INFORMATIQUE THÉORIQUE ET APPLICATIONS

HELMUT SEIDL
DAMIAN NIWIŃSKI

On distributive fixed-point expressions

Informatique théorique et applications, tome 33, n° 4-5 (1999), p. 427-446

<http://www.numdam.org/item?id=ITA_1999__33_4-5_427_0>

Theoretical Informatics and Applications
Theoret. Informatics Appl. **33** (1999) 427–446

ON DISTRIBUTIVE FIXED-POINT EXPRESSIONS *

HELMUT SEIDL[1] AND DAMIAN NIWIŃSKI[2]

Abstract. For every fixed-point expression e of alternation-depth r, we construct a new fixed-point expression e' of alternation-depth 2 and size $\mathcal{O}(r \cdot |e|)$. Expression e' is equivalent to e whenever operators are distributive and the underlying complete lattice has a co-continuous least upper bound. We show that our transformation is optimal not only w.r.t. alternation-depth but also w.r.t. the increase in size of the resulting expression.

AMS Subject Classification. 68Q60, 03D70, 06D99, 68Q25.

1. INTRODUCTION

Fixed-Point calculus is a logical formalism based on explicit notation for inductive and co-inductive definitions. It is recognized as a useful framework especially for reasoning about temporal properties of finite state systems.

The role of *alternation* of least (μ) and greatest (ν) fixed-point operators as a source of a sharp expressive power for the fixed-point calculus has been recognized in the early 1980's [8, 9, 17]. In particular, Park in his studies on the semantics of parallelism [17] observed that the *fair merge* of two infinite sequences can be adequately characterized only using *both* extremal fixed-point operators. It was shown somewhat later that the alternation of μ and ν gives rise to a strict hierarchy of properties of infinite *trees*, corresponding to some hierarchy of Rabin automata [13,15]. The strongest result in this direction is the infinity of the alternation hierarchy in Kozen's propositional modal μ-calculus, which has been only

Keywords and phrases: Fixed-point expressions, distributivity, alternation-depth, lower bounds.

* *Supported by Polish KBN grant No. 8 T11C 002 11.*

[1] FB IV – Informatik, Universität Trier, 54286 Trier, Germany; e-mail: seidl@uni-trier.de

[2] Institute of Informatics, University of Warsaw, Ul. Banacha 2, 02-097 Warsaw, Poland; e-mail: niwinski@mimuw.edu.pl

recently established independently by Bradfield [5] and Lenzi [11]. A new and very elegant proof has been found by Arnold [2].

These facts can be contrasted with the early result by Park [18] stating that the hierarchy collapses to the $\nu\mu$ level in the algebra of ω-languages with the basic operations consisting of prefixing by a single letter, and binary set union[1]. In this framework the fixed-point definable ω-languages are precisely ω-regular languages, and collapsing of the hierarchy to the $\nu\mu$ level follows from the structure of acceptance condition of Büchi automata. The result was subsequently generalized in the second author's doctoral dissertation [14] to what is called there *monadic Kleene algebras*; it was however never published elsewhere.

In recent years, an interest is coming back in fixed-point calculi *weaker* than the original Kozen's modal μ-calculus. This is because of the *model checking* problem. The general question whether a given finite state transition system satisfies or not a μ-calculus formula, appears to be computationally hard[2]. Therefore, it would be of interest to separate a (sufficiently comprehensive) *sub-calculus* enjoying better complexity. One such successful case is a sub-logic L_2 proposed by Emerson *et al.* [7], and proven to be strong enough to subsume, and in fact to be equivalent to, the program logic ECTL*.

Now, an observation can be made that the logic L_2 admits a collapsing property exactly as the aforementioned calculus of Park, and in fact for the same reason: the structure of Büchi automata.

This implies that model-checkers for L_2-formulas can be constructed according to (at least) three strategies. The first one simply applies methods for general μ-calculus model-checking. The disadvantage is that efficiency problems may arise due to arbitrary nesting of fixed-points. A second alternative, as proposed by Bhat and Cleaveland [4], tries to improve upon this by exploiting special properties of L_2-formulas. The complication still is that potentially arbitrarily nested fixed-points have to be dealt with. The third strategy, therefore, relies on the collapsing property. It first eliminates deep nesting of alternating fixed-points by transforming the formula into an equivalent one of small alternation-depth (namely, 2). To the resulting formula either a specialized algorithm is applied or ordinary fixed-point iteration. The efficiency of this third approach, however, crucially depends on how small the extra overhead through transformation of the formula can be made.

In this context, we believe it worthy to understand the early result by Park in its full generality, and also to clarify its complexity aspect, *i.e.* the complexity of reduction of a fixed-point expression to the $\nu\mu$ level. To our best knowledge, the last question has not yet been considered in literature.

To attack this question, we first give a general algebraic condition sufficient for the hierarchy of fixed-point expressions to collapse to the $\nu\mu$-level. It requires of all the basic operations (over a complete lattice) to be distributive, and of the binary least upper bound "$\sqcup$", additionally, to commute with the greatest lower bound of

[1] As the second author points out, this result was obtained in collaboration with J. Tiuryn.

[2] The model checking problem for Kozen's μ-calculus is known to be in $NP \cap co\text{-}NP$ [7], and even in $UP \cap co\text{-}UP$ [10], but it is not known whether it is in P.

downward directed sets. (This generalization slightly improves the aforementioned one of [16].) Next, we show that a fixed-point expression e of the alternation-depth r can be transformed uniformly over the structures of the aforementioned property to an expression of alternation-depth 2 (in fact, $\nu\mu$) and size $\mathcal{O}(r \cdot |e|)$. We finally prove that this estimation of the size of the resulting expression is optimal not only w.r.t. alternation-depth but also w.r.t. the increase in size of the resulting expression.

2. Distributive functions

Assume D_1, D_2 are complete lattices. We call a function $f : D_1 \to D_2$ *distributive* iff $f\,(d_1 \sqcup d_2) = f\,d_1 \sqcup f\,d_2$. For any set V, we consider the set $D^V = \{f \mid f : V \to D\}$ as a complete lattice with componentwise ordering.

Fact 2.1. *For every non-empty finite set V, a function $f : D_1^V \to D_2$ is distributive iff*

$$f = \bigsqcup_{x \in V} f_x \circ \pi_x$$

for unary distributive functions $f_x : D_1 \to D_2$ where for $x \in V$, $\pi_x : D_1^V \to D_1$ is given by $\pi_x \rho = \rho\, x$.

Proof. Define $f_x\, d = f\,\langle x, d\rangle$ where $\langle x, d\rangle \in D_1^V$ maps x to d and all $y \neq x$ to $\bot$. □

This implies that a k–ary function $D^k \to D$ that is distributive in our sense is a $\sqcup$–combination of unary distributive functions. Therefore, we usually consider just *unary* distributive functions.

Assume D is a complete lattice. For monotonic $f : D \to D$, function f^* is defined as the least fixed-point

$$f^* = \mu F.f \circ F \sqcup I.$$

It follows easily from the iterative characterization of fixed points, that whenever f is distributive, then f^* is distributive as well. Non-empty subset $X \subseteq D$ is called *co-directed* iff $x_1, x_2 \in X$ implies that X also contains some z with $z \sqsubseteq x_1$ and $z \sqsubseteq x_2$. Least upper bound "$\sqcup$" is called *co-continuous* if it commutes with co-directed greatest lower bounds, *i.e.*,

$$d \sqcup \bigsqcap X = \bigsqcap_{x \in X} (d \sqcup x)$$

for every $d \in D$ and co-directed subset $X \subseteq D$.

The last condition holds in particular in any complete Boolean algebra [19]. Especially, this is the case for $D = 2^S$ (ordered by "$\subseteq$") for any base set S.

Our considerations rely onto the following key observation. (In what follows, for $f : D \to D$, the least fixed point of f, $\mu x.f(x)$, is abbreviated by μf; the same for ν.)

Theorem 2.2. *Assume D is a complete lattice, and $f, g : D \to D$ are monotonic functions where $g\,x = f\,x \sqcup d$ for some $d \in D$. Then*

1. $\mu\, g = f^*\, d$;
2. *If binary "$\sqcup$" is co-continuous and f is distributive, then*

$$\nu\, g = \nu\, f \sqcup \mu\, g.$$

Note that similar identities have been considered by Doornbos *et al.* in [6] for relation algebras. Identity (2) turns out to be especially useful for the simplification of temporal logic formulas. The following example for LTL (linear time logic on infinite words) has been brought to our attention by Arnold.

Example 2.3. In LTL, formulas denote languages of infinite words. Formula $p\,\mathbf{U}\,q$ (p "until" q) is equivalent to the μ-formula $\mu x.q \vee (p \wedge \mathbf{X}\,x)$ where $\mathbf{X}$ is the temporal "next" operator. We would like to demonstrate how Theorem 2.2 can be used to derive a useful identity for $\neg(p\,\mathbf{U}\,q)$. By duality, we first obtain

$$\neg(p\,\mathbf{U}\,q) = \nu x.\neg q \wedge (\neg p \vee \mathbf{X}\,x) = \nu x.(\neg q \wedge \neg p) \vee (\neg q \wedge \mathbf{X}\,x). \tag{1}$$

Expressions $\neg q$ and $\neg p$ are interpreted as constants and $\mathbf{X}$ as well as intersection with $\neg q$ as unary distributive operators. Furthermore, $\vee$ represents binary lub. Therefore, Theorem 2.2 is applicable and gives us

$$\begin{array}{rcll} \neg(p\,\mathbf{U}\,q) & = & (\nu x.\neg q \wedge \mathbf{X}\,x) & \vee \quad (\mu x.(\neg q \wedge \neg p) \vee (\neg q \wedge \mathbf{X}\,x)) \\ & = & \mathbf{G}\,\neg q & \vee \quad \neg q\,\mathbf{U}\,(\neg q \wedge \neg p) \end{array}$$

where $\mathbf{G}$ denotes the "always" operator.

Proof of Theorem 2.2. In order to prove the first assertion, consider g^τ and F^τ, τ an ordinal, defined by

- $g^0 = \bot$, $F^0 = \lambda x.\bot$;
- $g^{\tau+1} = g\,g^\tau$, $F^{\tau+1} = f \circ F^\tau \sqcup I$; and
- for τ a limit ordinal, $g^\tau = \bigsqcup_{\tau' < \tau} g^{\tau'}$, $F^\tau = \bigsqcup_{\tau' < \tau} F^{\tau'}$.

Then $\mu\, g = g^\sigma$ as well as $f^* = F^\sigma$ for some ordinal σ. Therefore, it suffices to verify through transfinite induction that, for every ordinal τ, $g^\tau = F^\tau\, d$.

Now consider the second assertion. By definition, $\mu\, g \sqsubseteq \nu\, g$. Since $f \sqsubseteq g$, we also have $\nu\, f \sqsubseteq \nu\, g$. Therefore, $\nu\, g \sqsupseteq \nu\, f \sqcup \mu\, g$, and it remains to prove the reverse inclusion. Therefore define

- $g^0 = \top = f^0$;
- $g^{\tau+1} = g\,g^\tau$, $f^{\tau+1} = f\,f^\tau$; and
- for τ a limit ordinal, $g^\tau = \bigsqcap_{\tau' < \tau} g^{\tau'}$, $f^\tau = \bigsqcap_{\tau' < \tau} f^{\tau'}$.

Then $\nu\, g = g^\sigma$ and $\nu\, f = f^\sigma$ for some ordinal σ, and it suffices to prove that, for every ordinal τ, $g^\tau \sqsubseteq f^\tau \sqcup \mu\, g$. Again, we proceed by transfinite induction.

The case $\tau = 0$ trivially holds.

If $\tau = \tau' + 1$, then

$$g^\tau = g\, g^{\tau'} \sqsubseteq g\left(f^{\tau'} \sqcup \mu g\right) = f\, f^{\tau'} \sqcup g\,(\mu g) = f^\tau \sqcup \mu g$$

by ind. hypothesis and distributivity of f.

If τ is a limit ordinal, then, applying the induction hypothesis to all $\tau' < \tau$, we obtain:

$$g^\tau = \bigsqcap_{\tau' < \tau} g^{\tau'} \sqsubseteq \bigsqcap_{\tau' < \tau} \left(f^{\tau'} \sqcup \mu g\right) = \left(\bigsqcap_{\tau' < \tau} f^{\tau'}\right) \sqcup \mu g = f^\tau \sqcup \mu g$$

and the assertion follows.

3. Fixed-point expressions

Let us now study in detail the impact of identity (2) of Theorem 2.2 on fixed-point expressions in the spirit of Example 2.3. We fix a countably infinite set of variables $\mathcal{X}$, a set of constant symbols $\mathbf{C}$, and a set of unary function symbols $\mathbf{F}$ ($\mathbf{C}$ and $\mathbf{F}$ are not necessarily finite).

A *fixed-point expression* e is given by the following grammar:

$$e \quad ::= \quad c \mid x \mid f\, e \mid \mu x.e \mid \nu x.e \mid e_1 \sqcup e_2 \mid (e)$$

where $c \in \mathbf{C}$ and $f \in \mathbf{F}$.

An *interpretation* is a complete lattice D given together with an element c^D, for each $c \in \mathbf{C}$, and a distributive function $f^D : D \to D$, for each $f \in \mathbf{F}$.

The *meaning* $[\![e]\!]_D$ of a fixed-point expression e is a function $D^{\mathcal{X}} \to D$ mapping variable assignments to values. It is defined inductively as follows. (We usually omit the subscript D for simplicity. The symbols $\bigsqcap, \bigsqcup, \sqcup$ refer to the corresponding operations in D.)

$$\begin{aligned}
[\![c]\!]\,\rho &= c^D \\
[\![x]\!]\,\rho &= \rho\, x \\
[\![(e)]\!]\,\rho &= [\![e]\!]\,\rho \\
[\![f\, e']\!]\,\rho &= f^D\,([\![e']\!]\,\rho) \\
[\![\mu x.e']\!]\,\rho &= \bigsqcap\{d \mid [\![e']\!]\,\rho\{x \mapsto d\} \sqsubseteq d\} \\
[\![\nu x.e']\!]\,\rho &= \bigsqcup\{d \mid [\![e']\!]\,\rho\{x \mapsto d\} \sqsupseteq d\} \\
[\![e_1 \sqcup e_2]\!]\,\rho &= [\![e_1]\!]\,\rho \sqcup [\![e_2]\!]\,\rho
\end{aligned}$$

where $\rho\{x \mapsto d\}$ is an assignment that maps x to d and otherwise agrees with ρ.

We assume that the scope of the operator θ in $\theta x.e'$ extends to the end of the expression e'. *e.g.* $[\![\nu\, x.e_1 \sqcup e_2]\!]$ coincides with $[\![\nu\, x.(e_1 \sqcup e_2)]\!]$ and (in general) not with $[\![(\nu\, x.e_1) \sqcup e_2]\!]$.

It is easy to see that $[\![e]\!]\ \rho$ depends only on the values that ρ assigns to the free variables of e. For this reason we may safely write $[\![e]\!]\ \rho$ also for a partial assignment $\rho \in D^{\mathcal{Y}}$, whenever $\mathcal{Y} \subseteq \mathcal{X}$ contains the free variables of e. For two (partial) assignments ρ and ρ' with disjoint domains, we denote (the assignment being) their set–theoretic union by $\rho + \rho'$.

In case $[\![e_1]\!] = [\![e_2]\!]$ we also will write $e_1 = e_2$, as opposed to $e_1 \equiv e_2$ which denotes *syntactical identity.*

Proposition 3.1. *If all f^D are distributive and the operation $\sqcup$ is co-continuous in D, then $[\![e]\!]_D$ is distributive.*

In fact, it is this proposition which allows us to call e itself distributive and thus justifies the title of our paper.

Proof. We proceed by induction on the structure of e. The only non-trivial case is where e equals a fixed-point expression $\theta x.e'$ ($\theta \in \{\mu, \nu\}$). Let $\mathcal{Y}$ denote the set of free variables of e, and $\rho \in D^{\mathcal{Y}}$. By Fact 2.1, $[\![e']\!]\,(\rho + \{x \mapsto d\}) = f_x\, d \sqcup f'\, \rho$ for distributive functions $f_x : D \to D$ and $f' : D^{\mathcal{Y}} \to D$. If e is a least-fixed-point expression (*i.e.*, $\theta = \mu$), then by Theorem 2.2,

$$[\![e]\!] = [\![\mu x.e']\!] = f_x^* \circ f'.$$

Since distributive functions are closed under composition and "$*$", the assertion follows.

Accordingly, if e is a greatest-fixed-point expression (*i.e.*, $\theta = \nu$), then by Theorem 2.2,

$$[\![e]\!] = [\![\nu x.e']\!] = (\lambda\rho.\, \nu f_x) \sqcup (f_x^* \circ f').$$

Since constant functions are distributive and distributive functions are closed under "$\sqcup$", the assertion follows. □

4. Alternation-depth

For closed fixed-point expression e, we need to determine the maximal depth of nesting of greatest and least fixed-points. Let e' be a non-closed subexpression e' of e. Then the *hook* of e' is the smallest superexpression e'' of e' with the property that some variable that occurs free in e' is bound in e''. (Clearly, the hook of e' must start with a fixed-point operator.) Given this definition, the *level* of each subexpression of e is determined by the following topdown traversal of e.

Every closed subexpression e' receives level 1. Now assume e' is not closed but all proper superexpressions of e' have already received levels. Thus especially, the hook e'' of e' has received some level, say r. Then the level of e' is calculated as

follows. If $e' \equiv \theta_1 x.e_1$ and $e'' \equiv \theta_2 x.e_2$ with $\theta_1 \neq \theta_2$ then e' receives level $r+1$. Otherwise, e' receives level r.

The *alternation–depth* of closed expression e is then defined as the maximal level of a subexpression of e. Expressions of alternation–depth 1 are also called *alternation–free*. (The above definition is consistent with the alternation hierarchy of fixed-point expressions considered in [16]. More specifically, an expression has the alternation-depth n here iff it is in the class Comp $(\Sigma_n^\mu \cup \Pi_n^\mu)$ considered there.)

We illustrate this definition by the following example.

Example 4.1. Let

$$e \equiv \mu x.\nu y.f\, y \sqcup (\mu z.g\, z \sqcup x).$$

Since e is closed, e receives level 1. e serves as the hook for its subexpression $e_1 \equiv \nu y.f\, y \sqcup (\mu z.g\, z \sqcup x)$. Since e and e_1 are of different type (e is a least-fixed-point expression whereas e_1 is a greatest-fixed-point expression), e_1 receives level 2. e also serves as hook for subexpression $e_2 \equiv \mu z.g\, z \sqcup x$. Since e and e_2 are of the same type (both are least-fixed-point expressions), e_2 receives the same level as e, namely 1. Overall, we find that the whole expression is of alternation-depth 2.

The level numbers of subexpressions can be used to group several fixed-point iterations into one joint iteration. In the following we will always assume that all bound variables have distinct names which are also distinct from the names of all free variables. The next proposition is a useful special instance of the Bekič-Park principle [16, 17].

Proposition 4.2. *Assume $\bar{e}$ is a closed fixed-point expression, and $e \equiv \theta x_1.e_1$ is a fixed-point subexpression of $\bar{e}$ at level r ($\theta \in \{\mu, \nu\}$) where $\mathcal{Y}$ is the set of free variables of e. Let $\mathcal{X} = \{x_1, \ldots, x_m\}$, and assume that $\theta x_2.e_2, \ldots, \theta x_m.e_m$ are fixed-point subexpressions of e_1 which are also of level r and have free variables only in $\mathcal{X} \cup \mathcal{Y}$. Then these fixed-points can be computed jointly:*

For $j = 1, \ldots, m$, let e'_j denote the expression obtained from e_j by replacing each occurrence of $\theta x_i.e_i$ with x_i $(i = 1, \ldots, m)$. Then for every $\rho \in D^{\mathcal{Y}}$,

$$\begin{array}{lcll} [\![e]\!]\,\rho & = & (\theta F)\, x_1 & \text{where} \\ F & : & D^{\mathcal{X}} \to D^{\mathcal{X}} & \text{is given by} \\ F\,\beta\, x_j & = & [\![e'_j]\!]\,(\rho + \beta) & \text{for } j = 1, \ldots, m. \end{array}$$

We now arrive at our main theorem of this section.

Theorem 4.3. *For every fixed-point expression e, an expression $\tilde{e}$ can be constructed such that the following holds.*

1. $[\![e]\!]_D = [\![\tilde{e}]\!]_D$ *for every interpretation D with distributive operators, provided that operation "$\sqcup$" is co-continuous.*
2. *$\tilde{e}$ is of alternation-depth at most 2, and every greatest fixed-point subexpression of $\tilde{e}$ has level 1.*

3. $\tilde{e}$ has size $\mathcal{O}(r \cdot |e|)$ where r is the alternation-depth of e.

Expressions with property (2) are also said to be of class $\nu\mu$ (Π_2^μ, in the notation of [16]).

Proof. For the following we assume an interpretation D where operation "$\sqcup$" is co-continuous. In order to get an intuition about the proof, let us first consider the expression

$$e \equiv \nu x_r.\mu x_{r-1} \ldots \nu x_2.\mu x_1.f_1\, x_1 \sqcup \ldots \sqcup f_r\, x_r \sqcup d$$

where $r = 2 \cdot k$. In order to transform the innermost greatest fixed-point subexpression $e_2 \equiv \nu x_2.\mu x_1.f_1\, x_1 \sqcup \ldots \sqcup f_r\, x_r \sqcup d$, we calculate:

$$\begin{aligned}
e_2 &= \nu x_2.f_1^*\,(f_2\, x_2 \sqcup \ldots \sqcup f_r\, x_r \sqcup d) \\
&= \nu x_2.f_1^*\,(f_2\, x_2) \sqcup \ldots \sqcup f_1^*\,(f_r\, x_r) \sqcup f^*\, d \\
&= z_2 \sqcup \mu x_2.f_1^*\,(f_2\, x_2) \sqcup \ldots \sqcup f_1^*\,(f_r\, x_r) \sqcup f^*\, d \\
&= z_2 \sqcup \mu x_2.\mu x_1.f_1\, x_1 \sqcup \ldots \sqcup f_r\, x_r \sqcup d \qquad \text{where} \\
z_2 &= \nu x_2.f_1^*\,(f_2\, x_2) \\
&= \nu x_2.\mu x_1.f_1\, x_1 \sqcup f_2\, x_2.
\end{aligned}$$

Corresponding calculations also for greatest-fixed-point subexpressions corresponding to variables $x_4, \ldots, x_r$ finally result in:

$$\begin{aligned}
e &= z_r \sqcup (\mu x_r.\mu x_{r-1}.z_{r-2} \sqcup (\mu x_{r-2}.\mu x_{r-3}.z_{r-4} \sqcup \ldots \\
&\qquad (\mu x_2.\mu x_1.f_1\, x_1 \sqcup \ldots \sqcup f_r\, x_r \sqcup d) \ldots)) \qquad \text{where} \\
z_j &\equiv \nu x_j.\mu x_{j-1}.\mu x_{j-2} \ldots \mu x_1.f_1\, x_1 \sqcup \ldots \sqcup f_j\, x_j
\end{aligned}$$

for $j = 2, 4, \ldots, r$. Using the general identity

$$\mu x.\mu y.e \;=\; \mu x.(e\,\{y \mapsto x\})$$

we can simplify this to:

$$\begin{aligned}
e &= z_r \sqcup (\mu x_{r-1}.z_{r-2} \sqcup (\mu x_{r-3}.z_{r-4} \sqcup \ldots \\
&\qquad (\mu x_1.(f_1 \sqcup f_2)\, x_1 \sqcup \ldots \sqcup (f_{r-1} \sqcup f_r)\, x_{r-1} \sqcup d) \ldots)) \qquad \text{where} \\
z_j &\equiv \nu x_j.\mu y_j.(f_1 \sqcup \ldots \sqcup f_{j-1})\, y_j \sqcup f_j\, x_j
\end{aligned}$$

for $j = 2, 4, \ldots, r$. Each f_j occurs at most $r/2 + 1$ times in the new expression. Thus, the size of the resulting expression is bounded by $\mathcal{O}(r \cdot |e|)$ – as claimed by Theorem 4.3.

One possibility to generalize this idea to a transformation for arbitrary distributive fixed-point expressions is through *vectorial* fixed-points. Applying Proposition 3.1, we can bring every distributive fixed-point expression into form e – provided we allow functions f_j to operate on a suitable *power* of the base

distributive lattice D. Then we may apply the construction above which increases the dimension of involved vectors at most by a factor of $r+1$. Finally, we may transform the result back into an ordinary expression. In general however, the latter transformation may cause an *exponential* blow-up in expression size. Therefore, we prefer to present another, *direct* transformation which works for arbitrary distributive expressions.

We need some terminology. Greatest-fixed-point expression $e \equiv \nu x.e_1$ is called *simple* if

- e is closed;
- e_1 does not contain closed subexpressions;
- all greatest-fixed-point subexpressions have level 1.

By definition, simple expressions are of alternation-depth at most 2. Simple subexpressions can be seen as a (slight) generalization of the greatest fixed-point expressions z_j above. Thus, expression $\nu x.f\, x \sqcup (\mu y.g\, y \sqcup x)$ is simple whereas expression $\nu x.f\, x \sqcup (\mu y.g\, y)$ is not, since subexpression $\mu y.g\, y$ is closed.

Assume we are given an arbitrary expression $\bar{e}$. W.l.o.g. we assume that all fixed-point subexpressions are non-trivial, *i.e.*, fixed-point variables occur in the respective bodies at least once. Our key idea is to proceed by removing as many greatest fixed-points *simultaneously* as possible.

An expression $\bar{e}$ is called *r-clean* if it is obtained from an expression without ν-variables at levels $\geq r$ by replacing some constant symbols with simple expressions. Since simple expressions are closed, we conclude that closed expressions which are 1-clean are of alternation-depth at most 2 as well. Therefore, we aim at a construction which takes an $(r+1)$-clean expression $\bar{e}$ and constructs an equivalent r-clean expression. This construction will be made up of a sequence of local improvement steps.

So, let us assume $\bar{e}$ is $(r+1)$-clean. Consider a greatest-fixed-point subexpression $e = \nu x_1.e_1'$ in $\bar{e}$ of level r. In order to serve as a candidate for local improvement, e should be "maximal" in the sense that e should comprise as many greatest fixpoint variables as possible. On the other hand e also should be "minimal" in the sense that e itself should not contain candidates for improvement.

Formally, we grasp these two ideas as follows. We call e a *candidate* if

1. e is not simple,
2. e is closed (in case $r = 1$) or the hook of e has level $r-1$; and
3. e is minimal with this property.

Since e is chosen minimal, it may not contain further candidates as proper subexpressions. Since e is closed or has a hook of level $r-1$, e is not situated within the scope of another greatest fixed-point variable of the same level r being mutually recursive with those in e.

Assume e is a candidate with set $\mathcal{Y}$ of free variables.

Let $\mathcal{X} = \{x_1, \ldots, x_m\}$ where $\nu x_2.e_2', \ldots, \nu x_m.e_m'$ precisely are the greatest fixed-point subexpressions of e_1' at level r. By definition of a candidate, the free variables of each $e_i', i = 1, \ldots, m$, are contained in $\mathcal{X} \cup \mathcal{Y}$. Furthermore,

let $e'_s, s = m+1, \ldots, n$, denote the maximal subexpressions of e'_1 which contain free variables *only* from $\mathcal{Y}$. None of the e'_s may contain any of the greatest-fixed-point subexpressions of level r, because otherwise they themselves would contain candidates – which is excluded by minimality of e. Thus, the e'_s are precisely the closed subexpressions of e'_1 (in case $r = 1$) or the maximal subexpressions of e'_1 of level $< r$ (in case $r > 1$).

Our goal is to replace e by an expression $e_1 \sqcup e_2$ where e_1 is simple and e_2 is r-clean.

Expression e_1 is obtained by replacing all subexpressions $e'_s, s > m$, with $\bot$ and then absorbing lubs $e' \sqcup \bot$ resp. $\bot \sqcup e'$ into e'. Expression e_2 is constructed by replacing in e all occurrences of binders νx_j, $j = 1, \ldots, m$, with μx_j, respectively.

We illustrate the construction by the following example.

Example 4.4. Consider expression

$$\bar{e} \equiv \mu\, y_1.f_1\,(\nu\, x_1.f_2\,(\mu\, y_2.g_2\, y_2 \sqcup(\nu x_2.h_1\, x_1 \sqcup h_2\, x_2 \sqcup g_1\, y_1)) \sqcup d).$$

Expression $\bar{e}$ is of alternation-depth 3 where the maximal level of a greatest-fixed-point expression is 2. The only candidate in $\bar{e}$ is given by:

$$e \equiv \nu\, x_1.f_2\,(\mu\, y_2.g_2\, y_2 \sqcup(\nu x_2.h_1\, x_1 \sqcup h_2\, x_2 \sqcup g_1\, y_1)) \sqcup d.$$

Observe that greatest-fixed-point subexpression $\nu x_2.h_1\, x_1 \sqcup h_2\, x_2 \sqcup g_1\, y_1$ is also of level 2, but not a candidate since its hook (which is the expression e) has level 2 as well. Candidate e is transformed into $e_1 \sqcup e_2$ where

$$\begin{aligned} e_1 &\equiv \nu\, z_1.f_2\,(\mu\, y.g_2\, y \sqcup (\nu\, z_2.h_1\, z_1 \sqcup h_2\, z_2)) \\ e_2 &\equiv \mu\, x_1.f_2\,(\mu\, y_2.g_2\, y_2 \sqcup(\mu x_2.h_1\, x_1 \sqcup h_2\, x_2 \sqcup g_1\, y_1)) \sqcup d. \end{aligned}$$

In the newly generated simple expression e_1, we could eliminate not only constants but also closed subexpressions and even subexpressions which become closed if all free variables are replaced with $\bot$.

Thus overall, expression $\bar{e}$ is transformed into:

$$\begin{aligned} &\mu\, y_1.f_1\,(\nu\, z_1.f_2\,(\mu\, y.g_2\, y \sqcup (\nu\, z_2.h_1\, z_1 \sqcup h_2\, z_2)) \sqcup \\ &\quad (\mu\, x_1.f_2\,(\mu\, y_2.g_2\, y_2 \sqcup(\mu x_2.h_1\, x_1 \sqcup h_2\, x_2 \sqcup g_1\, y_1)) \sqcup d)). \end{aligned}$$

The greatest-fixed-point subexpressions of new expression e_1 indeed are all of level 1.

Let $\tilde{e}$ be obtained from $\bar{e}$ by replacing candidate e with $e_1 \sqcup e_2$. For the correctness of our transformation, we verify:

1. e_1 is simple;
2. If $r = 1$, then all fixed-point subexpressions of e_2 outside simple subexpressions are least fixed-points of level 1;
3. If $r > 1$, then all fixed-point subexpressions of e_2 outside simple subexpressions have level at most $r - 1$ where variables in $\mathcal{X}$ have precisely level $r - 1$;

4. $e = e_1 \sqcup e_2$.

Note that the new levels of subexpressions of e_2 are always determined relative to the transformed expression $\tilde{e}$, *i.e.*, the context of e in $\bar{e}$.

We only prove assertion 4. Assume e_i'' is obtained from e_i' by replacing occurrences of greatest-fixed-point expressions $\nu x_j.e_j', j \neq i$, with x_j, respectively. Since all free variables of e_i' are contained in $\mathcal{X} \cup \mathcal{Y}$, the same holds true also for the $\nu x_i.e_i''$. For $\rho \in D^{\mathcal{Y}}$, we therefore can define $G : D^{\mathcal{X}} \to D^{\mathcal{X}}$ by

$$G\,\beta\,x_i = [\![e_i'']\!]\,(\beta + \rho).$$

We claim:

5. $(\nu G)\,x_1 = [\![e]\!]\,\rho$;

6. $(\mu G)\,x_1 = [\![e_2]\!]\,\rho$.

Assertion 5 follows since all the greatest-fixed-point expressions of e have identical level r. Therefore by Proposition 4.2, the corresponding fixed-points can be computed jointly. Assertion 6 follows analogously for the least fixed-point expressions $\mu x_i.e_i'$ of e_2 which (according to assertion 2 and 3) all have the same level.

By distributivity, function G can be decomposed. For $i \in \{1, \ldots, m\}$, define expression $e_i^{\mathcal{X}}$ by replacing in e_i'' all expressions $e_s, s > m$, with $\perp$. Additionally, define expression $e_i^{\mathcal{Y}}$ by replacing in e_i'' all variables x_j, $j \in \{1, \ldots, m\}$, with $\perp$.

According to this construction, $e_i^{\mathcal{X}}$ contains free variables only from $\mathcal{X}$, whereas $e_i^{\mathcal{Y}}$ contains free variables only from $\mathcal{Y}$. Consequently,

$$[\![e_i'']\!]\,(\beta + \rho) = [\![e_i^{\mathcal{X}}]\!]\,\beta \sqcup [\![e_i^{\mathcal{Y}}]\!]\,\rho$$

for $\beta \in D^{\mathcal{X}}$. Therefore, $G\,\beta = F\,\beta \sqcup H$ where $F : D^{\mathcal{X}} \to D^{\mathcal{X}}$ and $H \in D^{\mathcal{X}}$ are given by

$$F\,\beta\,x_i = [\![e_i^{\mathcal{X}}]\!]\,\beta \qquad H\,x_i = [\![e_i^{\mathcal{Y}}]\!]\,\rho.$$

By Theorem 2.2, we have:

$$\nu G = \nu\,\beta.F\,\beta \sqcup H = \nu F \sqcup \mu G.$$

Projecting this equality to the component for x_1, we obtain:

$$[\![e]\!]\,\rho = (\nu G)\,x_1 = (\nu F)\,x_1 \sqcup (\mu G)\,x_1 = [\![e_1]\!]\,\emptyset \sqcup [\![e_2]\!]\,\rho.$$

Here, the equality between $(\nu F)\,x_1$ and $[\![e_1]\!]\,\emptyset$ is another instance of Proposition 4.2. Overall, assertion 4 follows.

Applying the given local improvement successively to candidates at level r, we obtain a r-clean expression equivalent to $\bar{e}$.

We are now going to estimate the size of the resulting expression. For that and in the following, we feel free to view expressions as (ordered finite) unary-binary trees where the leaves are labeled with variables or constants, unary nodes are

labeled with $\mu x, \nu x$ (x a variable) or $f \in \Sigma$ whereas binary nodes are labeled with $\sqcup$. Then the size $|e|$ of e is given by the total number of nodes of e. Moreover, we define the *basic* size $|e|_b$ of e as the number of the non-binary nodes of e. Clearly,

$$|e|_b \leq |e| \leq 2 \cdot |e|_b$$

for all expressions e. Therefore, it suffices to compute an upper bound for the basic size of the resulting expression. The basic size of expression e is now split into $|e|_b = |e|_r + |e|_s$ where $|e|_s$ is the sum of the basic sizes of all simple subexpressions of e, and $|e|_r$ is just the rest. Let us call $|e|_r$ the *reduced* basic size of e. Observe that each local improvement step adds a new simple subexpression but leaves the reduced basic size invariant. Therefore, it suffices to prove that the basic sizes of all new simple subexpressions created by the transformation starting from the initial expression $\bar{e}$ sum up to at most $r \cdot |\bar{e}|_b$.

Creating new simple expressions is done by copying certain parts of the current candidate. Therefore, instead of measuring their sizes directly, we as well may count for every node in $\bar{e}$ of arity 0 or 1 how often it is possibly copied into a new simple expression. We claim:

Claim: If node a has level r then a gives rise to at most r nodes in simple subexpressions of the result.

For a proof of this claim, consider candidate e at level $r \geq 1$ inside expression $\bar{e}$ which, in one improvement step, is replaced by $e_1 \sqcup e_2$ according to our definition above. Let us first consider the case $r = 1$. By construction, node a in e gives rise to a node in simple expression e_1 iff a is *not* contained in a proper closed subexpression of e. Thus, a has level 1 or 2 (w.r.t. $\bar{e}$). Let us consider what happens to a in the residual expression e_2 *after* the improvement step. According to assertion 2 above, a receives level 1 and either is the root of a closed least fixed-point expression or has a hook which is a least fixed-point expression of level 1. Therefore, the residual of a in e_2 will never be copied into a simple expression again.

Now assume $r > 1$. By construction, node a in e gives rise to a node in simple expression e_1 iff a is *not* contained in a subexpression of level $< r$. Therefore, a has level r or $r + 1$ (w.r.t. $\bar{e}$). After application of the improvement step, a has level at most $r - 1$ (see assertion 3). Therefore, the level of (the residual of) a in subexpression e_2 through the improvement step has been decreased at least by 1.

We conclude that any node gives rise to at most r nodes of simple expressions created by our transformation.

Remark. By fact 2.1, the above result can be easily extended to the case where the expressions involve function symbols of arbitrary finite arity, provided that the interpretation $g^D : D^k \to D$ of a k-ary symbol g is distributive over D^k.

Theorem 4.3 results in a quantitative version of the fact that fragment L_2 of the propositional μ-calculus (interpreted over finite or infinite transition systems) is no more expressive than its alternation-depth-2 fragment.

If $\Phi(m)$ is the complexity of evaluating expressions of size at most m and alternation-depth ≤ 2 in D with co-continuous "$\sqcup$" and distributive operators,

this results in a procedure evaluating expressions of alternation-depth $r > 0$. We obtain:

Corollary 4.5. *Given a complete lattice D with co-continuous "$\sqcup$" and distributive operators, every closed fixed-point expression e of alternation-depth $r > 0$ can be evaluated in time $\mathcal{O}(r \cdot \Phi(|e|))$.*

In the case of finite-state model-checking, Corollary 4.5 recovers the complexity bounds given by Bhatt and Cleaveland in [4] from the methods of Andersen and Vergauwen [1] for alternation-depth 2.

5. Optimality

In terminology of [16] Theorem 4.3 implies that the alternation-depth hierarchy of fixed-point expressions collapses for lattices D with co-continuous "$\sqcup$" and distributive operators not only at alternation-depth 2 but, more precisely, at the class $\nu\mu$. One may wonder whether this result could be even further improved upon, *e.g.*, whether or not class $\mu\nu$ might be sufficient as well. Formally, expression e is in class $\mu\nu$ iff all μ-variables are at level 1. For such expressions, however, the transformation of Theorem 4.3 reveals the following.

Fact 5.1. *Every fixed-point expression e in $\mu\nu$ is equivalent to an expression e' with the following properties:*

1. *e' is alternation-free;*
2. $|e'| = \mathcal{O}(|e|)$.

Thus, class $\mu\nu$ is no more expressive than alternation-free fixed-point expressions.

Example 5.2. Consider expression

$$e \equiv \mu x.\nu y.f\, y \sqcup (\mu z.g\, z \sqcup x)$$

of Example 4.1 which is contained in class $\mu\nu$. Applying the transformation from Theorem 4.3, we obtain

$$\mu x.(\nu y.f\, y) \sqcup (\mu y.f\, y \sqcup (\mu z.g\, z \sqcup x))$$

which is alternation-free.

It remains to prove that alternation-depth 2 fixed-point expressions in general are more expressive than alternation-free ones. Let Σ be a finite alphabet and let $D_\Sigma = 2^{\Sigma^\omega}$ be the complete lattice of all subsets of infinite words over Σ, ordered by subset inclusion. Let $\mathbf{C} = \emptyset$ and $\mathbf{F} = \Sigma$. We interpret a symbol $\sigma \in \Sigma$, as left-multiplication with σ, *i.e.*, the operation $\lambda L \in D_\Sigma.\sigma L$.

Example 5.3. Consider expression

$$e \equiv \nu x_2.\mu x_1.1\, x_1 \sqcup 2\, x_2$$

of alternation-depth 2 (more precisely, of class $\nu\mu$). When interpreted over D_Σ (where $\Sigma = \{1,2\}$), expression e denotes the language

$$[\![e]\!]\,\emptyset = [\![\nu\, x_2.1^*(2\, x_2)]\!]\,\emptyset = (\{1\}^* \cdot \{2\})^\omega$$

i.e., the set of all infinite words containing infinitely many 2.

The following fact belongs to the "folklore" of the fixed–point community. It was discovered by Park and Tiuryn around 1980 and gave an impact to the subsequent study of fixed–point hierarchy. The result was communicated to the second author by Tiuryn as early as in 1981. (It is suggested in the discussion of fair–merge problem in [17], however we do not know any place where the result is explicitly stated.)

Theorem 5.4 (Park and Tiuryn). *The language $(\{1\}^* \cdot \{2\})^\omega$ of Example 5.3 is not definable over D_Σ by an alternation–free expression. Consequently, alternation-depth-2 fixed-point expressions (from the class $\nu\mu$) are strictly more expressive than alternation-free fixed-point expressions.*

The original proof used a topological argument. Consider the Cantor topology on $\{1,2\}^\omega$ (*cf.* [20]). Then it can be easily shown that any language definable by an alternation–free expression (in fact, even by $\mu\nu$ expression) is in the class F_σ (countable unions of closed sets), while the set $(\{1\}^* \cdot \{2\})^\omega$ of Example 5.3 is not (precisely, it is in $G_\delta - F_\sigma$).

Another argument can be given on the basis of the following fact, which is not hard to prove, and also can be inferred from a more general characterization of the alternation–free μ–calculus in terms of *weak* alternating automata of Muller *et al.* [3, 12].

Fact 5.5. *For $L \subseteq \Sigma^\omega$ the following statements are equivalent:*

1. *$L = [\![e]\!]\,\emptyset$ for some closed alternation-free fixed-point expression e over D_Σ.*
2. *L can be accepted by a non-deterministic finite Büchi automaton where each strong component* either *consists of accepting states* or *of non-accepting states.*

It is easy to show that the language $(\{1\}^* \cdot \{2\})^\omega$ cannot be recognized by any automaton satisfying the condition 2 of Fact 5.5.

A generalization of the expression in Example 5.3 provides us with a witness to show that the size of the $\nu\mu$-level expression stated in Theorem 4.3 is optimal (up to a constant factor).

Let $\Sigma_r = \{0, 1, 2, \ldots, r\}$, where $r = 2k$. Let $L_{r,0} \subseteq (\Sigma_r \backslash \{0\})^\omega$ be the set of infinite words u such that $\limsup_{j\to\infty} u(j)$ is even, in other words, the highest number occurring infinitely often in u is even. (Here and further we identify an infinite word u with the sequence of its letters $u(0), u(1), \ldots$) Finally, let $L_{r,n}$ denote the language of infinite words obtained from $L_{r,0}$ by inserting 0^n before every letter $s > 0$. It is well known that $L_{r,n}$ can be defined by a fixed-point expression

$$e_{r,n} \equiv \nu x_r.\mu x_{r-1}. \ldots \nu x_2.\mu x_1.0^n\,(1\, x_1 \sqcup \ldots \sqcup r\, x_r)$$

(see, *e.g.*, [16], Sect. 5). Note that the above expression has alternation-depth r and size $\mathcal{O}(n+r)$. Hence Theorem 4.3 gives us an equivalent $\nu\mu$ expression of size $\mathcal{O}(r \cdot (n+r))$. We will argue that this size is indeed required. Please note that this is slightly more than just to say that the estimation in Theorem 4.3, given in terms of two parameters r and $|e|$, is exact. The example shows that the size factor $|e|$ matters also for fixed r – and *vice versa*.

Theorem 5.6. *Every fixed-point expression for $L_{r,n}$ of class $\nu\mu$ has size $\Omega(r \cdot (n+r))$.*

Proof. In order to prove the claim, we will use the correspondence between $\nu\mu$ expressions and Büchi automata. A translation of $\nu\mu$ expressions over D_Σ into Büchi automata was presented already by Park [17]. A construction described in [16] translates a fixed-point expression e (with arbitrary nesting of μ and ν) into a Rabin automaton with no more than $|e|$ states, and if e is of class $\nu\mu$, the result is a Büchi automaton.

However, the number of states is not a convenient measure for us here, and neither would be the number of transitions. Instead, we introduce another simple concept.

Let A be a Büchi automaton with the set of states Q and the transition table given by relation $Q \times \Sigma \times Q$. An *arrow* of A is any pair (σ, p) such that, for some $q \in Q$, the triple (q, σ, p) is a transition of A. It is plain to see that the construction of [16] translates a fixed-point expression e into an automaton with no more than $|e| + |\Sigma|$ arrows. The component $|\Sigma|$ is required because the resulting automaton may need a subprogram recognizing the set of *all* infinite words over Σ which in fixed-point expression is described by a short expression $\nu x.x$. Clearly, one such subprogram suffices, and it costs Σ arrows.

In our case, this means that any $\nu\mu$ expression e defining $L_{r,n}$ is equivalent to a Büchi automaton with no more than $|e| + r$ arrows. On the other hand, we will show that any nondeterministic Büchi automaton recognizing $L_{r,n}$ must have at least $k \cdot (n + \frac{k+1}{2})$ arrows, for $r = 2k$. From there, the postulated lower bound for $|e|$ will clearly follow.

Let A be a nondeterministic Büchi automaton recognizing $L_{r,n}$. We may assume $r \geq 2$. Our argument is based on two claims.

Claim 1. For each $i = 1, 2, \ldots, k$, there are $k + 1 - i$ transitions $(p_i, 2i, q_i)$, $(p_{i+1}, 2i, q_{i+1})$, $\ldots$, $(p_k, 2i, q_k)$, such that the states $q_i, q_{i+1}, \ldots, q_k$ are distinct. Consequently, there are *at least* $k + (k-1) + \ldots + 2 + 1 = \frac{k \cdot (k+1)}{2}$ arrows of the form (s, q), $s > 0$.

Claim 2. Let $(2, q_1), (2, q_2), \ldots, (2, q_k)$ be distinct arrows that exist by Claim 1 for the letter $s = 2$. Then there are distinct states $q_{j,m}$, for $j = 1, 2, \ldots, k$, $m = 1, \ldots, n$, such that the automaton has transitions $(q_j, 0, q_{j,1})$, and $(q_{j,m-1}, 0, q_{j,m})$, for $j = 1, 2, \ldots, k$, and $m = 1, \ldots, n$. Consequently, there are at least $k \cdot n$ arrows of the form $(0, q)$.

Together Claims 1 and 2 give us the required lower bound for the number of arrows in an automaton recognizing $L_{r,n}$.

Proof of Claim 1. Fix a constant M greater than the number of states of A. Recall that a run of a Büchi automaton on an infinite word u can be presented as an infinite word, r say, over the automaton's states, such that $(r(t), u(t), r(t+1))$ is a transition, for $t = 0, 1, \ldots$ A run is accepting if some of (especially designated) accepting states occur infinitely often [20].

We first give an idea of the proof by showing that if $r \geq 6$ then there must be at least *three* different arrows of the form $(2, p)$. In what follows, we let for simplicity $n = 0$. Experienced readers can skip this part and go directly to the *General case* discussed below.

Consider infinite words

$$u = 12\,12\,12\,12\ldots$$

and

$$v = (123)^M\,4\,(123)^M\,4\ldots$$

Fix some accepting runs of A on u and v, say r_u and r_v, respectively. Pick a state p such that, for infinitely many t's it holds that $r_u(t+1) = p$ while $u(t) = 2$ (we say that p occurs *after reading* 2). Next, by construction of v, we can find positions $i \leq \ell < \ell + 1 < j \leq 3M$, such that $r_v(i) = r_v(j)$, $v(\ell) = 2$, and $v(\ell+1) = 3$. We claim that $q = r_v(\ell+1)$ must be different from p, and consequently the arrows $(2, p)$ and $(2, q)$ are different. The intuitive reason for this is that, in the state q, the automaton waits for 4, while in the state p it is satisfied with having seen 2. To prove the claim, suppose the contrary. We can find positions $t_1 < t_2$ in the run r_u such that $r_u(t_1) = r_u(t_2) = p$, and some accepting state occurs between t_1 and t_2. Using the assumption that $p = q$, we can substitute to the position $\ell + 1$ in the run r_v, the segment cut out of r_u by the positions t_1 and t_2. Note that in the resulting run (and the corresponding infinite word), we have two positions (specifically, i and $j + (t_2 - t_1)$) with the same state, and such that an accepting state occurs between these positions while the highest number in underlying word is 3. Then, we can pump this segment infinitely many times, and eventually obtain an accepting run of A on an infinite word with lim sup equal to 3, a contradiction.

Note that the same argument would apply if, instead of selecting the positions i, ℓ, j in the first block of the run r_v, (*i.e.* $\leq 3M$), we would select them in any other block between $m \cdot (3M+1)$ and $(m+1) \cdot (3M+1) - 1$. Now, by counting argument, we can find a state q such that, for infinitely many positions ℓ in the run r_v, it happens that $v(\ell) = 2$, $r_v(\ell+1) = q$, and there are two positions within the same block $(123)^M$, say i_ℓ, j_ℓ, such that $i_\ell \leq \ell < \ell + 1 < j_\ell$, and $r_v(i_\ell) = r_v(j_\ell)$. By remark above, we may conclude that $q \neq p$.

Now let

$$w = (12345)^M\,6\,(12345)^M\,6\ldots$$

and let r_w be an accepting run on w.

We claim that a state different from p and q must occur in the run r_w after reading some occurrence of 2. Intuitively, in this third state the automaton waits for 6. To prove this claim, we can apply to w the trick previously applied to v. That is, we select positions $i \leq \ell < \ell + 3 < j \leq 5M$, such that $r_w(i) = r_w(j)$, $w(\ell) = 2$, and $w(\ell + 3) = 5$. Now suppose that $r_w(\ell + 1)$ is equal to p or to q. Then, by substituting at the position ℓ a part of the run r_u or r_v, respectively, containing an accepting state, we will be able to construct accepting runs on words with lim sup equal to 5.

General case. Let A be a Büchi automaton recognizing $L_{r,n}$, $r = 2k$, and let M be a constant greater than the number of states of A. Let $i \in \{1, 2, \ldots, k\}$. Recall that according to the claim, we wish to find $k + 1 - i$ different arrows of the form $(2i, p)$. For $j = i, i + 1, \ldots, k$, let

$$u_j = ((0^n 1 0^n 2 \ldots 0^n (2j-1))^M (2j))^\omega$$

and let r_j be an accepting run of A on u_j. Now, by counting argument, for each $j = i, i+1, \ldots, k$ we can find a state, say q_j, such that, for infinitely many positions ℓ in the run r_j, we have $u_j(\ell) = 2i$, $r_j(\ell + 1) = q_j$, and moreover there exist two positions i_ℓ and j_ℓ, such that $i_\ell \leq \ell \leq \ell + n < j_\ell$, $r_j(i_\ell) = r_j(j_\ell)$, and the highest digit occurring in the underlying finite segment of the word u_j, that is, in the segment $u_j(i_\ell)u_j(i_\ell + 1) \ldots u_j(j_\ell - 2)u_j(j_\ell - 1)$, is precisely $2j - 1$, whenever $j > i$, and it is (obviously) $2i = 2j$, for $j = i$. (The requirement $\ell + n < j_\ell$ will be used later on in the proof of *Claim 2*. At present we need only the inequality $\ell < j_\ell$.)

We claim that the states $q_i, q_{i+1}, \ldots, q_k$ are distinct, which will give us $k+1-i$ different arrows $(2i, q_i)$, $(2i, q_{i+1})$, $\ldots$ $(2i, q_k)$, as desired.

Suppose to the contrary that there are some $i \leq \alpha < \beta \leq k$ such that $q_\alpha = q_\beta$. Since q_α occurs infinitely often in r_α, we can find two positions $t_1 < t_2$ such that $r_\alpha(t_1) = r_\alpha(t_2) = q_\alpha$, and the segment $r_\alpha(t_1)r_\alpha(t_1 + 1) \ldots r_\alpha(t_2 - 1)r_\alpha(t_2)$ of r_α contains an occurrence of an accepting state of A. Let v be the underlying finite segment of the word u_α, *i.e.*,

$$v = u_\alpha(t_1)u_\alpha(t_1 + 1) \ldots u_\alpha(t_2 - 2)u_\alpha(t_2 - 1).$$

Note that the highest digit that may occur in v is not greater than $\limsup_{t \to \infty} u_\alpha = 2\alpha < 2\beta - 1$.

Now, by the choice of q_β, we can find positions $i_\ell \leq \ell < j_\ell$ in the run r_β, such that $u_\beta(\ell) = 2i$, $r_\beta(\ell + 1) = q_\beta$, $r_\beta(i_\ell) = r_\beta(j_\ell)$, and the highest digit occurring in the underlying segment of u_β is $2\beta - 1$ (recall that $i < \beta$). Let us decompose this segment by $w_1 w_2$, where

$$\begin{aligned} w_1 &= u_\beta(i_\ell)u_\beta(i_\ell + 1) \ldots u_\beta(\ell) \\ w_2 &= u_\beta(\ell + 1)u_\beta(\ell + 2) \ldots u_\beta(j_\ell - 1). \end{aligned}$$

Let

$$u = u_\beta(0)u_\beta(1) \ldots u_\beta(i_\ell - 1).$$

Finally, consider the infinite word

$$u_\infty \quad = \quad u(w_1 v w_2)^\omega.$$

Clearly $\limsup_{t\to\infty} u_\infty(t) = 2\beta - 1$, so this word should not be accepted. Now, assuming $q_\alpha = q_\beta$, we can easily get an accepting run on u_∞ by "cut and past" construction. Specifically, we can take the infinite word

$$r_\beta(0)\dots r_\beta(i_\ell - 1)(r_\beta(i_\ell)\dots r_\beta(\ell)\, r_\alpha(t_1)\dots r_\alpha(t_2)\, r_\beta(\ell+1)\dots r_\beta(j_\ell - 1))^\omega.$$

This contradiction completes the proof of Claim 1.

Proof of Claim 2. We may assume $n > 0$. We will use concepts introduced in the proof of Claim 1. Let $u_1, u_2, \dots, u_k$ be the infinite words defined for $i = 1$. Let $r_1, \dots, r_k$ and $q_1, \dots, q_k$ be as above. Recall that for each q_j, there are infinitely many positions ℓ in the run r_j, such that $u_j(\ell) = 2 \cdot 1 = 2$, $r_j(\ell+1) = q_j$, and moreover there exist two positions i_ℓ and j_ℓ, such that $i_\ell \le \ell < \ell + n < j_\ell$, $r_j(i_\ell) = r_j(j_\ell)$, and, whenever $j > 1$, the highest digit occurring in the underlying finite segment of the word u_j, that is, in the segment $u_j(i_\ell)u_j(i_\ell+1)\dots u_j(j_\ell - 2)u_j(j_\ell - 1)$, is precisely $2j - 1$. Let, for $j = 1, \dots, k$, $P_j \subseteq \omega$ be the set of all positions ℓ with above property.

Now, by counting argument, we can additionally find states $q_{j,m}$, for $j = 1, 2, \dots, k$, $m = 1, \dots, n$, such that the automaton has transitions $(q_j, 0, q_{j,1})$, and $(q_{j,m-1}, 0, q_{j,m})$, for $j = 1, 2, \dots, k$, and $m = 2, 3, \dots, n$, and moreover, for each $j = 1, 2, \dots, k$, the segment $q_j q_{j,1} q_{j,2} \dots q_{j,n}$ repeats infinitely often in the run r_j starting from some position ℓ in P_j (that is, $r_j(\ell+1) = q_j$, $r_j(\ell+2) = q_{j,1}$, $\dots$, $r_j(\ell+1+n) = q_{j,n}$).

We claim that $q_{j,m} = q_{j',m'}$ only if $j = j'$ and $m = m'$. Indeed, if there were $q_{j,m} = q_{j',m'}$ with $m \neq m'$ then we could easily construct an accepting run over an infinite word having a block of 0's of length different than n between two digits different from 0, an obvious contradiction. Now suppose $q_{\alpha,m} = q_{\beta,m}$, for some $1 \le m \le n$, and $1 \le \alpha < \beta \le k$. We will arrive at the contradiction along the same line as in the proof of Claim 1. (Intuitively, our hypothesis means that, although $q_\alpha \neq q_\beta$, the automaton may forget this difference after m steps.) Indeed, since the block $q_\alpha q_{\alpha,1} q_{\alpha,2} \dots q_{\alpha,n}$ repeats infinitely often in the run r_α starting from some position ℓ in P_α, we can find two positions $t_1 < t_2$ in P_α at which this block occurs, such that moreover the segment $r_\alpha(t_1 + 1 + m) \dots r_\alpha(t_2 + 1 + m)$ of the run r_α contains an occurrence of an accepting state.

Let v be the underlying finite segment of the word u_α, *i.e.*, $v = u_\alpha(t_1 + 1 + m) \dots u_\alpha(t_2 + m)$. As before, we note that the highest digit occurring in v is not greater than $\limsup_{t\to\infty} u_\alpha = 2\alpha < 2\beta - 1$.

Now, by the choice of the block $q_\beta q_{\beta,1} q_{\beta,2} \dots q_{\beta,n}$, we can find a position ℓ in P_β at which this block occurs. That is, we have two positions $i_\ell \le \ell < \ell + n < j_\ell$ such that $r_\beta(i_\ell) = r_\beta(j_\ell)$, and the highest digit in the underlying segment of u_β is $2\beta - 1$. Note that the block $q_\beta q_{\beta,1} q_{\beta,2} \dots q_{\beta,n}$ occurs in r_β between the positions

ℓ and j_ℓ; in particular we have $\ell + 1 + m \leq j_\ell$. We will now use the fact that $r_\beta(\ell + 1 + m) = q_{\beta,m} = q_{\alpha,m}$ in order to "fool" the automaton, similarly as in the proof of Claim 1. That is, we let

$$\begin{aligned} w_1 &= u_\beta(i_\ell)u_\beta(i_\ell + 1)\dots u_\beta(\ell)\dots u_\beta(\ell + 1 + m) \\ w_2 &= u_\beta(\ell + 1 + m + 1)\dots u_\beta(j_\ell - 1) \end{aligned}$$

(not harmful if $w_2 = \epsilon$), and

$$u = u_\beta(0)u_\beta(1)\dots u_\beta(i_\ell - 1)$$

and consider the infinite word

$$u_\infty = u(w_1 v w_2)^\omega.$$

Again, we have $\limsup_{t\to\infty} u_\infty(t) = 2\beta - 1$, and so this word should not be accepted. On the other hand, whenever an automaton enters in v in the state $q_{\beta,m} = q_{\alpha,m}$, it may leave this word in the same state passing by an accepting state. Therefore, by using a "cut and past" construction, we can obtain an accepting run on u_∞ similarly as in the proof of Claim 1.

This remark completes the proof. □

6. Conclusion

We have presented a transformation on fixed-point expressions which allows to reduce the alternation-depth to 2 in case all operators are distributive and binary "⊔" is co-continuous. More precisely, we have proved that under this proviso the alternation-depth hierarchy collapses at the level $\nu\mu$. We have also shown that our transformation is not only optimal w.r.t. the resulting alternation-depth, but it is also optimal (up to a constant factor) w.r.t. the size of the transformed expression.

We like to thank André Arnold for many useful comments and Thomas Wilke for interesting remarks on classes of Büchi automata.

References

[1] H.R. Andersen and B. Vergauwen, Efficient Checking of Behavioral Relations and Modal Assertions Using Fixed-Point Inversion. In *7th International Conference on Computer-Aided Verification (CAV)*. Springer, *Lecture Notes in Comput. Sci.* **939** (1995) 142–154.

[2] A. Arnold, The μ-Calculus Alternation-Depth Hierarchy is Strict on Binary Trees. *Theoret. Informatics. Appl.*, Special issue on FICS'98, to appear.

[3] A. Arnold and D. Niwiński, Fixed Point Characterization of Weak Monadic Logic Definable Sets of Trees, M. Nivat and A. Podelski, Eds. Elsevier, Amsterdam, *Tree Automata and Languages* (1992) 159–188.

[4] G. Bhat and R. Cleaveland, Efficient Local Model Checking for Fragments of the Modal μ-Calculus. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. Springer, *Lecture Notes in Comput. Sci.* **1055** (1996) 107–126.

[5] J.C. Bradfield, The Modal Mu-Calculus Alternation Hierarchy is Strict. In *7th International Conference on Concurrency Theory (CONCUR)*. Springer, *Lecture Notes in Comput. Sci.* **1119** (1986) 233–246.

[6] H. Doornbos, R. Backhouse and J. van der Woude, A Calculational Approach to Mathematical Induction. *Theoret. Comput. Sci.* **179** (1997) 103–135.

[7] E.A. Emerson, C.S. Jutla and A.P. Sistla, On Model-Checking for Fragments of μ-Calculus. In *5th International Conference on Computer-Aided Verification (CAV)*. Springer, *Lecture Notes in Comput. Sci.* **697** (1993) 385–396.

[8] E.A. Emerson and E.M. Clarke, Characterizing Correctness Properties of Parallel Programs Using Fixpoints. In *7th International Colloquium on Automata, Languages and Programming (ICALP)*. Springer, *Lecture Notes in Comput. Sci.* **85** (1980) 169–181.

[9] L. Flon and N. Suzuki, Consistent and Complete Proof Rules for the Total Correctness of Parallel Programs. In *19th IEEE Symp. on Foundations of Computer Science (FOCS)* (1978).

[10] M. Jurdziński, Deciding the Winnner in Parity Games is in UP $\cap$ co-UP. *Inform. Process. Lett.* **68** (1998) 119–124.

[11] G. Lenzi, A Hierarchy Theorem for the μ–Calculus. In *23rd International Colloquium on Automata, Languages and Programming (ICALP)*. Springer, *Lecture Notes in Comput. Sci.* **1099** (1996) 87–109.

[12] D.E. Muller, A. Saoudi and P.E. Schupp, Alternating Automata, the Weak Monadic Theory of the Tree and its Complexity. In *ICALP'86* (1986).

[13] D. Niwiński, On Fixed Point Clones. In *13th International Colloquium on Automata, Languages and Programming (ICALP)*. Springer, *Lecture Notes in Comput. Sci.* **226** (1986) 464–473.

[14] D. Niwiński, *Hierarchy of Objects Definable in the Fixed Point Calculus,* in Polish. Ph.D. Thesis, University of Warsaw (1987).

[15] D. Niwiński, Fixed Points *vs.* Infinite Generation. In *3rd Annual IEEE Symposium on Logic in Computer Science (LICS)*, IEEE (1988) 402–409.

[16] D. Niwiński, Fixed Point Characterization of Infinite Behavior of Finite State Systems. *Theoret. Comput. Sci.* **189** (1997) 1–69.

[17] D.M.R. Park, On the Semantics of Fair Parallelism. In *Abstract Software Specification*. Springer, *Lecture Notes in Comput. Sci.* **86** (1980) 504–526.

[18] D.M.R. Park, Concurrency and Automata on Infinite Sequences. In *Theoret. Comput. Sci.* Springer, *Lecture Notes in Comput. Sci.* **104** (1981) 167–183.

[19] H. Rasiowa and R. Sikorski, *Mathematics of Metamathematics*. Państowe Wydawnictwo Naukowe (1970).

[20] W. Thomas, Automata on Infinite Objects, J. van Leeuwen, Ed., *Handbook of Theoretical Computer Science*. Elsevier, Amsterdam (1990).

Received November 25, 1998. Revised April 22, 1999.