An Implicit Characterization of the Polynomial-Time Decidable Sets by Cons-Free Rewriting

Daniel de Carvalho and Jakob Grue Simonsen

Department of Computer Science, University of Copenhagen (DIKU)
Njalsgade 128-132, DK-2300 Copenhagen S, Denmark
{daniel.carvalho,simonsen}@diku.dk

Abstract. We define the class of *constrained cons-free* rewriting systems and show that this class characterizes P, the set of languages decidable in polynomial time on a deterministic Turing machine. The main novelty of the characterization is that it allows very liberal properties of term rewriting, in particular non-deterministic evaluation: no reduction strategy is enforced, and systems are allowed to be non-confluent.

We present a class of constructor term rewriting systems that characterizes the complexity class P—the set of languages decidable in polynomial time on a deterministic Turing machine. The class is an analogue of similar classes in functional programming that use *cons-freeness*–the inability of a program to construct new compound data during its evaluation–to characterize a range of complexity classes, including L and P [1, 2], and for higher-order programs $PSPACE$ and hierarchies of exponential space and time classes [3]. The primary novelty is that while previous work has crucially utilized the deterministic evaluation (in particular, call-by-value) and typing disciplines usually found in functional programming languages, we allow for the full rewriting relation to be used, and we allow non-orthogonal systems.

The ability to use non-orthogonal and non-confluent systems means that we do not have access to standard results on orthogonality such as normalization or finite developments of sets of redexes, and we cannot appeal to results connecting deterministic Turing machines to confluent rewriting [4], or to functional programming without overlapping function declarations [1, 3, 5]. These are the main reasons that our proofs are substantially more difficult than similar work by Bonfante showing that introducing non-determinism to a cons-free functional language characterizes P [2].

Related work

The original impetus for devising languages or calculi characterizing complexity classes was the seminal work of Bellantoni and Cook [6] who introduced a scheme of constrained recursion in function declarations in applicative languages, called *safe recursion*, later followed by similar constraints, *tiered* or *ramified recursion*

[7, 5]. Roughly, the idea of this approach is to partition the arguments of every function into "normal" and "safe" variables, where only normal variables are used for recursion. Our approach contains no such constraints. Other approaches have used type systems, typically based on variants of linear logic [8–10]; in contrast, we employ no type system, but enforce a simple syntactic criterion to constrain copying.

Much effort has been directed towards performing *polynomial complexity analysis* in term rewriting, that is, devising methods to automatically infer that specific TRSs have polynomial runtime or derivational complexity. This work has almost invariably considered analogues of call-by-value semantics, e.g. innermost evaluation; in this vein of research, several reduction orders have been defined such that TRSs are compatible with the orders iff they have polynomial runtime complexity [11, 12]. The main difference with our work is that we do not necessarily enforce polynomial runtime complexity, but use a form of memoization to ensure that our class of systems can be evaluated in polynomial time on a Turing machine. For full rewriting with no constraints on reduction strategy, Avanzini and Moser [4] have shown that a *confluent* constructor rewriting system characterizes a language in P iff it has polynomial runtime complexity, that is, if the maximal reduction lengths starting from appropriately formed terms are polynomially bounded. Most research in this vein has focused on *functional* complexity classes, whereas we only consider the case of decision problems; we believe our results can be extended to the function classes, but with some difficulty as input constructors may not be used as output constructors in cons-free systems.

The restriction to cons-free systems was originally developed in functional programming by Jones [1], [3] inspired by similar work by Goerdt in recursion-theoretic settings [13, 14], and leading to similar characterizations in other language paradigms [15], [2], [16]. The primary difference between this work and ours is that we do not consider a particular reduction order, and work in a completely untyped setting, that is, the standard liberal setting of term rewriting; the cost of this freedom is that we need to enforce technical demands on our class of systems leading to *constrained* cons-free systems, rather than merely cons-free ones.

1 Constrained cons-free term rewriting systems

We presuppose basic knowledge about rewriting, corresponding to the introductory chapters of [17]. Throughout the text, we assume a denumerable set $\mathcal{X}$ of *variables*.

Let Σ be a signature (i.e., a function from a set $\mathcal{F}$ of *function symbols* to $\mathbb{N}$ which associates with every $f \in \mathcal{F}$ its arity $ar(f)$); we then denote by $\mathcal{T}(\Sigma)$ the set of terms built from Σ and $\mathcal{X}$. The set of ground terms over Σ is denoted by $\mathcal{T}_0(\Sigma)$. By abuse of language, if $\mathcal{F}_0$ is a set of function symbols, we will write also $\mathcal{T}(\mathcal{F}_0)$ instead of $\mathcal{T}(\Sigma|_{\mathcal{F}_0})$ (or $\mathcal{T}_0(\mathcal{F}_0)$ instead of $\mathcal{T}_0(\Sigma|_{\mathcal{F}_0})$). The set of positions in a term t is denoted by $Pos(t)$ (a position q is said to be below the position p if $p \leq q$): if p is a position in a term t, then p determines the subterm $t_{|p}$ of t occurring at position p and the symbol $t(p)$ occurring in t at p. If s and t are terms, we write $s \trianglelefteq t$ (resp. $s \triangleleft t$) if s is a subterm of t (resp. if s is a subterm of t and $s \neq t$); note that $s \trianglelefteq t$ iff $(\exists p \in Pos(t))s = t_{|p}$. For any term t, we denote by $Oc(t)$ the set of variables occurring in t, that is, $Oc(t) = \{x \in \mathcal{X};\ (\exists p \in Pos(t))t(p) = x\}$ and, for any $x \in \mathcal{X}$, by $Oc(x,t)$ the number of occurrences of x in t, that is, $Oc(x,t) = Card(\{p \in Pos(t);\ t(p) = x\})$.

A *constructor TRS* is a term rewriting system (TRS) in which the set of function symbols $\mathcal{F}$ is partitioned into a set $\mathcal{D}$ of *defined* function symbols and a set $\mathcal{C}$ of *constructors*, such that for every rewrite rule $(l,r) \in \mathcal{R}$, the left-hand side l has the form $f(t_1,\ldots,t_n)$ with $f \in \mathcal{D}$ and $t_1,\ldots,t_n \in \mathcal{T}(\mathcal{C})$, the set of terms built from variables and constructors.

We introduce *cons-free TRS* that corresponds essentially to the functional programming language called "F+ro" (ro for "read-only") in [18].

Definition 1. *A* cons-free TRS *is a finite constructor TRS such that, for every rewrite rule (l,r), for any $c(u_1,\ldots,u_n) \trianglelefteq r$ such that c is a constructor, we have $c(u_1,\ldots,u_n) \trianglelefteq l$ or $c(u_1,\ldots,u_n) \in \mathcal{T}_0(\mathcal{C})$.*

The functional programming languages considered in [3] and in [18] have a call-by-value semantics, and proofs generally assume terminating programs; in contrast, terms in (cons-free) term rewriting systems may be subjected to different reduction strategies, are not necessarily terminating, and terms may have more than one normal form. To obviate technical problems due to these facts, we restrict the class of term rewriting systems to the *constrained* cons-free term rewriting systems.

Definition 2. *A cons-free TRS R is said to be* constrained *if there exists some subset $\mathcal{A} \subseteq \mathcal{D}$ such that, for any rule $(f(c_1,\ldots,c_q),r) \in \mathcal{R}$ and for any $x \in \mathcal{X}$ such that $x \not\triangleleft c_1,\ldots c_q$, we have:*

- $(\forall p,p' \in Pos(r))(r(p) = x \Rightarrow (p' < p \Rightarrow r(p') \in \mathcal{A}))$
- *and* $f \in \mathcal{A} \Rightarrow Oc(x,r) \leq 1$.

Every variable occurring just below the root symbol of a left-hand side of a rule occur only below defined symbols of a certain kind that do not allow for non-linear recursion. Note that duplication may occur in constrained cons-free TRSs, both for variables that occur "deep" in a left-hand side (i.e., below constructor symbols), and for variables occurring just below the root of defined symbols not in the special subset $\mathcal{A} \subseteq \mathcal{D}$. E.g., if $f/1, g/2 \in \mathcal{D}$ and $c/1 \in \mathcal{C}$, the TRS $\{f(c(x)) \to g(x,x), g(c(x),y) \to y, g(c(x),c(y)) \to g(x,y)\}$ is constrained cons-free (set $\mathcal{A} = \{g\}$ or $\mathcal{A} = \{f,g\}$).

In Sections 2 and Section 3, we will prove some properties of cons-free term, respectively constrained cons-free TRSs that will allow for efficient simulation on Turing machines. The main aim of the two sections is to prove Proposition 1, respectively Corollary 1.

2 Computation in cons-free TRS

In this section, we introduce a class of "generalized terms", and we show that any reduction sequence in a cons-free TRS from a ground term to a ground constructor term can be simulated by some "innermost" reduction sequence of such "generalized terms" (i.e. some sequence of $\triangleright$-reductions).

We are given a cons-free TRS with $\mathcal{R}$ the set of rules, $\mathcal{D}$ the set of defined function symbols, and $\mathcal{C}$ the set of constructors. Moreover, for any $m \in \mathbb{N}$, we denote by $\mathcal{D}_m$ the set of defined function symbols of arity m and by $\mathcal{C}_m$ the set of constructors of arity m. We first set notations used in the remainder of the paper.

Notations: As usual, the reflexive transitive closure of a relation E is denoted by E^*. Throughout the text, $A \rightharpoonup B$ refers to the type of partial maps with domain A and co-domain B. If $f : A \rightharpoonup B$, we denote by $dom(f)$ the set of $x \in A$ such that $f(x)$ is defined and by $im(f)$ the set $f(dom(f)) = \{f(x) : x \in dom(f)\}$.

For any $t \in \mathcal{T}(\mathcal{D} \cup \mathcal{C})$, we denote by $|t|$ the *size* of t, i.e., $|x| = |c| = 1$ for all variables x and nullary $c \in \mathcal{D}_0 \cup \mathcal{C}_0$, and $|f(s_1, \ldots, s_m)| = 1 + \sum_{i=1}^{m} |s_i|$ for $f \in \mathcal{D}_m \cup \mathcal{C}_m$.

Let $u, v, t \in \mathcal{T}(\mathcal{D} \cup \mathcal{C})$. We denote by $\mathbf{Seq}(u, v)$ the set of (finite) reduction sequences from u to v and we set $\mathbf{Seq}(u, -) = \bigcup_{v \in \mathcal{T}(\mathcal{D} \cup \mathcal{C})} \mathbf{Seq}(u, v)$. If $\rho_1 \in \mathbf{Seq}(t, u)$ and $\rho_2 \in \mathbf{Seq}(u, v)$, then we denote by $(\rho_1;\ \rho_2)$ the reduction sequence from t to v consisting in ρ_1 followed by ρ_2.

For any reduction step $\rho : t \rightarrow_{C_0[],(l,r)} u$, for any occurrence $\langle v|C[]\rangle$ of v in $t = C[v]$, we denote by $\langle v|C[]\rangle \setminus \rho$ the set of descendants of $\langle v|C[]\rangle$ in u after ρ.

We denote by $\mathcal{U}_0$ the set of ground terms that may be written as $C[t_1, \ldots, t_n]$ for some $n \in \mathbb{N}$ where $C[\cdot, \ldots, \cdot]$ is an n-hole context over $\mathcal{D}$, and $t_1, \ldots, t_n$ are ground terms over $\mathcal{C}$. Notice that, if $v \in \mathcal{U}_0$ and $v \rightarrow u$ in some cons-free TRS, then $u \in \mathcal{U}_0$.

For any $i \in \mathbb{N}$, we denote by Φ_i the set of i-hole contexts obtained by substituting exactly i distinct occurrences of constants in an element of $\mathcal{U}_0$ such that, for any hole, the unique path from the root to the hole passes through only elements of $\mathcal{D}$.

Recall that a semi-ring is an algebraic structure $(R, \cdot, +)$ satisfying the standard ring axioms with the exceptions that every element need not have a $+$-inverse. Recall further that a semi-module is an algebraic structure satisfying the usual module axioms over a commutative semi-ring. We denote by $\mathbf{2}$ the semi-ring with exactly two elements 0 and 1, where $1 + 1 = 1$.

Let $\mathcal{E}$ be some set. We denote by $\mathbf{2}\langle\mathcal{E}\rangle$ the free $\mathbf{2}$-semi-module on $\mathcal{E}$. For any $V \in \mathbf{2}\langle\mathcal{E}\rangle$, we denote by $Supp(V)$ the unique $\mathcal{F} \subseteq \mathcal{E}$ such that $V = \sum_{v \in \mathcal{F}} v$. If $\mathcal{F} = \{v\}$, then we still denote by v the vector $\sum_{v \in \mathcal{F}} v \in \mathbf{2}\langle\mathcal{E}\rangle$.

We will use the notation $\mathbf{2}\langle\mathcal{E}\rangle$ either with $\mathcal{E} = \mathcal{T}_0(\mathcal{C})$ or $\mathcal{E} = \Delta$, the set of "generalized terms" defined just below. In those cases, an element of $\mathbf{2}\langle\mathcal{E}\rangle$ may be thought of as a "formal sum" of (generalized) terms, and $Supp(V)$ as the set of (generalized) terms occurring in the sum. A benefit of considering formal sums instead of finite sets is that it allows to painlessly identify a term with the singleton containing this term. In later developments, we shall use the sum to track the possible reducts of subterms, i.e. each summand will correspond to a possible reduct.

Definition 3. *For any $i \in \mathbb{N}$, we define Δ_i by induction on i:*

- $\Delta_0 = \mathcal{T}_0(\mathcal{C})$;
- $\Delta_{i+1} = \Delta_i \cup \left(\bigcup_{m\in\mathbb{N}}\{f(U_1,\ldots,U_m);\ f \in \mathcal{D}_m \text{ and } U_1,\ldots,U_m \in \mathbf{2}\langle\Delta_i\rangle\}\right)$.

We set $\Delta = \bigcup_{i\in\mathbb{N}} \Delta_i$. For any $u \in \Delta$, we set level$(u) = \min\{i \in \mathbb{N};\ u \in \Delta_i\}$.

Thus, e.g., if $\mathcal{D} = \{f/1, g/2\}$ and $\mathcal{C}' = \{s/1, n/0\}$, then $f(s(n) + s(s(n))) \in \Delta_1$ and $g(f(s(n)) + f(s(s(n))), s(n) + s(s(n))) \in \Delta_2$. Note further that $\mathcal{U}_0 \subseteq \Delta$ and every term on the form $C[c_1,\ldots,c_m]$, where $C[\cdot,\ldots,\cdot]$ is an m-hole context over $\mathcal{D}$ and $C_1,\ldots,C_m \in \mathbf{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$, is an element of Δ.

Now, we want to define a notion of reduction on $\mathbf{2}\langle\Delta\rangle$: we will denote this reduction by $\rhd$. First, we define an auxiliary binary relation $\rhd_\Delta$. For any $r \in \mathcal{T}(\mathcal{D}\cup\mathcal{C}_0)$, we homomorphically extend the notation r^φ with $\varphi : \mathcal{X} \rightharpoonup \mathcal{T}(\mathcal{D}\cup\mathcal{C}_0)$ to any $\varphi : \mathcal{X} \rightharpoonup \mathbf{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$ such that $Oc(r) \subseteq dom(\varphi)$: instead of having $r^\varphi \in \mathcal{T}(\mathcal{D}\cup\mathcal{C}_0)$, we have $r^\varphi \in \mathbf{2}\langle\Delta\rangle$.

Definition 4. *We define the relation $\rhd_\Delta \subseteq (\Delta_1 \setminus \Delta_0) \times \mathbf{2}\langle\Delta\rangle$ as follows: $u \rhd_\Delta V$ if, and only if, there exist $q \in \mathbb{N}$, $f \in \mathcal{D}_q$, $(f(c_1,\ldots,c_q), r) \in \mathcal{R}$, $V_1,\ldots,V_q \in \mathbf{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$ and $\varphi : \mathcal{X} \rightharpoonup \mathbf{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$ such that, for any $j \in \{1,\ldots,q\}$, we have*

- $Oc(c_j) \subseteq dom(\varphi)$ *and* $(c_j \notin \mathcal{X} \Rightarrow (\forall x \in Oc(c_j))\varphi(x) \in \mathcal{T}_0(\mathcal{C}))$;
- *and* $V_j = c_j{}^\varphi$

and $u = f(V_1,\ldots,V_q)$ and $V = r^\varphi$.

If $u \rhd_\Delta V$, then U may be replaced by V inside a one-hole generalized context $C[]$; this gives rise to a reduction step $C[u] \rhd_{C[]} C[V]$. We also write $C[u] \rhd_{C[]} C[0]$, i.e. whenever u is erased. However, in this last case, we will not count this step when we define the length of $\rhd$-reductions (see Definition 6). The set of one-hone generalized contexts is denoted by Θ_1 and is defined by setting $\Theta_1 = \bigcup_{i\in\mathbb{N}} \Delta_i^\square$, where $\Delta_i^\square$ is defined by induction on i:

- $\Delta_0^\square = \{U + \square;\ U \in \mathbf{2}\langle\Delta\rangle\}$;
- $\Delta_{i+1}^\square = \bigcup_{m\in\mathbb{N}} \left\{ \begin{array}{c} U+ \\ f(U_1,\ldots,U_m) \end{array};\ \begin{array}{l} U \in \mathbf{2}\langle\Delta\rangle, f \in \mathcal{D}_m \text{ and} \\ (\exists j \in \{1,\ldots,m\})(U_j \in \Delta_i^\square \text{ and} \\ U_1,\ldots,U_{j-1},U_{j+1},\ldots,U_m \in \mathbf{2}\langle\Delta\rangle) \end{array} \right\}$.

More generally, we can define the set Θ_i of i-hole generalized contexts: if $i = 0$, then $\Theta_i = \mathbf{2}\langle\Delta\rangle$; if $i = 1$, then Θ_i is already defined. Now, for $i > 1$, if $D[] \in \Theta_i$, then $D[] = C[f(U_1,\ldots U_m)]$ with $C[] \in \Theta_1$ and $U_1 \in \Theta_{i_1},\ldots,U_m \in \Theta_{i_m}$, $i_1 + \ldots + i_m = i$, so the several holes have to be in the same summand.

Definition 5. *Let $C[] \in \Theta_1$. We define the binary relation $\rhd_{C[]}$ on $\mathbf{2}\langle\Delta\rangle$ as follows: for any $U, U' \in \mathbf{2}\langle\Delta\rangle$, we have $U \rhd_{C[]} U'$ if, and only if, there exist $u \in \Delta$ and $V \in \mathbf{2}\langle\Delta\rangle$ such that $U = C[u]$, $U' = C[V]$ and ($u \rhd_\Delta V$ or $V = 0$).*

Then we define the binary relation $\rhd$ on $\mathbf{2}\langle\Delta\rangle$ by writing (as usual) $U \rhd V$ if, and only if, there exists $C[] \in \Theta_1$ such that $U \rhd_{C[]} V$.

For any generalized term $t \in \Delta$, we denote by $\|t\|$ the maximum number of distict summands occurring anywhere in t. In particular, if $t \in \Delta_0 = \mathcal{T}_0(\mathcal{C})$, we have $\|t\| = 1$. We generalize this notation to any element U of $\mathbf{2}\langle\Delta\rangle$ by setting $\|U\| = \begin{cases} 0 & \text{if } U = 0; \\ \max\{\|u\|;\ u \in \mathit{Supp}(U)\} & \text{otherwise.} \end{cases}$

For any $k \in \mathbb{N}$, we denote by $\mathbf{2}\langle\Delta\rangle_k$ the set $\{U \in \mathbf{2}\langle\Delta\rangle;\ \|U\| \leq k\}$ and by $\rhd_k$ the restriction of the binary relation $\rhd$ to $\mathbf{2}\langle\Delta\rangle_k$, i.e. $\mathbf{2}\langle\Delta\rangle_k = \rhd_{|\mathbf{2}\langle\Delta\rangle_k \times \mathbf{2}\langle\Delta\rangle_k}$. For any $k \in \mathbb{N}$, the relation $\rhd_k^*$ enjoys the following properties:

- For any $U, V \in \mathbf{2}\langle\Delta\rangle_k$, we have $U + V \rhd_k^* V$.
- Let $q \in \mathbb{N}$. Let $W_1, \ldots, W_q, V_1, \ldots, V_q \in \mathbf{2}\langle\Delta\rangle_k$ such that $W_1 \rhd_k^* V_1, \ldots, W_q \rhd_k^* V_q$. Then we have $\sum_{j=1}^{q} W_j \rhd_k^* \sum_{j=1}^{q} V_j$. Moreover, for any $C[] \in \Theta_q$ such that $C[W_1, \ldots, W_q] \in \mathbf{2}\langle\Delta\rangle_k$, we have $C[W_1, \ldots, W_q] \rhd_k^* C[V_1, \ldots, V_q]$.

Definition 6. *Let $k \in \mathbb{N}$ and let $U, V \in \mathbf{2}\langle\Delta\rangle$.*

We denote by $\boldsymbol{Seq}_\Delta(U, V)$ (resp. $\boldsymbol{Seq}_{\Delta,k}(U, V)$) the set of finite sequences $(U_1, \ldots, U_n) \in \mathbf{2}\langle\Delta\rangle^{<\infty}$ such that $U = U_1$, $V = U_n$ and, for any $i \in \{1, \ldots, n-1\}$, we have $U_i \rhd U_{i+1}$ (resp. $U_i \rhd_k U_{i+1}$).

For any $(U_1, \ldots, U_n) \in \boldsymbol{Seq}_\Delta(U, V)$, we denote by $\mathit{length}_\Delta((U_1, \ldots, U_n))$ the integer $\mathit{Card}(\left\{ i \in \{1, \ldots, n-1\};\ \begin{array}{l} (\exists C[] \in \Theta_1, u \in \Delta_1 \setminus \Delta_0, V \in \mathbf{2}\langle\Delta\rangle) \\ (u \rhd_\Delta V, U_i = C[u] \text{ and } U_{i+1} = C[V]) \end{array} \right\})$.

Definition 7. *For any $(\rho, v, C[]) \in \boldsymbol{Seq} \times \mathcal{U}_0 \times \Phi_1$ such that $\rho \in \boldsymbol{Seq}(C[v], -)$, we define $\mathcal{R}(\rho, v, C[]) \subseteq \boldsymbol{Seq}(v, -)$ by induction on $\mathit{length}(\rho)$ as follows:*

- *if $\mathit{length}(\rho) = 0$, then $\mathcal{R}(\rho, v, C[]) = \{id_v\}$;*
- *if $\rho = C_0[u] \to_{C_0[],(l,r)} C_0[u'];\ \rho_0$ with $C_0[] = C[C'[]]$, then $\mathcal{R}(\rho, v, C[]) = \{(v \to_{C'[],(l,r)} C'[u'];\ \rho_0');\ \rho_0' \in \mathcal{R}(\rho_0, C'[u'], C[])\}$;*
- *if $\rho = C_0[u] \to_{C_0[],(l,r)} C_0[u'];\ \rho_0$ and there is no $C'[] \in \Phi_1$ such that $C_0[] = C[C'[]]$, then $\mathcal{R}(\rho, v, C[])$ is the set*

$$\{id_v\} \cup \left(\bigcup_{\substack{C''[] \in \Phi_1 \\ \langle v | C''[] \rangle \in \langle v | C[] \rangle \setminus C_0[u] \to_{C_0[],(l,r)} C_0[u']}} \mathcal{R}(\rho_0, v, C''[]) \right)$$

and we set $\mathcal{N}(\rho, v, C[]) = \{c \in \mathcal{T}_0(\mathcal{C});\ \mathcal{R}(\rho, v, C[])) \cap \boldsymbol{Seq}(v, c) \neq \emptyset\}$.

In other words, $\mathcal{N}(\rho, v, C[])$ is the set of constructor terms that are descendants of the occurrence $\langle v | C[] \rangle$ of v in $C[v]$ during the reduction ρ. Notice that in the case ρ is an innermost reduction sequence, the set $\mathcal{R}(\rho, v, C[])$ is a singleton.

Lemma 1. *Let $m \in \mathbb{N}$. Let $E[] \in \Phi_m$. Let $u_1, \ldots, u_m \in \mathcal{U}_0$. Let $c \in \mathcal{T}_0(\mathcal{C})$. Let $\rho \in \boldsymbol{Seq}(E[u_1, \ldots, u_m], c)$. For any $l \in \{1, \ldots, m\}$, let $U_l \in \mathbf{2}\langle\Delta\rangle$ such that $\mathcal{N}(\rho, u_l, E[u_1, \ldots, u_{l-1}, \Box, u_{l+1}, \ldots, u_m]) \subseteq \mathit{Supp}(U_l)$. Then $E[U_1, \ldots, U_m] \rhd^* c$.*

Definition 8. *Let* $t \in \mathcal{U}_0$*. For any* $u \in \Delta$*, we define the relation* $t \downarrow u$ *by induction on* level(u) *as follows:*

- *if* $u \in \Delta_0$*, then* $t \downarrow u$ *if, and only if,* $t \to^* u$*;*
- *if* $u = f(V_1, \ldots, V_q) \in \Delta_{i+1} \setminus \Delta_i$*, then* $t \downarrow u$ *if, and only if, there exist* $v_1, \ldots, v_q \in \mathcal{U}_0$ *such that* $t \to^* f(v_1, \ldots, v_q)$ *and, for any* $j \in \{1, \ldots, q\}$*, for any* $v \in Supp(V_j)$*,* $v_j \downarrow v$*.*

This relation is extended to the relation $\downarrow \subseteq \mathcal{U}_0 \times \mathit{2}\langle\Delta\rangle$ *defined by:* $t \downarrow U$ *if, and only if, for any* $u \in Supp(U)$*,* $t \downarrow u$*.*

Notice that, for any $t, u \in \mathcal{U}_0$, we have $t \downarrow u$ if, and only if, $t \to^* u$.

Lemma 2. *Let* $C[] \in \Theta_1$*. Let* $t \in \mathcal{U}_0$*,* $U, V \in \mathit{2}\langle\Delta\rangle$ *such that* $t \downarrow U$ *and* $U \rhd_{C[]} V$*. Then* $t \downarrow V$*.*

Proposition 1. *Let* $t \in \mathcal{U}_0$*,* $c \in \mathcal{T}_0(\mathcal{C})$*. We have* $t \to^* c$ *if, and only if,* $t \rhd^* c$*.*

Proof: Assume that $t \to^* c$. We have $t \in \Phi_0$ and $\mathbf{Seq}(t, c) \neq \emptyset$. Therefore, by Lemma 1, we have $t \rhd^* c$.

Conversely, we prove, by induction on n and applying Lemma 2, that, for any $n \in \mathbb{N}$, for any $U_0, \ldots, U_n \in \mathbf{2}\langle\Delta\rangle$ such that $t = U_0 \rhd U_1 \ldots U_{n-1} \rhd U_n$, we have $t \downarrow U_n$.

Example 1. Consider the following (constrained) cons-free TRS: the set $\mathcal{D}$ is $\{k/1, h/2, p/1\}$ and the set $\mathcal{C}$ is $\{c/2, n/0, \mathsf{true}/0, \mathsf{false}/0\}$ with the following rewrite rules:

- $p(c(x, c(y, z))) \to x$
- $p(c(x, c(y, z))) \to y$
- $h(x, \mathsf{false}) \to x$
- $k(x) \to h(x, x)$

We have $k(p(c(\mathsf{true}, c(\mathsf{false}, n)))) \to h(p(c(\mathsf{true}, c(\mathsf{false}, n))), p(c(\mathsf{true}, c(\mathsf{false}, n))))$ $\to^* h(\mathsf{true}, \mathsf{false}) \to \mathsf{true}$ (notice that there is no innermost reduction sequence from $k(p(c(\mathsf{true}, c(\mathsf{false}, n))))$ to true, so in particular the algorithm considered in [16] applied to the evaluation of $k(p(c(\mathsf{true}, c(\mathsf{false}, n))))$ is not able to find this normal form). Now, we have

$$\begin{aligned} k(p(c(\mathsf{true}, c(\mathsf{false}, n)))) &\rhd_2 k(p(c(\mathsf{true}, c(\mathsf{false}, n))) + \mathsf{false}) \\ &\rhd_2 k(\mathsf{true} + \mathsf{false}) \\ &\rhd_2 h(\mathsf{true} + \mathsf{false}, \mathsf{true} + \mathsf{false}) \\ &\rhd_2^* h(\mathsf{true}, \mathsf{false}) \\ &\rhd_2 \mathsf{true} \end{aligned}$$

3 Computation in constrained cons-free TRS

In this section, we show that, for any constrained cons-free TRS, it is enough to consider $\rhd$-reduction sequences $(U_i)_{i\in\mathbb{N}}$ of elements of $\mathbf{2}\langle\Delta\rangle_K$ (i.e. $\rhd_K$-reduction sequences) for some integer K depending only on the TRS.

We are given a constrained cons-free TRS and we set $\mathcal{B} = \mathcal{D} \setminus \mathcal{A}$ and as the TRS is finite, we let $K \geq 1$ be an integer such that, for any $(f(c_1, \ldots, c_q), r) \in \mathcal{R}$ for any $x \in \mathcal{X} \cap \{c_1, \ldots, c_q\}$, $Oc(x, r) \leq K$.

Definition 9. *Let $i \in \mathbb{N}$. For any $U \in \mathit{2}\langle\Delta_i\rangle$, we define $U^* \in \mathit{2}\langle\Delta_i\rangle \cap \mathit{2}\langle\Delta\rangle_K$ by induction on i:*

- *$i = 0$: we set $U^* = U$;*
- *$i > 0$, $U = f(U_1, \ldots, U_q) \in \Delta_i \setminus \Delta_{i-1}$: we set*

$$U^* = \sum_{\substack{W_1, \ldots, W_q \in \mathit{2}\langle\Delta_{i-1}\rangle \\ \mathit{Supp}(W_j) \subseteq \mathit{Supp}({U_j}^*) \\ \mathit{Card}(\mathit{Supp}(W_j)) = \min\{Z, \mathit{Card}(\mathit{Supp}({U_j}^*))\} \\ \text{for } j \in \{1, \ldots, q\}}} f(W_1, \ldots, W_q),$$

$$\text{where } Z = \begin{cases} 1 & \text{if } f \in \mathcal{A}; \\ K & \text{if } f \in \mathcal{B}. \end{cases}$$

- *$i > 0$, $U \in \mathit{2}\langle\Delta_i\rangle \setminus (\Delta_i \cup \mathit{2}\langle\Delta_{i-1}\rangle)$: we set $U^* = \sum_{u \in \mathit{Supp}(U)} u^*$.*

Definition 10. *Let $q \in \mathbb{N}$. Let $c_1, \ldots, c_q \in \mathcal{T}(\mathcal{C})$. Let $\varphi : \mathcal{X} \rightharpoonup \mathit{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$. Assume that, for any $l \in \{1, \ldots, q\}$, we have $Oc(c_l) \subseteq dom(\varphi)$ and $(c_l \notin \mathcal{X} \Rightarrow {c_l}^\varphi \in \mathcal{T}_0(\mathcal{C}))$. Let $W_1, \ldots, W_q \in \mathit{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$ such that, for any $l \in \{1, \ldots, q\}$, $(c_l \notin \mathcal{X} \Rightarrow W_l = {c_l}^\varphi)$ and $\mathit{Supp}(W_l) \subseteq \mathit{Supp}({c_l}^\varphi)$. Then we denote by $\varphi_{(W_1, \ldots, W_q)}$ the partial function $\mathcal{X} \rightharpoonup \mathit{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$ such that, for any $x \in \mathcal{X}$, we have $\varphi_{(W_1, \ldots, W_q)}(x) = \begin{cases} W_l & \text{if } x = c_l; \\ \varphi(x) & \text{otherwise.} \end{cases}$*

Lemma 3. *Let $(f(c_1, \ldots, c_q), r) \in \mathcal{R}$. Let $\varphi : \mathcal{X} \rightharpoonup \mathit{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$. Assume that, for any $j \in \{1, \ldots, q\}$, we have $Oc(c_j) \subseteq dom(\varphi)$ and $(c_j \notin \mathcal{X} \Rightarrow {c_j}^\varphi \in \mathcal{T}_0(\mathcal{C}))$. Then $\sum_{W_1 \in \mathcal{W}_1, \ldots, W_q \in \mathcal{W}_q} r^{\varphi(W_1, \ldots, W_q)} \rhd_K^* (r^\varphi)^*$, where, for any $j \in \{1, \ldots, q\}$, $\mathcal{W}_j$ is the following subset of $\mathit{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$:*

- *$\{{c_j}^\varphi\}$ in the case $c_j \notin \mathcal{X}$;*
- *$\left\{ W \in \mathit{2}\langle\mathcal{T}_0(\mathcal{C})\rangle; \begin{array}{l} \mathit{Supp}(W) \subseteq \mathit{Supp}({c_j}^\varphi) \text{ and} \\ \mathit{Card}(\mathit{Supp}(W)) = \min\{Oc(c_j, r), \mathit{Card}(\mathit{Supp}({c_j}^\varphi))\} \end{array} \right\}$ in the case $c_j \in \mathcal{X}$.*

Proposition 2. *Let $i \in \mathbb{N}$. Let $C[] \in \Delta_i^\square$. For any $U, V \in \mathit{2}\langle\Delta\rangle$ such that $U \rhd_{C[]} V$, we have $U^* \rhd_K^* V^*$.*

Proof: The proof is by induction on i.

– If $i = 0$, then $C[] = U_0 + \Box$ for some $U_0 \in \mathbf{2}\langle \Delta \rangle$; we distinguish between two cases:
  - $V = U_0$ and there exists $u \in \Delta$ such that $U = U_0 + u$: in this case, we have $\|U^*\|, \|{U_0}^*\| \leq K$. Thus we have $U^* = {U_0}^* + u^* \rhd_K^* {U_0}^* = V^*$.
  - $U = U_0 + f(c_1, \ldots, c_q)^\varphi$, $V = U_0 + r^\varphi$ with $f \in \mathcal{D}_q$, $c_1, \ldots, c_q \in \mathcal{T}(\mathcal{C})$, $\varphi : \mathcal{X} \rightharpoonup \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$ such that, for any $j \in \{1, \ldots, q\}$, $c_j \notin \mathcal{X} \Rightarrow {c_j}^\varphi \in \mathcal{T}_0(\mathcal{C})$: For any $j \in \{1, \ldots, q\}$, we define $\mathcal{W}_j$ as in Lemma 3 and we define $\mathcal{W'}_j$ as follows:
    - if $c_j \notin \mathcal{X}$, then $\mathcal{W'}_j = \{{c_j}^\varphi\}$;
    - if $c_j \in \mathcal{X}$, then $\mathcal{W'}_j$ is the set

$$\left\{ W \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle;\ \begin{array}{l} \mathit{Supp}(W) \subseteq \mathit{Supp}({c_j}^\varphi) \text{ and} \\ \mathit{Card}(\mathit{Supp}(W)) = \min\{Z, \mathit{Card}(\mathit{Supp}({c_j}^\varphi))\} \end{array} \right\}$$

    with $Z = \begin{cases} 1 & \text{if } f \in \mathcal{A}; \\ K & \text{if } f \in \mathcal{B}. \end{cases}$

    We have

$$\begin{aligned}
(f(c_1, \ldots, c_q)^\varphi)^* &= \sum_{W_1 \in \mathcal{W'}_1, \ldots, W_q \in \mathcal{W'}_q} f({c_1}^{\varphi(W_1, \ldots, W_q)}, \ldots, {c_q}^{\varphi(W_1, \ldots, W_q)}) \\
&\rhd_K^* \sum_{W_1 \in \mathcal{W}_1, \ldots, W_q \in \mathcal{W}_q} f(c_1, \ldots, c_q)^{\varphi(W_1, \ldots, W_q)} \\
&\rhd_K^* \sum_{W_1 \in \mathcal{W}_1, \ldots, W_q \in \mathcal{W}_q} r^{\varphi(W_1, \ldots, W_q)} \\
&\rhd_K^* (r^\varphi)^* \text{ (by Lemma 3).}
\end{aligned}$$

  We have $U_0 \in \mathbf{2}\langle \Delta \rangle_K$, so $U^* = {U_0}^* + (f(c_1, \ldots, c_q)^\varphi)^* \rhd_K^* {U_0}^* + (r^\varphi)^* = V^*$.

– If $i > 0$, $U = C[U_0]$, $V = C[V_0]$, $C[] = U' + f(U_1, \ldots, U_m)$, $k \in \{1, \ldots, m\}$, $U_k \in \Delta_{i-1}^\Box$ and $U_0 \rhd_\Box V_0$, then, by the induction hypothesis, $U_k[{U_0}^*] \rhd_K^* U_k[{V_0}^*]$. Let $V_1, \ldots, V_m \in \mathbf{2}\langle \Delta \rangle$ such that
  - for any $j \in \{1, \ldots, m\} \setminus \{k\}$, we have

$$\mathit{Supp}(V_j) \subseteq \mathit{Supp}({U_j}^*) \text{ and } \mathit{Card}(\mathit{Supp}(V_j)) = \min\{1, \mathit{Card}(\mathit{Supp}({U_j}^*))\}$$

  - and $\mathit{Supp}(V_k) \subseteq \mathit{Supp}(U_k[V_0]^*)$ and

$$\mathit{Card}(\mathit{Supp}(V_k)) = \begin{cases} \min\{1, \mathit{Card}(\mathit{Supp}(U_k[V_0]^*))\} & \text{if } f \in \mathcal{A}; \\ \min\{K, \mathit{Card}(\mathit{Supp}(U_k[V_0]^*))\} & \text{if } f \in \mathcal{B}. \end{cases}$$

  There exists $V \in \mathbf{2}\langle \Delta \rangle$ such that $\mathit{Supp}(V) \subseteq \mathit{Supp}(U_k[U_0])$, $\mathit{Card}(\mathit{Supp}(V)) \leq \mathit{Card}(\mathit{Supp}(V_k))$ and $V \rhd_K^* V_k$, and hence $f(V_1, \ldots, V_{k-1}, V, V_{k+1}, \ldots, V_m) \rhd_K^* f(V_1, \ldots, V_k)$. We obtain

$$\begin{aligned}
& f(U_1, \ldots, U_{k-1}, U_k[U_0], U_{k+1}, \ldots, U_m)^* \\
\rhd_K^* \ & f(U_1, \ldots, U_{k-1}, U_k[V_0], U_{k+1}, \ldots, U_m)^*;
\end{aligned}$$

moreover we have $\|U'^*\| \leq K$, hence

$$\begin{aligned} U^* &= U'^* + f(U_1, \ldots, U_{k-1}, U_k[U_0], U_{k+1}, \ldots, U_m)^* \\ &\rhd_K^* U'^* + f(U_1, \ldots, U_{k-1}, U_k[V_0], U_{k+1}, \ldots, U_m)^* \\ &= V^* \end{aligned}$$

Corollary 1. *Let $q \in \mathbb{N}$. Let $C_1, \ldots, C_q \in \mathit{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$ such that $Card(Supp(C_1))$, $\ldots$, $Card(Supp(C_q)) \leq K$, let $c \in \mathcal{T}_0(\mathcal{C})$ and $f \in \mathcal{D}_q$. We have $f(C_1, \ldots, C_q) \rhd^* c$ if, and only if, $f(C_1, \ldots, C_q) \rhd_K^* c$.*

Proof: Apply Proposition 2, noticing that $f(C_1, \ldots, C_q)^* = f(C_1, \ldots, C_q)$ and $c^* = c$.

4 A polynomial time algorithm

In this section we describe a polynomial time algorithm that computes the constructor terms obtained by $\rhd_K$-reduction sequences in a constrained cons-free TRS. Assume that we are given a constrained cons-free TRS. We assume that $\mathcal{R} = \{(l_1, r_1), \ldots, (l_R, r_R)\}$ and $\bigcup_{j=1}^{R} Oc(l_j) = \{x_1, \ldots, x_V\}$. We set $\mathcal{T}_\mathcal{R} = \bigcup \{t \in \mathcal{T}(\mathcal{D} \cup \mathcal{C});\ (\exists j \in \{1, \ldots, R\})(t \trianglelefteq l_j \text{ or } t \trianglelefteq r_j)\}$. We set $A = \max\{ar(f);\ f \in \mathcal{D} \cup \mathcal{C}\}$, $O = \max\{1, \max\{Oc(f, r_j);\ f \in \mathcal{D} \text{ and } j \in \{1, \ldots, R\}\}\}$, $Q = Card(\mathcal{D})$ and $S = Card(\{c \in \mathcal{T}_0(\mathcal{C});\ (\exists (l, r) \in \mathcal{R}) c \trianglelefteq r\})$. For any $c_0 \in \mathcal{T}_0(\mathcal{C})$, we set $\mathcal{I}(c_0) = \{c \in \mathcal{T}_0(\mathcal{C});\ c \trianglelefteq c_0 \textit{ or } (\exists (l, r) \in \mathcal{R}) c \trianglelefteq r\}$.

Remark 1. For any $c_0 \in \mathcal{T}_0(\mathcal{C})$, we have $Card(\mathcal{I}(c_0)) = S + |c_0|$.

Definition 11. *For any $c_0 \in \mathcal{T}_0(\mathcal{C})$, we set $\mathcal{V}(c_0) = \bigcup_{i \in \mathbb{N}} \mathcal{V}_i(c_0)$, where $\mathcal{V}_i(c_0)$ is a subset of $\{U \in \mathit{2}\langle \Delta_i \rangle;\ \|U\| \leq K\}$ defined by induction on i:*

- $\mathcal{V}_0(c_0) = \{U \in \mathit{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle;\ Card(Supp(U)) \leq K \textit{ and } Supp(U) \subseteq \mathcal{I}(c_0)\}$
- $\mathcal{V}_{i+1}(c_0) = \mathcal{V}_i(c_0) \cup \bigcup_{m=1}^{A} \{f(V_1, \ldots, V_m);\ f \in \mathcal{D}_m \textit{ and } V_1, \ldots, V_m \in \mathcal{V}_i(c_0)\}$

Given $c_0 \in \mathcal{T}_0(\mathcal{C})$, the algorithm will compute, for every element u of $\mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0)$, the set of constructor terms c such that $u \rhd_K^* c$. In particular, by Proposition 1 and Corollary 1, if K is large enough, then, for every $f \in \mathcal{D}_m$ and every $c_1, \ldots, c_m \in \mathcal{I}(c_0)$, it will return exactly all the constructor terms c such that $f(c_1, \ldots, c_m) \to^* c$.

Remark 2. For any $c_0 \in \mathcal{T}_0(\mathcal{C})$, we have $Card(\mathcal{V}_0(c_0)) \leq Card(\mathcal{I}(c_0))^{K+1}$, hence $Card(\mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0)) \leq Q \cdot Card(\mathcal{I}(c_0))^{A \cdot (K+1)}$.

Definition 12. *For any $c_0 \in \mathcal{T}_0(\mathcal{C})$, for any $i \in \mathbb{N} \setminus \{0\}$, for any $V \in \mathcal{V}_i(c_0) \setminus \mathcal{V}_{i-1}(c_0)$, we define, by induction on i,* the leftmost-innermost redex *$\langle U | E[] \rangle$ of V with $U \in \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0)$ and $E[] \in \Theta_1$:*

- *if $i = 1$, then the leftmost-innermost redex of V is $\langle V | \Box \rangle$;*

– *if* $i > 1$ *and* $V = f(V_1, \ldots, V_m)$*, then the leftmost-innermost redex of* V *is* $\langle W | f(V_1, \ldots, V_{j-1}, C[], V_{j+1}, \ldots, V_m)\rangle$*, where* $j = \min\{k \in \{1, \ldots, m\};\ V_k \notin \mathcal{V}_0\}$ *and* $\langle W | C[]\rangle$ *is the leftmost-innermost redex of* V_j*.*

From now M will be an integer and E the subset $\{1, \ldots, M\}$ of $\mathbb{N}$; L will be a function $E \to \mathcal{D} \cup \mathcal{C} \cup \{x_1, \ldots, x_V, \bot\}$, $Succ$ will be a partial function $E \rightharpoonup (E \cup \{\bot\})^{\{1,\ldots,A\}}$ and $Comp$ will be a partial function $E \rightharpoonup (E \cup \{\bot\})^{\{1,\ldots,K\}}$.

Definition 13. *For any* $t \in \mathcal{T}(\mathcal{D} \cup \mathcal{C})$*, for any* $n \in E$*, we define, by induction on* t*,* $F_{\mathcal{T}}(t, n) \in \{0, 1\}$ *as follows:* $F_{\mathcal{T}}(f(t_1, \ldots, t_m), n) = 1$ *if, and only if,* $n \in dom(Succ)$*,* $L(n) = f$ *and* $F_{\mathcal{T}}(t_1, Succ(n)(1)) = \ldots = F_{\mathcal{T}}(t_m, Succ(n)(m)) = 1$*.*

Notice that $F_{\mathcal{T}}(t, n) = F_{\mathcal{T}}(t', n) = 1 \Rightarrow t = t'$, hence we can define a partial function $[\![\cdot]\!]_{\mathcal{T}} : E \rightharpoonup \mathcal{T}(\mathcal{D} \cup \mathcal{C})$ by setting $[\![n]\!]_{\mathcal{T}} = t$ if, and only if, $F_{\mathcal{T}}(t, n) = 1$. In the same way, we define a partial function $[\![\cdot]\!]_{\mathcal{V},c_0} : E \rightharpoonup \mathcal{V}(c_0)$ for any $c_0 \in \mathcal{T}_0(\mathcal{C})$:

Definition 14. *Let* $c_0 \in \mathcal{T}_0(\mathcal{C})$*. For any* $i \in \mathbb{N}$*, for any* $V \in \mathcal{V}_i(c_0)$*, for any* $n \in E$*, we define, by induction on* i*,* $F_{\mathcal{V},c_0}(V, n) \in \{0, 1\}$*:*

– *if* $i = 0$*, then* $F_{\mathcal{V},c_0}(V, n) = 1$ *if, and only if,* $[\![n]\!]_{\mathcal{T}} = V$ *or the following holds:* $L(n) = \bot$*,* $n \in dom(Comp)$ *and* $\sum_{\substack{k \in \{1,\ldots,K\} \\ Comp(n)(k) \neq \bot}} [\![Comp(n)(k)]\!]_{\mathcal{T}} = V$*;*
– *if* $i > 0$ *and* $V = f(V_1, \ldots, V_m) \notin \mathcal{V}_{i-1}$*, then* $F_{\mathcal{V},c_0}(V, [\![n]\!]_{\mathcal{V},c_0}) = 1$ *if, and only if,* $n \in dom(Succ)$*,* $L(n) = f$ *and* $[\![Succ(n)(1)]\!]_{\mathcal{V},c_0} = V_1$*, …,* $[\![Succ(n)(ar(f))]\!]_{\mathcal{V},c_0} = V_{ar(f)}$*.*

Since, for any $c_0 \in \mathcal{T}_0(\mathcal{C})$, we have $F_{\mathcal{V},c_0}(V, v) = F_{\mathcal{V},c_0}(V', n) = 1 \Rightarrow V = V$, we can define a partial function $[\![\cdot]\!]_{\mathcal{V},c_0} : E \rightharpoonup \mathcal{V}(c_0)$ by setting $[\![n]\!]_{\mathcal{V},c_0} = V$ if, and only if, $F_{\mathcal{V},c_0}(V, n) = 1$.

In the two following definitions, we restrict the partial functions $[\![\cdot]\!]_{\mathcal{T}}$ and $[\![\cdot]\!]_{\mathcal{V},c_0}$ to elements of E that *unshare* (hence the symbol U) defined symbol functions.

Definition 15. *For any* $n \in dom([\![\cdot]\!]_{\mathcal{T}})$*, we define* $Reach_{\mathcal{T}}(n) \subseteq E$ *by induction on* $[\![n]\!]_{\mathcal{T}}$*: if* $L(n) \notin \mathcal{D}$*, then* $Reach_{\mathcal{T}}(n) = \emptyset$*; if* $L(n) \in \mathcal{D}$*, then* $Reach_{\mathcal{T}}(n) = \{n\} \cup \bigcup_{j=1}^{ar(L(n))} Reach_{\mathcal{T}}(Succ(n)(j))$*.*

We set $U_{\mathcal{T}} = \{n \in dom([\![\cdot]\!]_{\mathcal{T}});\ (\forall m, m' \in Reach_{\mathcal{T}}(n))(L(m) = L(m') \Rightarrow (m \neq m' \text{ or } L(m) \notin \mathcal{D}))\}$ *and* $[\![\cdot]\!]_{\mathcal{T},U} = [\![\cdot]\!]_{\mathcal{T}}|_{U_{\mathcal{T}}}$*.*

Definition 16. *For any* $c_0 \in \mathcal{T}_0(\mathcal{C})$*, for any* $i \in \mathbb{N}$*, for any* $V \in \mathcal{V}_i(c_0)$*, for any* $n \in E$ *such that* $[\![n]\!]_{\mathcal{V},c_0} = V$*, we define* $Reach_{\mathcal{V},c_0}(n)$ *by induction on* i*:*

– *if* $i = 0$ *and* $L(n) = \bot$*, then*

$$Reach_{\mathcal{V},c_0}(n) = \bigcup_{\substack{k \in \{1,\ldots,K\} \\ Comp(n)(k) \neq \bot}} Reach_{\mathcal{T}}(Comp(n)(k))$$

- *if* $i = 0$ *and* $L(n) \neq \bot$*, then* $Reach_{\mathcal{V},c_0}(n) = Reach_{\mathcal{T}}(n)$
- *if* $i > 0$*, then* $Reach_{\mathcal{V},c_0}(n) = \{n\} \cup \bigcup_{j=1}^{ar(L(n))} Reach_{\mathcal{V},c_0}(Succ(n)(j))$

For any $c_0 \in \mathcal{T}_0(\mathcal{C})$*, we denote by* $U_{\mathcal{V}}(c_0)$ *the set of* $n \in dom([\![\cdot]\!]_{\mathcal{V},c_0})$ *such that* $(\forall m, m' \in Reach_{\mathcal{V},c_0}(n))(L(m) = L(m') \Rightarrow (m \neq m' \text{ or } L(m) \notin \mathcal{D}))$ *and we set* $[\![\cdot]\!]_{\mathcal{V},c_0,U} = [\![\cdot]\!]_{\mathcal{V},c_0}|_{U_{\mathcal{V}}(c_0)}$.

From now, we assume that, for any $c_0 \in \mathcal{T}_0(\mathcal{C})$, we are given a bijection $inp(c_0) : \begin{array}{ccc} \{1, \ldots, Inp\text{-}Max\} & \to & \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0) \\ i & \mapsto & [\![i]\!]_{\mathcal{V},c_0} \end{array}$. By Remark 2, we can assume that $Inp\text{-}Max \leq Q \cdot Card(\mathcal{I}(c_0))^{A \cdot (K+1)}$.

The algorithm begins with a procedure $\mathsf{Inst\text{-}Init}()$ which performs the following one: for any $c_0 \in \mathcal{T}_0(\mathcal{C})$, after the execution of the procedure, we have

- $\{[\![Inp(i)]\!]_{\mathcal{V},c_0};\ 1 \leq i \leq Inp\text{-}Max\} = \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0)$
- and, for any $i \in \{1, \ldots, Inp\text{-}Max\}$, we have $\{[\![Inst(i)(s)]\!]_{\mathcal{V},c_0,U};\ 1 \leq s \leq R\} \setminus \{\bot\} = \{V \in \mathbf{2}\langle\Delta\rangle;\ [\![Inp(i)]\!]_{\mathcal{V},c_0} \rhd_{\Delta} V\}$.

Here we used the following crucial property of cons-free term rewriting systems: whenever we perform a reduction step $u \rhd_{\Delta} V$ with $u \in \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0)$, we have $V \in \mathcal{V}(c_0)$ (and not only in $\mathbf{2}\langle\Delta\rangle$).

The algorithm calls the procedure $\mathsf{Inf\text{-}Inp}(i, j)$ with $i, j \in \{1, \ldots, Inp\text{-}Max\}$. This procedure performs the following one: for any $c_0 \in \mathcal{T}_0(\mathcal{C})$, if there exist $q \in \mathbb{N}$, $f \in \mathcal{D}_q$ and $C_1, \ldots, C_q, C'_1, \ldots, C'_q \in \mathbf{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$ such that

- $[\![Inp(i)]\!]_{\mathcal{V},c_0} = f(C'_1, \ldots, C'_q)$,
- $[\![Inp(j)]\!]_{\mathcal{V},c_0} = f(C_1, \ldots, C_q)$
- and $Supp(C'_1) \subseteq Supp(C_1)$, $\ldots$, $Supp(C'_q) \subseteq Supp(C_q)$,

then the procedure $\mathsf{Inf\text{-}Inp}(i, j)$ returns true; otherwise it returns false.

Definition 17. *Let* $c_0 \in \mathcal{T}_0(\mathcal{C})$*. For any* $i \in \mathbb{N}$*, we define some subset* $\Psi_i(c_0)$ *of* $\Delta_i^{\Box}$ *by induction on* i *as follows:* $\Psi_0(c_0) = \{\Box\}$ *and*

$$\Psi_{i+1}(c_0) = \bigcup_{m \in \mathbb{N}} \left\{ f(V_1, \ldots, V_m);\ \begin{array}{c} f \in \mathcal{D}_m \text{ and } (\exists j \in \{1, \ldots, m\}) \\ (V_j \in \Psi_i(c_0) \text{ and} \\ V_1, \ldots, V_{j-1}, V_{j+1}, \ldots, V_m \in \mathcal{V}(c_0)) \end{array} \right\}.$$

We set $\Psi(c_0) = \bigcup_{i \in \mathbb{N}} \Psi_i(c_0)$.

Definition 18. *Let* $c_0 \in \mathcal{T}_0(\mathcal{C})$*. Let* $Y : \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0) \to \{\mathsf{true}, \mathsf{false}\}^{\mathcal{I}}$*. For any* $C[] \in \Psi(c_0)$*, we define the binary relation on* $\mathcal{V}(c_0)$ *as follows:* $V \rhd_{Y,C[]} V'$ *if, and only if, there exist* $u \in \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0)$ *and* $V_0 \in \mathcal{V}_0(c_0)$ *such that* $Supp(V_0) \subseteq \{c \in \mathcal{I}(c_0);\ Y(u)(c) = \mathsf{true}\}$*,* $V = C[u]$ *and* $V' = C[V_0]$.

We define the binary relation $\rhd_Y$ *on* $\mathcal{V}(c_0)$ *as follows:* $V \rhd_Y V'$ *if, and only if, there exists* $W \in \mathbf{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$ *such that*

- $Card(Supp(W)) \leq K$*,*
- *for any* $w \in Supp(W)$*,* $Y(U)(w) = \mathsf{true}$
- *and* $V' = E[W]$*,*

where $\langle U|E[]\rangle$ is the leftmost-innermost redex of V.

The algorithm uses a procedure Computation, which has the following properties: Let $c_0 \in \mathcal{T}_0(\mathcal{C})$. Let $Q \leq O$. Let $V \in \mathcal{V}_Q(c_0)$. Let $n \in E$ such that $[\![n]\!]_{\mathcal{V},c_0,U} = V$. Let $Y : \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0) \to \{\mathsf{true}, \mathsf{false}\}^{\mathcal{I}(c_0)}$ such that, for any $V' \in \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0)$, for any $c \in \mathcal{I}(c_0)$, $Y(V')(c) = \mathsf{true}$ if, and only if, $\mathit{Val}(\mathit{inp}(c_0)^{-1}(V'))(c) = \mathsf{true}$. After the execution of the procedure $\mathsf{Computation}(n)$, we have:

- apart from M and D, which increased, and apart from *Result*, no value of any global variable changed;
- the increasing of D is bound by $(\mathit{Card}(\mathcal{I}(c_0)) + 1)^{Q \cdot K} \cdot K$;
- for any $c \in \mathcal{I}(c_0)$, there exists $j \in \{1, \ldots, D\}$ such that $\mathit{Result}(j) = c$ if, and only if, there exists $V' \in \mathcal{V}_0(c_0)$ such that $c \in \mathit{Supp}(V')$ and $V \rhd^*_Y V'$.

The execution time of the procedure Computation is polynomial in the size of c_0.

```
Inst-Init(); change := true;
while change do
    change := false;
    for i := 1 to Inp-Max do
        D := 0; for s := 1 to R do Computation(Inst(i)(s)); od;
        for o := 1 to Inp-Max do
            if Inf-Inp(i, o) then
                                  for j := 1 to D do
                                      if Val(o)(Result(j)) ≠ false
                                        then change := true;
                                             Val(o)(Result(j)) := true;
                                      fi;
                                  od;
            fi;
        od; od; od;
```

Fig. 1. The algorithm

The key-point to notice is that the execution time of the algorithm is in $\mathcal{O}(|c_0|^H)$ for some constant H is that the size of the table *Val* is $\mathit{Card}(\mathit{Inp\text{-}Max}) \times \mathit{Card}(\mathcal{I}(c_0)) \leq (Q \cdot \mathit{Card}(\mathcal{I}(c_0))^{A \cdot (K+1)}) \times (S + |c_0|)$. Hence, for any $c_0 \in \mathcal{T}_0(\mathcal{C})$, for any $m \in \mathbb{N}$, for any $f \in \mathcal{D}_m$, for any $C_1, \ldots, C_m \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C})\rangle$ such that $\mathit{Card}(\mathit{Supp}(C_1)), \ldots, \mathit{Card}(\mathit{Supp}(C_m)) \leq K$, for any $c \in \{c \in \mathcal{T}_0(\mathcal{C});\ c \trianglelefteq c_0 \ \mathit{or}\ (\exists (l,r) \in \mathcal{R}) c \trianglelefteq r\}$, the problem of deciding whether $f(C_1, \ldots, C_m) \rhd^*_K c$ holds is solvable in time polynomial in the size of c_0.

5 Characterizing *P*

Let Γ be the signature $\{\mathrm{one}/1, \mathrm{zero}/1, \mathrm{nil}/0\}$. For each $t = f_1(f_2(\cdots f_n(\mathrm{nil}))) \in \mathcal{T}_0(\Gamma)$, we define the string $\langle t\rangle$ to be $\langle f_1\rangle\langle f_2\rangle \cdots \langle f_n\rangle$ where $\langle \mathrm{one}\rangle$ ='1' and

$\langle \text{zero} \rangle =$'0'. Clearly, $\mathcal{T}_0(\Gamma)$ is in bijective correspondence with $\{0,1\}^{<\infty}$ under $\langle\cdot\rangle$.

Jones [3] considers (deterministic) *cons-free functional programs*. Now, the following lemma holds:

Lemma 4. *Any (deterministic) cons-free functional program taking only zeroth-order data and involving only terminating functions can be simulated by an orthogonal cons-free TRS.*

Proof: Given a (deterministic) cons-free functional program p taking only zeroth-order data, we consider the following cons-free TRS: for any declaration of the form $f\,x_1\,\ldots\,x_n = e^f$ in p, we have the rewrite rule $f(x_1,\ldots,x_n) \to (e^f)^*$, where $(e^f)^*$ is defined by induction on e^f: for instance, if $e^f = \mathsf{if}\,e_1\,e_2\,e_3$, then $(e^f)^* = \mathsf{if}(e_1{}^*, e_2{}^*, e_3{}^*)$; moreover we have the rewrite rules $\mathsf{if}(\mathsf{true}, x, y) \to x$ and $\mathsf{if}(\mathsf{false}, x, y) \to y$.

As the language of [3] involves only a single function declaration per function name, and all left-hand sides of such declaration have the form $f(x_1,\ldots,x_n)$ (for distinct $x_1,\ldots,x_n$), it is straightforward that the cons-free TRS we obtained is orthogonal. The operational semantics in [3] is essentially call-by-value and can be straightforwardly simulated by innermost reduction steps (the exceptions are whenever we have expressions of the form $\mathsf{if}\,e_1\,e_2\,e_3$: following the evaluation of e_1, either e_2 or e_3 will not be evaluated). Hence, if $f\,c_1\ldots c_n$ evaluates to some normal form in the functional program, then t reduces to the same normal form in the corresponding TRS. Conversely, as orthogonal TRSs are confluent (hence each term has at most one normal form), and the functions are terminating, if $f(c_1,\ldots,c_n)$ reduces to some normal form c in the TRS, then $f\,c_1\ldots c_n$ evaluates to the value c in the functional program.

Theorem 1. *Let $L \subseteq \{0,1\}^{<\infty}$. Then, $L \in P$ if, and only if, there exists a constrained cons-free TRS over some signature $\mathcal{F} = \mathcal{D} \cup \mathcal{C}$ such that (i) $\Gamma \subseteq \mathcal{C}$, and there is $f \in \mathcal{D}$ and $\mathsf{true} \in \mathcal{C}_0$ such that, for any $t \in \mathcal{T}_0(\Gamma)$, we have $f(t) \to^* \mathsf{true}$ if, and only if, $\langle t \rangle \in L$.*

Proof: Corollary 24.2.4 of [18] (or Theorem 6.12 of [3] in the case $k = 0$) shows that we can simulate any polynomial-time Turing Machine by a (deterministic) cons-free (called *read-only* in [18]) functional program taking only zeroth-order data (Note that cons-free in the above setting is slightly stronger than our notion: No constructors are allowed in the right-hand side of function declarations). This simulation involves only terminating functions, hence, by Lemma 4, any polynomial-time Turing Machine can be simulated by an orthogonal cons-free TRS.

The cons-free term rewriting system R obtained from a functional program is not necessarily constrained. To obtain a constrained system, we do the following for each function declaration def $f(x_1,\ldots,x_n) = e^f$ (where the function body e^f is an expression in the functional language): Let $f(x_1,\ldots,x_n) \to r$ be the corresponding cons-free rule. For every such rule, let $\{x_1,\ldots,x_n\}$ be the set of variables that occur immediately beneath the defined symbol at the root of the left-hand side. Choose a set $\{y_1, \ldots, y_n\}$ of distinct variables, and let M be the set of all n-tuples $w = (s_1, \ldots, s_n)$ where s_i (for $1 \leq i \leq n$) is either zero(y_i), one(y_i), or nil. Then replace the rule $f(x_1, \ldots, x_n) \to r$ by the $|M|$ rules on the form $f(s_1, \ldots, s_n) \to r[s_1/x_1, \ldots, s_n/x_n]$, where $(s_1, \ldots, s_n) \in M$ and $r[s_1/x_1, \ldots, s_n/x_n]$ denotes the obvious substitution. Observe that (i) each of the new rules is left-linear if the original rule was, and (ii) that the only overlaps between these rules occur when the left-hand sides are equal. Thus, as R was orthogonal, so is R', and it is clearly constrained as no variable in a left-hand side occurs immediately below the defined symbol at the root.

It is obvious that, for any terms t and t' such that $t \to t'$ in R', we have $t \to t'$ in R: indeed if $t \to_{(l,r)} t'$ and (l, r) is not a rule of R, then there exists a unique rule (l_0, r_0) of R such that (l, r) is obtained from (l_0, r_0); we have $t \to_{(l_0,r_0)} t'$. Reciprocally, if t and t' are two terms such that $t = C[f(t_1, \ldots, t_m)^\sigma] \to_{(f(t_1,\ldots,t_m),r)} C[r^\sigma] = t'$ is a innermost reduction step in R, then ${t_1}^\sigma$, ..., ${t_m}^\sigma$ are constructor terms, hence there exists a rule (l, r) in R' such that $t \to_{(l,r)} t'$. Now, since R is confluent and the functions are terminating, for any term t and any constructor term c, we have $t \to^* c$ in R if, and only if, t reduces to c in R by some innermost strategy.

To see that every constrained, cons-free TRS can be suitably simulated by a polynomial-time Turing machine, let $K \geq 1$ be an integer such that, for any rule $(f(c_1, \ldots, c_q), r)$, for any $x \in \mathcal{X} \cap \{c_1, \ldots, c_q\}$, $Oc(x, r) \leq K$. By Proposition 1 and Corollary 1, we have $f(t) \to^* \mathsf{true}$ if, and only if, $f(t) \rhd_K^* \mathsf{true}$. And the previous section showed that the problem of deciding whether $f(t) \rhd_K^* \mathsf{true}$ holds is solvable in time polynomial in the size of t.

References

1. Jones, N.D.: Logspace and ptime characterized by programming languages. Theor. Comput. Sci. **228**(1-2) (1999) 151–174
2. Bonfante, G.: Some programming languages for logspace and ptime. In Johnson, M., Vene, V., eds.: AMAST. Volume 4019 of Lecture Notes in Computer Science., Springer (2006) 66–80
3. Jones, N.D.: The expressive power of higher-order types or, life without cons. J. Funct. Program. **11**(1) (2001) 5–94
4. Avanzini, M., Moser, G.: Closing the gap between runtime complexity and polytime computability. In Lynch, C., ed.: RTA. Volume 6 of LIPIcs., Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2010) 33–48
5. Hofmann, M.: Type systems for polynomial-time computation (1999) Habilitationsschrift.
6. Bellantoni, S., Cook, S.A.: A new recursion-theoretic characterization of the polytime functions. Computational Complexity **2** (1992) 97–110
7. Bellantoni, S.J., Niggl, K.H., Schwichtenberg, H.: Higher type recursion, ramification and polynomial time. Ann. Pure Appl. Logic **104**(1-3) (2000) 17–30
8. Baillot, P., Lago, U.D.: Higher-Order Interpretations and Program Complexity. In Cégielski, P., Durand, A., eds.: Computer Science Logic (CSL'12). Volume 16 of Leibniz International Proceedings in Informatics (LIPIcs)., Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2012) 62–76

9. Baillot, P.: From proof-nets to linear logic type systems for polynomial time computing. In Rocca, S.R.D., ed.: TLCA. Volume 4583 of Lecture Notes in Computer Science., Springer (2007) 2–7
10. Baillot, P., Gaboardi, M., Mogbil, V.: A polytime functional language from light linear logic. In Gordon, A.D., ed.: ESOP. Volume 6012 of Lecture Notes in Computer Science., Springer (2010) 104–124
11. Avanzini, M., Eguchi, N., Moser, G.: A new order-theoretic characterisation of the polytime computable functions. In Jhala, R., Igarashi, A., eds.: APLAS. Volume 7705 of Lecture Notes in Computer Science., Springer (2012) 280–295
12. Avanzini, M., Moser, G.: Polynomial path orders. Logical Methods in Computer Science **9**(4) (2013)
13. Goerdt, A.: Characterizing complexity classes by general recursive definitions in higher types. Inf. Comput. **101**(2) (1992) 202–218
14. Goerdt, A.: Characterizing complexity classes by higher type primitive recursive definitions. Theor. Comput. Sci. **100**(1) (1992) 45–66
15. Kristiansen, L., Niggl, K.H.: On the computational complexity of imperative programming languages. Theor. Comput. Sci. **318**(1-2) (2004) 139–161
16. Bonfante, G.: Observation of implicit complexity by non confluence. In Baillot, P., ed.: Proceedings International Workshop on Developments in Implicit Computational complExity, DICE 2010, Paphos, Cyprus, 27-28th March 2010. Volume 23 of EPTCS. (2010) 15–29
17. Terese, ed.: Term Rewriting Systems. Volume 55 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press (2003)
18. Jones, N.D.: Computability and complexity: from a programming perspective. The MIT Press (1997)

A Proofs of Sections 2 and 3 omitted from the main text

A.1 Proofs of properties of the relations $\rhd_k^*$:

The statement: "For any $U, V \in \mathbf{2}\langle\Delta\rangle_k$, we have $U + V \rhd_k^* V$." is proved by induction on $Card(Supp(U+V) \setminus Supp(V))$:

- $Card(Supp(U+V)\setminus Supp(V)) = 0$: $Supp(U+V) = Supp(V)$, hence $U+V = V$, hence $U + V \rhd_k^* V$.
- $Card(Supp(U+V) \setminus Supp(V)) > 0$: let $u \in Supp(U) \setminus Supp(V)$; we set $U' = \sum_{v \in Supp(U)\setminus\{u\}} v$; by the induction hypothesis we have $U' + V \rhd_k^* V$; moreover we have $U + V \rhd_k U' + V$, hence $U + V \rhd_k^* V$.

In order to prove the property "Let $q \in \mathbb{N}$. Let $W_1, \ldots, W_q, V_1, \ldots, V_q \in \mathbf{2}\langle\Delta\rangle_k$ such that $W_1 \rhd_k^* V_1$, $\ldots$, $W_q \rhd_k^* V_q$. Then, for any $C[] \in \Theta_q$ such that $C[W_1, \ldots, W_q] \in \mathbf{2}\langle\Delta\rangle_k$, we have $C[W_1, \ldots, W_q] \rhd_k^* C[V_1, \ldots, V_q]$, first we prove the following fact:

Fact 1 *Let $k \in \mathbb{N}$. Let $W, V \in \mathit{2}\langle\Delta\rangle$ such that $W \rhd_k V$. Let $C[] \in \Theta_1$ such that $\|C[W]\| \leq k$. Then $C[W] \rhd_k C[V]$.*

Proof: Let $C'[] \in \Theta_1$ such that $W \rhd_{C'[]} V$. We have $C[C'[]] \in \Theta_1$ and $C[W] \rhd_{C[C'[]]} C[V]$.

Now, we can prove the property by induction on q:

- If $q = 0$, then $C[W_1, \ldots, W_q] = C[V_1, \ldots, V_q]$.
- If $q = 1$, then apply Fact 1.
- If $q > 1$ and $C[] = C'[f(U_1, \ldots, U_m)]$ with $C' \in \Theta_1$, $U_1 \in \Theta_{i_1}, \ldots, U_m \in \Theta_{i_m}$, then we set $j_0 = \max\{j \in \{1, \ldots, m\};\ U_j \notin \Theta_0\}$. We set $n = \sum_{i=1}^{j_0-1} m_i$. By the induction hypothesis, we have

$$\begin{aligned} &C'[U_1, \ldots, U_{j_0-1}, U_{j_0}[W_{n+1}, \ldots, W_{n+m_{j_0}}], U_{j_0+1}, \ldots, U_m][W_1, \ldots, W_n] \\ \rhd_k^* \ &C'[U_1, \ldots, U_{j_0-1}, U_{j_0}[W_{n+1}, \ldots, W_{n+m_{j_0}}], U_{j_0+1}, \ldots, U_m][V_1, \ldots, V_n] \end{aligned}$$

  and, again by the induction hypothesis, we have

$$\begin{aligned} &C'[U_1[V_1, \ldots, V_{m_1}], \ldots, U_{j_0-1}[V_{n-m_{j_0-1}}, \ldots, V_n], U_{j_0}, \ldots, U_m][W_{n+1}, \ldots, W_{n+m_{j_0}}] \\ \rhd_k^* \ &C'[U_1[V_1, \ldots, V_{m_1}], \ldots, U_{j_0-1}[V_{n-m_{j_0-1}}, \ldots, V_n], U_{j_0}, \ldots, U_m][V_{n+1}, \ldots, V_{n+m_{j_0}}]. \end{aligned}$$

In order to prove the property "Let $q \in \mathbb{N}$. Let $W_1, \ldots, W_q, V_1, \ldots, V_q \in \mathbf{2}\langle\Delta\rangle_k$ such that $W_1 \rhd_k^* V_1$, $\ldots$, $W_q \rhd_k^* V_q$. Then we have $\sum_{j=1}^q W_j \rhd_k^* \sum_{j=1}^q V_j$.", first we prove the two following facts:

Fact 2 *Let $C[] \in \Theta_1$. Let $U', V' \in \mathit{2}\langle\Delta\rangle$ such that $U' \rhd_{C[]} V'$. Then, for any $U \in \mathit{2}\langle\Delta\rangle$, we have $U + U' \rhd_{U+C[]} U + V'$.*

Fact 3 *Let $U', V' \in \mathit{2}\langle\Delta\rangle$ such that $U' \rhd^* V'$. Then, for any $U \in \mathit{2}\langle\Delta\rangle$, we have $U + U' \rhd^* U + V'$.*

Proof: We prove, by induction on n, that, for any $n \in \mathbb{N}$, for any $U, U_0, \ldots, U_n \in \mathbf{2}\langle \Delta \rangle$ such that $U_0 = U'$, $V' = U_n$ and, for any $i \in \{0, \ldots, n-1\}$, $U_i \rhd U_{i+1}$, we have $U + U' \rhd^* U + V'$:

- $n = 0$: $U' = V'$, hence $U + U' \rhd^* U + V'$;
- $n > 0$: by the induction hypothesis, we have $U + U' \rhd^* U + U_{n-1}$; now, by Fact 2, we have $U + U_{n-1} \rhd U + V'$.

Now, we can prove the property by induction on m:

- If $m = 0$, then $\sum_{i=1}^{m} U_i = \sum_{i=1}^{m} V_i$.
- If $m > 0$, then, by the induction hypothesis, we have $\sum_{i=1}^{m-1} U_i \rhd_k^* \sum_{i=1}^{m-1} V_i$. By Fact 3, we have $\sum_{i=1}^{m} U_i \rhd_k^* \sum_{i=1}^{m-1} V_i + U_m$. Again by Fact 3, we have $\sum_{i=1}^{m-1} V_i + U_m \rhd_k^* \sum_{i=1}^{m} V_i$.

A.2 Proof of Lemma 1:

We set $\Omega = \{(\rho, v, C[]) \in \mathbf{Seq} \times \mathcal{U}_0 \times \Phi_1;\ \rho \in \mathbf{Seq}(C[v], -)\}$.

First, we prove the following fact:

Fact 4 *Let $(\rho, v, C[]) \in \Omega$. For any $\rho' \in \mathcal{R}(\rho, v, C[])$, we have $length(\rho') \leq length(\rho)$.*

Proof: By induction on $length(\rho)$.

- If $length(\rho) = 0$, then, for any $\rho' \in \mathcal{R}(\rho, v, C[])$, we have $\rho' = id_v$, hence $length(\rho') = 0$.
- If $length(\rho) > 0$ and $\rho = C_0[u] \rightarrow_{C_0[],(l,r)} C_0[u'];\ \rho_0$ with $C_0[] = C[C'[]]$, then, for any $\rho' \in \mathcal{R}(\rho, v, C[])$, there exists $\rho'_0 \in \mathcal{R}(\rho_0, C'[u'], C[])$ such that $\rho' = (v \rightarrow_{C'[],(l,r)} C'[u'];\ \rho'_0)$; by the induction hypothesis, we have $length(\rho'_0) \leq length(\rho_0) = length(\rho) - 1$, hence $length(\rho') = length(\rho'_0) + 1 \leq length(\rho) - 1 + 1 = length(\rho)$.
- If $length(\rho) > 0$ and $\rho = C_0[u] \rightarrow_{C_0[],(l,r)} C_0[u'];\ \rho_0$ with no $C'[] \in \Phi_1$ such that $C_0[] = C[C'[]]$, then, for any $\rho' \in \mathcal{R}(\rho, v, C[])$,
  - $\rho' = id_v$, and in this case $length(\rho') = 0$;
  - or there exists $C''[] \in \Phi_1$ such that $\langle v | C''[] \rangle \in \langle v | C[] \rangle \setminus C_0[u] \rightarrow_{C_0[],(l,r)} C_0[u']$ and $\rho' \in \mathcal{R}(\rho_0, v, C''[])$, and in this case we just apply the induction hypothesis to obtain $length(\rho') \leq length(\rho_0) = length(\rho) - 1$.

Then we can prove the following fact:

Fact 5 *Let $v \in \mathcal{U}_0$, $C[] \in \Phi_1$ and $c \in \mathcal{T}_0(\mathcal{C})$. Let $\rho \in \mathbf{Seq}(C[v], c)$. Let $\rho' \in \mathcal{R}(\rho, v, C[])$. If $length(\rho') = length(\rho)$, then $C[] = \Box$.*

Proof: By induction on $length(\rho)$, using Fact 4.

Now we prove, by induction on r, the following fact:

Fact 6 *Let $r \in \mathcal{T}(\mathcal{C}_0 \cup \mathcal{D})$. Let $\sigma : \mathcal{X} \rightharpoonup \mathcal{U}_0$ such that $Oc(r) \subseteq dom(\sigma)$. Then $r^\sigma \in \mathcal{U}_0$.*

Finally we can prove Lemma 1:

Proof: By induction on $length(\rho)$. If $length(\rho) = 0$, then $E[u_1, \ldots, u_m] = c \in \mathcal{T}_0(\mathcal{C})$, hence $m = 0$ or ($m = 1$ and $E[] = \Box$): in case $m = 1$, we have $\{c\} \subseteq Supp(U_1)$, hence $E[U_1] = U_1 \rhd^* c$. If $length(\rho) > 0$, then we assume that $\rho = C_0[f(v_1, \ldots, v_q)] \to_{C_0[],(l,r)} C_0[v'];\ \rho_0$ with $C_0[] \in \Phi_1$: first, notice that, by Fact 6, $v' \in \mathcal{U}_0$; now, we distinguish between two cases:

- case $E[] = E'[\underbrace{\Box, \ldots, \Box}_{j\ times}, f(E'_1[], \ldots, E'_q[]), \underbrace{\Box, \ldots, \Box}_{k\ times}]$, where $E'_1[] \in \Phi_{m_1}, \ldots, E'_q[] \in \Phi_{m_q}$ with $j+m_1+\ldots+m_q+k = m$ and $E'[] \in \Phi_{j+k+1}$ such that $E'[u_1, \ldots, u_j, \Box, u_{m-k}, \ldots, u_m] = C_0[]$:
  We set $D_0[] = E'[U_1, \ldots, U_j, \Box, U_{m-k}, \ldots, U_m] \in \Theta_1$.
  For any $l \in \{1, \ldots, q\}$, we set $V_l = \mathcal{N}(\rho, v_l, C_0[f(v_1, \ldots, v_{l-1}, \Box, v_{l+1}, \ldots, v_q)])$ and $W_l = E'_l[U_{j+m_1+\ldots+m_{l-1}+1}, \ldots, U_{j+m_1+\ldots+m_l}]$: by Fact 5, for any $c' \in Supp(V_l)$, there exists $\rho' \in \mathcal{R}(\rho, v_l, C_0[f(v_1, \ldots, v_{l-1}, \Box, v_{l+1}, \ldots, v_q)]) \cap \mathbf{Seq}(v_l, c')$ such that $length(\rho') < length(\rho)$, hence, by the induction hypothesis, we have $W_l \rhd^* V_l$.
  We set $C[] = E'[U_1, \ldots, U_j, f(\Box, \ldots, \Box), U_{m-k}, \ldots, U_m] \in \Theta_q$. We obtain
$$\begin{aligned} E[U_1, \ldots, U_m] &= C[W_1, \ldots, W_q] \\ &\rhd^* C[V_1, \ldots, V_q] \\ &= D_0[f(V_1, \ldots, V_q)]. \end{aligned}$$
  Let $\sigma : \mathcal{X} \rightharpoonup \mathcal{U}_0$ such that $l^\sigma = f(v_1, \ldots, v_q)$ and $r^\sigma = v'$.
  We set $\sigma'' : \begin{array}{l} \mathcal{X} \rightharpoonup \mathcal{U}_0 \cup \{\Box\} \\ x \mapsto \begin{cases} \Box & if\ x = c_l; \\ \sigma(x) & if\ there\ is\ no\ l \in \{1, \ldots, q\}\ such\ that\ x = c_l. \end{cases} \end{array}$
  We set $\sigma' : \begin{array}{l} \mathcal{X} \rightharpoonup \mathbf{2}\langle \Delta \rangle \\ x \mapsto \begin{cases} V_l & if\ x = c_l; \\ \sigma(x) & if\ there\ is\ no\ l \in \{1, \ldots, q\}\ such\ that\ x = c_l. \end{cases} \end{array}$
  Let $M \in \mathbb{N}$ such that $r^{\sigma''} \in \Phi_M$. Let $w_1, \ldots, w_M \in \{v_1, \ldots, v_q\}$ such that $r^{\sigma''}[w_1, \ldots, w_M] = r^{\sigma'}$. Let $F : \{1, \ldots, M\} \to \{1, \ldots, q\}$ such that, for any $l \in \{1, \ldots, M\}$, $w_l = v_{F(l)}$. For any $l \in \{1, \ldots, M\}$, we set $E''_l[] = r^{\sigma''}[w_1, \ldots, w_{l-1}, \Box, w_{l+1}, \ldots, w_M]$ and $T_l = \sum_{c \in \mathcal{N}(\rho_0, w_l, E'[u_1, \ldots, u_j, E''_l[], u_{m-k}, \ldots, u_m])} c$: we have $Supp(T_l) \subseteq Supp(V_{F(l)})$; indeed, we have
$$\begin{aligned} &\mathcal{R}(\rho_0, w_l, E'[u_1, \ldots, u_j, E''_l[], u_{m-k}, \ldots, u_m]) \\ &\subseteq \mathcal{R}(\rho, v_{F(l)}, C_0[f(v_1, \ldots, v_{F(l)-1}, \Box, v_{F(l)+1}, \ldots, v_q)]) \end{aligned}$$
  and, from this inclusion we immediately obtain
$$\begin{aligned} &\mathcal{N}(\rho_0, w_l, E'[u_1, \ldots, u_j, E''_l[], u_{m-k}, \ldots, u_m]) \\ &\subseteq \mathcal{N}(\rho, v_{F(l)}, C_0[f(v_1, \ldots, v_{F(l)-1}, \Box, v_{F(l)+1}, \ldots, v_q)]) \end{aligned}$$
  i.e. $Supp(T_l) \subseteq Supp(V_{F(l)})$.

Moreover, for any $l \in \{1, \ldots, j\}$, we have

$$\mathcal{N}(\rho_0, u_l, E'[u_1, \ldots, u_{l-1}, \Box, u_{l+1}, \ldots, u_j, v', u_{m-k}, \ldots, u_m]) =$$
$$\mathcal{N}(\rho, u_l, E'[u_1, \ldots, u_{l-1}, \Box, u_{l+1}, \ldots, u_j, f(v_1, \ldots, v_q), u_{m-k}, \ldots, u_m])$$

and, for any $l \in \{m-k, \ldots, m\}$, we have

$$\mathcal{N}(\rho_0, u_l, E'[u_1, \ldots, u_j, v', u_{m-k}, \ldots, u_{l-1}, \Box, u_{l+1}, \ldots, u_m]) =$$
$$\mathcal{N}(\rho, u_l, E'[u_1, \ldots, u_j, f(v_1, \ldots, v_q), u_{m-k}, \ldots, u_{l-1}, \Box, u_{l+1}, \ldots, u_m])$$

Since $\rho_0 \in \mathbf{Seq}(E'[u_1, \ldots, u_j, v', u_{m-k}, \ldots, u_m], c)$ and $v' \in \mathcal{U}_0$, by the induction hypothesis, we have $D_0[r^{\sigma'}] \rhd^* c$.
Lastly, we have $f(V_1, \ldots, V_q) \rhd_\Box r^{\sigma'}$, hence $D_0[f(V_1, \ldots, V_q)] \rhd_{D_0[]} D_0[r^{\sigma'}]$.

– case $C_0[] = E[u_1, \ldots, u_{j-1}, C'[], u_{j+1}, \ldots, u_m]$ with $C'[] \in \Phi_1$ and $u_j = C'[f(v_1, \ldots, v_q)]$: for any $l \in \{1, \ldots, q\}$, we set $u'_l = \begin{cases} u_l \ \textit{if}\ l \neq j; \\ C'[v'] \ \textit{if}\ l = j; \end{cases}$
we have $\mathcal{R}(\rho_0, u'_l, E[u'_1, \ldots, u'_{l-1}, \Box, u'_{l+1}, \ldots, u'_m]) =$
$\begin{cases} \mathcal{R}(\rho, u_l, E[u_1, \ldots, u_{l-1}, \Box, u_{l+1}, \ldots, u_m]) \ \textit{if}\ l \neq j; \\ \left\{ \begin{array}{l} (C'[f(v_1, \ldots, v_q)] \rightarrow_{C'[],(l,r)} C'[v'];\ \rho'_0); \\ \rho'_0 \in \mathcal{R}(\rho_0, u'_l, E[u'_1, \ldots, u'_{l-1}, \Box, u'_{l+1}, \ldots, u'_m]) \end{array} \right\} \ \textit{if}\ l = j; \end{cases}$
Hence,

$$\mathcal{N}(\rho_0, u'_l, E[u'_1, \ldots, u'_{l-1}, \Box, u'_{l+1}, \ldots, u'_m]) =$$
$$\mathcal{N}(\rho, u_l, E[u_1, \ldots, u_{l-1}, \Box, u_{l+1}, \ldots, u_m])$$

Since $\rho_0 \in \mathbf{Seq}(E[u'_1, \ldots, u'_{l+1}, \ldots, u'_m], c)$ and $u'_j \in \mathcal{U}_0$, by the induction hypothesis, we have $E[U_1, \ldots, U_m] \rhd^* c$.

A.3 Proof of Lemma 2:

First we prove the following lemma:

Lemma 5. *Let $t \in \mathcal{U}_0$. Let $(l, r) \in \mathcal{R}$ and $\varphi : \mathcal{X} \rightharpoonup 2\langle \mathcal{T}_0(\mathcal{C}) \rangle$ such that $t \downarrow u$ and $u \rhd_\Delta r^\varphi$. Then we have $t \downarrow r^\varphi$.*

Proof: For any $j \in \{1, \ldots, q\}$, we set $u'_j = \begin{cases} u_l \ \textit{if}\ c_j \in \mathcal{X}; \\ U_j \ \textit{if}\ c_j \notin \mathcal{X}. \end{cases}$

We set $\sigma : \begin{array}{l} \mathcal{X} \rightharpoonup \mathcal{U}_0 \\ x \ \mapsto \begin{cases} u_j \ \textit{if}\ c_j = x; \\ \varphi(x) \ \textit{otherwise.} \end{cases} \end{array}$

Notice that we have $f(u'_1, \ldots, u'_q) \rhd_\Box r^\sigma$. Now we have $u_1 \rightarrow^* u'_1, \ldots, u_q \rightarrow^* u'_q$, hence $t \rightarrow^* f(u_1, \ldots, u_q) \rightarrow^* f(u'_1, \ldots, u'_q) \rightarrow^\sigma_r$. So we just have to check that $r^\sigma \downarrow r^\varphi$. This is done by induction on r:

– $r = c_j \in \mathcal{X}$: $r^\sigma = u_j$ and $r^\varphi = U_j$;
– $r \in \mathcal{C}_0$ or ($r \in \mathcal{X}$ and there is no $j \in \{1, \ldots, q\}$ such that $r = c_j$): $r^\sigma = r^\varphi$, hence $r^\sigma \downarrow r^\varphi$;

– $r = g(v_1, \ldots, v_m)$: $r^\sigma = g(v_1{}^\sigma, \ldots, v_m{}^\sigma)$ and $r^\varphi = g(v_1{}^\varphi, \ldots, v_m{}^\varphi)$; by the induction hypothesis, we have $v_1{}^\sigma \downarrow v_1{}^\varphi$, ..., $v_m{}^\sigma \downarrow v_m{}^\varphi$, hence $r^\sigma \downarrow r^\varphi$.

Now, we can prove Lemma 2:

Proof: We prove, by induction on i, that, for any $i \in \mathbb{N}$, fo any $C[] \in \Delta_i^\Box$, for any $t \in \mathcal{U}_0$, $U, V \in \mathbf{2}\langle\Delta\rangle$ such that $t \downarrow U$ and $U \rhd_{C[]} V$, we have $t \downarrow V$.

– $i = 0$: $C[] = W + \Box$: we distinguish between two cases.
  1. There exist $u \in \Delta$ and $V' \in \mathbf{2}\langle\Delta\rangle$ such that $U = W + u$, $V = W + V'$ and $u \rhd_\Delta V'$: let $v \in \mathit{Supp}(V)$; if $v \in \mathit{Supp}(W)$, then, by assumption, $t \downarrow v$; if $v \in \mathit{Supp}(V')$, then we apply Lemma 5.
  2. There exists $u \in \Delta$ such that $U = V + u$: since $t \downarrow U$, we have $t \downarrow V$.
– $i > 0$: $C[] = W + f(U_1, \ldots, U_q)$ with $W \in \mathbf{2}\langle\Delta\rangle$ and $U_j = C'[] \in \Delta_{i-1}^\Box$: let $U'', V'' \in \mathbf{2}\langle\Delta\rangle$ such that $U = C[U'']$, $V = C[V'']$ and $U'' \rhd_\Box V''$; there exist $u_1, \ldots, u_q \in \mathcal{U}_0$ such that $t \to^* f(u_1, \ldots, u_q)$ and $u_1 \downarrow U_1$, ..., $u_{j-1} \downarrow U_{j-1}$, $u_{j+1} \downarrow U_{j+1}$, ..., $u_q \downarrow U_q$, $u_j \downarrow C'[U'']$. We have $C'[U''] \rhd_{C'[]} C'[V'']$, hence, by the induction hypothesis, $u_j \downarrow C'[V'']$. We thus obtain $t \downarrow f(U_1, \ldots, U_{j-1}, C'[V''], U_{j+1}, \ldots, U_q)$; moreover $t \downarrow W$, hence $t \downarrow V$.

A.4 Proof of Lemma 3:

First we state the following fact, which will used also in the proof of Proposition 2:

Fact 7 *Let $k \in \mathbb{N}$. Let $q \in \mathbb{N}$. Let $c_1, \ldots, c_q \in \mathcal{T}(\mathcal{C})$. Let $\varphi : \mathcal{X} \rightharpoonup \mathit{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$. Assume that, for any $j \in \{1, \ldots, q\}$, we have $Oc(c_j) \subseteq dom(\varphi)$ and $(c_j \notin \mathcal{X} \Rightarrow c_j{}^\varphi \in \mathcal{T}_0(\mathcal{C}))$. Let $t \in \Delta$. Let $W_1, \ldots, W_q, W_1', \ldots, W_q' \in \mathit{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$ such that*

– *for any $j \in \{1, \ldots, q\}$, $(c_j \notin \mathcal{X} \Rightarrow W_j = c_j{}^\varphi)$ and $\mathit{Supp}(W_j) \subseteq \mathit{Supp}(c_j{}^\varphi)$;*
– *for any $j \in \{1, \ldots, q\}$, $(c_j \notin \mathcal{X} \Rightarrow W_j' = c_j{}^\varphi)$ and $\mathit{Supp}(W_j') \subseteq \mathit{Supp}(W_j)$;*
– *and, for any $j \in \{1, \ldots, q\}$, $\mathit{Card}(\mathit{Supp}(W_j')) \leq \mathit{Card}(\mathit{Supp}(W_j)) \leq k$.*

Then $t^{\varphi(W_1, \ldots, W_q)} \rhd_k^* t^{\varphi(W_1', \ldots, W_q')}$.

Now, we can prove the lemma:

Proof: By induction on r:

– $r \in \mathcal{X}$: let $u \in \mathit{Supp}((r^\varphi)^*)$; for any $j \in \{1, \ldots, q\}$, let $W_j \in \mathbf{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$ such that $W_j = \begin{cases} u \ \textit{if } j = i; \\ c_j{}^\varphi \ \textit{if } c_j \notin \mathcal{X}; \\ 0 \ \textit{otherwise;} \end{cases}$
We then have $r^{\varphi(W_1, \ldots, W_q)} = u$, hence we have $r^{\varphi(W_1, \ldots, W_q)} \rhd_K^* u$;
– $r = g(u_1, \ldots, u_m)$ with $g \in \mathcal{D}_m \cap \mathcal{A}$: Let $W_1', \ldots, W_m' \in \mathbf{2}\langle\Delta\rangle$ such that, for any $j' \in \{1, \ldots, m\}$, $\mathit{Supp}(W_{j'}') \subseteq \mathit{Supp}((u_{j'}{}^\varphi)^*)$ and $\mathit{Card}(\mathit{Supp}(W_{j'}')) = \min\{1, \mathit{Card}(\mathit{Supp}((u_{j'}{}^\varphi)^*))\}$. For any $j \in \{1, \ldots, q\}$, for any $j' \in \{1, \ldots, m\}$, we set

$$\mathcal{W}_j^{j'} = \begin{cases} \{{c_j}^{\varphi}\} \textit{ if } c_j \notin \mathcal{X}; \\ \left\{ W \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle ; \begin{array}{l} \mathit{Supp}(W) \subseteq \mathit{Supp}({c_j}^{\varphi}) \textit{ and} \\ \mathit{Card}(\mathit{Supp}(W)) = \min\{ \mathit{Oc}(c_j, u_{j'}), \mathit{Card}(\mathit{Supp}({c_j}^{\varphi}))\} \end{array} \right\} \textit{ if } c_j \in \mathcal{X}. \end{cases}$$

For any $j' \in \{1, \ldots, m\}$, let $W_1^{j'} \in \mathcal{W}_1^{j'}, \ldots, W_q^{j'} \in \mathcal{W}_q^{j'}$ such that ${u_{j'}}^{\varphi({W_1}^{j'}, \ldots, {W_q}^{j'})} \rhd_K^* W'_{j'}$. For any $j \in \{1, \ldots, q\}$, we set $W_j = \sum_{j'=1}^{m} W_j^{j'}$: we have $W_j \in \mathcal{W}_j$. We have

$$\begin{aligned} g({u_1}^{\varphi(W_1, \ldots, W_q)}, \ldots, {u_m}^{\varphi(W_1, \ldots, W_q)}) &\rhd_K^* g({u_1}^{\varphi(W_1^1, \ldots, W_q^1)}, \ldots, {u_m}^{\varphi(W_1^m, \ldots, W_q^m)}) \\ &\qquad \textit{(by Fact 7)} \\ &\rhd_K^* g(W'_1, \ldots, W'_m) \end{aligned}$$

- $r = h(u_1, \ldots, u_m)$ with $h \in \mathcal{C}$: There exists $j \in \{1, \ldots, q\}$ such that $r \trianglelefteq c_j$, hence $r^{\varphi} \in \Delta_0$ and $(\forall W_1 \in \mathcal{W}_1, \ldots, W_q \in \mathcal{W}_q) r^{\varphi(W_1, \ldots, W_q)} = r^{\varphi}$. Now, since $r^{\varphi} \in \Delta_0$, we have $r^{\varphi *} = r^{\varphi}$.
- $r = g(u_1, \ldots, u_m)$ with $g \in \mathcal{D}_m \cap \mathcal{B}$: Notice that, for any $j \in \{1, \ldots, q\}$ such that $c_j \in \mathcal{X}$, we have $\mathit{Oc}(c_j, r) = 0$, hence $\mathit{Card}(\mathcal{W}_j) = 1$ and ${u_j}^{\varphi *} = {u_j}^{\varphi}$. Let $W_1 \in \mathcal{W}_1, \ldots, W_q \in \mathcal{W}_q$. For any $j' \in \{1, \ldots, m\}$, by the induction hypothesis, we have ${u_{j'}}^{\varphi(W_1, \ldots, W_q)} \rhd_K^* ({u_{j'}}^{\varphi})^* = {u_{j'}}^{\varphi}$; therefore $g({u_1}^{\varphi(W_1, \ldots, W_q)}, \ldots, {u_m}^{\varphi(W_1, \ldots, W_q)}) \rhd_K^* g({u_1}^{\varphi}, \ldots, {u_m}^{\varphi}) = g(u_1, \ldots, u_m)^{\varphi} = (g(u_1, \ldots, u_m)^{\varphi})^*$.

A.5 Lemmas and Facts used in the complete proof of Proposition 2 and which do not appear in the principal part of the text:

Lemma 6. *Let $k \in \mathbb{N}$. Let $U, V \in \mathit{2}\langle \Delta \rangle$ such that, for any $v \in \mathit{Supp}(V)$, there exists $u \in \mathit{Supp}(U)$ such that $u \rhd_k^* v$. Then $U \rhd_k^* V$.*

Proof: There exists a function $\psi : \mathit{Supp}(V) \to \mathit{Supp}(U)$ such that, for any $v \in \mathit{Supp}(V)$, we have $\varphi(v) \rhd_k^* v$. We obtain $\sum_{w \in im(\psi)} w \rhd_k^* V$, hence $U \rhd_k^* \sum_{w \in im(\psi)} w$.

Lemma 7. *Let $n, k \in \mathbb{N}$. Let $U, V \in \mathit{2}\langle \Delta \rangle$ such that $U \rhd_k^n V$. Let $V_0 \in \mathit{2}\langle \Delta \rangle$ such that $\mathit{Supp}(V_0) \subseteq \mathit{Supp}(V)$. Then there exist $n' \leq n$ and $U_0 \in \mathit{2}\langle \Delta \rangle$ such that*

- *$\mathit{Supp}(U_0) \subseteq \mathit{Supp}(U)$;*
- *$\mathit{Card}(\mathit{Supp}(U_0)) \leq \mathit{Card}(\mathit{Supp}(V_0))$;*
- *and $U_0 \rhd_k^{n'} V_0$.*

Proof of Lemma 7: First we prove the following fact:

Fact 8 *Let $k \in \mathbb{N}$. Let $U, V \in \mathit{2}\langle \Delta \rangle$ such that $U \rhd_k V$. Let $V_0 \in \mathit{2}\langle \Delta \rangle$ such that $\mathit{Supp}(V_0) \subseteq \mathit{Supp}(V)$. Then there exists $U_0 \in \mathit{2}\langle \Delta \rangle$ such that*

- *$\mathit{Supp}(U_0) \subseteq \mathit{Supp}(U)$;*
- *$\mathit{Card}(\mathit{Supp}(U_0)) \leq \mathit{Card}(\mathit{Supp}(V_0))$;*
- *and $U_0 \rhd_k^* V_0$.*

Proof: Let $C[] \in \Theta_1$ such that $U \rhd_{C[]} V$. We split on cases according to the shape of $C[]$:

- If $C[] = W + \Box$, $U = W + u$, $V = W + V'$ and $u \rhd_\Delta V'$, let $V_1, V_2 \in \mathbf{2}\langle\Delta\rangle$ such that $\mathit{Supp}(V_1) \subseteq \mathit{Supp}(W)$, $\mathit{Supp}(V_2) \subseteq \mathit{Supp}(V')$, $\mathit{Supp}(V_1) \cap \mathit{Supp}(V_2) = \emptyset$ and $V = V_1 + V_2$; if $V_2 = 0$, then we set $U_0 = V_1$, otherwise we set $U_0 = V_1 + u$.
- If $C[] = W + \Box$, $U = W + u$ and $V = W$, then we set $U_0 = V_0$.
- If $C[] = W + f(U_1, \ldots, U_m)$ with $U_j \in \Theta_1$, then $U = W + f(U_1, \ldots, U_{j-1}, U_j[U'], U_{j+1}, \ldots, U_m)$ and $V = W + f(U_1, \ldots, U_{j-1}, U_j[V'], U_{j+1}, \ldots, U_m)$ and $U' \rhd_\Box V'$, let $V_1, V_2 \in \mathbf{2}\langle\Delta\rangle$ such that $\mathit{Supp}(V_1) \subseteq \mathit{Supp}(W)$ and such that $\mathit{Supp}(V_2) \subseteq \mathit{Supp}(f(U_1, \ldots, U_{j-1}, U_j[V'], U_{j+1}, \ldots, U_m))$, $\mathit{Supp}(V_1) \cap \mathit{Supp}(V_2) = \emptyset$ and $V = V_1 + V_2$; if $V_2 = 0$, then we set $U = V_1$, otherwise we set $U = V_1 + f(U_1, \ldots, U_{j-1}, U_j[U'], U_{j+1}, \ldots, U_m)$.
- If $C[] = W + f(U_1, \ldots, U_m)$ with $U_j \in \Theta_1$ and $V = W$, then we set $U_0 = V_0$.

Now, we can prove the lemma:

Proof:

- If $n = 0$, then $U = V$, so we can set $n' = 0$ and $U_0 = V_0$.
- If $n > 0$, then there exists $T \in \mathbf{2}\langle\Delta\rangle$ such that $U \rhd_k^{n-1} T$ and $T \rhd_k V$; by Fact 8, there exists $U' \in \mathbf{2}\langle\Delta\rangle$ such that
  - $\mathit{Supp}(U') \subseteq \mathit{Supp}(T)$;
  - $\mathit{Card}(\mathit{Supp}(U')) \leq \mathit{Card}(\mathit{Supp}(V_0))$;
  - and $U' \rhd_k V_0$;

  now, by induction hypothesis, there exist $n'' \leq n - 1$ and $U_0 \in \mathbf{2}\langle\Delta\rangle$ such that
  - $\mathit{Supp}(U_0) \subseteq \mathit{Supp}(U')$;
  - $\mathit{Card}(\mathit{Supp}(U_0)) \leq \mathit{Card}(\mathit{Supp}(U'))$;
  - and $U_0 \rhd_k^{n''} U'$.

  We set $n' = n'' + 1$.

Fact 9 *Let $q \in \mathbb{N}$. Let $c_1, \ldots, c_q \in \mathcal{T}(\mathcal{C})$. Let $\varphi : \mathcal{X} \rightharpoonup \mathit{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$. Assume that, for any $l \in \{1, \ldots, q\}$, we have $\mathit{Oc}(c_l) \subseteq \mathit{dom}(\varphi)$ and $(c_l \notin \mathcal{X} \Rightarrow {c_l}^\varphi \in \mathcal{T}_0(\mathcal{C}))$. Let $W_1, \ldots, W_q \in \mathit{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$ such that, for any $l \in \{1, \ldots, q\}$, $(c_l \notin \mathcal{X} \Rightarrow W_l = {c_l}^\varphi)$ and $\mathit{Supp}(W_l) \subseteq \mathit{Supp}({c_l}^\varphi)$. Then, for any $l \in \{1, \ldots, q\}$, we have $W_l = {c_l}^{\varphi(W_1, \ldots, W_q)}$.*

Proof: For any $l \in \{1, \ldots, q\}$ such that $c_l \notin \mathcal{X}$, for any $x \in \mathit{Oc}(c_l)$, we have $\varphi(x) = \varphi_{(W_1, \ldots, W_q)}(x)$, hence ${c_l}^{\varphi(W_1, \ldots, W_q)} = {c_l}^\varphi = W_l$.

A.6 Complete proof of Proposition 2:

Proof: The proof is by induction on i.

- If $i = 0$, then $C[] = U_0 + \Box$ for some $U_0 \in \mathbf{2}\langle\Delta\rangle$; we distinguish between two cases:
  - $V = U_0$ and there exists $u \in \Delta$ such that $U = U_0 + u$: in this case, we have $\|U^*\|, \|{U_0}^*\| \leq K$. Thus we have $U^* = {U_0}^* + u^* \rhd_K^* {U_0}^* = V^*$.

- $U = U_0 + f(c_1,\ldots,c_q)^\varphi$, $V = U_0 + r^\varphi$ with $f \in \mathcal{D}_q$, $c_1,\ldots,c_q \in \mathcal{T}(\mathcal{C})$, $\varphi : \mathcal{X} \rightharpoonup \mathbf{2}\langle\mathcal{T}_0(\mathcal{C})\rangle$ such that, for any $j \in \{1,\ldots,q\}$, $c_j \notin \mathcal{X} \Rightarrow {c_j}^\varphi \in \mathcal{T}_0(\mathcal{C})$:
  For any $j \in \{1,\ldots,q\}$, we define $\mathcal{W}_j$ as in Lemma 3 and we set

$$\mathcal{W'}_j = \begin{cases} \{{c_j}^\varphi\} \textit{ if } c_j \notin \mathcal{X}; \\ \left.\left\{W \in \mathbf{2}\langle\mathcal{T}_0(\mathcal{C})\rangle; \begin{array}{l} \mathit{Supp}(W) \subseteq \mathit{Supp}({c_j}^\varphi) \textit{ and} \\ \mathit{Card}(\mathit{Supp}(W)) = \min\{1, \mathit{Card}(\mathit{Supp}({c_j}^\varphi))\} \end{array}\right.\right\} \begin{array}{l} \textit{if } f \in \mathcal{A} \\ \textit{and } c_j \in \mathcal{X}; \end{array} \\ \left.\left\{W \in \mathbf{2}\langle\mathcal{T}_0(\mathcal{C})\rangle; \begin{array}{l} \mathit{Supp}(W) \subseteq \mathit{Supp}({c_j}^\varphi) \textit{ and} \\ \mathit{Card}(\mathit{Supp}(W)) = \min\{K, \mathit{Card}(\mathit{Supp}({c_j}^\varphi))\} \end{array}\right.\right\} \begin{array}{l} \textit{if } f \in \mathcal{B} \\ \textit{and } c_j \in \mathcal{X}. \end{array} \end{cases}$$

  We have

$$\begin{aligned} (f(c_1,\ldots,c_q)^\varphi)^* &= \sum_{W_1\in\mathcal{W'}_1,\ldots,W_q\in\mathcal{W'}_q} f({c_1}^{\varphi(W_1,\ldots,W_q)},\ldots,{c_q}^{\varphi(W_1,\ldots,W_q)}) \textit{ (by Fact 9)} \\ &\rhd_K^* \sum_{W_1\in\mathcal{W}_1,\ldots,W_q\in\mathcal{W}_q} f(c_1,\ldots,c_q)^{\varphi(W_1,\ldots,W_q)} \textit{ (by Fact 7)} \\ &\rhd_K^* \sum_{W_1\in\mathcal{W}_1,\ldots,W_q\in\mathcal{W}_q} r^{\varphi(W_1,\ldots,W_q)} \\ &\rhd_K^* (r^\varphi)^* \textit{ (by Lemma 3).} \end{aligned}$$

  We have $\|{U_0}^*\| \leq K$, hence we obtain $U^* = {U_0}^* + (f(c_1,\ldots,c_q)^\varphi)^* \rhd_K^* {U_0}^* + (r^\varphi)^* = V^*$.

– If $i > 0$, $U = C[U_0]$, $V = C[V_0]$, $C[] = U' + f(U_1,\ldots,U_m)$, $k \in \{1,\ldots,m\}$, $U_k \in \Delta_{i-1}^{\square}$ and $U_0 \rhd_\square V_0$, then, by the induction hypothesis, $U_k[{U_0}^*] \rhd_K^* U_k[{V_0}^*]$; Let $V_1,\ldots,V_m \in \mathbf{2}\langle\Delta\rangle$ such that
  - for any $j \in \{1,\ldots,m\}\backslash\{k\}$, $\mathit{Supp}(V_j) \subseteq \mathit{Supp}({U_j}^*)$ and $\mathit{Card}(\mathit{Supp}(V_j)) = \min\{1, \mathit{Card}(\mathit{Supp}({U_j}^*))\}$
  - and $\mathit{Supp}(V_k) \subseteq \mathit{Supp}(U_k[V_0]^*)$ and $\mathit{Card}(\mathit{Supp}(V_k)) = \begin{cases} \min\{1, \mathit{Card}(\mathit{Supp}(U_k[V_0]^*))\} & \text{if } f \in \mathcal{A}; \\ \min\{K, \mathit{Card}(\mathit{Supp}(U_k[V_0]^*))\} & \text{if } f \in \mathcal{B}. \end{cases}$

  By Lemma 7, there exists $V \in \mathbf{2}\langle\Delta\rangle$ such that $\mathit{Supp}(V) \subseteq \mathit{Supp}(U_k[U_0])$, $\mathit{Card}(\mathit{Supp}(V)) \leq \mathit{Card}(\mathit{Supp}(V_k))$ and $V \rhd_K^* V_k$, and hence $f(V_1,\ldots,V_{k-1},V,V_{k+1},\ldots,V_m) \rhd_K^* f(V_1,\ldots,V_k)$. By Lemma 6, we obtain $f(U_1,\ldots,U_{k-1},U_k[U_0],U_{k+1},\ldots,U_m)^* \rhd_K^* f(U_1,\ldots,U_{k-1},U_k[V_0],U_{k+1},\ldots,U_m)^*$; moreover we have $\|U'^*\| \leq K$, hence $U^* = U'^* + f(U_1,\ldots,U_{k-1},U_k[U_0],U_{k+1},\ldots,U_m)^* \rhd_K^* U'^* + f(U_1,\ldots,U_{k-1},U_k[V_0],U_{k+1},\ldots,U_m)^* = V^*$.

B Details of Section 4

We extend the notion of size to any element of of $\mathbf{2}\langle \Delta \rangle$. First, for any $u \in \Delta$, we define $|u|$ by induction on $level(u)$ as follows: $|f(V_1, \ldots, V_m)| = 1 + \sum_{j=1}^{m} |V_j|$. Then, for any $V \in \mathbf{2}\langle \Delta \rangle$, for any $V \in \mathbf{2}\langle \Delta \rangle$, we set $|V| = \sum_{v \in Supp(V)} |v|$. Notice that, for any $n \in \mathbb{N}$, for any $V \in \mathcal{V}_n(c)$, we have $|V| \leq n + A^n \cdot K \cdot |c|$.

The following lemma wil be used in the proof of Proposition 3.

Lemma 8. *Let $q \in \mathbb{N}$. Let $C_1, \ldots, C_q \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$ such that $Card(Supp(C_1))$, $\ldots$, $Card(Supp(C_q)) \leq K$, let $c \in \mathcal{T}_0(\mathcal{C})$ and $f \in \mathcal{D}_q$. There exists $\rho = (U_1, \ldots, U_n) \in \boldsymbol{Seq}_{\Delta,K}(f(C_1, \ldots, C_q), c)$ if, and only if, there exist $C'_1, \ldots, C'_q \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$, $V \in \mathbf{2}\langle \Delta \rangle$ and $\rho' \in \boldsymbol{Seq}_{\Delta,K}(V, c)$ such that $Supp(C'_1) \subseteq Supp(C_1)$, $\ldots$, $Supp(C'_q) \subseteq Supp(C_q)$, $f(C'_1, \ldots, C'_q) \rhd_\Delta V$ and $length(\rho') < length(\rho)$.*

Proof: By induction on n.

- If there exists $V \in \mathbf{2}\langle \Delta \rangle$ such that $f(C_1, \ldots, C_q) \rhd_\Delta V$ and $U_2 = f(C_1, \ldots, C_q) + V$, then, by Lemma 7, we have $V \rhd_K^{n-1} c$ or $f(C_1, \ldots, C_q) \rhd_K^{n-1} c$; now, if $f(C_1, \ldots, C_q) \rhd_K^{n-1} c$, then we can apply the induction hypothesis.
- If there exist $C'_1, \ldots, C'_q \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$ such that $Supp(C'_1) \subseteq Supp(C_1)$, $\ldots$, $Supp(C'_q) \subseteq Supp(C_q)$ and $U_2 = f(C'_1, \ldots, C'_q)$, then we just apply the induction hypothesis.

We proceed to prove that certain reduction sequences may be shortened using an analogue of memoization.

Definition 19. *For any $C[] \in \Theta_1$, for any $V \in \mathbf{2}\langle \Delta \rangle$ such that $C[V] \in \Delta$, for any $\rho = (U_1, \ldots, U_n) \in \boldsymbol{Seq}_\Delta(C[V], -)$, we define $\mathcal{R}_\Delta(\rho, V, C[]) \in \boldsymbol{Seq}_\Delta(V, -)$ by induction on n:*

- *if $n = 0$, then $\mathcal{R}_\Delta(\rho, V, C[]) = id_V$;*
- *if $n > 0$ and $C[] = \Box$, then $\mathcal{R}_\Delta(\rho, V, C[]) = \rho$;*
- *if $\rho = (C_0[u] \rhd_{C_0[]} C_0[U'];\ \rho_0)$, $u \rhd_\Box U'$ and $C[] \neq \Box$, then we distinguish between the following cases:*
  - *if there exists $D[] \in \Theta_2$ such that $C_0 = D[V, \Box]$ and $C[] = D[\Box, u]$, then $\mathcal{R}_\Delta(\rho, V, C[]) = \mathcal{R}_\Delta(\rho_0, V, D[\Box, U'])$;*
  - *if $C_0[] = C[C'[]]$ with $C'[] \in \Theta_1$, then $\mathcal{R}_\Delta(\rho, V, C[]) = (V \rhd_{C'[]} C'[U'];\ \mathcal{R}_\Delta(\rho_0, C'[U'], C[]))$;*
  - *if $u \rhd_\Delta U'$ and $C[] = C_0[C'[]]$ with $C'[] \in \Theta_1 \setminus \{\Box\}$, then $\mathcal{R}_\Delta(\rho, V, C[]) = id_V$;*
  - *if $U' = 0$ and $C[] = C_0[C'[]]$ with $C'[] \in \Theta_1 \setminus \{\Box\}$, then $\mathcal{R}_\Delta(\rho, V, C[]) = id_0$.*

Fact 10 *For any $C[] \in \Theta_1$, for any $V \in \mathbf{2}\langle \Delta \rangle$ such that $C[V] \in \Delta$, for any $V' \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$, for any $\rho = (U_1, \ldots, U_n) \in \boldsymbol{Seq}_\Delta(C[V], V')$, there exists $W' \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$ such that $\mathcal{R}_\Delta(\rho, V, C[]) \in \boldsymbol{Seq}_\Delta(C[V], W')$.*

Proof: Let $W' \in \mathbf{2}\langle \Delta \rangle$ such that $\rho \in \mathbf{Seq}_\Delta(C[V], W')$. We prove, by induction on n, that $W' \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$.

If $n = 0$, then $W' = V$ and $C[V] = V' \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$, hence $W' \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$.

If $\rho = (C_0[u] \rhd_{C_0[]} C_0[U'];\ \rho_0)$, $u \rhd_\Delta U'$, $C[] \neq \Box$ and $C[] = C_0[C'[]]$ with $C'[] \in \Theta_1 \setminus \{\Box\}$, then $W' = V$. Since $u = C'[V] \in \Delta_1 \setminus \Delta_0$, we have $W' \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$.

The other cases are trivial.

Definition 20. *For any $C[] \in \Theta_1$, for any $V \in \mathit{2}\langle \Delta \rangle$ such that $C[V] \in \Delta$, for any $V' \in \mathit{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$, for any $\rho \in \boldsymbol{Seq}_\Delta(C[V], V')$, we define $\mathcal{N}_\Delta(\rho, V, C[]) \in \mathit{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$ as follows: $\mathcal{N}_\Delta(\rho, V, C[])$ is the unique $W' \in \mathit{2}\langle \Delta \rangle$ such that $\mathcal{R}_\Delta(\rho, V, C[]) \in \boldsymbol{Seq}_\Delta(C[V], W')$.*

Remark 3. By Fact 10, we have $\mathcal{N}_\Delta(\rho, V, C[]) \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$.

The two following lemmas will be used in the proof of Proposition 3.

Lemma 9. *For any $C[] \in \Theta_1$, for any $V \in \mathit{2}\langle \Delta \rangle \setminus \mathit{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$ such that $C[V] \in \Delta \setminus \Delta_0$, for any $U \in \mathit{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle \setminus \{0\}$, for any $\rho = (U_1, \ldots, U_n) \in \boldsymbol{Seq}_{\Delta,K}(C[V], U)$, there exists $\rho' \in \boldsymbol{Seq}_{\Delta,K}(C[\mathcal{N}_\Delta(\rho, V, C[])], U)$ such that $\mathit{length}_\Delta(\rho') < \mathit{length}_\Delta(\rho)$.*

Proof: By induction on n.

We have $C[V] \in \Delta \setminus \Delta_0$ and $U \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle \setminus \{0\}$, hence $n \neq 0$ and $\mathit{length}_\Delta(\rho) > 0$.

If $C[] = \Box$, then $\mathcal{N}_\Delta(\rho, V, C[]) = U$ and we can set $\rho' = \mathit{id}_U$.

If $\rho = (C_0[u] \rhd_{C_0[]} C_0[U'];\ \rho_0)$, $u \rhd_\Box U'$ and $C[] \neq \Box$, then we distinguish between the following cases:

- if there exists $D[] \in \Theta_2$ such that $C_0 = D[V, \Box]$ and $C[] = D[\Box, u]$, then $\mathcal{N}_\Delta(\rho, V, C[]) = \mathcal{N}_\Delta(\rho_0, V, D[\Box, U'])$; by the induction hypothesis, there exists $\rho'_0 \in \mathbf{Seq}_{\Delta,K}(D[\mathcal{N}_\Delta(\rho_0, V, D[\Box, U']), U'], U)$ such that $\mathit{length}_\Delta(\rho'_0) < \mathit{length}_\Delta(\rho_0)$; we set $\rho' = (D[\mathcal{N}_\Delta(\rho_0, V, D[\Box, U']), u] \rhd_{D[\mathcal{N}_\Delta(\rho_0, V, D[\Box, U']), \Box]} D[\mathcal{N}_\Delta(\rho_0, V, D[\Box, U']), U'];\ \rho'_0)$;
- if $C_0[] = C[C'[]]$ with $C'[] \in \Theta_1$, then $\mathcal{N}_\Delta(\rho, V, C[]) = \mathcal{N}_\Delta(\rho_0, C'[U'], C[])$; by the induction hypothesis, there exists $\rho'_0 \in \mathbf{Seq}_{\Delta,K}(C[\mathcal{N}_\Delta(\rho_0, C'[U'], C[])], U)$ such that $\mathit{length}_\Delta(\rho'_0) < \mathit{length}_\Delta(\rho_0)$; we set $\rho' = \rho'_0$;
- if $u \rhd_\Delta U'$ and $C[] = C_0[C'[]]$ with $C'[] \in \Theta_1 \setminus \{\Box\}$, then $V \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$;
- if $U' = 0$ and $C[] = C_0[C'[]]$ with $C'[] \in \Theta_1 \setminus \{\Box\}$, then $\mathcal{N}_\Delta(\rho, V, C[]) = 0$; we have $\rho_0 \in \mathbf{Seq}_{\Delta,K}(C_0[0], U)$: there exists $\rho'_0 \in \mathbf{Seq}_{\Delta,K}(C_0[C'[0]], U)$ such that $\mathit{length}_\Delta(\rho'_0) \leq \mathit{length}_\Delta(\rho_0)$; now, if $V \notin \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$, then $\mathit{length}_\Delta(\rho_0) < \mathit{length}_\Delta(\rho)$.

Lemma 10. *For any $C[] \in \Theta_1 \setminus \{\Box\}$, for any $V \in \mathit{2}\langle \Delta \rangle \setminus \mathit{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$ such that $C[V] \in \Delta \setminus \Delta_0$, for any $U \in \mathit{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle \setminus \{0\}$, for any $\rho = U_1 \cdots U_n \in \boldsymbol{Seq}_{\Delta,K}(C[V], U)$, we have $\mathit{length}_\Delta(\mathcal{R}_\Delta(\rho, V, C[])) < \mathit{length}_\Delta(\rho)$.*

Proof: By induction on n.

We have $C[V] \in \Delta \setminus \Delta_0$ and $U \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle \setminus \{0\}$, hence $n \neq 0$ and $\mathit{length}_\Delta(\rho) > 0$.

So we have $\rho = (C_0[u] \rhd_{C_0[]} C_0[U'];\ \rho_0)$ with $\rho_0 \in \mathbf{Seq}_{\Delta,K}(U', U)$, $u \rhd_{\Box} U'$ and $C[] \neq \Box$; we distinguish between the following cases:

- there exists $D[] \in \Theta_2$ such that $C_0 = D[V, \Box]$ and $C[] = D[\Box, u]$: $\mathcal{R}_\Delta(\rho, V, C[]) = \mathcal{R}_\Delta(\rho_0, V, D[\Box, U'])$; by the induction hypothesis, we have $\mathit{length}_\Delta(\mathcal{R}_\Delta(\rho_0, V, D[\Box, U'])) < \mathit{length}_\Delta(\rho_0)$;
- if $C_0[] = C[C'[]]$ with $C'[] \in \Theta_1$ and $u \notin \mathbf{2}\langle \mathcal{T}_0(\mathcal{C})\rangle$, then $\mathcal{R}_\Delta(\rho, V, C[]) = (V \rhd_{C'[]} C'[U'];\ \mathcal{R}_\Delta(\rho_0, C'[U'], C[]))$; by the induction hypothesis, we have $\mathit{length}_\Delta(\mathcal{R}_\Delta(\rho_0, C'[U'], C[]))) < \mathit{length}_\Delta(\rho_0)$, hence $\mathit{length}_\Delta(\mathcal{R}_\Delta(\rho, V, C[])) = \mathit{length}_\Delta(\mathcal{R}_\Delta(\rho_0, C'[U'], C[]))) + 1 < \mathit{length}_\Delta(\rho_0) + 1 = \mathit{length}_\Delta(\rho)$;
- if $C_0[] = C[C'[]]$ with $C'[] \in \Theta_1$ and $u \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C})\rangle$, then $\mathcal{R}_\Delta(\rho, V, C[]) = (V \rhd_{C'[]} C'[U'];\ \mathcal{R}_\Delta(\rho_0, C'[U'], C[]))$; by the induction hypothesis, we have $\mathit{length}_\Delta(\mathcal{R}_\Delta(\rho_0, C'[U'], C[]))) < \mathit{length}_\Delta(\rho_0)$, hence $\mathit{length}_\Delta(\mathcal{R}_\Delta(\rho, V, C[])) = \mathit{length}_\Delta(\mathcal{R}_\Delta(\rho_0, C'[U'], C[]))) < \mathit{length}_\Delta(\rho_0) = \mathit{length}_\Delta(\rho)$;
- if $u \rhd_\Delta U'$ and $C[] = C_0[C'[]]$ with $C'[] \in \Theta_1 \setminus \{\Box\}$, then $V \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C})\rangle$;
- if $U' = 0$ and $C[] = C_0[C'[]]$ with $C'[] \in \Theta_1 \setminus \{\Box\}$, then $\mathcal{R}_\Delta(\rho, V, C[]) = id_0$: $\mathit{length}_\Delta(\mathcal{R}_\Delta(\rho, V, C[])) = 0$.

We will use a procedure called Equal that performs the following one: for any $c_0 \in \mathcal{T}_0(\mathcal{C})$, for any $u, v \in E$, after the execution of the procedure Equal(u, v), no value of any global variable changed; moreover if $[\![u]\!]_{\mathcal{V},c_0} = [\![v]\!]_{\mathcal{V},c_0}$, then the procedure Equal(u, v) returns true; otherwise, it returns false.

```
Redex(n) :=
Local aux;
if Def(L(n)) then for j := 1 to A do
                     if L(Succ(n)(A − j + 1)) ≠ ⊥ then aux := Redex(Succ(n)(A − j + 1));
                                                      if aux ≠ ⊥ then return aux;  fi;
                     fi;
                  od;
                  return n;
             else return ⊥;
```

Fig. 2. Procedure Redex

In the procedure Redex, the procedure Def is a procedure executed in constant time such that, for any $l \in \mathcal{D} \cup \mathcal{C} \cup \{x_1, \ldots, x_V\} \cup \{\bot\}$, if $l \in \mathcal{D}$, then the procedure returns true; otherwise it returns false.

The procedure Computation calls two procedures Redex and Subst, which have the following properties:

- For any $V \in \mathcal{V}(c_0)$, for any $n \in E$ such that $[\![n]\!]_{\mathcal{V},c_0,U} = V$, the execution time of the procedure Redex(n) is in $\mathcal{O}(|V|)$.
- Let $c_0 \in \mathcal{T}_0(\mathcal{C})$. Let $V \in \mathcal{V}(c_0) \setminus \mathcal{V}_0(c_0)$. Let $n \in E$ such that $[\![n]\!]_{\mathcal{V},c_0,U} = V$. Let $w_1, \ldots, w_K \in \mathcal{I}(c_0) \cup \{\bot\}$. Let m be the result of the procedure Redex(n). Let $\langle Y | D[] \rangle$ be the leftmost-innermost redex of V. Assume that $p = M$.

$\mathsf{Subst}(p, n, m, w_1, \ldots, w_K) :=$
if $n = m$ then $Comp(p)(1) := w_1;\ \ldots;\ Comp(p)(K) := w_K;\ L(p) := \perp;$
    else $L(p) := L(n);$
        $M := M + 1;\ Succ(p)(1) := M;\ \mathsf{Subst}(M, Succ(n)(1), m, w_1, \ldots, w_K);$
        $\ldots;$
        $M := M + 1;\ Succ(p)(A) := M;\ \mathsf{Subst}(M, Succ(n)(A), m, w_1, \ldots, w_K);$
fi;

Fig. 3. Procedure Subst

Then, after the execution of the procedure $\mathsf{Subst}(p, n, m, w_1, \ldots, w_K)$, we have $[\![p]\!]_{\mathcal{V}, c_0, U} = D[\sum_{\substack{j \in \{1, \ldots, K\} \\ w_j \neq \perp}} w_j]$. Moreover, apart from M, which increased, no value of any global variable changed. The execution time is in $\mathcal{O}(|V|)$.

$\mathsf{Computation}(n) :=$
Local $k, m, i, m', M_0, C, w_1, \ldots, w_K;$
if $L(n) = \perp$ then
        for $k := 1$ to K do
            if $Comp(n)(k) \neq \perp$ then
                $D := D + 1;$
                $Result(D) := Comp(n)(k);$
            fi;
        od;
    else $m := \mathsf{Redex}(n);$
        for $i := 1$ to $Inp\text{-}Max$ do
            if $\mathsf{Equal}(Inp(i), m)$ then $m' := i;$ fi;
        od;
        $C := 0;\ W(0) := \perp;$
        for j in $\mathcal{C}_0 \cup \{c \in \mathcal{T}_0(\mathcal{C});\ c \trianglelefteq c_0\}$ do
            if $Val(m')(j)$ then $C := C + 1;\ W(C) := j;$ fi;
        od;
        for $h_1, \ldots, h_K := 0$ to C do
            $w_1 := W(h_1);\ \ldots;\ w_K := W(h_K);$
            $M := M + 1;$
            $\mathsf{Subst}(M, n, m, w_1, \ldots, w_K);$
            $\mathsf{Computation}(M);$
        od;
fi;

Fig. 4. Procedure Computation

Lemma 11. *Let $c_0 \in \mathcal{T}_0(\mathcal{C})$. Let $Q \leq O$. Let $V \in \mathcal{V}_Q(c_0)$. Let $n \in E$ such that $[\![n]\!]_{\mathcal{V},c_0,U} = V$. Let $Y : \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0) \to \{\mathsf{true}, \mathsf{false}\}^{\mathcal{I}(c_0)}$ such that, for any $V' \in \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0)$, for any $c \in \mathcal{I}(c_0)$, $Y(V')(c) = \mathsf{true}$ if, and only if, $Val(inp(c_0)^{-1}(V'))(c) = \mathsf{true}$. After the execution of the procedure $\mathsf{Computation}(n)$, the following properties hold:*

- *apart from M and D, which increased, and apart from Result, no value of any global variable changed;*
- *the increasing of D is bound by $(Card(\mathcal{I}(c_0)) + 1)^{Q \cdot K} \cdot K$;*
- *for any $c \in \mathcal{I}(c_0)$, there exists $j \in \{1, \ldots, D\}$ such that $Result(j) = c$ if, and only if, there exists $V' \in \mathcal{V}_0(c_0)$ such that $c \in Supp(V')$ and $V \rhd_Y^* V'$.*

Proof: The proof is by induction on Q. Notice that $L(n) \neq \perp \Leftrightarrow V \notin \mathcal{V}_0(c_0)$.

The key-point to notice that the execution time of the procedure $\mathsf{Computation}(n)$ is polynomial in the size of c_0 is to notice that we can bound *Inp-Max* by $Q \cdot Card(\mathcal{I}(c_0))^{A \cdot (K+1)} = Q \cdot (S + |c_0|)^{A \cdot (K+1)}$.

B.1 The correctness of the algorithm follows from the two following propositions:

Proposition 3. *Let $c_0 \in \mathcal{T}_0(\mathcal{C})$. Let $m \in \mathbb{N}$, let $f \in \mathcal{D}_m$, let $C_1, \ldots, C_m \in \mathit{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$, let $c \in \mathcal{I}(c_0)$, let $o \in \{1, \ldots, Inp\text{-}Max\}$, let $\rho \in \boldsymbol{Seq}_{\Delta,K}(f(C_1, \ldots, C_m), c)$ such that $[\![o]\!]_{\mathcal{V},c_0} = f(C_1, \ldots, C_m)$. Then, at some moment of the execution of the algorithm, we have $Val(o)(c) = \mathsf{true}$.*

Proof: By induction on $length_\Delta(\rho)$. By Lemma 8, there exist $C'_1, \ldots, C'_m \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$, $U \in \mathbf{2}\langle \Delta \rangle$ and $\rho' \in \mathbf{Seq}_{\Delta,K}(U, c)$ such that $Supp(C'_1) \subseteq Supp(C_1)$, ..., $Supp(C'_m) \subseteq Supp(C_m)$, $f(C'_1, \ldots, C'_m) \rhd_\Delta U$ and $length_\Delta(\rho') < length_\Delta(\rho)$.

Notice that we cannot have $length_\Delta(\rho) = 0$.

Let q be the number of occurrences of elements of $\mathcal{D}$ in r. For any $j \in \{1, \ldots, q\}$, we define $U_j \in \mathcal{V}(c_0)$, $E_j[] \in \Theta_1$, $Y_j \in \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0)$, $G_j \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$ and $\rho_j \in \mathbf{Seq}_{\Delta,K}(U_j, c)$ by induction on j in such a way that $U_1, \ldots, U_{q-1} \notin \mathcal{V}_1(c_0)$ and $U_q \in \mathcal{V}_1(c_0)$:

- we set $U_1 = U$ and $\rho_1 = \rho'$; let $\langle Y_1 | E_1[] \rangle$ be the leftmost-innermost redex of U_1; we set $G_1 = \mathcal{N}_\Delta(\rho_1, Y_1, E_1[])$;
- for any $j \in \{1, \ldots, q-1\}$, we set $U_{j+1} = E_j[G_j]$; by Lemma 9, there exists $\rho_{j+1} \in \mathbf{Seq}_{\Delta,K}(U_{j+1}, c)$ such that $length_\Delta(\rho_{j+1}) < length_\Delta(\rho_j)$; let $\langle Y_{j+1} | E_{j+1}[] \rangle$ be the leftmost-innermost redex of U_{j+1}; we set $G_{j+1} = \mathcal{N}_\Delta(\rho_{j+1}, Y_{j+1}, E_{j+1}[])$.

Applying q times Lemma 10 and the induction hypothesis, we obtain: for any $j \in \{1, \ldots, q\}$, at some moment, for any $o \in \{1, \ldots, Inp\text{-}Max\}$ such that $[\![o]\!]_{\mathcal{V},c_0} = Y_j$, for any $c' \in Supp(G_j)$, we have $Val(o)(c') = \mathsf{true}$. But for any $j \in \{1, \ldots, q\}$, for any $o \in \{1, \ldots, Inp\text{-}Max\}$, for any $c' \in \mathcal{I}$, if at some moment, we have $Val(o)(c') = \mathsf{true}$, then, after this moment, we always have $Val(o)(c') = \mathsf{true}$. Therefore, at some moment, for any $j \in \{1, \ldots, q\}$, for any $o \in \{1, \ldots, Inp\text{-}Max\}$ such that $[\![o]\!]_{\mathcal{V},c_0} = Y_j$, for any $c' \in Supp(G_j)$, we have $Val(o)(c') = \mathsf{true}$.

Let $Y : \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0) \to \{\mathsf{true}, \mathsf{false}\}^{\mathcal{I}(c_0)}$ such that, for any $j \in \{1, \ldots, q\}$, for any $c' \in \mathit{Supp}(G_j)$, $Y(Y_j)(c') = \mathsf{true}$. We have $c_0 \in \mathit{Supp}(G_q)$ and $U_1 \rhd_Y U_2$, $\ldots$, $U_{q_1} \rhd_Y U_q$, $U_q \rhd_Y G_q$: we apply Lemma 11.

Proposition 4. *Let $c_0 \in \mathcal{T}_0(\mathcal{C})$. Let $m \in \mathbb{N}$, let $f \in \mathcal{D}_m$, let $C_1, \ldots, C_m \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$, let $c \in \mathcal{I}(c_0)$, let $o_0 \in \{1, \ldots, \mathit{Inp\text{-}Max}\}$ such that $[\![o_0]\!]_{\mathcal{V},c_0} = f(C_1, \ldots, C_m)$ and, at some moment of the execution of the algorithm, we have $\mathit{Val}(o_0)(c) =$ true. Then $f(C_1, \ldots, C_m) \rhd_K^* c$.*

Proof: Let N be the number of times that the line 4 is executed before that $\mathit{Val}(o)(c) = \mathsf{true}$ and let $i_0 \in \{1, \ldots, \mathit{Inp\text{-}Max}\}$ be the value of i when we execute the line 4 with $o = o_0$ and $\mathit{Result}(j) = c$ for the first time. The proof is by induction on (N, i_0) lexicographically ordered. Due to the procedure $\mathsf{Inf\text{-}Inp}$, there exist $C'_1, \ldots, C'_m \in \mathbf{2}\langle \mathcal{T}_0(\mathcal{C}) \rangle$ such that $\mathit{Card}(\mathit{Supp}(C'_1)), \ldots, \mathit{Card}(\mathit{Supp}(C'_m)) \leq K$, $\mathit{Supp}(C'_1) \subseteq \mathit{Supp}(C_1), \ldots, \mathit{Supp}(C'_m) \subseteq \mathit{Supp}(C_m)$ and $[\![i_0]\!]_{\mathcal{V},c_0} = f(C'_1, \ldots, C'_m)$.

Notice that we cannot have $N = 0$.

Let $Y : \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0) \to \{\mathsf{true}, \mathsf{false}\}^{\mathcal{I}(c_0)}$ such that, for any $V' \in \mathcal{V}_1(c_0) \setminus \mathcal{V}_0(c_0)$, for any $c' \in \mathcal{I}(c_0)$, $Y(V')(c') = \mathsf{true}$ if, and only if, $\mathit{Val}(\mathit{inp}^{-1}(V'))(c') = \mathsf{true}$. By Lemma 11, there exist $s \in \{1, \ldots, R\}$ and $C \in \mathcal{V}_0(c_0)$ such that $c \in \mathit{Supp}(C)$ and $[\![\mathit{Inst}(i_0)(s)]\!]_{\mathcal{V},c_0} \rhd_Y^* C$. Applying the induction hypothesis, we obtain $[\![\mathit{Inst}(i_0)(s)]\!]_{\mathcal{V},c_0} \rhd_K^* C$. Due to the procedure $\mathsf{Inst\text{-}Init}()$, we have $f(C'_1, \ldots, C'_m) \rhd_\Delta [\![\mathit{Inst}(i_0)(s)]\!]_{\mathcal{V},c_0}$. Due to the procedure $\mathsf{Inst\text{-}Init}()$, we have $\mathit{Card}(\mathit{Supp}(C_1)), \ldots, \mathit{Card}(\mathit{Supp}(C_m)) \leq K$. If follows that $f(C_1, \ldots, C_m) \rhd_K^* c$.