarrow\mathfrak{D}_b)$ but $\mathfrak{f} \notin \boldsymbol{C}^{\sim}_{k_0}$. Then, since $\{\mathfrak{f}_\emptyset\} = \boldsymbol{C}_0 \subseteq \boldsymbol{C}_{k_0}$ there must be one such $\mathfrak{f}$ so that $\mathfrak{f} \precsim \mathfrak{f}'$ for some $\mathfrak{f}' \in \mathfrak{D}_b$ where $\mathfrak{f}' = \mathfrak{f}_1 + \mathfrak{f}_2$ for $\mathfrak{f}_1, \mathfrak{f}_2 \in \boldsymbol{C}^{\sim}_{k_0}$, or $\mathfrak{f}' \in (a,\bar{d})\mathfrak{f}''$ for $\mathfrak{f}'' \in \boldsymbol{C}^{\sim}_{k_0}$, $(a,\bar{d}) \in \mathbb{A} \times \mathbb{D}^{\mathbf{k}}$.

- If $\mathfrak{f}' = \mathfrak{f}'_1 + \mathfrak{f}'_2$ where $\mathfrak{f}'_1, \mathfrak{f}'_2 \in \boldsymbol{C}^{\sim}_{k_0}$, then $\mathfrak{f}' \in \boldsymbol{R}_+(\uparrow\downarrow\boldsymbol{C}_{k_0})$, and hence $\mathfrak{f} \in \textsc{min}(\boldsymbol{R}_+(\uparrow\downarrow\boldsymbol{C}_{k_0})) \subseteq \boldsymbol{C}^{\sim}_{k_0+1} = \boldsymbol{C}^{\sim}_{k_0}$, which is an absurd.
- If $\mathfrak{f}' \in (a,\bar{d})\mathfrak{f}''$ where $\mathfrak{f}'' \in \boldsymbol{C}^{\sim}_{k_0}$, then $\mathfrak{f}' \in \boldsymbol{R}^{(a,\bar{d})}_{up}(\uparrow\downarrow\boldsymbol{C}_{k_0})$, and hence $\mathfrak{f} \in \textsc{min}(\boldsymbol{R}^{(a,\bar{d})}_{up}) \subseteq \boldsymbol{C}^{\sim}_{k_0+1} = \boldsymbol{C}^{\sim}_{k_0}$, which is also an absurd.

Therefore, $\textsc{min}(\downarrow\mathfrak{D}_b) \subseteq \boldsymbol{C}^{\sim}_{k_0}$ and thus $\boldsymbol{C}^{\sim}_{k_0} = \textsc{min}(\downarrow\mathfrak{D}_b)$. □

We further have that this computation is in 2ExpSpace. Since $|\textsc{min}(\mathfrak{F}_b)/{\sim}|$ is doubly exponential in $|\mathcal{P}|$, we have the following.

Lemma B.10. *k_0 is bounded by a doubly exponential function on $|\mathcal{P}|$.*

Proof of Lemma **??**. Remember that $\mathfrak{F}_b$ is the set of profiles that have less than $2|\Pi|$ rigid values. Then,

$$\left(2^\Pi \times 2^\Pi \times 2^\Pi \times 2^\Pi\right)^{2|\Pi|}$$

represent all the possible profiles of the rigid values. We must also remember which profiles have either 0, or 1 or more flexible values, and this information can be represented with an element of

$$2^{2^\Pi \times 2^\Pi \times 2^\Pi \times 2^\Pi}.$$

Therefore, there are doubly exponentially many elements in $\textsc{min}(\mathfrak{F}_b)/{\sim}$, and hence k_0 is bounded by a doubly exponential function in $|\Pi|$. Since $|\Pi|$ is polynomial in $|\mathcal{P}|$, the statement follows. □

Lemma B.11. *Given $G \subseteq \mathfrak{F}_b$ and $\mathfrak{f} \in \mathfrak{F}_b$, testing $\mathfrak{f} \in \textsc{min}(\downarrow\boldsymbol{R}(\uparrow\downarrow G))$ is computable in 2ExpSpace in $|G|$ and $|\mathcal{P}|$.*

Proof of Lemma **??**. First we check that $\mathfrak{f} \in \textsc{min}(\downarrow\mathfrak{F}_b)$.

To check $\mathfrak{f} \in \textsc{min}(\downarrow\boldsymbol{R}_+(\uparrow\downarrow G))$, we verify if $\mathfrak{f} \precsim \mathfrak{f}_1 + \mathfrak{f}_2$ in 2ExpSpace, where $\mathfrak{f}_1, \mathfrak{f}_2$ is a pair of the (doubly exponentially many) bounded extensions of elements of $\textsc{min}(\downarrow G)$. By Lemma 5.3, there are such $\mathfrak{f}_1, \mathfrak{f}_2$ if, and only if, $\mathfrak{f} \in \textsc{min}(\downarrow\boldsymbol{R}_+(\uparrow\downarrow G))$.

Finally, to check $\mathfrak{f} \in \textsc{min}(\downarrow\boldsymbol{R}_{up}(\uparrow\downarrow G))$, we verify that $\mathfrak{f} \precsim \mathfrak{f}_3 \in \textsc{min}((a,\bar{d})\mathfrak{f}_4)$ in 2ExpSpace, where $\mathfrak{f}_4$ is a bounded extension of a profile of $\textsc{min}(\downarrow G)$. If the condition holds of course $\mathfrak{f} \in \textsc{min}(\downarrow\boldsymbol{R}_{up}(\uparrow\downarrow G))$. On the other hand, by Lemma 5.4 cum Remark 5.6, if $\mathfrak{f} \in (a,\bar{d})\mathfrak{f}_2$ with $\mathfrak{f}_2 \succsim \mathfrak{f}_5$, then $\mathfrak{f} \succsim \mathfrak{f}_3 \in \textsc{min}((a,\bar{d})\mathfrak{f}_4)$ for some bounded extension $\mathfrak{f}_4$ of $\mathfrak{f}_5$, and hence the condition holds. □

By the lemma above, we immediately obtain the following.

Lemma B.12. *$\boldsymbol{C}_{i+1}$ can be computed from $\boldsymbol{C}_i$ in 2ExpSpace in $|\boldsymbol{C}_i|$ and $|\mathcal{P}|$.*

Proof of Lemma **??**. For each $\mathfrak{f} \in \textsc{min}(\mathfrak{F}_b)/{\sim}$ we can check, in 2ExpSpace, wether $\mathfrak{f} \in \textsc{min}(\downarrow\boldsymbol{R}(\uparrow\downarrow\boldsymbol{C}_i))$ thanks to Lemma **??**. We can therefore compute $\textsc{min}(\downarrow\boldsymbol{R}(\uparrow\downarrow\boldsymbol{C}_i))/{\sim}$ in 2ExpSpace, and thus we can also compute $\boldsymbol{C}_{i+1}$ in 2ExpSpace. □

Proposition 5.12. *The derivation problem is decidable in 2ExpSpace.*

Proof of Proposition 5.12. We can compute all

$$\boldsymbol{C}_0, \ldots, \boldsymbol{C}_{k_0}$$

in 2ExpSpace by Lemmas **??** and **??**. Since $\boldsymbol{C}^{\sim}_{k_0} = \textsc{min}(\downarrow\mathfrak{D}_b)$ by Lemma 5.11, it follows that $\textsc{min}(\downarrow\mathfrak{D}_b)$ is hence computable in 2ExpSpace.

In order to test if there is a derivable profile in $\mathfrak{D}_b$ we choose some $\mathfrak{f} \in \textsc{min}(\{\mathfrak{f} \in \mathfrak{F}_b \mid \xi(\mathfrak{f}) \in \Gamma\})$ with no external values (*i.e.*, so that $\overleftarrow{\chi}^{\mathrm{E}} = \overrightarrow{\chi}^{\mathrm{E}} = \emptyset$), $\mathfrak{f}' \in \textsc{min}(\downarrow\mathfrak{D}_b)$ and $(a,\bar{d}) \in \mathbb{A}_{root} \times \mathbb{D}^{\mathbf{k}}$, and test, in 2ExpSpace (Lemma **??**), whether $\mathfrak{f} \in \textsc{min}(\downarrow\boldsymbol{R}^{(a,\bar{d})}_{up}(\uparrow\downarrow\{\mathfrak{f}'\}))$.

Claim B.12.1. *The following statements are equivalent.*

1. *There is some* $\mathfrak{f} \in \text{MIN}(\{\mathfrak{f} \in \mathfrak{F} \mid \xi(\mathfrak{f}) \in \Gamma\})$ *with no external values so that*
   - $\mathfrak{f} \in \text{MIN}(\downarrow \boldsymbol{R}_{up}^{(a,\bar{d})}(\uparrow\downarrow\{\mathfrak{f}'\}))$,
   - $(a, \bar{d}) \in \mathbb{A}_{root} \times \mathbb{D}^{\mathbf{k}}$ *and*
   - $\mathfrak{f}' \in \text{MIN}(\downarrow \mathfrak{D}_b)$.
2. *There is a derivable root profile in* $\mathfrak{D}_b$.

PROOF. [**??**⇒**??**] Assume first that condition **??** holds. We change the names for the profiles to make the explanation clear. Suppose that there is some $\mathfrak{f}_1' \in \text{MIN}(\downarrow \boldsymbol{R}_{up}^{(a,\bar{d})}(\uparrow\downarrow\{\mathfrak{f}\}))$ with no external data values, for some $(a, \bar{d}) \in \mathbb{A}_{root} \times \mathbb{D}^{\mathbf{k}}$, $\mathfrak{f}_1' \in \text{MIN}(\{\mathfrak{f} \in \mathfrak{F}_b \mid \xi(\mathfrak{f}) \in \Gamma\})$, and $\mathfrak{f} \in \text{MIN}(\downarrow \mathfrak{D}_b)$. This means that there must be some $\mathfrak{f}'' \in \mathfrak{D}_b$ so that $\mathfrak{f} \precsim \mathfrak{f}''$ and some $\mathfrak{f}'$ so that $\mathfrak{f} \precsim \mathfrak{f}'$ and $\mathfrak{f}_1' \in (a, \bar{d})\mathfrak{f}'$. We can then apply Lemma **??** obtaining that there is some $\mathfrak{f}'''$ and $\mathfrak{f}_1''$ so that $\mathfrak{f}'' \precsim \mathfrak{f}'''$, $\mathfrak{f}' \precsim \mathfrak{f}'''$, $\mathfrak{f}_1' \precsim \mathfrak{f}_1''$, and $\mathfrak{f}_1'' \in (a, \bar{d})\mathfrak{f}'''$. Since $\mathfrak{D}_b$ is upward closed (Remark 5.8), $\mathfrak{f}''' \in \mathfrak{D}_b$ and hence $\mathfrak{f}_1'' \in \mathfrak{D}_b$. By definition of $\precsim$, it follows that, since $\mathfrak{f}_1'$ has no external data values, $\mathfrak{f}_1''$ has no external data values either. Thus, there is a derivable root profile, namely $\mathfrak{f}_1''$.

[**??**⇐**??**] Suppose, on the other hand, that condition **??** holds. If there is a derivable root profile in $\mathfrak{D}_b$, then there are $\mathfrak{f}, \mathfrak{f}' \in \mathfrak{D}_b$ and $(a, \bar{d}) \in \mathbb{A}_{root} \times \mathbb{D}^{\mathbf{k}}$ so that $\overleftarrow{\chi}^{\text{E}} = \overrightarrow{\chi}^{\text{E}} = \emptyset$ and $\mathfrak{f} \in (a, \bar{d})\mathfrak{f}'$. Let $\mathfrak{f}_1' \in \text{MIN}(\downarrow\{\mathfrak{f}'\})/\!\sim$, that is, $\mathfrak{f}_1'$ is a minimal element corresponding to $\mathfrak{f}'$. Also, let $\mathfrak{f}_1 \in \text{MIN}(\downarrow\{\mathfrak{f}\})/\!\sim$, that is, $\mathfrak{f}_1$ is a minimal element corresponding to $\mathfrak{f}$. Of course, it follows that $\mathfrak{f}_1 \in \text{MIN}(\downarrow \boldsymbol{R}_{up}^{(a,\bar{d})}(\uparrow\downarrow\{\mathfrak{f}_1'\}))$. Note that $\mathfrak{f}_1 \in \text{MIN}(\{\mathfrak{f} \in \mathfrak{F}_b \mid \xi(\mathfrak{f}) \in \Gamma\})$, and that $\mathfrak{f}_1$ has no external values since $\mathfrak{f}$ has no external values. Hence condition **??** holds. □

Since there is a derivable root profile in $\mathfrak{D}_b$ if and only if there is a derivable root profile in $\mathfrak{D}$ by Lemma 5.7, the proposition follows. □

C. APPENDIX TO SECTION 6

Appendix to Section 6.1

LEMMA 6.1 (DIRECT NORMAL FORM). *There exists an exponential time translation that for every node expression* $\varphi \in \text{XPath}({}^{*}\!\leftarrow, \downarrow_{*}, \rightarrow^{*}, =)$ *returns an equivalent node expression* ψ *in direct normal form.*

PROOF OF LEMMA 6.1. The idea is that every data test expression $\langle \alpha@_i = \beta@_j \rangle$ is translated into a big disjunction of expressions in direct normal form, where loops in the tree navigation of α, β are factored as node expressions, as done in (‡).

For any finite alphabet $\mathbb{B}$, we define **tree order morphisms** between forests over a powerset alphabet $2^{\mathbb{B}}$. Given two forests $\bar{t}$, $\bar{t}'$ over $2^{\mathbb{B}}$, a tree order morphism from $\bar{t}$ to $\bar{t}'$ is a function f from the nodes of $\bar{t}$ to the nodes of $\bar{t}'$ so that

- for every node x of $\bar{t}$, the label of x in $\bar{t}$ is a subset of the label of $f(x)$ in $\bar{t}'$, and
- for every two nodes x, y of $\bar{t}$, if (x, y) is in the reflexive-transitive closure of the next-sibling relation (resp. of the child relation) in $\bar{t}$, then $(f(x), f(y))$ is also in the reflexive-transitive closure of the next-sibling relation (resp. of the child relation) in $\bar{t}'$.

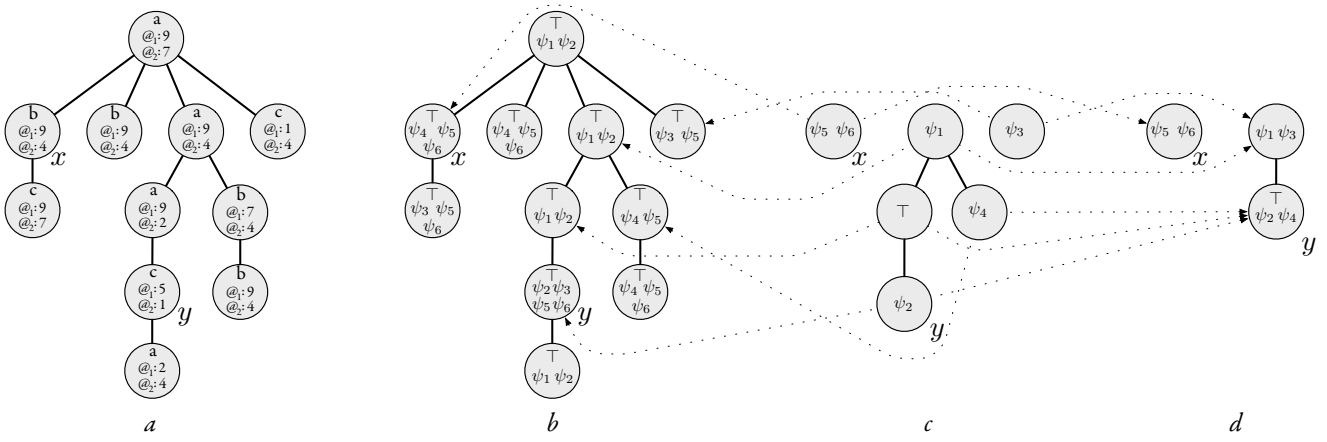
We say that $\bar{t}', (x', y')$ is a **contraction** of $\bar{t}, (x, y)$ if there is a tree order morphism f from $\bar{t}$ to $\bar{t}'$ so that $f(x) = x'$ and $f(y) = y'$. We also say that $\bar{t}, (x, y)$ is an **expansion** of $\bar{t}', (x', y')$. Notice that the function need not be surjective, and hence the fact that one forest is a contraction of another does not have any implication on the sizes of the forests: a forest may have less or more nodes than its contraction. For example, both the tree of Figure **??**-*b* and the forest of Figure **??**-*d* are contractions of the forest of Figure **??**-*c*.

Let α be a path expression, and let $ne(\alpha)$ be the set of all node expressions of α. Consider $\hat{\mathbf{t}}^{\alpha}$ the tree over the alphabet $2^{ne(\alpha)}$, where every node is labeled by those $\psi \in ne(\alpha)$ that are true at the node. For example, for the multi-attribute data tree $\mathbf{t}$ defined in Figure **??**-*a* and α as defined in Figure **??**-*e*, the tree $\hat{\mathbf{t}}^{\alpha}$ is the one of Figure **??**-*b*.

Let $\bar{t}$ be any forest over $2^{ne(\alpha)}$, and let x, y be two nodes of $\bar{t}$, so that x is a root. For such a forest and nodes, one can build a direct path expression $\mu^{\alpha}_{\bar{t},x,y}$ that tests if there is a path between x and y satisfying the relative order of nodes satisfying the labels as in the tree. For example, for the forest $\bar{t}$ of Figure **??**-*c* and the nodes x, y depicted, the corresponding expression $\hat{\mu}^{\alpha}_{\bar{t},x,y}$ would be the one appearing in Figure **??**-*e*.

CLAIM C.0.1. *A a direct path expression* $\mu^{\alpha}_{\bar{t},x,y}$ *can be built from* $\bar{t}$ *in polynomial time, such that for every multi-attribute data tree* $\mathbf{t}$ *we have that* $\mathbf{t}, (x', y') \models \mu^{\alpha}_{\bar{t},x,y}$ *if and only if* $\hat{\mathbf{t}}^{\alpha}, (x', y')$ *is a contraction of* $\bar{t}, (x, y)$.

PROOF. Given a forest $\bar{t}$ and nodes x, y so that x is a root, one can build $\mu^{\alpha}_{\bar{t},x,y}$ iteratively. We first build a path expression that starts in x and and ends in y and checks all the labels between x and y in the unique path between x and y that corresponds to a direct navigation. For example if $\bar{t}, x, y$ are as in Figure **??**-*a*, we build a path as in Figure **??**-*b* that tests all the node expressions of the nodes in



$$e \begin{cases} \alpha = {}^*\!\leftarrow[\psi_6]\rightarrow^*[\psi_3]^*\!\leftarrow[\psi_7]\rightarrow^*[\psi_1]\downarrow_*[\psi_4]^*\!\leftarrow[\top]\downarrow_*[\psi_2] \\ \mu^{\alpha}_{\bar{t},x,y} = [\psi_6 \wedge \psi_7]\rightarrow^*[\psi_1 \wedge \langle\rightarrow^*[\psi_3]\rangle]\downarrow_*[\top \wedge \langle\rightarrow^*[\psi_4]\rangle]\downarrow_*[\psi_2] \\ \quad \text{where} \quad \psi_1 = a \qquad \psi_3 = c \qquad \psi_5 = \neg a \\ \qquad\qquad \psi_2 = \langle\downarrow_*[a]\rangle \qquad \psi_4 = b \qquad \psi_6 = \neg\langle{}^*\!\leftarrow[a]\rangle \end{cases}$$

Figure 11: Example for the proof of Lemma 6.1.

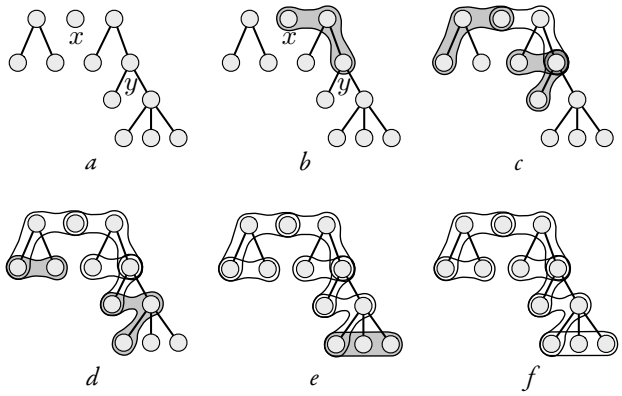


Figure 12: Idea of construction of $\hat{\mu}^{\alpha}_{\bar{t},x,y}$.

the path. We then build a more complex path expression nesting expressions that test the existence of paths as depicted in Figure **??**-*c*. We iterate until we have covered the whole tree $\bar{t}$, at each iteration, we add tests for paths that are at a deeper nesting degree in the resulting direct path expression. For example, the paths of Figure **??**-*c* are at depth 1, the paths of Figure **??**-*d* at depth 2, and the path of Figure **??**-*e* at depth 3. Note that if we apply this construction to the forest of Figure **??**-*c* and the path α of Figure **??**-*e*, we obtain the expression $\mu^{\alpha}_{\bar{t},x,y}$ of Figure **??**-*e*. □

Given a multi-attribute data tree $\mathbf{t}$ so that $\mathbf{t},(x,y) \models \alpha$, we call a *witness forest* of α for $\mathbf{t},(x,y)$ to a forest $\bar{t}$ over the alphabet $2^{ne(\alpha)}$ together with some nodes x, y, so that $\bar{t}$ contains only the nodes involved in the satisfaction of α, and they are labeled with the node expressions that they must verify. For example, given the multi-attribute data tree $\mathbf{t}$ of Figure **??**-*a*, and α as defined in Figure **??**-*e*, a possible witness forest of α for $\mathbf{t},(x,y)$ is depicted in Figure **??**-*c*. Of course, there may be several witness forests of α for a given $\mathbf{t},(x,y)$. Let $||\alpha||$ be the size of α, computed as the number of axes in α, irrespective of the size of node expressions. For example the path expression α defined in Figure **??**-*e* is so that $||\alpha|| = 7$. Notice that the number of nodes of any witness forest of α for $\mathbf{t},(x,y)$ is bounded by $\boldsymbol{p}(||\alpha||)$ for some polynomial $\boldsymbol{p}(\ )$.

Given a path expression α, let $\hat{\alpha}$ be the path expression over the alphabet $2^{ne(\alpha)}$, where every node expressions ψ is replaced by $\bigvee_{S \subseteq ne(\alpha), \psi \in S} S$, that is, it is treated as a disjunction of tests for labels from the alphabet. Notice that any witness forest $\bar{t},(x',y')$ of α for $\mathbf{t},(x,y)$ is an expansion of $\hat{\mathbf{t}}^{\alpha},(x,y)$, and we have $\bar{t},(x',y') \models \hat{\alpha}$. We then have the following.

CLAIM C.0.2. $\mathbf{t},(x,y) \models \alpha$ *if, and only if,* $\bar{t},(x',y') \models \hat{\alpha}$*, for some expansion* $\bar{t},(x',y')$ *of* $\hat{\mathbf{t}}^{\alpha},(x,y)$.

For every contraction $\bar{t},(x,y)$ of $\bar{t}',(x',y')$, if $\bar{t}',(x',y') \models \hat{\alpha}$ then $\bar{t}',(x',y') \models \hat{\alpha}$. This observation, together with the previous claim and the fact that every witness forest $\bar{t},(x',y')$ of α for $\mathbf{t},(x,y)$ is bounded by $\boldsymbol{p}(||\alpha||)$ yields the following.

CLAIM C.0.3. *There is an expansion* $\bar{t},(x',y')$ *of* $\hat{\mathbf{t}}^{\alpha},(x,y)$ *of size at most* $\boldsymbol{p}(||\alpha||)$ *so that* $\bar{t},(x',y') \models \hat{\alpha}$ *if, and only if,* $\mathbf{t},(x,y) \models \alpha$.

For every forest $\bar{t}$ over $2^{ne(\alpha)}$ of size $\leq \boldsymbol{p}(||\alpha||)$ and nodes x,y so that $\bar{t},(x,y) \models \hat{\alpha}$, we say that $\mu^{\alpha}_{\bar{t},x,y}$ is a *direct normal form linearization* of α. It follows that $\mathbf{t},(x,y) \models \mu^{\alpha}_{\bar{t},x',y'}$ if, and only if, a contraction of $\bar{t},(x',y')$ is a witness forest of α for $\mathbf{t},(x,y)$. Thus, the following claim follows.

CLAIM C.0.4. $\mathbf{t},(x,y) \models \alpha$ *if, and only if,* $\mathbf{t},(x,y) \models \mu^{\alpha}_{\bar{t},x',y'}$ *for some direct normal form linearization* $\mu^{\alpha}_{\bar{t},x',y'}$ *of* α.

PROOF. If $\mathbf{t},(x,y) \models \alpha$, then by Claim **??** there is a forest $\bar{t},(x',y')$ over $2^{ne(\alpha)}$ (the witness forest) of size $\leq \boldsymbol{p}(||\alpha||)$ so that $\bar{t},(x',y') \models \hat{\alpha}$. Hence, $\mu^{\alpha}_{\bar{t},x',y'}$ is a direct normal form linearization of α. Since $\bar{t},(x',y')$ is an expansion of $\hat{\mathbf{t}}^{\alpha},(x,y)$ and $\bar{t},(x',y') \models \hat{\alpha}$, then $\mathbf{t},(x,y) \models \mu^{\alpha}_{\bar{t},x',y'}$.

Suppose now that $\mathbf{t},(x,y) \models \mu^{\alpha}_{\bar{t},x',y'}$ for some $\bar{t},(x',y')$ over $2^{ne(\alpha)}$ of size $\leq \boldsymbol{p}(||\alpha||)$ so that $\bar{t},(x',y') \models \hat{\alpha}$. Then, $\bar{t},(x',y')$ must be an expansion of $\hat{\mathbf{t}}^{\alpha},(x,y)$ by Claim **??**. Since $\bar{t},(x',y') \models \hat{\alpha}$, we then have that $\mathbf{t},(x,y) \models \alpha$ by Claim **??**. □

Note that there are exponentially many trees in $||\alpha||$ over $ne(\alpha)$ of size $\leq \boldsymbol{p}(||\alpha||)$. Hence, all the direct normal form linearizations of a given path expression α are computable in exponential time $||\alpha||$. Hence, given $\varphi \in \text{XPath}({}^{*}\!\leftarrow, \downarrow_{*}, \rightarrow^{*}, =)$, consider φ' as the result of replacing every appearance of $\langle \alpha @_i = \beta @_j \rangle$ (resp. $\neq$) by a disjunction of $\langle \mu^{\alpha}_{\bar{t},x,y} @_i = \mu^{\beta}_{\bar{t}',x',y'} @_j \rangle$ (resp. $\neq$) for every direct normal form linearization $\mu^{\alpha}_{\bar{t},x,y}$ of α and $\mu^{\beta}_{\bar{t}',x',y'}$ of β. Since these are all the possible linearizations, the translated formula is satisfied in the same multi-attribute data trees and nodes as the original formula. Further, φ' is in direct normal form. Since every replacement is exponential in $||\alpha||$, and since $||\alpha||$ does not depend on the size of sub-node expressions, it follows that $|\varphi'|$ in time exponential in $|\varphi|$. Thus, the lemma follows. □

LEMMA 6.2 (UNNESTED NORMAL FORM). *There exists an exponential time translation that for every formula* $\eta \in \text{XPath}({}^{*}\!\leftarrow, \downarrow_{*}, \rightarrow^{*}, =)$ *returns a formula* φ *in unnested normal form such that* η *is satisfiable iff* φ *is satisfiable. Further, the translation of a formula in direct normal form is in direct normal form.*

PROOF OF LEMMA 6.2. Given a formula η we define the alphabet $\mathbb{A}_{\varphi}$ of the translation φ as all the locally consistent sets of subformulas of η. That is, the sets S such that for every subformula ψ of η: (1) if $\psi = \neg\psi'$ then $\{\psi', \neg\psi'\} \not\subseteq S$; (2) if $\psi = \psi' \wedge \psi''$ then $\psi \in S$ iff $\{\psi', \psi''\} \subseteq S$; and (3) if $\psi = \psi' \vee \psi''$ then $\psi \in S$ iff $\psi' \in S$ or $\psi'' \in S$.

Given a formula ψ, $tr(\psi)$ denotes the result of replacing

- every instance of a path expression $o_1[\psi_1]o_2 \cdots o_n[\psi_n]$ in ψ (where $o_1, \ldots, o_n \in \{{}^{*}\!\leftarrow, \downarrow_{*}, \rightarrow^{*}\}$) which does not appear nested inside another path expression, with the expression
$$o_1[\zeta_{\psi_1}]o_2 \cdots o_n[\zeta_{\psi_n}],$$
and
- every test for label a which does not appear inside a path expression by ζ_a,

where $\zeta_{\psi} \stackrel{def}{=} \bigvee_{S \in \mathbb{A}_{\varphi}, \psi \in S} S$.

To build the formula $\varphi = \varphi_1 \wedge \varphi_2$ in normal form, we define $\varphi_1 = \zeta_{\eta}$, and we build φ_2 as a conjunction of formulas
$$\neg\langle \downarrow_{*}[\zeta_{\psi} \wedge \neg tr(\psi)] \rangle$$
for all subformulas ψ of η. It is easy to see that this translation preserves satisfiability. □

COROLLARY 6.3. *About the translation of Lemma 6.2:*

1. *The set of path subformulas resulting from the translation has cardinality polynomial in* η.
2. *Every path subformula resulting from the translation can be written using polynomial space.*

PROOF OF COROLLARY 6.3. The blowup in the exponential translation comes only from the formulas ζ_{ψ}. In fact, φ can be symbolically written in polynomial space just as we did, using a symbol ζ_{ψ} instead of a big exponential disjunction. Remark that testing whether a label $S \in \mathbb{A}_{\varphi}$ satisfies ζ_{ψ} reduces to testing $\psi \in S$. □

Appendix to Section 6.2

LEMMA 6.4. *Given a direct non-recursive formula* ψ *that is a boolean combination of subformulas of* ϕ*, and two forest profiles* $\mathfrak{f}, \mathfrak{f}' \in \mathfrak{F}$ *so that* $\xi(\mathfrak{f}) = \xi(\mathfrak{f}')$ *then* $\mathfrak{f} \vdash \psi$ *if, and only if,* $\mathfrak{f}' \vdash \psi$.

PROOF. If $\psi \in \mathcal{B}(\mathbb{A})$, then it is immediate that $\mathfrak{f} \vdash \psi$ iff $\xi(\mathfrak{f})((\psi, \epsilon, i), \circ) \neq 0$ for some $i \in [\mathbf{k}]$.

Suppose now that $\psi = \langle \alpha \cdot \beta @_i \neq \gamma \cdot \delta @_j \rangle$ where α is leftward, ε or empty and γ is rightward, ε or empty, and β, δ are downward or empty. We show that whether $\mathfrak{f} \vdash \psi$ depends only on $\xi(\mathfrak{f})$. We show

one of the possible cases in the definition of $\vdash$, while the remaining ones are analogous or simpler. Suppose, for example, that $\alpha \neq \epsilon$, $\alpha \neq \varepsilon$, $\gamma = \epsilon$. We then have that $\mathfrak{f} \vdash \psi$ iff

- there is at least one data value reachable with $\alpha\beta@_i$, that is, $\xi(\mathfrak{f})(\bar{\alpha}, \overleftarrow{\circ}) \neq 0$, with $\bar{\alpha} = (\mathrm{P}^{-1}_{*\leftarrow}(\alpha), \mathrm{P}^{-1}_{\downarrow_*}(\beta), i)$, and
- there is at least one data value reachable with $\gamma\delta@_j$, that is, $\xi(\mathfrak{f})(\bar{\beta}, \circ^{\downarrow}) \neq 0$, with $\bar{\beta} = (\top, \mathrm{P}^{-1}_{\downarrow_*}(\delta), j)$ and either
  - there are two data values reachable with $\alpha\beta@_i$, that is, $\xi(\mathfrak{f})(\bar{\alpha}, \overleftarrow{\circ}) = 2+$, or
  - there are two data values reachable with $\gamma\delta@_j$, that is, $\xi(\mathfrak{f})(\bar{\beta}, \circ^{\downarrow}) = 2+$, or
  - there is one data value reachable with $\alpha\beta@_i$ and only one data value reachable with $\gamma\delta@_j$, but they are different, that is, $\xi(\mathfrak{f})(\bar{\alpha}, \overleftarrow{\circ}) = 1$, $\xi(\mathfrak{f})(\bar{\beta}, \circ^{\downarrow}) = 1$, and $\xi(\mathfrak{f})(\bar{\alpha}, \overleftarrow{\circ}, \bar{\beta}, \circ^{\downarrow}) = 0$.

These are indeed the necessary and sufficient conditions for the existence of two data values $d \neq d'$ so that $d \in \chi_a(\bar{\alpha})$ and $d' \in \chi_{a'}(\bar{\beta})$, where $a = \overrightarrow{\circ}$, $a' = \circ^{\downarrow}$. Hence, these are the necessary and sufficient conditions to verify $\mathfrak{f} \vdash \psi$.

The remaining cases are analogous or easier. □

LEMMA 6.5. *$\mathfrak{f} \vdash \bigwedge_{a \in \mathbb{A}}(a \Rightarrow \gamma_\varphi(a))$ if, and only if, $\xi(\mathfrak{f}) \in \Gamma_\phi$.*

PROOF OF LEMMA 6.5. This is a direct consequence of Lemma 6.4 and the definition of Γ_ϕ. □

Abstractions.

We now define $abs(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}, \bar{\mathbf{t}}_r)$ for any forests $\bar{\mathbf{t}}_l, \bar{\mathbf{t}}, \bar{\mathbf{t}}_r$. Let $\mathbf{t}_l = \mathbf{a}_l \otimes \mathbf{d}_l$, $\mathbf{t} = \mathbf{a} \otimes \mathbf{d}$, $\mathbf{t}_r = \mathbf{a}_r \otimes \mathbf{d}_r$ be the multi-attributes data trees $(a, \bar{d})\bar{\mathbf{t}}_l$, $(a, \bar{d})\bar{\mathbf{t}}$, and $(a, \bar{d})\bar{\mathbf{t}}_r$ respectively, for some fixed arbitrary $(a, \bar{d})$. We then define $abs(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}, \bar{\mathbf{t}}_r) = \bar{\chi}$ where

$$\overrightarrow{\chi}^{\mathrm{I}} = \{(\mathbf{d}(y)(i), \alpha, \beta, i) \mid (1, y) \in [\![\mathrm{P}_{\rightarrow^*}(\alpha)\cdot\mathrm{P}_{\downarrow_*}(\beta)]\!]_{\mathbf{t}}, \text{ for } (\alpha, \beta, i) \in \Pi \text{ and } y \in pos(\mathbf{t})\}$$

$$\overleftarrow{\chi}^{\mathrm{I}} = \{(\mathbf{d}(y)(i), \alpha, \beta, i) \mid (n, y) \in [\![\mathrm{P}_{*\leftarrow}(\alpha)\cdot\mathrm{P}_{\downarrow_*}(\beta)]\!]_{\mathbf{t}}, \text{ for } n = |\bar{\mathbf{t}}|, (\alpha, \beta, i) \in \Pi \text{ and } y \in pos(\mathbf{t})\}$$

$$\overrightarrow{\chi}^{\mathrm{E}} = \{(\mathbf{d}_r(y)(i), \alpha, \beta, i) \mid (1, y) \in [\![\mathrm{P}_{\rightarrow^*}(\alpha)\cdot\mathrm{P}_{\downarrow_*}(\beta)]\!]_{\mathbf{t}_r}, \text{ for } (\alpha, \beta, i) \in \Pi \text{ and } y \in pos(\mathbf{t}_r)\}$$

$$\overleftarrow{\chi}^{\mathrm{E}} = \{(\mathbf{d}_l(y)(i), \alpha, \beta, i) \mid (n, y) \in [\![\mathrm{P}_{*\leftarrow}(\alpha)\cdot\mathrm{P}_{\downarrow_*}(\beta)]\!]_{\mathbf{t}_l}, \text{ for } n = |\bar{\mathbf{t}}_l|, (\alpha, \beta, i) \in \Pi \text{ and } y \in pos(\mathbf{t}_l)\}.$$

For any $R \subseteq \mathbb{D}$ we define $abs_R(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}, \bar{\mathbf{t}}_r) \stackrel{def}{=} (abs(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}, \bar{\mathbf{t}}_r), R)$.

REMARK C.1. *Note that $abs_\emptyset(\epsilon, \epsilon, \epsilon) = \mathfrak{f}_\emptyset$.*

We show that *abs* is basically an algebra morphism between multi-attribute data forests with rooting and concatenation and forest profiles with profile rooting and profile concatenation. This is necessary to show our reduction from SAT-XPath($^*\!\leftarrow, \downarrow_*, \rightarrow^*, =$) into the derivation problem for forest profiles.

LEMMA C.2. *Given $R \subseteq \mathbb{D}$, and multi-attribute data forests $\bar{\mathbf{t}}, \bar{\mathbf{t}}_l, \bar{\mathbf{t}}_r, \bar{\mathbf{t}}_1, \bar{\mathbf{t}}_2$,*

1. *provided that $abs_R(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}_1, \bar{\mathbf{t}}_2\cdot\bar{\mathbf{t}}_r)$, $abs_R(\bar{\mathbf{t}}_l\cdot\bar{\mathbf{t}}_1, \bar{\mathbf{t}}_2, \bar{\mathbf{t}}_r)$, $abs_R(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}_1\cdot\bar{\mathbf{t}}_2, \bar{\mathbf{t}}_r) \in \mathfrak{F}$,*

$$abs_R(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}_1, \bar{\mathbf{t}}_2\cdot\bar{\mathbf{t}}_r) + abs_R(\bar{\mathbf{t}}_l\cdot\bar{\mathbf{t}}_1, \bar{\mathbf{t}}_2, \bar{\mathbf{t}}_r) = abs_R(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}_1\cdot\bar{\mathbf{t}}_2, \bar{\mathbf{t}}_r).$$

2. *Given $(a, \bar{d}) \in \mathbb{A}\times\mathbb{D}^{\mathbf{k}}$, $R' \subseteq \mathbb{D}$ so that $abs_{R'}(\epsilon, \bar{\mathbf{t}}, \epsilon) \in \mathfrak{F}$, $abs_R(\bar{\mathbf{t}}_l, (a, \bar{d})\bar{\mathbf{t}}, \bar{\mathbf{t}}_r) \in \mathfrak{F}$, and $\xi(abs_R(\bar{\mathbf{t}}_l, (a, \bar{d})\bar{\mathbf{t}}, \bar{\mathbf{t}}_r)) \in \Gamma_\phi$, we have*

$$abs_R(\bar{\mathbf{t}}_l, (a, \bar{d})\bar{\mathbf{t}}, \bar{\mathbf{t}}_r) \in (a, \bar{d})\ abs_{R'}(\epsilon, \bar{\mathbf{t}}, \epsilon).$$

PROOF OF LEMMA **??**.

??. Condition (a) is obvious because we have the same set R of rigid values in all profiles. Conditions (b) and (c) follow straight from the semantics of XPath and the definition of *abs*. By definition of *abs*, the set $\overrightarrow{\chi}^{\mathrm{E}}_1$ where $\mathfrak{f}_1 = abs_R(\bar{\mathbf{t}}_l\cdot\bar{\mathbf{t}}_1, \bar{\mathbf{t}}_2, \bar{\mathbf{t}}_r)$ is composed by all the paths that can reach data values in $\bar{\mathbf{t}}_2\cdot\bar{\mathbf{t}}_r$, which is precisely $\overrightarrow{\chi}^{\mathrm{I}}_2 \cup [\mathfrak{f}]\cdot\overrightarrow{\chi}^{\mathrm{E}}_2$ where $\mathfrak{f}_2 = abs_R(\bar{\mathbf{t}}_l\cdot\bar{\mathbf{t}}_1, \bar{\mathbf{t}}_2, \bar{\mathbf{t}}_r)$. In the same way, $\overleftarrow{\chi}^{\mathrm{E}}_2$ is composed by all the paths that can reach data values in $\bar{\mathbf{t}}_l\cdot\bar{\mathbf{t}}_1$, which is precisely $\overleftarrow{\chi}^{\mathrm{I}}_1 \cup \langle\mathfrak{f}_1\rangle\cdot\overleftarrow{\chi}^{\mathrm{E}}_1$. Finally, conditions (+2)–(+5) follow from the definition of *abs*.

??. Since $\xi(abs_R(\bar{\mathbf{t}}_l, (a, \bar{d})\bar{\mathbf{t}}, \bar{\mathbf{t}}_r)) \in \Gamma_\phi$, we only need to show that $abs_R(\bar{\mathbf{t}}_l, (a, \bar{d})\bar{\mathbf{t}}, \bar{\mathbf{t}}_r)$ is so that the rooting conditions (b) and (c) hold. Condition (b) is immediate since by definition of *abs* we have that $abs_{R'}(\epsilon, \bar{\mathbf{t}}, \epsilon)$ has empty external descriptions. Condition (c) holds by definition of *abs* and the semantics of XPath. □

For any derivation forest $\bar{t}$ (as defined in Section 4.3) we associate a multi-attribute data forest $\bar{\mathbf{t}}_t$ which is the result of removing all leaf nodes from t and projecting the tree onto $\mathbb{A} \times \mathbb{D}^{\mathbf{k}}$.

LEMMA C.3. *For every boolean combination ψ of non-recursive subformulas of ϕ, we have that $abs_R(\bar{\mathbf{t}}_l, \mathbf{t}, \bar{\mathbf{t}}_r) \vdash \psi$ if, and only if, $\bar{\mathbf{t}}_l\cdot\mathbf{t}\cdot\bar{\mathbf{t}}_r, i_0 \models \psi$, for $i_0 = |\bar{\mathbf{t}}_l| + 1$ and $R = data(\bar{\mathbf{t}}_l\cdot\mathbf{t}\cdot\bar{\mathbf{t}}_r)$.*

PROOF OF LEMMA **??**. Let $\mathfrak{f} = abs_R(\bar{\mathbf{t}}_l, \mathbf{t}, \bar{\mathbf{t}}_r)$.

• Suppose first that $\psi = b$ with $b \in \mathbb{A}$. Since $b \in \mathcal{P}_\phi$, we have that $\mathfrak{f} \vdash b$ iff $(b, \epsilon, i) \in \overrightarrow{\chi}^{\mathrm{I}}(d)$ for some d, i iff $(1, y) \in [\![\rightarrow^*[b]@_i]\!]_{(a,\bar{d})\mathbf{t}}$ for some y by definition of *abs* iff $(1,1) \in [\![\rightarrow^*[b]@_i]\!]_{(a,\bar{d})\mathbf{t}}$ iff $\mathbf{t} \models b$ iff $\bar{\mathbf{t}}_l \cdot \mathbf{t} \cdot \bar{\mathbf{t}}_r, i_0 \models b$, since $b \in \mathbb{A}$.

• If $\psi = \psi_1 \wedge \psi_2$ or $\psi = \neg\psi'$, we proceed by induction.

• Now suppose $\psi = \langle \alpha \cdot \beta @_i = \gamma \cdot \delta @_j \rangle$, where α is leftward or empty, γ is rightward or empty, and β, δ are downward or empty. Let $\mathbf{t}' = \mathbf{a}' \otimes \mathbf{d}' = (a, \bar{d})(\bar{\mathbf{t}}_l \cdot \mathbf{t} \cdot \bar{\mathbf{t}}_r)$.

[$\Rightarrow$] If $\mathfrak{f} \vdash \psi$ there must be some data value $d \in \mathbb{D}$ that witnesses this fact, as required by the definition of $\vdash$. Suppose first that $\alpha \neq \epsilon$, $\alpha \neq \varepsilon$. Hence, there must be some $d \in \mathbb{D}$ so that $(\mathrm{P}^{-1}_{*\leftarrow}(\alpha), \mathrm{P}^{-1}_{\downarrow_*}(\beta), i) \in (\langle\!\langle \mathfrak{f} \rangle\!\rangle \cdot \overleftarrow{\chi}^{\mathrm{E}} \cup \overleftarrow{\chi}^{\mathrm{I}})(d)$.

1. If $(\mathrm{P}^{-1}_{*\leftarrow}(\alpha), \mathrm{P}^{-1}_{\downarrow_*}(\beta), i) \in \overleftarrow{\chi}^{\mathrm{I}}(d)$, by definition of *abs* we have that $\overleftarrow{\chi}^{\mathrm{I}}$ is the set of all $(\mathbf{d}(y)(k), \alpha_1, \alpha_2, k)$ so that $(1, y) \in [\![\mathrm{P}_{\rightarrow^*}(\alpha_1)\mathrm{P}_{\downarrow_*}(\alpha_2)@_k]\!]_{(a,\bar{d})\mathbf{t}}$. It then follows that there must be some y' so that $(i_0, y') \in [\![\alpha \cdot \beta @_i]\!]_{(a,\bar{d})(\bar{\mathbf{t}}_l \cdot \mathbf{t} \cdot \bar{\mathbf{t}}_r)}$ with $\mathbf{d}'(y')(i) = d$.
2. If on the other hand $(\mathrm{P}^{-1}_{*\leftarrow}(\alpha), \mathrm{P}^{-1}_{\downarrow_*}(\beta), i) \in \langle\!\langle \mathfrak{f} \rangle\!\rangle \cdot \overleftarrow{\chi}^{\mathrm{E}}(d)$, either $(\mathrm{P}^{-1}_{*\leftarrow}(\alpha), \mathrm{P}^{-1}_{\downarrow_*}(\beta), i) \in \overleftarrow{\chi}^{\mathrm{E}}(d)$, or there must be some α'_1, α'_2 so that $\alpha'_1 \in \langle\!\langle \mathfrak{f} \rangle\!\rangle$ and $(\alpha'_2, \mathrm{P}^{-1}_{\downarrow_*}(\beta), i) \in \overleftarrow{\chi}^{\mathrm{E}}(d)$ and $\alpha'_1 \cdot \alpha'_2 = \mathrm{P}^{-1}_{*\leftarrow}(\alpha)$. Suppose that the latter occurs (the former being only easier). Since by definition of *abs* we have that $\overleftarrow{\chi}^{\mathrm{E}}$ is the set of all $(\mathbf{d}_l(y)(k), \alpha_1, \beta_1, k)$ so that $(n, y) \in [\![\mathrm{P}_{*\leftarrow}(\alpha_1)\mathrm{P}_{\downarrow_*}(\beta_1)@_k]\!]_{(a,\bar{d})\bar{\mathbf{t}}_l}$ for $n = |\bar{\mathbf{t}}_l| = i_0 - 1$ there is some y' so that $(i_0 - 1, y') \in [\![\alpha_2 \cdot \beta]\!]_{(a,\bar{d})(\bar{\mathbf{t}}_l \cdot \mathbf{t} \cdot \bar{\mathbf{t}}_r)}$ with $\mathbf{d}'(y')(i) = d$ and $\alpha_2 = \mathrm{P}_{*\leftarrow}(\alpha'_2)$. Also, since $\alpha'_1 \in \langle\!\langle \mathfrak{f} \rangle\!\rangle$, we have $(i_0, i_0 - 1) \in [\![\alpha_1]\!]_{(a,\bar{d})(\bar{\mathbf{t}}_l \cdot \mathbf{t} \cdot \bar{\mathbf{t}}_r)}$ where $\alpha_1 = \mathrm{P}_{*\leftarrow}(\alpha'_1)$. Then, there is some y' so that $(i_0, y') \in [\![\alpha \cdot \beta]\!]_{(a,\bar{d})(\bar{\mathbf{t}}_l \cdot \mathbf{t} \cdot \bar{\mathbf{t}}_r)}$ with $\mathbf{d}'(y')(i) = d$.

The case where $\alpha = \epsilon$, or $\alpha = \varepsilon$, $\gamma = \epsilon$ is only easier (it is basically as in the item 1). In any case we obtain that there is some y' so that $(i_0, y') \in [\![\alpha \cdot \beta]\!]_{(a,\bar{d})(\bar{\mathbf{t}}_l \cdot \mathbf{t} \cdot \bar{\mathbf{t}}_r)}$ with $\mathbf{d}'(y')(i) = d$.

By similar arguments, we have that there must be some $z' \in pos((a, \bar{d})(\bar{\mathbf{t}}_l \cdot \mathbf{t} \cdot \bar{\mathbf{t}}_r))$, $z' \neq \epsilon$ with $(i_0, z') \in [\![\alpha \cdot \beta]\!]_{(a,\bar{d})(\bar{\mathbf{t}}_l \cdot \mathbf{t} \cdot \bar{\mathbf{t}}_r)}$ and $\mathbf{d}'(z')(j) = d$. Hence, $\bar{\mathbf{t}}_l \cdot \mathbf{t} \cdot \bar{\mathbf{t}}_r, i_0 \models \langle \alpha \cdot \beta @_i = \gamma \cdot \delta @_j \rangle$.

[$\Leftarrow$] Suppose that $\bar{\mathbf{t}}_l \cdot \mathbf{t} \cdot \bar{\mathbf{t}}_r, i_0 \models \langle \alpha \cdot \beta @_i = \gamma \cdot \delta @_j \rangle$. In other words, there are $y, z \in pos(\mathbf{t}')$, $y \neq \epsilon$, $z \neq \epsilon$, so that $(i_0, y) \in [\![\alpha \cdot \beta]\!]_{\mathbf{t}'}$, $(i_0, z) \in [\![\gamma \cdot \delta]\!]_{\mathbf{t}'}$, and $d = \mathbf{d}'(z)(i) = \mathbf{d}'(y)(j)$ for some $d \in \mathbb{D}$. Let $\alpha' = \top$ if $\alpha = \varepsilon$ or $\alpha = \epsilon$ and $\alpha' = \mathrm{P}^{-1}_{*\leftarrow}(\alpha)$ otherwise.

1. If y is inside the subtree $\mathbf{t}$ of $\mathbf{t}'$, then $(\alpha', \mathrm{P}^{-1}_{\downarrow_*}(\beta), i) \in \overleftarrow{\chi}^{\mathrm{I}}(d)$, by definition of *abs*.
2. If y is inside the subforest $\bar{\mathbf{t}}_l$ of $\mathbf{t}'$, then we have that either $(\alpha', \mathrm{P}^{-1}_{\downarrow_*}(\beta), j) \in \overleftarrow{\chi}^{\mathrm{E}}(d)$, or there must be some α_1, α_2 so that $\alpha_1 \cdot \alpha_2 = \alpha$, $\alpha'_1 \in \langle\!\langle \mathfrak{f} \rangle\!\rangle$ and $(\alpha'_2, \mathrm{P}^{-1}_{\downarrow_*}(\beta), j) \in \overleftarrow{\chi}^{\mathrm{E}}(d)$, where $\alpha'_1 = \mathrm{P}^{-1}_{*\leftarrow}(\alpha_1), \alpha'_2 = \mathrm{P}^{-1}_{*\leftarrow}(\alpha_2)$. If the latter case holds (the former being only easier), we have that $(\alpha'_1 \cdot \alpha'_2, \mathrm{P}^{-1}_{\downarrow_*}(\beta), j) \in (\langle\!\langle \mathfrak{f} \rangle\!\rangle \cdot \overleftarrow{\chi}^{\mathrm{E}})(d)$, which implies that $(\alpha', \mathrm{P}^{-1}_{\downarrow_*}(\beta), j) \in (\langle\!\langle \mathfrak{f} \rangle\!\rangle \cdot \overleftarrow{\chi}^{\mathrm{E}})(d)$.

By similar arguments, we obtain some $(\gamma', \mathrm{P}^{-1}_{\downarrow_*}(\delta), j) \in (\overrightarrow{\chi}^{\mathrm{I}} \cup [\mathfrak{f}\rangle\!\rangle \cdot \overrightarrow{\chi}^{\mathrm{E}})(d)$, where $\gamma' = \top$ if $\gamma = \varepsilon$ or $\gamma = \epsilon$ and $\gamma' = \mathrm{P}^{-1}_{\rightarrow^*}(\gamma)$ otherwise. Hence, $\mathfrak{f} \vdash \langle \alpha \cdot \beta @_i = \gamma \cdot \delta @_j \rangle$.

• The cases of $\langle \alpha \cdot \beta @_i = \gamma \cdot \delta @_j \rangle$ where both α and γ are rightwards or leftwards is similar. The same applies to the inequality formulas $\langle \alpha \cdot \beta @_i \neq \gamma \cdot \delta @_j \rangle$. □

The next lemma basically shows that every derivable root profile is the abstraction of some tree satisfying ϕ.

LEMMA C.4. *Let $\mathfrak{f}_{root} \in \mathfrak{D}$ be a root profile, and let t be its derivation tree. Let $\bar{t}$ be a maximal subforest of t, where $\bar{\mathbf{t}}$ is the multi-attribute data forest associated with $\bar{t}$, and $\mathfrak{f}_1, \ldots, \mathfrak{f}_n$ are the profile labels of the roots of $\bar{t}$. Let R be the rigid set for $\mathfrak{f}_1, \ldots, \mathfrak{f}_n$. Then,*

- *$abs_R(\epsilon, \bar{\mathbf{t}}, \epsilon) = \mathfrak{f}_1 + \cdots + \mathfrak{f}_n$, and*
- *$\bar{\mathbf{t}}, i \models \phi_2$ for every $1 \leq i \leq |\bar{\mathbf{t}}|$.*

PROOF OF LEMMA **??**. We will make use of the following easy fact.

CLAIM C.4.1. *Given $\mathfrak{f}_1, \ldots, \mathfrak{f}_n \in \mathfrak{F}$, so that $\mathfrak{f}_+ = \mathfrak{f}_1 + \cdots + \mathfrak{f}_n$ where $\overleftarrow{\chi}^{\mathrm{E}}_+ = \overrightarrow{\chi}^{\mathrm{E}}_+ = \emptyset$, then the sets $\{\overrightarrow{\chi}^{\mathrm{I}}_i, \overleftarrow{\chi}^{\mathrm{I}}_i \mid i \in [n]\}$ determine the sets $\{\overrightarrow{\chi}^{\mathrm{E}}_i, \overleftarrow{\chi}^{\mathrm{E}}_i \mid i \in [n]\}$.*

PROOF. This is by conditions (b) and (c) of profile concatenation:

- $\overleftarrow{\chi}^{\mathrm{I}}_i$ and $\overleftarrow{\chi}^{\mathrm{E}}_i$ determine $\overleftarrow{\chi}^{\mathrm{E}}_{i+1}$. Hence, $\{\overleftarrow{\chi}^{\mathrm{E}}_i \mid i \in [n]\}$ are determined by $\overleftarrow{\chi}^{\mathrm{E}}_1$ and $\{\overleftarrow{\chi}^{\mathrm{I}}_i \mid i \in [n]\}$.
- $\overrightarrow{\chi}^{\mathrm{I}}_{i+1}$ and $\overrightarrow{\chi}^{\mathrm{E}}_{i+1}$ determine $\overrightarrow{\chi}^{\mathrm{E}}_i$. Hence, $\{\overrightarrow{\chi}^{\mathrm{E}}_i \mid i \in [n]\}$ are determined by $\overrightarrow{\chi}^{\mathrm{E}}_n$ and $\{\overrightarrow{\chi}^{\mathrm{I}}_i \mid i \in [n]\}$.

Once we fix $\overrightarrow{\chi}^{\mathrm{E}}_n = \overleftarrow{\chi}^{\mathrm{E}}_1 = \emptyset$, then $\{\overrightarrow{\chi}^{\mathrm{I}}_i, \overleftarrow{\chi}^{\mathrm{I}}_i \mid i \in [n]\}$ determine $\{\overrightarrow{\chi}^{\mathrm{E}}_i, \overleftarrow{\chi}^{\mathrm{E}}_i \mid i \in [n]\}$. □

We now prove the main statement by generalized induction on the height of $\bar{t}$. Suppose that $\bar{t}$ has height $m \geq 1$. Then there must be forests $\bar{\mathbf{t}}_1, \ldots, \bar{\mathbf{t}}_n$ so that

$$\bar{\mathbf{t}} = (a_1 \bar{d}_1)\bar{\mathbf{t}}_1 \cdots (a_n, \bar{d}_n)\bar{\mathbf{t}}_n$$

and each $\bar{\mathbf{t}}_i$ has height less than m. Let

$$\hat{\mathfrak{f}}_i = abs_R\big(\ ((a_j, \bar{d}_j)\bar{\mathbf{t}}_j)_{j<i}, (a_i, d_i)\bar{\mathbf{t}}_i, ((a_j, \bar{d}_j)\bar{\mathbf{t}}_j)_{j>i}\ \big)$$

for every i. We then have the following.

CLAIM C.4.2. *For every $i \in [n]$, $\mathfrak{f}_i = \hat{\mathfrak{f}}_i$.*

PROOF. For every $\bar{\mathbf{t}}_i \neq \epsilon$, we have that $\mathfrak{f}'_1 + \cdots + \mathfrak{f}'_l = abs_{R_i}(\epsilon, \bar{\mathbf{t}}_i, \epsilon)$ by inductive hypothesis, where $\mathfrak{f}'_1, \ldots, \mathfrak{f}'_l$ are the profile labels of the children of the root of the ith tree of $\bar{t}$. Then, since $\mathfrak{f}_i \in (a_i, \bar{d}_i)(\mathfrak{f}'_1 + \cdots + \mathfrak{f}'_l)$ by definition of derivation tree, we have that $\mathfrak{f}_i \in (a_i, \bar{d}_i) abs_{R_i}(\epsilon, \bar{\mathbf{t}}_i, \epsilon)$. If $\bar{\mathbf{t}}_i = \epsilon$, then, since $abs_\emptyset(\epsilon, \epsilon, \epsilon) = \mathfrak{f}_\emptyset$ and $\mathfrak{f}_i \in (a_i, \bar{d}_i)\mathfrak{f}_\emptyset$, we also have that $\mathfrak{f}_i \in (a_i, \bar{d}_i) abs_{R_i}(\epsilon, \bar{\mathbf{t}}_i, \epsilon)$. Hence, for all $i \in [n]$, $\mathfrak{f}_i \in (a_i, \bar{d}_i) abs_{R_i}(\epsilon, \bar{\mathbf{t}}_i, \epsilon)$.

Now, since

- all the internal descriptions of any forest profile inside $(a_i, \bar{d}_i) abs_\emptyset(\epsilon, \bar{\mathbf{t}}_i, \epsilon)$ is completely determined by $(a_i, \bar{d}_i)$ and $abs_\emptyset(\epsilon, \bar{\mathbf{t}}_i, \epsilon)$, and hence $\mathfrak{f}_i$ and $\hat{\mathfrak{f}}_i$ have the same internal descriptions,
- the external descriptions of $\mathfrak{f}_1 + \cdots + \mathfrak{f}_n$ are $\emptyset$,
- $abs_R(\epsilon, (a_1, \bar{d}_1)\bar{\mathbf{t}}_1 \cdots (a_n, \bar{d}_n)\bar{\mathbf{t}}_n, \epsilon) = \hat{\mathfrak{f}}_1 + \cdots + \hat{\mathfrak{f}}_n$ by Lemma **??-??**, and the external descriptions of $\hat{\mathfrak{f}}_1 + \cdots + \hat{\mathfrak{f}}_n$ are $\emptyset$

we can apply Claim **??**, concluding that $\mathfrak{f}_i = \hat{\mathfrak{f}}_i$ for every i. □

Therefore, since $abs_R(\epsilon, (a_1, \bar{d}_1)\bar{\mathbf{t}}_1 \cdots (a_n, \bar{d}_n)\bar{\mathbf{t}}_n, \epsilon) = \hat{\mathfrak{f}}_1 + \cdots + \hat{\mathfrak{f}}_n$ by Lemma **??-??**, we have that

$$abs_R(\epsilon, (a_1, \bar{d}_1)\bar{\mathbf{t}}_1 \cdots (a_n, \bar{d}_n)\bar{\mathbf{t}}_n, \epsilon) = \mathfrak{f}_1 + \cdots + \mathfrak{f}_n.$$

On the other hand, since $\xi(\mathfrak{f}_1), \ldots, \xi(\mathfrak{f}_n) \in \Gamma_\phi$ by condition (a) of rooting, we have $\hat{\mathfrak{f}}_i = \mathfrak{f}_i \vdash \bigwedge_{a \in \mathbb{A}_\phi}(a \Rightarrow \gamma_\phi(a))$ for every i. Hence, by Lemma **??**, $(a_1, \bar{d}_1)\bar{\mathbf{t}}_1 \cdots (a_n, \bar{d}_n)\bar{\mathbf{t}}_n, i \models \bigwedge_{a \in \mathbb{A}}(a \Rightarrow \gamma_\varphi(a))$ for every i. Also, by inductive hypothesis, for all $\bar{\mathbf{t}}_i \neq \epsilon$ we have that $\bar{\mathbf{t}}_i, j \models \phi_2$ for all $1 \leq j \leq |\bar{\mathbf{t}}_i|$. Then, by definition of ϕ_2, it follows that

$$(a_1, \bar{d}_1)\bar{\mathbf{t}}_1 \cdots (a_n, \bar{d}_n)\bar{\mathbf{t}}_n, i \models \phi_2$$

for every i. □

We now show that if ϕ is satisfiable, then there must be a derivable root profile.

LEMMA C.5. *Let $\mathbf{t}_\phi$ be a tree so that $\mathbf{t}_\phi \models \phi$. For every maximal subforest $\bar{\mathbf{t}}_l \cdot \bar{\mathbf{t}} \cdot \bar{\mathbf{t}}_r$ of $\mathbf{t}_\phi$ so that $\bar{\mathbf{t}} \neq \epsilon$, there is a derivable forest profile $\mathfrak{f} \in \mathfrak{D}$ so that $\mathfrak{f} = abs_R(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}, \bar{\mathbf{t}}_r)$, where $R = data(\bar{\mathbf{t}}_l \cdot \bar{\mathbf{t}} \cdot \bar{\mathbf{t}}_r)$. Further, if $\bar{\mathbf{t}}$ is a tree, $\mathfrak{f}$ is so that $\mathfrak{f} \in (a, \bar{d})\mathfrak{f}'$, with $\mathfrak{f}' \in \mathfrak{D}$.*

PROOF OF LEMMA **??**. We proceed by induction on the size of $\bar{\mathbf{t}}$, defined as its number of nodes. Suppose first that $\bar{\mathbf{t}}$ consists in one tree with just one node $(a, \bar{d})$. We show that $abs_R(\bar{\mathbf{t}}_l, (a, \bar{d}), \bar{\mathbf{t}}_r) \in (a, \bar{d})\mathfrak{f}_\emptyset$.

Since $\bar{\mathbf{t}}_l \cdot (a, \bar{d}) \cdot \bar{\mathbf{t}}_r, j \models \phi_2$, for $j = |\bar{\mathbf{t}}_l| + 1$, we have that $abs_R(\bar{\mathbf{t}}_l, \mathbf{t}, \bar{\mathbf{t}}_r) \vdash \gamma_\phi(a)$ by Lemma **??**. In other words, we have $\xi(abs_R(\bar{\mathbf{t}}_l, \mathbf{t}, \bar{\mathbf{t}}_r)) \in \Gamma_\phi$. Then, by Lemma **??-??** we have $abs_R(\bar{\mathbf{t}}_l, (a, \bar{d}), \bar{\mathbf{t}}_r) \in (a, \bar{d}) abs_\emptyset(\epsilon, \epsilon, \epsilon)$. Since $\mathfrak{f}_\emptyset = abs_\emptyset(\epsilon, \epsilon, \epsilon)$ (Remark **??**), we have that $(a, \bar{d}) abs_\emptyset(\epsilon, \epsilon, \epsilon) = (a, \bar{d})\mathfrak{f}_\emptyset$. Therefore, $abs_R(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}, \bar{\mathbf{t}}_r) \in \mathfrak{D}$.

Suppose now that $\bar{\mathbf{t}}$ consists in one tree $(a, \bar{d})\bar{\mathbf{t}}'$ where $\bar{\mathbf{t}}' \neq \epsilon$. By inductive hypothesis there is some $\mathfrak{f}' \in \mathfrak{D}$ so that $\mathfrak{f}' = abs_{R'}(\epsilon, \bar{\mathbf{t}}', \epsilon)$ for some R'. By Lemma **??** since $\bar{\mathbf{t}}_l \cdot (a, \bar{d})\bar{\mathbf{t}}' \cdot \bar{\mathbf{t}}_r, j \models \phi_2$ for $j = |\bar{\mathbf{t}}_l| + 1$, we have that $\xi(abs_R(\bar{\mathbf{t}}_l, (a, \bar{d})\bar{\mathbf{t}}', \bar{\mathbf{t}}_r)) \in \Gamma_\phi$. We then have that

$$abs_R(\bar{\mathbf{t}}_l, (a, \bar{d})\bar{\mathbf{t}}', \bar{\mathbf{t}}_r) \in (a, \bar{d})\mathfrak{f}' = (a, \bar{d}) abs_{R'}(\epsilon, \bar{\mathbf{t}}', \epsilon)$$

again by Lemma **??-??**. Therefore, $abs_R(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}, \bar{\mathbf{t}}_r) \in \mathfrak{D}$.

Suppose now that $\bar{\mathbf{t}}$ consists in more than one tree, let $\bar{\mathbf{t}}_1$ and $\bar{\mathbf{t}}_2$ be non-empty forests so that $\bar{\mathbf{t}} = \bar{\mathbf{t}}_1 \cdot \bar{\mathbf{t}}_2$. By applying twice the inductive hypothesis there must be $\mathfrak{f}_1, \mathfrak{f}_2 \in \mathfrak{D}$ so that $\mathfrak{f}_1 = abs_R(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}_1, \bar{\mathbf{t}}_2 \cdot \bar{\mathbf{t}}_r)$ and $\mathfrak{f}_2 = abs_R(\bar{\mathbf{t}}_l \cdot \bar{\mathbf{t}}_1, \bar{\mathbf{t}}_2, \bar{\mathbf{t}}_r)$. As $abs_R(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}_1 \cdot \bar{\mathbf{t}}_2, \bar{\mathbf{t}}_r) = abs_R(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}_1, \bar{\mathbf{t}}_2 \cdot \bar{\mathbf{t}}_r) + abs_R(\bar{\mathbf{t}}_l \cdot \bar{\mathbf{t}}_1, \bar{\mathbf{t}}_2, \bar{\mathbf{t}}_r)$ by Lemma **??-??**, it follows that $abs_R(\bar{\mathbf{t}}_l, \bar{\mathbf{t}}_1 \cdot \bar{\mathbf{t}}_2, \bar{\mathbf{t}}_r) = \mathfrak{f}_1 + \mathfrak{f}_2 \in \mathfrak{D}$ since $\mathfrak{f}_1, \mathfrak{f}_2 \in \mathfrak{D}$. □

COROLLARY 6.6. *There is a derivable root forest profile if, and only if, ϕ is satisfiable.*

PROOF OF COROLLARY 6.6. [⇒] Let $\mathfrak{f}$ be a derivable root forest profile. Then, there must be a derivation tree t for $\mathfrak{f} = (R, \bar{\chi})$. Let $\mathbf{t}$ be the multi-attribute data tree associated to t. By Lemma **??**, we have that $abs_R(\epsilon, \mathbf{t}, \epsilon) = \mathfrak{f}$ and that $\mathbf{t} \models \phi_2$. Since $\mathfrak{f}$ is a root profile, the root of $\mathbf{t}$ must have a label $(a, \bar{d})$ with $a \in \mathbb{A}_{root}$, and hence $\mathbf{t} \models \phi_1$ as well. Therefore, ϕ is satisfiable.

[⇐] Let $\mathbf{t}$ be so that $\mathbf{t} \models \phi$. Then by Lemma **??** we have that $\mathfrak{f} = abs_{data(\mathbf{t})}(\epsilon, \mathbf{t}, \epsilon) \in \mathfrak{D}$. Further, since $\mathbf{t}$ is a tree, we have that in fact $\mathfrak{f} = (a, \bar{d})\mathfrak{f}'$ for some $\mathfrak{f}' \in \mathfrak{D}$. Further, since $\mathbf{t} \models \phi_1$, $a \in \mathbb{A}_{root}$. Hence, $\mathfrak{f}$ is a derivable root profile. □