Definability and Transformations for Cost Logics and Automatic Structures

Martin Lang[1]*, Christof Löding[1], and Amaldev Manuel[2]**

[1] RWTH Aachen University, Lehrstuhl für Informatik 7, D-52056 Aachen, Germany
[2] LIAFA, CNRS & Université Paris Diderot – Paris 7, France

Abstract. We provide new characterizations of the class of regular cost functions (Colcombet 2009) in terms of first-order logic. This extends a classical result stating that each regular language can be defined by a first-order formula over the infinite tree of finite words with a predicate testing words for equal length. Furthermore, we study interpretations for cost logics and use them to provide different characterizations of the class of resource automatic structures, a quantitative version of automatic structures. In particular, we identify a complete resource automatic structure for first-order interpretations.

1 Introduction

The theory of regular cost functions [5] has emerged in recent years as a general theory for extensions of automata and logics that have been studied in the context of boundedness problems. In these problems, the exact values of the functions are not of specific interest but rather whether the function is bounded on specific subsets of the domain. For this reason, two cost functions are considered to be equivalent if they are bounded on the same subsets of the domain. It turns out that this coarser view renders decision problems for some classes of automaton-definable cost functions decidable. The central automaton model in this setting is the one of B-automata, which associate a value to the input words using counters that can be incremented or reset by the transitions (the execution of the transitions, however, does not depend on the counter values).

Together with the development of regular cost functions, several logical formalisms appeared. The logics introduced in this area extend normal first- or monadic second-order logics by special quantitative operators. In this paper, we are concerned with two such logics, namely cost MSO (CMSO) and cost FO (CFO). In CMSO (cf. [5]) atomic formulas $|X| \leq N$ for a set variable X and a free variable N can be used. The value of a formula is the least value for N such that the formula becomes true (and infinity otherwise). In CFO (cf. [11]) a quantifier of the form $\forall^{\leq N} x \varphi(x)$ can be used, which states that $\varphi(x)$ is true

* Supported by DFG research grant *Automatentheoretische Verifikationsprobleme mit Ressourcenschranken*

** Supported by funding from European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement №259454.

for almost all elements with at most N exceptions. As for CMSO, the value of the formula is the least value for N such that the formula is true. In order to ensure monotonicity, the newly introduced operators are only allowed to appear positively in a formula.

In the classical setting of languages, it is known that the regular languages are precisely those that are definable in MSO over word structures. This correspondence extends to B-automata and CMSO (cf. [5]). The FO definable languages correspond to the strict subclass of counter-free or aperiodic regular languages. There is also an analogue of this theorem in the cost setting, which is formulated using the temporal logic CLTL instead of CFO (cf. [12]) (the connection between CLTL and CFO was made in [11]).

However, there is a different way of looking at FO-definability of languages, as taken in [7]. Instead of considering the words as structures (with the word positions as elements), one can consider the set of all words as the domain of a structure. On such a structure one can define languages (subsets of the domain) by using formulas with one free element variable. If one equips the structure with successor relations for appending a letter to a word, the prefix relation, and a predicate for testing whether two words have the same length, then it turns out that the FO-definable languages are precisely the regular ones. Over a binary alphabet we refer to this structure as $\mathcal{T}_2^{\mathsf{el}}$. More generally, one can show that $\mathcal{T}_2^{\mathsf{el}}$ is complete for the class of automatic structures in the sense that all automatic structures can be defined (or interpreted) in $\mathcal{T}_2^{\mathsf{el}}$ by FO formulas. An automatic structure is a structure whose domain and relations can be defined by finite automata (for accepting relations the automata read all the input words synchronously in parallel). See [10, 3] for a more detailed introduction.

In this paper we mainly study this notion of definability and the class of automatic structures in the quantitative setting of regular cost functions. A notion of automatic structures with costs has already been introduced in [13]. There, the cost is not coming from specific operators in the logic, but is part of the structure: a tuple of elements is not simply in relation or not, but a value is associated to the tuple, which could be interpreted as the cost of being in relation (where the value infinity means that the tuple is not in the relation at all). This is achieved by using B-automata instead of classical automata in the definition of automatic structures. In [13] these costs have been considered as a model for the consumption of resources, and therefore these structures are called resource automatic structures and FO is referred to as FO with resource relations (FO+RR). As first main contribution, we define a complete resource automatic structure $c\mathcal{T}_2^{\mathsf{el}}$ as extension of $\mathcal{T}_2^{\mathsf{el}}$ and show that basically FO over $c\mathcal{T}_2^{\mathsf{el}}$ has the same expressive power as CFO over $\mathcal{T}_2^{\mathsf{el}}$.

Another way of obtaining a complete automatic structure is to consider finite sets of natural numbers as the elements of the structure (represented by finite words using the characteristic vector of the set) with the standard order of natural numbers on singleton sets, and the subset relation between sets. This structure can easily be defined in weak MSO over the structure $(\mathbb{N}, \mathrm{Succ})$ of the natural numbers with successor relation (in weak monadic second-order logic, set quantification only ranges over finite sets). In this transformation, we proceed from the structure $(\mathbb{N}, \mathrm{Succ})$ to its weak powerset structure (restricted to finite sets). By definition, WMSO on the original structure corresponds to FO on the powerset structure. This connection has already been observed in [8] and has been studied in more detail in [6]. Our second main contribution is a corresponding result for CWMSO and CFO. Furthermore, we extend the weak powerset structure by a size predicate for sets, show that this yields a complete resource automatic structure and establish the correspondence between CWMSO formulas and FO+RR formulas on this extended weak powerset structure.

The remainder of this paper is structured as follows. In Section 2 we give basic definitions and results. In Section 3 we characterize the class of regular cost functions in terms of CFO and FO+RR, thereby also relating these two logics over the class of resource automatic structures. In Section 4 we show how to obtain the class of resource automatic structures using CWMSO and establish a connection between CWMSO and the first-order logics CFO and FO+RR on the powerset structure. Furthermore, we would like to thank the anonymous reviewers for their constructive comments that helped us to improve this work.

2 Preliminaries

We start with providing a formal basis for the concepts mentioned before. A *cost function* is a function of the form $f : A \to \mathbb{N} \cup \{\infty\}$ that maps elements of its domain to natural numbers or infinity. In order to define the equivalence relation between cost functions, we first introduce the notion of a *correction function.* A correction function $\alpha : \mathbb{N} \cup \{\infty\} \to \mathbb{N} \cup \{\infty\}$ is a monotone mapping that maps ∞ and only ∞ to ∞. Let $f, g : A \to \mathbb{N} \cup \{\infty\}$ be two cost functions. We say f is α-dominated by g and write $f \preceq_\alpha g$ if for all $a \in A : f(a) \leq \alpha(g(a))$. We call f and g α-equivalent and write $f \approx_\alpha g$ if $f \preceq_\alpha g$ and $g \preceq_\alpha f$. Additionally, we may also drop the annotation α to indicate that there exists an α such that the relation holds. Then, we have $f \approx g$ iff they are bounded on the same subsets of the domain. A proof of this fact can be found in [5]. In this work, we often have the case $\alpha(n) = 2^n$. We use $\approx_{\exp}$ as a shorthand notation for $\approx_\alpha$ with this α in order to provide explicit bounds without too much notational overhead.

The basic model we consider to define *regular* cost functions are B-automata. They can be seen as NFAs extended by a finite set of non-negative integer counters. These counters support three kinds of operations[3]. First, the counter can be incremented ($\mathtt{i}$). Second, the counter can be reset to zero ($\mathtt{r}$) and lastly the counter can be left unchanged (ε). Each transition assigns one of these operations to every counter. So, formally, we have:

Definition 1. *A B-automaton is a tuple* $\mathfrak{A} = (Q, \Sigma, \mathsf{In}, \Delta, \mathsf{Fin}, \Gamma)$*. Where the components* $Q, \Sigma, \mathsf{In}, \mathsf{Fin}$ *are defined as for normal NFAs. The component* Γ *is a finite set of counters and* Δ *is a subset of* $Q \times \Sigma \times Q \times \{\mathtt{i}, \mathtt{r}, \varepsilon\}^\Gamma$*.*

[3] Please note that we do not consider the *check* operation originally introduced for B-automata in order to simplify the notation. This does not change their expressive power.

A run of a B-automaton is defined as usual as a sequence of (connected) transitions. The counter operations are executed along a run. This way, we associate the value of a run with the maximal occurring counter value. In total, the B-automaton induces a function that maps every word w to the inf of the values of all accepting runs on w. We write $[\![\mathfrak{A}]\!](w)$ to refer to this value. We also consider B-automata that read tuples of words synchronously, and thus defining a cost function over tuples of words. We call these automata (synchronous) B-transducers. Formally, they can be seen as standard B-automata whose alphabet consists of tuples of letters, using a padding symbol to extend all words to the same length (see, e.g., [13] for a detailed definition).

The logics C(W)MSO and CFO extend usual (W)MSO and FO logic by special quantitative operators (with the abbreviations as used in the introduction). These quantitative operators are only allowed to appear positively in formulas (within an even number of negations). A C(W)MSO formula is built by the normal MSO quantifiers and connectives. It additionally may have the atomic operation $|X| \leq N$ for set variables X (and a special symbol N later interpreted as some natural number). For a C(W)MSO formula φ we write $\mathfrak{S}, n \models \varphi$ if the structure $\mathfrak{S}$ satisfies φ as normal (W)MSO formula when we replace all the $|X| \leq N$ by a formula which checks that X is at most of size n. Moreover, we write $[\![\varphi]\!]^{\mathfrak{S}}$ to indicate the infimum over $n \in \mathbb{N}$ such that $\mathfrak{S}, n \models \varphi$. For example, the formula $\exists X(|X| \leq N \wedge \forall x(x \in X))$ counts the number of elements in a finite structure and evaluates to infinity on infinite structures. A CFO formula is built by normal FO quantifiers and connectives. It additionally may have the new quantifier $\forall^{\leq N} x\psi$. Similar to C(W)MSO, we write $\mathfrak{S}, n \models \varphi$ if $\mathfrak{S}$ satisfies φ as normal FO formula when we replace all occurrences of $\forall^{\leq N} x\psi$ by "ψ is true for all x with at most n exceptions". The notation $[\![\varphi]\!]^{\mathfrak{S}}$ is used accordingly. For example, the formula $\forall^{\leq N} x(x \neq x)$ counts the number of elements in a structure as the above CMSO formula. We remark that the restriction to appear positively ensures the monotonicity of the model relation and thereby that the quantitative semantics is well-defined. As a convention, we use upper case letters for set variables and lower case variables for element variables.

The two structures that are of main interest to us are the infinite binary tree with equal level predicate ($\mathcal{T}_2^{\mathsf{el}}$) and the natural numbers with successor predicate ($\mathbb{N}^{+1}$). We formally define $\mathcal{T}_2^{\mathsf{el}} = (\{0,1\}^*, \preceq, S_0, S_1, \mathsf{el})$. This follows the idea to identify a node with the word that describes the path leading from the root to it. The letter 0 indicates that the path in the tree branches to the left and the letter 1 indicates a branch to the right, respectively. With this view on the universe, the relation $\preceq$ is the prefix relation on words, the relations S_0 and S_1 are appending one letter (0 or 1, respectively) and el is the equal length predicate for a pair of words. The structure $\mathbb{N}^{+1}$ is defined by $\mathbb{N}^{+1} = (\mathbb{N}, \mathrm{Succ})$ where $\mathrm{Succ}(x, y)$ holds iff $y = x + 1$.

In addition to the cost logics, we also consider the logic First-Order+Resource Relations (for short FO+RR), which is evaluated over quantitative structures. A resource- or cost-structure is similar to normal relational structures but the relations have a quantitative valuation.

Definition 2 ([13]). *A resource structure $\mathfrak{S} = (S, R_1, \ldots, R_n)$ is a tuple consisting of a universe S and relation symbols R_1 up to R_n. The relation symbols are valuated by functions $R_i^{\mathfrak{S}} : S^k \to \mathbb{N} \cup \{\infty\}$ where k is the arity of the relation R_i.*

The syntax of the logic FO+RR is normal first-order logic without negation. The semantics of the relations is given by the resource structure, the semantics of complete formulas is given inductively by:

$$[\![R_i x_1 \ldots x_{k_i}]\!]^{\mathfrak{S}} := R_i^{\mathfrak{S}}(x_1, \ldots, x_{k_i})$$

$$[\![x = y]\!]^{\mathfrak{S}} := \begin{cases} 0 & \text{if } x = y \\ \infty & \text{otherwise} \end{cases} \qquad [\![x \neq y]\!]^{\mathfrak{S}} := \begin{cases} \infty & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

$$[\![\varphi \wedge \psi]\!]^{\mathfrak{S}} := \max([\![\varphi]\!]^{\mathfrak{S}}, [\![\psi]\!]^{\mathfrak{S}}) \qquad [\![\varphi \vee \psi]\!]^{\mathfrak{S}} := \min([\![\varphi]\!]^{\mathfrak{S}}, [\![\psi]\!]^{\mathfrak{S}})$$

$$[\![\exists x \varphi(x)]\!]^{\mathfrak{S}} := \inf_{s \in S} [\![\varphi(s)]\!]^{\mathfrak{S}} \qquad [\![\forall x \varphi(x)]\!]^{\mathfrak{S}} := \sup_{s \in S} [\![\varphi(s)]\!]^{\mathfrak{S}}$$

Intuitively, the value of a formula is the amount of resources needed to make the formula true. Interpreting ∞ as false and 0 as true, one obtains the standard semantics of FO. Therefore, we can interpret classical relations also as resource relations using the characteristic function operator χ that transforms a relation R into a function that maps tuples in relation to 0 and all other tuples to ∞. Moreover, we remark that although negation is not allowed in FO+RR, we may use negation on relations that are defined as characteristic function of a normal relation because we can always add the characteristic function of the complement.

Logical interpretations are a classical tool to represent one relational structure in another (cf. [9]). The key idea is to use logical formulas with free variables to define the universe as well as the relations of some structure $\mathfrak{B}$ in another structure $\mathfrak{A}$. This provides a systematic way to transfer decidability results for the logic on $\mathfrak{A}$ to $\mathfrak{B}$. One can effectively rewrite questions about formulas on $\mathfrak{B}$ to questions about formulas on $\mathfrak{A}$ by just replacing the relations with their defining formulas and relativizing quantification to the definition of the new universe. We extend this basic concept to define resource structures in two ways: We can either start with a normal relational structure and use a quantitative logic such as CMSO to define the quantitative relations of a resource structure or we may start with a resource structure and use FO+RR to define another resource structure. Moreover, we have to provide an additional formula that describes the negation of the universe in order to be able to relativize universal formulas.

Definition 3. *Let $\mathcal{L}$ be a logic with quantitative semantics. A* quantitative interpretation *$\mathcal{I}$ is a tuple $(\delta, \overline{\delta}, \varphi_1, \ldots, \varphi_k)$ of $\mathcal{L}$ formulas. For an interpretation $\mathcal{I}$ and an appropriate structure $\mathfrak{A} = (A, R_1, \ldots, R_n)$, we define the resource structure $\mathcal{I}(\mathfrak{A}) = (B, R_1, \ldots, R_k)$ where $B = \{a \in A \mid [\![\delta(a)]\!]^{\mathfrak{A}} = 0\}$ and $R_i^{\mathcal{I}(\mathfrak{A})}(\bar{x}) = [\![\varphi_i(\bar{x})]\!]^{\mathfrak{A}}$. The formulas δ and $\overline{\delta}$ are only allowed to assume the values 0 and ∞ and have to be inverse in the sense that $[\![\delta(x)]\!]^{\mathfrak{A}} = \infty \Leftrightarrow [\![\overline{\delta}(x)]\!]^{\mathfrak{A}} = 0$. Moreover, we say a resource structure $\mathfrak{S}$ is $\mathcal{L}$-interpretable in $\mathfrak{A}$ if there is an interpretation $\mathcal{I}$ such that $\mathfrak{S} \cong \mathcal{I}(\mathfrak{A})$.*

In the same way as in the classical case, an algorithm to compute the value of a formula can be transferred with quantitative $\mathcal{L}$-interpretations. However, the semantics of the logical connectives $\wedge$, $\vee$ and the quantifications has to coincide in FO+RR and $\mathcal{L}$. Since this is the case for FO+RR and cost logics, we obtain:

Proposition 4. *Let $\mathfrak{A}$ be a relational or resource structure, $\mathcal{L}$ be a logic of* $\mathrm{CMSO}, \mathrm{CWMSO}, \mathrm{CFO}, \mathrm{FO{+}RR}$*, and $\mathcal{I}$ be an $\mathcal{L}$-interpretation for $\mathfrak{A}$. Then each* FO+RR *formula over $\mathcal{I}(\mathfrak{A})$ can be transformed into an equivalent $\mathcal{L}$-formula over $\mathfrak{A}$.*

Proof. As usual, we replace the occurrences of the $\mathcal{I}(\mathfrak{A})$ relations by their defining formulas given in the interpretation and relativize existential quantification by using δ and universal quantification by using $\overline{\delta}$. The resulting transformed $\mathcal{L}$ formula is then a formula over $\mathfrak{A}$. The equivalence of the semantics follows from a straight forward induction over the structure of φ (using an inductive definition for the semantics of the logic $\mathcal{L}$). Note that the domain of $\mathcal{I}(\mathfrak{A})$ is, in general, a subset of the domain of $\mathfrak{A}$. For the equivalence statement we view a cost function over the domain of $\mathcal{I}(\mathfrak{A})$ as a cost function of the domain of $\mathfrak{A}$ that maps all elements in the difference to ∞. □

We remark that although the formula $\overline{\delta}$ is needed in an interpretation for technical reasons, we can omit it in most of the cases relevant to us. If $\mathcal{L}$ is one of the cost logics and the formula δ makes no use of the special quantitative operators, $\overline{\delta}$ can be obtained by taking the normal negation. This also applies to the case that $\mathcal{L}$ is FO+RR and the formula δ only uses relations that are defined using the χ operator. For the sake of simplicity, we do not define $\overline{\delta}$ explicitly in such situations.

3 Quantitative First-Order Logics

In the classical setting, it is known that FO formulas with one free variable over $\mathcal{T}_2^{\mathsf{el}}$ characterize the regular languages:

Theorem 5 ([7]). *For every regular language $L \subseteq \{0,1\}^*$ there is an* FO *formula $\varphi(x)$ with one free variable such that $w \in L$ iff $\mathcal{T}_2^{\mathsf{el}}, w \models \varphi$.*

We aim to show that this result extends in a very natural way to the setting of CFO and regular cost functions. Moreover, we relate CFO and FO+RR by introducing $c\mathcal{T}_2^{\mathsf{el}}$ as a variant of $\mathcal{T}_2^{\mathsf{el}}$ in form of a resource structure and show that the expressive power of CFO on $\mathcal{T}_2^{\mathsf{el}}$ equals FO+RR on $c\mathcal{T}_2^{\mathsf{el}}$ despite their apparent differences. Hence, both formalisms yield a new characterization of regular cost functions in terms of first-order like logics. We consider this to be an example for the robustness of the notion of regular cost functions.

With the quantitative semantics of CFO in mind, each formula with one free variable defines a function from the universe of the structure to $\mathbb{N} \cup \{\infty\}$. We claim that the definable functions are exactly the regular cost functions:

Theorem 6. *For every regular cost function $f : \{0,1\}^* \to \mathbb{N} \cup \{\infty\}$, there is a* CFO *formula $\varphi(x)$ such that $f \approx_{\exp} [\![\varphi]\!]^{\mathcal{T}_2^{\mathsf{el}}}$. Moreover, every function $[\![\varphi]\!]^{\mathcal{T}_2^{\mathsf{el}}}$ defined by a* CFO *formula $\varphi(x)$ is a regular cost function.*

We do not give a direct proof of this fact here but focus on establishing the above mentioned connection to resource structures and FO+RR. Theorem 6 follows from the translation between CFO and FO+RR (Propositions 8 and 9) and the characterization of regular cost functions with FO+RR (Theorem 12). We start with formally defining $c\mathcal{T}_2^{\mathsf{el}}$. It consists of all the relations present in $\mathcal{T}_2^{\mathsf{el}}$ but now valuated with their characteristic function and one new (truly) quantitative relation $|\cdot|_1$ that counts the number of ones in a word. Formally, we define $c\mathcal{T}_2^{\mathsf{el}}$ by:

Definition 7. *Let $c\mathcal{T}_2^{\mathsf{el}} = (\{0,1\}^*, \preceq, S_0, S_1, \mathsf{el}, |\cdot|_1)$ with $|w|_1^{c\mathcal{T}_2^{\mathsf{el}}}$ counting the number of letters 1 in w and all the other relations valuated by the characteristic functions of their valuations in $\mathcal{T}_2^{\mathsf{el}}$.*

We observe that one can easily define $c\mathcal{T}_2^{\mathsf{el}}$ with CFO formulas in $\mathcal{T}_2^{\mathsf{el}}$, which means by Proposition 4 that FO+RR over $c\mathcal{T}_2^{\mathsf{el}}$ can be translated into CFO over $\mathcal{T}_2^{\mathsf{el}}$. In combination with Theorem 12 this yields one direction of Theorem 6.

Proposition 8. *$c\mathcal{T}_2^{\mathsf{el}}$ is* CFO*-interpretable in $\mathcal{T}_2^{\mathsf{el}}$.*

Proof. The universe of $c\mathcal{T}_2^{\mathsf{el}}$ is identical to $\mathcal{T}_2^{\mathsf{el}}$. Consequently, we can set $\delta(x) := x = x$. The relations $\preceq$, S_0, S_1, el directly define their quantitative counterparts because their quantitative semantics is just the characteristic function. So, it remains to define $|\cdot|_1$ as a CFO formula over $\mathcal{T}_2^{\mathsf{el}}$:

$$|w|_1 := \forall^{\leq N} x(\exists y(S_0yx \vee S_1yx) \wedge x \preceq w) \rightarrow \exists y S_0yx$$

The idea behind this formula is that all elements x that are a predecessor of w (except the empty word) have to be 0-successors with at most N exceptions that are exactly the 1-positions in w. □

In order to provide the other direction of Theorem 6, we show how to translate CFO formulas over $\mathcal{T}_2^{\mathsf{el}}$ into equivalent FO+RR formulas over $c\mathcal{T}_2^{\mathsf{el}}$.

Proposition 9. *For every* CFO *formula φ, there is an* FO+RR *formula $\tilde{\varphi}$ such that $[\![\varphi]\!]^{\mathcal{T}_2^{\mathsf{el}}} \approx_{\exp} [\![\tilde{\varphi}]\!]^{c\mathcal{T}_2^{\mathsf{el}}}$.*

Proof (sketch). We provide an inductive translation from CFO to FO+RR. The key difficulty here arises from replacing the $\forall^{\leq N}$ quantifier with an FO+RR-expressible equivalent. We do so by approximating the set with the exception elements in form of one tree element. The path from the root to this tree element branches to the right in each level such that there are two exception elements with longest common ancestor in this level. With this idea, we can approximate the number of exceptions up to one exponential. □

Now that we established the connection between CFO and FO+RR, we show that FO+RR formulas with one free variable on $c\mathcal{T}_2^{\mathsf{el}}$ capture regular cost functions. The translation from FO+RR into B-automata can easily be done with the concept of resource automatic structures in mind. We recall the definition from [13] and show that it applies to $c\mathcal{T}_2^{\mathsf{el}}$:

Definition 10. *A resource structure $\mathfrak{S} = (S, R_1, \ldots, R_n)$ is called* resource automatic *if $S \subseteq \{0,1\}^*$ is a regular language and and there are synchronous B-transducers $\mathfrak{T}_1, \ldots, \mathfrak{T}_n$ such that $R_i^{\mathfrak{S}}(\bar{x}) := [\![\mathfrak{T}_i]\!](\bar{x})$.*

Proposition 11. *$c\mathcal{T}_2^{\mathsf{el}}$ is a resource automatic structure.*

Proof. It is known that $\mathcal{T}_2^{\mathsf{el}}$ is an automatic structure (cf. [3]). By interpreting the NFAs defining the classical relations as B-automata in which the counters are not used, we directly obtain automata for the characteristic functions. The relation $|\cdot|_1$ can be defined by an automaton that just counts the letters 1. □

With this result in mind we can now characterize regular cost functions in terms of FO+RR on $c\mathcal{T}_2^{\mathsf{el}}$:

Theorem 12. *For every regular cost function $f : \{0,1\}^* \to \mathbb{N} \cup \{\infty\}$, there is an* FO+RR *formula $\varphi(x)$ such that $f = [\![\varphi]\!]^{c\mathcal{T}_2^{\mathsf{el}}}$. Moreover, the function $[\![\varphi]\!]^{c\mathcal{T}_2^{\mathsf{el}}}$ defined by an* FO+RR *formula with one free variable is a regular cost function.*

Proof. We first show the second part. Let $\varphi(x)$ be a FO+RR formula with one free variable. Since $c\mathcal{T}_2^{\mathsf{el}}$ is resource automatic, the inductive automaton translation from [13] provides us with a B-automaton $\mathfrak{A}$ such that $[\![\varphi]\!]^{c\mathcal{T}_2^{\mathsf{el}}} \approx_\alpha [\![\mathfrak{A}]\!]$.

For the converse, we encode the run of a B-automaton in $c\mathcal{T}_2^{\mathsf{el}}$. Let $\mathfrak{A} = (Q, \Sigma, \mathsf{In}, \Delta, \mathsf{Fin}, \Gamma)$ be a B-automaton with $|Q| = n$ and $|\Gamma| = m$. The basic construction follows the classical approach. We simulate the behavior of $\mathfrak{A}$ on a word w with a formula $\varphi(w)$ by existentially guessing a run, verifying that it is accepting and now additionally computing its value. The state sequence is encoded along the levels in the tree in the following way: For each of the n states, we guess a position p_i on the same level as w. The path to p_i branches to the right in all levels in which the run is currently in the i-th state. Now we only have to verify that in every level up to $|w|$ exactly one of the paths branches right and that there are transitions that enable the respective state change given the letter of w in the corresponding level. We extend this with $2m$ additional positions on the level of w to describe the behavior of the counters. For each counter, there is a position $c_i^{\mathtt{i}}$ that branches right in every level where the transition increments the respective counter and a position $c_i^{\mathtt{r}}$ that branches right in every level where the transition resets the counter. We can then calculate the value of a run by selecting maximal segments of the path given by $c_i^{\mathtt{i}}$ that are not interrupted by a right-branch of $c_i^{\mathtt{r}}$. We use the $|\cdot|_1$ relation on these segments to count the increments. □

A Complete Resource Automatic Structure

The study of *complete* structures for certain classes of logical structures provides insight into the whole class of structures by looking at a single structure. Hence, we are interested in finding a complete structure for resource automatic structures. This not only provides a characterization of the expressive power of the formalism but also enables us to better understand the type of quantitative extension realized by resource automatic structures. First, we formally fix the notion of completeness:

Definition 13. *Let $\mathcal{C}$ be a class of resource structures. We call a structure $\mathfrak{S}$* complete *for $\mathcal{C}$ if $\mathfrak{S} \in \mathcal{C}$ and for all structures $\mathfrak{A} \in \mathcal{C}$, there is an* FO+RR-*interpretation $\mathcal{I}$ such that $\mathfrak{A} \cong \mathcal{I}(\mathfrak{S})$.*

By Proposition 11, we already know that $c\mathcal{T}_2^{\mathsf{el}}$ is a resource automatic structure. In order to show that it is complete, we have to extend the ideas of Theorem 12 to synchronous B-transducers. Consider an arbitrary resource automatic structure $\mathfrak{S}$. The universe is represented by a regular language. As a consequence, the classical Theorem 5 provides us with a formula δ to encode the domain. It remains to find formulas that define the relations in $c\mathcal{T}_2^{\mathsf{el}}$. For this we take the synchronous B-transducer that defines the relation. Since this is essentially only a B-automaton working over a vector of the original alphabet, the same approach as in Theorem 12 can be used to obtain the formula φ that defines the relation in $c\mathcal{T}_2^{\mathsf{el}}$. Although this involves some technical difficulties such as necessary padding when working with synchronous transducers, no new ideas are required in principle. Altogether, we obtain:

Corollary 14. *The structure $c\mathcal{T}_2^{\mathsf{el}}$ is complete for resource automatic structures.*

This result concisely illustrates the quantitative extension that is provided by resource automatic structures compared to the standard model. The idea of characteristic functions provides an embedding that shows that resource automatic structures extend the classical concept. The quantitative aspect boils down to a relation that counts the number of letters 1 in the word presentation of the elements.

4 (Finite)Set Transformations

In the classical setting, it is a well-known fact that (W)MSO formulas over a structure are equivalent to FO formulas over the (weak) powerset structure (cf. [2]). We aim at providing a generalization of this fact to the area of cost logics and resource structures. First, we fix the notation used in this section. Let $\mathfrak{A} = (A, R_1, \ldots, R_k)$ be a relational structure. For a j-ary relation R, let the *set extension* of R be given by $\mathcal{P}(R) := \{(\{x_1\}, \ldots, \{x_j\}) \mid (x_1, \ldots, x_j) \in R\}$. The powerset structure of $\mathfrak{A}$ is $\mathcal{P}(\mathfrak{A}) = (\mathcal{P}(A), \mathrm{Sing}, \subseteq, \mathcal{P}(R_1), \ldots, \mathcal{P}(R_k))$ where $\subseteq$ is the normal subset relation and Sing is a unary predicate that indicates singleton sets. Additionally, we also show that there is a correspondence to a canonical resource extension of the powerset structure. The resource powerset structure of $\mathfrak{A}$ is $c\mathcal{P}(\mathfrak{A}) = (\mathcal{P}(A), \mathrm{Sing}, \mathrm{size}, \subseteq, \mathcal{P}(R_1), \ldots, \mathcal{P}(R_k))$ where size is a unary resource relation mapping a set to its size, the other relations are valuated with the characteristic functions of their valuations in the classical powerset structure. Analogously, we also consider the weak variant with only finite subsets of A in the universe. We denote the weak variants by an index w.

Proposition 15. *The following correspondence holds for all relational structures $\mathfrak{A}$. For every* CMSO *formula φ, there is a* CFO *formula φ_1 (respectively a* FO+RR *formula φ_2) and vice versa such that for all $X_1, \ldots, X_k \subseteq A$ and all $x_1, \ldots, x_\ell \in A$ it is the case that*

$$\llbracket \varphi(X_1, \ldots, X_k, x_1, \ldots, x_\ell) \rrbracket^{\mathfrak{A}} \approx_{\exp} \llbracket \varphi_1(X_1, \ldots, X_k, \{x_1\}, \ldots, \{x_\ell\}) \rrbracket^{\mathcal{P}(\mathfrak{A})}$$
$$\llbracket \varphi(X_1, \ldots, X_k, x_1, \ldots, x_\ell) \rrbracket^{\mathfrak{A}} = \llbracket \varphi_2(X_1, \ldots, X_k, \{x_1\}, \ldots, \{x_\ell\}) \rrbracket^{c\mathcal{P}(\mathfrak{A})}.$$

The same holds for CWMSO *and the respective weak powerset structures.*

We obtain the previous result by inductively transforming logical formulas in a way that preserves the semantics up to α. Most of the translations are relatively straightforward encodings of the missing operators. However, transforming a CFO formula over the powerset structure back into a CMSO formula over the original structure involves counting the number of "exceptions" in $\forall^{\leq N} x \varphi(x)$ formulas. Since the x are sets of the original structure, we cannot simply existentially quantify the set of exceptions and bound its size. We solve this by instead bounding the size of sets that contain only elements with a distinct membership profile w.r.t. the exception sets, i.e., a pair of element z, z' can be member in this set iff there is an exception that contains exactly one of z, z'. For these sets, we recognize that their size approximates the number of exception sets up to an exponentiation. This can be seen as a refinement of the approximation idea used in Proposition 2.1 in [1].

The translation from Proposition 15 has the following immediate consequence for the definable cost functions:

Corollary 16. *Let f be a cost function. The following are equivalent:*

1. *f is definable in (weak)* CMSO *over a structure $\mathfrak{A}$.*
2. *f is definable in* CFO *over (weak) $\mathcal{P}(\mathfrak{A})$.*
3. *f is definable in* FO+RR *over (weak) $c\mathcal{P}(\mathfrak{A})$.*

In order to close our study of logical formalisms for regular cost functions, we connect our results in the area of first-order logics with CWMSO on $\mathbb{N}^{+1}$. For this we recognize that $c\mathcal{T}_2^{\mathsf{el}}$ is almost $c\mathcal{P}_{\mathrm{w}}(\mathbb{N}^{+1})$. A word from $\{0,1\}$ can be seen as a set over the natural numbers that contains the positions in the word with letter 1. Nevertheless, we additionally need WMSO-definable coding in order to distinguish tree elements with trailing zeros. Hence, we obtain that $c\mathcal{T}_2^{\mathsf{el}}$ and $c\mathcal{P}_{\mathrm{w}}(\mathbb{N}^{+1})$ are FO+RR-interpretable in each other. The connection between the two first-order logics stated in Theorem 17 below was already established in Proposition 9 and Proposition 8.

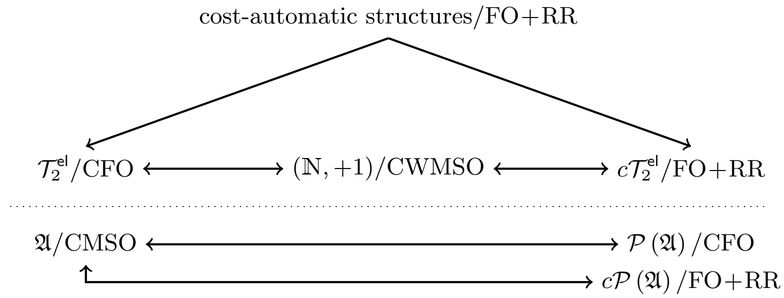

Fig. 1. Cost Logics Correspondences Overview

Theorem 17. *Logical formulas can be transformed in a semantics preserving way among* CFO *on* $\mathcal{T}_2^{\mathsf{el}}$, FO+RR *on* $c\mathcal{T}_2^{\mathsf{el}}$ *and* CWMSO *on* $\mathbb{N}^{+1}$.

The interpretations to transform $c\mathcal{T}_2^{\mathsf{el}}$ in $c\mathcal{P}_{\mathrm{w}}\left(\mathbb{N}^{+1}\right)$ and vice versa in the proof of the previous theorem also yield the following result as immediate consequence.

Corollary 18. $c\mathcal{P}_{\mathrm{w}}\left(\mathbb{N}^{+1}\right)$ *is a complete resource automatic structure.*

As an example application of Theorem 17, we settle the open question of the last section in [13]. Based on the results known at that time, it was unclear whether the full *bounded reachability problem* for pushdown systems with B-counters can be encoded in form of a CWMSO formula on $\mathcal{T}_2$. The problem asks, given a pushdown system with counter operations as in B-automata, and two regular sets A, B of pushdown configurations, whether there is a uniform upper bound $k \in \mathbb{N}$ such that it is possible to reach from all elements in A some element in B with a configuration sequence whose counter values are bounded by k. In [13] it was already established that the problem can be encoded in FO+RR over the configuration graph, which is a resource automatic structure. With Corollary 14 and Theorem 17 we obtain that the problem is expressible in CWMSO over $\mathbb{N}^{+1}$. Hence, it is also expressible in CWMSO over $\mathcal{T}_2$ since $\mathbb{N}^{+1}$ is CWMSO definable in $\mathcal{T}_2$. However, the argument here uses general properties of resource automatic structures and does not provide the insight of the direct formulation in CWMSO over $\mathcal{T}_2$ given in [13] for the special case of a single counter.

5 Conclusion

In this work we established connections among the different logics that arose around regular cost functions. Figure 1 provides an overview of the obtained results. We showed that the two quantitative first-order logics CFO and FO+RR are essentially equally expressive on the infinite binary tree with equal level predicate despite their rather different mechanisms for defining costs. Their expressive power on the tree could be summarized on an intuitive level by first-order queries plus the ability to count elements satisfying first-order properties. Furthermore, both formalisms provide another characterization for regular cost functions. The extension of this result to cost functions over tuples of words provided the insight that $c\mathcal{T}_2^{\mathsf{el}}$ is a complete structure for the class of resource automatic structures. These results nicely extend the classical results for regular languages and automatic structures, and can be seen as another sign that the notion of regular cost functions is a good quantitative generalization of regular languages.

In the second part, we showed extensions of the classical results that allow to exchange between MSO logic and FO logic over the power set structure. These results enabled particular transformations among CFO on $\mathcal{T}_2^{\mathsf{el}}$, FO+RR on $c\mathcal{T}_2^{\mathsf{el}}$ and CWMSO on $\mathbb{N}^{+1}$.

We are currently working on continuing this research in two directions. First, we want to extend the ideas of resource automatic structures towards resource tree automatic structures. This could lead to new decision methods for quantitative WMSO logics over the infinite tree. Second, we try to connect our results to the world of (W)MSO with the unbounding quantifier (for short (W)MSO+U) as introduced by Mikołaj Bojańczyk in [4].

References

1. Bárány, V., Kaiser, Ł., Rabinovich, A.: Expressing Cardinality Quantifiers in Monadic Second-Order Logic over Trees. Fundamenta Informaticae 100, 1–18 (2010)
2. Blumensath, A., Colcombet, T., Löding, C.: Logical theories and compatible operations. In: Logic and Automata. pp. 73–106 (2008)
3. Blumensath, A., Grädel, E.: Automatic structures. In: Logic in Computer Science, 2000. Proceedings. 15th Annual IEEE Symposium on. pp. 51–62. IEEE (2000)
4. Bojanczyk, M.: A bounding quantifier. In: CSL. pp. 41–55. LNCS, Springer (2004)
5. Colcombet, T.: Regular cost functions over words. Online Manuscript (2009)
6. Colcombet, T., Löding, C.: Transforming structures by set interpretations. Logical Methods in Computer Science 3(2) (2007)
7. Eilenberg, S., Elgot, C.C., Shepherdson, J.C.: Sets recognized by n-tape automata. J. Algebra 13, 447–464 (1969)
8. Elgot, C.C., Rabin, M.O.: Decidability and undecidability of extensions of second (first) order theory of (generalized) successor. Journal of Symbolic Logic 31(2), 169–181 (1966)
9. Grädel, E.: Finite Model Theory and Descriptive Complexity. In: Finite Model Theory and Its Applications, pp. 125–230. Springer (2007)
10. Khoussainov, B., Nerode, A.: Automatic presentations of structures. In: Logic and Computational Complexity, LNCS, vol. 960, pp. 367–392. Springer (1995)
11. Kuperberg, D.: Study of classes of regular cost functions. Ph.D. thesis, LIAFA Paris (December 2012)
12. Kuperberg, D.: Linear temporal logic for regular cost functions. In: STACS. LIPIcs, vol. 9, pp. 627–636. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2011)
13. Lang, M., Löding, C.: Modeling and verification of infinite systems with resources. Logical Methods in Computer Science 9(4) (2013)

A Proofs of Section 3

Proof (of Proposition 9). We provide an inductive translation over the structure of the formula. The only relevant case is the $\forall^{\leq N}$ operator. All other quantifiers and connectives can be translated directly. The translation uses the idea of the *longest common ancestor*. For two nodes $n_1, n_2 \in \{0,1\}^*$ in the tree, we define the longest common ancestor $\mathrm{lca}(n_1, n_2)$ to be the longest node n (in terms of word length) such that $n \preceq n_1$ and $n \preceq n_2$. Moreover, we define a formula $\varphi_{\mathrm{lca}}(z, x, y)$ to test if z is the lca of x and y by

$$\varphi_{\mathrm{lca}}(z,x,y) := z \preceq x \wedge z \preceq y \wedge \forall z'(z \neq z' \wedge z \preceq z') \rightarrow \neg(z' \preceq x \wedge z' \preceq y)$$

Now consider a formula $\psi(\bar{a}) = \forall^{\leq N} x \varphi(\bar{a}, x)$. Let $\tilde{\varphi}$ be the inductively given translation of φ. We define the translation (see below for a description of the idea, the definition of φ_{1at} can be found in the Proof of Theorem 12):

$$\tilde{\psi}(\bar{a}) := \exists e |e|_1 \wedge \forall x \forall y \left[(\exists p \varphi_{\mathrm{lca}}(p, x, y) \wedge \varphi_{1\mathrm{at}}(e, p)) \vee \tilde{\varphi}(\bar{a}, x) \vee \tilde{\varphi}(\bar{a}, y) \right]$$

The idea of this construction is as follows: Using the means of FO+RR on $c\mathcal{T}_2^{\mathsf{el}}$, we need to count the number of "exceptions" of the $\forall^{\leq N}$ operator. Since cost functions are only compared up to equivalence, this counting needs not to be exact. However, we need to ensure that the two values are related by some correction function (in this particular case the function is 2^n). For this purpose, we count the number of levels in $c\mathcal{T}_2^{\mathsf{el}}$ with a certain property. This number counting the levels, is defined by the element e: it counts each level on which the path to e goes to the right (captured by $|e|_1$). As a first approximation, consider the levels in the tree on which at least one exception occurs. However, this approximation fails if there are many exceptions on few levels. For this reason, we also count the levels on which the lca of two exceptions is located.

In the following, we show that this translation yields $[\![\varphi]\!]^{\mathcal{T}_2^{\mathsf{el}}} \approx_\alpha [\![\tilde{\varphi}]\!]^{c\mathcal{T}_2^{\mathsf{el}}}$ for $\alpha(n) = 2^n$. The only interesting part is the inductive step for the case $\psi(\bar{a}) = \forall^{\leq N} x \varphi(\bar{a}, x)$. The other cases follow directly from the identical inductive semantics in both logics. We start with $[\![\psi]\!]^{\mathcal{T}_2^{\mathsf{el}}} \preceq_\alpha [\![\tilde{\psi}]\!]^{c\mathcal{T}_2^{\mathsf{el}}}$. Let $[\![\psi(\bar{a})]\!]^{\mathcal{T}_2^{\mathsf{el}}} \leq n$. Then, we have $\mathcal{T}_2^{\mathsf{el}}, n \models \forall^{\leq N} x \varphi(\bar{a}, x)$. By definition of the operator $\forall^{\leq N}$, there is a set Z with $|Z| \leq n$ such that $\forall x \notin Z$ we have $\mathcal{T}_2^{\mathsf{el}}, n \models \varphi(\bar{a}, x)$. Let Z_t be the set of the lcas of all pairs of points in Z. Clearly, $|Z_t| \leq |Z|^2 \leq n^2$. Since Z_t is finite, we can find an element $e \in \{0,1\}^*$ that branches to the right in all levels in which Z_t contains an element. Formally, $|e| = \max_{t \in Z_t} |t|$ and $e(i) = 1$ iff $\exists t \in Z_t$ with $|t| = i$. For this e, we have $[\![|e|_1]\!]^{c\mathcal{T}_2^{\mathsf{el}}} \leq |Z_t| \leq n^2 \leq 2^n$. Now, let x and y be arbitrary elements of the tree. We distinguish two cases. First, at least one of x, y is not in Z (w.l.o.g. let $x \notin Z$). Then, we have $\mathcal{T}_2^{\mathsf{el}}, n \models \varphi(\bar{a}, x)$ and thus by induction hypothesis $[\![\tilde{\varphi}(\bar{a}, x)]\!]^{c\mathcal{T}_2^{\mathsf{el}}} \leq 2^n$. Consequently

$$[\![(\exists p \varphi_{\mathrm{lca}}(p, x, y) \wedge \varphi_{1\mathrm{at}}(e, p)) \vee \tilde{\varphi}(\bar{a}, x) \vee \tilde{\varphi}(\bar{a}, y)]\!]^{c\mathcal{T}_2^{\mathsf{el}}} \leq 2^n$$

because the semantics of $\vee$ is min and $\tilde{\varphi}(\bar{a}, x)$ is one of the disjuncts.

In the second case, we have $x, y \in Z$. However, by choice of e, we have that e is right-branching at the level of $\mathrm{lca}(x,y)$. Consequently, we have $[\![(\exists p \varphi_{\mathrm{lca}}(p,x,y) \wedge \varphi_{1\mathrm{at}}(e,p))]\!]^{c\mathcal{T}_2^{\mathsf{el}}} = 0$. Thus also

$$[\![(\exists p \varphi_{\mathrm{lca}}(p,x,y) \wedge \varphi_{1\mathrm{at}}(e,p)) \vee \tilde{\varphi}(\bar{a},x) \vee \tilde{\varphi}(\bar{a},y)]\!]^{c\mathcal{T}_2^{\mathsf{el}}} = 0 \leq 2^n$$

So we have seen that the first part of the translated formula $\tilde{\psi}$ is bounded by 2^n and the second part is also bounded by 2^n by all possible choices of x, y. Altogether, we obtain $[\![\tilde{\psi}(\bar{a})]\!]^{c\mathcal{T}_2^{\mathsf{el}}} \leq 2^n$, which concludes this direction of the proof.

Now, we prove the other direction. So, let $[\![\tilde{\psi}(\bar{a})]\!]^{c\mathcal{T}_2^{\mathsf{el}}} \leq n$. We claim that $\mathcal{T}_2^{\mathsf{el}}, 2^n \models \forall^{\leq N} x \varphi(\bar{a},x)$. That directly implies $[\![\psi(\bar{a})]\!]^{\mathcal{T}_2^{\mathsf{el}}} \leq 2^n$. Since the semantics is over a discrete domain with lower bound, there is an element e that assumes the infimum (from the semantics of the existential quantifier) and witnesses that $\tilde{\psi}$ is bounded by n. Fix such an e. We first notice that $[\![|e|_1]\!]^{c\mathcal{T}_2^{\mathsf{el}}} \leq n$. Moreover, the second part of $\tilde{\psi}$ yields that for every pair of elements x, y, we have either $[\![\tilde{\varphi}(\bar{a},x)]\!]^{c\mathcal{T}_2^{\mathsf{el}}} \leq n$ or $[\![\tilde{\varphi}(\bar{a},y)]\!]^{c\mathcal{T}_2^{\mathsf{el}}} \leq n$ or e is 1-branching in the level of $\mathrm{lca}(x,y)$. For all elements x with $[\![\tilde{\varphi}(\bar{a},x)]\!]^{c\mathcal{T}_2^{\mathsf{el}}} \leq n$, we obtain from the induction hypothesis that $\mathcal{T}_2^{\mathsf{el}}, 2^n \models \varphi(\bar{a},x)$ (‡). So, it suffices to show that there are at most 2^n elements x such that $[\![\tilde{\varphi}(\bar{a},x)]\!]^{c\mathcal{T}_2^{\mathsf{el}}} > n$. Let Z be the set of these x. Now, we consider the subtree $\mathcal{T}_{\mathrm{lca}}^{Z}$ that is induced by the elements $\{\mathrm{lca}(x,y) \mid x, y \in Z\}$, i.e., has these elements as universe and the descendant relation $\preceq$ inherited from $c\mathcal{T}_2^{\mathsf{el}}$.

First, we recognize that $Z \subseteq \mathcal{T}_{\mathrm{lca}}^{Z}$ and show that $\mathcal{T}_{\mathrm{lca}}^{Z}$ has branching degree 2. The major reason for this is that the $\mathcal{T}_{\mathrm{lca}}^{Z}$ is closed under taking lca, i.e., the lca of two lca elements is again in $\mathcal{T}_{\mathrm{lca}}^{Z}$ because it can be written as lca of the original elements in Z. For a detailed formal argument consider the following: Assume $\mathcal{T}_{\mathrm{lca}}^{Z}$ has a node n with at least 3 direct descendants n_1, n_2, n_3. Since $c\mathcal{T}_2^{\mathsf{el}}$ has only branching degree 2, there are at least two of n_1, n_2, n_3 in either the left of the right subtree induced by n (we write $n0{\downarrow}$ and $n1{\downarrow}$ for the two subtrees). W.l.o.g. let n_1, n_2 be in $n0{\downarrow}$. By the definition of $\mathcal{T}_{\mathrm{lca}}^{Z}$, there are nodes $n_1^1, n_1^2, n_2^1, n_2^2 \in Z$ such that $n_1 = \mathrm{lca}(n_1^1, n_1^2)$ and $n_2 = \mathrm{lca}(n_2^1, n_2^2)$. Moreover, not both of n_2^1 and n_2^2 can be in the subtree $n_1{\downarrow}$. Otherwise $n_1 \preceq \mathrm{lca}(n_2^1, n_2^2) = n_2$. W.l.o.g. let n_2^1 be not in the subtree $n_1{\downarrow}$. Then, we have $n0 \preceq n_1^1$ and $n0 \preceq n_2^1$ and thus also $n0 \preceq \mathrm{lca}(n_2^1, n_2^1)$. However, since n_2^1 is not in the subtree $n_1{\downarrow}$ and $\mathrm{lca}(n_2^1, n_2^1) \in \mathcal{T}_{\mathrm{lca}}^{Z}$, we have in total $n \prec n0 \preceq \mathrm{lca}(n_1^1, n_2^1) \prec n_1$. This is a contradiction to the assumption that n_1 is a direct child of n in $\mathcal{T}_{\mathrm{lca}}^{Z}$. Hence, $\mathcal{T}_{\mathrm{lca}}^{Z}$ has branching degree 2.

By construction, e is right-branching in all levels with a lca of elements from Z. Consequently, the tree $\mathcal{T}_{\mathrm{lca}}^{Z}$ has at most $[\![|e|_1]\!]^{c\mathcal{T}_2^{\mathsf{el}}} \leq n$ levels. Hence, we obtain that $|\mathcal{T}_{\mathrm{lca}}^{Z}| \leq 2^{[\![|e|_1]\!]^{c\mathcal{T}_2^{\mathsf{el}}}} \leq 2^n$ and thus $|Z| \leq |\mathcal{T}_{\mathrm{lca}}^{Z}| \leq 2^n$ by $Z \subseteq \mathcal{T}_{\mathrm{lca}}^{Z}$. This proves $\mathcal{T}_2^{\mathsf{el}}, 2^n \models \forall^{\leq N} x \varphi(\bar{a},x)$ since (‡) for all $x \notin Z$ and $|Z| \leq 2^n$. Altogether, $[\![\psi(\bar{a})]\!]^{\mathcal{T}_2^{\mathsf{el}}} \leq 2^n$. □

Proof (of Theorem 12). More detailed explanation:

Formally, we construct the formula in the following way. Let $Q = \{1, \ldots, n\}$, $\Gamma = \{1, \ldots, m\}$. We first define some helper formulas. Note that we use ε as a constant as it is definable.

$$\begin{aligned}
\varphi_{\text{between}}(x, u, v) &:= u \preceq x \wedge x \preceq v \\
\varphi_{\text{lvl}-\text{between}}(x, u, v) &:= \exists z \varphi_{\text{between}}(z, u, v) \wedge \mathsf{el}(z, x) \\
\varphi_{S1}(x) &:= \exists y S_1(y, x) \\
\varphi_{S0}(x) &:= \exists y S_0(y, x) \\
\varphi_{\text{at}}(x, y, l) &:= x \preceq y \wedge \mathsf{el}(x, l) \\
\varphi_{0\text{at}}(x, l) &:= \exists z \varphi_{\text{at}}(z, y, l) \wedge \varphi_{S0}(z) \\
\varphi_{1\text{at}}(x, l) &:= \exists z \varphi_{\text{at}}(z, y, l) \wedge \varphi_{S1}(z) \\
x \preceq_1 y &:= x \preceq y \wedge \exists z \mathsf{el}(z, y) \wedge \forall u((\varepsilon \preceq u \wedge u \preceq x) \to \varphi_{0\text{at}}(z, u)) \\
&\qquad (u \neq x \wedge x \preceq u \wedge u \preceq y) \to (\varphi_{1\text{at}}(z, u) \leftrightarrow \varphi_{1\text{at}}(y, u)) \wedge |z|_1 \\
\varphi_{0\text{between}}(x, u, v) &:= \forall z(u \preceq z \wedge z \preceq v) \to \varphi_{0\text{at}}(x, z) \\
\varphi_{\text{part}}(x_1, \ldots, x_k, w) &:= \forall y \varphi_{\text{between}}(y, \varepsilon, w) \to \\
&\qquad \Big[(\bigvee_{i=1}^{k} (\varphi_{1\text{at}}(x_i, y)) \wedge \neg (\bigwedge_{i \neq j} \varphi_{1\text{at}}(x_i, y) \wedge \varphi_{1\text{at}}(x_j, y) \Big] \\
\varphi_{\text{cval}}(c^{\mathtt{i}}, c^{\mathtt{r}}, w) &:= \forall u \forall v(u \preceq v \wedge v \preceq c^{\mathtt{i}} \wedge \varphi_{0\text{between}}(c^{\mathtt{r}}, u, v)) \to u \preceq_1 v
\end{aligned}$$

Moreover, we define a family of formulas to that can verify whether a certain transition $\delta = (i, k, j, f)$ is applied on a certain level l. As a parameter this formula gets all positions p_i for the states, all counter positions $c_i^{\mathtt{i}}, c_i^{\mathtt{r}}$ and the input word w.

$$\begin{aligned}
\psi_\delta(l) &:= \exists l'(S_0(l', l) \vee S_1(l', l)) \wedge \varphi_{1\text{at}}(p_i, l') \wedge \varphi_{1\text{at}}(p_j, l) \wedge \varphi_{k\text{at}}(w, l) \\
&\wedge \left(\bigwedge_{\substack{r=1 \\ f(r) \neq \mathtt{r}}}^{m} \varphi_{0\text{at}}(c_r^{\mathtt{r}}) \right) \wedge \left(\bigwedge_{\substack{r=1 \\ f(r) = \mathtt{r}}}^{m} \varphi_{1\text{at}}(c_r^{\mathtt{i}}) \right)
\end{aligned}$$

Additionally, we need a bootstraping variant for the first level that does not check that the previous state is correct.

$$\begin{aligned}
\hat{\psi}_\delta(l) &:= \wedge \varphi_{1\text{at}}(p_j, l) \wedge \varphi_{k\text{at}}(w, l) \\
&\wedge \left(\bigwedge_{\substack{r=1 \\ f(r) \neq \mathtt{r}}}^{m} \varphi_{0\text{at}}(c_r^{\mathtt{r}}) \right) \wedge \left(\bigwedge_{\substack{r=1 \\ f(r) = \mathtt{r}}}^{m} \varphi_{1\text{at}}(c_r^{\mathtt{i}}) \right)
\end{aligned}$$

We remark here that it suffices to ensure that the run has no resets at positions where there are no resets possible and that it has increments whenever the transition dictates an increment. The other implications can be neglected because the existential guessing of the parameters in order to obtain the smallest result.

We now define as a last intermediate step a run checker formula that checks if a guessed run is valid and accepting. It also gets all the parameters mentioned above.

$$\begin{aligned}\Theta := \exists l \exists l' S_0(\varepsilon, l) \wedge S_0(l, l') \wedge & \bigvee_{\substack{\delta=(i,j,j,f)\in\Delta \\ i\in\mathsf{In}}} \hat{\psi}_\delta(l) \\ \wedge & \left(\forall z \varphi_{\text{lvl}-\text{between}}(z, l', w) \rightarrow \bigvee_{\delta\in\Delta} \psi_\delta(z)\right) \\ \wedge & \left(\bigvee_{i\in\mathsf{Fin}} \varphi_{\text{1at}}(p_i, w)\right)\end{aligned}$$

The total formula $\varphi(w)$ is then obtained by:

$$\begin{aligned}\varphi(w) := \exists p_1 \ldots \exists p_n \exists c_1^{\mathtt{i}} \exists c_1^{\mathtt{r}} \ldots \exists c_m^{\mathtt{i}} \exists c_m^{\mathtt{r}} & \left(\bigwedge_{i=1}^{n} \mathsf{el}(p_i, w)\right) \wedge \left(\bigwedge_{i=1}^{m} \mathsf{el}(c_i^{\mathtt{i}}, w) \wedge \mathsf{el}(c_i^{\mathtt{r}}, w)\right) \\ & \wedge \varphi_{\text{part}}(p_1, \ldots, p_n, w) \\ & \wedge \Theta \\ & \wedge \left(\bigwedge_{i=1}^{m} \varphi_{\text{cval}}(c_i^{\mathtt{i}}, c_i^{\mathtt{r}}, w)\right)\end{aligned}$$

Note that only the last line of the formula makes real use of the quantitative predicate $\preceq_1$. The upper lines guess a run and would return the value 0 if the run is valid and ∞ otherwise. Consequently, taking the max with the counter values induced by this run results in ∞ for invalid runs and in the maximal counter value over all counters (as desired) in the case of valid runs. ☐

B Proofs of Section 4

Proof (of Proposition 15). As a preparation for the proof we first establish a technical result that allows us to approximate the number of sets by the size of another set. For this, let F be a set of sets. We will show that $|F| \approx_\alpha |X_F|$ for $\alpha(n) := 2^n$ and a set $X_F \subseteq \bigcup F$. Let X_F be a maximal set that satisfies the following property:

$$(\forall X \in F \exists x \in X x \in X_F) \wedge (\forall x \in X_F \forall y \in X_F x \neq y \rightarrow \exists Z \in F x \in Z \leftrightarrow y \notin Z)$$

We call such an X_F a *witness set* for F. The idea behind the set X_F is that it contains elements from all sets but all elements have pairwise different membership vectors with respect to the sets in F. W.l.o.g. let $|F| < \infty$. The case of

$|F| = \infty$ follows from considering an increasing chain of finite subsets from F. Formally, let $F = \{F_1, \dots F_n\}$. A membership vector v^x for some $x \in X_F$ is a $|F|$ dimensional vector over $\{0,1\}$ such that $v^x_i = 1$ iff $x \in F_i$, i.e., the vector indicates in which sets from F the element x is. We claim ($\star$) that $|F| \approx_\alpha |X_F|$.

First, we show that $|X_F| \leq 2^{|F|}$. By definition of X_F, all elements in X_F have pairwise different membership vectors. However, there are only $2^{|F|}$ different membership vectors of dimension $|F|$.

Now, we show that $|F| \leq 2^{|X_F|}$. For this, we define the equivalence relation $\sim_F$ on elements of F_i by $x \sim_F y$ if $v^x = v^y$. Now, we consider the factors of the sets in F w.r.t. $\sim_F$: ${}^{F}\!/_{\sim_F} := \{{}^{F_1}\!/_{\sim_F}, \dots, {}^{F_n}\!/_{\sim_F}\}$ and claim that $|F| = |{}^{F}\!/_{\sim_F}|$. Clearly $|F| \geq |{}^{F}\!/_{\sim_F}|$, so assume $|F| \gneqq |{}^{F}\!/_{\sim_F}|$. Then there are $i \neq j$ such that ${}^{F_i}\!/_{\sim_F} = {}^{F_j}\!/_{\sim_F}$ although $F_i \neq F_j$. So there is an element f in the symmetric difference of F_i and F_j. W.l.o.g. $f \in F_i, f \notin F_j$. Consequently, we have $v^f_i = 1$ but $v^f_j = 0$. Since for all $x \in F_j$ we have $v^x_j = 1$, we have that there is no $x' \in F_j$ such that $x' \sim_F f$. Thus, we obtain a contradiction because $[f]_{\sim_F} \notin {}^{F_j}\!/_{\sim_F}$ and $[f]_{\sim_F} \in {}^{F_i}\!/_{\sim_F}$. Moreover, we notice that X_F is a set of representative for all $\sim_F$ classes (because we require that the set is maximal with the given condition). We identify the classes with elements from X_F and obtain that ${}^{F_i}\!/_{\sim_F} \subseteq X_F$ or ${}^{F_i}\!/_{\sim_F} \in \mathcal{P}(X_F)$. In total:

$$|F| = |{}^{F}\!/_{\sim_F}| \leq 2^{|X_F|}$$

Next using the above result we prove the proposition. Observe that in C(W)MSO one can define the predicates $\mathrm{Sing}(X)$ and $X \subseteq Y$ which hold if X is singleton and X is a subset of Y respectively. We identify these predicates by their respective definitions in the logic. Assume we are given a CFO formula $\varphi(x_1, \dots, x_j)$ over the powerset structure $\mathcal{P}(\mathfrak{A})$. In $\varphi(x_1, \dots, x_j)$ we replace each first order variable x in φ by a second order variable X, each formula $\mathrm{Sing}(x)$ by $\mathrm{Sing}(X)$, each $x \subseteq y$ by $X \subseteq Y$ to obtain the formula $\psi(X_1, \dots, X_j)$. For a subformula $\forall^{\leq N} X \phi(X)$ of $\psi(X_1, \dots, X_j)$ on $\mathfrak{A}$ let n the least number such that there are at most n subsets of A for which $[\![\phi(X)]\!]^{\mathcal{P}(A)} > n$. With this definition of the semantics of ψ, it is clear that $[\![\varphi(x_1, \dots, x_j)]\!]^{\mathcal{P}(\mathfrak{A})} = [\![\psi(X_1, \dots, X_j)]\!]^{\mathfrak{A}}$. Hence, it suffices to show that for every formula $\forall^{\leq N} X \phi(X)$ there is a cost-equivalent C(W)MSO formula. Let $\zeta = \forall X_\phi \ (|X_\phi| \leq N \vee \zeta'(X_\phi))$ where

$$\zeta'(X_\phi) := \exists x \in X_\phi \exists y \in X_\phi \forall X \ \phi(X) \vee (x \neq y \wedge (x \in X \leftrightarrow y \in X))$$

The formula says the following: if $\mathfrak{A}$ satisfies ζ with value N, then for any set X either X is of size at most N, or X is not a witness set for the family $F = \{A' \subseteq A \mid \mathfrak{A}, N \not\models \phi(A')\}$. The second disjunct is stated in terms of the presence of two elements x and y in X such that they have the same membership vector for the elements in F. Next we prove that $\forall^{\leq N} X \phi(X)$ and ζ are cost equivalent. Assume $\mathfrak{A}, n \models \forall^{\leq N} X \phi(X)$. We claim that $[\![\zeta]\!]^{\mathfrak{A}} \leq \alpha(n)$. To see this, take F^α_ϕ to be the family $F^\alpha_\phi = \{A' \subseteq A \mid \mathfrak{A}, \alpha(n) \not\models \phi(A')\}$. A set $Y \subseteq A$ satisfies ζ' iff there exist two elements in Y which have the same membership vector for F^α_ϕ

iff Y does not have a superset which is a witness. Consequently, ζ' has value 0 for all sets Y that are not extensible to a witness set and ∞ otherwise. So, $[\![\zeta]\!]^{\mathfrak{A}}$ is determined by $|Y| \leq N$ for the case that Y is extensible to a witness set for F. These sets are bounded by the size of witness sets for F^{α}_{ϕ} which is by $(\star)$ bounded by $\alpha(n)$.

Conversely, let now $\mathfrak{A}, n \models \zeta$, then for all sets $X \subseteq A$ it is the case that either X is of size at most n or X is not a witness set for $F_\phi = \{A' \subseteq A \mid \mathfrak{A}, n \not\models \phi(A')\}$ with the same ideas as in the previous direction. Continuing, we deduce that all witness sets for F_ϕ are of size at most n. This implies that F_ϕ itself is of size at most $\alpha(n)$ (by claim $(\star)$). Hence, $\mathfrak{A}, \alpha(n) \models \forall^{\leq N} X\, \phi(X)$.

Note that $\phi(X)$ appears positively in the formula ζ. Therefore, we can inductively replace each quantification of the form $\forall^{\leq N} X \phi(X)$ in $\psi(X_1, \ldots, X_j)$ with an equivalent formula ζ to obtain a formula in C(W)MSO. Call it $\psi_1(X_1, \ldots, X_j)$. A structural induction on the formulas shows that

$$[\![\psi(X_1, \ldots, X_j)]\!]^{\mathfrak{A}} \approx_\alpha [\![\psi_1(X_1, \ldots, X_j)]\!]^{\mathfrak{A}}$$

This proves the direction from CFO to C(W)MSO.

Next we treat the other direction, that is, from a C(W)MSO formula over structure to a CFO formula over the (weak) powerset structure. Let $\mathfrak{A}'$ be the structure $\mathfrak{A}' = (A, \mathrm{Sing}, \subseteq, \mathcal{P}(R_1), \ldots, \mathcal{P}(R_k))$ where Sing and $\subseteq$ are predicates on subsets of A. Formulas over $\mathfrak{A}'$ which uses only second order variables are said to be in normal form. Observe that for every C(W)MSO formula $\varphi(X_1, \ldots, X_j, y_1, \ldots, y_k)$ there is an equivalent formula $\varphi'(X_1, \ldots, X_j, Y_1, \ldots, Y_k)$ in normal form (and vice versa) such that for all sets $X_1 \ldots, X_j \subseteq A$ and elements $y_1, \ldots, y_k \in A$,

$$[\![\varphi(X_1, \ldots, X_j, y_1, \ldots, y_k)]\!]^{\mathfrak{A}} = [\![\varphi'(X_1, \ldots, X_i, \{y_1\}, \ldots, \{y_k\})]\!]^{\mathfrak{A}'}.$$

Let $\varphi'(X_1, \ldots, X_j)$ be a formula in normal form and $\varphi_1(x_1, \ldots, x_j)$ is obtained from φ' by replacing every second order variable by a first order variable. Then

$$[\![\varphi'(X_1, \ldots, X_j)]\!]^{\mathfrak{A}'} = [\![\varphi_1(x_1, \ldots, x_j)]\!]^{\mathcal{P}(\mathfrak{A})}$$

It remains to eliminate all predicates of the form $\psi(x) = |x| \leq N$ from φ_1. For every $\psi(x)$ in φ_1 we substitute $\psi_1(x) = \forall^{\leq N} y\, y \not\subseteq x$. First of all observe that since $\psi(x)$ occurs positively in φ_1 after substitution $\psi_1(x)$ occurs positively as well. Next observe that $\mathfrak{A}', n \models \psi(x)$ iff $\mathfrak{A}', \alpha(n) \models \psi_1(x)$. This concludes the claim.

To prove the second correspondence with the cost poweset structure, we need to translate a C(W)MSO formula φ over $\mathfrak{A}$ to an FO+RR formula over $c\mathcal{P}(\mathfrak{A})$. For this, it suffices to transform the predicate $|x| \leq N$ into the predicate size(x). Similarly every FO+RR formula is translated to back a C(W)MSO formula by replacing each size(x) to $|x| \leq N$. □

Proof (of Theorem 17). We first notice that it suffices to connect FO+RR on $c\mathcal{T}_2^{\mathsf{el}}$ with CWMSO on $\mathbb{N}^{+1}$ since the equivalence of FO+RR on $c\mathcal{T}_2^{\mathsf{el}}$ with CFO on $\mathcal{T}_2^{\mathsf{el}}$ was already shown in Proposition 9. First, we use Proposition 15 to translate CWMSO formulas on $\mathbb{N}^{+1}$ into FO+RR on $c\mathcal{P}_{\mathrm{w}}\left(\mathbb{N}^{+1}\right)$. In a second step we show that $c\mathcal{P}_{\mathrm{w}}\left(\mathbb{N}^{+1}\right)$ and $c\mathcal{T}_2^{\mathsf{el}}$ are mutually FO+RR-interpretable in each other. For both directions we use our definition of $\mathcal{T}_2$ where nodes are strings from $\{0,1\}^*$. Additionally, finite subsets of $\mathbb{N}$ can also be represented in form of strings from $\varepsilon + \{0,1\}^*1$ based on the idea that position i in the string is 1 iff i is in the set. First, we interpret $c\mathcal{P}\left(\mathbb{N}^{+1}\right)$ in $c\mathcal{T}_2^{\mathsf{el}}$. We define $\delta(x)$ such that it is true for all nodes from $\varepsilon + \{0,1\}^*1$. We represent $\mathrm{Sing}(x)$ by $\exists y S_1 yx \wedge \forall z (z \precneqq x) \rightarrow (z = \varepsilon \vee \exists z' S_0 z'z)$ and $\mathrm{Size}(x)$ by $\varepsilon \preceq_1 x$. Moreover, one can encode $\mathrm{Succ}(x,y)$ by $\mathrm{Sing}(x) \wedge \exists p \exists x' S_1 px \wedge S_0 px' \wedge S_1 x'y$.

For the converse direction we encode every node t from $\mathcal{T}_2$ by the set represented by the string $t1$. Accordingly, we define $\delta(X)$ to select all non-empty sets. From literature it is known that the transitive closure of Succ is definable in WMSO. We use this fact and write $X \leq Y$ as a shorthand for this. Moreover, we use $\varphi_{\max}(X,Y) = X \subseteq Y \wedge \mathrm{Sing}(X) \wedge \forall X' (\mathrm{Sing}(X') \wedge X' \neq X \wedge X \leq X') \rightarrow \neg (X' \subseteq Y)$. With this we first define the prefix relation by $X \preceq Y := \exists X' \exists Y' \varphi_{\max}(X', X) \wedge \varphi_{\max}(Y', Y) \wedge X' \leq Y' \wedge \forall Z (\mathrm{Sing}(Z) \wedge Z \subseteq X \wedge Z \neq X') \rightarrow Z \subseteq Y$. In a very similar way, we realize the counting prefix relation: $X \preceq_1 Y := X \preceq Y \wedge \exists X' \exists Y' \varphi_{\max}(X', X) \wedge \varphi_{\max}(Y', Y) \wedge X' \leq Y' \wedge \exists S\, \mathrm{Size}(S) \wedge \forall Z (\mathrm{Sing}(Z) \wedge X' \leq Z \wedge Z \neq Y' \wedge Z \leq Y' \wedge Z \subseteq Y) \rightarrow Z \subseteq S$ □