# Fixpoint Games on Continuous Lattices

(full version)

PAOLO BALDAN, Università di Padova, Italy
BARBARA KÖNIG, Universität Duisburg-Essen, Germany
CHRISTINA MIKA-MICHALSKI, Universität Duisburg-Essen, Germany
TOMMASO PADOAN, Università di Padova, Italy

Many analysis and verifications tasks, such as static program analyses and model-checking for temporal logics, reduce to the solution of systems of equations over suitable lattices. Inspired by recent work on lattice-theoretic progress measures, we develop a game-theoretical approach to the solution of systems of monotone equations over lattices, where for each single equation either the least or greatest solution is taken. A simple parity game, referred to as fixpoint game, is defined that provides a correct and complete characterisation of the solution of systems of equations over continuous lattices, a quite general class of lattices widely used in semantics. For powerset lattices the fixpoint game is intimately connected with classical parity games for $\mu$-calculus model-checking, whose solution can exploit as a key tool Jurdziński's small progress measures. We show how the notion of progress measure can be naturally generalised to fixpoint games over continuous lattices and we prove the existence of small progress measures. Our results lead to a constructive formulation of progress measures as (least) fixpoints. We refine this characterisation by introducing the notion of selection that allows one to constrain the plays in the parity game, enabling an effective (and possibly efficient) solution of the game, and thus of the associated verification problem. We also propose a logic for specifying the moves of the existential player that can be used to systematically derive simplified equations for efficiently computing progress measures. We discuss potential applications to the model-checking of latticed $\mu$-calculi and to the solution of fixpoint equations systems over the reals.

Additional Key Words and Phrases: fixpoint equation systems, continuous lattices, parity games, $\mu$-calculus

## 1 INTRODUCTION

Systems of fixpoint equations are ubiquitous in formal analysis and verification. For instance, program analysis [Nielson et al. 1999] uses the flow graph of a program to generate a set of constraints specifying how the information of interest at the different program points is interrelated. The set of constraints can be viewed as a system of fixpoint equations, whose (least or greatest) solution provides a sound approximation of the properties of the program. Invariant/safety properties can be characterised as greatest fixpoints, while liveness/reachability properties as least fixpoints. Behavioural equivalences (for instance for process calculi) are typically defined as the solution of a fixpoint equation. The most famous example is bisimilarity that can be characterised as the greatest fixpoint of a suitable operator over the lattice of binary relations on the set of states (see, e.g., [Sangiorgi 2011]).

Almost invariably, in the mentioned settings, the involved functions are monotone and the domains of interest are complete lattices where the key result for deriving the existence of (least or greatest) fixpoints is Knaster-Tarski's fixpoint theorem [Tarski 1955].

Least and greatest fixpoint can be profitably mixed, in order to obtain expressive specification logics, among which the $\mu$-calculus [Kozen 1983] is a classical example. The $\mu$-calculus is very

Authors' addresses: Paolo Baldan, Dipartimento di Matematica "Tullio Levi-Civita", Università di Padova, Via Trieste, 63, Padova, I-35121, Italy, baldan@math.unipd.it; Barbara König, Fakultät für Ingenieurwissenschaften, Abteilung Informatik und Angewandte Kognitionswissenschaft, Universität Duisburg-Essen, Lotharstraße 65, Duisburg, 47048, Germany, barbara_koenig@uni-due.de; Christina Mika-Michalski, Fakultät für Ingenieurwissenschaften, Abteilung Informatik und Angewandte Kognitionswissenschaft, Universität Duisburg-Essen, Lotharstraße 65, Duisburg, 47048, Germany, christina.mika-michalski@uni-due.de; Tommaso Padoan, Dipartimento di Matematica "Tullio Levi-Civita", Università di Padova, Via Trieste, 63, Padova, I-35121, Italy, padoan@math.unipd.it.

expressive, but the nesting of fixpoints increases the complexity of model-checking. Common approaches to the model-checking problem rely on an encoding in terms of parity games (see, e.g., [Bradfield and Walukiewicz 2018; Emerson and Jutla 1991; Stirling 1995]). The seminal paper [Jurdziński 2000] provides an algorithm for the solution of parity games which is polynomial in the number of states and exponential in (half of) the alternation depth, recently improved to quasi-polynomial in [Calude et al. 2017]. A detailed discussion of the complexity of $\mu$-calculus model-checking can be found in [Bradfield and Walukiewicz 2018].

It has been recently observed in [Hasuo et al. 2016] that progress measures, a key ingredient in Jurdzisńki's algorithm for solving parity games, are amenable to a generalisation to systems of fixpoint equations over general lattices. A constructive characterisation of such progress measures is given in the case of powerset lattices and used to derive model-checking procedures for (branching and linear) coalgebraic logic. For general lattices, however, the notion of progress measure in [Hasuo et al. 2016] does not exactly correspond to Jurdzinski's notion. In particular, there is no algorithm for actually computing such progress measures, they rather play the role of invariants respectively ranking functions that have somehow to be provided. While the possibility of deriving generic algorithms for solving systems of equations is very appealing, the restriction to powerset lattices limits the applicability of the technique. Often program analysis relies on lattices which are not powerset lattices (and neither distributive, hence they cannot be seen as sublattices of powerset lattices). Moreover also settings involving fuzziness, probabilities or in general quantitative information are not captured by restricting to powerset lattices.

Inspired by the mentioned work, in this paper we devise a game-theoretical approach to the solution of systems of fixpoint equations over a vast class of lattices, the so-called continuous lattices. Originally studied by Scott in connection with the semantics of the $\lambda$-calculus [Scott 1972], they have later been recognised as a fundamental structure, with a plethora of applications in the semantics of programming languages and, more generally, in the theory of computation [Abramsky and Jung 1994; Gierz et al. 2003]. They include discrete structures, such as most domains used in program analysis, and continuous structures, such as the real interval $[0, 1]$ or the lattice of open sets of a locally compact Hausdorff space.

The possibility of characterising the least or the greatest fixpoint of a (single) monotone function over a powerset lattice in terms of a game between an existential and an universal player is probably folklore and has been observed in [Venema 2008] where the game is referred to as an unfolding game. As a first result, here we show how the unfolding game can be extended to work for a *system* of fixpoint equations over lattices, resulting in a surprisingly simple game that we refer to as a *fixpoint game*. Mixing least and greatest fixpoint equations requires a non-trivial winning condition, which however arises as a natural adaptation to our setting of the one for parity games.

For the simpler case of powerset lattices the interaction between the players in the fixpoint game fundamentally relies on the possibility of testing the presence of elements in the image of a set and on the fact that a subset is completely determined by the elements that belongs to it. When moving to a more general class of lattices we need to ensure that this kind of interaction can be suitably mimicked. We argue in the paper that continuous lattices provide an extremely natural setting for this extension, providing exactly the necessary machinery for stating results in a way which is analogous to the powerset case. In fact, they come equipped with a notion of "finitary" approximation based on the *way-below* relation and each element arises as the join of the elements (possibly restricted to a selected basis) which are way-below it, in the same way as a subset is the union of its singletons.

The proof that our fixpoint game provides a correct and complete characterisation of the solution of a system of fixpoint equations over a continuous lattice relies on $\mu$- and $\nu$-approximants that provide a clear notion of approximation of the solution. In particular, $\mu$-approximants turn out to be closely related to the progress measures of [Hasuo et al. 2016], a connection that we will make precise in Appendix B.

We show how Jurdziński's approach for solving parity games [Jurdziński 2000] can be generalised to systems of fixpoint equations over continuous lattices. In particular we introduce a notion of progress measure for fixpoint games over continuous lattices. Intuitively, given an element $b$ of the basis of the lattice and an equation index $i$, the progress measure provides a vector of ordinals, specifying how many iterations are needed for each equation to cover $b$ in the $i$-th component of the solution. Then we prove the existence of suitably defined small progress measures. This result enables a constructive characterisation of progress measures as (least) fixpoints and provides a recipe for computing the progress measure that can be straightforwardly implemented, at least for finite lattices.

We refine the fixpoint characterisation of progress measures by introducing the notion of selection, which basically constrains the moves of the existential player in the parity game, still preserving correctness and completeness, thus enabling a more efficient solution of the game. We also define a logic for providing a symbolic representation of the moves of the existential player that can be directly translated into a system of fixpoint equations describing the progress measure. In particular, we discuss selections and logical formulae that are needed to handle $\mu$-calculus model-checking.

As an example of application beyond standard $\mu$-calculus model-checking we will discuss the case of latticed $\mu$-calculi, where the evaluation of a formula for a state gives a lattice element, generalising the standard truth values 0, 1 (see, e.g., [Eleftheriou et al. 2012; Grumberg et al. 2005; Kupfermann and Lustig 2007]). This happens naturally also when $\mu$-calculus formulae are evaluated over weighted transition systems or over probabilistic automata [Huth and Kwiatkowska 1997].

The lattice under consideration – and hence its basis – might be infinite and in this case it is not even guaranteed that the fixpoint iteration terminates in $\omega$ steps. In fact, it is known that, despite the functions involved in the equations of the system being continuous, due to alternation of least and greatest fixpoints, discontinuous functions may arise and we possibly have to refer to ordinals beyond $\omega$ [Fontaine 2008; Mio and Simpson 2015]. In order to solve these cases, one has to restrict to a finite part of the lattice, approximate or resort to symbolic representations. We take some preliminary steps in this direction proposing a technique to deal with infinite lattices that is always correct, and complete under certain conditions (for instance, on well-orders without other restrictions, or on the reals, suitably restricting the functions). In particular, inspired by [Mio and Simpson 2015, 2017], we show how the game for solving fixpoint equations over the reals can be encoded into an SMT formula of fixed size, capturing the winning condition of the existential player.

Summing up, our main contributions are the following:

- We propose a game-theoretical characterisation of the solution of systems of fixpoint equations over lattices and we identify continuous lattices as a general and appropriate setting for such theory.
- We develop a theory of progress measures à la Jurdziński in this general framework, with a clear recipe for their computation. This can be seen as a generalisation of the MC progress measures, proposed in [Hasuo et al. 2016] for coalgebraic logics over powerset lattices.
- We devise strategies for the computation of such progress measures based on selections and a logic for the symbolic representation of players' moves, along with a complexity analysis.
- We explicitly discuss two application scenarios, beyond standard $\mu$-calculus over powerset lattices: model-checking of latticed $\mu$-calculi via progress measures and the solution of fixpoint equation systems over the reals via SMT solvers.

We believe that due to the generality of our results, there is the potential for several more interesting applications for different lattices.

The rest of the paper is structured as follows. In § 2 we recap the basics of continuous lattices and introduce some notation that will be used throughout the paper. In § 3 we introduce the systems of fixpoint equations over a lattice, we define their solution and devise a corresponding notion of approximation. In § 4 we present a game-theoretical approach to the solution of a system of equations over a continuous lattice, together with several case studies. In § 5 we introduce the notion of progress measure for (the game associated with) systems of fixpoint equations over a continuous lattice. In § 6 we discuss the application of our framework to the model-checking of latticed $\mu$-calculi. In § 7 we present some results for the solutions of systems of fixpoint equations over infinite lattices, with special focus on real intervals. In § 8 we conclude the paper and outline future research. All proofs, further details on the encoding of $\mu$-calculus formulae into fixpoint equation systems (and vice versa) and a detailed comparison to [Hasuo et al. 2016] can be found in the appendix.

## 2 PRELIMINARIES ON ORDERED STRUCTURES

In this section we provide the basic order theoretic notions that will be used throughout the paper. In particular, we define continuous lattices and we provide some notation about tuples of elements that will be useful for compactly describing the solution of systems of equation.

### 2.1 Lattices

A preordered or partially ordered set $\langle P, \sqsubseteq \rangle$ is often denoted simply as $P$, omitting the (pre)order relation. It is *well-ordered* if every non-empty subset $X \subseteq P$ has a minimum. The *join* and the *meet* of a subset $X \subseteq P$ (if they exist) are denoted $\bigsqcup X$ and $\bigsqcap X$, respectively.

*Definition 2.1 (complete lattice, basis, irreducibles).* A *complete lattice* is a partially ordered set $(L, \sqsubseteq)$ such that each subset $X \subseteq L$ admits a join $\bigsqcup X$ and a meet $\bigsqcap X$. A complete lattice $(L, \sqsubseteq)$ always has a least element $\bot = \bigsqcup \emptyset$ and a greatest element $\top = \bigsqcap \emptyset$. Given an element $l \in L$ we define its upward-closure $\uparrow l = \{l' \mid l' \in L \wedge l \sqsubseteq l'\}$. A *basis* for a lattice is a subset $B_L \subseteq L$ such that for each $l \in L$ it holds that $l = \bigsqcup \{b \in B_L \mid b \sqsubseteq l\}$. An element $l \in L$ is *completely join-irreducible* if whenever $l = \bigsqcup X$ for some $X \subseteq L$ then $l \in X$.

Since all lattices in this paper will be complete, we will often omit the qualification "complete". Similarly, since we are only interested in completely join-irreducible elements we will often omit the qualification "completely". Note that $\bot$ is never an irreducible since $\bot = \bigsqcup \emptyset$ and $\bot \notin \emptyset$.

*Example 2.2.* Three simple examples of lattices, that we will refer to later, are:

- The powerset of any set $X$, ordered by subset inclusion $(2^X, \sqsubseteq)$. Join is union, meet is intersection, top is $X$ and bottom is $\emptyset$. A basis is the set of singletons $B_{2^X} = \{\{x\} \mid x \in X\}$. These are also the the join-irreducible elements. Any set $Y \subseteq X$ with $|Y| > 1$ is not irreducible, since $Y = \bigsqcup_{x \in Y} \{x\}$ but clearly $Y \neq \{x\}$ for any $x \in Y$.
- The real interval $[0, 1]$ with the usual order $\leq$. Join and meet are the sup and inf over real numbers, 0 is bottom and 1 is top. The rationals $\mathbb{Q} \cap (0, 1]$ are a basis. There are no irreducible elements (in fact, for any $x \in [0, 1]$ we have that $x = \bigsqcup \{y \mid y < x\}$ and clearly $x \notin \{y \mid y < x\}$).
- Consider the partial order $W = \mathbb{N} \cup \{\omega, a\}$ depicted in Fig. 1. It is easy to see that it is a lattice. All elements are irreducible apart from the bottom 0 and the top $\omega$. For the latter notice that, e.g., $\omega = \bigsqcup \{1, a\}$.

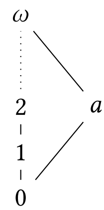



Fig. 1. A complete lattice $W$ which is not continuous.

A lattice is *completely distributive* if

$$\bigsqcup_{k \in K} \bigsqcap_{j \in J_k} l_{k,j} = \bigsqcap \{ \bigsqcup_{k \in K} r_{k,j^k} \mid j^k \in J_k, k \in K \}$$

where $K, J_k, K$ are index sets and $l_{k,j} \in L$.

A function $f : L \longrightarrow L$ is *monotone* if for all $l, l' \in L$, if $l \sqsubseteq l'$ then $f(l) \sqsubseteq f(l')$. By Knaster-Tarski's theorem [Tarski 1955], any monotone function on a complete lattice has a least and a greatest fixpoint, denoted respectively $\mu f$ and $\nu f$, characterised as the meet of all pre-fixpoints respectively the join of all post-fixpoints:

$$\mu f = \bigsqcap \{ l \mid f(l) \sqsubseteq l \} \qquad \nu f = \bigsqcup \{ l \mid l \sqsubseteq f(l) \}$$

The least and greatest fixpoint can also be obtained by iterating the function on the bottom and top elements of the lattice. This is often referred to as Kleene's theorem (at least for continuous functions) and it is one of the pillars of abstract interpretation [Cousot and Cousot 1979]. Given a lattice $L$, define its *height* $\lambda_L$ as the supremum of the length of any strictly ascending, possibly transfinite, chain. Then we have the following result.

Theorem 2.3 (Kleene's iteration [Cousot and Cousot 1979]). *Let $L$ be a lattice and let $f : L \longrightarrow L$ be a monotone function. Consider the (transfinite) ascending chain $(f^{\beta}(\bot))_{\beta}$ where $\beta$ ranges over the ordinals, defined by $f^0(\bot) = \bot$, $f^{\alpha+1}(\bot) = f(f^{\alpha}(\bot))$ for any ordinal $\alpha$ and $f^{\alpha}(\bot) = \bigsqcup_{\beta < \alpha} f^{\beta}(\bot)$ for any limit ordinal $\alpha$. Then $\mu f = f^{\gamma}(\bot)$ for some ordinal $\gamma \leq \lambda_L$. The greatest fixpoint $\nu f$ can be characterised dually, via the (transfinite) descending chain $(f^{\alpha}(\top))_{\alpha}$.*

Note also that $f^{\alpha}(\bot)$ is always a post-fixpoint and $f^{\alpha}(\top)$ is always a pre-fixpoint.

We will focus on special lattices where elements are generated by suitably defined approximations. Given a lattice $L$, a subset $X \subseteq L$ is *directed* if $X \neq \emptyset$ and every pair of elements in $X$ has an upper bound in $X$.

*Definition 2.4 (way-below relation, continuous lattices).* Let $L$ be a lattice. Given two elements $l, l' \in L$ we say that $l$ is *way-below* $l'$, written $l \ll l'$ when for every directed set $D \subseteq L$, if $l' \sqsubseteq \bigsqcup D$ then there exists $d \in D$ such that $l \sqsubseteq d$. We denote by $\twoheaddownarrow l$ the set of elements way-below $l$, i.e., $\twoheaddownarrow l = \{ l' \mid l' \in L \ \wedge \ l' \ll l \}$.

The lattice $L$ is called *continuous* if $l = \bigsqcup \twoheaddownarrow l$ for all $l \in L$.

Intuitively, the way-below relation captures a form of finitary approximation: if one imagines that $\sqsubseteq$ is an order on the information content of the elements, then $x \ll y$ means that whenever $y$ can be "covered" by joining (possibly small) pieces of information, then $x$ is covered by one of those pieces. Then a lattice is continuous if any element can be built by joining its approximations.

Concerning the origin of the name "continuous lattice", we can quote [Scott 1972] that says that "One of the justifications (by euphony at least) of the term *continuous lattice* is the fact that such spaces allow for so many continuous functions." For instance, one indication is the fact that meet and join are both continuous in such lattices.

It can be shown that if $L$ is a continuous lattice and $B_L$ is a basis, for all $l \in L$, it holds that $l = \bigsqcup(\twoheaddownarrow l \cap B_L)$.

Various lattices that are commonly used in semantics enjoy a property stronger than continuity, defined below.

*Definition 2.5 (compact element, algebraic lattice).* Let $L$ be a lattice. An element $l \in L$ is called *compact* whenever $l \ll l$. The lattice $L$ is *algebraic* if the set of compact elements is a basis.

*Example 2.6.* Some examples are as follows:

- All finite lattices are continuous (since every finite directed set has a maximum). More generally, all algebraic lattices (which include all finite lattices) are continuous. The way-below relation is $x \ll y$ if $x$ compact and $x \sqsubseteq y$.
- Given a set $X$, the powerset lattice $\mathbf{2}^X$, ordered by inclusion, is an algebraic lattice. The compact elements are the finite subsets. In fact, any set $Y$ is the union of its finite subsets, i.e., $Y = \bigcup\{F \mid F \subseteq Y \ \land\ F \text{ finite}\}$. Since $\{F \mid F \subseteq Y \ \land\ F \text{ finite}\}$ is directed set, compactness requires that $Y \subseteq F$ for some finite $F \subseteq Y$, hence $Y = F$. Therefore $Y \ll Z$ holds when $Y$ is finite and $Y \subseteq Z$.
- The interval $[0,1]$ with the usual order $\leq$ is a continuous lattice. For $x, y \in [0,1]$, we have $x \ll y$ when $x < y$ or $x = 0$. In fact, each $\emptyset \neq Y \subseteq [0,1]$ is directed. Imagine that $y \leq \bigsqcup Y$ for such a $Y$. Then by standard properties of the reals there always exists a $y' \in Y$ such that $x \leq y'$ if and only if $x < y$ or $x = 0$. Note that this lattice is not algebraic since the only compact element is 0.
- The lattice $W$ in Fig. 1 is not continuous. In fact, $a \not\ll a$ since $a \sqsubseteq \bigsqcup \mathbb{N}$ but $a \not\sqsubseteq i$ for all $i \in \mathbb{N}$. Therefore $\twoheaddownarrow a = \{0\}$ and thus $a \neq \bigsqcup \twoheaddownarrow a$.

## 2.2 Tuples and Ordinals

We will often consider tuples of elements. Given a set $A$, an $n$-tuple in $A^n$ will be denoted by a boldface letter $\boldsymbol{a}$. The components of a tuple $\boldsymbol{a}$ will be denoted by using the same name of the tuple, not in boldface style and with an index, i.e., $\boldsymbol{a} = (a_1, \ldots, a_n)$. For an index $n \in \mathbb{N}$ we use the notation $\underline{n}$ to denote the integer interval $\{1, \ldots, n\}$. Given $\boldsymbol{a} \in A^n$ and $i, j \in \underline{n}$ we write $\boldsymbol{a}_{i,j}$ for the subtuple $(a_i, a_{i+1}, \ldots, a_j)$.

*Definition 2.7 (pointwise order).* Given a lattice $(L, \sqsubseteq)$ we will denote by $(L^n, \sqsubseteq)$ the set of $n$-tuples endowed with the *pointwise order* defined, for $\boldsymbol{l}, \boldsymbol{l'} \in L^n$, by $\boldsymbol{l} \sqsubseteq \boldsymbol{l'}$ if $l_i \sqsubseteq l'_i$ for all $i \in \underline{n}$.

The structure $(L^n, \sqsubseteq)$ is a lattice and it is continuous if $L$ is continuous, with the way-below relation given by $\boldsymbol{l} \ll \boldsymbol{l'}$ iff $l_i \ll l'_i$ for all $i \in \underline{n}$ [Gierz et al. 2003, Proposition I-2.1]. More generally, for any set $X$, the set of functions $L^X = \{f \mid f : X \to L\}$, endowed with pointwise order, is a lattice (continuous when $L$ is).

*Definition 2.8 (lexicographic order).* Given a partial order $(P, \sqsubseteq)$ we will denote by $(P^n, \preceq)$ the set of $n$-tuples endowed with the *lexicographic order* (where the last component is the most relevant), i.e., inductively, for $\boldsymbol{l}, \boldsymbol{l'} \in P^n$, we let $\boldsymbol{l} \preceq \boldsymbol{l'}$ if either $l_n \sqsubset l'_n$ or $l_n = l'_n$ and $\boldsymbol{l}_{1,n-1} \preceq \boldsymbol{l'}_{1,n-1}$.

When $(L, \sqsubseteq)$ is a lattice also $(L^n, \preceq)$ is a lattice. Given a set $X \subseteq L^n$, the meet of $X$ with respect to $\preceq$ can be obtained by taking the meet of the single components, from the last to the first, i.e., it is defined inductively as $\bigsqcap X = \boldsymbol{l}$ where $l_i = \bigsqcap\{l'_i \mid \boldsymbol{l'} \in X \ \land\ \boldsymbol{l'}_{i+1,n} = \boldsymbol{l}_{i+1,n}\}$. The join can be defined analogously. Similarly, one can show that $\preceq$ is a well-order whenever $\sqsubseteq$ is.

As in [Hasuo et al. 2016; Jurdziński 2000], we will also need to consider tuples with a preorder arising from the lexicographic order, when some components are considered irrelevant.

*Definition 2.9 (truncated lexicographic order).* Let $(P, \sqsubseteq)$ be a partial order and let $n \in \mathbb{N}$. For $i \in \underline{n}$ we define a preorder $\preceq_i$ on $P^n$ by letting, for $\boldsymbol{x}, \boldsymbol{y} \in P^n$, $\boldsymbol{x} \preceq_i \boldsymbol{y}$ if $\boldsymbol{x}_{i,n} \preceq \boldsymbol{y}_{i,n}$. We write $=_i$ for the equivalence induced by $\preceq_i$ and $\boldsymbol{x} \prec_i \boldsymbol{y}$ for $\boldsymbol{x} \preceq_i \boldsymbol{y}$ and $\boldsymbol{x} \neq_i \boldsymbol{y}$. Whenever $\sqsubseteq$ is a well-order, given $X \subseteq P^n$ with $X \neq \emptyset$ and $i \in \underline{n}$, we write $\min_{\preceq_i} X$ for the vector $\boldsymbol{x} = (\bot, \dots, \bot, x_i, \dots, x_n)$ where $\boldsymbol{x}_{i,n} = \min_{\preceq}\{\boldsymbol{l}_{i,n} \mid \boldsymbol{l} \in X\}$.

In words, $\preceq_i$ is the lexicographic order restricted to the components $i, i+1, \dots, n$. For instance, if $P = \mathbb{N}$ with the usual order, then $(6, 1, 4, 7) \prec_2 (5, 2, 4, 7)$, while $(6, 1, 4, 7) =_3 (5, 2, 4, 7)$.

We denote *ordinals* by Greek letters $\alpha, \beta, \gamma, \dots$ and their order by $\leq$. The collection of all ordinals is well-ordered. Given any ordinal $\alpha$, the collection of ordinals dominated by $\alpha$ is a set $[\alpha] = \{\lambda \mid \lambda \leq \alpha\}$, which, seen as an ordered structure, is a lattice. Meet and join of a set $X$ of ordinals will be denoted by $\inf X$ (which equals $\min X$ if $X \neq \emptyset$) and $\sup X$. The lattice $[\alpha]$ is completely distributive, which follows from classical results. In fact, the complete join-irreducibles are all ordinals which are not limit ordinals. Hence, from [Raney 1952, Theorems 1 and 2], since every element is the join of completely join-irreducible elements, we can conclude that $[\alpha]$ is completely distributive. A similar argument shows that, for a fixed $n \in \mathbb{N}$ and ordinal $\alpha$, the lattice of $n$-tuples of ordinals, referred to as *ordinal vectors*, endowed with the lexicographic order $([\alpha]^n, \preceq)$ is completely distributive. In fact, the only elements that are not complete join-irreducibles are vectors of the kind $(0, \dots, 0, \alpha, \beta_i, \dots, \beta_n)$ where $\alpha$ is a limit ordinal and such vectors can be obtained as the join of the vectors $(0, \dots, 0, \beta, \beta_i, \dots, \beta_n)$, with $\beta < \alpha$ and $\beta$ a successor ordinal.

## 3 FIXPOINT EQUATIONS: SOLUTIONS AND APPROXIMANTS

In this section we introduce the systems of fixpoint equations we will work with in the paper. We define the solution of a system and we devise some results concerning its approximations that will play a major role later.

### 3.1 Systems of Fixpoint Equations

We focus on systems of (fixpoint) equations over some lattice, where, for each equation one can be interested either in the least or in the greatest solution.

*Definition 3.1 (system of equations).* Let $L$ be a lattice. A system of equations $E$ over $L$ is a list of equations of the following form

$$\begin{array}{lll} x_1 & =_{\eta_1} & f_1(x_1, \dots, x_m) \\ & \dots & \\ x_m & =_{\eta_m} & f_m(x_1, \dots, x_m) \end{array}$$

where $f_i \colon L^m \to L$ are monotone functions and $\eta_i \in \{\mu, \nu\}$. The system will often be denoted as $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$, where $\boldsymbol{x}$, $\boldsymbol{\eta}$ and $\boldsymbol{f}$ are the obvious tuples. We denote by $\emptyset$ the system with no equations.

Systems of equations of this kind have been considered by various authors, e.g., [Cleaveland et al. 1992; Hasuo et al. 2016; Seidl 1996]. In particular, [Hasuo et al. 2016] works on general lattices.

We next define the pre-solutions of a system as tuples of lattice elements that, replacing the variables, satisfy all the equations of the system. The solution will be a suitably chosen pre-solution, taking into account also the $\eta_i$ annotations that specify for each equation whether the least or greatest solution is required.

*Definition 3.2 (pre-solution).* Let $L$ be a lattice and let $E$ be a system of equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. A *pre-solution* of $E$ is any tuple $\boldsymbol{u} \in L^m$ such that $\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{u})$.

Note that $\boldsymbol{f}$ can be seen as a function $\boldsymbol{f}\colon L^m \to L^m$. In this view, pre-solutions are the fixpoints of $\boldsymbol{f}$. Since all components $f_i$ are monotone, also $\boldsymbol{f}$ is monotone over $(L^m, \sqsubseteq)$. Then, it is well-known that the set of fixpoints of $\boldsymbol{f}$, i.e., the pre-solutions of the system, are a sublattice. In order to define the solution of a system we need some further notation.

*Definition 3.3 (substitution).* Given a system $E$ of $m$ equations over a lattice $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$, an index $i \in \underline{m}$ and $l \in L$ we write $E[x_i := l]$ for the system of $m-1$ equations obtained from $E$ by removing the $i$-th equation and replacing $x_i$ by $l$ in the other equations, i.e., if $\boldsymbol{x} = \boldsymbol{x}' x_i \boldsymbol{x}''$, $\boldsymbol{\eta} = \boldsymbol{\eta}' \eta_i \boldsymbol{\eta}''$ and $\boldsymbol{f} = \boldsymbol{f}' f_i \boldsymbol{f}''$ then $E[x_i := l]$ is $\boldsymbol{x}'\boldsymbol{x}'' =_{\boldsymbol{\eta}', \boldsymbol{\eta}''} \boldsymbol{f}'\boldsymbol{f}''(\boldsymbol{x}', l, \boldsymbol{x}'')$.

Let $f[x_i := l]\colon L^{m-1} \to L$ be defined by $f[x_i := l](\boldsymbol{x}', \boldsymbol{x}'') = f(\boldsymbol{x}', l, \boldsymbol{x}'')$. Then, explicitly, the system $E[x_i := l]$ has $m-1$ equations,

$$x_j =_{\eta_j} f_j[x_i := l](\boldsymbol{x}', \boldsymbol{x}'') \qquad j \in \{1, \dots, i-1, i+1, \dots, n\}$$

We can now recursively define the solution of a system of equations. The notion is the same as in [Hasuo et al. 2016], although we find it convenient to adopt a more succinct formulation (an explicit proof of the equivalence of the two notions can be found in Appendix B.1).

*Definition 3.4 (solution).* Let $L$ be a lattice and let $E$ be a system of $m$ equations on $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. The *solution* of $E$, denoted $sol(E) \in L^m$, is defined inductively as follows:

$$\begin{aligned} sol(\emptyset) &= () \\ sol(E) &= (sol(E[x_m := u_m]),\, u_m) \text{ where } u_m = \eta_m(\lambda x.\, f_m(sol(E[x_m := x]), x)) \end{aligned}$$

The $i$-th component of the solution will be denoted $sol_i(E)$.

In words, for solving a system of $m$ equations, the last variable is considered as a fixed parameter $x$ and the system of $m-1$ equations that arises from dropping the last equation is recursively solved. This produces an $(m-1)$-tuple parametric on $x$, i.e., we get $\boldsymbol{u}_{1,m-1}(x) = sol(E[x_m := x])$. Inserting this parametric solution into the last equation, we get an equation in a single variable

$$x =_{\eta_m} f_m(\boldsymbol{u}_{1,m-1}(x), x)$$

that can be solved by taking for the function $\lambda x.\, f_m(\boldsymbol{u}_{1,m-1}(x), x)$, the least or greatest fixpoint, depending on whether the last equation is a $\mu$- or $\nu$-equation. This provides the $m$-th component of the solution $u_m = \eta_m(\lambda x.\, f_m(\boldsymbol{u}_{1,m-1}(x), x))$. The remaining components of the solution are obtained inserting $u_m$ in the parametric solution $\boldsymbol{u}_{1,m-1}(x)$ previously computed, i.e., $\boldsymbol{u}_{1,m-1} = \boldsymbol{u}_{1,m-1}(u_m)$.

The next lemma will be helpful in several places. In particular, it shows that the definition above is well-given, since we are taking (least or greatest) fixpoints of monotone functions.

Lemma 3.5 (solution is monotone). *Let $E$ be a system of $m > 0$ equations of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$ over a lattice $L$. For $i \in \underline{m}$ the function $g\colon L \to L^{m-1}$ defined by $g(x) = sol(E[x_i := x])$ is monotone.*

Proof. The proof proceeds by induction on $m$. The base case $m = 1$ holds trivially since necessarily $i = 1$ and for any $x \in L$, the system $E[x_i := x]$ is empty, with empty solution.

Let us assume $m > 1$. We distinguish two subcases according to whether $i = m$ or $i < m$. If $i = m$ then by definition of solution

$$g(x) = sol(E[x_m := x]) = (sol(E[x_m := x][x_{m-1} := u_{m-1}(x)]), u_{m-1}(x)) \tag{1}$$

where $u_{m-1}(x) = \eta_{m-1}(\lambda y.\, f_{m-1}(sol(E[x_m := x][x_{m-1} := y]), y, x)$.

Next observe that the function $h\colon L^2 \to L^{m-2}$ defined by $h(x, y) = sol(E[x_m := x][x_{m-1} := y])$ is monotone. In fact, it is monotone in $y$ by inductive hypothesis, and also in $x$, again by inductive hypothesis, since $E[x_m := x][x_{m-1} := y] = E[x_{m-1} := y][x_m := x]$. Observe that $u_{m-1}$ can be written as

$$u_{m-1}(x) = \eta_{m-1}(\lambda y.\, f_{m-1}(h(x, y), y, x))$$

Recalling that also $f_{m-1}$ is monotone, we deduce that $u_{m-1}$ is monotone.

Finally, using the definition of $g$ and $u_{m-1}$, from (1) we can derive

$$g(x) = (h(x, u_{m-1}(x)), u_{m-1}(x))$$

which allows us to conclude that $g$ is monotone.

If instead, $i < m$, just note that

$$g(x) = sol(E[x_i := x]) = (sol(E[x_i := x][x_m := u_m(x)]), u_m(x)) \tag{2}$$

where $u_m(x) = \eta_m(\lambda y. f_m\,(sol(E[x_i := x][x_m := y]), y, x)$. Then the proof proceeds as in the previous case. □

It can be easily proved that the solution of a system is, as anticipated, a special pre-solution.

Lemma 3.6 (solution is pre-solution). *Let $E$ be a system of $m$ equations over a lattice $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$ and let $\boldsymbol{u}$ be its solution. Then $\boldsymbol{u}$ is a pre-solution, i.e., $\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{u})$.*

Proof. The proof proceeds by induction on $m$. The base case $m = 0$ trivially holds. For any $m > 0$, let $\boldsymbol{u} = \boldsymbol{u}' u_m$, $\boldsymbol{f} = \boldsymbol{f}' f_m$ and $\boldsymbol{x} = \boldsymbol{x}' x_m$. Since $\boldsymbol{u}' = sol(E[x_m := u_m])$, by inductive hypothesis, we have that

$$\boldsymbol{u}' = \boldsymbol{f}'[x_m := u_m](\boldsymbol{u}') = \boldsymbol{f}'(\boldsymbol{u}). \tag{3}$$

Moreover, again by definition of solution, we have that $u_m = \eta_m(\lambda x.\, f_m(sol(E[x_m := x]), x))$. Hence $u_m = f_m(sol(E[x_m := u_m]), u_m))$. Recalling that $sol(E[x_m := u_m]) = \boldsymbol{u}'$ we deduce $u_m = f_m(\boldsymbol{u}', u_m) = f_m(\boldsymbol{u})$, that together with (3) gives $\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{u})$ as desired. □

### 3.2 A Prototypical Example: the $\mu$-Calculus

As a prototypical example, we discuss how $\mu$-calculus formulae can be equivalently seen as systems of fixpoint equations. We focus on a standard $\mu$-calculus syntax. For fixed disjoint sets *PVar* of propositional variables, ranged over by $x, y, z, \ldots$ and *Prop* of propositional symbols, ranged over by $p, q, r, \ldots$, formulae are defined by

$$\varphi \ ::= \ \mathbf{t} \mid \mathbf{f} \mid p \mid x \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \eta x.\, \varphi$$

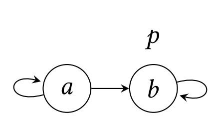
where $p \in Prop$, $x \in PVar$ and $\eta \in \{\mu, \nu\}$. Formulae of the kind $\eta x.\, \varphi$ are called fixpoint formulae.

The semantics of a formula is given with respect to an unlabelled transitions system (or Kripke structure) $(\mathbb{S}, \to)$ where $\mathbb{S}$ is the set of states and $\to \subseteq \mathbb{S} \times \mathbb{S}$ is the transition relation. Given a formula $\varphi$ and an environment $\rho\colon Prop \cup PVar \to \mathbf{2}^{\mathbb{S}}$ mapping each proposition or propositional variable to the set of states where it holds, we denote by $\|\varphi\|_\rho$ the semantics of $\varphi$ defined as usual (see, e.g., [Bradfield and Walukiewicz 2018]).

First note that any $\mu$-calculus formula can be expressed in equational form, by inserting an equation for each propositional variable (see also [Cleaveland et al. 1992; Seidl 1996]). The reverse translation is also possible, hence these specification languages are equally expressive. Here, we will only depict the relation via an example, the formal treatment is given in Appendix A.

$$\begin{aligned} x_1 &=_\mu p \lor \Diamond x_1 \\ x_2 &=_\nu x_1 \land \Box x_2 \end{aligned}$$

(a)




(b)

$$\begin{aligned} x_1 &=_\mu \{b\} \cup \Diamond x_1 \\ x_2 &=_\nu x_1 \cap \Box x_2 \end{aligned}$$

(c)

Fig. 2

*Example 3.7.* Let $\varphi = \nu x_2.((\mu x_1.(p \vee \Diamond x_1)) \wedge \Box x_2)$ be a formula requiring that from all reachable states there exists a path that eventually reaches a state where $p$ holds. The equational form is quite straightforward and is reported in Fig. 2a. Consider a transition system $(\mathbb{S}, \to)$ where $\mathbb{S} = \{a, b\}$ and $\to$ is as depicted in Fig. 2b, with $p$ that holds only on state $b$. The resulting system of equations on the lattice $\mathbf{2}^{\mathbb{S}}$ is given in Fig. 2c, where $\Diamond, \Box \colon \mathbf{2}^{\mathbb{S}} \to \mathbf{2}^{\mathbb{S}}$ are defined as $\Diamond(S) = \{s \in \mathbb{S} \mid \exists s' \in \mathbb{S}.(s \to s' \wedge s' \in S)\}$, $\Box(S) = \{s \in \mathbb{S} \mid \forall s' \in \mathbb{S}.(s \to s' \Rightarrow s' \in S)\}$ for $S \subseteq \mathbb{S}$.

The solution is $x_1 = x_2 = \mathbb{S}$. In particular, $x_2 = \mathbb{S}$ corresponds to the fact that the formula $\varphi$ holds in every state.

*Example 3.8.* Consider the formula $\varphi' = \nu x_2.(\Box x_2 \wedge \mu x_1.((p \wedge \Diamond x_2) \vee \Diamond x_1))$ requiring that from all reachable states there is a path along which $p$ holds infinitely often. The equational form of $\varphi'$ is:

$$\begin{aligned} x_1 &=_\mu (p \wedge \Diamond x_2) \vee \Diamond x_1 \\ x_2 &=_\nu \Box x_2 \wedge x_1 \end{aligned}$$

On the same transition system of the previous example (Fig. 2b), the solution of the corresponding system is $x_1 = x_2 = \mathbb{S}$. Notice that this time the order of the equations is relevant, while in the previous example it was not. Indeed, if we swap the two equations in the system, the solution becomes $x_1 = x_2 = \emptyset$. In general, the order of the equations is important whenever there is alternation of fixpoints (mutual dependencies between least and greatest fixpoint equations).

### 3.3 Data-Flow Analysis

In order to give further intuition, we revisit another area where fixpoints play a major role, namely data-flow analysis of programs. One can easily state a program analysis question in this setting as a system of fixpoint equations, based on the flow graph of the program under consideration.

We focus on the well-known *constant propagation analysis* (see, e.g., [Nielson et al. 1999]). Its aim is to show that the value of a variable is always constant at a certain program point, allowing us to optimise the program by replacing the variable by the constant. Consider for instance the while program in Fig. 3a, where variables contain integer values and blocks are numbered in order to easily reference them. The condition for the while loop (block 3) is irrelevant and is hence replaced by $\star$. Note that variable x always has value 7 in block 4 and hence the assignment in this block could be replaced by y:=7+y.

```
[y:=6]¹;
[x:=y+1]²;
while [*]³ do
    [y:=x+y]⁴
od
```

(a)

$$\begin{aligned} \rho_1 &=_\nu \bot \\ \rho_2 &=_\nu \rho_1[\mathsf{y} \mapsto 6] \\ \rho_3 &=_\nu \rho_2[\mathsf{x} \mapsto \rho_2(\mathsf{y}) + 1] \sqcap \rho_4[\mathsf{y} \mapsto \rho_4(\mathsf{x}) + \rho_4(\mathsf{y})] \\ \rho_4 &=_\nu \rho_3 \end{aligned}$$

(b)

Fig. 3. (a) A simple while program and (b) the equation system for the corresponding constant propagation analysis.

Following [Nielson et al. 1999] we analyse such programs by setting up an instance of a monotone framework. In particular we will use the following lattice to record the results of the analysis:

$$L = (\mathbb{Z} \cup \{\bot\})^{\mathit{Var}} \cup \{\top\}$$

where *Var* is the set of variables. That is, a lattice element is either $\top$ or a function $\rho \colon \mathit{Var} \to \mathbb{Z} \cup \{\bot\}$ that assigns a variable x to a value in $\mathbb{Z}$ (if x is known to have constant value $\rho(\mathsf{x})$ at this program point) or to $\bot$ (to indicate that x is possibly non-constant). As usual, we are allowed to over-approximate and $\bot$ might be assigned although the value of the variable is actually constant.

The lattice order is defined as follows: two assignments $\rho_1, \rho_2 \colon \mathit{Var} \to \mathbb{Z} \cup \{\bot\}$ are ordered, i.e. $\rho_1 \sqsubseteq \rho_2$, if for each $\mathsf{x} \in \mathit{Var}$ either $\rho_1(\mathsf{x}) = \rho_2(\mathsf{x})$ or $\rho_1(\mathsf{x}) = \bot$. That is, we consider a flat order where $\bot$ is smaller than the integers and the integers themselves are incomparable, and extend it pointwise to functions. Clearly, $\top$ is the largest lattice element and we use some overloading and denote by $\bot$ the function that maps every variable to $\bot$. Note that this order deviates from the usual convention in program analysis which states that smaller values should be more precise than larger values. We do this since our game characterises whether a lattice element is *below* the solution. Since we want to check that the solution is more precise than a given threshold, we have to reverse the order.

Let us write $\rho' = \rho[\mathsf{x} \mapsto z]$ for $z \in \mathbb{Z}$ to denote function update, that is $\rho'(\mathsf{x}) = z$ and $\rho'(\mathsf{y}) = \rho(\mathsf{y})$ for $\mathsf{y} \neq \mathsf{x}$. When $\rho = \top$ we define $\rho[\mathsf{x} \mapsto z] = \top$.

Observe that $L$ is algebraic (and hence continuous). The compact elements are $\top$ and those functions which have finite support, i.e., functions of the kind $\bot[\mathsf{x1} \mapsto z_1, \ldots, \mathsf{xn} \mapsto z_n]$ where only finitely many variables are not mapped to $\top$. In particular we can use as a basis the functions $\bot[\mathsf{x} \mapsto z]$ for some $\mathsf{x} \in \mathit{Var}$ and $z \in \mathbb{Z}$. Note also that $L$ is not distributive. For instance if $\rho_i = \bot[\mathsf{x} \mapsto i]$ for $i \in \{1,2,3\}$ then $(\rho_1 \sqcap \rho_2) \sqcup \rho_3 = \bot \sqcup \rho_3 = \rho_3$ while $(\rho_1 \sqcup \rho_3) \sqcap (\rho_2 \sqcup \rho_3) = \top \sqcap \top = \top$.

From the program in Fig. 3a we can easily derive the system of fixpoint equations in Fig. 3b, where we use $\rho_i$ to record the lattice value for the entry of block $i$.

At the beginning, no variable is constant. Then the equation system mimics the control flow of the program. In block 3 we have to take the meet to obtain an analysis result that is less precise than the results coming from block 2 respectively block 4. Furthermore, since precision increases with the order, we are interested in the greatest solution, which means that we have only $\nu$-equations.

The expected solution is $\rho_1 = \bot$, $\rho_2 = \bot[\mathsf{y} \mapsto 6]$, $\rho_3 = \rho_4 = \bot[\mathsf{x} \mapsto 7]$ witnessing that at block 2 variable y has constant value 6 and at blocks 3 and 4 variable x has constant value 7.

### 3.4 Approximating the Solution

The game theoretical characterisation of the solution of a system of fixpoint equations discussed later will rely on a notion of approximation of the solution that is reminiscent of the lattice progress measure in [Hasuo et al. 2016].

*Definition 3.9 (approximants).* Let $E$ be a system of $m$ equations over the lattice $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. Given any tuple $\boldsymbol{l} \in L^m$, let $f_{i,\boldsymbol{l}} \colon L \to L$ be the function defined by

$$f_{i,\boldsymbol{l}}(x) = f_i(sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := x]), x, \boldsymbol{l}_{i+1,m}).$$

We say that a tuple $\boldsymbol{l} \in L^m$ is a $\mu$-*approximant* when for all $i \in \underline{m}$, if $\eta_i = \nu$ then $l_i = \nu(f_{i,\boldsymbol{l}})$, else if $\eta_i = \mu$ then $l_i = f_{i,\boldsymbol{l}}^{\alpha}(\bot)$ for some ordinal $\alpha$. Dually, $\boldsymbol{l} \in L^m$ is a $\nu$-*approximant* when for all $i \in \underline{m}$, if $\eta_i = \nu$ then $l_i = f_{i,\boldsymbol{l}}^{\alpha}(\top)$ for some ordinal $\alpha$, else if $\eta_i = \mu$ then $l_i = \mu(f_{i,\boldsymbol{l}})$.

Whenever $\boldsymbol{l}$ is a $\mu$-approximant we write $ord(\boldsymbol{l})$ to denote the least $m$-tuple of ordinals $\boldsymbol{\alpha}$ such that for any $i \in \underline{m}$, if $\eta_i = \mu$ then $l_i = f_{i,\boldsymbol{l}}^{\alpha_i}(\bot)$ else, if $\eta_i = \nu$, $l_i = f_{i,\boldsymbol{l}}^{\alpha_i}(\top) = \nu(f_{i,\boldsymbol{l}})$.

Observe that, spelling out the definition of the solution of a system of equations, it can be easily seen that $sol_i(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}]) = \eta_i(f_{i,\boldsymbol{l}})$. Then a $\mu$-approximant is obtained by taking under-approximations for the least fixpoints and the exact value for greatest fixpoints. In fact, in the case of $\mu$-approximants, for each $i \in \underline{m}$, if $\eta_i = \nu$, the $i$-th component is set to $\nu(f_{i,\boldsymbol{l}})$ which is $i$-th component $sol_i(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}])$ of the solution. Instead, if $\eta_i = \mu$ the component $l_i$ is set to $f_{i,\boldsymbol{l}}^{\alpha}(\bot)$ for some ordinal $\alpha$, which is an underapproximation of $\mu(f_{i,\boldsymbol{l}}) = sol_i(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}])$, obtained by iterating $f_{i,\boldsymbol{l}}$ over $\bot$ up to ordinal $\alpha$. For $\nu$-approximants the situation is dual.

We remark that the function $f_{i,\boldsymbol{l}}$ depends only on the subvector $\boldsymbol{l}_{i+1,m}$. In particular $f_{m,\boldsymbol{l}}$ does not depend on $\boldsymbol{l}$. In fact, $f_{m,\boldsymbol{l}} = \lambda x.\, f_m(sol(E[x_m := x]), x)$. Using $\boldsymbol{l}$ as subscript instead of the subvector is a slight abuse of notation that makes the notation lighter.

Approximants can be given an inductive characterisation. Besides shedding some light on the notion of approximant, the following easy result will be useful at a technical level.

Lemma 3.10 (inductive characterisation of approximants). *Let $E$ be a system of $m > 0$ equations over the lattice $L$, of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$ and let $g_m \colon L \to L$ be the function $g_m(x) = f_m(sol(E[x_m := x]), x)$. A tuple $\boldsymbol{l} \in L^m$ is a $\mu$-approximant iff the following conditions hold*

*(1) either $\eta_m = \mu$ and $l_m = g_m^{\alpha}(\bot)$ for some ordinal $\alpha$, or $\eta_m = \nu$ and $l_m = \nu g_m$*
*(2) $\boldsymbol{l}_{1,m-1}$ is a $\mu$-approximant of $E[x_m := l_m]$.*

Proof. Immediate. □

As mentioned above, $\mu$-approximants are closely related to lattice progress measures in the sense of [Hasuo et al. 2016]. In fact, from Lemma 3.10 we can infer that, given a vector $\boldsymbol{\alpha}$ of ordinals, the $\mu$- or $\nu$-approximant $\boldsymbol{l} \in L^m$ with $ord(\boldsymbol{l}) = \boldsymbol{\alpha}$ is uniquely determined. More precisely, a $\mu$-approximant $\boldsymbol{l}$ is determined by the subvector of $ord(\boldsymbol{l})$ consisting only of the $m$-tuple of components of $ord(\boldsymbol{l})$ corresponding to $\mu$-indices. Hence we can define a function that maps each such $m$-tuple of ordinals to the corresponding $\mu$-approximant and this turns out to be a lattice progress measures in the sense of [Hasuo et al. 2016]. Actually, as proved in the Appendix B.2, it is the greatest one. It can be seen to coincide with the measure used in [Hasuo et al. 2016, Theorem 2.13] (completeness part).

We next observe that the name approximant is appropriate, i.e., $\mu$-approximants provide an approximation of the solution from below, while $\nu$-approximants from above. The solution is thus the only pre-solution which is both a $\mu$- and a $\nu$-approximant.

Lemma 3.11 (solution and approximants). *Let $E$ be a system of $m$ equations over the lattice $L$, of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. The solution of $E$ is the greatest $\mu$-approximant and the least $\nu$-approximant.*

Proof. The solution $\boldsymbol{u}$ is clearly a $\mu$-approximant. We prove that it is the greatest $\mu$-approximant by induction on $m$. If $m = 0$ the thesis is vacuously true. If $m > 0$, consider another $\mu$-approximant $\boldsymbol{l}$. We distinguish two subcases according to whether $\eta_m = \mu$ or $\eta_m = \nu$. If $\eta_m = \mu$, we know that $l_m = f_{m,\boldsymbol{l}}^{\alpha}(\bot)$ for some ordinal $\alpha$. Observe that $f_{m,\boldsymbol{l}} = \lambda x.\, f_m(sol(E[x_m := x]), x))$ is the function for which $u_m$ is the least fixpoint, hence

$$l_m \sqsubseteq u_m. \tag{4}$$

Moreover, by Lemma 3.10, $\boldsymbol{l}_{1,m-1}$ is a $\mu$-approximant for the system $E[x_m := l_m]$. Hence, by inductive hypothesis

$$\boldsymbol{l}_{1,m-1} \sqsubseteq sol(E[x_m := l_m]) \tag{5}$$

Moreover, by monotonicity of the solution (Lemma 3.5), since $l_m \sqsubseteq u_m$, we get $sol(E[x_m := l_m]) \sqsubseteq sol(E[x_m := u_m]) = \boldsymbol{u}_{1,m-1}$. Therefore, combined with (4) and (5), we conclude $\boldsymbol{l} \sqsubseteq \boldsymbol{u}$.

The proof for $\nu$-approximants is dual. □

We conclude with a technical lemma that will be used to locally modify approximations in the game.

Lemma 3.12 (updating approximants). *Let $E$ be a system of $m$ equations over the lattice $L$, of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$ and let $\boldsymbol{l}$ be a $\mu$-approximant with $ord(\boldsymbol{l}) = \boldsymbol{\alpha}$. For any $i \in \underline{m}$ and ordinal $\alpha \leq \alpha_i$*

*(1) if $\eta_i = \mu$, then $\boldsymbol{l}' = (sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := l'_i]), l'_i, \boldsymbol{l}_{i+1,m})$, with $l'_i = f_{i,\boldsymbol{l}}^{\alpha}(\bot)$ for some ordinal $\alpha$, is a $\mu$-approximant*

*(2) if $\eta_i = \nu$, then $\boldsymbol{l}' = (sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}]), \boldsymbol{l}_{i+1,m})$ is a $\mu$-approximant*
*and in both cases $ord(\boldsymbol{l}') \preceq_i ord(\boldsymbol{l})$. A dual result holds for $\nu$-approximants.*

Proof. Let us focus on (1). In order to show that $\boldsymbol{l}' = (sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := l'_i]), l'_i, \boldsymbol{l}_{i+1,m})$ is a $\mu$-approximant, first observe that the components $l_{i+1}, \ldots, l_m$ do not change. Component $l'_i$ is of the desired shape by definition. Finally, for $j < i$ the component $l'_j$ is defined as $sol_j(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := l'_i])$ and thus, by definition of solution of a system, if $\eta_j = \nu$ then $l'_j = \nu(f_{j,\boldsymbol{l}'})$ and if $\eta_j = \mu$ then $l'_j = \mu(f_{j,\boldsymbol{l}'}) = f^{\beta}_{j,\boldsymbol{l}'}(\bot)$ for some ordinal $\beta$, as desired. Finally observe that since $\boldsymbol{l}$ and $\boldsymbol{l}'$ coincide on components $i+1, \ldots, m$, and $l_i = f^{\alpha_i}_{i,\boldsymbol{l}}(\bot)$, while $l'_i = f^{\alpha}_{i,\boldsymbol{l}}(\bot)$, with $\alpha \leq \alpha_i$, clearly $ord(\boldsymbol{l}') \preceq_i ord(\boldsymbol{l})$.

The proof for (2) is analogous. In fact, also in this case the components $i+1 \ldots, m$ are unchanged and finally, for $j \leq i$ the component $l'_j$ is defined as $sol_j(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := l'_i])$, thus the same reasoning as above applies. Both can easily dualised for $\nu$-approximants. □

## 4 FIXPOINT GAMES

In this section we present a game-theoretical approach to the solution of a system of fixpoint equations over a continuous lattice. More precisely, given a lattice with a fixed basis, the game allows us to check whether an element of the basis is smaller (with respect to $\sqsubseteq$) than the solution of a selected equation. This corresponds to solving the associated verification problem. For instance, when model-checking the $\mu$-calculus, one is interested in establishing whether a system satisfies a formula $\varphi$, which amounts to check whether $\{s_0\} \subseteq u_\varphi$ where $s_0$ is the initial state and $u_\varphi$ is the solution of the system of equations associated with $\varphi$.

### 4.1 Definition of the Game

The fixpoint game that we introduce has been inspired by the unfolding game described in [Venema 2008], that works for a single fixpoint equation over the powerset lattice. We adopted the name *fixpoint game*, analogously to [Hansen et al. 2017].

*Definition 4.1 (fixpoint game).* Let $L$ be a continuous lattice and let $B_L$ be a basis of $L$ such that $\bot \notin B_L$. Given a system $E$ of $m$ equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$, the corresponding fixpoint game is a parity game, with an existential player $\exists$ and a universal player $\forall$, defined as follows:

- The positions of $\exists$ are pairs $(b, i)$ where $b \in B_L$ and $i \in \underline{m}$ and those of $\forall$ are tuples $\boldsymbol{l} \in L^m$.
- From $(b, i)$ the possible moves of $\exists$ are $\mathrm{E}(b, i) = \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge \ b \sqsubseteq f_i(\boldsymbol{l})\}$.
- From $\boldsymbol{l} \in L^m$ the possible moves of $\forall$ are $\mathrm{A}(\boldsymbol{l}) = \{(b, i) \mid i \in \underline{m} \ \wedge \ b \in B_L \ \wedge \ b \ll l_i\}$.

The game is schematised in Table 1. For a finite play, the winner is the player whose opponent is unable to move. For an infinite play, let $h$ be the highest index that occurs infinitely often in a pair $(b, i)$. If $\eta_h = \nu$ then $\exists$ wins, else $\forall$ wins.

| Position | Player | Moves |
|---|---|---|
| $(b, i)$ | $\exists$ | $(l_1, \ldots, l_m)$ such that $b \sqsubseteq f_i(l_1, \ldots, l_m)$ |
| $(l_1, \ldots, l_m)$ | $\forall$ | $(b', j)$ such that $b' \ll l_j$ |

Table 1. The fixpoint game

Observe that the fixpoint game is a parity game [Emerson and Jutla 1991; Zielonka 1998] (on an infinite graph) and the winning condition is the natural formulation of the standard winning condition in this setting.

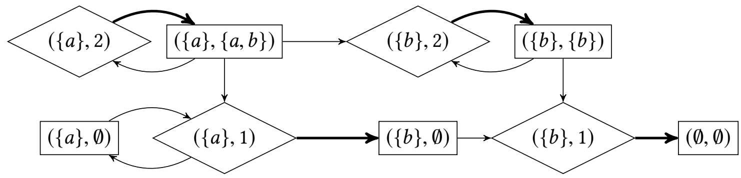



Fig. 4. Graphical representation of a fixpoint game

Hereafter, whenever we consider a continuous lattice $L$, we assume that a basis $B_L$ is fixed such that $\bot \notin B_L$. Elements of the basis will be denoted by letters $b$ with super or subscripts.

We will prove correctness and completeness of the game, i.e., we will show that if $\boldsymbol{u}$ is the solution of the system, given a basis element $b \in B_L$ and $i \in \underline{m}$, if $b \sqsubseteq u_i$ then starting from $(b, i)$ the existential player has a winning strategy, otherwise the universal player has a winning strategy.

*Example 4.2.* As an example, consider the equation system of Example 3.7, as depicted in Fig. 2c, corresponding to the $\mu$-calculus formula $\varphi = \nu x_2.((\mu x_1.(p \vee \Diamond x_1)) \wedge \Box x_2)$. Recall that the lattice is $(\mathbf{2}^{\mathbb{S}}, \subseteq)$ and let us fix as a basis the set of singletons $B_{\mathbf{2}^{\mathbb{S}}} = \{\{a\}, \{b\}\}$.

A portion of the fixpoint game is graphically represented as a parity game in Fig. 4. Diamond nodes correspond to positions of player $\exists$ and the box nodes to positions of player $\forall$. Only a subset of the possible positions for $\forall$ are represented. The positions which are missing, such as $(\{a, b\}, \{a, b\})$, can be shown to be redundant, in a sense formalised later (see § 5.3.1), so that the subgame is equivalent to the full game. Numbers in the diamond nodes correspond to priorities. Box nodes do not have priorities (or we can assume priority 0). Since index 1 and 2 corresponds to a $\mu$ and a $\nu$ equation, respectively, in this specific case the winning condition for player $\exists$ is exactly the same as for parity games: either the play is finite and $\exists$ plays last or the play is infinite and the highest priority that occurs infinitely often is even (in this case 2).

Let $(u_1, u_2)$ be the solution of the system. We can check if $a \in u_2$, i.e., if $a$ satisfies $\varphi$, by playing the game from the position $(\{a\}, 2)$. In fact, $\{a\} \sqsubseteq u_2$ amounts to $a \in u_2$. Indeed player $\exists$ has a winning strategy that we can represent as a function $\varsigma$ from the positions of the game (for any play) to the corresponding moves of player $\exists$, i.e., $\varsigma \colon B_{\mathbf{2}^{\mathbb{S}}} \times \underline{2} \to \mathbf{2}^{\mathbb{S}} \times \mathbf{2}^{\mathbb{S}}$. A winning strategy for $\exists$ is given by $\varsigma(\{a\}, 1) = (\{b\}, \emptyset)$, $\varsigma(\{a\}, 2) = (\{a\}, \{a, b\})$, $\varsigma(\{b\}, 1) = (\emptyset, \emptyset)$ and $\varsigma(\{b\}, 2) = (\{b\}, \{b\})$. In Fig. 4 we depict by bold arrows the choices prescribed by the strategy.

A possible play of the game could be the following, where $\overset{x}{\rightsquigarrow}$ denotes a move of $x \in \{\exists, \forall\}$:

$$(\{a\}, 2) \overset{\exists}{\rightsquigarrow} (\{a\}, \{a, b\}) \overset{\forall}{\rightsquigarrow} (\{a\}, 1) \overset{\exists}{\rightsquigarrow} (\{b\}, \emptyset) \overset{\forall}{\rightsquigarrow} (\{b\}, 1) \overset{\exists}{\rightsquigarrow} (\emptyset, \emptyset) \overset{\forall}{\not\rightsquigarrow},$$

hence $\exists$ wins. Another (infinite) play is the following. It is also won by $\exists$ since the highest index that occurs infinitely often is 2, which is a $\nu$-index:

$$(\{a\}, 2) \overset{\exists}{\rightsquigarrow} (\{a\}, \{a, b\}) \overset{\forall}{\rightsquigarrow} (\{a\}, 2) \overset{\exists}{\rightsquigarrow} (\{a\}, \{a, b\}) \overset{\forall}{\rightsquigarrow} \ldots$$

Note that if $\exists$ always plays as specified by $\varsigma$, she will always win.

## 4.2 Correctness and Completeness

Before proving correctness and completeness of the game in the general case, as a warm up, we give some intuition and outline the proof for the case of a single equation. Let $f : L \to L$ be a monotone function on a continuous lattice $L$ and consider the equation $x =_\eta f(x)$, where $\eta \in \{\nu, \mu\}$, with solution $u = \eta f$. In this case the positions for $\exists$ are simply basis elements $b \in B_L$ and $\exists$ must choose $l \in L$ such that $b \sqsubseteq f(l)$. Positions of $\forall$ are lattice elements $l \in L$ and moves are elements of the basis $b \in B_L$, with $b \ll l$. In the case of $\eta = \mu$, player $\forall$ wins infinite plays and in the case of $\eta = \nu$, player $\exists$ wins infinite plays.

When $\eta = \mu$, if $b \sqsubseteq u$, then $b \sqsubseteq f^{\alpha}(\bot)$ for some ordinal $\alpha$. The idea is that $\exists$ can win by descending the chain $f^{\beta}(\bot)$. E.g., if $\beta = \gamma + 1$ is a successor ordinal, then $\exists$ can play $f^{\gamma}(\bot)$. If instead, $\eta = \nu$, then the existential player can win just by identifying some post-fixpoint $l$ such that $b \sqsubseteq l$. In fact, if $l$ is a post-fixpoint, i.e., $l \sqsubseteq f(l)$ we know that $l \sqsubseteq u$. Moreover, if $b \sqsubseteq l$ then $b \sqsubseteq f(l)$ and thus $\exists$ can cycle on $l$ and win. More formally:

*(Case $\eta = \mu$).* In this case $u = f^{\alpha}(\bot)$ for some ordinal $\alpha$.

- *Completeness:* We show that whenever $b \sqsubseteq f^{\beta}(\bot)$, for some ordinal $\beta$ (i.e., $b$ is below some $\mu$-approximant), then $\exists$ has a winning strategy, by transfinite induction on $\beta$. First observe that $\beta > 0$. In fact, otherwise $b \sqsubseteq f^{0}(\bot) = \bot$, hence $b = \bot$, while $\bot \notin B_L$ by hypothesis. Hence we have two possibilities:
  - If $\beta$ is a limit ordinal, player $\exists$ plays $l = f^{\beta}(\bot)$, which is a post-fixpoint and hence $b \sqsubseteq f^{\beta}(\bot) \sqsubseteq f(f^{\beta}(\bot))$. Then $\forall$ chooses $b' \ll f^{\beta}(\bot) = \bigsqcup_{\gamma<\beta} f^{\gamma}(\bot)$. Since this is a directed join, by definition of the way-below relation there exists $\gamma < \beta$ with $b' \sqsubseteq f^{\gamma}(\bot)$.
  - If $\beta = \gamma + 1$, $\exists$ plays $l = f^{\gamma}(\bot)$ and $\forall$ chooses $b' \ll f^{\gamma}(\bot)$, hence $b' \sqsubseteq f^{\gamma}(\bot)$.

  Note that $\exists$ always has a move and the answer of $\forall$ is some $b' \sqsubseteq f^{\gamma}(\bot)$, with $\gamma < \beta$, from which there exists a winning strategy for $\exists$ by the inductive hypothesis.
- *Correctness:* We show that whenever $b \not\sqsubseteq u$, player $\forall$ has a winning strategy.
  Observe that a move of $\exists$ will be some $l$ such that $b \sqsubseteq f(l)$. Note that there must be a $b' \ll l$ with $b' \not\sqsubseteq u$. In fact, otherwise, if for all $b' \ll l$ it holds that $b' \sqsubseteq u$, since $L$ is a continuous lattice, we would have $l = \bigsqcup\{b' \mid b' \ll l\} \sqsubseteq u$ and furthermore $b \sqsubseteq f(l) \sqsubseteq f(u) = u$, which is a contradiction.
  Hence $\forall$ can choose such a $b' \ll l$ with $b' \not\sqsubseteq u$ and the game can continue. Then either $\exists$ runs out of moves at some point or we end up in an infinite play. In both cases $\forall$ wins.

*(Case $\eta = \nu$).* In this case $u = f^{\alpha}(\top)$ for some ordinal $\alpha$.

- *Completeness:* We show that when $b \sqsubseteq u$, then $\exists$ has a winning strategy. In fact, in this case $\exists$ simply plays $l = u$, which satisfies $b \sqsubseteq u = f(u)$ and $\forall$ answers with some $b \ll u$, hence $b \sqsubseteq u$. The game can thus continue forever, leading to an infinite play which is won by $\exists$.
- *Correctness:* We show that whenever $b \not\sqsubseteq f^{\beta}(\top)$, for some ordinal $\beta$ (i.e., $b$ is not below some $\nu$-approximant), then $\forall$ has a winning strategy, by transfinite induction on $\beta$. First observe that $\beta > 0$. In fact, otherwise $b \not\sqsubseteq f^{0}(\top) = \top$ would be a contradiction. Hence we distinguish two cases:
  - If $\beta$ is a limit ordinal $b \not\sqsubseteq f^{\beta}(\top) = \bigsqcap_{\gamma<\beta} f^{\gamma}(\top)$, which means that there exists $\gamma < \beta$ such that $b \not\sqsubseteq f^{\gamma}(\top)$.
    Now any move of $\exists$ is some $l$ with $b \sqsubseteq f(l)$. Therefore $l \not\sqsubseteq f^{\gamma}(\top)$, since otherwise $b \sqsubseteq f(l) \sqsubseteq f(f^{\gamma}(\top)) = f^{\gamma+1}(\top) \sqsubseteq f^{\beta}(\top)$ (since $\gamma + 1 < \beta$). Hence there must be $b' \ll l$ with $b' \not\sqsubseteq f^{\gamma}(\top)$. Otherwise, as above, if for all $b' \ll l$ we had $b' \sqsubseteq f^{\gamma}(\top)$, then by continuity of the lattice, we would conclude $l = \bigsqcup\{b' \mid b' \ll l\} \sqsubseteq f^{\gamma}(\top)$. Such a $b'$ can be chosen by $\forall$, and the game continues.
  - If $\beta = \gamma + 1$ we know that $b \not\sqsubseteq f^{\beta}(\top) = f(f^{\gamma}(\top))$.
    Any move of $\exists$ is $l$ with $b \sqsubseteq f(l)$, which as above implies that $l \not\sqsubseteq f^{\gamma}(\top)$ and thus the existence of $b' \ll l$ with $b' \not\sqsubseteq f^{\gamma}(\top)$. The basis element $b'$ is chosen by $\forall$ and the game continues.

  Hence $\forall$ always has a move, ending up in $b' \not\sqsubseteq f^{\gamma}(\top)$, from which there exists a winning strategy for $\forall$ by the induction hypothesis.

Observe that cases of a $\mu$- and a $\nu$-equation are not completely symmetric. In the completeness part, for showing that $l \sqsubseteq \nu f$ we use the fact that $\nu f$ is the greatest post-fixpoint. Instead, for showing that $l \sqsubseteq \mu f$ we use the fact that $l \sqsubseteq f^{\alpha}(\bot)$ for some $\alpha$ and provide a proof that we can descend to $\bot$, similarly to what happens for ranking functions in termination analysis. Note that in order to guarantee that we truly descend, also below limit ordinals, we require that $\forall$ plays $b$ with $b \ll l$. Then we can use the fact that whenever $b$ is way-below a directed join, then it is smaller than one of the elements over which the join is taken. We remark that choosing $b$ with $b \sqsubseteq l$ instead would not be sufficient (see Proposition 4.10). In the correctness part, despite the asymmetry, both proofs use the fact that each element is the join of all elements way-below it, for which it is essential to be in a continuous lattice (see Proposition 4.9). Instead, for completeness, the continuity hypothesis does not play a role.

For the general case, correctness and completeness of the game are proved by relying on the notions of $\mu$- and $\nu$-approximant. We prove the two properties separately. Completeness exploits a result that shows how $\exists$ can play descending along a chain of $\mu$-approximants and, as in the case of a single equation, it can be proved for general lattices, without assuming the continuity hypothesis.

Lemma 4.3 (descending on $\mu$-approximants). *Let $E$ be a system of $m$ equations over a lattice $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. For each $\mu$-approximant $\boldsymbol{l} \in L^m$ and $(b, i) \in \mathbf{A}(\boldsymbol{l})$ there exists a $\mu$-approximant $\boldsymbol{l'} \in \mathbf{E}(b, i)$ such that $ord(\boldsymbol{l}) \succeq_i ord(\boldsymbol{l'})$. Moreover, if $\eta_i = \mu$, the $i$-th component strictly decreases and thus the inequality is strict.*

Proof. Let $\boldsymbol{l} \in L^m$ be a $\mu$-approximant and let $(b, i)$ in $\mathbf{A}(\boldsymbol{l})$, i.e., $b \in B_L$ and $i \in \underline{m}$ with $b \ll l_i$. We distinguish various cases:

(1) ($\eta_i = \mu$) This means that $l_i = f^{\alpha}_{i,\boldsymbol{l}}(\bot)$ for some ordinal $\alpha$. Since $f^{0}_{i,\boldsymbol{l}}(\bot) = \bot$ and $b \ll \bot$ would imply $b = \bot$, while $\bot \notin B_L$, necessarily $\alpha \neq 0$. We distinguish two subcases:

(a) $\alpha = \beta + 1$ *is a successor ordinal*

Let $l'_i = f^{\beta}_{i,\boldsymbol{l}}(\bot)$ and $(l'_1, \ldots, l'_{i-1}) = sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := l'_i])$. Then define

$$\boldsymbol{l'} = (l'_1, \ldots, l'_{i-1}, l'_i, \boldsymbol{l}_{i+1,m})$$

Observe that $\boldsymbol{l'}$ is a $\mu$-approximant by Lemma 3.12. Moreover $\boldsymbol{l'} \in \mathbf{E}(b, i)$. In fact

$$\begin{aligned} b \sqsubseteq l_i &= f^{\beta+1}_{i,\boldsymbol{l}}(\bot) \\ &= f_{i,\boldsymbol{l}}(f^{\beta}_{i,\boldsymbol{l}}(\bot)) \\ &= f_{i,\boldsymbol{l}}(l'_i) \\ &= f_i(sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := l'_i]), l'_i, \boldsymbol{l}_{i+1,m}) \\ &= f_i(l'_1, \ldots, l'_{i-1}, l'_i, \boldsymbol{l}_{i+1,m}) \\ &= f_i(\boldsymbol{l'}) \end{aligned}$$

Finally, note that $ord(\boldsymbol{l'}) \prec_i ord(\boldsymbol{l})$ since vectors $\boldsymbol{l}$ and $\boldsymbol{l'}$ coincide on the components $i + 1, \ldots, m$, and $l_i = f^{\beta+1}_{i,\boldsymbol{l}}(\bot)$ while $l'_i = f^{\beta}_{i,\boldsymbol{l}}(\bot)$.

(b) $\alpha$ *is a limit ordinal*

Since $b \ll l_i = f^{\alpha}_{h,\boldsymbol{l}}(\bot) = \bigsqcup_{\beta<\alpha} f^{\beta}_{i,\boldsymbol{l}}(\bot)$, which is a directed join, by definition of the way-below relation, there is $\beta < \alpha$ such that $b \sqsubseteq f^{\beta}_{i,\boldsymbol{l}}(\bot)$. We set $l'_i = f^{\beta}_{i,\boldsymbol{l}}(\bot)$ and $(l'_1, \ldots, l'_{i-1}) = sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := l'_i])$. Then we define

$$\boldsymbol{l'} = (l'_1, \ldots, l'_{i-1}, l'_i, \boldsymbol{l}_{i+1,m})$$

The vector $\boldsymbol{l}'$ is a $\mu$-approximant by Lemma 3.12. Moreover $\boldsymbol{l}' \in \mathbf{E}(b, i)$ since

$$
\begin{aligned}
b &\sqsubseteq l'_i \\
&\sqsubseteq f_{i,\boldsymbol{l}}(l'_i) && \text{[since } l'_i = f^{\beta}_{i,\boldsymbol{l}}(\bot) \text{ is a post-fixpoint]} \\
&= f_i(sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := l'_i]), l'_i, \boldsymbol{l}_{i+1,m}) \\
&= f_i(l'_1, \ldots, l'_{i-1}, l'_i, \boldsymbol{l}_{i+1,m}) \\
&= f_i(\boldsymbol{l}')
\end{aligned}
$$

Finally, note that $ord(\boldsymbol{l}') \prec_i ord(\boldsymbol{l})$ since vectors $\boldsymbol{l}$ and $\boldsymbol{l}'$ coincide on the components $i+1, \ldots, m$, and $l_i = f^{\alpha}_{i,\boldsymbol{l}}(\bot)$ while $l'_i = f^{\beta}_{i,\boldsymbol{l}}(\bot)$, with $\beta < \alpha$.

(2) ($\eta_i = \nu$)
In this case $l_i = \nu(f_{i,\boldsymbol{l}})$. Let $(l'_1, \ldots, l'_{i-1}) = sol(E[\boldsymbol{x}_{i,m} := \boldsymbol{l}_{i,m}])$. Then define

$$\boldsymbol{l}' = (l'_1, \ldots, l'_{i-1}, \boldsymbol{l}_{i,m})$$

The vector $\boldsymbol{l}'$ is a $\mu$-approximant by Lemma 3.12. Moreover, observe that $\boldsymbol{l}' \in \mathbf{E}(b, i)$, since

$$
\begin{aligned}
b &\sqsubseteq l_i \\
&= f_{i,\boldsymbol{l}}(l_i) && \text{[since } l_i \text{ is a fixpoint]} \\
&= f_i(sol(E[\boldsymbol{x}_{i,m} := \boldsymbol{l}_{i+1,m}]), \boldsymbol{l}_{i,m}) \\
&= f_i(l'_1, \ldots, l'_{i-1}, \boldsymbol{l}_{i,m}) \\
&= f_i(\boldsymbol{l}')
\end{aligned}
$$

Finally, note that $ord(\boldsymbol{l}') \preceq_i ord(\boldsymbol{l})$ since vectors $\boldsymbol{l}$ and $\boldsymbol{l}'$ coincide on the components $i, \ldots, m$.

□

The previous result allows us to prove that player ∃ can always win starting from a $\mu$-approximant. Roughly, relying on Lemma 4.3, we can prove that player ∃ can play on $\mu$-approximants in a way that each time the $i$-th equation is chosen, the ordinal vector associated to the approximant decreases with respect to $\preceq_i$, and it strictly decreases when the $i$-th equation is a $\mu$-equation. This, together with the fact that the order on ordinals is well-founded, allows one to conclude that either the play is finite and ∃ plays last or the highest index on which one can cycle is necessarily the index of a $\nu$-equation. In both cases player ∃ wins.

Lemma 4.4 (∃ wins on $\mu$-approximants). *Let $E$ be a system of $m$ equations over a lattice $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$ and let $\boldsymbol{l} \in L^m$ be a $\mu$-approximant. Then in a game starting from $\boldsymbol{l}$ (which is a position of ∀) player ∃ has a winning strategy.*

Proof. We first describe the strategy for player ∃ and then prove that it is a winning strategy.

The key observation is that ∃ can always play a $\mu$-approximant, where she plays the solution in the first step. In fact, let $\boldsymbol{l}' \in L^m$ be the current $\mu$-approximant. For any possible move $(b', i') \in \mathbf{A}(\boldsymbol{l}')$ of ∀, by Lemma 4.3 there always exists a move $\boldsymbol{l}'' \in \mathbf{E}(b', i')$ of ∃ which is a $\mu$-approximant such that $ord(\boldsymbol{l}') \succeq_i ord(\boldsymbol{l}'')$. Additionally, if $\eta_i = \mu$ the inequality is strict.

Since ∃ player has always a move, either the play finishes because ∀ has no moves, hence ∃ wins or the play continues forever.

In this last case, note that, if $h$ is the largest index occurring infinitely often, then necessarily $\eta_h = \nu$, hence ∃ wins. In fact, assume by contradiction that $\eta_h = \mu$. Consider the sequence of turns of the play starting from the point where all indexes repeat infinitely often.

Let $\boldsymbol{l}'$, $(b', j)$, $\boldsymbol{l}''$ be consecutive turns. By the choice of $h$, necessarily $j \leq h$. Moreover, by construction, if

$$ord(\boldsymbol{l'}) \succeq_j ord(\boldsymbol{l''})$$

Observing that for $j \leq j'$ it holds $\boldsymbol{\alpha} \succeq_j \boldsymbol{\alpha'}$ implies $\boldsymbol{\alpha} \succeq_{j'} \boldsymbol{\alpha'}$, we deduce that

$$ord(\boldsymbol{l'}) \succeq_h ord(\boldsymbol{l''})$$

i.e., the sequence is decreasing. Moreover, since $\eta_h = \mu$, whenever $j = h$, $ord(\boldsymbol{l'}) \succ_h ord(\boldsymbol{l''})$, i.e., the sequence strictly decreases. This contradicts the well-foundedness of $\succ_h$. □

Since the solution of a system of equation is a $\mu$-approximant (the greatest one), completeness is an easy corollary of Lemma 4.4.

Corollary 4.5 (completeness). *Let $E$ be a system of $m$ equations over a lattice $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. Given any $\mu$-approximant $\boldsymbol{l} \in L^m$, $b \in B_L$ and $i \in \underline{m}$, if $b \sqsubseteq l_i$ then $\exists$ has a winning strategy from position $(b, i)$.*

Proof. Just observe that at the first turn $\exists$ can play the $\mu$-approximant $\boldsymbol{l}$ that is in $\mathbf{E}(b, i)$ by hypotheses. Then using Lemma 4.4 we conclude that $\exists$ wins. □

For correctness we rely on a result, dual to Lemma 4.3, that allows to ascend along $\nu$-approximants. However, in this case, the fact of working in a continuous lattice is crucial (see Proposition 4.9).

Lemma 4.6 (ascending on $\nu$-approximants). *Let $E$ be a system of $m$ equations over a continuous lattice $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. Given a $\nu$-approximant $\boldsymbol{l} \in L^m$, an element $b \in B_L$ and an index $i \in \underline{m}$ with $b \not\sqsubseteq l_i$, for all tuples $\boldsymbol{l'} \in \mathbf{E}(b, i)$ there are a $\nu$-approximant $\boldsymbol{l''}$ and $(b'', j) \in \mathbf{A}(\boldsymbol{l'})$ such that (1) $b'' \not\sqsubseteq l''_j$ and (2) $ord(\boldsymbol{l}) \succeq_i ord(\boldsymbol{l''})$. Moreover, if $\eta_i = \nu$, the $i$-th component strictly decreases and thus the inequality in item 2 above is strict.*

Proof. Let $\boldsymbol{l} \in L^m$ be a $\nu$-approximant, let $b \in B_L$ and let $i \in \underline{m}$ with $b \not\sqsubseteq l_i$. Take $\boldsymbol{l'} \in \mathbf{E}(b, i)$, i.e., such that $b \sqsubseteq f_i(\boldsymbol{l'})$. We prove that there are a $\nu$-approximant $\boldsymbol{l''}$ and $(b'', j) \in \mathbf{A}(\boldsymbol{l'})$ satisfying (1) and (2) above, by distinguishing various cases:

(i) ($\eta_i = \mu$) Define $\boldsymbol{l''} = (sol(E[\boldsymbol{x}_{i,m} := \boldsymbol{l}_{i,m}]), l_i, \boldsymbol{l}_{i+1,m})$, which is a $\nu$-approximant by Lemma 3.12. Note that, since $l_i = \mu(f_{i,\boldsymbol{l}})$,

$$l_i = f_{i,\boldsymbol{l}}(l_i) = f_i(sol(E[\boldsymbol{x}_{i,m} := \boldsymbol{l}_{i,m}]), l_i, \boldsymbol{l}_{i+1,m}) = f_i(\boldsymbol{l''})$$

We first prove prove (1), i.e., that there exists $(b'', j) \in \mathbf{A}(\boldsymbol{l'})$, i.e., $j \in \underline{m}$ and $b'' \in B_L$, $b'' \ll l'_j$ with $b'' \not\sqsubseteq l''_j$. In fact, otherwise, if for any $j$ and $b'' \ll l'_j$ we had $b'' \sqsubseteq l''_j$, then for any $j$, since $B_L$ is a basis and $L$ a continuous lattice:

$$l'_j = \bigsqcup\{b'' \mid b'' \in B_L \land\ b'' \ll l'_j\} \sqsubseteq l''_j.$$

However, by monotonicity of $f_i$, this would imply $f_i(\boldsymbol{l'}) \sqsubseteq f_i(\boldsymbol{l''}) = l_i$, that together with the hypothesis $b \sqsubseteq f_i(\boldsymbol{l'})$, would contradict $b \not\sqsubseteq l_i$.

For point (2), note that $ord(\boldsymbol{l'}) \preceq_i ord(\boldsymbol{l})$ since vectors $\boldsymbol{l}$ and $\boldsymbol{l'}$ coincide on all components $i, \ldots, m$, and $l_i = f_{i,\boldsymbol{l}}^{\alpha}(\bot)$.

(ii) ($\eta_i = \nu$) This means that $l_i = f_{i,\boldsymbol{l}}^{\alpha}(\top)$ for some ordinal $\alpha$, necessarily $\alpha \neq 0$ (since otherwise $l_i = \top$ and $b \not\sqsubseteq l_i$ could not hold). We distinguish two subcases

(a) $\alpha = \beta + 1$ *is a successor ordinal*
Define $\boldsymbol{l}'' = (sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := f_{i,\boldsymbol{l}}^{\beta}(\top)]), f_{i,\boldsymbol{l}}^{\beta}(\top), \boldsymbol{l}_{i+1,m})$. Then we have

$$\begin{aligned} b &\not\sqsubseteq l_i \\ &= f_{i,\boldsymbol{l}}^{\beta+1}(\top) \\ &= f_{i,\boldsymbol{l}}(f_{i,\boldsymbol{l}}^{\beta}(\top)) \\ &= f_i(sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := f_{i,\boldsymbol{l}}^{\beta}(\top)]), f_{i,\boldsymbol{l}}^{\beta}(\top), \boldsymbol{l}_{i+1,m}) \\ &= f_i(\boldsymbol{l}'') \end{aligned}$$

Recalling that $b \sqsubseteq f_i(\boldsymbol{l}')$, as in case (i) we deduce point (1), i.e., that there exists $(b'', j) \in \mathbf{A}(\boldsymbol{l}')$ such that $b'' \not\sqsubseteq l_j''$.
Concerning point (2), note that $ord(\boldsymbol{l}') \prec_i ord(\boldsymbol{l})$ since vectors $\boldsymbol{l}$ and $\boldsymbol{l}'$ coincide on the components $i + 1, \ldots, m$, and $l_i = f_{i,\boldsymbol{l}}^{\beta+1}(\bot)$ while $l_i' = f_{i,\boldsymbol{l}}^{\beta}(\bot)$.

(b) $\alpha$ *is a limit ordinal*
In this case

$$b \not\sqsubseteq l_i = f_{i,\boldsymbol{l}}^{\alpha}(\top) = \textstyle\bigsqcap_{\beta<\alpha} f_{i,\boldsymbol{l}}^{\beta}(\top) = \bigsqcap_{\beta<\alpha} f_{i,\boldsymbol{l}}^{\beta+1}(\top)$$

Therefore there exists $\beta < \alpha$ such that $b \not\sqsubseteq f_{i,\boldsymbol{l}}^{\beta+1}(\top)$. Hence, we can define $l_i'' = f_{i,\boldsymbol{l}}^{\beta}(\top)$ and take the $\nu$-approximant

$$\boldsymbol{l}'' = (sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := l_i'']), l_i'', \boldsymbol{l}_{i+1,m})$$

Then we have

$$\begin{aligned} b &\not\sqsubseteq f_{i,\boldsymbol{l}}^{\beta+1}(\top) \\ &= f_{i,\boldsymbol{l}}(f_{i,\boldsymbol{l}}^{\beta}(\top)) \\ &= f_i(sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := f_{i,\boldsymbol{l}}^{\beta}(\top)]), f_{i,\boldsymbol{l}}^{\beta}(\top), \boldsymbol{l}_{i+1,m}) \\ &= f_i(\boldsymbol{l}'') \end{aligned}$$

and thus, again, recalling that $b \sqsubseteq f_i(\boldsymbol{l}')$, as in case (i) we deduce point (1), i.e., that there exists $(b'', j) \in \mathbf{A}(\boldsymbol{l}')$ such that $b'' \not\sqsubseteq l_j''$.
Concerning point (2), note that $ord(\boldsymbol{l}'') \prec_i ord(\boldsymbol{l})$ since vectors $\boldsymbol{l}$ and $\boldsymbol{l}''$ coincide on the components $i + 1, \ldots, m$, and $l_i = f_{i,\boldsymbol{l}}^{\alpha}(\bot)$ while $l_i'' = f_{i,\boldsymbol{l}}^{\beta}(\bot)$, with $\beta < \alpha$.

□

As in the dual case, correctness is an easy corollary of the above lemma, recalling that the solution is the least $\nu$-approximant.

Lemma 4.7 (correctness). *Let $E$ be a system of $m$ equations over a continuous lattice $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. For a $\nu$-approximant $\boldsymbol{l} \in L^m$, $b \in B_L$ and $i \in \underline{m}$, if $b \not\sqsubseteq l_i$ then $\forall$ has a winning strategy from position $(b, i)$.*

Proof. We first describe the strategy for the universal player and then prove that it is a winning strategy.
Let $\boldsymbol{l} \in L^m$ be a $\nu$-approximant, $b \in L$ and $i \in \underline{m}$ such that $b \not\sqsubseteq l_i$. Starting from $(b, i)$, for any possible move $\boldsymbol{l}' \in \mathbf{E}(b, i)$ of $\exists$. Then $\forall$ can play a pair $(b', j) \in \mathbf{A}(\boldsymbol{l}')$, whose existence is ensured by Lemma 4.6, such that there is a $\nu$-approximant $\boldsymbol{l}''$ satisfying $b'' \not\sqsubseteq l_j''$ and $ord(\boldsymbol{l}'') \prec_i ord(\boldsymbol{l})$. Additionally, if $\eta_i = \nu$ the inequality is strict.

According to the strategy defined above $\forall$ player has always a move. Thus either the play finishes because $\exists$ has no moves, hence $\forall$ wins or the play continues forever.

In this last case, with an argument dual with respect to that in Lemma 4.4, we can show that if $h$ is the largest index occurring infinitely often, then necessarily $\eta_h = \mu$, hence $\forall$ win. $\square$

Combining Corollary 4.5 and Lemma 4.7 we reach the desired result.

Theorem 4.8 (correctness and completeness). *Given a system of $m$ equations $E$ over a continuous lattice $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} f(\boldsymbol{x})$ with solution $\boldsymbol{u}$, then for all $b \in B_L$ and $i \in \underline{m}$,*

$$b \sqsubseteq u_i \quad \text{iff} \quad \exists \text{ has a winning strategy from position } (b, i).$$

Proof. Immediate corollary of Lemma 3.11, Corollary 4.5 and Lemma 4.7. $\square$

Note that even when the fixpoint is reached in more than $\omega$ steps, thanks to the fact that the order on the ordinals is well-founded and players "descend" over the order, ordinals do not play an explicit role in the game. In particular plays are not transfinite and whenever $\forall$ or $\exists$ win due to the fact that the other player cannot make a move, this happens after a *finite* number of steps. This can be a bit surprising at first since the game works for general continuous lattices, including, for instance, intervals over the reals.

We close this subsection by proving two results that, in a sense, show that the choice of continuous lattices and the design of the game based on the way-below relation are "the right ones". We first observe that the restriction to continuous lattices is not only sufficient but also necessary for the correctness of the game.

Proposition 4.9 (correctness holds exactly in continuous lattices). *Let $L$ be a lattice and let $B_L$ be a fixed basis with $\bot \notin B_L$. The game is correct for every system of equations over $L$ if and only if $L$ is continuous.*

Proof. We already know from Lemma 4.7 that when $L$ is continuous the game is correct.

Conversely, let $L$ be a non-continuous lattice. This means that there is an element $l \in L$ such that $l \neq \bigsqcup \twoheaddownarrow l$. Note that since $\twoheaddownarrow l \subseteq \downarrow l$, where $\downarrow$ is the downward-closure with respect to $\sqsubseteq$, we have $\bigsqcup \twoheaddownarrow l \sqsubset \bigsqcup \downarrow l = l$. We prove that, for any basis $B_L$ for $L$ such that $\bot \notin B_L$, there are a monotone function $f \colon L \to L$ and an element $b \in B_L$ such that $b \not\sqsubseteq \mu f$, for which there is a winning strategy for the existential player for the corresponding fixpoint game starting from position $b$, while such a strategy should not exists. The function $f$ is defined by:

$$f(x) = \begin{cases} \bigsqcup \twoheaddownarrow l & \text{if } x \sqsubseteq \bigsqcup \twoheaddownarrow l \\ \top & \text{otherwise.} \end{cases}$$

Notice that necessarily $\bigsqcup \twoheaddownarrow l \neq \top$, since $\bigsqcup \twoheaddownarrow l \sqsubset l \sqsubseteq \top$. Then, clearly $f$ is monotone and its least fixpoint is $\mu f = \bigsqcup \twoheaddownarrow l$. Moreover, since $B_L$ is a basis, we know that $\bigsqcup(\downarrow l \cap B_L) = l$. Then there must be $b \in B_L$ such that $b \sqsubseteq l$ but $b \not\sqsubseteq \bigsqcup \twoheaddownarrow l$. Otherwise, if for all $b \in B_L$ such that $b \sqsubseteq l$, $b \sqsubseteq \bigsqcup \twoheaddownarrow l$, then we would have $\bigsqcup(\downarrow l \cap B_L) \sqsubseteq \bigsqcup \twoheaddownarrow l$, contradicting the hypothesis $\bigsqcup \twoheaddownarrow l \sqsubset l = \bigsqcup(\downarrow l \cap B_L)$. Now we show that player $\exists$ is able to win any play of the game, with a single equation $x =_\mu f(x)$, for checking whether such a $b \sqsubseteq \mu f$. The strategy is actually quite simple and can be described by just one family of plays. We use the fact that $f(l) = \top$.

$$b \overset{\exists}{\rightsquigarrow} l \overset{\forall}{\rightsquigarrow} b' \overset{\exists}{\rightsquigarrow} \bot \overset{\forall}{\not\rightsquigarrow}$$

for any $b' \ll l$, since $b' \sqsubseteq \bigsqcup \twoheaddownarrow l = f(\bot)$. Thus player $\exists$ can always win, despite the fact that $b \not\sqsubseteq \mu f$. $\square$

As a counterexample, consider the lattice $W$ in Fig. 1, which is not continuous and let $B_W$ be any basis such that $0 \notin B_W$. First note that necessarily $a \in B_W$, otherwise $a \neq \bigsqcup\{x \in B_W \mid x \sqsubseteq a\} = \bigsqcup \emptyset = 0$. Secondly, $\twoheaddownarrow a = \{0\}$ since $a \not\ll a$. Then, consider the equation $x =_\mu f(x)$, where the function $f \colon W \to W$ is defined by $f(0) = 0$, and $f(x) = \omega$ for $x \neq 0$. Clearly $f$ is monotone and its least fixpoint is $\mu f = 0$. However, the player $\exists$ can win any play of the game from position $a$, despite the fact that $a \not\sqsubseteq \mu f = 0$. In fact, the first move of $\exists$ can be $a$, since $a \sqsubseteq f(a) = \omega$. But then player $\forall$ has no moves since $\twoheaddownarrow a \cap B_W = \emptyset$. And so player $\exists$ always wins while she should not.

The second observation is that using the lattice order instead of the way-below relation may break completeness. More precisely, consider the natural variant of the game where the way-below relation is replaced by the lattice order. Let us call it *weak game*. Since the set of possible moves of player $\forall$ is enlarged, correctness clearly continues to hold. Instead, as we hinted before, completeness could fail. We show that it is exactly on algebraic lattices that completeness still holds for the weak game.

Proposition 4.10 (way-below is needed in non-algebraic lattices). *Let $L$ be a lattice. The weak game is complete on every system of equations over $L$ if and only if $B_L$ consists of compact elements (which in turn means that $L$ is algebraic).*

Proof. Let $K_L$ be the set of compact elements of $L$. If $B_L \subseteq K_L$, then for any $b \in B_L$ and $l \in L$, we have that $b \ll l$ if and only if $b \sqsubseteq l$. Therefore the weak game coincides with the original one and hence completeness clearly holds.

Conversely, assume that $B_L \not\subseteq K_L$. We show that we can identify a system consisting of a single equation $x =_\mu f(x)$ for which the weak game is not complete. Let $b \in B_L \setminus K_L$ be a non-compact element in the basis. Therefore $b \not\ll b$ which means that there exists a directed set $D$ such that $b \sqsubseteq \bigsqcup D$ and $b \not\sqsubseteq d$ for all $d \in D$. Without loss of generality we can assume that $D$ is a transfinite chain $D = (d_\alpha)_\alpha$ (see, e.g., [Markowsky 1976, Theorem 1]).

Consider the function $f \colon L \to L$ defined as

$$f(x) = \begin{cases} \top & \text{if } b \sqsubseteq x \\ d_\alpha & \text{otherwise, where } \alpha = \min\{\beta \mid d_\beta \not\sqsubseteq x\} \end{cases}$$

Observe that the function $f$ is well-defined. In fact, when $b \not\sqsubseteq x$ there exists $\beta$ such that $d_\beta \not\sqsubseteq x$ and hence the set $\{\beta \mid d_\beta \not\sqsubseteq x\}$ is not empty. In fact, if we had $d_\beta \sqsubseteq x$ for all elements of the chain, we would deduce $\bigsqcup D \sqsubseteq x$ and thus, recalling $b \sqsubseteq \bigsqcup D$, we would conclude $b \sqsubseteq x$.

Observe that $f$ is monotone. In fact, let $x, y \in L$ with $x \sqsubseteq y$. If $b \sqsubseteq y$ and thus $f(y) = \top$, we trivially conclude $f(x) \sqsubseteq \top = f(y)$. Let us then consider the case in which $b \not\sqsubseteq y$ and thus $f(y) = d_\alpha$ where $\alpha = \min\{\beta \mid d_\beta \not\sqsubseteq y\}$. Obviously $b \not\sqsubseteq x$ and thus $f(x) = d_{\alpha'}$, where $\alpha' = \min\{\beta \mid d_\beta \not\sqsubseteq x\}$. Since $x \sqsubseteq y$, we have $\{\beta \mid d_\beta \not\sqsubseteq x\} \supseteq \{\beta \mid d_\beta \not\sqsubseteq y\}$, hence $\alpha' \leq \alpha$ and thus $f(x) = d_{\alpha'} \sqsubseteq d_\alpha = f(y)$, as desired.

By construction $\top$ is the only fixpoint of $f$, hence $\top = \mu f$. Thus $b \sqsubseteq \mu f = \top$. Now, if we play the weak game, since $b \not\sqsubseteq d_\alpha$ for all $\alpha$, the possible moves for $\exists$ are initially only those $x \in L$ such that $b \sqsubseteq x$ and thus $b \sqsubseteq f(x) = \top$. However, if $\exists$ play such an $x$, in the weak game $\forall$ can answer $b$, getting back to the initial situation. Hence $\forall$ wins, providing the desired counterexample to completeness. $\square$

Note that when the elements of the basis are compact, the way-below relation with respect to elements of the basis is the lattice order. Hence the result above essentially states that the weak game is complete exactly when it coincides with the original game, thus further supporting the appropriateness of our formulation of the game.

As a counterexample, consider the continuous lattice $[0, 1]$ with the usual order and basis $B_{[0,1]} = \mathbb{Q} \cap (0, 1]$. Recall that $[0, 1]$ is not algebraic (the only compact element is 0) and way-below relation is the strict order $<$. Let $g\colon [0,1] \to [0,1]$ be the function defined by $g(x) = \frac{x+1}{2}$. The fixpoint equation $x =_\mu g(x)$ has solution $\mu g = 1$.

In the weak game, from position $l \in [0,1]$, player $\forall$ can play any $b \leq l$ (instead, of $b < l$). Then player $\exists$ loses any play starting from position 1, despite the fact that $1 \leq \mu g = 1$. In fact, the only possible move of player $\exists$ is 1, and $\forall$ can play any $x \leq 1$. In particular, playing 1 the game will continue forever and will thus be won by $\forall$.

Notice that, instead, in the original game, from position 1, player $\forall$ has to play an element $1-\epsilon$ for some $\epsilon > 0$. Then, it is easy to see that at each step $i$ player $\exists$ will be able to play some $z_i \leq 1 - 2^i\epsilon$. This means that after finitely many steps $\exists$ will be allowed to play 0, thus leaving no possible answer to $\forall$ and winning the game.

### 4.3 Relation to $\mu$-Calculus Model-Checking

We discuss how our fixpoint game over systems arising from $\mu$-calculus formulae relates to classical techniques for model-checking the $\mu$-calculus, which can be presented interchangeably in terms of parity games, tableaux, and automata (see, e.g., [Emerson 1985]). Specifically, we compare our game with classical tableau systems for the $\mu$-calculus (e.g., as in [Cleaveland 1990; Stirling and Walker 1991]) where the similarities can be presented more directly.

Recall that a tableau is a (finite) proof tree whose nodes are labelled by sequents. Usually sequents are of the kind $s \models \varphi$, where $s$ is a state of the model and $\varphi$ formula. The fact that a state $s$ satisfies a formula $\varphi$ amounts to the existence of a tableau, rooted in $s \models \varphi$ and that it is *successful*, according to a suitable definition.

Given a closed $\mu$-calculus formula $\varphi$ and a state $s$ in a transition system $(\mathbb{S}, \to)$, let $E$ be the corresponding system of $m$ equations as in Definition A.1. The model-checking problem using tableaux is solved by searching for a successful tableau for the sequent $s \models \varphi$. Instead, using the fixpoint game, it is reduced to the existence of a winning strategy for player $\exists$ starting from position $(\{s\}, m)$, where $m$ is the highest equation index.

We discuss the two approaches using Example 3.7. Recall that the formula of interest is $\varphi = \nu x_2.((\mu x_1.(p \vee \Diamond x_1)) \wedge \Box x_2)$. Let $\psi$ denote the subformula $\mu x_1.(p \vee \Diamond x_1)$. Using the tableau rules in the style of [Cleaveland 1990] (omitting assumptions for the sake of the presentation), we can build a successful tableau for the sequent $a \models \varphi$ as in Fig. 5a. It is not difficult to see that this tableau corresponds to the winning strategy $\varsigma$ for $\exists$ discussed in Example 4.2. In fact, consider the reduced tree in Fig. 5b, which is obtained from the tableau by keeping only the sequents corresponding to fixpoint formulae (i.e., either $\varphi$ or $\psi$) and replacing such formulae with the corresponding variable ($\varphi$ with $x_2$ and $\psi$ with $x_1$).

Each sequent $s \models x_i$ can be seen as a position $(\{s\}, i) \in B_{2^{\mathbb{S}}} \times \underline{2}$ of $\exists$ in the fixpoint game. The successor sequents correspond to the move prescribed on $(\{s\}, i)$ by the strategy $\varsigma$. More precisely, the move should be $(y_1, y_2)$ where $y_j = \{s' \mid s' \models x_j \text{ is a successor of } s \models x_i\}$ for $j \in \underline{2}$. For instance, the sequent $a \models x_2$ corresponds to the position $(\{a\}, 2)$. The three successors $a \models x_1$, $a \models x_2$ and $b \models x_2$ determine the move prescribed by the strategy $\varsigma(\{a\}, 2) = (\{a\}, \{a, b\})$. Instead, the sequent $a \models x_1$ has only one successor $b \models x_1$ and, correspondingly, we have $\varsigma(\{a\}, 1) = (\{b\}, \emptyset)$, since there are no successors containing variable $x_2$. When a sequent appears on a leaf of the reduced tree which was already a leaf in the original tableau, by definition of the tableau rules it must have an ancestor labelled by the same sequent and in this case the strategy is defined by the ancestor. For instance, in Fig. 5, the sequent $b \models x_2$ labels the leaf ① which was already a leaf in the original tableau, marked by ②. The strategy is thus defined by the ancestor ③, labelled by the same sequent $b \models x_2$, as $\varsigma(\{b\}, 2) = (\{b\}, \{b\})$.

Additionally, it can be seen that plays of the fixpoint game correspond to paths in the reduced tree. For example, the first play discussed in Example 4.2 corresponds to the leftmost path in the

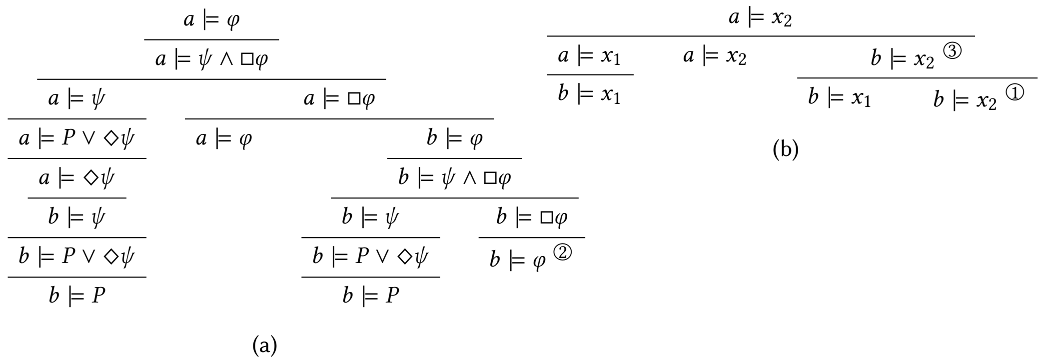



Fig. 5. $\mu$-calculus tableaux vs strategies in the fixpoint game

tree. In fact, while successors of sequents define the strategy for player $\exists$, the moves of player $\forall$ determine the path to follow.

For general, possibly non-successful tableaux, if we consider the reduced tree, then for each subtree the sequents at the leaves can be read as a set of assumptions that player $\exists$ has taken to show that the root sequent holds. Player $\forall$ chooses among such assumptions which one player $\exists$ should develop next. If there is no winning strategy for player $\exists$, the winning strategy for player $\forall$ is such that he always chooses a path in the tableau that cannot be successfully concluded at a leaf.

### 4.4 Fixpoint Games in Data-Flow Analysis

We get back to constant propagation example in § 3.3. Recall that the system of fixpoint equations expressing the analysis in Fig. 3b had solution $\rho_1 = \bot$, $\rho_2 = \bot[\mathsf{y} \mapsto 6]$, $\rho_3 = \rho_4 = \bot[\mathsf{x} \mapsto 7]$. We next describe a game that shows that indeed $\bot[\mathsf{x} \mapsto 7] \sqsubseteq \rho_4$ and hence $\mathsf{x}$ has constant value 7 at block 4. The game starts as follows:

$$(\bot[\mathsf{x} \mapsto 7], 4) \overset{\exists}{\leadsto} (\bot, \bot, \bot[\mathsf{x} \mapsto 7], \bot) \overset{\forall}{\leadsto} (\bot[\mathsf{x} \mapsto 7], 3) \overset{\exists}{\leadsto} (\bot, \bot[\mathsf{y} \mapsto 6], \bot, \bot[\mathsf{x} \mapsto 7]) \overset{\forall}{\leadsto}$$

Now the universal player has two options: either choose $(\bot[\mathsf{x} \mapsto 7], 4)$, which brings him back to an earlier game situation and might potentially lead to an infinite game. Since we are considering greatest fixpoints, this means that $\exists$ wins. If he chooses the other option, the game continues as follows, where eventually $\forall$ has no move left and $\exists$ wins as well:

$$(\bot[\mathsf{y} \mapsto 6], 2) \overset{\exists}{\leadsto} \bot \overset{\forall}{\not\leadsto}$$

## 5 STRATEGIES AS PROGRESS MEASURES

Along the lines of [Jurdziński 2000], influenced by [Hasuo et al. 2016], in this section we introduce a general notion of progress measure for fixpoint games over continuous lattices. We will show how a complete progress measure characterises the winning positions for the two players. The existence of a so-called small progress measure will allow us to express a complete progress measure as a least fixpoint, thus providing a technique for computing the progress measure and solving the corresponding system of equations.

## 5.1 General Definition

Given an ordinal $\alpha$ we denote by $[\alpha]^m_\star = \{\beta \mid \beta \leq \alpha\}^m \cup \{\star\}$, the set of ordinal vectors with entries smaller or equal than $\alpha$, with an added bound $\star$.

*Definition 5.1 (progress measure).* Let $L$ be a continuous lattice and let $E$ be a system of $m$ equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. Given an ordinal $\lambda$, a $\lambda$*-progress measure* for $E$ is a function $R \colon B_L \to \underline{m} \to [\lambda]^m_\star$ such that for all $b \in B_L$, $i \in \underline{m}$, either $R(b)(i) = \star$ or there exists $\boldsymbol{l} \in \mathbf{E}(b, i)$ such that for all $(b', j) \in \mathbf{A}(\boldsymbol{l})$ it holds

- if $\eta_i = \mu$ then $R(b)(i) \succ_i R(b')(j)$;
- if $\eta_i = \nu$ then $R(b)(i) \succeq_i R(b')(j)$

A progress measure maps any basis element of the lattice and index $i \in \underline{m}$ to an $m$-tuple of ordinals, with one component for each equation. Components relative to $\mu$-equations roughly measure how many unfolding steps for the equation would be needed to reach an under-approximation $l_i$ above $b$, and thus, for $\exists$, to win the game. Components relative to $\nu$-equations, as in the original work of [Jurdziński 2000], are less relevant, as we will see.

Intuitively, whenever $R(b)(i) \neq \star$, the progress measure $R$ provides an evidence of the existence of a winning strategy for $\exists$ in a play starting from $(b, i)$. The tuple $\boldsymbol{l}$, whose existence is required by the definition, is a move of player $\exists$ such that for any possible answer of $\forall$, the progress measure will not increase with respect to $\preceq_i$, and it will strictly decrease in the case of $\mu$-equations. Since $\prec_i$ is well-founded, this ensures that we cannot cycle on a $\mu$-equation. Also note that whenever the current index is $i$, all indices lower than $i$ are irrelevant (expressed by the orders $\succeq_i$ resp. $\succ_i$), which is related to the fact that the highest index which is visited infinitely often is the only relevant index for determining the winner of the game. This idea is formalised in the following lemma.

Lemma 5.2 (progress measures are strategies). *Let $L$ be a continuous lattice and let $E$ be a system of $m$ equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$ with solution $\boldsymbol{u}$. For any $b \in B_L$ and $i \in \underline{m}$, if there exists some ordinal $\lambda$ and a $\lambda$-progress measure $R$ such that $R(b)(i) \preceq_i (\lambda, \ldots, \lambda)$, then $b \sqsubseteq u_i$.*

Proof. We show that $\exists$ has a winning strategy from $(b, i)$. The strategy consists in choosing a move $\boldsymbol{l} \in \mathbf{E}(b, i)$ such that for all $(b', j) \in \mathbf{A}(\boldsymbol{l})$, it holds

- $R(b)(i) \succ_i R(b')(j)$, if $\eta_i = \mu$
- $R(b)(i) \succeq_i R(b')(j)$, if $\eta_i = \nu$

which exists by definition of progress measure.

Now, observe that player $\exists$ can always make its turn. Therefore either the play stops because $\forall$ runs out of moves, hence $\exists$ win. Otherwise, the play is infinite, and, if we denote by $h$ the largest index occurring infinitely often, then $\eta_h = \nu$, hence $\exists$ wins. In fact, assume by contradiction that $\eta_h = \mu$. Consider the sequence of turns of the play starting from the point where all indexes repeat infinitely often and take the $m$-tuples of ordinals $R(b')(h)$ corresponding to the positions $(b', i)$ where $\exists$ plays. For any two successive elements, say $(b', i)$ and $(b'', j)$, by construction

$$R(b')(i) \succeq_i R(b'')(j)$$

Observing that for $i \leq j$ it holds $\boldsymbol{\alpha} \succeq_i \boldsymbol{\alpha}'$ implies $\boldsymbol{\alpha} \succeq_j \boldsymbol{\alpha}'$, we deduce that

$$R(b')(i) \succeq_h R(b'')(j)$$

i.e., the sequence is decreasing. Moreover, since $\eta_h = \mu$, whenever $i = h$, $R(b')(i) \succ_h R(b'')(j)$, i.e., the sequence strictly decreases. This contradicts well-foundedness of $\prec_h$. □

The above lemma, in a sense, says that progress measures provide sound characterisations of the solution. However, in general, they are not complete, since whenever $R(b)(i) = \star$ we cannot derive any information on $(b, i)$, i.e., if $\boldsymbol{u}$ is the solution of the system, we cannot conclude that $b \not\sqsubseteq u_i$. This motivates the following definition.

*Definition 5.3 (complete progress measures).* Let $L$ be a continuous lattice and let $E$ be a system of equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$ with solution $\boldsymbol{u}$. A $\lambda$-progress measure $R \colon B_L \to \underline{m} \to [\lambda]_\star^m$ is called *complete* if for all $b \in B_L$ and $i \in \underline{m}$, if $b \sqsubseteq u_i$ then $R(b)(i) \preceq_i (\lambda, \ldots, \lambda)$.

Observe that in search of a complete progress measure, in principle, we would have to try all ordinals as a bound. We next show that we can take as bound the height $\lambda_L$ of the lattice $L$. This provides a generalisation of the *small progress measure* in [Jurdziński 2000].

*Definition 5.4 (small progress measure).* Let $L$ be a continuous lattice and let $E$ be a system of $m$ equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. Given an $m$-tuple of ordinals $\boldsymbol{\alpha}$, let us denote by $z_E(\boldsymbol{\alpha})$ the $m$-tuple of ordinals where $\nu$-components are set to 0, i.e., $z_E(\boldsymbol{\alpha}) = \boldsymbol{\beta}$ with $\beta_i = \alpha_i$ if $\eta_i = \mu$ and $\beta_i = 0$ otherwise. We define the *small progress measure* $R_E \colon B_L \to \underline{m} \to [\lambda_L]_\star^m$

$$R_E(b)(i) = \min_{\preceq_i}\{z_E(\mathit{ord}(\boldsymbol{l})) \mid \boldsymbol{l} \text{ is a } \mu\text{-approximant } \wedge\ \boldsymbol{l} \in \mathbf{E}(b, i)\}$$

where $\min_{\preceq_i}$ is the minimum on $\preceq_i$ as given in Definition 2.9, with the convention that $\min_{\preceq_i} \emptyset = \star$.

Observe that $R_E$ is well-defined, i.e., it actually takes values in $[\lambda_L]_\star^m$. In fact, the components of $z_E(\mathit{ord}(\boldsymbol{l}))$ corresponding to $\mu$-indices are ordinals expressing the number of Kleene iterations needed to reach under-approximations of the least fixpoint. These are clearly bounded by $\lambda_L$, since for a monotone function $f \colon L \to L$, the sequence $f^\alpha(\bot)$ is strictly increasing until it reaches the least fixpoint of $f$. For $\nu$-indices, $z_E(\mathit{ord}(\boldsymbol{l}))$ is always 0.

Observe that while formally $R_E(b)(i)$ takes values in $[\lambda_L]_\star^m$, whenever $j < i$ or $\eta_j = \nu$, due to the effect of the $\min_{\preceq_i}$ and of the $z_E$ operations, the only possible value for the $j$-th component is 0. Despite such components are then clearly irrelevant, we keep them for notational convenience.

The fact that $R_E$ is indeed a progress measure follows from Lemma 4.3. Moreover, we can easily show that it is complete.

Lemma 5.5 (small progress measure). *Let $L$ be a continuous lattice and let $E$ be a system of $m$ equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. Then $R_E \colon B_L \to \underline{m} \to [\lambda_L]_\star^m$ is a progress measure and it is complete.*

Proof. For the first part, let $R_E(b)(i) = \boldsymbol{\alpha} \neq \star$. Hence $R_E(b)(i) =_i z_E(\mathit{ord}(\boldsymbol{l}))$ for some $\mu$-approximant $\boldsymbol{l}$ such that $\boldsymbol{l} \in \mathbf{E}(b, i)$. By Lemma 4.3, for all $(b', j) \in \mathbf{A}(\boldsymbol{l})$ there exists $\boldsymbol{l}' \in \mathbf{E}(b', j)$ such that $\mathit{ord}(\boldsymbol{l}) \succeq_i \mathit{ord}(\boldsymbol{l}')$ and, if $\eta_i = \mu$, the inequality is strict since the $i$-th component strictly decreases. Clearly, this implies $z_E(\mathit{ord}(\boldsymbol{l})) \succeq_i z_E(\mathit{ord}(\boldsymbol{l}'))$. Additionally, if $\eta_i = \mu$, the inequality remains strict since the $i$-th component is left unchanged by the $z_E$ operation.

Therefore, by definition of $R_E$, for all $(b', j) \in \mathbf{A}(\boldsymbol{l})$ we have

$$R_E(b')(j) \preceq_i z_E(\mathit{ord}(\boldsymbol{l}')) \preceq_i z_E(\mathit{ord}(\boldsymbol{l})) =_i R_E(b)(i). \tag{6}$$

where, if $\eta_i = \mu$, the inequality is strict, as desired.

Let us now show that $R_E$ is complete. Let $b \in B_L$ be such that $b \sqsubseteq u_i$. We know that the solution $\boldsymbol{u}$ is a $\mu$-approximant. Moreover, since $b \sqsubseteq u_i = f_i(\boldsymbol{u})$, we have that $\boldsymbol{u} \in \mathbf{E}(b, i)$. Hence $R_E(b)(i) \preceq_i z_E(\mathit{ord}(\boldsymbol{u}))$ and thus $R_E(b)(i) \neq \star$.

□

## 5.2 Progress Measures as Fixpoints

Here we show that a complete progress measure can be characterised as the least solution of a system of equations over tuples of ordinals, naturally induced by Definition 5.1.

*Definition 5.6 (progress measure equations).* Let $L$ be a continuous lattice and let $E$ be a system of $m$ equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. Let $\boldsymbol{\delta}_i^{\eta}$, with $i \in \underline{m}$, be, for $\eta = \nu$, the null vector and, for $\eta = \mu$, the vector such that $\delta_j = 0$ if $j \neq i$ and $\delta_i = 1$. The *progress measure equations* for $E$ over the lattice $[\lambda_L]_\star^m$, are defined, for $b \in B_L$, $i \in \underline{m}$, as:

$$R(b)(i) = \min_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b', j) \in \mathrm{A}(\boldsymbol{l})\} \mid \boldsymbol{l} \in \mathrm{E}(b, i)\}$$

We will denote by $\Phi_E$ the corresponding endofunction on $L \to \underline{m} \to [\lambda_L]_\star^m$ which is defined, for $R\colon B_L \to \underline{m} \to [\lambda_L]_\star^m$, by

$$\Phi_E(R)(b)(i) = \min_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b', j) \in \mathrm{A}(\boldsymbol{l})\} \mid \boldsymbol{l} \in \mathrm{E}(b, i)\}$$

Observe that, since $[\lambda_L]_\star^m$ is a lattice, also the corresponding set of progress measures, endowed with pointwise $\preceq$-order, is a lattice. It is immediate to see that $\Phi_E$ is monotone with respect to such order, i.e., if $R \preceq R'$ pointwise then $\Phi_E(R) \preceq \Phi_E(R')$ pointwise. This allows us to obtain a complete progress measure as a (least) fixpoint of $\Phi_E$.

Lemma 5.7 (complete progress measure as a fixpoint). *Let $L$ be a continuous lattice and let $E$ be a system of $m$ equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. Then the least solution $R_M$ of the progress measure equations (least fixpoint of $\Phi_E$ with respect to $\preceq$) is the least $\lambda_L$-progress measure, hence it is smaller than $R_E$ and it is complete.*

Proof. We first observe that $\lambda_L$-progress measures $R$ are all and only pre-fixpoints of $\Phi_E$. This implies that $R_M$, which is the least pre-fixpoint, is the least progress measure.

In fact, if $R$ is a pre-fixpoint, i.e., for all $b \in B_L$, $i \in \underline{m}$, $\Phi_E(R)(b)(i) \preceq R(b)(i)$, which implies $\Phi_E(R)(b)(i) \preceq_i R(b)(i)$. Then, for $b \in B_L$ and $i \in \underline{m}$, if $R(b)(i) \neq \star$, necessarily $\Phi_E(R)(b)(i) \neq \star$. Hence we can take $\boldsymbol{l} \in \mathrm{E}(b, i)$ that realises the minimum in the definition of $\Phi_E(R)(b)(i)$, namely such that $\Phi_E(R)(b)(i) = \sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b', j) \in \mathrm{A}(\boldsymbol{l})\}$ and we have that for all $(b', j) \in \mathrm{A}(\boldsymbol{l})$

$$R(b)(i) \succeq_i \Phi_E(R)(b)(i) \succeq R(b')(j) + \boldsymbol{\delta}_i^{\eta_i}$$

which amounts to the validity of the progress measure property (it gives strict inequality for $\eta_i = \mu$ and general inequality for $\eta_i = \nu$).

Conversely, let $R$ be a progress measure. We have to show that for all $b \in B_L$, $i \in \underline{m}$

$$R(b)(i) \succeq_i \min_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b', j) \in \mathrm{A}(\boldsymbol{l})\} \mid \boldsymbol{l} \in \mathrm{E}(b, i)\} \tag{7}$$

Given $b \in L$, $i \in \underline{m}$, by definition of progress measure, there is $\boldsymbol{l} \in \mathrm{E}(b, i)$ such that for all $(b', j) \in \mathrm{A}(\boldsymbol{l})$, it holds $R(b)(i) \succeq_i R(b')(j)$, with strict inequality if $\eta_i = \mu$. This can be equivalently stated $R(b)(i) \succeq_i R(b')(j) + \boldsymbol{\delta}^{\eta_i}$. Hence $R(b)(i) \succeq_i \sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b', j) \in \mathrm{A}(\boldsymbol{l})\}$. Namely, $R(b)(i)$ is larger than an element of the set of which we take the minimum, hence (7) immediately follows. Since in the right-hand side all entries with an index below $i$ are 0, we even have $\succeq$ (instead of $\succeq_i$ in (7), which implies that $R$ is a pre-fixpoint of $\Phi_E$.

For completeness, recall that by Lemma 5.5, $R_E$ is a $\lambda_L$-progress measure and it is complete. Therefore for all $b \in B_L$ and $i \in \underline{m}$, we have $R_M(b)(i) \preceq R_E(b)(i)$, from which completeness of $R_M$ immediately follows. □

Observe that, since $R_M \preceq R_E$, in particular, for all $b \in B_L$ and $i \in \underline{m}$, if $R_M(b)(i) \neq \star$, then all components of $R_M(b)(i)$ corresponding to $\nu$-indices are 0.

*Example 5.8.* If we consider the system of equations of Example 3.7 we obtain as least fixpoint the progress measure $R_M(\{a\})(1) = (1, 0)$ while $R_M(\{a\})(2) = R_M(\{b\})(1) = R_M(\{b\})(2) = (0, 0)$. Note that $R_M$ never assumes the top value $\star$, consistently with the fact that the solution is $(u_1, u_2) = (\mathbb{S}, \mathbb{S})$. We will discuss how $R_M$ is obtained later when providing a more "efficient" way for computing it.

We next observe that the operator $\Phi_E$ creates monotone functions and, applied to functions that respect joins, it produces functions enjoying the same property. We first introduce the formal definitions.

*Definition 5.9 (monotonicity and sup-respecting).* Let $L$ be a lattice. A function $R \colon B_L \to \underline{m} \to [\lambda_L]_\star^m$ is *monotone* if for all $b, b' \in L$, $i \in \underline{m}$, if $b \sqsubseteq b'$ then $R(b)(i) \preceq R(b')(i)$. It is *sup-respecting* if for all $b \in B_L$ and $X \subseteq B_L$, if $b \sqsubseteq \bigsqcup X$ then $R(b)(i) \preceq \sup\{R(b')(i) \mid b' \in X\}$.

Note that the notion of monotonicity for $R \colon B_L \to \underline{m} \to [\lambda_L]_\star^m$ is the standard one, with respect to the pointwise order on $\underline{m} \to [\lambda_L]_\star^m$.

Observe that $R$ is defined only on the basis elements, which are possibly (and typically) not closed under joins. The requirement of being sup-respecting ensures that $R$ extends to a function on $L$ which preserves joins. Also note that a sup-respecting function $R$ is always monotone.

Lemma 5.10 ($\Phi_E(R)$ is monotone). *Let $L$ be a lattice and let $E$ be a system of $m$ equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. For every function and $R \colon B_L \to \underline{m} \to [\lambda_L]_\star^m$, the function $\Phi_E(R)$ is monotone.*

Proof. Given $b \sqsubseteq b'$ we have to show that $\Phi_E(R)(b)(i) \preceq \Phi_E(R)(b')(i)$. Note that $\mathbf{E}(b, i) = \{\boldsymbol{l} \mid b \sqsubseteq f_i(\boldsymbol{l})\} \supseteq \{\boldsymbol{l} \mid b' \sqsubseteq f_i(\boldsymbol{l})\} = \mathbf{E}(b', i)$, hence in order to determine $\Phi_E(R)(b)(i)$ we take the $\min_{\preceq_i}$ over a larger set, resulting in a smaller vector of ordinals than for $\Phi_E(R)(b')(i)$. □

The fact that $\Phi_E$ preserves sup-respecting functions is proved in Lemma C.2 in Appendix C.1.

## 5.3 Computing Progress Measures

*5.3.1 Selections.* In principle, at least on finite lattices, the previous results allow one to compute the progress measure and thus to prove properties of the solutions of the system of equations. However, the computation can be quite inefficient due to the fact that the existential player has a (uselessly) large number of possible moves. In fact, given a system $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$ on a lattice $L$, from a position $(b, i)$, given any move $\boldsymbol{l} \in \mathbf{E}(b, i)$ for player $\exists$, i.e., any tuple such that $b \sqsubseteq f_i(\boldsymbol{l})$, it is immediate to see that all $\boldsymbol{l}'$ such that $\boldsymbol{l} \sqsubseteq \boldsymbol{l}'$ are valid moves for $\exists$, since by monotonicity of $f_i$ we have $b \sqsubseteq f_i(\boldsymbol{l}) \sqsubseteq f_i(\boldsymbol{l}')$. In other words, $\mathbf{E}(b, i)$ is upward-closed. However, player $\exists$, in order to win, has to try to descend as much as possible, hence playing large elements is inconvenient.

We next introduce some machinery that formalises the above intuition and allows us to make the calculation more efficient. The idea is discussed for a single function first, and then for a system of equations. For this we need some additional notation. Given a monotone function $f \colon L^m \to L$ and $b \in B_L$, we write $\mathbf{E}(b, f) = \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge \ b \sqsubseteq f(\boldsymbol{l})\}$.

*Definition 5.11 (selection).* Let $L$ be a lattice. Given a monotone function $f \colon L^m \to L$, a *selection for $f$* is a function $\sigma \colon B_L \to 2^{L^m}$ such that for all $b \in B_L$ it holds $\mathbf{E}(b, f) = \uparrow\sigma(b)$. Given a system $E$ of $m$ equations on $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$, a *selection for $E$* is an $m$-tuple of functions $\boldsymbol{\sigma}$ such that, for each $i \in \underline{m}$, the function $\sigma_i$ is a selection for $f_i$.

Intuitively, a selection provides for each element of the basis and function $f_i$, a subset of the moves $\mathbf{E}(b, i)$ that are sufficient to "cover" $b$ in all possible ways. Indeed, we can show that when computing the complete progress measure $R_M$ according to the equations in Lemma 5.7, we can restrict the moves of the existential player to a selection. Dually, since the moves of the universal player $\mathbf{A}(\boldsymbol{l})$ are downward-closed and the progress measures of interest are monotone (see Lemma 5.10), we can restrict also such moves to a subset whose downward-closure is $\mathbf{A}(\boldsymbol{l})$.

Lemma 5.12 (progress measure on a selection). *Let $L$ be a continuous lattice, let $E$ be a system of equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$ and let $\boldsymbol{\sigma}$ be a selection for $E$. Moreover, for all $\boldsymbol{l} \in L^m$ let $\mathbf{A}_r(\boldsymbol{l}) \subseteq B_L \times \underline{m}$ be such that $\mathbf{A}(\boldsymbol{l}) = \{(b', i) \mid (b, i) \in \mathbf{A}_r(\boldsymbol{l}) \ \wedge \ b' \sqsubseteq b\}$. The system of equations over the lattice $[\lambda_L]^m_\star$, defined, for $b \in L$, $i \in \underline{m}$, as:*

$$R(b)(i) = \min\nolimits_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b', j) \in \mathbf{A}_r(\boldsymbol{l})\} \mid \boldsymbol{l} \in \sigma_i(b)\}$$

*has the same least solution as that in Lemma 5.7.*

Proof. Let $\Phi'_E$ be the operator associated with the equations in the statement of the lemma. We prove that $\Phi_E$ and $\Phi'_E$ have the same fixpoint by showing that they coincide on monotone $R$'s.

Let $b \in B_L$ and $i \in \underline{m}$. Let us write

$$\begin{aligned}
\boldsymbol{\beta}_b = \Phi'_E(R)(b)(i) &= \min\nolimits_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b', j) \in \mathbf{A}_r(\boldsymbol{l})\} \mid \boldsymbol{l} \in \sigma_i(b)\} \\
\boldsymbol{\gamma}_b = \Phi_E(R)(b)(i) &= \min\nolimits_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b', j) \in \mathbf{A}(\boldsymbol{l})\} \mid \boldsymbol{l} \in \mathbf{E}(b, i)\}
\end{aligned}$$

and we show $\boldsymbol{\beta}_b =_i \boldsymbol{\gamma}_b$. First observe that $\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b', j) \in \mathbf{A}_r(\boldsymbol{l})\} = \sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b', j) \in \mathbf{A}(\boldsymbol{l})\}$ since $\mathbf{A}(\boldsymbol{l})$ is the downward-closure of $\mathbf{A}_r(\boldsymbol{l})$ and $R$ is monotone. Then, the fact that $\boldsymbol{\gamma}_b \preceq_i \boldsymbol{\beta}_b$ follows from the observation that, by Definition 5.11, $\sigma_i(b) \subseteq \mathbf{E}(b, i)$, i.e., the first is a minimum over a smaller set. The converse inequality follows from the fact that, by Definition 5.11, for each $b \in B_L$, $i \in \underline{m}$ if $\boldsymbol{l} \in \mathbf{E}(b, i)$ then there exists $\boldsymbol{l}' \in \sigma_i(b)$ such that $\boldsymbol{l}' \sqsubseteq \boldsymbol{l}$, hence $\mathbf{A}(\boldsymbol{l}') \subseteq \mathbf{A}(\boldsymbol{l})$ and thus $\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b', j) \in \mathbf{A}(\boldsymbol{l}')\} \preceq_i \sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b', j) \in \mathbf{A}(\boldsymbol{l})\}$ □

Since the complete progress measure $R_M$ witnesses the existence of a winning strategy for $\exists$, the above result implies that whenever $\exists$ has a winning strategy, it has one also in the game where the moves of $\exists$ are restricted to be in the selection. A similar property holds for $\forall$ and $\mathbf{A}_r(\boldsymbol{l})$.

Clearly, for computational purposes, we are interested in having the selections as small as possible. Given a monotone function $f \colon L^m \to L$, and two selections $\sigma, \sigma' \colon B_L \to 2^{L^m}$ for $f$, we write $\sigma \subseteq \sigma'$ if for all $b \in B_L$ it holds $\sigma(b) \subseteq \sigma'(b)$. We will use the same notation for the pointwise order on selections for systems of equations.

*Example 5.13 (selections for $\mu$-calculus operators).* Given a transition system $(\mathbb{S}, \to)$, consider the powerset lattice $\mathbf{2}^{\mathbb{S}}$ ordered by subset inclusion, with basis $B_{\mathbf{2}^{\mathbb{S}}} = \{\{s\} \mid s \in \mathbb{S}\}$. Then standard $\mu$-calculus operators admit a least selection, as detailed below.

- Given $f \colon (\mathbf{2}^{\mathbb{S}})^2 \to \mathbf{2}^{\mathbb{S}}$ defined by $f(X_1, X_2) = X_1 \cup X_2$, then $\sigma \colon B_{\mathbf{2}^{\mathbb{S}}} \to 2^{(\mathbf{2}^{\mathbb{S}})^2}$ is $\sigma(\{s\}) = \{(\emptyset, \{s\}), (\{s\}, \emptyset)\}$
- Given $f \colon (\mathbf{2}^{\mathbb{S}})^2 \to \mathbf{2}^{\mathbb{S}}$ defined by $f(X_1, X_2) = X_1 \cap X_2$, then $\sigma \colon B_{\mathbf{2}^{\mathbb{S}}} \to 2^{(\mathbf{2}^{\mathbb{S}})^2}$ is $\sigma(\{s\}) = \{(\{s\}, \{s\})\}$
- Given $f \colon \mathbf{2}^{\mathbb{S}} \to \mathbf{2}^{\mathbb{S}}$ defined by $f(X) = \Diamond X$, then $\sigma \colon B_{\mathbf{2}^{\mathbb{S}}} \to 2^{\mathbf{2}^{\mathbb{S}}}$ is $\sigma(\{s\}) = \{\{s'\} \mid s \to s'\}$
- Given $f \colon \mathbf{2}^{\mathbb{S}} \to \mathbf{2}^{\mathbb{S}}$ defined by $f(X) = \Box X$, then $\sigma \colon B_{\mathbf{2}^{\mathbb{S}}} \to 2^{\mathbf{2}^{\mathbb{S}}}$ is $\sigma(\{s\}) = \{\{s' \mid s \to s'\}\}$

We next provide sufficient conditions for a function to admit a least selection.

Lemma 5.14 (existence of least selections). *Let $L$ be a lattice with a basis $B_L$ and let $f \colon L^m \to L$ be a monotone function. If $f$ preserves the meet of descending chains, then it admits a least selection $\sigma_m$ that maps each $b \in B_L$ to the set of minimal elements of $\mathbf{E}(b, f)$.*

Proof. Assume that $f$ preserves the meet of descending chains. First observe that given a descending chain $(\boldsymbol{l}_\alpha)_\alpha$ in $\mathbf{E}(b, f)$ we have that $\bigsqcap_\alpha \boldsymbol{l}_\alpha \in \mathbf{E}(b, f)$. In fact, for each $\alpha$ we have $b \sqsubseteq f(\boldsymbol{l}_\alpha)$ and thus $b \sqsubseteq \bigsqcap_\alpha f(\boldsymbol{l}_\alpha) = f(\bigsqcap_\alpha \boldsymbol{l}_\alpha)$.

The above implies that for each $\boldsymbol{l} \in \mathbf{E}(b, f)$ there exists $\boldsymbol{l}' \in \mathbf{E}(b, f)$, minimal, such that $\boldsymbol{l}' \sqsubseteq \boldsymbol{l}$. In fact, consider the (possibly transfinite) chain of tuples $\boldsymbol{l}_\alpha$ in $\mathbf{E}(b, f)$ defined as follows. Start from

$\boldsymbol{l}_0 = \boldsymbol{l}$. For any ordinal $\alpha$, if there is $\boldsymbol{l}' \in \mathbf{E}(b, f)$, such that $\boldsymbol{l}' \neq \boldsymbol{l}_\alpha$ and $\boldsymbol{l}' \sqsubseteq \boldsymbol{l}_\alpha$, let $\boldsymbol{l}_{\alpha+1} = \boldsymbol{l}'$. If $\alpha$ is a limit ordinal $\boldsymbol{l}_\alpha = \bigsqcap_{\beta<\alpha} \boldsymbol{l}_\beta$.

This is a strictly descending chain, that thus necessarily stops at some ordinal $\lambda$ bounded by the length of the longest descending chain in $L$. By construction $\boldsymbol{l}_\lambda \sqsubseteq \boldsymbol{l}$, $\boldsymbol{l}_\lambda \in \mathbf{E}(b, f)$ and it is minimal in $\mathbf{E}(b, f)$.

Define $\sigma_m(b)$ as the set of minimal elements of $\mathbf{E}(b, f)$ for each $b \in B_L$. It is immediate to see that this is a selection. Moreover, it is the least selection. In fact, let $\sigma'$ be another selection for $f$. Let $\boldsymbol{l} \in \sigma_m(b)$. Since $b \sqsubseteq f(\boldsymbol{l})$ and $\sigma'$ is a selection, there is $\boldsymbol{l}' \in \sigma'(b)$ such that $\boldsymbol{l}' \sqsubseteq \boldsymbol{l}$. Now, $b \sqsubseteq f(\boldsymbol{l}')$ and thus there must be $\boldsymbol{l}'' \in \sigma_m(b)$ such that $\boldsymbol{l}'' \sqsubseteq \boldsymbol{l}'$. Therefore by transitivity $\boldsymbol{l}'' \sqsubseteq \boldsymbol{l}$, but $\boldsymbol{l}$ is minimal and thus $\boldsymbol{l} = \boldsymbol{l}' = \boldsymbol{l}'' \in \sigma'(b)$. This shows $\sigma_m(b) \subseteq \sigma'(b)$ for all $b \in B_L$. Thus $\sigma_m \subseteq \sigma'$, as desired. □

*Example 5.15.* Consider our running example in Example 3.7. Minimal selections for the functions $f_1$ and $f_2$ associated with the first and second equation are given by

- $\sigma_1(\{a\}) = \{(\{a\}, \emptyset), (\{b\}, \emptyset)\}$ and $\sigma_1(\{b\}) = \{(\emptyset, \emptyset)\}$;
- $\sigma_2(\{a\}) = \{(\{a\}, \{a, b\})\}$ and $\sigma_2(\{b\}) = \{(\{b\}, \{b\})\}$.

Observe that the winning strategy for $\exists$ discussed in Example 4.2 is a subset of the selection. We already noticed that this is a general fact: if a winning strategy exists, we can find one that is a subset of any given selection.

Selections can be constructed "compositionally", i.e., if a function $f$ arises as the composition of some component functions then we can derive a selection for $f$ from selections of the components. The details are presented in Appendix C.2.

*5.3.2 A Logic for Characterising the Moves of the Existential Player.* The set of possible moves of the existential player is an upward-closed set in the lattice. Such sets can be conveniently represented and manipulated in logical form (see, e.g., [Delzanno and Raskin 2000]). Intuitively, (minimal) selections describe a disjunctive normal form, but more compact representations can be obtained using arbitrary nesting of conjunction and disjunction. For instance, the minimal selection for the monotone function $f(X_1, \ldots, X_{2n}) = (X_1 \cup X_2) \cap (X_3 \cup X_4) \cap \cdots \cap (X_{2n-1} \cup X_{2n})$ would be of exponential size (think of the corresponding disjunctive normal form), but we can easily give a formula of linear size.

This motivates the introduction of a propositional logic for expressing the set of moves of the existential player along with a technique for deriving the fixpoint equations for computing the progress measure, avoiding the potential exponential explosion.

*Definition 5.16 (logic for upward-closed sets).* Let $L$ be a continuous lattice and let $B_L$ be a basis for $L$. Given $m \in \mathbb{N}$, the logic $\mathcal{L}_m(B_L)$ has formulae defined as follows, where $b \in B_L$ and $j \in \underline{m}$:

$$\varphi ::= [b, j] \mid \bigvee_{k \in K} \varphi_k \mid \bigwedge_{k \in K} \varphi_k$$

We will write *true* for the empty conjunction. The semantics of a formula $\varphi$ is an upward-closed set $[\![\varphi]\!] \subseteq L^m$, defined as follows:

$$\begin{array}{rcl} [\![[b,j]]\!] & = & \{\boldsymbol{l} \in L^m \mid b \sqsubseteq l_j\} \\ [\![\bigvee_{k\in K} \varphi_k]\!] & = & \bigcup_{k\in K} [\![\varphi_k]\!] \\ [\![\bigwedge_{k\in K} \varphi_k]\!] & = & \bigcap_{k\in K} [\![\varphi_k]\!] = \{ \bigsqcup_{k\in K} m_k \mid m_k \in [\![\varphi_k]\!], k \in K\} \\ & = & \{ \bigsqcup_{k\in K} f(k) \mid f : K \to L \ \wedge \ \forall k \in K.\, f(k) \in \ [\![\varphi_k]\!]\} \end{array}$$

The last equality in the definition above holds since every formula represents an upward-closed set.

It is easy to see that indeed each upward-closed set is denoted by a formula, showing that the logic is sufficiently expressive.

Lemma 5.17 (formulae for upward-closed sets). *Let $L$ be a continuous lattice with basis $B_L$ and let $X \subseteq L^m$ be upward-closed. Then $X = [\![\varphi]\!]$ where $\varphi$ is the formula in $\mathcal{L}_m(B_L)$ defined as follows:*

$$\varphi = \bigvee_{\boldsymbol{l}\in X} \bigwedge \{[b,j] \mid j \in \underline{m} \ \wedge \ b \sqsubseteq l_j\}.$$

Proof. We have to show that $[\![\varphi]\!] = X$:

- ($\subseteq$) Let $\boldsymbol{l}' \in [\![\varphi]\!]$, hence
$$\boldsymbol{l}' \in \bigcup_{\boldsymbol{l}\in X} \bigcap \{\{\boldsymbol{l}'' \in L^m \mid b \sqsubseteq l''_k\} \mid k \in \underline{m} \wedge b \sqsubseteq l_k\}.$$
Hence there exists $\boldsymbol{l} \in X$ such that for all $j \in \underline{m}$ and $b \sqsubseteq l_j$ it holds that $b \sqsubseteq l'_j$. Then
$$l_j = \bigsqcup\{b \mid b \sqsubseteq l_j\} \sqsubseteq \bigsqcup\{b \mid b \sqsubseteq l'_j\} = l'_j.$$
Hence $\boldsymbol{l} \sqsubseteq \boldsymbol{l}'$ and since $X$ is upward-closed $\boldsymbol{l}' \in X$.
- ($\supseteq$) Let $\boldsymbol{l} \in X$. We show that $\boldsymbol{l} \in [\![\psi_{\boldsymbol{l}}]\!]$ where $\psi_{\boldsymbol{l}} = \bigwedge \{[b,j] \mid j \in \underline{m} \ \wedge \ b \sqsubseteq l_j\}$. In fact
$$[\![\psi_{\boldsymbol{l}}]\!] = \bigcap \{\{\boldsymbol{l}' \in L^m \mid b \sqsubseteq l'_j\} \mid j \in \underline{m} \wedge b \sqsubseteq l_j\}.$$
Now, if $j \in \underline{m}$ and $b \sqsubseteq l_j$ then clearly $\boldsymbol{l} \in \{\boldsymbol{l}' \mid b \sqsubseteq l'_j\}$ and hence $\boldsymbol{l}$ is contained in the intersection.

$\square$

For practical purposes we should restrict to finite formulae. This can surely be done in the case of finite lattices, but also for well-quasi orders (see, e.g., [Delzanno and Raskin 2000]).

*Definition 5.18 (symbolic $\exists$-moves).* Let $L$ be a continuous lattice and let $f : L^m \to L$ be a monotone function. A *symbolic $\exists$-move* for $f$ is a family $(\varphi_b)_{b\in B_L}$ of formulae in $\mathcal{L}_m(B_L)$ such that $[\![\varphi_b]\!] = \mathbf{E}(b, f)$ for all $b \in B_L$.

If $E$ is a system of $m$ equations of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$ over a continuous lattice $L$, a *symbolic $\exists$-move* for $E$ is a family of formulae $(\varphi^i_b)_{b\in B_L, i\in \underline{m}}$ such that for all $i \in \underline{m}$, the family $(\varphi^i_b)_{b\in B_L}$ is a symbolic $\exists$-move for $f_i$.

Interestingly, symbolic $\exists$-moves can be obtained compositionally, namely, the formulae corresponding to a functions arising as a composition can be obtained from those of the components.

Lemma 5.19 (symbolic $\exists$-moves, compositionally). *Let $L$ be a continuous lattice with a basis $B_L$, and let $f : L^n \to L$, $f_j : L^m \to L$ for $j \in \underline{n}$ be monotone functions and let $(\varphi_b)_{b \in B_L}$, $(\varphi^j_b)_{b \in B_L}$, $j \in \underline{n}$ be symbolic $\exists$-moves for $f, f_1, \ldots, f_n$. Consider the function $h : L^m \to L$ obtained as the composition $h(\boldsymbol{l}) = f(f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l}))$. Define $(\varphi'_b)_{b \in B_L}$ as follows. For all $b \in B_L$, the formula $\varphi'_b$ is obtained from $\varphi_b$ by replacing each occurrence of $[b', j]$ by $\varphi^j_{b'}$. Then $(\varphi'_b)_{b \in B_L}$ is a symbolic $\exists$-move for $h$.*

Proof. We first show that given a formula $\varphi \in \mathcal{L}_n(B_L)$, if $\varphi'$ is the formula in $\mathcal{L}_m(B_L)$ obtained from $\varphi$ by replacing each occurrence of an atom $[b, j]$ by $\varphi^j_b$, then

$$[\![\varphi']\!] = \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge\ (f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l})) \in [\![\varphi]\!]\}$$

We proceed by induction on $\varphi_b$.

- ($\varphi = [b, j]$): In this case $\varphi' = \varphi^j_b$. Therefore we have

$$\begin{aligned}
[\![\varphi']\!] &= [\![\varphi^j_b]\!] \\
&= \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge\ b \sqsubseteq f_j(\boldsymbol{l})\} \\
&= \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge\ (f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l})) \in [\![[b, j]]\!]\} \\
&= \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge\ (f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l})) \in [\![\varphi]\!]\}
\end{aligned}$$

- ($\varphi = \bigvee_{k \in K} \varphi_k$): In this case $\varphi' = \bigvee_{k \in K} \varphi'_k$, where each $\varphi'_k$ is obtained from $\varphi_k$ by by replacing each occurrence of an atom $[b, j]$ by $\varphi^j_b$. Then

$$\begin{aligned}
[\![\varphi']\!] &= [\![\bigvee_{k \in K} \varphi'_k]\!] \\
&= \bigcup_{k \in K} [\![\varphi'_k]\!] \\
&= \bigcup_{k \in K} \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge\ (f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l})) \in [\![\varphi_k]\!]\} && \text{[by inductive hyp.]} \\
&= \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge\ (f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l})) \in \bigcup_{k \in K} [\![\varphi_k]\!]\} \\
&= \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge\ (f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l})) \in [\![\bigvee_{k \in K} \varphi_k]\!]\} \\
&= \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge\ (f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l})) \in [\![\varphi]\!]\}
\end{aligned}$$

  as desired.
- ($\varphi = \bigwedge_{k \in K} \varphi_k$): Analogous.

Now, given $b \in B_L$, we have to show that

$$[\![\varphi'_b]\!] = \mathbf{E}(b, h) = \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge\ b \sqsubseteq h(\boldsymbol{l})\} = \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge\ b \sqsubseteq f(f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l}))\}.$$

This is almost immediate. In fact

$$\begin{aligned}
&[\![\varphi'_b]\!] = \\
&\quad = \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge\ (f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l})) \in [\![\varphi_b]\!]\} && \text{[by the property proved above]} \\
&\quad = \{\boldsymbol{l} \mid \boldsymbol{l} \in L^m \ \wedge\ b \sqsubseteq f(f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l}))\} && \text{[by def. of symbolic } \exists\text{-move]}
\end{aligned}$$

□

*Example 5.20.* Consider again our running example in Example 3.7. The selections specified in Example 5.15 can be expressed in the logic as follows:

$$\varphi^1_{\{a\}} = [\{a\},1] \vee [\{b\},1] \qquad \varphi^1_{\{b\}} = \mathit{true}$$
$$\varphi^2_{\{a\}} = [\{a\},1] \wedge [\{a\},2] \wedge [\{b\},2] \qquad \varphi^2_{\{b\}} = [\{b\},1] \wedge [\{b\},2]$$

These formulae can be obtained compositionally. For instance the formula $\varphi^2_{\{a\}}$ for the equation $x_2 =_\nu x_1 \cap \Box x_2$ is obtained by combining a logical formula for $x_1$ (namely $[\{a\},1]$) via conjunction with a logical formula for $\Box x_2$ (namely $[\{a\},2] \wedge [\{b\},2]$).

A symbolic $\exists$-move for a system can be directly converted into a recipe for evaluating the fixpoint expressions for progress measures. Essentially, every disjunction simply has to be replaced by a minimum and every conjunction by a supremum (although the proof, which relies on complete distributivity of the lattice $[\lambda_L]^m_\star$ is not trivial). Furthermore, in the case of an algebraic lattice, where we can ensure that the elements of the basis are compact, an atom translates to a straightforward lookup of the progress measure without additional computation.

Proposition 5.21 (progress measure from symbolic $\exists$-moves). *Let $E$ be a system of $m$ equations over a continuous lattice $L$ and let $B_L$ be a basis for $L$. Let $(\varphi^i_b)_{b\in B_L, i\in \underline{m}}$ be a symbolic $\exists$-move for $E$.*

*Then the system of fixpoint equations for computing the progress measure can be written, for all $b \in B_L$ and $i \in \underline{m}$, as $R(b)(i) = R^i_{\varphi^i_b}$ where $R^i_\psi$ is defined inductively as follows:*

$$R^i_{[b,j]} = \min\nolimits_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}^{\eta_i}_i \mid b' \ll b\}\} \qquad R^i_{\bigvee_{k\in K}\varphi_k} = \min_{k\in K} R^i_{\varphi_k} \qquad R^i_{\bigwedge_{k\in K}\varphi_k} = \sup_{k\in K} R^i_{\varphi_k}$$

*Whenever the basis element $b$ is compact it holds that $R^i_{[b,j]} = \min_{\preceq_i}\{R(b)(j) + \boldsymbol{\delta}^{\eta_i}_i\}$.*

Proof. First observe that due to Lemma C.2 $\Phi_E$ preserves sup-respecting progress measures. Furthermore the supremum of sup-respecting progress measures is again sup-respecting. This means that the fixpoint iteration generates only sup-respecting functions and we can in the following assume that $R$ is sup-respecting.

Since $(\varphi^i_b)_{b\in B_L, i\in\underline{m}}$ is a symbolic $\exists$-move for $E$, the equations of Definition 5.6 can be written as

$$R(b)(i) = \min\nolimits_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}^{\eta_i}_i \mid (b',j) \in \mathrm{A}(\boldsymbol{l})\} \mid \boldsymbol{l} \in [\![\varphi^i_b]\!]\}.$$

We conclude by proving that, when $R$ is monotonic

$$R^i_\psi = \min\nolimits_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}^{\eta_i}_i \mid (b',j) \in \mathrm{A}(\boldsymbol{l})\} \mid \boldsymbol{l} \in [\![\psi]\!]\}.$$

We proceed by induction on the structure of $\psi$.

- ($\psi = [b,k]$): By definition

$$\min\nolimits_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}^{\eta_i}_i \mid (b',j) \in \mathrm{A}(\boldsymbol{l})\} \mid \boldsymbol{l} \in [\![[b,k]]\!]\} =$$
$$\min\nolimits_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}^{\eta_i}_i \mid j \in \underline{m} \ \wedge\ b' \ll l_j\} \mid \boldsymbol{l} \in L^m \ \wedge\ b \sqsubseteq l_k\}$$

  A vector $\boldsymbol{l} \in L^m$ satisfying $b \ll l_k$ has the form $(l_1,\ldots,l_m)$ where $l_j$ is arbitrary if $j \neq k$ and $b \ll l_k$. Since we can assume that $R$ is monotonic and hence the inner supremum is monotone in $\boldsymbol{l}$, we can conclude that the minimum is reached for a vector $\boldsymbol{\ell}$ where $l_j = \bot$ if $j \neq k$ and $b \sqsubseteq l_k$. Hence we obtain

$$\min\nolimits_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}^{\eta_i}_i \mid j \in \underline{m} \ \wedge\ b' \ll l_j\} \mid \boldsymbol{l} \in L^m, b \sqsubseteq l_k, l_j = \bot \text{ if } j \neq k\}.$$

  Since there is no basis element $b'$ with $b' \ll \bot$, it is sufficient if one takes the inner suprema only for elements with $j = k$ and $b' \sqsubseteq l_k$. And so we obtain

$$\min\nolimits_{\preceq_i}\{\sup\{R(b')(k) + \boldsymbol{\delta}^{\eta_i}_i \mid b' \ll l\} \mid l \in L \ \wedge\ b \sqsubseteq l\}$$

We can now infer that $b$ is the least value $l \in L$ such that $b \sqsubseteq l$ and hence – again by monotonicity – the above can be rewritten as

$$\min_{\leq_i}\{\sup\{R(b')(k) + \boldsymbol{\delta}_i^{\eta_i} \mid b' \ll b\}\}$$

which is exactly $R^i_{[b,k]}$, as desired.
If $b$ is compact, we know that $b$ itself is the least element of all $l$ such that $b \ll l$ and we can write the above as

$$R^i_{[b,k]} = \min_{\leq_i}\{R(b)(k) + \boldsymbol{\delta}_i^{\eta_i}\}.$$

- Disjunction:

$$\begin{aligned} R^i_{\bigvee_{k\in K}\varphi_k} &= \min_{\leq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid j \in \underline{m} \ \wedge\ b' \ll l_j\} \mid \boldsymbol{l} \in \bigcup_{k\in K}[\![\varphi_k]\!]\} \\ &= \min_{\leq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid j \in \underline{m} \ \wedge\ b' \ll l_j\} \mid \boldsymbol{l} \in [\![\varphi_k]\!], k \in K\} \\ &= \min_{k\in K}\min_{\leq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid j \in \underline{m} \ \wedge\ b' \ll l_j\} \mid \boldsymbol{l} \in [\![\varphi_k]\!]\} \\ &= \min_{k\in K} R^i_{\varphi_k} \end{aligned}$$

- Conjunction: since every set $[\![\varphi_k]\!]$ is upward-closed we can immediately apply Lemma C.1 and obtain

$$\begin{aligned} R^i_{\bigwedge_{k\in K}\varphi_k} &= \min_{\leq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid j \in \underline{m} \ \wedge\ b' \ll l_j\} \mid \boldsymbol{l} \in \bigcap_{k\in K}[\![\varphi_k]\!]\} \\ &= \sup_{k\in K}\min_{\leq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid j \in \underline{m} \ \wedge\ b' \ll l_j\} \mid \boldsymbol{l} \in [\![\varphi_k]\!]\} \\ &= \sup_{k\in K} R^i_{\varphi_k} \end{aligned}$$

□

Note that the operator $\min_{\leq_i}$ in the definition of $R^i_{[b,j]}$ above is just there to ensure that all entries in positions smaller than $i$ are set to 0.

*Example 5.22.* Using the logical formulae from Example 5.20, we obtain the following equations for the progress measure (where $\max_{\leq_i}$ works analogously to $\min_{\leq_i}$: it sets all vector entries in positions smaller than $i$ to 0):

$R(\{a\})(1) = \min_{\leq_1}\{R(\{a\})(1) + (1,0), R(\{b\})(1) + (1,0)\}$ $\quad R(\{b\})(1) = (0,0)$

$R(\{a\})(2) = \max_{\leq_2}\{R(\{a\})(1), R(\{a\})(2), R(\{b\})(2)\}$ $\quad R(\{b\})(2) = \max_{\leq_2}\{R(\{b\})(1), R(\{b\})(2)\}$

The solution for the progress measure equations has already been given in Example 5.8.

*5.3.3 Complexity Analysis.* The benefit of the progress measures introduced in [Jurdziński 2000] is to ensure that model-checking is polynomial in the number of states and exponential in (half of) the alternation depth. We will now perform a corresponding complexity analysis for our setting, based on symbolic $\exists$-moves and by assuming that we are working on a finite lattice.

Let $E$ be a fixed system of $m$ equations over a finite lattice $L$, let $k$ be the number of $\mu$-equations and let $B_L$ be a basis for $L$. Let $(\varphi^i_b)_{b\in B_L, i\in\underline{m}}$ be a symbolic $\exists$-move for $E$ and assume that the size of every such formula is bounded by $s$. Note that the formulae are typically of moderate size. For instance, $\mu$-calculus model-checking, the branching of a transition system (i.e., the number of successors of a single state) is a determining factor. In fact, as it can be grasped from our running example (see Example 5.20), the size of the symbolic $\exists$-move $\varphi^i_b$ will be linear in the number of propositional operators and, in the presence of modal operators, linear in the branching degree of the transition system. For arbitrary monotone functions it is more difficult to give a general rule.

The shape of the formulae in the symbolic $\exists$-move determine how the values of the progress measure at various positions $(b, i)$ of the games are interrelated. These dependencies clearly play a role in the computation and thus are made explicit by following definition.

*Definition 5.23 (dependency graph).* Given two game positions $(b, i), (b', j) \in B_L \times \underline{m}$ of $\exists$ we say that $(b, i)$ is a *predecessor* of $(b', j)$ if $[b', j]$ occurs in $\varphi_b^i$. We will write *pred*$(b', j)$ for the set of predecessors of $(b', j)$. In this situation we will also call the pair $((b, i), (b', j))$ an *edge* and refer to corresponding graph as the *dependency graph* for $E$.

As a first step we provide a bound to the number of edges in the dependency graph.

Proposition 5.24 (edges in the dependency graph). *The number $e$ of edges in the dependency graph for system $E$ is such that $e \leq \min\{|B_L| \cdot m \cdot s, (|B_L| \cdot m)^2\}$, where $m$ is the number of equations and $s$ is the bound on the size of symbolic $\exists$-moves.*

Proof. There are at most $|B_L| \cdot m$ game positions and hence the number of edges is obviously bounded by $(|B_L| \cdot m)^2$. Moreover, each game position, the number of outgoing edges is bounded by the size of the formula (symbolic $\exists$-move) associate to the position. Hence the thesis. □

In order to bound the complexity of the overall computation of the progress measure, first note that the lattice $[\lambda_L]_\star^m$ contains $(\lambda_L + 1)^m + 1$ elements. However only $h = (\lambda_L + 1)^k + 1$ are relevant, since the entries of $\nu$-indices are always set to 0. As an example, when model-checking a $\mu$-calculus formula over a finite state system, $\lambda_L$ is the size of the state space of the Kripke structure. In fact, the lattice is $(\mathbf{2}^{\mathbb{S}}, \subseteq)$ where $\mathbb{S} = \{s_0, \ldots, s_n\}$ is the state space, then the longest ascending chain is $\emptyset \subseteq \{s_0\} \subseteq \{s_0, s_1\} \subseteq \ldots \subseteq \mathbb{S}$.

This fact and the observation that we can perform the fixpoint iteration for the progress measure using a worklist algorithm on the dependency graph, lead to the following result.

Theorem 5.25 (computing progress measures). *The time complexity for computing the least fixpoint progress measure for system $E$ is $O(s \cdot k \cdot e \cdot h)$, where $s$ is the bound on the size of symbolic $\exists$-moves, $k$ is the number of $\mu$-equations, $e$ the number of edges in the dependecy graph, and $h = (\lambda_L + 1)^k + 1$.*

Proof. We use a worklist algorithm, and the worklist initially contains all edges.

Processing an edge $((b, i), (b', j))$ means to update the value $R(b)(i)$ by evaluating the formula $\varphi_b^i$. Afterwards all edges originating from $(b, i)$ can be removed from the worklist. Whenever a value $R(b')(j)$ increases, all edges $((b, i), (b', j))$ with $(b, i) \in$ *pred*$(b', j)$ will be again inserted into the worklist. Hence, at most $\sum_{(b',j) \in B_L \times \underline{m}} h \cdot pred(b', j) = e \cdot h$ edges will be inserted into the worklist and processed later.

In turn, processing an edge has complexity at most $O(s \cdot k)$, since we inductively evaluate a formula of size $s$ on ordinal vectors of length $k$. (Since the lattice is finite, it is automatically algebraic and the simpler case for compact elements of Theorem 5.21 applies.) Everything combined, we obtain a runtime of $O(s \cdot k \cdot e \cdot h)$. □

We compare the above with the runtime in [Jurdziński 2000], which is $O(dg\left(\frac{n}{d}\right)^{\lceil \frac{d}{2} \rceil})$, where $d$ is the alternation depth of the formula, $g$ the number of edges and $n$ the number of nodes of the parity game.

The correspondence is as follows: $g$ corresponds to our number $e$ and $n$ to $\lambda_L$ (where we cannot exploit the optimisation by Jurdziński which uses the fact that every node in the parity game is associated with a single parity, leading to the division by $d$). Furthermore $s$ is a new factor, which is due to the fact that we are working with arbitrary functions. But this is mitigated by the fact that we often obtain smaller parity games than in the standard $\mu$-calculus case (see for instance Example 4.2,

Figure 4). The number $\frac{d}{2}$ corresponds to our $k$. However $\frac{d}{2}$ could potentially be strictly lower than $k$, since we did not take into account the fact that some equations might not be dependent on other equations.

To incorporate this and possibly further optimisations into the complexity analysis we need a notion of alternation depth for equation systems. This can be easily obtained by extending the one introduced in [Cleaveland et al. 1992; Schneider 2004]. A system of equations can be split into closed subsystems corresponding to the strongly connected components of the dependency graph for the system. Then the alternation depth of the system is defined as the length of the longest chain of mutually dependent $\mu$ and $\nu$-equations within a closed subsystem. By solving every component separately we obtain a more efficient algorithm.

In particular, systems of fixpoint equations that consist only of $\mu$-equations or $\nu$-equations can be solved by a single fixpoint iteration on $L^m$, where $m$ is the number of equations [Venema 2008]. Similarly, equations with indices $i, i+1$ where $\eta_i = \eta_{i+1}$ can be merged. This results in an equation system where subsequent equations alternate between $\mu$ and $\nu$. (Notice that this transformation means that the equations are over $L^j$ instead of $L$, but this can be easily adapted in our setting.)

Note also that the runtime might be substantially improved by finding a good strategy for computing the progress measure, as spelled out in [Jurdziński 2000], in the same way as efficient ways can be found for implementing the worklist algorithm in program analysis.

## 6 MODEL-CHECKING LATTICED $\mu$-CALCULI

As explained earlier, model-checking for $\mu$-calculus formulae can be reduced to solving fixpoint equations over the powerset lattice $2^{\mathbb{S}}$ where $\mathbb{S}$ is the state space of the system under consideration. A state $x \in \mathbb{S}$ can either satisfy or not satisfy a formula, meaning it either belongs to the solution or not. However, there are also multi-valued logics for modelling uncertainty, disagreement or relative importance in which it is natural to have "non-binary" truth values (see, e.g., [Eleftheriou et al. 2012; Fitting 1991; Grumberg et al. 2005; Kupfermann and Lustig 2007]). Such a setting, as detailed later, can also be used to model and verify conditional (or featured) transition systems with upgrades. Here we discuss latticed $\mu$-calculi, inspired by the work cited above, and discuss a corresponding model checking procedure.

A lattice of truth values $L$ is fixed, which is typically finite. and then formulae are evaluated over the lattice $L^{\mathbb{S}}$, endowed with the pointwise order. Also transitions are associated with an element in the lattice of truth values.

*Definition 6.1 (multi-valued transition system).* A *multi-valued transition system* over $L$ is a function $R \colon \mathbb{S} \times \mathbb{S} \to L$, where $\mathbb{S}$ is the set of states.

Since $L$ can be non-boolean, multi-valued modal logics express forms of negation or implication by relying on *residuation* or *relative pseudo-complement* operation which is well defined for all complete lattices $L$.

*Definition 6.2 (residuation).* Let $L$ be a lattice. Given $l, m \in L$, we define $(l \Rightarrow m) = \bigsqcup\{l' \in L \mid l \sqcap l' \sqsubseteq m\}$.

Latticed $\mu$-calculi use atoms, conjunction, disjunction and residuation. The modal operators $\Diamond$ and $\Box$ are interpreted as follows. Given $u \in L^{\mathbb{S}}$ we define $\Diamond u, \Box u \in L^{\mathbb{S}}$ as

$$(\Diamond u)(x) = \bigsqcup_{y \in \mathbb{S}} (R(x,y) \sqcap u(y)) \qquad\qquad (\Box u)(x) = \bigsqcap_{y \in \mathbb{S}} (R(x,y) \Rightarrow u(y))$$

The approach discussed in § 3.2 for model-checking the $\mu$-calculus can be easily adapted to this setting. Instead of the powerset lattice we now have $L^{\mathbb{S}}$ and, as a basis $B_{L^{\mathbb{S}}}$ we can take the functions $b_x \in B_{L^{\mathbb{S}}}$, with $x \in \mathbb{S}$, $b \in B_L$, defined by $b_x(x) = b$ and $b_x(y) = \bot$ for all $y \neq x$.

In order to perform the calculation of the progress measure efficiently, we use symbolic $\exists$-moves as defined in § 5.3.2. Here we assume that $L$ is a finite distributive lattice. In this case $\ll$ and $\sqsubseteq$ coincide. Moreover, for finite distributive lattice it is is well-known from the Birkhoff duality (see also [Davey and Priestley 2002]) that every element can be uniquely represented as the join of a downward-closed set of join-irreducibles. Note that if $B_L$ is the set of join-irreducibles in $L$, then the basis $B_{L^{\mathbb{S}}} = \{b_x \mid x \in \mathbb{S}, b \in B_L\}$, given above is the set of join-irreducibles of $L^{\mathbb{S}}$.

Proposition 6.3 (symbolic $\exists$-moves in latticed $\mu$-calculi). *Let $L$ be a finite distributive lattice, let $B_L$ be the set of its join-irreducibles. The following are symbolic $\exists$-moves for the semantic functions:*

- *For $\sqcup\colon L^{\mathbb{S}} \times L^{\mathbb{S}} \to L^{\mathbb{S}}$, we let $\psi^{\sqcup}_{b_x} = [b_x, 1] \vee [b_x, 2]$.*
- *For $\sqcap\colon L^{\mathbb{S}} \times L^{\mathbb{S}} \to L^{\mathbb{S}}$, we let $\psi^{\sqcap}_{b_x} = [b_x, 1] \wedge [b_x, 2]$.*
- *For $l \Rightarrow \_\colon L^{\mathbb{S}} \to L^{\mathbb{S}}$ (where $l \in L$ is fixed and seen as a constant function $\mathbb{S} \to L$), we let $\psi^{\Rightarrow}_{b_x} = \bigwedge\{[b'_x, 1] \mid b' \sqsubseteq l \ \wedge \ b' \sqsubseteq b\}$.*
- *For $\Diamond\colon L^{\mathbb{S}} \to L^{\mathbb{S}}$ we let $\psi^{\Diamond}_{b_x} = \bigvee\{[b_y, 1] \mid y \in Y \ \wedge b \sqsubseteq R(x, y)\}$*
- *For $\Box\colon L^{\mathbb{S}} \to L^{\mathbb{S}}$ we let $\psi^{\Box}_{b_x} = \bigwedge\{[b'_y, 1] \mid y \in Y \ \wedge \ b' \sqsubseteq R(x, y) \ \wedge b' \sqsubseteq b\}$.*

Proof. We will only consider two cases, since the remaining ones cases are analogous.

- *$\sqcup$:* Let $u_1, u_2 \in L^{\mathbb{S}}$. Since $b$ and hence $b_x$ are join-irreducibles, it holds that $b_x \sqsubseteq u_1 \sqcup u_2$ iff $b_x \sqsubseteq u_1$ or $b_x \sqsubseteq u_2$. Hence we can define
$$\mathbf{E}(b_x, \sqcup) = \{(u_1, u) \mid b_x \sqsubseteq u_1, u \in L^{\mathbb{S}}\} \cup \{(u, u_2) \mid b_x \sqsubseteq u_2, u \in L^{\mathbb{S}}\} = [\![\psi_{b_x}]\!].$$
- *$\Box$-operator:* Let $u \in L^{\mathbb{S}}$. It holds that
$$\begin{aligned} & b_x \sqsubseteq \Box u \\ \iff\quad & b \sqsubseteq (\Box u)(x) = \bigsqcap_{y \in \mathbb{S}} (R(x, y) \Rightarrow u(y)) \\ \iff\quad & \text{for all } y \in \mathbb{S}\text{: } b \sqsubseteq (R(x, y) \Rightarrow u(y)) \\ \iff\quad & \text{for all } y \in \mathbb{S}\text{: } b \sqcap R(x, y) \sqsubseteq u(y) \\ \iff\quad & \text{for all } y \in \mathbb{S}, b' \in B_L \text{ with } b' \sqsubseteq R(x, y), b' \sqsubseteq b\text{: } b' \sqsubseteq u(y) \\ \iff\quad & \text{for all } y \in \mathbb{S}, b' \in B_L \text{ with } b' \sqsubseteq R(x, y), b' \sqsubseteq b\text{: } b'_y \sqsubseteq u \end{aligned}$$
Note that we used that $(l \Rightarrow m)$ is the maximal element in the downward-closed set $\{l' \in L \mid l \sqcap l' \sqsubseteq m\}$, which holds for distributive lattices.
Hence can define $\mathbf{E}(b_x, \Box) = \{u \in L^{\mathbb{S}} \mid b_x \sqsubseteq \Box u\} = [\![\psi_{b_x}]\!]$.

$\square$

Note that residuation is only monotone in the second argument and that distributivity is essential for this definition of symbolic $\exists$-moves. For instance, if $b$ is not a join-irreducible then $b \sqsubseteq l_1 \sqcup l_2$ is not equivalent to $b \sqsubseteq l_1 \vee b \sqsubseteq l_2$.

*Example 6.4 (conditional transition systems with upgrades).* An interesting special case are conditional transition systems with upgrades [Beohar et al. 2017] for which a logic satisfying the Hennessy-Milner property has been studied in [Poltermann 2017]. This logic uses the operators given above, enriched with constants. This kind of systems extend the well-known featured transition systems for modelling software product lines [Cordy et al. 2012] by upgrades.

Let $(P, \leq)$ be a given partial order where $P$ is the set of products and $\leq$ is the upgrade relation. If $p \leq q$, it is possible to make an upgrade from $q$ to $p$ during the runtime of the system, i.e., $p$ is the more advanced product compared to $q$. We consider the lattice $L = (\mathcal{O}(P), \sqsubseteq)$, where $\mathcal{O}(P)$ is the set of all downward-closed subsets of $P$. (In fact the sets $\downarrow p$, for $p \in P$, where $\downarrow$ denotes downward-closure, are the join-irreducibles of $L$.) A transition system that compactly specifies the system behaviour for all possible products is given by $R\colon \mathbb{S} \times \mathbb{S} \to \mathcal{O}(P)$ where $p \in R(x, y)$ means that there exists a transition from $x$ to $y$ if one is in possession of product $p$. More advanced products lead to additional transitions, due to the downward-closure. It is possible to spontaneously perform upgrades during runtime.

Now one can study the latticed modal logic or latticed $\mu$-calculus arising in such a setting. Evaluating a formula $\varphi$ yields a function $\|\varphi\|\colon \mathbb{S} \to \mathcal{O}(P)$ which intuitively gives us for every state those products on which $\varphi$ holds (taking upgrades into account).

The approach outlined in the first part of the section can be directly used for model checking the Hennessy-Milner logic on product lines. Note that, as it happens in this case, the lattice $L$ of truth values can have a considerable size and thus the availability of general approaches for handling latticed $\mu$-calculi can be of great help.

## 7 SOLVING FIXPOINT EQUATIONS OVER INFINITE LATTICES

We present some initial but promising results concerning the solution of fixpoint equations in infinite lattices. We will mainly concentrate on equations over the real interval $[0, 1]$, as considered also in [Mio and Simpson 2017] as a precursor to model-checking PCTL or probabilistic $\mu$-calculi. We adapt our fixpoint game in a way that it can be encoded into a finite first-order formula. If this formula is in a decidable fragment – such as linear arithmetic – we can use an SMT solver to determine its satisfiability. In this way one can either check that a value is smaller or equal than the solution or even let the SMT solver calculate the solution.

The starting observation is that the existence of a winning strategy for player $\exists$ in the game can be expressed as a first-order formula with nested quantifiers (existential quantifiers for the $\exists$ player, universal quantifiers for the $\forall$ player). However, the formula is in general of infinite size, since plays are unbounded or even infinite. Starting from $(b, i)$ the formula would be of the kind $\exists \boldsymbol{l}_0 \in \mathbf{E}(b, i).\, \forall (b_0, i_0) \in \mathbf{A}(\boldsymbol{l}_0).\, \exists \boldsymbol{l}_1 \in \mathbf{E}(b_0, i_0).\, \forall (b_1, i_1) \in \mathbf{A}(\boldsymbol{l}_0).\, \ldots$ In order to make the formula finite we need a stopping condition: if an equation index is visited for the second time (without any higher index in between), we make sure that the game can be continued if we jump back to the previous occurrence of the index. For $\nu$-indices this simply amounts to checking that the tuple's values seen at the two occurrences are in the $\sqsubseteq$-relation. For $\mu$-indices the situation is more complicated: ensuring that we can cycle on that equation is not sufficient (since by continuing forever $\exists$ would lose) and thus we have to provide a proof that the values truly decrease and that we will eventually reach $\bot$. We will see that for lattices based on a well-founded order, it is sufficient to require a strict inequality, for lattices which do not enjoy this property (such as the real interval $[0, 1]$), we have to find a different condition. In fact, for the reals we will present a condition which is correct, i.e., if the formula is satisfiable, we know that the considered value is bounded by the solution. However, this method is not always complete. We will discuss the limitations of the approach and provide a preliminary characterisation of functions for which we obtain completeness.

We will first adapt our game (Definition 4.1) to a modified version, which incorporates the stopping condition mentioned above. We define a game parametrised over a predicate $\mathit{decrease}(v, b, l)$ which takes three lattice elements as parameters.

*Definition 7.1 (modified fixpoint game).* Let $L$ be a lattice and let $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$ be a system of equations over $L$ and let $\mathit{decrease} \subseteq L^3$ be a fixed predicate. The game has a state consisting of a vector $\boldsymbol{v} \in L^m$, whose entries are defined while playing the game, and the current index $j$.

The game starts on some $(v_i, i) \in L \times \underline{m}$, namely, initially $v_i$ is the only component of $\boldsymbol{v}$ which is set and the current index is set $j := i$. Player $\forall$ chooses $b_i \ll v_i$. At a generic step:

- $\exists$ chooses $\boldsymbol{l} \in \mathbf{E}(b_j, j)$
- $\forall$ chooses an index $k \in \underline{m}$ and set $j := k$. Then:
  - if the current index $j$ was already set to $k$ earlier in the play and no higher index has occurred in between, then there are two possibilities:
    - if $\eta_k = \nu$ check whether $l_k \sqsubseteq v_k$. If yes, $\exists$ wins, otherwise $\forall$ wins.
    - if $\eta_k = \mu$ check whether $l_k \sqsubseteq b_k$ and *decrease*$(v_k, b_k, l_k)$. If yes, $\exists$ wins, otherwise $\forall$ wins.
  - otherwise set $v_k := l_k$ and $\forall$ chooses $b_k \in B_L$ with $b_k \ll v_k$ (hence $(b_k, k) \in \mathbf{A}(\boldsymbol{l})$), and continue.

We can imagine the game as being played on game trees as depicted below for the case $m = 2$. In the first tree we start with index $i = 1$ and in the second with $i = 2$ and – depending on the choice of $\forall$ – we descend in the tree. Once we reach a leaf (i.e., a node with a repeated index with no larger index in between) we can stop and determine the winner of the game. Note that in the left-hand tree we need one extra level of nodes, since we cannot yet stop at the 1-node on level three, since there is a higher index (2) on the path between this node and the root.

$$
\begin{tikzcd}
& 1 \arrow[dl] \arrow[dr] & & \\
1 & & 2 \arrow[dl] \arrow[dr] & \\
& 1 \arrow[dl] \arrow[dr] & & 2 \\
1 & & 2 &
\end{tikzcd}
\qquad
\begin{tikzcd}
& & 2 \arrow[dl] \arrow[dr] & \\
& 1 \arrow[dl] \arrow[dr] & & 2 \\
1 & & 2 &
\end{tikzcd}
$$

It is possible to show that every such tree is finite, which follows from the fact that there are no infinite paths and that it is finitely branching.

Lemma 7.2 (modified plays are finite). *The game of Definition 7.1 does not admit infinite plays.*

Proof. Assume that there exists an infinite play, i.e., whenever an index is reached for the second time, there is always a higher index in between. Consider the suffix of the play which contains only indices that occur infinitely often. Assume that $k$ is the highest among those indices. Then the play will stop when $k$ is reached for the second time in the suffix, which is a contradiction. $\square$

We will next show that a winning strategy in the modified game implies a winning strategy in the original game. The basic idea is that we follow the winning strategy in the modified game and once we have reached a leaf, we "jump" back to the predecessor node with the same index and continue to follow the strategy. A crucial point is to show that the *decrease* predicate ensures that there cannot exist an infinite path where the highest index occurring infinitely often is a $\mu$-index.

*Definition 7.3 (well-foundedness).* Let *decrease* be a ternary predicate on $L$. We say that *decrease* is well-founded if there exist no $v \in L$ and $b^m, l^m \in L$ for $m \in \mathbb{N}$ such that

$$b^{m+1} \ll l^m \sqsubseteq b^m \ll v$$

and *decrease*$(v, b^m, l^m)$ for all $m$.

Intuitively we want to ensure that there is no infinite play underneath a fixed starting value $v$. Obviously, if the lattice order $\sqsubset$ is well-founded one can define *decrease*$(v, b, l) = (l \sqsubset b)$. (Or even *true* if the way-below relation $\ll$ should be well-founded.) For the real interval $[0, 1]$ we need a more sophisticated predicate, whose shape will be explained in more detail in Lemma 7.8.

Lemma 7.4. *Let $a_i \in [0, 1]$, $i \in \{0, \dots, \ell\}$ be a finite set of real constants with $a_0 = 0 < a_1 < \cdots < a_\ell$ and let $c \in [0, 1]$. Given $v, b, l \in L$ we define that decrease$(v, b, l)$ holds if*

- *$l = a_i$ for some $i \in \{0, \dots, \ell\}$*
- *or $a_\ell \le b$ and $b - l \ge c \cdot (v - b)$*
- *or $a_i \le b < a_{i+1}$ and $b - l \ge c \cdot (a_{i+1} - b)$ for some $i \in \{0, \dots, \ell\}$*

*Then decrease is a well-founded predicate.*

Proof. Assume, by contradiction, that there exist $v, b^m, l^m \in L$, $m \in \mathbb{N}_0$ such that $b^{m+1} < l^m \leq b^m < v$ and $decrease(v, b^m, l^m)$ for all $m$.

Since the sequence is infinite and strictly decreasing, it must have a suffix for which $b^m, l^m \neq a_i$ for all $i \in \{0, \ldots, \ell\}$. Hence there exists an index $n$ such that for all $m \geq n$ we have $b^m - l^m \geq c \cdot (a - b^m)$ where either $a = v$ or $a = a_i$ for some $i \in \underline{m}$. Furthermore $b^n < a$.

We show by induction on $m$ that for these indices

$$a - l^m \geq (a - b^n) \cdot (1 + (m - n + 1) \cdot c)$$

- $m = n$: we know that $b^n - l^n \geq c \cdot (a - b^n)$. By rearranging we obtain $l^n \leq b^n - c \cdot (a - b^n)$. We subtract both sides of the inequality from $a$ and get $a - l^n \geq a - b^n + c \cdot (a - b^n) = (a - b^n) \cdot (1 + c)$.
- $m - 1 \to m$: We have

$$\begin{array}{rcl}
a - l^m & = & a - l^{m-1} + l^{m-1} - l^m \\
& \geq & (a - b^n) \cdot (1 + (m - n) \cdot c) + l^{m-1} - l^m \\
& > & (a - b^n) \cdot (1 + (m - n) \cdot c) + b^m - l^m \\
& \geq & (a - b^n) \cdot (1 + (m - n) \cdot c) + c \cdot (a - b^m) \\
& \geq & (a - b^n) \cdot (1 + (m - n) \cdot c) + c \cdot (a - b^n) \\
& = & (a - b^n) \cdot (1 + (m - n + 1) \cdot c)
\end{array}$$

  where the first inequality ($\geq$) is due to the induction hypothesis, the second inequality ($>$) holds since $l^{m-1} > b^m$, the third inequality ($\geq$) holds because of the *decrease*-constraint and the fourth inequality ($\geq$) is satisfied since $b^m \leq b^n$.

This implies that $l^m \leq a - (a - b^n) \cdot (1 + (m - n + 1) \cdot c)$ for all $m$. Since $a - b^n > 0$, this is a contradiction, since the right-hand side of the inequality will eventually be negative. □

Naturally, one has to determine suitable constants $a_i, c$. These can either be derived in some way from the given functions or one can existentially quantify over the constants. Note that it is sound to let *decrease* hold for $l = a_0 = 0$, since $\exists$ automatically wins in this case.

We can now show the correctness of the modified game, provided that *decrease* is well-founded.

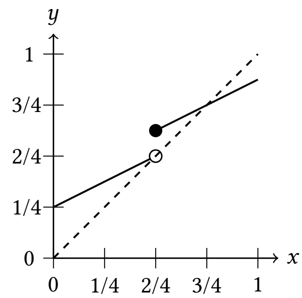
Proposition 7.5 (correctness of the modified game). *Let $E$ be a system of equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$ with solution $\boldsymbol{u} \in L^m$ and let decrease be a well-founded predicate. Then the modified game is correct: for all $(v_i, i) \in L \times \underline{m}$, if $\exists$ has a winning strategy in a play starting from $(v_i, i)$ then $v_i \sqsubseteq u_i$.*

Proof. We show that whenever $\exists$ has a winning strategy in the modified game for $(v_i, i)$, then $\exists$ has a winning strategy in the original game for all $(b_i, i)$ with $b_i \ll v_i$. This implies $b_i \sqsubseteq u_i$ and finally $v_i = \bigsqcup_{b_i \ll v_i} b_i \sqsubseteq u_i$.

Let $b_i \ll v_i$. We start the game with $(v_i, i)$ and assume that $\forall$ chooses $b_i$ in the first step. Then $\exists$ follows her winning strategy in the modified game until she reaches a leaf, i.e., an index $k$ which already appeared earlier in the game and no higher index has occurred in between. At this point $\forall$ will choose $b \ll l_k$ in the original game. Note that $l_k \sqsubseteq v_k$ since $\exists$ wins the game: in the case of a $\nu$-index this follows directly and in the case of a $\mu$-index we have $l_k \sqsubseteq b_k \ll v_k$.

We will now restart the game after the prefix of the play which ends at the first occurrence of the index $k$. We keep the value $v_k$ but set $b_k := b$. This is a valid choice since $b \ll l_k \sqsubseteq v_k$ and hence $b \ll v_k$.

Note that $\exists$ always has an available move, since she can move in the modified game. It is left to show that she can win infinite games: assume that we have an infinite run where the highest index




(a)

```
; Predicate encoding the game
(define-fun win-game ((v Real)) Bool
(forall ((b Real))                              ; forall chooses b
    (=> (and (< 0.0 b) (< b v))                 ; with 0 < b < v
        (exists ((l Real))                      ; exists chooses l
            (and (<= 0.0 l) (<= l 1.0)          ; with 0 <= l <= 1
                (>= (f l) b)                    ; with f(l) >= b
                (<= l b) (decrease v b l))))))  ; and we decrease

; Specify that we can win the game for v
(assert (win-game v))

; v is the greatest value for which one can win the game
(assert (forall ((w Real))
    (=> (and (<= 0.0 w) (<= w 1.0) (win-game w))
        (<= w v))))
(check-sat)
(get-model)
```

(b) SMT formula encoding the modified game

Fig. 6

that occurs infinitely often is $k$ with $\eta_k = \mu$. Consider the run from the point onwards where we do not visit any indices $\ell > k$ any more. Then we will eventually find a $k$-index (either seeing it for the first time or via a restart). The next occurrence of $k$ will be in a restart situation (since the condition that there is no higher index in between is automatically satisfied). In this case we will verify *decrease*$(v_k, b_k, l_k)$ for the current values $b_k, l_k$, which will be denoted $b^0, l^0$ ($v_k$ is always left unchanged). This continues and we obtain lattice elements $b^m, l^m$, $m \in \mathbb{N}_0$ where $b^{m+1} \ll l^m$ (since the choice of $\forall$ is restricted accordingly) and $l^m \sqsubseteq b^m$ (since we check this condition at the restart for each $\mu$-index). All are way-below $v_k$. Furthermore *decrease*$(v_k, b^m, l^m)$ holds for all these values. But this is a contradiction to the fact that the predicate *decrease* is well-founded. $\square$

*Example 7.6.* As an example, before discussing completeness, consider the monotone, but discontinuous function $f\colon [0,1] \to [0,1]$ defined by:

$$f(x) = \begin{cases} \frac{1}{4} + \frac{1}{2}x & \text{if } 0 \le x < \frac{1}{2} \\ \frac{3}{8} + \frac{1}{2}x & \text{if } \frac{1}{2} \le x \le 1 \end{cases}$$

The graph of the function looks as shown in Figure 6a. The dashed diagonal intersects the graph at the position of the only fixpoint.

We are interested in computing the least fixpoint, i.e., we consider the equation $x =_\mu f(x)$. We set $c = 1$, $a_1 = 0$, $a_2 = \frac{1}{2}$ (the discontinuity point of $f$) and consider the corresponding *decrease*-constraint. The basis contains all elements of $[0,1]$, apart from 0. Then we can easily encode the modified game in the SMT-LIB format, see Figure 6b, which shows the relevant part (the rest is the definition of the functions `f`, `decrease` and the declaration of the constant `v`). Note that SMT-LIB uses a prefix notation. We define a predicate `win-game` which encodes the fact that $\exists$ win the modified game for a value `v` by simply spelling out the definition. Then we require that the game can be won for `v` and that `v` is the largest such value.

By running the SMT solver `cvc4`, we obtain $\frac{3}{4}$. Since we only showed correctness we can only guarantee that the value found is smaller or equal than the true solution. However, in this case $\frac{3}{4}$ is the true solution, and this is not by chance. In fact, we will discuss sufficient conditions that ensure completeness that cover also this specific example.

We have also run successful experiments with the SMT solver `z3` involving equation systems and non-linear (quadratic) equations, where it is less obvious to compute fixpoints.

It is also possible to encode the solution of a fixpoint equation systems into SMT solvers in a more direct way (see [Mio and Simpson 2017] for a more detailed explanation). In the above example one would simply search for a fixpoint (which can be determined by solving linear equations) such that all other fixpoints are larger or equal. While the direct encoding is reasonably straightforward, it has been shown in [Mio and Simpson 2017] that due to the nesting of equations the encoding will be of a size exponential in the number of equations. This can be also the case in our setting (due to the growth of the trees depicted above), however if every function $f_i$ depends only on few parameters (preferably the $i$-th parameter and one other), then the game trees can be of linear size and we obtain also formulae of linear size. To our knowledge, such an efficiency gain cannot be achieved in the direct encoding.

We will now discuss the issue of completeness. We will first prove that it holds when the lattice order is well-ordered, hence well-founded and total. This is for instance the case for the lattice of integers (enriched with a top element).

Proposition 7.7 (completeness on well-orders). *Let $(L, \sqsubseteq)$ be a lattice where $\sqsubset$ is a well-order and define the decrease-predicate decrease$(v, b, l) = (l \sqsubset b)$. Furthermore assume that the solution is reached in at most $\omega$ steps, i.e., all entries in ord$(\boldsymbol{u})$ for the solution $\boldsymbol{u}$ are at most $\omega$. Then the modified game is complete in the following sense: $\exists$ has a winning strategy for $(u_i, i)$ for every $i \in \underline{m}$.*

Proof. Since the modified game starts with $v_i = u_i$ we can always assume that $v_i$ is the component of a $\mu$-approximant $\boldsymbol{v}$. This means that we can follow the winning strategy of $\exists$ in the original game, described in Lemmas 4.3 and 4.4 by descending along the $\mu$-approximants.

We only make a slight modification in the choice of the new $\mu$-approximant whenever the current game index $j$ satisfies $\eta_j = \mu$. Assume that $v_j$ is fixed and that $v_j$ is a component of a $\mu$-approximant $\boldsymbol{v}$. Furthermore $b_j \ll v_j$. Take the least ordinal such that $b_j \sqsubseteq f^{\gamma}_{j,\boldsymbol{v}}(\bot)$. Since $v_j = f^{\alpha}_{j,\boldsymbol{v}}(\bot)$ for some $\alpha \leq \omega$ it holds that $\gamma < \omega$. Hence either $\gamma = 0$ and $b_j = \bot$ (which cannot occur since $\bot \notin B_L$) or $\gamma = \delta + 1$ is a successor ordinal. We define $l_j = f^{\delta}_{j,\boldsymbol{v}}(\bot)$ and we can show with the same arguments as in Lemma 4.3 that one can define a new $\mu$-approximant $\boldsymbol{l} = (l_1, \ldots, l_j, v_{j+1}, \ldots, v_m)$ where $(l_1, \ldots, l_{j-1}) = sol(E[\boldsymbol{x}_{j+1,m} := \boldsymbol{v}_{j+1,m}][x_j := l_j])$ which satisfies $\boldsymbol{l} \in \mathrm{E}(b_j, j)$ and $ord(\boldsymbol{v}) >_j ord(\boldsymbol{l})$. Furthermore $l_j \sqsubset b_j$, due to the fact that $\gamma$ is minimal and the order is total.

Now assume that we have reached a leaf in the game tree with index $k$, i.e., there is an earlier occurrence of $k$ with no larger index in between, and let $v_k, b_k, l_k \in L$ the lattice elements which are recorded in the game. Furthermore $v_k$ is a component of a $\mu$-approximant $\boldsymbol{v}$ and $l_k$ is a component of another $\mu$-approximant $\boldsymbol{l}$. Since all indices $j$ that occur between the two occurrences of $k$ satisfy $j < k$ and the subsequent $\mu$-approximants are ordered by $\succeq_j$, we have $\boldsymbol{v} \succeq_k \boldsymbol{l}$ (if $\eta_k = \mu$ the equality is strict).

In particular the construction of new $\mu$-approximants is such that all components with $i > k$ are unchanged: $v_i = l_i$. If $\eta_k = \nu$ also $v_k = l_k$ and hence $l_k \sqsubseteq v_k$. If $\eta_k = \mu$ we only modify the $k$-the component of the $\mu$-approximant in the first step and it is left unchanged afterwards. The adapted construction of the $\mu$-approximant explained above ensures $l_k \sqsubset b_k$. Hence $\exists$ can make sure that the *decrease*-predicate is satisfied and wins the modified game. $\square$

Note that it does not hold that $\exists$ has a winning strategy for $(v_i, i)$ for every $v_i \sqsubseteq u_i$. The requirement that $v_i$ is a component of a $\mu$-approximant is important. Consider for instance the three-element lattice $L = \{\bot, a, \top\}$ and a monotone function $f : L \to L$ with $f(\bot) = f(a) = \bot$, $f(\top) = \top$. This function is monotone and has $\mu f = \bot$, $\nu f = \top$. We consider the equation $x =_\nu f(x)$ and start playing with $v_1 = a$, which is below the greatest fixpoint. If $\forall$ chooses $b_1 = a \ll a$, there does not exist a value $l_1$ with $f(l_1) \sqsupseteq b_1$ and $l_1 \sqsubseteq b_1$. This is connected to the fact that every post-fixpoint is below the greatest fixpoint, but not every lattice element below the greatest fixpoint is a post-fixpoint.

It would of course be desirable to extend Proposition 7.7 in such a way that it also covers fixpoint iteration beyond $\omega$ and non-well-founded lattice orders. Going beyond $\omega$ for well-founded orders requires the introduction of "special" lattice elements $a_i$ as in the *decrease*-predicate in order to cover the discontinuity points. However, this could be more complex for arbitrary lattices than for the reals, since there we know that $b \sqsubseteq v$ ($b \leq v$) and $b \not\ll v$ ($b \not< v$) imply $b = v$.

On the other hand, handling non-well-founded lattices orders would require an adaptation where, given $b_j \ll v_j$ and $v_j$ is below the least fixpoint, we are always able to choose $\boldsymbol{l}$ such that $l_j \sqsubseteq b_j$ and *decrease*$(b_j, v_j, l_j)$. We will characterise a class of functions on the real interval $[0, 1]$ for which this is the case and which also explains the shape of the *decrease*-condition of Lemma 7.4.

Lemma 7.8 (completeness for piecewise linear dominated functions). *Fix real numbers $a_0 = 0 < a_1 < \cdots < a_\ell < a_{\ell+1} \leq 1$, $0 < p_i < 1$, $0 \leq q_i \leq 1$, $i \in \underline{\ell+1}$ be given. Let $g_i(x) = p_i x + q_i$ and assume that $g_i(a_i) = a_i$ for all $i \in \underline{\ell+1}$. Let $g\colon [0,1] \to [0,1]$ be a function with*

$$g(x) = \begin{cases} g_i(x) & \text{if } a_{i-1} \leq x < a_i \\ g_{\ell+1}(x) & \text{if } a_\ell \leq x \end{cases} .$$

*Define the decrease predicate as in Lemma 7.4 with $c = \min_i \frac{1-p_i}{p_i}$ and with the values $a_0, \ldots, a_\ell$.*

*Then the modified game for the single equation $x =_\mu g(x)$ is complete, i.e., $\exists$ has a winning strategy for all $v \leq \mu f$. The same holds for any monotone function $f\colon [0,1] \to [0,1]$ with $f \geq g$ and $\mu f = a_{\ell+1}$.*

Proof. In this case the game is over after one iteration and we only have to show that $\exists$ can always find an answering move that ensures her win. In particular, we have to prove that for all $v, b \in [0,1]$ with $b < v \leq \mu f$ there exists $l \in [0,1]$ such that $0 \leq l \leq b$, $b \leq f(l)$ and *decrease*$(v, b, l)$.

First note that $g_{\ell+1}(\mu f) = g_{\ell+1}(a_{\ell+1}) = a_{\ell+1} = \mu f = f(\mu f)$ means that $g, f$ agree on the least fixpoint $\mu f$.

Now assume that $a_{i-1} \leq b < a_i$, where $i \in \underline{\ell+1}$. We have $f(b) \geq g_i(b)$. Define $l = \frac{b-q_i}{p_i}$ if $\frac{b-q_i}{p_i} \geq a_{i-1}$ otherwise $l = a_{i-1}$. We consider both cases:

- $l = \frac{b-q_i}{p_i}$:
  - It obviously holds that $f(l) \geq g_i(l) = p_i \cdot \frac{b-q_i}{p_i} + q_i = b$, since $a_{i-1} \leq l \leq a_i$ ($l \leq b < a_i$ is shown below).
  - Now we show that $b - l = \frac{1-p_i}{p_i} \cdot (a_i - b) \geq c \cdot (a_i - b)$.
    Since $g(a_i) = a_i$, we have $p_i a_i + q_i = a_i$, which implies $a_i = \frac{a_i - q_i}{p_i}$. Hence we get
    $$\begin{aligned} b &= a_i + (b - a_i) = \frac{a_i - q_i}{p_i} + (b - a_i) = \frac{a_i - q_i}{p_i} + \frac{b - a_i}{p_i} - \frac{b - a_i}{p_i} + (b - a_i) \\ &= \frac{b - q_i}{p_i} + \frac{-(b - a_i) + p_i(b - a_i)}{p_i} = l + \frac{1 - p_i}{p_i} \cdot (a_i - b) \end{aligned}$$
    This immediately implies $b - l = \frac{1-p_i}{p_i} \cdot (a_i - b) \geq c \cdot (a_i - b)$.
    If $a_\ell < b < a_{\ell+1} = \mu f$ we can infer $b - l \geq c \cdot (a_{\ell+1} - b) \geq c \cdot (v - b)$.
  - Note that $b - l \geq c \cdot (a_i - b)$ implies $l \leq b$ (since $b \leq a_i$) and hence $0 \leq l \leq b$.
- $l = a_{i-1}$:
  - It holds that $\frac{b-q_i}{p_i} < 0$ and hence $b < q_i$. This means that $f(l) = f(a_{i-1}) \geq g_i(a_{i-1}) = p_i a_{i-1} + q_i \geq q_i > b$, hence $f(l) \geq b$.
  - The *decrease*-predicate is automatically satisfied since $l = a_{i-1}$.

□

Lemma 7.8 states the following condition: the function $f$ must be larger or equal than a piecewise linear function where the pieces always end on the diagonal and the least fixpoints of $f, g$ are equal. (See Figure 7 where $g$ is drawn with a solid and $f$ with a dotted line.) The slope of these piecewise linear functions can be arbitrarily close to 1, that is close to the diagonal. Since monotone functions on $[0, 1]$ are always above the diagonal (in a post-fixpoint) before they reach the diagonal (the fixpoint), this is not a very strong restriction.

For instance, the function $f$ in Example 7.6 satisfies the requirements of Lemma 7.8 since it is itself a piecewise linear function with slopes $p_1 = p_2 = \frac{1}{2}$. This is why in the example the constant $c = \frac{1-p_1}{p_1} = 1$ is sufficient.

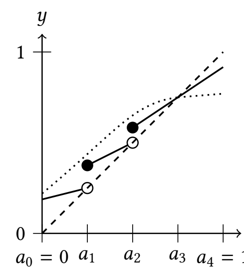



Fig. 7

Considering the case of multiple equations, note that if we could guarantee that every function $f_{i,I}$, which we use to determine $\mu$-approximants (see Definition 3.9) is of this kind, then we could generalise the proof. In [Mio and Simpson 2017] it is shown that the functions that arise from evaluating Lukasiewicz $\mu$-terms are piecewise linear, hence they would in principle fit our characterisation. On the other hand [Mio and Simpson 2017] also gives examples (based on the strong Lukasiewicz operators) that solutions of fixpoint expressions of continuous function need not be continuous themselves. Hence, the functions $f_{i,I}$ can be discontinuous even if all $f_i$ are continuous. It is an open question to find the discontinuity points $a_i$ that are required to define the *decrease*-predicates and it is unclear whether there are always only finitely many of them. We believe that this task might be easier for non-expansive functions, but we leave this as future work.

## 8 CONCLUSION

*Related work.* Our work is based on lattice theory and in particular on continuous lattices. The use of lattices in program analysis and verification has been pioneered by the work [Cousot and Cousot 1977]. Continuous lattices, which received this name due to their intimate connection with continuous functions, have originally been studied by Scott as a semantic domain for the $\lambda$-calculus [Scott 1972] and have since found many further applications in the semantics of programming languages [Abramsky and Jung 1994; Gierz et al. 2003].

The modal $\mu$-calculus is an expressive temporal logics, which originated in an unpublished manuscript by Scott and de Bakker and was further developed by Kozen [Kozen 1983]. For a good overview see [Bradfield and Walukiewicz 2018].

Its introduction posed the problem of efficient model-checking, which involves the solution of nested fixpoint equations, see, e.g., [Browne et al. 1997; Cleaveland et al. 1992; Seidl 1996]. The paper [Cleaveland et al. 1992] introduced the notion of a hierarchical system of fixpoint equations, on which our paper is based as well. One way to tackle the model-checking problem is to translate it into the question of finding winning strategies for parity games. The latter were first described in [Emerson and Jutla 1991].

A very satisfying technique for solving parity games was proposed in [Jurdziński 2000] resulting in an algorithm which is exponential only in half of the alternation depth. The approach crucially relies on the notion of progress measure, that can be seen as generalising both invariants and ranking functions. The complexity of computing progress measures has recently been improved to quasi-polynomial [Calude et al. 2017].

An extension to general lattices has been given in [Hasuo et al. 2016], which was very inspiring for our development. Compared to [Hasuo et al. 2016] we brought games back into the picture by introducing a game that generalises both parity games and the unfolding games in [Venema 2008]. This allowed us to define a notion of progress measures which is closer to the original definition of Jurdziński and, as such, admits a constructive characterisation as a least fixed point. This works in the general context of continuous lattices, providing a way of solving systems of fixpoint equations in settings that are beyond powerset lattices and were not covered by previous work. We devised the notion of selection and a logics for specifying the moves of the existential player, with the aim of making the computation of progress measures more efficient. We view as a valuable contribution the identification of continuous lattices as the right setting where these general results can be stated.

Usually, $\mu$-calculus formulae are evaluated over the state space of a transition system, i.e., over a powerset lattice. This changes if the $\mu$-calculus is not a classical logic, but lattice-valued as in [Kupfermann and Lustig 2007] or real-valued as in [Huth and Kwiatkowska 1997], which presents an algorithm based on the simplex method for the non-nested case. Solving equation systems over the reals was considered in [Gawlitza and Seidl 2011] and in [Mio and Simpson 2015, 2017]. In particular, [Mio and Simpson 2017] presents an algorithm for solving nested fixpoint equation systems over the interval $[0, 1]$ by a direct algorithm which represents and manipulates piecewise linear functions as conditioned linear expressions. As far as we know this algorithm has not been implemented. Our results can offer an alternative way to solve such equation systems.

Games for quantitative or probabilistic $\mu$-calculi have been studied in [McIver and Morgan 2007; Mio 2012]. As opposed to our game, such games closely follow the structure of the $\mu$-calculus formula on which the game is based (e.g., $\exists$ makes a choice at an $\vee$-node, $\forall$ at an $\wedge$-node). It is an interesting question whether the conceptual simplicity of our game can lead to a new perspective on existing games.

*Future work.* A parity game over a finite graph can be easily converted into a system of boolean equations whose solution characterises the winning positions for the players. Since our game is a standard parity game, possibly played on an infinite graph, the standard conversion would lead to infinitely many equations. Systems of equations of this kind are considered, e.g., in [Mader 1997]. An interesting question, still to be investigated, is under which conditions an infinite parity game can be converted into finitely many equations on an (infinite) powerset lattice.

The generality of continuous lattices suggests the possibility of instantiating our framework in various other application scenarios.

The use of our results for solving fixpoint equations over the reals via SMT solvers in § 7 appears to be promising, but it requires further investigation. In particular, we plan to deepen the issue of completeness for which we currently only have partial results (see § 7).

We also plan to study fixpoint equations on the (non-distributive, but continuous) lattices of equivalence relations and pseudo-metrics. As explained in the introduction, the computation of fixpoints for equivalence relations is essential for behavioural equivalences, and the same holds for pseudo-metrics and behavioural distances [van Breugel and Worrell 2005].

We would also like to determine whether we can handle quantitative logics whose modalities interact with (lattice) truth values in a non-trivial way, such as logics with discounted modalities as studied in [Almagor et al. 2014]. Expressing such logics as systems of fixed point equations over suitable continuous lattices and thus obtaining a game theoretical characterisation of the model checking problem seems reasonably easy. However, turning such characterisation into an effective technique requires some non-trivial symbolic approach due to the fact that the lattice is infinite.

Furthermore we would like to study situations in which local (or on-the-fly) algorithms rather than global fixpoint iteration can be used to check whether a lattice element is below the solution. Examples of such local algorithms are backtracking methods studied in [Hirschkoff 1998; Stevens and Stirling 1998]. In particular we are interested in the integration of local methods with up-to techniques for general lattices, see for instance [Bonchi et al. 2018; Pous 2007; Pous and Sangiorgi 2011].


## ACKNOWLEDGMENTS

Research is partially supported by DFG project BEMEGA and University of Padova project ASTA.

We want to thank the reviewers for their insightful and valuable comments. Furthermore we are grateful to Clemens Kupke for bringing fixpoint games to our attention.


## REFERENCES


Samson Abramsky and Achim Jung. 1994. Domain Theory. In *Handbook of Logic in Computer Science*, Samson Abramsky, Dov Gabbay, and Thomas Stephen Edward Maibaum (Eds.). Oxford University Press, 1–168.

Shaull Almagor, Udi Boker, and Orna Kupferman. 2014. Discounting in LTL. In *Proc. of TACAS '14 (Lecture Notes in Computer Science)*, Vol. 8413. Springer, 424–439.

Harsh Beohar, Barbara König, Sebastian Küpper, and Alexandra Silva. 2017. Conditional Transition Systems with Upgrades. In *Proc. of TASE '17*. IEEE Xplore, 1–8.

Filippo Bonchi, Pierre Ganty, Roberto Giacobazzi, and Dusko Pavlovic. 2018. Sound up-to techniques and Complete abstract domains. In *Proc. of LICS '18*. ACM, 175–184.

Julian Bradfield and Igor Walukiewicz. 2018. The mu-Calculus and Model Checking. In *Handbook of Model Checking*, Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem (Eds.). Springer, 871–919.

Anca Browne, Edmund M. Clarke, Somesh Jha, David E. Long, and Wilfredo R. Marrero. 1997. An improved algorithm for the evaluation of fixpoint expressions. *Theoretical Computer Science* 178, 1–2 (1997), 237–255.

Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. 2017. Deciding parity games in quasipolynomial time. In *Proc. of STOC '17*. ACM, 252–263.

Rance Cleaveland. 1990. Tableau-based model checking in the propositional mu-calculus. *Acta Informatica* 27, 8 (1990), 725–747.

Rance Cleaveland, Marion Klein, and Bernhard Steffen. 1992. Faster model checking for the modal Mu-Calculus. In *Proc. of CAV 1992 (Lecture Notes in Computer Science)*, Vol. 663. Springer, 410–422.

Maxime Cordy, Andreas Classen, Gilles Perrouin, Pierre-Yves Schobbens, Patrick Heymans, and Axel Legay. 2012. Simulation-based abstractions for software product-line model checking. In *Proc. of ICSE '12*. IEEE, 672–682.

Patrick Cousot and Radhia Cousot. 1977. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In *Proc. of POPL '77*. ACM, 238–252.

Radhia Cousot and Patrick Cousot. 1979. Constructive versions of Tarski's fixed point theorems. *Pacific J. Math.* 82, 1 (1979), 43–57.

Brian A. Davey and Hilary A. Priestley. 2002. *Introduction to lattices and order.* Cambridge University Press.

Giorgio Delzanno and Jean-François Raskin. 2000. Symbolic Representation of Upward-Closed Sets. In *Proc. of TACAS '00 (Lecture Notes in Computer Science)*, Susanne Graf and Michael I. Schwartzbach (Eds.), Vol. 1785. Springer, 426–440.

Pantelis E. Eleftheriou, Costas D. Koutras, and Christos Nomikos. 2012. Notions of bisimulation for Heyting-valued modal languages. *Journal of Logic and Computation* 22 (2012), 213–235.

E. Allen Emerson. 1985. Automata, tableaux, and temporal logics. In *Proceedings of Logics of Programs 1985 (Lecture Notes in Computer Science)*, R. Parikh (Ed.), Vol. 193. Springer, 79–88.

E. Allen Emerson and Charanjit S. Jutla. 1991. Tree automata, Mu-Calculus and determinacy. In *Proc. of SFCS '91*. IEEE, 368–377.

Melvin Fitting. 1991. Many-valued modal logics. *Fundamenta Informaticae* 15 (1991), 235–254.

Gaëlle Fontaine. 2008. Continuous Fragment of the $\mu$-Calculus. In *Proc. of CSL '08 (Lecture Notes in Computer Science)*, Vol. 5213. Springer, 139–153.

Thomas Martin Gawlitza and Helmut Seidl. 2011. Solving systems of rational equations through strategy iteration. *ACM Trans. Program. Lang. Syst.* 33, 3 (2011), 11:1–11:48.

Gerhard Gierz, Karl H. Hofmann, Klaus Keimel, Jimmie D. Lawson, Michael W. Mislove, and Dana S. Scott. 2003. *Continuous Lattices and Domains.* Cambridge University Press.

Orna Grumberg, Martin Lange, Martin Leucker, and Sharon Shoham. 2005. Don't Know in the $\mu$-Calculus. In *Proc. of VMCAI '05 (Lecture Notes in Computer Science)*, Radhia Cousot (Ed.), Vol. 3385. Springer, 233–249.

Helle Hvid Hansen, Clemens Kupke, Johannes Marti, and Yde Venema. 2017. Parity Games and Automata for Game Logic. In *Proc. of DALI '17 (Lecture Notes in Computer Science)*, Vol. 10669. Springer, 115–132.

Ichiro Hasuo, Shunsuke Shimizu, and Corina Cîrstea. 2016. Lattice-theoretic progress measures and coalgebraic model checking. In *Proc. of POPL '16*. ACM, 718–732.

Daniel Hirschkoff. 1998. Automatically Proving Up To Bisimulation. In *Proc. of MFCS '98 Workshop on Concurrency (Electronic Notes in Theoretical Computer Science)*. Elsevier, 75–89.

Michael Huth and Marta Kwiatkowska. 1997. Quantitative analysis and model checking. In *Proc. of LICS '97*. IEEE, 111–122.

Marcin Jurdziński. 2000. Small Progress Measures for Solving Parity Games. In *Proc. of STACS '00 (Lecture Notes in Computer Science)*, Vol. 1770. Springer, 290–301.

Dexter Kozen. 1983. Results on the Propositional $\mu$-Calculus. *Theoretical Computer Science* 27, 3 (1983), 333–354.

Orna Kupfermann and Yoad Lustig. 2007. Latticed Simulation Relations and Games. In *Proc. of ATVA '07 (Lecture Notes in Computer Science)*, Vol. 4672. Springer, 316–330.

Angelika Mader. 1997. *Verification of Modal Properties Using Boolean Equation Systems*. Ph.D. Dissertation. TU München.

George Markowsky. 1976. Chain-complete poscts and directed sets with spplications. *Algebra Universalis* 6 (1976), 53–68.

Annabelle McIver and Carroll Morgan. 2007. Results on the quantitative $\mu$-calculus qM$\mu$. *ACM Trans. Comp. Log.* 8, 1:3 (2007), 43.

Matteo Mio. 2012. On the Equivalence of Game and Denotational Semantics for the Probabilistic $\mu$-Calculus. *Logical Methods in Computer Science* 8, 2:07 (2012), 1–21.

Matteo Mio and Alex Simpson. 2015. Łukasiewicz $\mu$-calculus. https://arxiv.org/abs/1510.00797 arXiv:1510.00797.

Matteo Mio and Alex Simpson. 2017. Łukasiewicz $\mu$-calculus. *Fundamenta Informaticae* 150, 3-4 (2017), 317–346.

Flemming Nielson, Hanne Riis Nielson, and Chris Hankin. 1999. *Principles of Program Analysis*. Springer.

Katja Poltermann. 2017. *A Modal Logic for Conditional Transition Systems*. Bachelor's thesis. Universität Duisburg-Essen.

Damien Pous. 2007. Complete Lattices and Up-To Techniques. In *Proc. of APLAS '07*. Springer, 351–366. LNCS 4807.

Damien Pous and Davide Sangiorgi. 2011. Enhancements of the bisimulation proof method. In *Advanced Topics in Bisimulation and Coinduction*, Davide Sangiorgi and Jan Rutten (Eds.). Cambridge University Press.

George. N. Raney. 1952. Completely Distributive Complete Lattices. *Proceedings of the AMS* 3, 5 (1952), 677–680.

Davide Sangiorgi. 2011. *Introduction to Bisimulation and Coinduction*. Cambridge University Press.

Klaus Schneider. 2004. *Verification of Reactive Systems: Formal Methods and Algorithms*. Springer.

Dana Scott. 1972. Continuous lattices. In *Toposes, Algebraic Geometry and Logic (Lecture Notes in Mathematics)*, F. W. Lawvere (Ed.). Springer, 97–136.

Helmut Seidl. 1996. Fast and simple nested fixpoints. *Inform. Process. Lett.* 59, 6 (1996), 303–308.

Perdita Stevens and Colin Stirling. 1998. Practical Model-Checking Using Games. In *Proc. of TACAS '98 (Lecture Notes in Computer Science)*, Vol. 1384. Springer, 85–101.

Colin Stirling. 1995. Local Model Checking Games. In *Proc. of CONCUR '95 (Lecture Notes in Computer Science)*, Vol. 962. Springer, 1–11.

Colin Stirling and David Walker. 1991. Local Model Checking in the Modal mu-Calculus. *Theoretical Computer Science* 89, 1 (1991), 161–177.

Alfred Tarski. 1955. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.* 5 (1955), 285–309.

Franck van Breugel and James Worrell. 2005. A behavioural pseudometric for probabilistic transition systems. *Theoretical Computer Science* 331 (2005), 115–142.

Yde Venema. 2008. Lectures on the modal $\mu$-calculus. Lecture notes, Institute for Logic, Language and Computation, University of Amsterdam.

Wieslaw Zielonka. 1998. Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees. *Theoretical Computer Science* 200, 1-2 (1998), 135–183.

# A COMPARING FIXPOINT EQUATION SYSTEMS WITH $\mu$-CALCULUS FORMULAE

We will show how $\mu$-calculus formulae can be translated into equation systems and vice versa.

Hereafter we will assume that in every formula different bound variables have different names, a requirement that can always be fulfilled by alpha-renaming. In this way for every variable $x$ appearing in a closed formula $\varphi$, we can refer to "the" fixpoint subformula quantifying $x$, that will be denoted $\varphi_x$ (hence $\varphi_x$ is of the kind $\eta x.\psi$).

*Definition A.1 (equation system for a formula).* Given a closed fixpoint formula $\varphi$ of the $\mu$-calculus, let $(x_1, \ldots, x_m)$ be the tuple of variables in $\varphi$, in the order in which their quantification appears from right to left. The *equational form* of $\varphi$ is $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{\theta}$, where, for all $i \in \underline{m}$, if $\varphi_{x_i} = \eta_i x_i.\psi_i$ then $\theta_i$ is the (open) formula obtained from $\psi_i$ by replacing every fixpoint subformula with the corresponding propositional variable.

Observe that the restriction to fixpoint formulae is not limiting since any formula $\varphi$ is equivalent to a fixpoint formula $\mu x.\varphi$, where $x$ is a variable not occurring in $\varphi$.

Once a transition system $(\mathbb{S}, \to)$ is fixed, the formula in equational form can be interpreted as a system of equations over the powerset lattice $(\mathbf{2}^{\mathbb{S}}, \subseteq)$, by replacing formulae with their semantics, i.e., an equation $x_i =_{\eta_i} \theta_i$ becomes

$$x_i \quad =_{\eta_i} \quad \|\theta_i\|_\rho$$

where in the right-hand side $\|x_j\|_\rho$ is replaced by $x_j$ and $\rho$ is some fixed environments providing a meaning only for propositions.

It is not difficult to see that also a converse transformation is possible, i.e., a system of fixpoint equations of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{\psi}$ where each $\psi_i$ is an open formula with propositional variables in $\boldsymbol{x}$ and without fixpoints, can be translated into a tuple of $\mu$-calculus formulae, equivalent to the system in a sense formalised later.

*Definition A.2 (formulae for an equation system).* Let $E$ be a system of $m$ equations $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{\psi}$ where each $\psi_i$ is an open formula with variables in $\boldsymbol{x}$ and without fixpoints. The corresponding $m$-tuple of $\mu$-calculus formulae, denoted by $\boldsymbol{\varphi}^E$, is defined inductively as follows, where $E_i$ denotes the system consisting of the first $i$ equations of $E$, for all $i \in \underline{m}$.

$$\begin{array}{lcll} \boldsymbol{\varphi}^{\emptyset} & = & () & \\ \boldsymbol{\varphi}^{E_i} & = & (\boldsymbol{\varphi}^{E_{i-1}}[\varphi_i^{E_i} x_i], \varphi_i^{E_i}) & \qquad \text{where } \varphi_i^{E_i} = \eta_i x_i.\psi_i[\varphi_j^{E_{i-1}} x_j]_{\forall j \in \underline{i-1}} \end{array}$$

Then $\boldsymbol{\varphi}^E = \boldsymbol{\varphi}^{E_m}$.

Note that $\boldsymbol{\varphi}^E$ is a tuple of closed formulae.

A similar procedure is given in [Cleaveland et al. 1992] for the characterisation of $\mu$-calculus formulae in terms of equation systems, to allow an efficient model-checking algorithm. However such equation systems differ from ours. In particular, the solution of a system is defined just by means of the semantics of the formulae into which the system can be translated.

We finally prove that the proposed translations preserve the semantics. We will need the substitution lemma for the $\mu$-calculus as stated below.

Lemma A.3 (substitution in $\mu$-calculus). *For all $\mu$-calculus formulae $\varphi$ and $\psi$, variable $x$, and environment $\rho$, it holds $\|\varphi[\psi x]\|_\rho = \|\varphi\|_{\rho[x \mapsto \|\psi\|_\rho]}$.*

Proof. It can be easily proved by routine induction on $\varphi$. □

Proposition A.4 (correspondence between formulae and equation systems). *Let $\varphi$ be a closed fixpoint formula of the $\mu$-calculus and let $E$ be the system arising as its equational form. For any environment $\rho$, it holds $\|\varphi\|_\rho = sol_m(E)$, where $m$ is number of equations in $E$. Conversely, given a system $E$ of $m$ equations $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{\psi}$, for all $i \in \underline{m}$, it holds that $sol_i(E) = \|\varphi_i^E\|_\rho$, where $\rho$ is any environment.*

Proof. We prove the two statements separately.

For the first part, let $\varphi$ be a closed fixpoint formula of the $\mu$-calculus and let $E$ the system arising as its equational form. Recall that for every $i \in \underline{m}$, the $i$-th equation of the system is $x_i =_{\eta_i} \theta_i$, where $\theta_i$ is obtained from the subformula $\psi_i$ of the fixpoint formula $\varphi_{x_i} = \eta_i x_i.\psi_i$, as described in Definition A.1. In particular, $\varphi = \varphi_{x_m} = \eta_m x_m.\psi_m$, corresponds to the last equation of the system $x_m =_{\eta_m} \theta_m$. Then, we prove that for all $i \in \underline{m}$, $\|\varphi_{x_i}\|_{\rho'} = sol_i(E')$ where $\rho' = \rho[x_{i+1} \mapsto S_{i+1}, \ldots, x_m \mapsto S_m]$ and

$E' = E[x_m := S_m]\ldots[x_{i+1} := S_{i+1}]$, and every $S_j \subseteq \mathbb{S}$. Clearly this implies the desired result. The proof proceeds by induction on the index $i$.

($i = 1$) By definition of substitution we know that the system $E'$ consists of a single equation, i.e., $x_1 =_{\eta_1} \theta'_1$ where $\theta'_1 = \theta_1[S_m x_m]\ldots[S_2 x_2]$. By definition of solution of a system, we have that $sol_1(E') = sol(x_1 =_{\eta_1} \theta'_1) = \eta_1(\lambda S.\|\theta_1\|_{\rho'[x_1 \mapsto S]})$. Similarly, by definition of the semantics of $\mu$-calculus we know that $\|\varphi_{x_1}\|_{\rho'} = \|\eta_1 x_1.\psi_1\|_{\rho'} = \eta_1(\lambda S.\|\psi_1\|_{\rho'[x_1 \mapsto S]})$. By the definition of the ordering of the variables given in Definition A.1, we must have that $\psi_1$ does not contain any fixpoint subformula, otherwise its index could not be 1. Hence $\theta_1 = \psi_1$, and so $\|\varphi_{x_1}\|_{\rho'} = sol_1(E')$.

($i > 1$) In this case, by definition of solution, we have $sol_i(E') = \eta_i(\lambda S.\|\theta_i\|_{\rho''})$ where $\rho'' = \rho'[x_i \mapsto S][\boldsymbol{x}_{1,i-1} \mapsto sol(E'[x_i := S])]$. While, by definition of the semantics we have $\|\varphi_{x_i}\|_{\rho'} = \|\eta_i x_i.\psi_i\|_{\rho'} = \eta_i(\lambda S.\|\psi_i\|_{\rho'[x_i \mapsto S]})$. By an inspection of the definition of $\boldsymbol{\theta}$ and the ordering of the variables, one can notice that for all $j \in \underline{m}$, $\psi_j = \theta_j[\varphi_{x_{j-1}} x_{j-1}]\ldots[\varphi_{x_1} x_1]$. Then, we have that $\|\psi_i\|_{\rho'[x_i \mapsto S]} = \|\theta_i[\varphi_{x_{i-1}} x_{i-1}]\ldots[\varphi_{x_1} x_1]\|_{\rho'[x_i \mapsto S]}$. Furthermore, by repeatedly applying Lemma A.3 we obtain that $\|\theta_i[\varphi_{x_{i-1}} x_{i-1}]\ldots[\varphi_{x_1} x_1]\|_{\rho'[x_i \mapsto S]} = \|\theta_i\|_{\rho_i}$ where $\rho_1 = \rho'[x_i \mapsto S]$ and for all $j \in \underline{i-1}$, $\rho_{j+1} = \rho_j[x_{i-j} \mapsto \|\varphi_{x_{i-j}}\|_{\rho_j}]$. Note that actually $\rho_i = \rho'[x_i \mapsto S][x_{i-1} \mapsto \|\varphi_{x_{i-1}}\|_{\rho_1}]\ldots[x_1 \mapsto \|\varphi_{x_1}\|_{\rho_{i-1}}]$. Now we just need to prove that $\rho_i = \rho''$. To show this we can use the inductive hypothesis $i-1$ times, recalling the recursive structure of the solutions of systems. Therefore, we can conclude that $\|\psi_i\|_{\rho'[x_i \mapsto S]} = \|\theta_i\|_{\rho_i} = \|\theta_i\|_{\rho''}$, and so $\|\varphi_{x_i}\|_{\rho'} = sol_i(E')$.

Let us now focus on the second part. Let $E$ be a system of $m$ equations of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{\psi}$. We have to prove that, $sol_i(E) = \|\varphi_i^E\|_\rho$ for all $i \in \underline{m}$. The proof proceeds by induction on the number of equations $m$.

($m = 1$) Clearly there is only one possible index $i \in \underline{1}$, that is $i = 1$. Then, by definition of solution we know that $sol_1(E) = sol(x_1 =_{\eta_1} \psi_1) = \eta_1(\lambda S.\|\psi_1\|_{\rho[x_1 \mapsto S]})$. Moreover, by definition of the semantics of $\mu$-calculus we have that also $\|\varphi_1^E\|_\rho = \|\eta_1 x_1.\psi_1\|_\rho = \eta_1(\lambda S.\|\psi_1\|_{\rho[x_1 \mapsto S]})$. Thus we can immediately conclude.

($m > 1$) First we show the property for $i = m$. By definition of solution we have $sol_m(E) = \eta_m(\lambda S.\|\psi_m\|_{\rho'})$ where $\rho' = \rho[x_m \mapsto S][\boldsymbol{x}_{1,m-1} \mapsto sol(E')]$ and $E' = E[x_m := S]$. By Definition A.2 we know that $\varphi_m^E = \eta_m x_m.\psi_m[\varphi_j^{E_{m-1}} x_j]_{\forall j \in \underline{m-1}}$. Then, the semantics is $\|\varphi_m^E\|_\rho = \eta_m(\lambda S.\|\psi_m[\varphi_j^{E_{m-1}} x_j]_{\forall j \in \underline{m-1}}\|_{\rho[x_m}$ By repeatedly applying Lemma A.3 we obtain that $\|\psi_m[\varphi_j^{E_{m-1}} x_j]_{\forall j \in \underline{m-1}}\|_{\rho[x_m \mapsto S]} = \|\psi_m\|_{\rho''}$ where $\rho'' = \rho[x_m \mapsto S][x_j \mapsto \|\varphi_j^{E_{m-1}}\|_{\rho[x_m \mapsto S]}]_{\forall j \in \underline{m-1}}$. Now we just need to show that $\rho'' = \rho'$, that is, for all $j \in \underline{m-1}$, $\|\varphi_j^{E_{m-1}}\|_{\rho[x_m \mapsto S]} = sol_j(E')$. Since $E'$ has $m-1$ equations, by inductive hypothesis we know that $sol_j(E') = \|\varphi_j^{E'}\|_\rho$. Recalling that $E' = E[x_m := S]$, we can immediately conclude that $\|\varphi_j^{E'}\|_\rho = \|\varphi_j^{E_{m-1}}\|_{\rho[x_m \mapsto S]}$.

Instead, for all $i \in \underline{m-1}$, by definition of solution and what we just proved above we have $sol_i(E) = sol_i(E[x_m := sol_m(E)]) = sol_i(E[x_m := \|\varphi_m^E\|_\rho])$. Moreover, let $E' = E[x_m := \|\varphi_m^E\|_\rho]$, since $E'$ has $m-1$ equations, by inductive hypothesis we know that $sol_i(E') = \|\varphi_i^{E'}\|_\rho$. Observe that $\|\varphi_i^{E'}\|_\rho = \|\varphi_i^{E_{m-1}}\|_{\rho[x_m \mapsto \|\varphi_m^E\|_\rho]} = \|\varphi_i^{E_{m-1}}[\varphi_m^E x_m]\|_\rho$ by Lemma A.3 and since $E' = E[x_m := \|\varphi_m^E\|_\rho]$. Then, since by Definition A.2 we know that $\varphi_i^E = \varphi_i^{E_{m-1}}[\varphi_m^E x_m]$, we can conclude that $\|\varphi_i^E\|_\rho = \|\varphi_i^{E'}\|_\rho = sol_i(E') = sol_i(E)$. □

## B RESULTS CONCERNING THE COMPARISON WITH [Hasuo et al. 2016]

### B.1 Comparing the Definitions of Solutions

Here we show that the definition of the solution of an equational system in [Hasuo et al. 2016] is equivalent to our Definition 3.4. In both definitions the solution $\boldsymbol{u} = (u_1, \ldots, u_m)$ is solved recursively based on interim solutions by calculating fixpoints.

*Definition B.1 (Solution of an equational system [Hasuo et al. 2016]).* Let $L$ be a lattice and let $E$ be a system of $m \geq 1$ equations on $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. For each $i \in \underline{m}$ and $j \in \underline{i}$ we define monotone functions $f^{\ddagger} \colon L^{m-i+1} \to L$ and $l_j^{(i)} \colon L^{m-i} \to L$ as follows, inductively on $i$:

(1) $i = 1$ :

$$f_1^{\ddagger}(l_1, \ldots, l_m) := f_1(l_1, \ldots, l_m)$$
$$l_1^{(1)}(l_2, \ldots, l_m) := \eta_1[f_1^{\ddagger}(\_, l_2, \ldots, l_m) \colon L \to L]$$

with $\eta_1 \in \{\mu, \nu\}$.

(2) $i = i + 1$ :

$$f_{i+1}^{\ddagger}(l_{i+1}, \ldots, l_m) := f_{i+1}(l_1^{(i)}(l_{i+1}, \ldots, l_m), \ldots, l_i^{(i)}(l_{i+1}, \ldots, l_m), l_{i+1}, l_m)$$
$$l_{i+1}^{(i+1)}(l_{i+2}, \ldots, l_m) := \eta_{i+1}[f_{i+1}^{\ddagger}(\_, l_{i+2}, \ldots, l_m \colon L \to L]$$

with $\eta_{i+1} \in \{\mu, \nu\}$. The $l_{i+1}^{(i+1)}$ solution is then used to obtain the $(i+1)$-th interim solutions for each $j \in \underline{i}$:

$$l_j^{(i+1)}(l_{i+2}, \ldots, l_m) := l_j^{(i)}(l_{i+1}^{(i+1)}(l_{i+2}, \ldots, l_m), l_{i+2}, \ldots, l_m)$$

Proposition B.2. *Let $L$ be a lattice and let $E$ be a system of $m \geq 1$ equations on $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. Then the solution from Definition B.1 coincides with the solution from Definition 3.4.*

Proof. Let $l_{i+1}, \ldots, l_m \in L$ be given. We show $l_j^{(i)}(l_{i+1}, \ldots, l_m) = sol_j(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}])$ for $j \in \underline{i}$ by induction on $i$:

(1) $i = 1$: We define $E' = E[\boldsymbol{x}_{2,m} := \boldsymbol{l}_{2,m}]$ and according to Definition 3.4 we have

$$sol(E[\boldsymbol{x}_{2,m} := \boldsymbol{l}_{2,m}]) = sol(E') = sol(E'[x_1 := u_1]), u_1) = (u_1)$$

where $u_1 = \eta_1(\lambda x.\, f_1(x))$. In Definition B.1 for $i = 1$ we only have to consider $l_1^{(1)}(l_2, \ldots, l_m) = \eta_1[f_1^{\ddagger}(\_, l_2, \cdots, l_m)] = \eta_1[f_1(\_, l_2, \ldots, l_m)]$ which corresponds to $sol_1(E[\boldsymbol{x}_{2,m} := \boldsymbol{l}_{2,m}]) = u_1 = \eta_1(\lambda x.\, f_1(x))$.

(2) $i \to i + 1$ : We define $E' = E[\boldsymbol{x}_{i+2,m} := \boldsymbol{l}_{i+2,m}]$. Here we need to distinguish two cases to prove that $l_j^{(i+1)}(l_{i+2}, \ldots, l_m) = sol_j(E')$ for all $j \in \underline{i}$.

(a) $j = i + 1$: From Definition 3.4 we have

$$sol(E[\boldsymbol{x}_{i+2,m} := \boldsymbol{l}_{i+2,m}]) = sol(E') = sol(E'[x_{i+1} := u_{i+1}]), u_{i+1})$$

where $u_{i+1} = \eta_{i+1}(\lambda x.\, f_{i+1}(sol(E'[x_{i+1} := x]), x, l_{i+1}, \ldots, l_m))$. Hence

$$sol_{i+1}(E[\boldsymbol{x}_{i+2,m} := \boldsymbol{l}_{i+2,m}]) = u_{i+1}.$$

From Definition B.1 we obtain $l_{i+1}^{(i+1)}(l_{i+2}, \ldots, l_m) = \eta_{i+1}[f_{i+1}^{\ddagger}(x, l_{i+2}, \ldots, l_m)]$ where

$$f_{i+1}^{\ddagger}(x, l_{i+2}, \ldots, l_m) = f_{i+1}(l_1^{(i)}(x, l_{i+2}, \ldots, l_m), \ldots, l_i^{(i)}(x, l_{i+2}, \cdots, l_m), x, l_{i+2} \ldots, l_m).$$

From the induction hypothesis it follows that

$$l_j^{(i)}(x, l_{i+2}, \ldots, l_m) = sol_j(E[\boldsymbol{x}_{i+1,m} := x, \boldsymbol{l}_{i+2,m}]) = sol_j(E'[x_{i+1} := x]) = l_j$$

for $j \in \underline{i}$. We define $(l_1, \ldots, l_i) = sol(E'[x_{i+1} := x])$ and observe that $l_{i+1}^{(i+1)}(l_{i+2}, \ldots, l_m)$ is the $\eta_i$-fixpoint of $\lambda x. f_{i+1}(l_1, \ldots l_i, x, l_{i+2}, \ldots, l_m)$. The same is true for $u_{i+1}$ and hence we conclude $l_{i+1}^{(i+1)}(l_{i+2}, \ldots, l_m) = u_{i+1} = sol_{i+1}(E')$.

(b) $j \leq i$ : First, from Definition 3.4 we obtain

$$sol_j(E') = sol_j(E[\boldsymbol{x}_{i+2,m} := \boldsymbol{l}_{i+2,m}]) = sol_j(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}])$$

where $l_{i+1} = sol_{i+1}(E')$. From the induction hypothesis we know that $sol_j(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}]) = l_j^{(i)}(l_{i+1}, l_{i+2}, \ldots, l_m)$.
On the other hand we have from Definition B.1 that

$$l_j^{(i+1)}(l_{i+2} \ldots l_m) = l_j^{(i)}(l_{i+1}^{(i+1)}(l_{i+2}, \ldots l_m), l_{i+2}, \ldots, l_m)$$

and from (2a) we finally get $l_{i+1} = l_{i+1}^{(i+1)}(l_{i+2}, \ldots l_m)$, which concludes the proof.

□

### B.2 Comparing $\mu$-Approximants and Lattice Progress Measures [Hasuo et al. 2016]

As hinted in the main body of the paper, $\mu$-approximants can be seen as special lattice progress measures in the sense of [Hasuo et al. 2016], that we will refer here as hsc-measures. More precisely, as discussed below, the function that, for any $\mu$-approximant $\boldsymbol{l}$, maps the subvector of $ord(\boldsymbol{l})$ obtained by keeping only the components corresponding to $\mu$-indices to $\boldsymbol{l}$ is a hsc-measure. This is indeed the hsc-measure used in [Hasuo et al. 2016, Theorem 2.13] (completeness part).

*Definition B.3 (hsc-measure [Hasuo et al. 2016]).* Let $L$ be a lattice and let $E$ be a system of equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. We assume that $i_1, \ldots, i_k$ are the indexes of $\mu$-equations and $j_1, \ldots, j_{m-k}$ are the indexes of $\nu$-equations, i.e., $\eta_{i_h} = \mu$ for $h \in \{1, \ldots, k\}$ and $\eta_{j_h} = \nu$ for $h \in \{1, \ldots, m-k\}$. Given an $k$-tuple of ordinals $\boldsymbol{\gamma}$, a $\boldsymbol{\gamma}$-bounded *hsc-measure* is a tuple of functions $\boldsymbol{p} \colon [\boldsymbol{\gamma}] \to L^m$ satisfying

1. (*Monotonicity*) For $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in [\boldsymbol{\gamma}]$ and $a \in \underline{k}$, if $\boldsymbol{\alpha} \preceq_a \boldsymbol{\alpha}'$ then for all $i \geq i_a$ it holds $p_i(\boldsymbol{\alpha}) \sqsubseteq p_i(\boldsymbol{\alpha}')$.
2. ($\mu$-*case*) For $i \in \underline{m}$, $\eta_i = \mu$ and $i = i_a$ for some $a \in \underline{k}$ and $\boldsymbol{\alpha} = \boldsymbol{\alpha}'\alpha_a\boldsymbol{\alpha}'' \in [\boldsymbol{\gamma}]$, we have (i) $p_i(\boldsymbol{\alpha}', 0, \boldsymbol{\alpha}'') = \bot$; (ii) $p_i(\boldsymbol{\alpha}', \alpha{+}1, \boldsymbol{\alpha}'') \sqsubseteq f_i(\boldsymbol{p}(\boldsymbol{\beta}', \alpha, \boldsymbol{\alpha}''))$ for some $\boldsymbol{\beta}'$ and (iii) $p_i(\boldsymbol{\alpha}', \alpha, \boldsymbol{\alpha}'') \sqsubseteq \bigsqcup_{\beta<\alpha} f_i(\boldsymbol{p}^E(\boldsymbol{\alpha}', \beta, \boldsymbol{\alpha}''))$ for $\alpha$ a limit ordinal.
3. ($\nu$-*case*) For $i \in \underline{m}$, $\eta_i = \nu$, $i_{a-1} < i < i_a$ for some $a \in \underline{k}$ and $\boldsymbol{\alpha} = \boldsymbol{\alpha}'\alpha_a\boldsymbol{\alpha}'' \in [\boldsymbol{\gamma}]$, we have $p_i(\boldsymbol{\beta}', \alpha_a, \boldsymbol{\alpha}'') \sqsubseteq f_i(\boldsymbol{p}(\boldsymbol{\beta}', \alpha_a, \boldsymbol{\alpha}''))$ for some $\boldsymbol{\beta}'$.

Note that by point (1), for $a \in \underline{k}$ and $i \geq i_a$, the value of $p_i(\boldsymbol{\alpha})$ depends only the components of $\alpha$ of index greater or equal $a$. In fact for all $(m-a)$-tuples of ordinals $\boldsymbol{\alpha}', \boldsymbol{\beta}'$ and $a$-tuples of ordinals $\boldsymbol{\alpha}''$ we have $\boldsymbol{\alpha}'\boldsymbol{\alpha}'' \preceq_a \boldsymbol{\beta}'\boldsymbol{\alpha}'' \preceq_a \boldsymbol{\alpha}'\boldsymbol{\alpha}''$, hence $p_i(\boldsymbol{\alpha}', \boldsymbol{\alpha}'') \sqsubseteq p_i(\boldsymbol{\beta}', \boldsymbol{\alpha}'') \sqsubseteq p_i(\boldsymbol{\alpha}', \boldsymbol{\alpha}'')$. Thus $p_i(\boldsymbol{\alpha}', \boldsymbol{\alpha}'') = p_i(\boldsymbol{\beta}', \boldsymbol{\alpha}'')$.

As mentioned above, we can easily adapt the definition of $\mu$-approximant (Definition 3.9) to get a hsc-measure which can be shown to be the greatest one. Intuitively, the fact that $\mu$-approximants are closely related to the greatest hsc-meaure explains why our interest is mainly concentrated on $\mu$-approximants and their dual ($\nu$-approximants): the greatest hsc-measure surely provides a sound and complete approximation of the solution.

Theorem B.4 ($\mu$-approximants as hsc-measures). *Let $E$ be a system of $m$ equations over the lattice $L$, of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. Given $\boldsymbol{\alpha} \in [\lambda_L]^m$, define $\boldsymbol{p}^E(\alpha) = \boldsymbol{l}$ where for all $i \in \underline{m}$*

- *if $\eta_i = \nu$, then $l_i = \nu(f_{i,\boldsymbol{l}})$*
- *if $\eta_i = \mu$ with $i = i_a$ then $l_i = f_{i,\boldsymbol{l}}^{\alpha_a}(\bot)$*

*Then $\boldsymbol{p}$ is a $[\lambda_L]^m$-bounded hsc-measure and it is the greatest one.*

Proof. Let us start by proving that $p^E$ is a hsc-measure. Concerning monotonicity, let $a \in \underline{k}$, let $\boldsymbol{\alpha} \preceq_a \boldsymbol{\alpha}'$ and let $\boldsymbol{p}^E(\boldsymbol{\alpha}) = \boldsymbol{l}$ and $\boldsymbol{p}^E(\boldsymbol{\alpha}) = \boldsymbol{l}'$. We can show that for any $i \geq i_a$ it holds $l_i \sqsubseteq l_i'$, by means of an inductive argument (on $m - i$). The base case is $i = m$. Observe that $f_{m,\boldsymbol{l}} = f_{m,\boldsymbol{l}'}$ (as $f_{m,\boldsymbol{l}}$ is independent of $\boldsymbol{l}$). There are two possibilities.

- If $\eta_m = \mu$ then $i_k = m$. Since $\alpha_k \leq \alpha'_k$ we have $l_m = f_{m,\boldsymbol{l}}^{\alpha_k}(\bot) \sqsubseteq f_{m,\boldsymbol{l}}^{\alpha'_k}(\bot) = f_{m,\boldsymbol{l}'}^{\alpha'_k}(\bot) = l'_m$.
- If $\eta_m = \nu$ then we have $l_m = \nu(f_{m,\boldsymbol{l}}) = \nu(f_{m,\boldsymbol{l}'}) = l'_m$.

For $i_a \leq i < m$, observe that by inductive hypothesis $\boldsymbol{l}_{i+1,m} \sqsubseteq \boldsymbol{l'}_{i+1,m}$. Hence by monotonicity of the solution (Lemma 3.5) $sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := x]) \sqsubseteq sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l'}_{i+1,m}][x_i := x])$. Thus, using the fact that $f_i$ is monotonic, we have that

$$\begin{aligned} f_{i,\boldsymbol{l}}(x) &= f_i(sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := x]), x, \boldsymbol{l}_{i+1,m}) \\ &\sqsubseteq f_i(sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l'}_{i+1,m}][x_i := x]), x, \boldsymbol{l'}_{i+1,m}) = f_{i,\boldsymbol{l}'}(x). \end{aligned}$$

Given the above, reasoning as in the base case, we conclude $l_i \sqsubseteq l'_i$.

As a direct consequence of the definition of $\boldsymbol{p}^E$ we can show that the properties of the $\mu$-case and $\nu$-case in Definition B.3 hold with equality replacing $\sqsubseteq$ and the tuple of ordinals $\boldsymbol{\beta}' = (\lambda_L, \ldots, \lambda_L) = \boldsymbol{\lambda_L}$. More precisely

- (*$\mu$-case*) For $i \in \underline{m}$, $\eta_i = \mu$ and $i = i_a$ for some $a \in \underline{k}$ and $\boldsymbol{\alpha} = \boldsymbol{\alpha}\alpha_a\boldsymbol{\alpha''} \in [\boldsymbol{\gamma}]$, we have (i) $p_i(\boldsymbol{\alpha}', 0, \boldsymbol{\alpha}'') = \bot$; (ii) $p_i(\boldsymbol{\alpha}', \alpha + 1, \boldsymbol{\alpha}'') = f_i(\boldsymbol{p}^E(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}''))$, and (iii) $p_i(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') = \bigsqcup_{\beta<\alpha_a} f_i(\boldsymbol{p}^E(\boldsymbol{\alpha}', \beta, \boldsymbol{\alpha}''))$ for $\alpha$ a limit ordinal.
- (*$\nu$-case*) For $i \in \underline{m}$, $\eta_i = \nu$, $i_{a-1} < i < i_a$ for some $a \in \underline{k}$ and $\boldsymbol{\alpha} = \boldsymbol{\alpha}'\alpha_a\boldsymbol{\alpha''} \in [\boldsymbol{\gamma}]$, we have $p_i(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') = f_i(\boldsymbol{p}^E(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}''))$.

In fact

- (*$\mu$-case*) For $i \in \underline{m}$, $\eta_i = \mu$ and $i = i_a$ for some $a \in \underline{k}$ and $\boldsymbol{\alpha} = \boldsymbol{\alpha}\alpha_a\boldsymbol{\alpha''} \in [\boldsymbol{\gamma}]$, we have
  (i) (*Base*) If $\boldsymbol{p}^E(\boldsymbol{\alpha}', 0, \boldsymbol{\alpha}'') = \boldsymbol{l}$ then, by definition, we have
  $$p_i^E(\boldsymbol{\alpha}', 0, \boldsymbol{\alpha}'') = l_i = f_{i,\boldsymbol{l}}^0(\bot) = \bot$$
  (ii) (*Successor*) For the case of a successor ordinal, if we let $\boldsymbol{p}^E(\boldsymbol{\alpha}', \alpha_a + 1, \boldsymbol{\alpha}'') = \boldsymbol{l}$, we have:
  $$\begin{aligned} p_i^E(\boldsymbol{\alpha}', \alpha_a + 1, \boldsymbol{\alpha}'') &= f_{i,\boldsymbol{l}}^{\alpha_a+1}(\bot) \\ &= f_{i,\boldsymbol{l}}(f_{i,\boldsymbol{l}}^{\alpha_a}(\bot)) \\ &= f_i(sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := f_{i,\boldsymbol{l}}^{\alpha_a}(\bot)]), f_{i,\boldsymbol{l}}^{\alpha_a}(\bot), \boldsymbol{l}_{i+1,m}) \end{aligned}$$
  Let $\boldsymbol{l}' = (sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i+1,m}][x_i := f_{i,\boldsymbol{l}}^{\alpha_a}(\bot)]), f_{i,\boldsymbol{l}}^{\alpha_a}(\bot), \boldsymbol{l}_{i+1,m})$.
  Observe that for $j > i$ we have $l'_j = l_j = p_j(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}'')$ since $p_j$ only depends on $\boldsymbol{\alpha}_{j,m}$. Using this fact, we also get $l'_i = p_i(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}'')$. Finally, for $j < i$, it holds $l'_j = sol_j(E[\boldsymbol{x}_{i,m} := \boldsymbol{l'}_{i,m}])$. Hence, if $\eta_j = \nu$, we have $l'_j = \nu(f_{i,\boldsymbol{l}'}) = p_j^E(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}'')$ and, if $\eta_j = \mu$ we have $l'_j = \mu(f_{i,\boldsymbol{l}'}) = f_{i,\boldsymbol{l}'}^{\lambda_l}(\bot) = p_j^E(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}'')$.
  The above shows that $\boldsymbol{l}' = \boldsymbol{p}^E(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}'')$ and therefore $p_i^E(\boldsymbol{\alpha}', \alpha_a+1, \boldsymbol{\alpha}'') = f_i(\boldsymbol{p}^E(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}''))$, as desired.
  (iii) (*Limit*) If $\alpha_a$ is a limit ordinal, let $\boldsymbol{p}^E(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') = \boldsymbol{l}$. Then we have:
  $$p_i^E(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') = f_{i,\boldsymbol{l}}^{\alpha_a}(\bot) = \bigsqcup_{\beta<\alpha_a} f_{i,\boldsymbol{l}}^{\beta}(\bot) \tag{8}$$
  Now, observe that when taking the join above we can restrict to successor ordinals $\beta = \beta'+1$, and thus, reasoning as in the previous case, we get that $f_{i,\boldsymbol{l}}^{\beta}(\bot) = f_i(\boldsymbol{p}^E(\boldsymbol{\lambda_L}, \beta', \boldsymbol{\alpha}'')) =$

$p_i^E(\boldsymbol{\lambda_L}, \beta' + 1, \boldsymbol{\alpha}'') = p_i^E(\boldsymbol{\lambda_L}, \beta, \boldsymbol{\alpha}'') = p_i^E(\boldsymbol{\alpha}', \beta, \boldsymbol{\alpha}'')$, since $p_i^E$ only depends on the components $i, \ldots, m$ of its argument. Therefore, replacing in (8) we obtain

$$p_i^E(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') = \bigsqcup_{\beta < \alpha_a} p_i^E(\boldsymbol{\alpha}', \beta, \boldsymbol{\alpha}'') \tag{9}$$

as desired.

- ($\nu$-*case*) For $i \in \underline{m}$, $\eta_i = \nu$, $i_{a-1} < i < i_a$ for some $a \in \underline{k}$ and $\boldsymbol{\alpha} = \boldsymbol{\alpha}'\alpha_a\boldsymbol{\alpha}'' \in [\boldsymbol{\gamma}]$, if we let $\boldsymbol{p}^E(\boldsymbol{\alpha}', \beta', \boldsymbol{\alpha}'') = \boldsymbol{l}$ we have $p_i(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') = \nu(f_{i,\boldsymbol{l}})$. Therefore
$$\begin{aligned} p_i^E(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') &= l_i \\ &= f_{i,\boldsymbol{l}}(l_i) = \\ &= f_i(sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{l}_{i,m}]), \boldsymbol{l}_{i,m}) \end{aligned}$$
Let $\boldsymbol{l}' = (sol(E[\boldsymbol{x}_{i,m} := \boldsymbol{l}_{i,m}]), \boldsymbol{l}_{i,m})$. Observe that for $j \geq i$ we have $l'_j = l_j = p_j(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') = p_j(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}'')$ since $p_j$ only depends on $\boldsymbol{\alpha}_{j,m}$. Instead, for $j < i$, it holds $l'_j = sol_j(E[\boldsymbol{x}_{i,m} := \boldsymbol{l'}_{i,m}])$. Hence, if $\eta_j = \nu$, we have $l'_j = \nu(f_{i,\boldsymbol{l}'}) = p_j^E(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}'')$ and, if $\eta_j = \mu$ we have $l'_j = \mu(f_{i,\boldsymbol{l}'}) = f_{i,\boldsymbol{l}'}^{\lambda_l}(\bot) = p_j^E(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}'')$.
The above shows that $\boldsymbol{l}' = \boldsymbol{p}^E(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}'')$ and therefore $p_i^E(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') = f_i(\boldsymbol{p}^E(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}''))$, as desired.

This proves that $\boldsymbol{p}^E$ is a hsc-measure.

In addition, for any other progress measure $\boldsymbol{p}$ it holds that for any $\boldsymbol{\alpha} \in [\lambda_L]^m$ and $i \in \underline{m}$, we have $p_i(\boldsymbol{\alpha}) \sqsubseteq p_i^E(\boldsymbol{\alpha})$. The proof proceeds by induction on the ordinal vector $\boldsymbol{\alpha}$ with respect to the well-founded order $\preceq$. In order to show that for all $i \in \underline{m}$, $p_i(\boldsymbol{\alpha}) \sqsubseteq p_i^E(\boldsymbol{\alpha})$ we proceed by induction on $m - i$. If $\eta_i = \mu$, consider the index $a$ such that $i_a = i$. If $\alpha_a = 0$ then $p_i(\boldsymbol{\alpha}) = \bot = p_i^E(\boldsymbol{\alpha})$. For a successor ordinal $\alpha_a = \alpha + 1$,

$$\begin{aligned} p_i(\boldsymbol{\alpha}', \alpha + 1, \boldsymbol{\alpha}'') &\sqsubseteq f_i(\boldsymbol{p}(\boldsymbol{\beta}', \alpha, \boldsymbol{\alpha}'')) && \text{[for some } \boldsymbol{\beta}'\text{, by Def. B.3(2)]} \\ &\sqsubseteq f_i(\boldsymbol{p}(\boldsymbol{\lambda_L}, \alpha, \boldsymbol{\alpha}'')) && \text{[by Def. B.3(1) and monotonicity of } f_i\text{]} \\ &\sqsubseteq f_i(\boldsymbol{p}^E(\boldsymbol{\lambda_L}, \alpha_a, \boldsymbol{\alpha}'')) && \text{[by ind. hyp. and monotonicity of } f_i\text{]} \\ &= p_i^E(\boldsymbol{\lambda_L}, \alpha + 1, \boldsymbol{\alpha}'') && \text{[by } \mu\text{-case, property (ii) of } \boldsymbol{p}^E\text{]} \\ &= p_i^E(\boldsymbol{\alpha}', \alpha + 1, \boldsymbol{\alpha}'') && \text{[since } p_i^E \text{ only depends on components } i, \ldots, m\text{]} \end{aligned}$$

When $\alpha_a$ is a limit ordinal, by inductive hypothesis we know that for all $\beta < \alpha_a$ we have $p_i(\boldsymbol{\alpha}', \beta, \boldsymbol{\alpha}'') \sqsubseteq p_i^E(\boldsymbol{\alpha}', \beta, \boldsymbol{\alpha}'')$ and thus

$$\begin{aligned} p_i(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') &\sqsubseteq \bigsqcup_{\beta < \alpha_a} f_i(\boldsymbol{p}(\boldsymbol{\alpha}', \beta, \boldsymbol{\alpha}'')) && \text{[since } \boldsymbol{p} \text{ is a progress measure]} \\ &\sqsubseteq \bigsqcup_{\beta < \alpha_a} f_i(\boldsymbol{p}^E(\boldsymbol{\alpha}', \beta, \boldsymbol{\alpha}'')) && \text{[by ind. hyp. and monotonicity of } f_i\text{]} \\ &= \boldsymbol{p}^E(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') && \text{[by } \mu\text{-case, property (iii) of } \boldsymbol{p}^E\text{]} \end{aligned}$$

For the case $\eta_i = \nu$, let $a \in \underline{k}$ be the index such that $i_{a-1} < i < i_a$ and let $\boldsymbol{\alpha} = \boldsymbol{\alpha}'\alpha_a\boldsymbol{\alpha}'' \in [\boldsymbol{\gamma}]$. By Definition B.3(3), there must exist $\boldsymbol{\beta}'$ such that

$$p_i(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') \sqsubseteq f_i(\boldsymbol{p}(\boldsymbol{\beta}', \alpha_a, \boldsymbol{\alpha}'')). \tag{10}$$

Let $\boldsymbol{\beta} = (\boldsymbol{\beta}', \alpha_a, \boldsymbol{\alpha}'')$. By inner inductive hypothesis, for $j > i$

$$p_j(\boldsymbol{\beta}) \sqsubseteq p_j^E(\boldsymbol{\beta}) \tag{11}$$

Moreover, as shown in [Hasuo et al. 2016, Thm. 2.13] (property denoted by $(*)$), we have that for $j < i$ it holds that

$$p_j(\boldsymbol{\beta}) \sqsubseteq sol_j(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{p}_{i+1,m}(\boldsymbol{\beta})][x_i := p_i(\boldsymbol{\beta})])$$

In turn, by monotonicity of the solution (Lemma 3.5) and (11), we get

$$p_j(\boldsymbol{\beta}) \sqsubseteq sol_j(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{p^E}_{i+1,m}(\boldsymbol{\beta})][x_i := p_i(\boldsymbol{\beta})]) \tag{12}$$

Therefore, putting things together, from (10), we get

$$\begin{aligned}
&p_i(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'')\\
&\sqsubseteq f_i(\boldsymbol{p}(\boldsymbol{\beta}))\\
&\sqsubseteq f_i(sol(E[\boldsymbol{x}_{i+1,m} := \boldsymbol{p^E}_{i+1,m}(\boldsymbol{\beta})][x_i := p_i(\boldsymbol{\beta})]), p_i(\boldsymbol{\beta}), \boldsymbol{p^E}_{i+1,m}(\boldsymbol{\beta}))\\
&\quad \text{[by (11) and (12) and monotonicity of } f_i]\\
&\sqsubseteq f_{i,\boldsymbol{p}^E(\boldsymbol{\beta})}(p_i(\boldsymbol{\beta}))
\end{aligned}$$

Recalling that $p_i$ only depends on components $i+1, \ldots, m$, we have that $p_i(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') = p_i(\boldsymbol{\beta}', \alpha_a, \boldsymbol{\alpha}'') = p_i(\boldsymbol{\beta})$, i.e., the inequality above can be rewritten as

$$p_i(\boldsymbol{\beta}) \sqsubseteq f_{i,\boldsymbol{p}^E(\boldsymbol{\beta})}(p_i(\boldsymbol{\beta})).$$

This means that $p_i(\boldsymbol{\beta})$ is a post-fixpoint of $f_{i,\boldsymbol{p}^E(\boldsymbol{\beta})}$, and by definition of $\boldsymbol{p}^E$, we have that $p_i^E(\boldsymbol{\beta})$ is the greatest fixpoint of $f_{i,\boldsymbol{p}^E(\beta)}$. Therefore $p_i(\boldsymbol{\beta}) \sqsubseteq p_i^E(\boldsymbol{\beta})$. Recalling that $\boldsymbol{\beta} = (\boldsymbol{\beta}', \alpha_a, \boldsymbol{\alpha}'')$ and, again, that $p_i$ only depends on components $i+1, \ldots, m$, we conclude the desired inequality

$$p_i(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'') = p_i(\boldsymbol{\beta}) \sqsubseteq p_i^E(\boldsymbol{\beta}) = p_i^E(\boldsymbol{\alpha}', \alpha_a, \boldsymbol{\alpha}'').$$

□

Note that the characterisation of $\boldsymbol{p}^E$ used in the proof offers an alternative method for computing approximations of the fixpoint and could be interesting in its own right, for instance for cases where the basis is too large – for instance for the reals – and it is infeasible to determine the progress measure for every element of the basis.

The notion of *matrix progress measure* (MPM) in [Hasuo et al. 2016], which is introduced for powerset lattices, is closely related to the game-theoretical progress measure that we proposed for equations over continuous lattices: it can be seen as an instance of our notion for systems of equations arising from formulae in the coalgebraic $\mu$-calculus.

# C TECHNICAL RESULTS

## C.1 Sup-Respecting Progress Measures (§ 5.2)

In order to show that $\Phi_E$ preserves sup-respecting functions $R$ we first need a technical lemma that will also prove useful for the logic characterising symbolic $\exists$-moves.

LEMMA C.1. *Let $L$ be a continuous lattice and let $(U_k)_{k\in K}$ with $U_k \subseteq L^m$ be a collection of upward-closed sets. Assume that $R\colon L \to \underline{m} \to [\lambda_L]_\star^m$ is sup-respecting. Then it holds that:*

$$\begin{aligned}
&\min\nolimits_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid j \in \underline{m} \ \wedge\ b' \ll l_j\} \mid \boldsymbol{l} \in \bigcap_{k\in K} U_k\}\\
=\ &\sup_{k\in K} \min\nolimits_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid j \in \underline{m} \ \wedge\ b' \ll l_j\} \mid \boldsymbol{l} \in U_k\}
\end{aligned}$$

Proof. Since all $U_k$ are upward-closed, their intersection can be written as $\bigcap_{k\in K} U_k = \{\bigsqcup_{k\in K} \boldsymbol{l}^k \mid \boldsymbol{l}^k \in U_k, k \in K\}$ (where suprema of $L^m$ are taken pointwise). Hence the left-hand side of the equation can be rewritten to

$$\min\nolimits_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid j \in \underline{m} \ \wedge \ b' \ll \bigsqcup_{k\in K} l_j^k\} \mid \boldsymbol{l}^k \in U_k, k \in K\}$$

We first show that for $j \in \underline{m}$

$$\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid b' \ll \bigsqcup_{k\in K} l_j^k\} = \sup_{k\in K} \sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid b' \ll l_j^k\}$$

- ($\sqsupseteq$) This direction is obvious since $b' \ll l_j^k$ implies $b' \ll \bigsqcup_{k\in K} l_j^k$. Hence every ordinal vector of the form $R(b')(j) + \boldsymbol{\delta}_i^{\eta_i}$ which is contained in the right-hand side set is automatically a member of the left-hand side set.
- ($\sqsubseteq$) Let $b' \ll \bigsqcup_{k\in K} l_j^k$. This implies that $b' \ll \bigsqcup_{k\in K} l_j^k = \bigsqcup Y$ where $Y = \bigcup_{k\in K}(\Downarrow l_j^k \cap B_L)$, since we are in a continuous lattice. Then there exists a finite subset $Y' \subseteq Y$ such that $b' \sqsubseteq \bigsqcup Y'$ (see [Gierz et al. 2003, Remark on p. 50]).
  Since $R$ is sup-respecting we have

$$\begin{aligned} R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} &\preceq \big(\sup_{y\in Y'} R(y)(j)\big) + \boldsymbol{\delta}_i^{\eta_i} \\ &= \sup_{y\in Y'} \big(R(y)(j) + \boldsymbol{\delta}_i^{\eta_i}\big) \\ &\preceq \sup_{y\in Y} \big(R(y)(j) + \boldsymbol{\delta}_i^{\eta_i}\big) \\ &= \sup_{k\in K} \sup\{R(y)(j) + \boldsymbol{\delta}_i^{\eta_i} \mid y \in Y \cap \Downarrow l_j^k\} \\ &= \sup_{k\in K} \sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid b' \ll l_j^k\} \end{aligned}$$

  Note that the first equality is due to the fact that $Y'$ is finite and non-empty.
  Since the left-hand side of the equation is the supremum of all such $R(b')(j)$ and we have shown that the right-hand side is an upper bound, the result follows.

Now we can conclude by showing that

$$\begin{aligned} &\min\nolimits_{\preceq_i}\{\sup_{j\in\underline{m}} \sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid b' \ll \bigsqcup_{k\in K} l_j^k\} \mid \boldsymbol{l}^k \in U_k, k \in K\} \\ = \ &\min\nolimits_{\preceq_i}\{\sup_{j\in\underline{m}} \sup_{k\in K} \sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid b' \ll l_j^k\} \mid \boldsymbol{l}^k \in U_k, k \in K\} \\ = \ &\min\nolimits_{\preceq_i}\{\sup_{k\in K} \sup_{j\in\underline{m}} \sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid b' \ll l_j^k\} \mid \boldsymbol{l}^k \in U_k, k \in K\} \\ = \ &\sup_{k\in K} \min\nolimits_{\preceq_i}\{\sup_{j\in\underline{m}} \sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid b' \ll l_j\} \mid \boldsymbol{l} \in U_k\} \\ = \ &\sup_{k\in K} \min\nolimits_{\preceq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid j \in \underline{m} \wedge b' \ll l_j\} \mid \boldsymbol{l} \in U_k\} \end{aligned}$$

where the second-last equality is due to complete distributivity. □

Lemma C.2 ($\Phi_E$ preserves sup-respecting functions). *Let $L$ be a continuous lattice and let $E$ be a system of equations over $L$ of the kind $\boldsymbol{x} =_{\boldsymbol{\eta}} \boldsymbol{f}(\boldsymbol{x})$. If $R \colon B_L \to \underline{m} \to [\lambda_L]_\star^m$ is sup-respecting, then $\Phi_E(R)$ is sup-respecting as well.*

Proof. We assume that $R$ is sup-respecting and $R' = \Phi_E(R)$ is as follows:

$$R'(b)(i) = \min_{\leq_i}\{\sup\{R(b')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b', j) \in \mathbf{A}(\boldsymbol{l})\} \mid \boldsymbol{l} \in \mathbf{E}(b, i)\}$$

The aim is to show that $R'$ is sup-respecting as well. Let $X \subseteq B_L$ be a set of basis elements such that $b \sqsubseteq \bigsqcup X$. Note furthermore that $\mathbf{E}(b, i)$ is upwards-closed. We first show that

$$\bigcap_{b' \in X} \mathbf{E}(b', i) \subseteq \mathbf{E}(b, i)$$

Let $\boldsymbol{l} \in \mathbf{E}(b', i)$ for all $b' \in X$, which means that $b' \sqsubseteq f_i(\boldsymbol{l})$. So if we take the supremum over all $b' \in X$ we obtain $b \sqsubseteq \bigsqcup X \sqsubseteq f_i(\boldsymbol{l})$. Hence $\boldsymbol{l} \in \mathbf{E}(b, i)$, as required.

Now we can apply Lemma C.1 where $K = X$, $U_{b'} = \mathbf{E}(b', i)$ and we obtain:

$$\begin{aligned}
R'(b)(i) &= \min_{\leq_i}\{\sup\{R(b'')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b'', j) \in \mathbf{A}(\boldsymbol{l})\} \mid \boldsymbol{l} \in \mathbf{E}(b, i)\} \\
&\leq \min_{\leq_i}\{\sup\{R(b'')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b'', j) \in \mathbf{A}(\boldsymbol{l})\} \mid \boldsymbol{l} \in \bigcap_{b' \in X} \mathbf{E}(b', i)\} \\
&= \sup_{b' \in X} \min_{\leq_i}\{\sup\{R(b'')(j) + \boldsymbol{\delta}_i^{\eta_i} \mid (b'', j) \in \mathbf{A}(\boldsymbol{l})\} \mid \boldsymbol{l} \in \mathbf{E}(b', i)\} \\
&= \sup_{b' \in X} R'(b')(i)
\end{aligned}$$

□

## C.2 Compositionality for Selections (§ 5.3.1)

In order to define selections compositionally we first need a technical lemma that extends selections to generic elements of the lattice, possibly not part of the basis.

Lemma C.3 (extending the selection). *Let $L$ be a continuous lattice with a basis $B_L$, let $f\colon L^m \to L$ be a monotone functions and let $\sigma\colon B_L \to 2^{L^m}$ be a selection for $f$. Define $\bar{\sigma}\colon L \to 2^{L^m}$ by $\bar{\sigma}(b) = \sigma(b)$ for $b \in B_L$ and $\bar{\sigma}(l) = \{\bigsqcup_{b \ll l} \boldsymbol{l}^b \mid \boldsymbol{l}^b \in \sigma(b)\}$ for $l \in L \setminus B_L$. Then*

*(1) for all $\boldsymbol{l} \in \bar{\sigma}(l)$ it holds $l \sqsubseteq f(\boldsymbol{l})$;*
*(2) for all $\boldsymbol{l}' \in L^m$, if $l \sqsubseteq f(\boldsymbol{l}')$ then there exists $\boldsymbol{l} \in \bar{\sigma}(l)$ such that $\boldsymbol{l} \sqsubseteq \boldsymbol{l}'$.*

Proof. For $l \in B_L$, there is nothing to prove since the properties hold by definition of selection.

Let $l \in L \setminus B_L$. We start with point (1). Let $\boldsymbol{l} \in \sigma(l)$, hence $\boldsymbol{l} = \bigsqcup_{b \ll l} \boldsymbol{l}^b$ with $\boldsymbol{l}^b \in \sigma(b)$ for each $b \ll l$. By the properties of selections, for all $b \ll l$, since $\boldsymbol{l}^b \in \sigma(l)$ it holds $b \sqsubseteq f(\boldsymbol{l}^b)$, hence

$$b \sqsubseteq \bigsqcup_{b' \ll l} f(\boldsymbol{l}^b) \sqsubseteq f(\bigsqcup_{b' \ll l} \boldsymbol{l}^b) = f(\boldsymbol{l})$$

the last inequality following by monotonicity of $f$. Therefore $l = \bigsqcup_{b \ll l} b \sqsubseteq f(\boldsymbol{l})$, as desired.

Concerning point (2), let $\boldsymbol{l}' \in L^m$ be such that $l \sqsubseteq f(\boldsymbol{l}')$. For all $b \ll l$, since $b \sqsubseteq f(\boldsymbol{l}')$ there is $\boldsymbol{l}^b \in \sigma(b)$ such that $\boldsymbol{l}^b \sqsubseteq \boldsymbol{l}'$. Then we can consider $\boldsymbol{l} = \bigsqcup_{b \ll l} \boldsymbol{l}^b \sqsubseteq \boldsymbol{l}'$ which is in $\bar{\sigma}(l)$ by definition. □

We can now define the selection for a composition of functions.

Lemma C.4 (selection for composition). *Let $L$ be a continuous lattice with a basis $B_L$, and let $f\colon L^n \to L$ and $f_j\colon L^m \to L$, $j \in \underline{n}$ be monotone functions and let $\sigma\colon B_L \to 2^{L^n}$ and $\sigma_j\colon B_L \to 2^{L^m}$, $j \in \underline{n}$ be the corresponding selections. Consider the function $h\colon L^m \to L$ obtained as the composition $h(\boldsymbol{l}) = f(f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l}))$. Then $\sigma'\colon B_L \to 2^{L^m}$ defined by*

$$\sigma'(b) = \{\textstyle\bigsqcup_{i=1}^n \boldsymbol{l}^i \mid \exists \boldsymbol{l} \in \sigma(b). \forall i \in \underline{n}.\, \boldsymbol{l}^i \in \bar{\sigma}_i(l_i)\}$$

*is a selection for $h$.*

Proof. We show properties (1) and (2) of Definition 5.11. Let $b \in B_L$. Concerning (1), let $\boldsymbol{l}' \in \sigma'(b)$. Hence $\boldsymbol{l}' = \bigsqcup_{i=1}^{n} \boldsymbol{l}^i$ such that, for some $\boldsymbol{l} \in \sigma(b)$, for all $i \in \underline{n}$ we have $\boldsymbol{l}^i \in \bar{\sigma}_i(l_i)$. Since $\boldsymbol{l}^i \in \bar{\sigma}(l_i)$, by Lemma C.3 and monotonicity of $f_i$ we have $l_i \sqsubseteq f_i(\boldsymbol{l}^i) \sqsubseteq f_i(\bigsqcup_{i=1}^{n} \boldsymbol{l}^i) = f_i(\boldsymbol{l}')$. Hence $\boldsymbol{l} \sqsubseteq (f_1(\boldsymbol{l}'), \ldots, f_n(\boldsymbol{l}'))$ and thus, by monotonicity of $f$,

$$f(\boldsymbol{l}) \sqsubseteq f(f_1(\boldsymbol{l}'), \ldots, f_n(\boldsymbol{l}')) = h(\boldsymbol{l}')$$

Recalling that $\boldsymbol{l} \in \sigma(b)$ and thus $b \sqsubseteq f(\boldsymbol{l})$ we conclude, by transitivity, $b \sqsubseteq h(\boldsymbol{l}')$, as desired.

Let us focus on property (2). Let $\boldsymbol{l} \in L^m$ be such that $b \sqsubseteq h(\boldsymbol{l}) = h(f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l}))$. Since $\sigma$ is a selection for $f$, there exists $\boldsymbol{l}' \in \sigma(b)$ such that $\boldsymbol{l}' \sqsubseteq (f_1(\boldsymbol{l}), \ldots, f_n(\boldsymbol{l}))$. Now, for all $i \in \underline{n}$, since $l'_i \sqsubseteq f_i(\boldsymbol{l})$, by Lemma C.3, there is $\boldsymbol{l}^i \in \bar{\sigma}_i(l'_i)$ such that $\boldsymbol{l}^i \sqsubseteq \boldsymbol{l}$. If we let $\boldsymbol{l}'' = \bigsqcup_{i=1}^{n} \boldsymbol{l}^i$, by definition $\boldsymbol{l}'' \in \sigma'(b)$ and clearly $\boldsymbol{l}'' \sqsubseteq \boldsymbol{l}$, as desired. □

*Example C.5.* Consider again our running example in Example 3.7. The selection discussed in Example 5.15 is computed using the observation in Example 5.13 and Lemma C.4.

The fact that in this case the selections $\sigma_1$ and $\sigma_2$ arising from the construction in Lemma C.4 are the least ones is not a general fact. In order to ensure that starting from the least selections of the components we get the least selection we need to consider a more complex definition of extension (Lemma C.3) that is omitted since we favour the use of the logic for symbolic ∃-moves (§ 5.3.2).