Space Complexity of LogicGuard Revisited *

David M. Cerna
Research Institute for Symbolic Computation (RISC)
Johannes Kepler University, Linz, Austria
David.Cerna@risc.uni-linz.ac.at

October 8, 2015

Abstract

We analyse the space complexity of specific formula structures constructed using the core language of the LogicGuard framework. Unlike previous analysis which focused on the local space complex around a single quantifier, in this work we fix the global structure of the formula enough to allow analysis as a whole. Though we end up with precise results for two very specific cases, we show that these cases are upper and lower bounds of arbitrary formulae with similar structure. We ignore cases which need infinite space as previous work deals with them sufficiently.

1 Introduction

The LogicGuard framework [3, 4] was designed to monitor network traffic by adding a monitor to a firewall which reports violations of security assertions. The language used to write the monitors is essentially a variant of predicate logic designed specifically to write monitors which observe network traffic. Rather then deciding the truth of the predicate logic monitors by providing a model, the model is instead generated real-time as packets are received by the firewall from the network. A specific semantics was constructed to deal with this real-time model building which is referred to as the *operational semantics* (See [2]). In [2], resource requirements of an arbitrary monitor was studied. By resource, we are referring to the amount of information which needs to be collected from the network in order to evaluate an instance of the given monitor.

In previous work concerned with space complexity of LogicGuard monitors [1], we developed bounds for space complexity based on the formula structure around a single quantifier. What we mean by formula structure around single quantifiers is that we can take a larger formula and abstract away everything but one quantifier, from this

*Supported by the Austrian Research Promotion Agency (FFG) in the frame of the BRIDGE program by the projects 832207 "LogicGuard" and 846003 "LogicGuard II".

quantifier and the abstraction of the global structure we can get a precise space bound. Recursively applying this space bound to a formula results in a space bound for the entire formula. We had to make a few assumptions for this method to work. One important assumption being that the monitor formula only has one free variable, namely the *stream variable*. The stream variable being the variable which is instantiated by the network stream. Having only one free variable allowed us to construct a *dominating formula* of a given formula, that is a formula whose quantifier bounds only contain the stream variable. The downside of this method of analysis is that it is highly inaccurate in that the bound is always an upper bound, but it is a few orders of magnitude off.

In this work, we ignore monitors which, for evaluation, need infinite space as well as monitors which have a constant in the quantifier bounds. In the latter case, the space complexity can be bounded by adding a stream variable to the constants in the quantifier bounds. In [1], we provide a method for differentiating between monitors which require infinite space and finite space. Instead of focusing on the local formula structure around quantifiers, we instead focus on a simple global formula structure. This simple formula structure is a chain of quantifiers. Essentially, we have an outer most quantifier which has only one quantifier in its matrix. The quantifier in its matrix also has a single quantifier in its matrix and so on. The last quantifier in the chain has a quantifier-free matrix. Even though this structure is extremely simple, it is easy to see that analysis of such quantifier chains can be generalized to anaylsis of formula which have multiple quantifier chains by breaking up the formulae into single chains and adding the results together.

The rest of this paper is as follows, we first provide the necessary background and definitions needed for the anaylsis. We then analyse quantifier chains where every quantifier has the same bounds, we refer to these chains as *uniform standard quantifier chains*. We used this basic analysis to analyse quantifier chains which have quantifiers with different bounds, but have monotonically increasing or decreasing upper bounds, namely *standard quantifier chains* and *inverse-standard quantifier chains*. Finally, we show that these two examples are the upper and lower bounds of space complexity of unordered quantifier chains, namely *top-free quantifier chains*. In every case we are assuming that we are dealing with a dominating formula.

2 Background

Unlike resource anaylsis [4], the space complexity analysis is concerned with the problem, given access to the necessary resources, how much memory will be used to evaluate the given monitor. The core language, the language we will be focusing on in this work, can be written as a context-free grammar as described in Fig. 1; we refer to this grammar as **F**ull **F**ormula **g**rammar (FF_g).

To give a brief outline of the connectives used in the above grammars, & *sequential and*, i.e. first evaluate the left sub-formula and then the right, $\wedge$ *parallel and*, i.e. evaluate both sub-formula simultaneously, $\sim$ *negation*, and $\forall x_{[a,b]}\varphi(x)$ *interval quantification*, i.e. $\varphi(x)$ is true for all $x \in [a, b]$, where a and b are constructed from the term language of

$$
\begin{aligned}
M &\implies mon(X) : F(X). \\
F &\implies @X|\ {\sim}F(X)|F(X)\ \&\ F(X)|F(X)\ \land\ F(X)| \\
&\quad\ \ \forall X_{[B,B]} : F\ . \\
B &\implies 0|\infty|X|B+N|B-N\ . \\
N &\implies k \in \mathbb{N}. \\
X &\implies x \in \mathbb{V}.
\end{aligned}
$$

Figure 1: We will refer to this grammar as FF_g *Full Formula grammar.*

$$
\begin{aligned}
M &\implies mon(X) : F(X). \\
F &\implies @X|\ {\sim}F(X)|F(X)\ \&\ F(X)|F(X)\ \land\ F(X)| \\
&\quad\ \ \forall X_{[B,B]} : F\ . \\
B &\implies X|B+N|\ . \\
N &\implies k \in \mathbb{N}. \\
X &\implies x \in \mathbb{V}.
\end{aligned}
$$

Figure 2: We will refer to this grammar as SF_g *Simplified Formula grammar.*

minimal arithmetic plus the symbol ∞. Interval quantification can also be written as $\forall x((a < x \land x < b) \rightarrow \varphi(x))$. Atoms are of the form $@x$ where x is some variable, of which when it is evaluated, is replaced with a numeral. Essentially, $@\overline{n}$, where $\overline{n}$ is a numeric constant, is a propositional variable which is assigned true or false depending on the truth value assigned to the position $\overline{n}$ in the external stream. Every formulae φ we will evaluate is encapsulated in a construct of the form $mon(x) : \varphi(x)$, where x is referred to as the *stream variable*. Consider the stream variable as an external quantifier over which all propositional symbols and internal quantifiers are evaluated. From now on we will only use x as the stream variable.

The rest of this paper will focus on the space complexity of the Simplified Formula Grammar SF_g (See Fig. 2). The main difference is quantifier bounds cannot contain infinity or a constant without a variable. The SF_g still allows bounds to contain variables other then the stream variable, but we will show how to remove these variables from any formula, though this removal effects the space complexity.

Rather then explaining the operational semantics of [4], we instead provide Ex. 1 which outlines the evaluation of a single quantifier.

Example 1. *Let us consider the following sentence constructed using SF_g:*

$$mon(x) : \forall y_{[x+2,x+4]} : (@y \& @x)$$

Evaluation of this monitor is performed as following by instantiating the monitor variable x with a value starting at zero:

$x = 0$	$x = 1$	$x = 2$
$\forall y_{[2,4]} : (@y\&@0)$	$\forall y_{[2,4]} : (@y\&@0)$ $\forall y_{[3,5]} : (@y\&@1)$	$\forall y_{[3,4]} : (@y\&@0)$ $\forall y_{[3,5]} : (@y\&@1)$ $\forall y_{[4,6]} : (@y\&@2)$

$x = 3$	$x = 4$	$x = 5$
$(@4\&@0)$ $\forall y_{[4,5]} : (@y\&@1)$ $\forall y_{[4,6]} : (@y\&@2)$ $\forall y_{[5,7]} : (@y\&@3)$	$(@5\&@1)$ $\forall y_{[4,6]} : (@y\&@2)$ $\forall y_{[5,7]} : (@y\&@3)$ $\forall y_{[6,8]} : (@y\&@4)$	$(@6\&@2)$ $\forall y_{[6,7]} : (@y\&@3)$ $\forall y_{[6,8]} : (@y\&@4)$ $\forall y_{[7,9]} : (@y\&@5)$

We assume that the truth value of every position on the stream is saved, thus, we can decide any propositional statement with positional variable indexed by 0 *up to the current value of x. For example when $x = 5$ we cannot remove* $(@6\&@2)$ *from memory because it relies on a future position. Notice that at no position did we consider the true value of any of the propositional variables. We do not care for the truth value of the positional sentences in this work, but rather we care to know if it is possible to evaluate them given the currently constructed model.*

Now we discuss the concept of a dominating formula. Under the assumption that the only free variable in a given monitor is the stream variable and that every bound variable is unique, we can transform a formula of the form:

$$mon(x) : \forall y_{[x+1,x+5]} : \left(\left(\forall z_{[y,x+3]} :^{\sim} @z \wedge \wedge @y \right) \& \right.$$
$$\left. \left(\forall w_{[x-1,y+2]} : \left({}^{\sim}@y \wedge \wedge \left(\forall m_{[y,w]} :^{\sim} @x\&@m \right) \right) \right) \right)$$

in the following formula:

$$mon(x) : \forall y_{[x+1,x+5]} : \left(\left(\forall z_{[x+1,x+3]} :^{\sim} @z \wedge \wedge @y \right) \& \right.$$
$$\left. \left(\forall w_{[x-1,x+7]} : \left({}^{\sim}@y \wedge \wedge \left(\forall m_{[x+1,x+7]} :^{\sim} @x\&@m \right) \right) \right) \right) .$$

The second formula is essentially a worst case senerio, in terms of memory usage, of the first formula.

Definition 1. *Given a sentence F constructed using SF_g, we say F is* regular *if the each quantifier has a distinct variable name for its bound variable.*

Definition 2 (Dominating Formula transformation)**.** *Given a regular sentence F constructed using SF_g we construct the* dominating formula of F, F_D, *using the following transformation:*

$$\begin{aligned}
D(mon(X) : F, \emptyset, \emptyset) &\Longrightarrow mon(x) : D(F, \{x \leftarrow x\}, \{x \leftarrow x\}) \\
D(F \& G, \sigma_l, \sigma_h) &\Longrightarrow D(F, \sigma_l, \sigma_h) \& D(G, \sigma_l, \sigma_h) \\
D(F \wedge G, \sigma_l, \sigma_h) &\Longrightarrow D(F, \sigma_l, \sigma_h) \wedge D(G, \sigma_l, \sigma_h) \\
D({}^{\sim}F, \sigma_l, \sigma_h) &\Longrightarrow {}^{\sim}D(F, \sigma_l, \sigma_h) \\
D(\forall Y_{[b_1,b_2]} : F, \sigma_l, \sigma_h) &\Longrightarrow \forall Y_{[h_L(b_1),h_H(b_2)]} : D(F, \sigma_l \{Y \leftarrow h_L(b_1)\}, \sigma_2 \{Y \leftarrow h_H(b_2)\}) \\
D(@X, \sigma_l, \sigma_h) &\Longrightarrow @X
\end{aligned}$$

where $h_L(b_1)$ is defined as,

$$h_L(b_1) = \min \{b_1\sigma_l, b_1\sigma_h\}$$

and where $h_H(b_2)$ is defined as,

$$h_H(b_2) = \max \{b_2\sigma_l, b_2\sigma_h\}$$

We assume form now on that we are working with the dominating form of any formula constructed from SF_g.

3 Quantifier Chain Vectors and Evaluation

In this section we introduce the abstractions that we will use for the rest of this paper.

Definition 3. *By $\varphi[\bullet]$ we mean φ is a formula with holes which are filled by the formula in the braces. By $\varphi[[\bullet]]$ we mean φ is a formula with a single hole which is filled by the formula in the braces.*

Example 2. *Let $\varphi[\bullet] = \neg((\bullet \wedge A) \& \bullet)$, then*

$$\varphi[\forall y\, [x+2, x+5]\, @y] = \neg(((\forall y\, [x+2, x+5]\, @y) \wedge A) \& (\forall y\, [x+2, x+5]\, @y))$$

If instead we has $\varphi[[\bullet]]$ we can only have one hole, i.e. $\neg(\bullet \wedge A)$.

Definition 4. *Let F be a dominating formula of a formula $F' \in SF_g$, with free variable x, of the form*

$$\varphi[[\forall x_1\, [x+a, x+b]\, \psi(x, x_1)]]$$

such that $\varphi[[\bullet]]$ is quantifier free, $a \leq b$, and $a, b \in \mathbb{N}$. Then $\langle a, b, w\rangle$, where $w \in \{0, 1\}$, is a quantifier tuple abstraction of F (QTA of F).

Example 3. *Let us consider the formula*

$$@x\&\forall x_1\,[x, x+5]\,(@x_1 \wedge \forall x_2\,[x+3, x+7]\,(@x_1\&@x_2))\,.$$

We can, using the formula with a hole $\varphi\,[[\bullet]] \equiv @x\&\bullet$*, write the formula as follow:*

$$\varphi\,[[\forall x_1\,[x, x+5]\,(@x_1 \wedge \forall x_2\,[x+3, x+7]\,(@x_1\&@x_2))]]\,.$$

At this point we can write the formula as a quantifier tuple abstraction $\langle 0, 5, w\rangle$ *where* $w \in \{0, 1\}$*. Notice that we ignore the matrix of the quantifier. We even allow the matrix to contain quantifiers which can have higher or lower upper and lower bounds than the outer-most formula. The only quantifier which matters is the one in the outermost position when concerning a QTA of a formula.*

To deal with the issues concerning the matrix, we define the following variation of the quantifier tuple abstractions.

Definition 5. *Let* F *be a dominating formula of a formula* $F' \in SF_g$*, with free variable* x*, of the form:*

$$\varphi\,[[\forall x_1\,[x+a, x+b]\,\psi(x, x_1)]]$$

such that $\varphi\,[[\bullet]]$ *is quantifier free,* $a \leq b$*, and* $a, b \in \mathbb{N}$*, and* $\psi(x, x_1)$ *has* m *quantifiers connected using propositional connectives, then* $\langle a, b, w\rangle^m$*, where* $w \in \{0, 1\}$*, is an* m quantifier tuple abstraction of F (m-QTA of F) .

Example 4. *Going back to Ex. 3, any QTA of the formula would be a* 1*-QTA. A* 0*-QTA would be for formulae whose outermost quantifier has a quantifier-free matrix. Concerning formula with nested quantifiers in the matrix of the outer most quantifier, such as*

$$@x\&\forall x_1\,[x, x+5]\,(@x_1 \wedge \forall x_2\,[x+3, x+7]\,(@x_1\&\forall x_3\,[x+3, x+7]\,(@x_3\&@x_2)))\,,$$

a QTA of such a formula would still be a 1*-QTA.*

Definition 6. *A* quantifier chain vector v *is an* n*-tuple, for* $n \in \mathbb{N}$*, of the form:*

$$v = \left[\langle a_1, b_1, x_1\rangle^1_1, \langle a_2, b_2, x_2\rangle^1_2, \cdots, \langle a_n, b_n, x_n\rangle^0_n\right].$$

where for each 1*-QTA* $\langle a_i, b_i, x_i\rangle^1_i$*, the QTA* $\langle a_i, b_i, x_i\rangle^j_{i+1}$*, for* $j \in \{0, 1\}$*, is an abstraction of the matrix of the outer-most quantifier in the formula abstracted by* $\langle a_i, b_i, x_i\rangle^1_i$*. The* empty vector*, i.e.* $n = 0$ *is represented by* $[]$ *and can be consider an abstraction of any quantifier-free formula.*

Example 5. *Let us consider a simple quantifier chain vector of length three constructed from the following formula:*

$$\forall x_1\,[x, x+5]\,(\forall x_2\,[x, x+6]\,(\forall x_3\,[x, x+7]\,(@x_1\&@x_2\&@x_3)))\,.$$

Considering the outermost quantifier we see that a 1-QTA is enough to abstract the formula and the resulting QTA is either $\langle 0,5,0\rangle^1$ *or* $\langle 0,5,1\rangle^1_i$. *The matrix of this quantifer is*

$$\forall x_2\,[x, x+6]\,(\forall x_3\,[x, x+7]\,(@x_1\&@x_2\&@x_3))\,.$$

To abstract the matrix, a 1-QTA again suffices except the upper bound is larger. We can abstract the matrix of the second formula now, however, it is quantifier free, so in this case we need a 0-QTA of the form $\langle 0,7,0\rangle^0$ *or* $\langle 0,7,1\rangle^0$. *Assuming that* 0 *was chosen for the third component of each QTA we can put them together as a quantifier chain vector*

$$\left[\langle 0,5,0\rangle^1, \langle 0,6,0\rangle^1, \langle 0,7,0\rangle^0\right]$$

The idea here is that given a quantifier chain vector, $\langle a_1, b_1, x_1\rangle^1_1$ (the subscript is the position in the quantifier chain vector) is a formula whose outer most quantifier has lower bound a_1 and upper bound b_1, and whose matrix is abstracted as a 1-QTA by $\langle a_2, b_2, x_2\rangle^1_2$.

We will refer to quantifier chain vectors as just vectors whenever it is clear from context. Also if one wants to know the length of a quantifier chain vector v, we represent this by $|v|$, as in the number of QTAs the vector contains. We assume that the formula quantifier chain vectors are derived from are dominating formulae [1]. We will refer to the i^{th} component of a quantifier chain vector v by $v(i)$, i.e. $v(i) = \langle a_i, b_i, x_i\rangle^k_i$. By $v(i,j)$ for $1 \le j \le 3$ we refer to the individual components of $v(i)$, i.e. $v(i,3) = 0$ or $v(i,3) = 1$.

Definition 7. *Let* v *be the quantifier chain vector*

$$\left[\langle a, b_1, 0\rangle^1_1, \langle a, b_2, 0\rangle^1_2, \cdots, \langle a, b_n, 0\rangle^0_n\right].$$

We call v top-free. *If there exists an ordering on the* b_i, *for* $1 \le i \le n$, *we call* v standard *when the ordering is monotonically increasing from left to right, and inverse-standard when the ordering is monotonically decreasing from left to right . If* b_i, *for* $1 \le i \le n$, *is equivalent We refer to* v *as* uniform standard.

Whenever it is not necessary for understanding, we will leave out the superscripts of the QTAs. The idea behind Def. 7 is that such configurations of the QTAs in a quantifier chain vector represent some canonical orderings. We will see later that these orderings actually result in an upper and lower bound in terms of Quantifier vector chains constructed with from a given set of QTAs.

Definition 8. *Let* v *be the quantifier chain vector*

$$[\langle a_1, b_1, x_1\rangle_1, \langle a_2, b_2, x_2\rangle_2, \cdots, \langle a_n, b_n, x_n\rangle_n]\,.$$

We call a quantifier chain vector v' *a* sub-quantifier chain vector of v *if there exists* $1 \le i \le j \le n$, *such that* $v' = [v(i), v(i+1), \cdots, v(j)]$. *We call* v' proper *if* $i \ne 1$ *or* $j \ne n$.

Definition 9. *Let v be a quantifier chain vector*

$$\left[\langle a_1, b_1, x_1\rangle_1^1, \langle a_2, b_2, x_2\rangle_2^1, \cdots, \langle a_n, b_n, x_n\rangle_n^0\right]$$

and w be a quantifier chain vector

$$\left[\left\langle a_1', b_1', x_1'\right\rangle_1^1, \left\langle a_2', b_2', x_2'\right\rangle_2^1, \cdots, \left\langle a_m', b_m', x_m'\right\rangle_m^0\right].$$

We define the vector $v' = v \otimes w$ as the product of v and w. *The vector v' has the following form:*

$$\left[\langle a_1, b_1, x_1\rangle_1^1, \langle a_2, b_2, x_2\rangle_2^1, \cdots, \langle a_n, b_n, x_n\rangle_n^1, \left\langle a_1', b_1', x_1'\right\rangle_{n+1}^1, \cdots, \left\langle a_m', b_m', x_m'\right\rangle_{m+n}^0\right]$$

An interesting issue concerning the product of two standard quantifier vector chains is how does this influence the underlying formula given that the last QTA of v was a 0-QTA. Essentially, we throwing away the original formula which were used to construct the vectors and replacing them with a more complex formula containing both. It is easy to see how one would go about this using the examples above.

Definition 10. *Let v be a quantifier chain vector*

$$\left[\langle a, b_1, x_1\rangle_1^1, \langle a, b_2, x_2\rangle_2^1, \cdots, \langle a, b_n, x_n\rangle_n^0\right].$$

A section *s of v is a sub-quantifier chain vector of v which is uniform, and such that there does not exists a sub-quantifier chain vector of v, w, whose product with s, $s \otimes w$, is also a sub-quantifier chain vector of v and is uniform, nor does there exists a w' which is a sub-quantifier chain vector of v whose product with s, $w' \otimes s$, is uniform.*

Example 6. *Given the quantifier vector chain*

$$\left[\langle 0, 5, 0\rangle^1, \langle 0, 6, 0\rangle^1, \langle 0, 6, 0\rangle^1, \langle 0, 7, 0\rangle^0\right]$$

The sections are as follows:

$$s_1 = \left[\langle 0, 5, 0\rangle^1\right]$$

$$s_2 = \left[\langle 0, 6, 0\rangle^1, \langle 0, 6, 0\rangle^1\right]$$

$$s_3 = \left[\langle 0, 7, 0\rangle^1\right]$$

Proposition 1. *Let v be a quantifier chain vector*

$$\left[\langle a, b_1, x_1\rangle_1^1, \langle a, b_2, x_2\rangle_2^1, \cdots, \langle a, b_n, x_n\rangle_n^0\right].$$

and $s_1, \cdots s_m$ be its sections, Then the vector $v_s = s_1 \otimes \cdots \otimes s_m$ such that $s_1(1,2) \leq s_2(1,2) \leq \cdots \leq s_m(1,2)$ is standard and the vector $v_i = s_1 \otimes \cdots \otimes s_m$ such that $s_1(1,2) \geq s_2(1,2) \geq \cdots \geq s_m(1,2)$ is inverse-standard.

Proof. Follows straight from definitions. □

Definition 11. *Let v be a quantifier chain vector of the form*

$$\left[\langle a_1, b_1, 1\rangle_1^1, \cdots, \langle a_m, b_m, 1\rangle_m^1, \langle a_{m+1}, b_{m+1}, 0\rangle_{m+1}^1, \cdots, \langle a_n, b_n, 0\rangle_n^0\right],$$

for $0 \le m \le n$. When $m = 0$, $v(i, 3) = 0$ for all $1 \le i \le n$. A split of v *constructs a set of vectors from v, $spt(v)$, under the following constraints:*

- *When $0 \le m < n$ and $a_{m+1} < b_{m+1}$, then $spt(v) = \{v_1, v_2\}$ where*

$$v_1 = \left[\langle a_1, b_1, 1\rangle_1^1, \cdots, \langle a_m, b_m, 1\rangle_m^1, \langle a_{m+1}, b_{m+1}, 1\rangle_{m+1}^1, \cdots, \langle a_n, b_n, 0\rangle_n^0\right],$$

$$v_2 = \left[\langle a_1, b_1, 1\rangle_1^1, \cdots, \langle a_m, b_m, 1\rangle_m^1, \langle a_{m+1} + 1, b_{m+1}, 0\rangle_{m+1}^1, \cdots, \langle a_n, b_n, 0\rangle_n^0\right].$$

- *When $0 \le m < n$ and $a_{m+1} = b_{m+1}$, then then $spt(v) = \{v_1\}$ where*

$$v_1 = \left[\langle a_1, b_1, 1\rangle_1^1, \cdots, \langle a_m, b_m, 1\rangle_m^1, \langle b_{m+1}, b_{m+1}, 1\rangle_{m+1}^1, \cdots, \langle a_n, b_n, 0\rangle_n^0\right],$$

- *When $m = n$, then $spt(v) \equiv \{[]\}$.*

Definition 12. *We define the* iterated split *of a quantifier chain vector v as follows:*

$$SI(n+1, v) \equiv \left(\bigcup_{w \in SI(n,v)} spt(w)\right) \cup SI(n, v)$$

$$SI(0, v) \equiv \{v\}$$

Proposition 2. *There exists $n \in \mathbb{N}$ such that $SI(n+1, v) \equiv SI(n, v)$.*

Proof. Trivial from the definition of split. □

Definition 13. *We define the* split closure *of a quantifier chain vector v as $SC_v = SI(n, v)$ where $SI(n+1, v) \equiv SI(n, v)$.*

Definition 14. *Given the* split closure *of a quantifier chain vector v, the* degenerate vector set *DV_v of SC_v is defined as follows:*

$$DV_v = \{w \in SC_v | spt(w) = []\}$$

Definition 15. *Given a quantifier chain vector v, the* the set of derived vectors *of v is $D_v = \{v\} \cup (SC_v \setminus DV_v)$*

To simplify our discussion of quantifier chain vectors, we will consider an abbreviation of a given quantifier vector chain v in terms of three sub quantifier vector chains v_1, v_h, and v_0.

Proposition 3. *Let v be a non-empty quantifier chain vector, for every vector $w \in D_v$, there exists three quantifier chain vectors $\mathbf{1}_w$, $\mathbf{h}_w$, and $\mathbf{0}_w$, such that $w = \mathbf{1}_w \otimes \mathbf{h}_w \otimes \mathbf{0}_w$, where $\mathbf{1}_w(i,3) = 1$, $1 \leq i \leq |\mathbf{1}_w|$, $\mathbf{h}_w(1,3) = 0$ and $|\mathbf{h}_w| = 1$, and $\mathbf{0}_w(i,3) = 0$, $1 \leq i \leq |\mathbf{0}_w|$. Both $\mathbf{0}_w$ and $\mathbf{1}_w$ can be the empty vector.*

Proposition 4. *Let v be a non-empty quantifier chain vector. For every vector $w \in D_v$, $\mathbf{h}_w$ is non-empty.*

Definition 16. *Given a vector $w \in D_v$ for some non-empty quantifier chain vector v, we define the set $e(s,w)$ where $s \in \mathbb{N}$ as follows:*

E1) *If $s < \mathbf{h}_w(1,1)$, then*

$$e(s,w) \equiv \{w\}$$

E2) *If $\mathbf{h}_w(1,1) \leq s < \mathbf{h}_w(1,2)$ and $\mathbf{0}_w \neq []$, then*

$$e(s,w) \equiv \left(\bigcup_{i=\mathbf{h}_w(1,1)}^{s} e(s, \mathbf{1}_w \otimes v_i \otimes \mathbf{0}_w) \right) \cup \{\mathbf{1}_w \otimes w_{s+1} \otimes \mathbf{0}_w\},$$

where $v_i = [\langle i, \mathbf{h}_w(1,2), 1\rangle]$ and $w_{s+1} = [\langle s+1, \mathbf{h}_w(1,2), 0\rangle]$.

E3) *If $\mathbf{h}_w(1,2) \leq s$ and $\mathbf{0}_w \neq []$,then*

$$e(s,w) \equiv \left(\bigcup_{i=\mathbf{h}_w(1,1)}^{\mathbf{h}_w(1,2)} e(s, \mathbf{1}_w \otimes v_i \otimes \mathbf{0}_w) \right),$$

where $v_i = [\langle i, \mathbf{h}_w(1,2), 1\rangle]$.

E4) *If $\mathbf{h}_w(1,1) \leq s < \mathbf{h}_w(1,2)$ and $\mathbf{0}_w = []$, then*

$$e(s,w) \equiv \{\mathbf{1}_w \otimes w_{s+1} \otimes \mathbf{0}_w\},$$

where $w_{s+1} = [\langle s+1, \mathbf{h}_w(1,2), 0\rangle]$.

E5) *If $\mathbf{h}_w(1,2) \leq s$ and $\mathbf{0}_w = []$, then*

$$e(s,w) \equiv \emptyset$$

Proposition 5. *Given a vector $w \in D_v$ for some non-empty quantifier chain vector v, then $e(m,w) \subseteq D_v$ for $0 \leq m$.*

Definition 17. *Given a vector $w \in D_v$ for some non-empty quantifier chain vector v. We can define the function $E(m,w)$, for $0 \leq v(1,1) \leq m$ recursively as follows,*

$$E(m,w) := \begin{cases} e(m, E(m-1,w)) & v(1,1) < m \\ \\ e(v(1,1), m) & \text{otherwise} \end{cases}$$

Proposition 6. *Given a vector $w \in D_v$ for some non-empty quantifier chain vector v, then $E(m, w) \subseteq D_v$ for $0 \leq m$.*

Proposition 7. *Given a vector $w \in D_v$ for some non-empty quantifier chain vector v, then $E_v(m, w) \equiv E_v(m, \mathbf{h}_w \otimes \mathbf{0}_w)$ for $0 \leq m$.*

Proof. It is obvious from Def. 16. □

Definition 18. *Given a set of vectors $V \subseteq D_v$ for some non-empty quantifier chain vector v. We define $e(s, V)$, for $0 \leq s$, as follows:*

$$e(s, V) \equiv \bigcup_{w \in V} e(s, w)$$

Definition 19. *Given a set of vectors $V \subseteq D_v$ for some non-empty quantifier chain vector v. We define $\{V\}_1^i$ for all $1 \leq i \leq |v|$ as follows:*

$$\{V\}_1^i = \{w \in V | \mathbf{h}_w = w(i)\}\,.$$

The size of $\{V\}_1^i$ is defined as

$$|V|_1^i = |\, \{V\}_1^i \,|$$

Proposition 8. *Given a set of vectors $V \subseteq D_v$ for some non-empty quantifier chain vector v. For all $v(1, 1) \leq m \leq v(n, 2)$ the following holds:*

$$E(m, V) \equiv \bigcup_{i=1}^{n} E(m, \{V\}_1^i) \equiv \bigcup_{i=1}^{n} \bigcup_{w \in \{V\}_1^i} E(m, w) \tag{1a}$$

$$|E(m, V)| = \sum_{i=1}^{n} |E(m, \{V\}_1^i)| = \sum_{i=1}^{n} \sum_{w \in \{V\}_1^i} |E(m, w)| \tag{1b}$$

$$|E(m, V)|_1^i = \sum_{j=1}^{n} |E(m, \{V\}_1^j)|_1^i = \sum_{j=1}^{n} \sum_{w \in \{V\}_1^j} |E(m, w)|_1^i \tag{1c}$$

$$|E(m, V)|_1^i = |\bigcup_{j=1}^{i} E(m, \{V\}_1^j)|_1^i \tag{1d}$$

4 Analysis of Uniform Standard Quantifier Chain Vectors

In this section we compute the size of $E(i, v)$ where $v(1, 1) \leq i \leq v(1, 2)$ and v is a uniform standard quantifier chain vector. Another way of considering v is that it only has one section, thus, it is a quite simple case. However, it will play the role as a foundation for the more complex cases considered in later sections.

Theorem 1. *Given a non-empty uniform standard quantifier chain vector v, then*

$$|e(v(1,1), v)| = n$$

Proof. Let us assume that $|v| = 1$, then $v = [] \otimes \mathbf{h}_v \otimes []$. From now on we refer to $v(i,1)$ as a and $v(i,2)$ as b, for $1 \leq i \leq n$, being that the vector is uniform. When evaluating $e(a,v)$ using Def. 16, we apply **E4** based on our assumptions. We get the following derivation:

$$e(a,v) \equiv \{[\langle a+1, b, 0\rangle_1]\}$$

The resulting set has size one and thus the theorem holds for the basecase. Let us now assume the theorem holds for all $m \leq n$ and show that the theorem holds for n+1. By the definition of uniform standard quantifier vector chains $v = [] \otimes [\langle a, b, 0\rangle_1] \otimes \mathbf{0}_v$. Thus, when evaluating $e(a,v)$ using Def. 16 we apply **E2**. This application results in the following computation:

$$e(a,v) \equiv e(a, [\langle a, b, 1\rangle] \otimes [\langle a, b, 0\rangle] \otimes \mathbf{0}'_v) \cup \{[\langle a+1, b, 0\rangle] \otimes \mathbf{0}_v\}$$

Where $\mathbf{0}_v \equiv [\langle a, b, 0\rangle] \otimes \mathbf{0}'_v$ This implies that

$$|e(a,v)| = |e(a, \mathbf{1}_v \otimes [\langle a, b, 0\rangle] \otimes \mathbf{0}'_v)| + 1$$

Now considering the vector $[\langle a, b, 0\rangle] \otimes \mathbf{0}'_v$, which we will refer to as v', it is essentially the sub quantifier chain vector of v which has the component $\mathbf{1}_v$ removed. The length of v' is n, and thus by the induction hypothesis, $|e(a, v')| = n$. Thus, by induction, we have proven that $|e(a,v)| = n+1$ when v is of length $n+1$. □

Corollary 1. *Given a non-empty uniform standard quantifier chain vector v , then*

$$|e(a,v)|_1^i = 1$$

for $1 \leq i \leq n$.

Theorem 2. *Given a non-empty uniform standard quantifier chain vector v, then for $1 \leq i \leq |v|$ and $v(1,1) \leq m < v(1,2)$,*

$$|E(m,v)|_1^i = ((m - v(1,1)) + 1)^{i-1}$$

Proof. From now on in this proof we will refer to $v(1,1)$ as a and $v(1,2)$ as b and $|v|$ as n. To prove this theorem we perform an induction over the lexicographical ordering of the ordered pairs (m,i) for $a \leq m < b$ and $1 \leq i \leq n$. The position m in the ordered pair (m,i) refers to the first position in the function $E(m,v)$, i.e. the recursion depth. The position i in the ordered pair (m,i) refers to the position of $\mathbf{h}_w$ in a given vector $w \in D_v$. Fig. 3 provides an outline of the inductive argument used to prove the theorem.

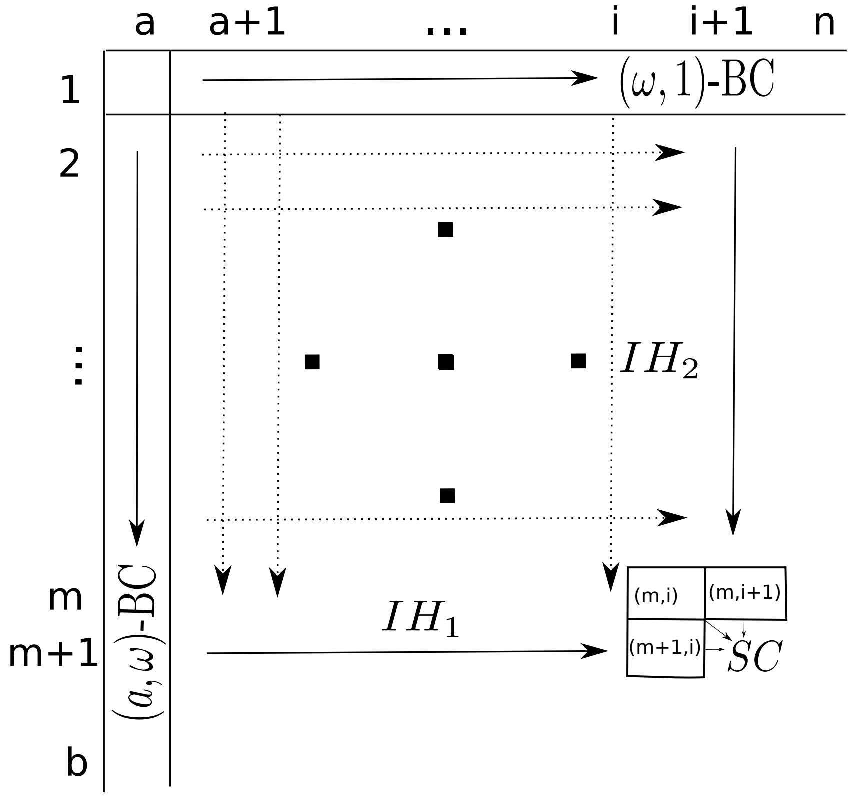

Figure 3: By SC in the diagram we are referring to the final stepcase $(m+1, i+1)$ which completes the inductive argument.

4.0.1 Fixed Recursion Depth Induction

From Thm. 1 and Cor. 1 we can gather that

$$|E(a, v)|_1^i = 1^{i-1}.$$

These equations are represented in our lexicographical ordering as (a, i) for $1 \leq i \leq n$. To fit these results to the structure of the current theorem, let us consider the argument leading up to $|E(a, v)|_1^1 = 1$ as the basecase for an induction. We will refer to it as $(a, 1)$-BC. The induction hypothesis based on $(a, 1)$-BC assumes that $|E(a, v)|_1^j = 1$ for $1 \leq j \leq i < n$, and we show that the theorem also holds for $i + 1$. We will refer to this induction hypothesis as $(a, \overrightarrow{\omega})$-IH and the theorem that follows from it as (a, ω)-BC, essentially Cor. 1. We refer the reader to the proof of Thm. 1 for more details concerning the argument.

4.0.2 Fixed Quantifier Chain Vector Position Induction

Now we consider induction over m, the recursion depth. We need to use $(a, 1)$-BC to construct our induction hypothesis $(\overrightarrow{\omega}, 1)$-IH, of which assumes that the theorem holds for $(r,1)$, when $a \le r \le m < b-1$, and we show that it also holds for $m+1$. Written in a less obfuscated form, we are showing that

$$|E_v(m+1,v)|_1^1 = (((m+1)-a)+1)^0 = 1.$$

We are assuming that

$$|E(r,v)|_1^1 = ((r-a)+1)^0 = 1$$

holds for $a \le r \le m < b-1$.

The resulting theorem will be referred to as $(\omega,1)$-BC. The argumentation is as follows, first consider Def. 17 which states that the following rewriting is valid:

$$E(m+1,v) = e(m+1,E(m,v)).$$

We know, by $(\overrightarrow{\omega},1)$-IH, that $|E(m,v)|_1^1 = 1$. Also, we know that the following holds by Def. 18:

$$e(m+1,a,E(m,v)) \equiv \bigcup_{w\in E(m,v)} e(m+1,w). \tag{2}$$

Eq. 2 is almost in the right form, we just need to consider the sets $\{E(m,v)\}_1^1$ and $E(m,v)\setminus\{E(m,v)\}_1^1$ instead of $E(m,v)$ in the union.

$$e(m+1,E(m,v)) \equiv \bigcup_{j=2}^{n} e(m+1,\{E(m,v)\}_1^j)) \cup \bigcup_{w\in\{E(m,v)\}_1^1} e(m+1,w) \tag{3}$$

Let us consider the cardinality of the two sets on the right side of Eq. 3, more specifically the cardinality $|\cdot|_1^1$ of Def. 19. Consider only the right-hand most set of Eq. 3 it is easy to see that the following holds

$$\left|\bigcup_{w\in\{E(m,v)\}_1^1} e(m+1,w)\right|_1^1 = 1$$

By Def. 16 and $(\overrightarrow{\omega},1)$-IH. Now consider the other set on the right-hand side of Eq. 3, we get the following:

$$\left|\bigcup_{j=2}^{n} e(m+1,\{E(m,v)\}_1^j))\right|_1^1 = \left|e(m+1,\bigcup_{j=2}^{n}\{E(m,v)\}_1^j))\right|_1^1 = 0. \tag{4}$$

Eq. 4 holds because every vector $w \in \bigcup_{j=2}^{n}\{E(m,v)\}_1^j)$ is of the form w $= \mathbf{1}_w \otimes \mathbf{h}_w \otimes \mathbf{0}_w$. By Prop. 7, application of Def. 16 on any of these vectors $w' = \mathbf{1}_{w'} \otimes \mathbf{h}_{w'} \otimes \mathbf{0}_{w'}$ is equivalent to the application of Def. 16 to the vector $\mathbf{h}_{w'} \otimes \mathbf{0}_{w'}$, essentially ignoring w_1, see Prop. 7. Thus, for every such w, $|e(m+1,w)|_1^1 = 0$, and it follows that Eq. 4 holds. Thus,

$$|E(m+1,v)|_1^1 = |e(m+1,a,E(m,v))|_1^1 = 1 \tag{5}$$

holds and we have proven the theorem by induction. This argument suffices to show that $(\omega,1)$-BC holds.

4.0.3 Induction on recursion depth and position in v

Using both (a, ω)-BC and $(\omega, 1)$-BC We now construct our final induction hypothesis. We assume that the theorem holds for $(m+1, j)$, $(r, i+1)$, and $(r, 1)$, where $a \leq r \leq m < b-1$ and $1 \leq j \leq i < n$. We refer to the first assumption, $(m+1, j)$, as IH_1 and the two other assumptions, $(r, i+1)$ and $(r, 1)$, as IH_2. Given these assumptions we now show that the theorem holds for $(m+1, i+1)$. Descriptively, we are trying to show that the following holds under the assumption of IH_1 and IH_2:

$$|E_v(m+1, v)|_1^{i+1} = (((m+1) - a) + 1)^i \tag{6}$$

when $a \leq m < b-1$ and $1 \leq i < n$. The first assumption IH_1 states

$$|E(m+1, v)|_1^j = (((m+1) - a) + 1)^{j-1} \tag{7}$$

when $a \leq m < b-1$ and $1 \leq j \leq i < n$. The second assumption IH_2 states

$$|E(r, v)|_1^i = ((r - a) + 1)^{i-1} \tag{8}$$

when $a \leq r \leq m < b-1$ and $1 \leq i \leq n$.

Now we show how one can rewrite the left-hand side of Eq. 6 into parts which are justified by either IH_1 or IH_2. We start as we did in the proof of $(\omega, 1)$-BC by unrolling the $E(m+1, v)$ using Def. 17.

$$E(m+1, v) \equiv e(m+1, E(m, v)) \tag{9}$$

Using Prop. 1c & 1d we can justify the following derivation:

$$|E(m+1, v)|_1^{i+1} = |e(m+1, E(m, v))|_1^{i+1} =$$

$$|e(m+1, \textstyle\bigcup_{k=1}^{i} \{E(m, v)\}_1^k) \cup \bigcup_{w=i+1}^{n} e(m+1, \{E(m, v)\}_1^w)|_1^{i+1} = \tag{10a}$$

$$|e(m+1, \textstyle\bigcup_{k=1}^{i} \{E(m, v)\}_1^k) \cup e(m+1, \{E(m, v)\}_1^{i+1})|_1^{i+1} = \tag{10b}$$

$$|e(m+1, \bigcup_{k=1}^{i} \{E(m, v)\}_1^k)|_1^{i+1} + |e(m+1, \{E(m, v)\}_1^{i+1})|_1^{i+1} = \tag{10c}$$

$$|e(m+1, \bigcup_{k=1}^{i} \{E(m, v)\}_1^k)|_1^{i+1} + |\bigcup_{w \in \{E(m,v)\}_1^{i+1}} e(m+1, w)|_1^{i+1} = \tag{10d}$$

$$|e(m+1, \bigcup_{k=1}^{i} \{E(m, v)\}_1^k)|_1^{i+1} + \sum_{w \in \{E(m,v)\}_1^{i+1}} |e(m+1, w)|_1^{i+1} \tag{10e}$$

The transition from Eq. 10c to Eq. 10d is justified by Def. 18, and the final transition from Eq. 10d to Eq. 10e is again justified by Prop. 1c. We now concern ourselves with Eq. 10e, more specifically, the summation on the right-hand side of the addition, i.e.

$$\sum_{w \in \{E(m,v)\}_1^{i+1}} |e(m+1, w)|_1^{i+1}. \tag{11}$$

By IH_2, we know that $|E(m,v)|_1^{i+1} = ((m-a)+1)^{i+1}$, thus, if the evaluation of $|e(m+1,w_1)|_1^{i+1} = |e(m+1,w_2)|_1^{i+1}$ for any $w_1, w_2 \in \{E(m,v)\}_1^{i+1}$, then Eq. 11 can be rewritten as follows:

$$((m-a)+1)^{i+1} \cdot |e(m+1,w')|_1^{i+1} \tag{12}$$

for some arbitrary $w' \in \{E(m,v)\}_1^{i+1}$. Incidental,by this property does hold as we shall show.

By Def. 16, there are two ways $e(m+1,w)$ can evaluate given a vector $w = \mathbf{1}_w \otimes \mathbf{h}_w \otimes \mathbf{0}_w$ such that $\mathbf{h}_w = \left[\langle m+1,b,0\rangle_i^1\right]$. Note that $\mathbf{h}_w(1,1) = m+1$ because every member in the set $\{E_v(m,v)\}_1^{i+1}$ was derived from prior evaluations. The following two equations are the possible evaluations (we leave out $\mathbf{1}_w$ by Prop. 7):

$$e(m+1,w) = e(m+1,[\langle a,b,0\rangle] \otimes \mathbf{0}'_w) \cup \{[\langle m+2,b,0\rangle] \otimes \mathbf{0}_w\} \tag{13a}$$

$$e(m+1,w) = \{[\langle m+2,b,0\rangle]\} \tag{13b}$$

where $\mathbf{0}_w = \mathbf{h}_w \otimes \mathbf{0}'_w$.In the case of Eq.13b, $\mathbf{0}_w = []$, thus obviously $|e(m+1,w)|_1^{i+1} = 1$. In the case of Eq.13a we just need to consider the same argument from Subsec. 4.0.2, that is that $|e(m+1,[\langle a,b,0\rangle] \otimes w'_0)|_1^{i+1} = 0$. This implies that only one vector is generated as in Eq.13b. Thus what we can conclude is $|e(m+1,w)|_1^{i+1} = 1$ for any $w \in \{E(m,v)\}_1^{i+1}$, and by Eq. 11 & 12 the following equality holds:

$$\sum_{w \in \{E(m,v)\}_1^{i+1}} |e(m+1,w)|_1^{i+1} = ((m-a)+1)^{i+1} \cdot |e(m+1,w')|_1^{i+1} = ((m-a)+1)^i. \tag{14}$$

Getting back to Eq. 10e, we can now write it as follows:

$$|e(m+1, \textstyle\bigcup_{k=1}^{i} \{E(m,v)\}_1^k)|_1^{i+1} + ((m-a)+1)^i \tag{15}$$

To compute,

$$|e(m+1, \bigcup_{k=1}^{i} \{E(m,v)\}_1^k)|_1^{i+1}$$

we need to consider vectors of the form $w = \mathbf{1}_w \otimes [\langle m+1,b,0\rangle] \otimes \mathbf{0}_w$, where $|\mathbf{1}_w| = i-1$ and how they generate vectors $w' = \mathbf{1}'_w \otimes [\langle m+2,b,0\rangle] \otimes \mathbf{0}'_w$, where $|\mathbf{1}'_w| = i$, using Def. 16. By IH_1 and Def. 16 we know the following:

$$|e(m+1, \bigcup_{k=1}^{i} \{E(m,v)\}_1^k)|_1^i = (((m+1)-a)+1)^{i-1}. \tag{16}$$

By IH_2 and Def. 16 it can also be derived that,

$$|\bigcup_{k=1}^{i} \{E(m,v)\}_1^k|_1^i = ((m-a)+1)^{i-1}. \tag{17}$$

We now concern ourselves with the set counterpart of the left-hand side of Eq. 17, namely the following:

$$\left\{ \bigcup_{k=1}^{i} \left\{ E_(m,v) \right\}_1^k \right\}_1^i .$$

Let us now consider $D = \bigcup_{k=1}^{i} \left\{ E(m,v) \right\}_1^k$. We are interested in the application of Def. 16 with recursion depth $m+1$ to D, more specifically $|e(m+1,D)|_1^i$. Using Prop.1a & 1c, we can derive the following:

$$|e(m+1,D)|_1^i = \left|\bigcup_{w \in D} e(m+1,w)\right|_1^i = \sum_{w \in D} |e(m+1,w)|_1^i . \tag{18}$$

I AM HERE WITH THE EDITING Concerning the right-most formula of Eq. 18, each w is of the form:

$$w = \mathbf{1}_w \otimes [\langle m+1, b, 0\rangle] \otimes \mathbf{0}_w,$$

where $|\mathbf{1}_w| = i$. We need to assume that $i < n$ at this point, because if $i = n$, then the results presented here can not be used to prove the case $i+1$. The reason we use $m+1$ in w_h instead of a is that each of these vectors has already been evaluated during the application of Def. 16 to a vector at a recursion depth m or less. Now we consider the evaluation of $e(m+1,w)$:

$$e(m+1,w) = e(m+1, \mathbf{1}'_w \otimes \mathbf{h}'_w \otimes \mathbf{0}'_w) \cup \{\mathbf{1}_w \otimes [\langle m+2, b, 0\rangle] \otimes \mathbf{0}_w\} \tag{19}$$

Where $\mathbf{1}'_w = \mathbf{1}_w \otimes [\langle m+1, b, 1\rangle]$ and $\mathbf{h}'_w = [\langle a, b, b\rangle]$ As in Eq. 13a,

$$\left\{ e(m+1, \mathbf{1}'_w \otimes \mathbf{h}'_w \otimes \mathbf{0}'_w) \right\}_1^i$$

is empty thus, $|e(m+1,w)|_1^i = 1$. Being that Eq. 19 is the only case we have to deal with, we can now go back to Eq. 18 and derive the following result:

$$|e(m+1,D)|_1^i = \sum_{w \in D} |e(m+1,w)|_1^i = \sum_{w \in D} 1 = ((m-a)+1)^{i-1} \tag{20}$$

The right most part Eq. 20 is derived from the size of D, of which is provided in Eq. 17. Getting back to Eq. 19 the derivation also provides us information about $|e(m+1,w)|_1^{i+1}$, essentially that $|e(m+1,w)|_1^{i+1} = 1$ as well. To see this, let us consider $e(m+1,w')$ ($w' = \mathbf{1}'_{w'} \otimes [\langle a, b, 0\rangle] \otimes \mathbf{0}'_{w'}$), where $\mathbf{h}'_{w'} = [\langle a, b, 0\rangle]$ and $|\mathbf{1}_{w'}| = i$. This time with brackets for $i+1$, i.e. $\{e(m+1,w')\}_1^{i+1}$. Now consider the application of Def. 16 to w' for recursion depth $m+1$, there are two cases as there were for Eq. 13a & Eq. 13b

$$e(m+1,w') = \bigcup_{k=a}^{m+1} e(m+1, \mathbf{1}''_{w'}(k) \otimes \mathbf{h}''_{w'} \otimes \mathbf{0}''_{w'}) \cup \left\{ \mathbf{1}'_{w'} \otimes \mathbf{h}'_{w'} \otimes \mathbf{0}'_{w'} \right\} \tag{21}$$

$$e(m+1,w') = \left\{ \mathbf{1}'_{w'} \otimes \mathbf{h}'_{w'} \otimes [] \right\} \tag{22}$$

where $\mathbf{1}''_{w'}(k) = \mathbf{1}'_{w'} \otimes [\langle k, b, 1\rangle]$ and $\mathbf{h}''_{w'} = [\langle a, b, 0\rangle]$. Obviously in the case of Eq. 22, $|e(m+1,w')|_1^{i+1} = 1$, even though in the case Eq. 21 we have to consider $m+1-a$ more function applications, it is quite obvious from prior argumentation that their $(i+1)^{\text{th}}$ cardinality is zero. Thus, Eq. 20 can be written for the $(i+1)$ case as well,

$$|e(m+1,D)|_1^{i+1} = \sum_{w\in D} |e(m+1,w)|_1^{i+1} = \sum_{w\in D} 1 = ((m-a)+1)^{i-1}. \tag{23}$$

Now we can get back to Eq. 16. Base on the work above and Prop. 8, we can do the following:

$$|e(m+1, \bigcup_{k=1}^{i} \{E(m,v)\}_1^k)|_1^i = \tag{24a}$$

$$|e(m+1, \bigcup_{k=1}^{i-1} \{E(m,v)\}_1^k) \cup e(m+1, \{E_v(m,v)\}_1^i)|_1^i = \tag{24b}$$

$$|e(m+1, \bigcup_{k=1}^{i-1} \{E(m,v)\}_1^k)|_1^i + |e(m+1, \{E(m,v)\}_1^i)|_1^i = \tag{24c}$$

$$|e(m+1, \bigcup_{k=1}^{i-1} \{E(m,v)\}_1^k)|_1^i + |e(m+1, \{E(m,v)\}_1^i)|_1^{i+1} = \tag{24d}$$

$$|e(m+1, \bigcup_{k=1}^{i-1} \{E(m,v)\}_1^k)|_1^i + ((m-a)+1)^{i-1} \tag{24e}$$

We know the value of Eq. 24a, thus we can deduce the value of

$$|e(m+1, \bigcup_{k=1}^{i-1} \{E(m,v)\}_1^k)|_1^i$$

to be Eq. 24a minus $((m-a)+1)^{i-1}$, or more precisely,

$$(((m+1)-a)+1)^{i-1} - ((m-a)+1)^{i-1} = \sum_{j=1}^{i-1} \binom{i-1}{j} \cdot ((m-a)+1)^{(i-1)-j}.$$

The equality

$$|e(m+1, \bigcup_{k=1}^{i-1} \{E(m,v)\}_1^k)|_1^i = \sum_{j=1}^{i-1} \binom{i-1}{j} \cdot ((m-a)+1)^{(i-1)-j} \tag{25}$$

is enough to show that Eq. 6 holds if we derive the following:

$$|e(m+1, \bigcup_{k=1}^{i-1} \{E(m,v)\}_1^k)|_1^{i+1}.$$

Essentially, what we are missing is how to go from knowing the size of

$$|e(m+1, \bigcup_{k=1}^{i-1} \{E(m,v)\}_1^k)|_1^i$$

to knowing the size of

$$|e(m+1, \bigcup_{k=1}^{i-1} \{E(m,v)\}_1^k)|_1^{i+1}.$$

An important fact concerning $\bigcup_{k=1}^{i-1} \{E(m,v)\}_1^k$ is that $|\bigcup_{k=1}^{i-1} \{E(m,v)\}_1^k|_1^i = 0$. This means that any vector $w \in \left\{e(m+1, \bigcup_{k=1}^{i-1} \{E(m,v)\}_1^k)\right\}_1^{i+1}$ was generated by $m+1$ applications of Def. 16. In other words, when Def. 16 is applied to one of these vectors w the head position will always be of the form $\mathbf{h}_w = \langle a, b, 0\rangle$. Thus, when we apply Def. 16 to $w = \mathbf{1}_w \otimes [\langle a,b,0\rangle] \otimes \mathbf{0}_w$, where $|\mathbf{1}_w| = i-1$ we will get the evaluation:

$$e(m+1, w') \equiv \left(\bigcup_{j=a}^{m+1} e(m+1, \mathbf{1}'_w(j) \otimes [\langle a,b,0\rangle] \otimes \mathbf{0}'_w)\right) \cup \{\mathbf{1}_w \otimes [\langle m+2, b, 0\rangle] \otimes \mathbf{0}_w\}$$

where $\mathbf{1}'_w(j) = \mathbf{1}_w \otimes [\langle j,b,0\rangle]$. Again we see that $|e(m+1,w)|_1^i = 1$, and,

$$\left|\left(\bigcup_{j=a}^{m+1} e(m+1, \mathbf{1}'_w(j) \otimes [\langle a,b,0\rangle] \otimes \mathbf{0}'_w)\right)\right|_1^i = 0,$$

but importantly ,

$$\left|\left(\bigcup_{j=a}^{m+1} e(m+1, \mathbf{1}'_w(j) \otimes [\langle a,b,0\rangle] \otimes \mathbf{0}'_w)\right)\right|_1^{i+1} = ((m+1)-a)+1. \tag{26}$$

Now we must put together the results of Eq. 14, 23,25, and26 to get the value of

$$|E(m+1,a,v)|_1^{i+1}.$$

The resulting term after putting all the parts together is

$$\begin{aligned} &\left|\left(\bigcup_{j=a}^{m+1} e(m+1, \mathbf{1}'_w(j) \otimes [\langle a,b,0\rangle] \otimes \mathbf{0}'_w)\right)\right|_1^{i+1} \cdot \left|e(m+1, \bigcup_{k=1}^{i-1} \{E(m,v)\}_1^k)\right|_1^i + \\ &\left|e(m+1, \left\{\bigcup_{k=1}^{i} \{E(m,v)\}_1^k\right\}_1^i)\right|_1^{i+1} + |e(m+1, \{E(m,v)\}_1^{i+1})|_1^{i+1} \end{aligned} . \tag{27}$$

Replacing each of these parts of Eq. 27 with their size we get Eq. 28:

$$\begin{gathered}(((m+1)-a)+1)\cdot\left(\sum_{j=1}^{i-1}\binom{i-1}{j}\cdot((m-a)+1)^{(i-1)-j}\right)+\\((m-a)+1)^{i-1}+((m-a)+1)^{i}\end{gathered}\tag{28}$$

Distribution over the summation and placing $((m-a)+1)^{i-1}$ in the right most summation results in the following:

$$\begin{gathered}\left(\sum_{j=1}^{i-1}\binom{i-1}{j}\cdot((m-a)+1)^{i-j}\right)+\\\left(\sum_{j=0}^{i-1}\binom{i-1}{j}\cdot((m-a)+1)^{(i-1)-j}\right)+((m-a)+1)^{i}\end{gathered}\tag{29}$$

What we have to consider for the next step is the following rule concerning “n choose k” statements.

$$\binom{n}{k}=\binom{n-1}{k}+\binom{n-1}{k-1}\tag{30}$$

This relationship can be used to add the two summations together and get the following result:

$$\begin{gathered}\left(\sum_{j=1}^{i-1}\binom{i}{j}\cdot((m-a)+1)^{i-j}\right)+\\\left(\sum_{j=i-1}^{i-1}\binom{i-1}{j}\cdot((m-a)+1)^{(i-1)-j}\right)+((m-a)+1)^{i}\end{gathered}\quad.\tag{31}$$

Next thing to note is the following equalities

$$\sum_{j=i-1}^{i-1}\binom{i-1}{j}\cdot((m-a)+1)^{(i-1)-j}=\sum_{j=i}^{i}\binom{i}{j}\cdot((m-a)+1)^{i-j}$$

$$((m-a)+1)^{i}=\sum_{j=0}^{0}\binom{i}{j}\cdot((m-a)+1)^{i-j}$$

Thus, Eq. 31 is actually

$$\begin{array}{c} \sum_{j=1}^{i-1} \binom{i}{j} \cdot ((m-a)+1)^{i-j} + \\ \left(\sum_{j=i}^{i} \binom{i}{j} \cdot ((m-a)+1)^{i-j} \right) + \sum_{j=0}^{0} \binom{i}{j} \cdot ((m-a)+1)^{i-j} \end{array} . \tag{32}$$

Thus, from Eq. 32 we can derive the following summation, of which has a well known form:

$$\sum_{j=0}^{i} \binom{i}{j} \cdot ((m-a)+1)^{i-j} = (((m-a)+1)+1)^{i+1} \tag{33}$$

The derivation leading up to Eq. 33 shows us that

$$|E(m+1,v)|_1^{i+1} = ((m-a)+1)+1)^{i+1}$$

Holds, and thus by induction the theorem is true. □

Corollary 2. *Given a non-empty uniform standard quantifier chain vector v, then*

$$|E(m,v)| = \sum_{j=0}^{n-1} ((m - v(1,1)) + 1)^j$$

for all $v(1,1) \leq m < v(1,2)$.

When applying the actual operational semantics to a monitor, at every step of the external stream a new instance of the monitor is created. This has not been addressed in the analysis carried out in this section. Let us consider two uniform standard quantifier chain vectors v and v' such that,

$$v = \left[\langle a,b,0 \rangle_1^1, \langle a,b,0 \rangle_2^1, \cdots, \langle a,b,0 \rangle_n^0 \right]$$

and

$$v' = \left[\langle a+1,b+1,0 \rangle_1^1, \langle a+1,b+1,0 \rangle_2^1, \cdots, \langle a+1,b+1,0 \rangle_n^0 \right].$$

For a fixed value $v(1,1) \leq m < v(1,2)$, $E_{v'}(m,v')$ will be evaluated one step less than $E_v(m,v)$ because $m - v'(1,1) < m - v(1,1)$. We can think of this as v represents the monitor at time $t = 0$ and v' the monitor made at $t = 1$. We formalize this in the following definition.

Definition 20. *given a non-empty quantifier chain vector v, we define a another non-empty quantifier chain vector called a* step-pair *of v, written as $v : n$ for $n \in \mathbb{N}$, as follows:*

$$v : n = \left[\langle a_1+n, b_1+n, 0 \rangle_1^1, \langle a_2+n, b_2+n, 0 \rangle_2^1, \cdots, \langle a_{|v|}+n, b_{|v|}+n, 0 \rangle_{|v|}^0 \right]$$

where

$$v = \left[\langle a_1, b_1, 0\rangle_1^1, \langle a_2, b_2, 0\rangle_2^1, \cdots, \langle a_{|v|}, b_{|v|}, 0\rangle_{|v|}^0\right].$$

Note that $v : 0 = v$.

Corollary 3. *Given a non-empty uniform standard quantifier chain vector* v*, then*

$$\left|\bigcup_{i=0}^{v(1,2)-v(1,1)} E(v(1,2) - v(1,1), v : i)\right| = \sum_{i=1}^{v(1,2)-v(1,1)} \sum_{j=0}^{n-1} i^j =$$

$$\left(\sum_{i=2}^{v(1,2)-v(1,1)} \frac{(1 - i^{n+1})}{1 - i} + 1\right) + (n+1) = \left(\sum_{i=2}^{v(1,2)-v(1,1)} \frac{(1 - i^{n+1})}{1 - i}\right) + n + v(1,2) - v(1,1)$$

The only thing missing from Cor. 3 is that we assume we have already reached the position $v(1,1)$ on the external stream, and don't consider what happened before this position. Essentially, if we always consider the external stream to start at 0, then we have to assume that it always starts at zero. We can change this property by varying the lower bound.

Corollary 4. *Given a non-empty uniform standard quantifier chain vector* v *and* $c \in \mathbb{N}$*, such that* $0 \leq c \leq v(1,1)$ *, then*

$$|\bigcup_{i=c}^{(v(1,2)-1)} E_{v:(i-c)}(v(1,2) - 1, v : (i - c))| =$$

$$(v(1,1) - c) + \left(\sum_{i=2}^{v(1,2)-v(1,1)} \frac{(1 - i^{n+1})}{1 - i}\right) + n + v(1,2) - v(1,1) =$$

$$\left(\sum_{i=2}^{v(1,2)-v(1,1)} \frac{(1 - i^{n+1})}{1 - i}\right) + n + v(1,2) - c$$

From now on, instead of using the cumbersome formalization for maximum memory usage during evaluation found in 9 we will instead use the following

Definition 21. *Given a(n) (inverse-)standard quantifier chain vector* $v = s_1 \otimes \cdots \otimes s_j$ *and* $0 \leq c \leq v(1,1)$*, the* space bound *of the* v *starting at* c *is as follows:*

$$SB(c, v) = |\bigcup_{i=c}^{\max_j\{s_j(1,2)-s_j(1,1)\}} E_{v:(i-c)}(\max_j \{s_j(1,2) - s_j(1,1)\}, v : (i - c))|$$

5 Analysis of Standard Quantifier Chain Vectors

Standard quantifier chain vectors allow for monotonically increasing upper bounds within the QTAs. Unlike uniform standard quantifier chain vectors, if the last position in the first quantifier interval is reached, this does not mean that the values are removed from memory, rather they stay in memory till the last value of the last quantifier interval is reached. We reduce the analysis of a standard quantifier chain vector v to the analysis of the sections S^v of v, such that for a given vector $w \in D_v$ the respective section of w, $S^w \in DV_{S^v}$. Essentially, we are counting the vectors which no longer have any computational value, however, we ignore the empty vector as it is added to the degenerate set for completeness reasons. This method is what we outline in this section.

Lemma 1. *Given a non-empty standard quantifier chain vector v, such that $v = s_1 \otimes \cdots \otimes s_j$, then*

$$|E(k,v)|_1^i = |E(k, s_1 \otimes \cdots \otimes s_l)|_1^i = 0$$

where $1 \leq l \leq j$, $s_l(1,2) \leq k \leq s_j(1,2)$ and $1 \leq i \leq \sum_{w=1}^{l} |s_w|$.

Proof. It is quite obvious from Def. 16 that if the statement holds for $k = s_l(1,2)$ then it holds for all $s_l(1,2) \leq k \leq s_j(1,2)$. We also know that for all $w = \mathbf{1}_w \otimes \mathbf{h}_w \otimes \mathbf{0}_w \in E(s_l(1,2), v)$ that $\sum_{w=1}^{l} |s_l| \leq |\mathbf{1}_w|$ by the fact that $k = s_l(1,2)$. Thus, be the definition of $\{\cdot\}_1^i$,$E(k,v)_1^i \equiv \emptyset$ and thus, its cardinality is 0. The section part of the inequality holds for the same reason. □

Before moving on to prove a cardinality bound for standard quantifier chain vectors, we need to consider, rather than what constitutes an object included in the calculation of the cardinality, what is considered not to count. what we mean is the rules which concern themselves with the removed of degenerate set vectors, i.e. **E4** and **E5** of Def. 16. Though, these two rules perform multiple splits at once and remove the degenerate set vectors, we would like to count the number of splits which result in a degenerate set vector. It is simple to see, from observing the way Def. 16 deals with vectors of length one, that in total $b - a + 1$ splits of this form will occur. The individual splits construct the degenerate vectors $\langle a, b, 1\rangle$,$\langle a+1, b, 1\rangle$, $\cdots$, $\langle b, b, 1\rangle$.

Note that the rules **E4** and **E5** of Def. 16 can only be applied to the last position in a standard quantifier chain vector. This means if we want to know the cardinality of the removed instances, we need to only consider what is happening in the last position. We will represent this cardinality as $||e(k,v)||$. When we would like to work with the set of vectors removed instead we write $\}e(k,v)\{$.

Lemma 2. *Given a non-empty uniform standard quantifier chain vector v,*

$$||E_v(v(1,2), v)|| = ((v(1,2) - v(1,1)) + 1)^{|v|}.$$

Proof. Let us consider the case when $|v| = 1$, the the only rules from Def. 16 which can be applied are **E4** and **E5**. from now on in this proof we will use a for $v(1,1)$ and b for $v(1,2)$. Now let us consider instead of $E_(b,v)$, $E(a,v)$ which is equivalent to $e(a,v)$, according to the rules of Def. 16 we apply rule **E4** and the result is $e(a,v) = \left\{\left[\langle a+1,b,0\rangle_1^0\right]\right\}$, thus $||E_v(a,v)|| = 1$. We constructed the vector $\left[\langle a,b,1\rangle_1^0\right]$ from the degenerate set. We assume for our induction hypothesis that $||E(k,v)|| = k-a+1$ for $a \leq k < b$ and we show for $k+1$. We know by definition that $E(k+1,v) = e(k+1, E(k,v))$. Also we know that $E(k,v) = \left\{\left[\langle k+1,b,0\rangle_1^0\right]\right\}$ and that $||E(k,v)|| = k-a+1$. Evaluating $e(k+1, \left[\langle k+1,b,0\rangle_1^0\right])$ results in two possibilities, first when $k+1 < b$ we again apply rule **E4** and get $||E(k+1,v)|| = k-a+2$. However, if $k+1 = b$, then we instead apply rule **E5** and remove $\left\{\left[\langle b,b,0\rangle_1^0\right]\right\}$, doing so results in

$$||E(b,v)|| = ((b-a)+1)$$

and thus we have proven the basecase.

Let us now assume that the theorem holds for vectors v such that $|v| = m$,for $1 \leq m \leq n$. We now show that the theorem holds for $|v| = n+1$. From the vector v we can define two sub uniform quantifier vector chains $v' = [v(1), \cdots, v(|v|)]$ and $v'' = [v(1)]$. We know by the induction hypothesis that $||E(b,v')|| = ((b-a)+1)^{|v|}$ and $||E(b,v'')|| = ((b-a)+1)$. Now let us consider the vector $v' \otimes v''$ and count how my vectors are removed. Every vector $w \in \}E(b,v')\{$ can be extended by the vector v'' to produce a vector $w \otimes v''$. Note that $w \otimes v'' \notin DV_{v' \otimes v''}$ and $\mathbf{1}_{w \otimes v''} = w$, $\mathbf{0}_{w \otimes v''} = []$, and $\mathbf{h}_{w \otimes v''} = v''$. We can deduce from these facts that $||E(b, w \otimes v'')|| = ((b-a)+1)$ being that it is essentially the basecase. We know that there are $((b-a)+1)^{|v|}$ vectors w, and thus,

$$||E(b,v)|| = \left|\left|E(b, v' \otimes v'')\right|\right| =$$

$$\left|\left|E(b,v')\right|\right| * \left|\left|E(b,v'')\right|\right| = ((b-a)+1)^{|v|} * \left|\left|E(b, w \otimes v'')\right|\right| = ((b-a)+1)^{|v|+1}.$$

By the above argument we have proven that the theorem holds by induction. □

Lemma 3. *Given a non-empty standard quantifier chain vector v such that $v = s_1 \otimes \cdots \otimes s_j$, then*

$$||E(s_j(1,2),v)|| = \left|\left|E(s_{j-1}(1,2),v')\right|\right| \cdot \left|\left|E(s_j(1,2),v'')\right|\right|$$

where $v = v' \otimes v''$ and $v' = s_1 \otimes \cdots \otimes s_{j-1}$.

Proof. we consider Lem. 2 as a basecase when $j = 1$. In this case v' is considered to be empty. Let us now consider as our induction hypothesis that the theorem holds for vectors with $1 \leq l \leq j$ sections and we show that it also holds for vectors with $j+1$ sections. Let us consider a vector $v = s_1 \otimes \cdots \otimes s_j$, we know the recurrence above holds for v by the induction hypothesis. Consider a uniform standard quantifier chain vector $v' = s_{j+1}$ such that $s_j(1,2) < s_{j+1}(1,2)$. Using these vectors we can make a new standard quantifier chain vector $w = v \otimes v'$. Note that w has $j+1$ sections. We know the size of $\left|\left|E(s'_j(1,2),v)\right|\right|$ and $\left|\left|E_{v'}(v'(s'_{j1}(1,2),v')\right|\right|$ by the induction hypothesis. Let us consider for every $w \in \}E(b,w)\{$, the vector $w \otimes v'$. Note that $w \otimes v' \notin DV_{v' \otimes v'}$ and

$\mathbf{1}_{w\otimes v'} = w$, $\mathbf{0}_{w\otimes v''} = [v'(2), \cdots, v'(|v'|)]$, and $\mathbf{h}_{w\otimes v''} = v'(1)$. We can deduce from these facts that $||E(b, w\otimes v')|| = ||E(b, v')||$ being that it is essentially the Lem. 2. There are exactly $\left|\left|E(s'_j(1,2), v)\right|\right|$ vectors are of the form $w \otimes v'$ and thus we can can deduce that

$$\left|\left|E(b, v\otimes v')\right|\right| = \left|\left|E(s'_j(1,2), v)\right|\right| * \left|\left|E(b, v')\right|\right|$$

proving the theorem holds by induction. □

Corollary 5. *Given a non-empty standard quantifier chain vector v such that $v = s_1 \otimes \cdots \otimes s_j$, then*

$$||E(s_j(1,2), v)|| = \prod_{i=1}^{j}((s_i(1,2) - s_i(1,1)) + 1)^{|s_i|}.$$

What is really nice about Lem. 2 & Lem. 3 is that it indirectly give us a method for calculating the cardinality of an evaluation of a given formula. we use this indirect counting for the following theorem.

Theorem 3. *Given a non-empty standard quantifier chain vector v such that $v = s_1 \otimes \cdots \otimes s_j$, then*

$$|E(k,v)|_1^q = \left|\left|E(s_l(1,2), v^l)\right|\right| \cdot |E(k, v^j)|_1^{q-\sum_{r=1}^{l}|s_r|} =$$

$$\prod_{i=1}^{l}((s_i(1,2) - s_i(1,1)) + 1)^{|s_i|} \cdot ((k - s_1(1,1)) + 1)^{\left(q-\sum_{r=1}^{l}|s_r|\right)-1}$$

where $0 \le l < j$, $s_l(1,2) \le k < s_{l+1}(1,2)$, $\sum_{r=1}^{l}|s_r| < q \le \sum_{r=1}^{j}|s_r|$, $v^l = s_1 \otimes \cdots \otimes s_l$, and $v^j = s_{l+1} \otimes \cdots \otimes s_j$.

Proof. The basecase for this theorem considers the case when $l = 0$, this means the first term drops from the right-hand side of the equation:

$$|E(k,v)|_1^q = |E(k, v^j)|_1^q$$

Being that $s_1(1,1) \le k < s_1(1,2)$ we can consider v as a uniform standard quantifier chain vector. In that case, this is exactly the case considered in Thm. 2 and thus we have the following:

$$|E_v(k,v)|_1^q = |E_{v^j}(k, v^j)|_1^q = ((k - s_1(1,1)) + 1)^{q-1}$$

Thus the theorem holds for $l = 0$. Let us now assume for our induction hypothesis that the theorem holds for all $0 \le m \le l < j - 1$ and show that it holds for $l + 1$. Let us consider the case:

$$|E(k,v)|_1^q = \left|\left|E(s_l(1,2), v^l)\right|\right| \cdot |E(k, v^l)|_1^{q-\sum_{r=1}^{l}|s_r|}$$

were $\sum_{r=1}^{l+1} |s_r| < q \leq \sum_{r=1}^{j} |s_r|$. If q was allowed to be smaller we can not use this case to move from l to $l+1$. Now we can take the left-hand side of the equation and break it into three parts instead of two:

$$\left|\left|E(s_l(1,2), v^l)\right|\right| \cdot ||E(k, s_{l+1})|| \cdot |E(k, v^{j'})|_1^{q-\sum_{r=1}^{l+1}|s_r|}$$

where $v^{j'} = s_{l+2} \otimes \cdots \otimes s_j$. This transformation can be done because for all

$$w = \mathbf{1}_w \otimes \mathbf{h}_w \otimes \mathbf{0}_w \in \left\{E(k, v^j)\right\}_1^{q-\sum_{r=1}^{l}|s_r|},$$

where $|s_{l+1}| \leq |v_1|$. In other words, the vectors would have been removed when computing $E_{s_{l+1}}(k, s_{l+1})$. Next we need to consider a larger value of k, i.e. $s_{l+1}(1,2) \leq k < s_{l+2}(1,2)$. It is enough to prove the theorem by first considering $k = s_{l+1}(1,2)$ and then complete the argument by comparing the remaining part of the computation to Thm. 2; this was done in the base case. So for $k = s_{l+1}(1,2)$ we have the following:

$$\left|\left|E(s_l(1,2), v^l)\right|\right| \cdot ||E(s_{l+1}(1,2), s_{l+1})|| \cdot |E(s_{l+1}(1,2), v^{j'})|_1^{q-\sum_{r=1}^{l+1}|s_r|}.$$

By Lem. 3 we know

$$\left|\left|E(s_l(1,2), v^l)\right|\right| \cdot ||E(s_{l+1}(1,2), s_{l+1})|| = \left|\left|E(s_l(1,2), v^{l'})\right|\right|,$$

where $v^{l'} = s_1 \otimes \cdots \otimes s_{l+1}$. Thus we have

$$\left|\left|E(s_l(1,2), v^{l'})\right|\right| \cdot |E(s_{l+1}(1,2), v^{j'})|_1^{q-\sum_{r=1}^{l+1}|s_r|}.$$

Extending this argument to higher values of k with Thm. 2 we get

$$\left|\left|E(s_l(1,2), v^{l'})\right|\right| \cdot |E(k, s_1(1,1), v^{j'})|_1^{q-\sum_{r=1}^{l+1}|s_r|}$$

which is equivalent to the following:

$$\prod_{i=1}^{l+1}((s_i(1,2) - s_i(1,1)) + 1)^{|s_i|} \cdot ((k - s_1(1,1)) + 1)^{\left(q-\sum_{r=1}^{l+1}|s_r|\right)-1}$$

and thus we have proven the theorem by induction. □

Thm. 4 provides us with a precise bound on the number of instances of a formula kept in memory when evaluating a standard quantifier chain vector, however, for a single starting formula. The following corollary is concerned with additional monitor instances

Corollary 6. *Given a non-empty standard quantifier chain vector v and $0 \leq c \leq v(1,1)$, such that $v = s_1 \otimes \cdots \otimes s_j$, then*

$$SB(c, v) = (v(1,1) - c) +$$

$$\sum_{i_1=1}^{j} \sum_{i_2=(s_{i_1-1}(1,2)-s_{i_1-1}(1,1))+1}^{s_{i_1}(1,2)-s_{i_1}(1,1)} \left(\prod_{i_3=1}^{i_1-1} ((s_{i_3}(1,2) - s_{i_3}(1,1)) + 1)^{|s_{i_3}|} \cdot \sum_{i_4=1}^{\sum_{i_5=i_1}^{j} |s_{i_5}|} (i_2)^{(i_4-1)} \right)$$

where $s_0(1,2) = 1$ *and* $s_0(1,1) = 0$.

6 Analysis of Inverse-standard Quantifier Chain Vectors

A problem which has to be addressed concerning standard quantifier chain vectors is that most formula will not have the fixed structure it deals with. Most formulae will not be structured as a standard quantifier chain vector. We are left with the questions, is this a worst case analysis, is there a best case analysis, and is there a way, given that the lower and upper bound exists, to quantify the memory usage of formulae which have other quantifier structure. In this section we will address formulae with another fixed, but interesting structure, namely, the inverse-standard quantifier chain vectors. Also, we show that when concerned with non-trivial quantifier chain vectors (more then one section), standard quantifier chain vectors always require more space than inverse-standard quantifier chain vectors.

Theorem 4. *Given a non-empty inverse-standard quantifier chain vector* v *such that* $v = s_1 \otimes \cdots \otimes s_j$*, then*

$$|E(k,v)|_1^q = k^l \cdot \left(\prod_{i=1}^{l} ((k - s_i(1,1)) + 1)^{|s_i|-1} \right) \cdot ((k - s_{l+1}(1,1)) + 1)^{q-(\sum_{i=1}^{l} |s_i|)-1}$$

where $\sum_{i=1}^{l} |s_i| < q \leq \sum_{i=1}^{l+1} |s_i|$*,* $v(1,1) \leq k < s_{l+1}(1,2)$*, and* $0 \leq l < j$*.*

Proof. When $l = 0$ Then we drop the product and get the following:

$$|E_v(k,v)|_1^q = ((k - s_1(1,1)) + 1)^q.$$

Being that we are only concerning ourselves with the first section of v, this is essentially the case handled by Thm. 2. Thus, by Thm. 2 the basecase holds. Let us assume the theorem holds for $0 \leq d \leq l < j$ and show that it hold for $l + 1$. We start with l which we know holds by the induction hypothesis.

$$|E(k,v)|_1^q = k^{l-1} \cdot \left(\prod_{i=1}^{l-1} ((k - s_i(1,1)) + 1)^{|s_i|-1} \right) \cdot ((k - s_l(1,1)) + 1)^{q-(\sum_{i=1}^{l-1} |s_i|)-1} \tag{34}$$

Let us assume that $q = \sum_{i=1}^{l} |s_i|$ and thus, using Eq. 34, we get

$$|E(k,v)|_1^q = k^{l-1} \cdot \left(\prod_{i=1}^{l-1} ((k - s_i(1,1)) + 1)^{|s_i|-1} \right) \cdot ((k - s_l(1,1)) + 1)^{|s_l|-1}$$

Now we have to consider what $((k-s_l(1,1))+1)^{|s_l|-1}$ is counting, essentially it is counting how many instances in memory. However, we want to transitions from l to $l+1$ and that means we have to consider how many vectors would be removed from memory if l was the last section of the vector. To compute this, we need to multiple $|E(k,v)|_1^q$ by k because each vector in $|E(k,v)|_1^q$ has had k instances removed. Thus, we get

$$|| \{E(k,v)\}_1^q || = \left(k^l \cdot \prod_{i=1}^{l}((k - s_i(1,1)) + 1)^{|s_i|-1}\right)$$

Now let us consider a $\sum_{i=1}^{l} |s_i| < q' \leq \sum_{i=1}^{l+1} |s_i|$. We know the number of occurrences of $|E(k,v)|_1^{(\sum_{i=1}^{l} |s_i|)+1}$ will be $|| \{E(k,v)\}_1^q ||$. We also know that the s_{l+1} section can be treated independently as a uniform standard quantifier chain vector. Thus, using Theorem 2 we get the following:

$$|E(k, s_{l+1})|_1^{q'-\sum_{i=1}^{l}|s_i|} = ((k - s_{l+1}(1,1)) + 1)^{q'-(\sum_{i=1}^{l}|s_i|)-1}$$

Putting this together with $|| \{E(k,v)\}_1^q ||$ we get

$$|E(k,v)|_1^{q'} = || \{E(k,v)\}_1^q || \cdot ((k - s_{l+1}(1,1)) + 1)^{q'-(\sum_{i=1}^{l}|s_i|)-1}$$

which is equivalent to

$$|E(k,v)|_1^q = k^l \cdot \left(\prod_{i=1}^{l}((k - s_i(1,1)) + 1)^{|s_i|-1}\right) \cdot ((k - s_{l+1}(1,1)) + 1)^{q-(\sum_{i=1}^{l}|s_i|)-1}$$

and thus, we have proven the theorem by induction. □

Lemma 4. *Given a non-empty inverse-standard quantifier chain vector v such that $v = s_1 \otimes \cdots \otimes s_j$, then*

$$|E(k,v)|_1^q = 0$$

where $\sum_{i=1}^{l-1} |s_i| < q \leq \sum_{i=1}^{j} |s_i|$, $s_l(1,2) \leq k < s_1(1,2)$, and $0 \leq l < j$.

Proof. Unlike standard quantifier chain vectors upper bounds of the QTAs are in monotonically decreasing order, thus, we remove instances from the end rather than the beginning as k increases. □

Corollary 7. *Given a non-empty inverse-standard quantifier chain vector v such that $v = s_1 \otimes \cdots \otimes s_j$, then*

$$|E(m,v)| = |E(m,s_1)| = \sum_{j=0}^{|s_1|-1} ((m - s_1(1,1)) + 1)^j$$

for all $s_2(1,2) \leq m < s_1(1,2)$

Corollary 8. *Given a non-empty inverse-standard quantifier chain vector v such that $v = s_1 \otimes \cdots \otimes s_j$, then*

$$|E(m,v)| = \sum_{r=1}^{i} m^{r-1} \cdot \left(\prod_{g=1}^{r-1} ((m - s_g(1,1)) + 1)^{|s_g|-1} \right) \cdot \sum_{j=0}^{|s_r|-1} ((m - s_r(1,1)) + 1)^j$$

for all $s_{i+1}(1,2) \leq m < s_i(1,2)$, for $1 \leq i \leq j$. We assume $s_{j+1}(1,2) = 0$.

Corollary 9. *Given a non-empty inverse-standard quantifier chain vector v and $0 \leq c \leq v(1,1)$ such that $v = s_1 \otimes \cdots \otimes s_j$, then*

$$SB(c,v) = (v(1,1) - c) + \sum_{m=0}^{s_1(1,2)-s_1(1,1)} \sum_{r=1}^{\max_i\{s_i(1,2) \leq m\}}$$
$$m^{r-1} \cdot \left(\prod_{g=1}^{r-1} ((m - s_g(1,1)) + 1)^{|s_g|-1} \right) \cdot \sum_{j=0}^{|s_r|-1} ((m - s_r(1,1)) + 1)^j$$

We assume $s_{j+1}(1,2) = 0$ and allow $1 \leq i \leq j+1$.

Now we just have to show that given the same sections, the inverse-standard quantifier chain vector constructed from the sections will always use less memory than the standard quantifier chain vector constructed from the sections.

Corollary 10. *Given a non-empty inverse-standard quantifier chain vector v such that $v = s_1 \otimes \cdots \otimes s_j$, $j > 2$, and $c \in \mathbb{N}$, such that $0 \leq c \leq v(1,1)$, then*

$$SB(c,v) < SB(c,v')$$

where $v' = s_j \otimes s_{j-1} \otimes \cdots \otimes s_1$.

Proof. We skip the case when we only have one section because the two formulae are equal, i.e. there is no inverse of a one section standard quantifier chain vector. In the case of two sections $s_2(1,2) \leq s_1(1,2)$ and $v' = s_2 \otimes s_1$, the calculations computed by the two formula will be the same up to but not including position $s_2(1,2)$. At the position $s_2(1,2)$ all vectors w with $w(i,3) = 0$ for $1 \leq i \leq |s_2|$ will be removed from memory. However, because all of these vectors w will have $w(i,3) = 0$ for $|s_2| + 1 \leq i \leq |s_1|$, we still have to count them and multiple the evaluation of section s_1 by the number of vectors removed this way. However, if we switch the sections around $v = s_1 \otimes s_2$, at position $s_2(1,2)$, we remove all vectors w such that $w(i,3) = 0$ for $1 \leq i \leq |s_2|$, but no multipliers for the evaluation of s_1 are generated because s_1 is before section two. Thus in the standard case v', we first evaluate both sections together and than s_1 with a multiplier. Though, in the inverse standard case v we first evaluate both sections together and then s_1 alone. Thus, the standard case has more vectors in memory at position $|s_2| - 1$.

For the induction hypothesis, we assume the theorem holds for $2 \leq l \leq j$ and show for $j+1$. The argument is exactly the same as the basecase, but we now consider j sections constructing the multiplier or being removed. □

7 Bounding Top-Free Quantifier Vector Chains

Our goal in this section is to show that even though a complete analysis of top-free quantifier vector chains has not been carried out, we can still show that every top-free quantifier chain vectors is bounded from above and from below by a top-free quantifier chain vector, namely the inverse-standard ordering of its sections and the standard ordering of its sections. Though, the precise ordering between two top-free quantifier chain vectors is an open problem.

Definition 22. *Given a set of non-empty top-free quantifier chain vectors* $v_1, \cdots v_j$ *, for* $1 \leq j$*, such that* $v_1(1,1) = \cdots = v_j(1,1)$*, we define* $V^{\otimes}(v_1, \cdots, v_j)$ *as the set of all vectors constructable using* $\otimes$ *and all* j *vectors.*

Definition 23. *Given* $V^{\otimes}(v_1, \cdots, v_j)$ *for* $1 \leq j$*, we define* the standard quantifier chain vector relative to $V^{\otimes}(v_1, \cdots, v_j)$*, namely* $\mathbf{S}^{\otimes}_{\{v_1,\cdots,v_j\}}$*, as the vector* $v \in V^{\otimes}(v_1, \cdots, v_j)$ *such that for all* $w \in V^{\otimes}(v_1, \cdots, v_j)$ $SB(c, w) \leq SB(c, v)$*, for* $0 \leq c \leq v_1(1,1)$*. Also,* the inverse-standard quantifier chain vector relative to $V^{\otimes}(v_1, \cdots, v_j)$*, namely* $\mathbf{I}^{\otimes}_{\{v_1,\cdots,v_j\}}$*, is the vector* $v \in V^{\otimes}(v_1, \cdots, v_j)$ *such that for all* $w \in V^{\otimes}(v_1, \cdots, v_j)$ $SB(c, v) \leq SB(c, w)$*, for* $0 \leq c \leq v_1(1,1)$*.*

Definition 24. *Given* $V^{\otimes}(v_1, \cdots, v_j)$ *for* $1 \leq j$*, we define* the standard quantifier chain vector of $V^{\otimes}(v_1, \cdots, v_j)$*, namely* $s^{\otimes}_{\{v_1,\cdots,v_j\}}$*, as the vector* $v \in V^{\otimes}(v_1, \cdots, v_j)$ *such that* $v = v^1 \otimes \cdots \otimes v^j$ *and* $v^1(1,2) \leq \cdots \leq v^j(1,2)$*. Also,we define* the inverse-standard quantifier chain vector of $V^{\otimes}(v_1, \cdots, v_j)$*, namely* $i^{\otimes}_{\{v_1,\cdots,v_j\}}$*, as the vector* $v \in V^{\otimes}(v_1, \cdots, v_j)$ *such that* $v = v^1 \otimes \cdots \otimes v^j$ *and* $v^1(1,2) \geq \cdots \geq v^j(1,2)$*.*

Proposition 9. *Given uniform standard quantifier vectors* $v_1, \cdots, v_j$*, such that* $v_1(1,1) = \cdots = v_j(1,1)$ *and each* $v_i(1,2)$ *is distinct, for* $1 \leq i \leq j$*, then* $|V^{\otimes}(s_1, \cdots, v_j)| = j!$

Lemma 5. *Let* $v_1, \cdots, v_j$ *and* $v'_1, \cdots, v'_m$ *be non-empty uniform standard quantifier vectors, such that* $v_1(1,1) = \cdots = v_j(1,1) = v'_1(1,1) = \cdots = v'_m(1,1)$*. Let* $w \in V^{\otimes}(v_1, \cdots, v_j)$ *and* $v^1, v^2 \in V^{\otimes}(v'_1, \cdots, v'_m)$ *and* $SB(c, v^1) \leq SB(c, v^2)$ *for* $0 \leq c \leq v_1(1,1)$*. Then* $SB(c, w \otimes v^1) \leq SB(c, w \otimes v^2)$ *and* $SB(c, v^1 \otimes w) \leq SB(c, v^2 \otimes w)$*.*

Proof. When evaluating $e(s, w \otimes v_1)$ using Def. 16, at some point in the computation vectors of the form $w' \otimes v_1$ are constructed where $w' \in DV_w$. Further evaluation of $w' \otimes v_1$ ignores w' and essentially we have an evaluation of v_1. The same happens in the case of v_2. Being that w is treated the same in both cases, we know that the ordering existing between $SB(c, v_1)$ and $SB(c, v_2)$ will exist between $SB(c, w \otimes v_1)$ and $SB(c, w' \otimes v_2)$. The same argument can be constructed for $v_1 \otimes w$ and $v_2 \otimes w$. □

Theorem 5. *Given non-empty uniform standard quantifier vectors* $v_1, \cdots, v_j$*, such that* $v_1(1,1) = \cdots = v_j(1,1)$*, then* $\mathbf{S}^{\otimes}_{\{v_1,\cdots,v_j\}} = s^{\otimes}_{\{v_1,\cdots,v_j\}}$*, and* $\mathbf{I}^{\otimes}_{\{v_1,\cdots,v_j\}} = i^{\otimes}_{\{v_1,\cdots,v_j\}}$*.*

Proof. For the basecase, let us assume that $j = 2$ because $j = 1$ does not distinguish between standard and inverse standard quantifier vector chains, i.e. $\mathbf{S}^{\otimes}_{\{v_1,\cdots,v_j\}} =$

$s^{\otimes}_{\{v_1,\cdots,v_j\}} = \mathbf{I}^{\otimes}_{\{v_1,\cdots,v_j\}} = i^{\otimes}_{\{v_1,\cdots,v_j\}}$. Also, even if $j = 2$ and $v_1(1,2) = v_2(1,2)$, the two cases are equivalent and the theorem trivially holds. Thus, the interesting case is when $v_1(1,2) \neq v_2(1,2)$. By Prop. 9, $|V^{\otimes}(v_1,v_2)| = 2$. These two vectors are $s^{\otimes}_{\{v_1,v_2\}}$ and $i^{\otimes}_{\{v_1,v_2\}}$. By *Cor.*10 we know that $SB(c, i^{\otimes}_{\{v_1,v_2\}}) < SB(c, s^{\otimes}_{\{v_1,v_2\}})$, for $0 \leq c \leq v_1(1,1)$. We use the same constant c from now on. Thus, it must be the case that $\mathbf{S}^{\otimes}_{\{v_1,v_2\}} = s^{\otimes}_{\{v_1,v_2\}}$ and $\mathbf{I}^{\otimes}_{\{v_1,v_2\}} = i^{\otimes}_{\{v_1,v_2\}}$. We have shown that the basecase holds.

For the stepcase, let us assume the theorem holds for $1 \leq l \leq j$ and show for $j+1$. We assume w.l.o.g that $v_1(1,2) \leq \cdots \leq v_{j+1}(1,2)$. We take an arbitrary vector $w \in V^{\otimes}(v_1,\cdots,v_{j+1})$ such that $w = v_{\pi(1)} \otimes \cdots \otimes v_{\pi(j+1)}$ for some permutation π of $1,\cdots,j+1$. We break w into two parts $w^1 = v_{\pi(1)}$ and $w^2 = v_{\pi(2)} \otimes \cdots \otimes v_{\pi(j+1)}$. Notice that $|w^2| = j$ and thus, by the induction hypothesis $SB(c, w^2) \leq SB(c, s^{\otimes}_{\{v_{\pi(2)},\cdots,v_{\pi(j+1)}\}})$. By Lem. 5, we know that $SB(c, w) \leq SB(c, w^1 \otimes s^{\otimes}_{\{v_{\pi(2)},\cdots,v_{\pi(j+1)}\}})$. Now we have to consider two cases, either $w^1 = v_{\pi(1)} = v_{j+1}$ or $v_{\pi(j+1)} = v_{j+1}$. If $v_{\pi(j+1)} = v_{j+1}$, we break $s^{\otimes}_{\{v_{\pi(2)},\cdots,v_{\pi(j+1)}\}}$ into two parts, $s_1 = s^{\otimes}_{\{v_{\pi(2)},\cdots,v_{\pi(j)}\}}$ and $s_2 = v_{\pi(j+1)}$. The vector $v_{\pi(1)} \otimes s_1$ has a length of j and thus by the induction hypothesis $SB(c, v_{\pi(1)} \otimes s_1) \leq SB(c, s^{\otimes}_{\{v_{\pi(1)},\cdots,v_{\pi(j)}\}})$. By Lem. 5, we know that $SB(c, v_{\pi(1)} \otimes s_1 \otimes s_2) \leq SB(c, s^{\otimes}_{\{v_{\pi(1)},\cdots,v_{\pi(j)}\}} \otimes s_2)$. We have shown that

$$s^{\otimes}_{\{v_{\pi(1)},\cdots,v_{\pi(j)}\}} \otimes s_2 = s^{\otimes}_{\{v_{\pi(1)},\cdots,v_{\pi(j+1)}\}} = \mathbf{S}^{\otimes}_{\{v_1,\cdots,v_{j+1}\}}$$

We now consider the case when $w^1 = v_{\pi(1)} = v_{j+1}$. we break $s^{\otimes}_{\{v_{\pi(2)},\cdots,v_{\pi(j+1)}\}}$ into two parts, $s_1 = s^{\otimes}_{\{v_{\pi(2)},\cdots,v_{\pi(j)}\}}$ and $s_2 = v_{\pi(j+1)} = v_j$. The vector $v_{\pi(1)} \otimes s_1$ has a length of j and thus by the induction hypothesis $SB(c, v_{\pi(1)} \otimes s_1) \leq SB(c, s^{\otimes}_{\{v_{\pi(1)},\cdots,v_{\pi(j)}\}})$. By Lem. 5, we know that $SB(c, v_{\pi(1)} \otimes s_1 \otimes s_2) \leq SB(c, s^{\otimes}_{\{v_{\pi(1)},\cdots,v_{\pi(j)}\}} \otimes s_2)$. We now have

$$s^{\otimes}_{\{v_{\pi(1)},\cdots,v_{\pi(j)}\}} \otimes s_2 = v_1 \otimes \cdots \otimes v_{j+1} \otimes v_j.$$

If $v_j(1,2) = v_{j+1}(1,2)$ we are done. Else, by Cor. 10 and the base case we know that $SB(c, v_{j+1} \otimes v_j) < SB(c, v_j \otimes v_{j+1})$. By Lem. 5, we get the following:

$$SB(c, s^{\otimes}_{\{v_1,\cdots,v_{j-1}\}} \otimes v_{j+1} \otimes v_j) < SB(c, s^{\otimes}_{\{v_1,\cdots,v_{j-1}\}} \otimes v_j \otimes v_{j+1}).$$

Again we have shown that

$$s^{\otimes}_{\{v_1,\cdots,v_{j-1}\}} \otimes v_j \otimes v_{j+1} = s^{\otimes}_{\{v_{\pi(1)},\cdots,v_{\pi(j+1)}\}} = \mathbf{S}^{\otimes}_{\{v_1,\cdots,v_{j+1}\}}$$

Putting both cases together we have proven the theorem by induction for the standard case. for the inverse standard case, the method of proof is the same except ordering is reversed. □

8 Conclusion

The majority of this work has been carried out on abstract vectors rather than on logical formulae. It is hard to make a comparison with results concerning the logical formulae directly, unless, rather than concerning ourselves with the truth value and semantic meaning of the formulae, let us only consider the syntactic structure. We are considering chains of quantifier which have been converted into dominating formula. If we forget about the propositional structure and only the only look at the quantifier bounds the work makes more sense. Essentially, we have been taking any permutations of the bounds of the quantifiers in a given formula without considering the underlying formula structure. Though this destroys the logic meaning, it allows us to discuss different formulae at the same time because they are a permutation of the quantifier bounds provided. The simplest case of such formulae are formulae such that all the quantifier bounds are equivalent, only one permutation exists. These were the first formulae we found a memory bound for, *uniform standard quantifier chains*.

Using uniform standard quantifier chains, we where able to find a memory bound for two important permutations, namely monotonically increasing and monotonically decreasing quantifier bounds. It turns out that these two cases upper and lower bound, respectively, the memory usage of any permutation of the quantifier bounds. Even though computing the memory usage for these two cases was relatively easy, this is not the case for arbitrary permutations. Thus, we leave to future work the development of precise memory bounds for arbitrary permutations of the quantifier bounds. Though, we think that precise bounds for all the possible cases will not be possible, what is more likely possible is estimated bounds based on the structure of the quantifier bounds, i.e. which permutation is the current permutation most like, the lower bound or the upper bound. More empirical work will have to be carried out towards this goal.

References

[1] David Cerna. Space Complexity of Operational Semantics for the LogicGuard Core Language. Technical report, Research Institute for Symbolic Computation (RISC), Johannes Kepler University Linz, Schloss Hagenberg, 4232 Hagenberg, Austria, May 2015.

[2] Temur Kutsia and Wolfgang Schreiner. Verifying the Soundness of Resource Analysis for LogicGuard Monitors (Revised Version) . RISC Report Series 14-08, Research Institute for Symbolic Computation (RISC), Johannes Kepler University Linz, Schloss Hagenberg, 4232 Hagenberg, Austria, 2014.

[3] Wolfgang Schreiner. Logicguard ii, 2015. http://www.risc.jku.at/projects/LogicGuard2/.

[4] Wolfgang Schreiner, Temur Kutsia, Michael Krieger, Ahmad Bashar, Helmut Otto, and Martin Rummerstorfer. Monitoring Network Traffic by Predicate Logic. Technical report, Research Institute for Symbolic Computation (RISC), Johannes Kepler University, Linz, Austria, September 2014.