Logical Methods in Computer Science
Volume 20, Issue 3, 2024, pp. 5:1–5:32
https://lmcs.episciences.org/

Submitted Apr. 17, 2023
Published Jul. 10, 2024

OPERATIONS ON FIXPOINT EQUATION SYSTEMS

THOMAS NEELE [a] AND JACO VAN DE POL [b]

[a] Eindhoven University of Technology, The Netherlands
e-mail address: t.s.neele@tue.nl

[b] Aarhus University, Denmark
e-mail address: jaco@cs.au.dk

Abstract. We study operations on fixpoint equation systems (FES) over arbitrary complete lattices. We investigate under which conditions these operations, such as substituting variables by their definition, and swapping the ordering of equations, preserve the solution of a FES. We provide rigorous, computer-checked proofs. Along the way, we list a number of known and new identities and inequalities on extremal fixpoints in complete lattices.

1. Introduction

This paper deals with operations on systems of fixpoint equations over an arbitrary complete lattice. We investigate when these operations preserve the solution of the equations. An example of a system of equations is the set $\mathcal{E} := \{X = f(X,Y,Z),\ Y = g(X,Y,Z),\ Z = h(X,Y,Z)\}$. For most results, it is required that the functions f, g, h are monotonic in the given lattice. Such systems may well have multiple solutions. In order to specify particular solutions, we introduce specifications, for example $\mathcal{S} := [\mu X, \nu Y, \mu Z]$, indicating for each variable whether we are interested in the minimal (μ) or maximal (ν) solution. The order of the variables in these specifications is relevant: the leftmost variable indicates the fixpoint with the highest priority. A *Fixpoint Equation System* (FES) [Mad97] is a pair $(\mathcal{E}, \mathcal{S})$, where $\mathcal{E}$ is a set of equations, and $\mathcal{S}$ is a specification. Several well known instances are obtained by instantiating the complete lattice.

Well-known instances of FES. *Boolean Equation Systems* (BES) arise as FES over the complete lattice $\bot < \top$, and were proposed in [And94, AV95] for solving the model checking and equivalence checking problems on finite labeled transition systems (LTS). BES received extensive study in [Mad97, MS03, GK04, Mat06]. A generalisation to the domain $\mathbb{R} \cup \{-\infty, \infty\}$ is *real equation systems* (RES) [GW23].

An equivalent notion to BES is two-player *parity games* [EJ91], see [Mad97] for a proof. Algorithms for solving parity games receive a lot of attention, since this is one of the few problems which is in NP and in co-NP, but not known to be in P. Recently, it has been shown that parity games (and thus BES) can be solved in quasi-polynomial time [CJK$^+$17]. This result has also been lifted to the general setting of FES on finite lattices [HS21, JMT22].

Key words and phrases: fixpoint equation system, complete lattice.

Other types of games can also be seen as an instance of FES, for example energy parity games [CD12] are FES on the lattice $\mathbb{Z} \to \{\bot, \top\}$, ordered pointwise. A modern parity game solver is Oink [vD18].

Parameterised Boolean Equation Systems (PBES, also known as first-order, or predicate BES) arise as FES over the powerset lattice $(2^D, \subseteq)$, with D some data type, typically representing the state space of a possibly infinite LTS. In [Mat98, GM99], PBES are proposed to encode the model-checking problem of first-order mu-calculus on infinite LTSs; they are studied in more detail in [GW05b, GW05a]. An encoding of (branching) bisimulation of infinite LTSs in PBES is proposed in [CPvdPW07]. Various procedures that operate on PBES have been studied, for example to simplify [Nee22, OW10] or solve PBES [NWG20, NWWV22, PWW11]. Algorithms for solving some timed fragments of PBES automatically are studied in [ZC05]. PBES are implemented in the mCRL2 [BGK$^+$19] and CADP [GLMS13] model checking toolsets. MuArith [KFG20] is similar to PBES, but the domain D is restricted to integers.

Fixpoint Equation Systems over arbitrary complete lattices (FES) are defined in [Mad97, TC02]. Some works refer to the same concept as *Hierarchical Equation Systems* (HES) [Sei96, KNIU19], *Systems of Fixpoint Equations* [BKP20] or *Nested Fixpoint Equations* [JMT22]. In [ZC05] it is recognized that BES and PBES (and also *Modal Equation Systems* [CS93], an equational representation of the modal mu-calculus) are instances of FES. FESs are mainly useful to provide generic definitions for all these kinds of equation systems. We claim that the generic semantics of a FES is more elegant than the semantics of PBES, as given in e.g. [GW05a]. In particular, equations in FES are defined in a semantic manner as functions on valuations, rather than on syntactic expressions (possibly with binders). Another advantage of FES is that one can derive a number of basic theorems for equation systems over all lattices in one stride, like in Chapter 3 of Mader's thesis [Mad97].

Abstract dependency graphs [EGLS19] are similar to FES, but variables range over a Noetherian partial order with a least element, instead of a complete lattice. When assuming every right-hand side is effectively computable, minimal fixpoints can be computed in an iterative fashion. Dependency graphs do not contain fixpoint alternations.

Contributions. Our main goal is to study basic operations on FES, related with substituting variables in the equations by their definition or final solution, or swapping the order of equations in the specification. Substitution operations form the basis of solving BES by so-called Gauss-elimination [Mad97]. Also for PBES, Gauss elimination plays a crucial role in their solution. Reordering the variables in the specification is useful, because it may give rise to independent subspecifications that can be solved separately. Also, swapping the order of variables may bring down the number of alternations between μ and ν, thus lowering the complexity of certain solution algorithms.

Our results consist of equalities and inequalities between FES, expressing under which conditions the basic operations preserve the solution of a FES. The main results are summarized in Table 1 (Section 7). In particular:

(1) Results on substitution for BES and PBES are generalized to FES.
(2) Results on swapping variables are generalized and sorted out, by weakening existing conditions, and by providing alternative conditions.
(3) We provide rigorous proofs of our results. All proofs in this paper have been proof-checked mechanically by the Coq theorem prover [Ber08, S$^+$23] (version 8.17) as well as the PVS theorem prover [OS08] (version 7.1). Our proofs are available online [NvdP24].

Overview. We first provide the basic theory of complete lattices in Section 2 and reprove all needed facts on fixpoints, in order to present a self-contained account. The formal definition and semantics of Fixpoint Equation Systems is provided in Section 3. The proofs (Section 4, 5 and 6) are quite elementary. They are mainly based on induction (to deal with the recursive definition of FES semantics, Section 3.1) and on identities and inequalities on fixpoints in complete lattices. In Section 7, we provide examples of applications of our theory and discuss its relation with the literature. Finally, we highlight several aspects of our Coq and PVS formalisations in Section 8.

2. Fixpoint Laws in Complete Lattices

A *partial order* on a universe U is a binary relation $\leq\ \subseteq U \times U$, which is *reflexive* ($\forall x.\, x \leq x$), *anti-symmetric* ($\forall x, y.\, x \leq y \wedge y \leq x \Rightarrow x = y$) and *transitive* ($\forall x, y, z.\, x \leq y \wedge y \leq z \Rightarrow x \leq z$), where in all cases $x, y, z \in U$.

Given partial orders $(U, \leq)$ and $(V, \leq)$, we define partial orders $(U \times V, \leq)$ and $(U \to V, \leq)$ pointwise: $(u_1, v_1) \leq (u_2, v_2)$ iff $u_1 \leq u_2 \wedge v_1 \leq v_2$, and $f \leq g$ iff $\forall x \in U. f(x) \leq g(x)$. Function $f : U \to V$ is called *monotonic*, iff $\forall x, y.\, x \leq y \Rightarrow f(x) \leq f(y)$.

Given a set $X \subseteq U$, we define its set of *lower bounds* in U as $lb(X) := \{y \in U \mid \forall x \in X.\, y \leq x\}$. If $y \in lb(X)$ and $z \leq y$ for all $z \in lb(X)$, then y is called the *greatest lower bound* of X. A *complete lattice* is a triple $(U, \leq, glb)$, where $\leq$ is a partial order, and $glb(X)$ returns the greatest lower bound of X in U, for all (finite or infinite) $X \subseteq U$.

Given a complete lattice $(U, \leq, glb)$, define the partial order $(U, \geq)$, by $x \geq y$ iff $y \leq x$. We define the set of *upper bounds* of $X \subseteq U$ by $ub(X) := \{y \in U \mid \forall x \in X.\, y \geq x\}$. Define $lub(X) := glb(ub(X))$. Clearly, for all $y \in ub(X)$, $lub(X) \leq y$. But also $lub(X) \in ub(X)$, for if $x \in X$, then $x \in lb(ub(X)$, hence $x \leq glb(ub(X)$. So $lub(X)$ yields the *least upper bound* of X, and $(U, \geq, lub)$ is a complete lattice as well.

Given a complete lattice $(U, \leq, glb)$, we define the *least fixpoint* (μ) and *greatest fixpoint* (ν) of any function $f : U \to U$ (not only for monotonic) as follows:

$$\begin{array}{rcl} \mu(f) & := & glb(\{x \mid f(x) \leq x\}) \\ \nu(f) & := & lub(\{x \mid x \leq f(x)\}) \end{array}$$

For $\sigma \in \{\mu, \nu\}$, we abbreviate $\sigma(\lambda x.f(x))$ by $\sigma x.f(x)$. Note that by definition, ν in $(U, \leq, glb)$ equals μ in $(U, \geq, lub)$, so theorems on $(\mu, \leq)$ hold for $(\nu, \geq)$ as well "by duality". Also note that $F : U \to U$ is monotonic in $(U, \leq)$ if and only if it is monotonic in $(U, \geq)$. A direct consequence of the definition of μ is the following principle (and its dual):

$$f(x) \leq x \;\Rightarrow\; \mu(f) \leq x \qquad (\mu\text{-fixpoint induction})$$
$$x \leq f(x) \;\Rightarrow\; x \leq \nu(f) \qquad (\nu\text{-fixpoint induction})$$

We now have the following identities on fixpoint expressions:

Lemma 2.1. *Let $(U, \leq, glb)$ be a complete lattice. Let $\sigma \in \{\mu, \nu\}$, $A \in U$, and let $F, G \in U \to U$ and $H, K \in U \times U \to U$ be monotonic functions. Then:*

(1) $F(\sigma(F)) = \sigma(F)$ *(computation rule)*
(2) $\sigma x.\, A = A$ *(constant rule)*
(3) $\sigma x.\, F(G(x)) = F(\sigma x.\, G(F(x)))$ *(rolling rule)*
(4) $\sigma x.\, F(F(x)) = \sigma x.\, F(x)$ *(square rule)*
(5) σ *is monotonic* *(fixpoint monotonicity)*
(6) $\sigma x.\, H(x, x) = \sigma x.\, \sigma y.\, H(x, y)$ *(diagonal rule)*

(7) $\sigma x.\, H(x,x) = \sigma x.\, H(x, H(x,x))$ *(unfolding rule)*
(8) $\sigma x.\, H(x,x) = \sigma x.\, H(x, \sigma x.\, H(x,x))$ *(solve rule)*
(9) $\sigma x.\, H(x, \sigma y.\, K(y,x)) = \sigma x.\, H(x, \sigma y.\, K(y, \sigma z.\, H(z,y)))$ *(Bekič rule)*

Proof. We first prove the theorem for $\sigma = \mu$. By the observations above, the theorem then follows for $\sigma = \nu$ as well ("by duality").

(1) (a) Let y with $F(y) \leq y$ be given. Then by fixpoint induction, $\mu(F) \leq y$. By monotonicity, $F(\mu(F)) \leq F(y) \leq y$. Since y is arbitrary, $F(\mu(F))$ is a lower bound of $\{x \mid F(x) \leq x\}$. Hence $F(\mu(F)) \leq glb(\{x \mid F(x) \leq x\}) = \mu(F)$
(b) $F(\mu(F)) \leq \mu(F)$ by (a), so by monotonicity, $F(F(\mu(F))) \leq F(\mu(F))$. By fixpoint induction, $\mu(F) \leq F(\mu(F))$.
Then by anti-symmetry $F(\mu(F)) = \mu(F)$.
(2) Follows directly from (1) by taking $F := \lambda x.A$ (which is monotonic)
(3) Obviously, $\lambda x.\, F(G(x))$ and $\lambda x.\, G(F(x))$ are monotonic.
(a) By (1), $F(G(F(\mu x.\, G(F(x))))) = F(\mu x.\, G(F(x)))$. Hence by fixpoint induction, $\mu x.\, F(G(x)) \leq F(\mu x.\, G(F(x)))$.
(b)

$$\begin{aligned}
& G(F(G(\mu x.\, F(G(x))))) \overset{(1)}{=} G(\mu x.\, F(G(x))) \\
\Rightarrow\ & \text{(by fixpoint induction)} \\
& \mu x.\, G(F(x)) \leq G(\mu x.\, F(G(x))) \\
\Rightarrow\ & \text{(by monotonicity)} \\
& F(\mu x.\, G(F(x))) \leq F(G(\mu x.\, F(G(x)))) \overset{(1)}{=} \mu x.\, F(G(x))
\end{aligned}$$

By anti-symmetry, we obtain $\mu x.\, F(G(x)) = F(\mu x.\, G(F(x)))$.
(4) (a) Using (1) twice, $F(F(\mu x.\, F(x))) = F(\mu x.\, F(x)) = \mu x.\, F(x)$. So by fixpoint induction, $\mu x.\, F(F(x)) \leq \mu x.\, F(x)$.
(b) By (3), we get $F(\mu x.\, F(F(x))) = \mu x.\, F(F(x))$. Hence by fixpoint induction, $\mu x.\, F(x) \leq \mu x.\, F(F(x))$.
Then by anti-symmetry, $\mu x.\, F(F(x)) = \mu x.\, F(x)$.
(5) Assume $f \leq g$. Let y with $g(y) \leq y$ be given. Then $f(y) \leq g(y) \leq y$, so $\mu(f) \leq y$ by fixpoint induction. Since y is arbitrary, $\mu(f)$ is a lower bound for $\{x \mid g(x) \leq x\}$. By definition $\mu(g)$ is its greatest lower bound, so $\mu(f) \leq \mu(g)$.
(6) (a)

$$\begin{aligned}
& H(\mu x.\, H(x,x), \mu x.\, H(x,x)) \overset{(1)}{=} \mu x.\, H(x,x) \\
\Rightarrow\ & \text{(by fixpoint induction, applied with } F := \lambda x.\, H(x,x)) \\
& \mu y.\, H(\mu x.\, H(x,x), y) \leq \mu x.\, H(x,x) \\
\Rightarrow\ & \text{(by fixpoint induction)} \\
& \mu x.\, \mu y.\, H(x,y) \leq \mu x.\, H(x,x)
\end{aligned}$$

(b) Let us abbreviate $A := \mu x.\, \mu y.\, H(x,y)$. Using (5) one can show that $\lambda x.\, \mu y.\, H(x,y)$ is monotonic. Then:

$$\begin{aligned}
& A \overset{(1)}{=} \mu y.\, H(A,y) \overset{(1)}{=} H(A, \mu y.\, H(A,y)) \\
\Rightarrow\ & \text{(by congruence and both equations above)} \\
& H(A,A) = H(A, \mu y.\, H(A,y)) = A
\end{aligned}$$

$$\begin{aligned} \Rightarrow\ & \text{(by fixpoint induction)} \\ & \mu x.\, H(x,x) \leq A = \mu x.\, \mu y.\, H(x,y) \end{aligned}$$

By anti-symmetry, we indeed get: $\mu x.\, H(x,x) = \mu x.\, \mu y.\, H(x,y)$.

(7) Using (4) on $\lambda y.\, H(x,y)$ yields $\mu x.\, \mu y.\, H(x,y) = \mu x.\, \mu y.\, H(x, H(x,y))$. Applying (6) to both sides yields $\mu x.\, H(x,x) = \mu x.\, H(x, H(x,x))$.

(8) We use (6) twice on the function $\lambda(y,x).H(x,y)$:

$$\begin{aligned} & \mu x.\, H(x,x) \\ \overset{(6)}{=}\ & \mu y.\, \mu x.\, H(x,y) \\ \overset{(1)}{=}\ & \mu x.\, H(x, \mu y.\, \mu x.\, H(x,y)) \\ \overset{(6)}{=}\ & \mu x.\, H(x, \mu x.\, H(x,x)) \end{aligned}$$

(9) Define $F(y) := \mu x.\, H(x,y)$ and $G(y) := \mu x.\, K(x,y)$. Then:

$$\begin{aligned} & \mu y.\, F(G(y)) \overset{(3)}{=} F(\mu y.\, G(F(y))) \\ \Rightarrow\ & \text{(by definition of } F,\ G\text{)} \\ & \mu y.\, \mu x.\, H(x, G(y)) = F(\mu y.\, \mu x.\, K(x, F(y))) \\ \Rightarrow\ & \text{(by 6, applied to left- and right-hand side)} \\ & \mu x.\, H(x, G(x)) = F(\mu y.\, K(y, F(y))) \\ \Rightarrow\ & \text{(by definition of } F,\ G\text{)} \\ & \mu x.\, H(x, \mu y.\, K(y,x)) = \mu x.\, H(x, \mu y.\, K(y, \mu z.\, H(z,y))) \end{aligned}$$ □

A careful analysis shows that all these identities can be derived in an equational style from the identities 1, 3, 4 and 6. A natural question is whether all true equalities (with μ as second order operation, and variables ranging over monotonic functions) can be derived from these four identities in an equational manner (thus excluding the fixpoint induction rule). We don't know the answer, but we expect that at least the equations $\mu x.\, F(x) = \mu x.\, F^p(x)$ are needed for all primes p. Results from universal algebra don't apply directly, due to the second order nature of the fixpoint operator.

By mixing least and greatest fixpoints, we also obtain a number of inequalities. In particular, 4 is new, as far as we know. Note the similarity of (4) with Bekič Rule, Lemma 2.1. We will call (4) Bekič Inequality.

Lemma 2.2. *Let $(U, \leq, glb)$ be a complete lattice. Let $\sigma \in \{\mu, \nu\}$, $A \in U$, and let $F, G \in U \to U$ and $H, K \in U \times U \to U$ be monotonic functions. Then:*

(1) $\mu(F) \leq \nu(F)$
(2) (a) $\mu x.x \leq A$
  (b) $A \leq \nu x.x$
(3) $\mu x.\, \nu y.\, H(x,y) \leq \nu y.\, \mu x.\, H(x,y)$
(4) (a) $\mu x.\, H(x, \nu y.\, K(y,x)) \leq \mu x.\, H(x, \nu y.\, K(y, \mu x.\, H(x,y)))$
  (b) $\nu x.\, H(x, \mu y.\, K(y,x)) \geq \nu x.\, H(x, \mu y.\, K(y, \nu x.\, H(x,y)))$

Proof.

(1) $F(\nu(F)) \overset{(2.1.1)}{=} \nu(F)$, so by fixpoint induction, $\mu(F) \leq \nu(F)$.
(2) (a) $A \leq A$, hence by fixpoint induction, $\mu x.x \leq A$. Then (b) follows by duality.

(3) Define $F(x) := \nu y.\, H(x, y)$ and $G(y) := \mu x.\, H(x, y)$. Note that both F and G are monotonic, using Lemma 2.1.5. Then:

$$
\begin{aligned}
&\mu(F) \overset{(2.1.1)}{=} F(\mu(F)) \overset{(2.1.1)}{=} H(\mu(F), F(\mu(F))) \overset{(2.1.1)}{=} H(\mu(F), \mu(F)) \\
\Rightarrow\ & \text{(by fixpoint induction)} \\
& G(\mu(F)) = \mu x.\, H(x, \mu(F)) \leq \mu(F) \\
\Rightarrow\ & \text{(monotonicity } F\text{)} \\
& F(G(\mu(F))) \leq F(\mu(F)) \overset{(2.1.1)}{=} \mu(F) \\
\Rightarrow\ & \text{(monotonicity } H\text{)} \\
& F(G(\mu(F)) \overset{(2.1.1)}{=} H(G(\mu(F)), F(G(\mu(F)))) \leq H(G(\mu(F)), \mu(F)) \overset{(2.1.1)}{=} G(\mu(F)) \\
\Rightarrow\ & \text{(by fixpoint induction)} \\
& \mu(F) \leq G(\mu(F)) \\
\Rightarrow\ & \text{(by fixpoint induction for } \nu\text{)} \\
& \mu(F) \leq \nu(G)
\end{aligned}
$$

(4) (a) Define $F(y) := \mu x.\, H(x, y)$ and $G(x) := \nu y.\, K(y, x)$. Note that both F and G are monotonic, using Lemma 2.1.5. Then:

$$
\begin{aligned}
& \mu x.\, H(x, \nu y.\, K(y, x)) \\
\overset{(2.1.6)}{=}\ & \mu x.\, \mu z.\, H(z, \nu y.\, K(y, x)) \\
=\ & \mu x.\, F(G(x)) \\
\overset{(2.1.3)}{=}\ & F(\mu x.\, G(F(x))) \\
\leq\ & \text{ (using 1, and monotonicity of } F\text{)} \\
& F(\nu x.\, G(F(x))) \\
=\ & F(\nu x.\, \nu y.\, K(y, F(x))) \\
\overset{(2.1.6)}{=}\ & F(\nu y.\, K(y, F(y))) \\
=\ & \mu x.\, H(x, \nu y.\, K(y, \mu x.\, H(x, y)))
\end{aligned}
$$

Then (b) follows by "duality" (reversing μ/ν and $\leq/\geq$). More precisely, (b) is (a) in the reversed complete lattice $(U, \geq, lub)$. □

Note that (1) and (3) are their own dual.

3. Fixpoint Equation Systems

In this section we first formally define Fixpoint Equation Systems (FES). We show by examples how they generalize Boolean and Predicate Equation Systems. Subsection 3.2 introduces the semantics of a FES by defining its solutions. Finally, Subsection 3.3 defines the variable dependency graph in a FES.

3.1. **Definition of Fixpoint Equation Systems.** Fix a complete lattice $(U, \leq, glb)$, and a set of variables $\mathcal{X}$. Throughout the paper, we assume that equality on variables is decidable. We define the set of *valuations* $\mathcal{V}al := \mathcal{X} \to U$. For $X \in \mathcal{X}$, $\eta \in \mathcal{V}al$, $P \in U$, we denote by $\eta[X := P]$ the valuation that returns P on X and $\eta(Y)$ on $Y \neq X$. As any function, valuations can be ordered pointwise, i.e. $\eta_1 \leq \eta_2$ iff $\forall X \in \mathcal{X}. \eta_1(X) \leq \eta_2(X)$. Note that valuation update is monotonic, that is, if $P \leq Q$, then $\eta[X := P] \leq \eta[X := Q]$. To indicate that two valuations agree on a set of variables $V \subseteq \mathcal{X}$, we write $\eta_1 \overset{V}{=} \eta_2$, formally defined as $\forall X \in V. \eta_1(X) = \eta_2(X)$. The complement of V in $\mathcal{X}$ is denoted $\overline{V}$.

A set of *mutually recursive equations* is a member of $\mathcal{E}qs := \mathcal{V}al \to \mathcal{V}al$. The set $\mathcal{E}qs$ is also ordered pointwise. $\mathcal{E}$ is *monotonic* iff it is a monotonic function on $\mathcal{V}al$. Note that this semantic view on equations escapes the need to introduce (and be limited) to a particular syntax.

Example 3.1. Take $\mathcal{X} = \{X, Y, Z\}$ and $U = \mathbb{B}$, the Boolean lattice $\bot < \top$. We write $(a, b, c) \in \mathbb{B}^3$ as a shorthand for the valuation $\{X{=}a, Y{=}b, Z{=}c\}$. The system of equations $\{X = Y \wedge Z,\ Y = X \vee Z,\ Z = \neg X\}$ is represented in our theory as the function

$$\mathcal{B} \ := \ \lambda(X, Y, Z) \in \mathbb{B}^3. (Y \wedge Z,\ X \vee Z,\ \neg X) \ .$$

It is not monotonic, because as valuations, $(\bot, \bot, \bot) \leq (\top, \top, \top)$, but

$$\mathcal{B}(\bot, \bot, \bot) = (\bot, \bot, \top) \not\leq (\top, \top, \bot) = \mathcal{B}(\top, \top, \top) \ .$$

Note that $\mathcal{E}qs$ is isomorphic with $\mathcal{X} \to \mathcal{V}al \to U$. This motivates the following slight abuse of notation: Given $\mathcal{E} \in \mathcal{E}qs$, we will often write $\mathcal{E}_X(\eta)$ for $\mathcal{E}(\eta)(X)$. This expression denotes the definition of X in $\mathcal{E}$, possibly depending on other variables as represented by the valuation η. Similar to valuations, agreement on variables from $V \subseteq \mathcal{X}$ is denoted $\mathcal{E}_1 \overset{V}{=} \mathcal{E}_2$, defined as $\forall \eta \in \mathcal{V}al, X \in V. \mathcal{E}_1(\eta)(X) = \mathcal{E}_2(\eta)(X)$.

The set of *specifications* consists of finite lists of signed variables: $\mathcal{S}pec := (\{\mu, \nu\} \times \mathcal{X})^*$. Note that a specification selects a subset of variables to be considered, assigns a fixpoint sign to these variables, and assigns an order to these variables. We use σX as a notation for (σ, X), write ε for the empty list, and use ; for list concatenation. We will identify a singleton list with its element. For instance, $[\mu X, \nu Y]; \mu Z$ denotes the specification $[(\mu, X), (\nu, Y), (\mu, Z)]$. We define $dom(\mathcal{S}) \subseteq \mathcal{X}$ as the set of variables that occur in some pair in $\mathcal{S}$. Decidability of $X \in dom(\mathcal{S})$ follows from finiteness of $\mathcal{S}$ and decidability of equality on variables. We define $disjoint(\mathcal{S}_1, \mathcal{S}_2)$ iff $dom(\mathcal{S}_1) \cap dom(\mathcal{S}_2) = \emptyset$.

We often require that valuations or equation systems agree on the variables in a specification. Accordingly, we overload $\doteq$ so that $\eta_1 \overset{\mathcal{S}}{=} \eta_2$ (*resp.* $\mathcal{E}_1 \overset{\mathcal{S}}{=} \mathcal{E}_2$) is defined as $\eta_1 \underset{dom(\mathcal{S})}{\overset{\mathcal{S}}{=}} \eta_2$ (*resp.* $\mathcal{E}_1 \overset{\mathcal{S}}{=} \mathcal{E}_2$). This also applies when a complement is involved: $\eta_1 \overset{\overline{\mathcal{S}}}{=} \eta_2$ is $\eta_1 \overset{\overline{dom(\mathcal{S})}}{=} \eta_2$.

Finally, a *fixpoint equation system* (FES) $\mathcal{F}$ on $(U, \mathcal{X})$ is simply a pair in $\mathcal{F}es := \mathcal{E}qs \times \mathcal{S}pec$. Before we present the semantics of FES, we first consider several instances of FES.

Example 3.2. The Boolean Equation System [Mad97] traditionally written as

$$\begin{array}{rcl} \mu X & = & Y \wedge Z \\ \nu Y & = & X \vee Z \\ \nu Z & = & \neg X \end{array}$$

is represented in our theory as the pair $(\mathcal{B}, [\mu X, \nu Y, \nu Z])$, where $\mathcal{B}$ is from Example 3.1. Note that this notation for BES integrates the set of equations and the specification into one, and these cannot be considered separately.

Example 3.3. A PBES (parameterized BES [GW05a], or predicate equation system [ZC05]) is a FES over the complete lattice $U := (\mathcal{P}(D), \subseteq)$ for some data set D, or equivalently $(D \to \mathbb{B}, \leq)$. PBES thus generalise BES: each variable is now a predicate over domain D, allowing one to create complex expressions over data. For our example, let $\mathcal{X} = \{X, Y\}$ and $D = \mathbb{N} \times \mathbb{B}$. Again using the shorthand $(X, Y) \in (D \to \mathbb{B})^2$ for valuations, $(\mathcal{B}', [\mu X, \nu Y])$ is a PBES, where

$$\begin{array}{rcl} \mathcal{B}' & := & \lambda(X,Y) \in (D \to \mathbb{B})^2. \\ & & \quad (\lambda m \in \mathbb{N}, b \in \mathbb{B}.\ (b \to m > 0 \wedge Y(m, \bot)) \wedge (\neg b \to m < 5 \wedge Y(m, \bot)), \\ & & \quad\ \lambda m \in \mathbb{N}, b \in \mathbb{B}.\ X(m-1, m > 4) \vee Y(m+1, \bot)) \end{array}$$

The function on the last line, which defines Y, does not contain an occurrence of the argument b. However, in our theory we are required to include it so that both variables in $\mathcal{X} := \{X, Y\}$ are predicates over the same $D := \mathbb{N} \times \mathbb{B}$. In the notation of [GW05a], the same PBES is simply written as a pair of predicate definitions with accompanying fixpoint signs. The argument b of Y may be left out:

$$\begin{array}{rcl} \mu X(m \colon \mathbb{N}, b \colon \mathbb{B}) & = & (b \to m > 0 \wedge Y(m)) \wedge (\neg b \to m < 5 \wedge Y(m)) \\ \nu Y(m \colon \mathbb{N}) & = & X(m-1, m > 4) \vee Y(m+1) \end{array}$$

Similar to BES, the PBES formalism as defined in [GW05a] does not consider the equations and the specification separately.

Remark 3.4. Our choice of separating the set of equations and the specification makes it easier to perform induction proofs over the specification (because one retains knowledge of all equations in the proof scope). However, we have not required that all variables in $\mathcal{S}$ are unique. This is not needed in our formalization, because in Lemma 3.7.3 we will show that if $X \in \mathit{dom}(\mathcal{S})$, then any set of equations $\mathcal{E}$ has the same semantics when combined with S or with $\sigma X; S$, for any $\sigma \in \{\mu, \nu\}$. Do note that there is a hidden assumption: even if X occurs multiple times in $\mathcal{S}$, possibly with different signs, there can only be one defining equation for it, because $\mathcal{E}$ is a function. So when transferring the results to traditional notation, one should add the (quite natural) requirement that all equations have unique variable names.

Example 3.5. *Modal equation systems* (MES) [CS93] is a very similar FES instance to PBES since it also uses the powerset lattice $(\mathcal{P}(D), \subseteq)$. However, a MES is interpreted on a *labelled transition system* (LTS), so that D is equal to the set of states in the LTS. MES includes, for every action a in the LTS, the modal operators $[a]\varphi$ ("φ must hold after every possible transition labelled with a") and $\langle a \rangle \varphi$ ("there exists an a-transition after which φ holds"). MES is an equational representation of the modal mu-calculus. Adopting the notation from the previous examples, an example of a MES is:

$$\begin{array}{rcl} \nu X & = & Y \\ \mu Y & = & [a]X \wedge [b]Y \end{array}$$

This MES expresses that on every path in the LTS, action a occurs infinitely often (and b may happen only finitely often in between).

3.2. **Semantics of FES and Basic Results.** Next, the semantics of a FES, $[\![\mathcal{E},\mathcal{S}]\!] : \mathcal{V}al \to \mathcal{V}al$, is defined recursively on $\mathcal{S}$:

$$\begin{cases} [\![\mathcal{E},\varepsilon]\!](\eta) & := & \eta \\ [\![\mathcal{E},\sigma X;\mathcal{S}]\!](\eta) & := & [\![\mathcal{E},\mathcal{S}]\!](\eta[X := \sigma(F)]), \\ & & \text{where } F\colon U \to U \text{ is defined as} \\ & & F(P) := \mathcal{E}_X([\![\mathcal{E},\mathcal{S}]\!](\eta[X := P])) \end{cases}$$

We now state the first results on the semantics of FES. The lemma below states monotonicity properties for the semantics. They ensure that fixpoints are well-behaved. Below, recall that we use pointwise ordering, *e.g.*, $\mathcal{E}_1 \leq \mathcal{E}_2$ iff $\mathcal{E}_1(\eta)(X) \leq \mathcal{E}_2(\eta)(X)$ for all η and X.

Lemma 3.6. *Let $\mathcal{E}$, $\mathcal{E}_1$, $\mathcal{E}_2 \in \mathcal{E}qs$ and $\mathcal{S} \in \mathcal{S}pec$, then*

(1) *If $\mathcal{E}$ is monotonic, then $[\![\mathcal{E},\mathcal{S}]\!]$ is monotonic.*
(2) *If $\mathcal{E}_1$ is monotonic and $\mathcal{E}_1 \leq \mathcal{E}_2$, then $[\![\mathcal{E}_1,\mathcal{S}]\!] \leq [\![\mathcal{E}_2,\mathcal{S}]\!]$.*

Proof.

(1) We prove $\forall \eta_1, \eta_2.\ \eta_1 \leq \eta_2 \Rightarrow [\![\mathcal{E},\mathcal{S}]\!](\eta_1) \leq [\![\mathcal{E},\mathcal{S}]\!](\eta_2)$ by induction on $\mathcal{S}$. The base case: assume $\eta_1 \leq \eta_2$, then $[\![\mathcal{E},\varepsilon]\!](\eta_1) = \eta_1 \leq \eta_2 = [\![\mathcal{E},\varepsilon]\!](\eta_2)$. Induction step: Let $\eta_1 \leq \eta_2$ be given, then for any $P \in U$, we have $\eta_1[X := P] \leq \eta_2[X := P]$. So

$$\begin{aligned} & \text{(induction hypothesis)} \\ & [\![\mathcal{E},\mathcal{S}]\!](\eta_1[X := P]) \leq [\![\mathcal{E},\mathcal{S}]\!](\eta_2[X := P]) \\ \Rightarrow\ & (\mathcal{E} \text{ is monotonic}) \\ & \mathcal{E}_X([\![\mathcal{E},\mathcal{S}]\!](\eta_1[X := P])) \leq \mathcal{E}_X([\![\mathcal{E},\mathcal{S}]\!](\eta_2[X := P])) \\ \Rightarrow\ & \text{(fixpoint monotonicity, Lemma 2.1.5)} \\ & \sigma P.\, \mathcal{E}_X([\![\mathcal{E},\mathcal{S}]\!](\eta_1[X := P])) \leq \sigma P.\, \mathcal{E}_X([\![\mathcal{E},\mathcal{S}]\!](\eta_2[X := P])) \\ \Rightarrow\ & (\text{define } F_i(P) := \mathcal{E}_X([\![\mathcal{E},\mathcal{S}]\!](\eta_i[X := P])), \text{ for } i = 1,2) \\ & \eta_1[X := \sigma(F_1)] \leq \eta_2[X := \sigma(F_2)] \\ \Rightarrow\ & \text{(induction hypothesis)} \\ & [\![\mathcal{E},\mathcal{S}]\!](\eta_1[X := \sigma(F_1)]) \leq [\![\mathcal{E},\mathcal{S}]\!](\eta_2[X := \sigma(F_2)]) \\ \Leftrightarrow\ & \text{(definition)} \\ & [\![\mathcal{E},\sigma X;\mathcal{S}]\!](\eta_1) \leq [\![\mathcal{E},\sigma X;\mathcal{S}]\!](\eta_2) \end{aligned}$$

(2) Assume $\mathcal{E}_1$ is monotonic, and $\mathcal{E}_1 \leq \mathcal{E}_2$ (pointwise). We prove by induction on $\mathcal{S}$ that $\forall \eta.\ [\![\mathcal{E}_1,\mathcal{S}]\!](\eta) \leq [\![\mathcal{E}_2,\mathcal{S}]\!](\eta)$. The base case is simple: for all η, we have $[\![\mathcal{E}_1,\varepsilon]\!](\eta) = \eta = [\![\mathcal{E}_2,\varepsilon]\!](\eta)$. For the induction step, let η be given, then for any $P \in U$:

$$\begin{aligned} & \text{(induction hypothesis)} \\ & [\![\mathcal{E}_1,\mathcal{S}]\!](\eta[X := P]) \leq [\![\mathcal{E}_2,\mathcal{S}]\!](\eta[X := P]) \\ \Rightarrow\ & (\text{monotonicity of } \mathcal{E}_1 \text{ and } \mathcal{E}_1 \leq \mathcal{E}_2, \text{ pointwise})) \\ & \mathcal{E}_{1,X}([\![\mathcal{E}_1,\mathcal{S}]\!](\eta[X := P])) \leq \mathcal{E}_{2,X}([\![\mathcal{E}_2,\mathcal{S}]\!](\eta[X := P])) \\ \Rightarrow\ & \text{(fixpoint monotonicity, Lemma 2.1.5)} \\ & \sigma P.\, \mathcal{E}_{1,X}([\![\mathcal{E}_1,\mathcal{S}]\!](\eta[X := P])) \leq \sigma P.\, \mathcal{E}_{2,X}([\![\mathcal{E}_2,\mathcal{S}]\!](\eta[X := P])) \\ \Rightarrow\ & (\text{define } F_i(P) := \mathcal{E}_{i,X}([\![\mathcal{E}_i,\mathcal{S}]\!](\eta[X := P])), \text{ for } i = 1,2) \end{aligned}$$

$$\begin{array}{ll} & \eta[X := \sigma(F_1)] \le \eta[X := \sigma(F_2)] \\ \Rightarrow & \quad [\![\mathcal{E}_1, \mathcal{S}]\!](\eta[X := \sigma(F_1)]) \\ & \le \quad \text{(monotonicity, part 1 of this lemma)} \\ & \quad [\![\mathcal{E}_1, \mathcal{S}]\!](\eta[X := \sigma(F_2)]) \\ & \le \quad \text{(induction hypothesis)} \\ & \quad [\![\mathcal{E}_2, \mathcal{S}]\!](\eta[X := \sigma(F_2)]) \\ \Rightarrow & \text{(definition)} \\ & [\![\mathcal{E}_1, \sigma X;\mathcal{S}]\!](\eta) \le [\![\mathcal{E}_2, \sigma X;\mathcal{S}]\!](\eta) \end{array}$$ □

The next lemma states three sanity properties of the semantics: (1) The semantics only modifies the valuation on elements in the domain; (2) the semantics only depends on equations mentioned in the domain; (3) the input valuation is only used for variables outside the domain.

Lemma 3.7. *Let $\mathcal{E}, \mathcal{E}_1, \mathcal{E}_2 \in \mathcal{E}qs$, $\mathcal{S} \in \mathcal{S}pec$, $\eta \in \mathcal{V}al$ and $X \in \mathcal{X}$.*

(1) *If $X \notin dom(\mathcal{S})$ then $[\![\mathcal{E}, \mathcal{S}]\!](\eta)(X) = \eta(X)$.*

(2) *If $\mathcal{E}_1 \overset{\mathcal{S}}{\equiv} \mathcal{E}_2$, then $[\![\mathcal{E}_1, \mathcal{S}]\!] = [\![\mathcal{E}_2, \mathcal{S}]\!]$.*

(3) *If $\eta_1 \overset{\overline{\mathcal{S}}}{\equiv} \eta_2$, then $[\![\mathcal{E}, \mathcal{S}]\!](\eta_1) = [\![\mathcal{E}, \mathcal{S}]\!](\eta_2)$.*

Proof.

(1) Induction on $\mathcal{S}$. The base case holds by definition. For the induction step, assume $X \notin dom(\sigma Y;S)$. Then

$$\begin{array}{ll} & [\![\mathcal{E}, \sigma Y;\mathcal{S}]\!](\eta)(X) \\ = & \text{(define } F(P) := \mathcal{E}_Y([\![\mathcal{E}, \mathcal{S}]\!](\eta[Y := P]))) \\ & [\![\mathcal{E}, \mathcal{S}]\!](\eta[Y := \sigma(F)])(X) \\ = & \text{(induction hypothesis, note: } X \notin dom(\mathcal{S})) \\ & \eta[Y := \sigma(F)](X) \\ = & (X \notin dom(\sigma Y;S), \text{ so } X \neq Y) \\ & \eta(X) \end{array}$$

(2) Induction on $\mathcal{S}$. The base case holds by definition. For the induction step, let η be given, and assume $\mathcal{E}_1 \overset{\sigma X;\mathcal{S}}{\equiv} \mathcal{E}_2$. Note that this implies $\mathcal{E}_1 \overset{\mathcal{S}}{\equiv} \mathcal{E}_2$, so we can use the induction hypothesis. For any $P \in U$ we have:

$$\begin{array}{ll} & \text{(induction hypothesis)} \\ & [\![\mathcal{E}_1, \mathcal{S}]\!](\eta[X := P]) = [\![\mathcal{E}_2, \mathcal{S}]\!](\eta[X := P]) \\ \Rightarrow & (X \in dom(\sigma X;\mathcal{S}), \text{ so } \mathcal{E}_{1,X} = \mathcal{E}_{2,X} \text{ by assumption}) \\ & \mathcal{E}_{1,X}([\![\mathcal{E}_1, \mathcal{S}]\!](\eta[X := P])) = \mathcal{E}_{2,X}([\![\mathcal{E}_2, \mathcal{S}]\!](\eta[X := P])) \\ \Rightarrow & \text{(define } F_i(P) := \mathcal{E}_{i,X}([\![\mathcal{E}_i, \mathcal{S}]\!](\eta[X := P])) \text{ for } i = 1, 2) \\ & \eta[X := \sigma(F_1)] = \eta[X := \sigma(F_2)] \\ \Rightarrow & \text{(induction hypothesis)} \\ & [\![\mathcal{E}_1, \mathcal{S}]\!](\eta[X := \sigma(F_1)]) = [\![\mathcal{E}_2, \mathcal{S}]\!](\eta[X := \sigma(F_2)]) \\ \Leftrightarrow & \text{(definition)} \end{array}$$

$$[\![\mathcal{E}_1, \sigma X;\mathcal{S}]\!](\eta) = [\![\mathcal{E}_2, \sigma X;\mathcal{S}]\!](\eta)$$

(3) Induction on $\mathcal{S}$. The base case is trivial, because both the assumption and the conclusion reduce to $\eta_1 = \eta_2$. The induction step is proved as follows (for arbitrary $P \in U$):

$$\begin{array}{ll} & \text{(assumption)} \\ & \eta_1 \overset{\overline{\sigma X;\mathcal{S}}}{=} \eta_2 \\ \Rightarrow & \text{(extensionality)} \\ & \eta_1[X := P] \overset{\overline{\mathcal{S}}}{=} \eta_2[X := P] \\ \Rightarrow & \text{(induction hypothesis)} \\ & [\![\mathcal{E}, \mathcal{S}]\!](\eta_1[X := P]) = [\![\mathcal{E}, \mathcal{S}]\!](\eta_2[X := P]) \\ \Rightarrow & \text{(define } F_i(P) := \mathcal{E}_X([\![\mathcal{E}, \mathcal{S}]\!](\eta_i[X := P])), \text{ for } i = 1, 2) \\ & \sigma(F_1) = \sigma(F_2) \\ \Rightarrow & \text{(extensionality)} \\ & \eta_1[X := \sigma(F_1)] \overset{\overline{\mathcal{S}}}{=} \eta_2[X := \sigma(F_2)] \\ \Rightarrow & \text{(induction hypothesis)} \\ & [\![\mathcal{E}, \mathcal{S}]\!](\eta_1[X := \sigma(F_1)]) = [\![\mathcal{E}, \mathcal{S}]\!](\eta_2[X := \sigma(F_2)]) \\ \Rightarrow & \text{(definition)} \\ & [\![\mathcal{E}, \sigma X;\mathcal{S}]\!](\eta_1) = [\![\mathcal{E}, \sigma X;\mathcal{S}]\!](\eta_2) \end{array}$$

□

Next, we can prove that the semantics as defined above indeed solves the equations for those variables occurring in the specification:

Lemma 3.8. *Let $\mathcal{E} \in \mathcal{E}qs$ be monotonic, $\mathcal{S} \in \mathcal{S}pec$, $\eta \in \mathcal{V}al$ and $X \in \mathcal{X}$. If $X \in dom(\mathcal{S})$, then $\mathcal{E}_X([\![\mathcal{E}, \mathcal{S}]\!](\eta)) = [\![\mathcal{E}, \mathcal{S}]\!](\eta)(X)$.*

Proof. Let $\mathcal{E}$ be monotonic. By induction on $\mathcal{S}$, we will prove that for all $X \in dom(\mathcal{S})$ and for all η, it holds that $\mathcal{E}_X([\![\mathcal{E}, \mathcal{S}]\!](\eta)) = [\![\mathcal{E}, \mathcal{S}]\!](\eta)(X)$. The base case trivially holds, because $X \notin dom(\varepsilon)$. For the induction step, assume $X \in dom(\sigma Y;\mathcal{S})$. We distinguish cases.

If $X \in dom(\mathcal{S})$:

$$\begin{array}{ll} & [\![\mathcal{E}, \sigma Y;\mathcal{S}]\!](\eta)(X) \\ = & \text{(define } F(P) := \mathcal{E}_Y([\![\mathcal{E}, \mathcal{S}]\!](\eta[Y := P]))\ ) \\ & [\![\mathcal{E}, \mathcal{S}]\!](\eta[Y := \sigma(F)])(X) \\ = & \text{(induction hypothesis; } X \in dom(\mathcal{S})) \\ & \mathcal{E}_X([\![\mathcal{E}, \mathcal{S}]\!](\eta[Y := \sigma(F)])) \\ = & \text{(definition)} \\ & \mathcal{E}_X([\![\mathcal{E}, \sigma Y;\mathcal{S}]\!](\eta)) \end{array}$$

Otherwise, if $X \notin dom(\mathcal{S})$ then $X = Y$. We compute:

$$\begin{array}{ll} & [\![\mathcal{E}, \sigma X;\mathcal{S}]\!](\eta)(X) \\ = & \text{(define } F(P) := \mathcal{E}_X([\![\mathcal{E}, \mathcal{S}]\!](\eta[X := P]))\ ) \\ & [\![\mathcal{E}, \mathcal{S}]\!](\eta[X := \sigma(F)])(X) \end{array}$$

$$\begin{aligned}
&= \text{(Lemma 3.7.1; } X \notin \mathit{dom}(\mathcal{S})\text{)}\\
&\quad \sigma(F)\\
&= \text{(computation rule, Lemma 2.1.1; } F \text{ is monotonic because } \mathcal{E}_X\\
&\qquad \text{is monotonic by assumption, and } [\![\mathcal{E},\mathcal{S}]\!] \text{ is by Lemma 3.6.1)}\\
&\quad F(\sigma(F))\\
&= \mathcal{E}_X([\![\mathcal{E},\mathcal{S}]\!](\eta[X := \sigma(F)]))\\
&= \text{(definition)}\\
&\quad \mathcal{E}_X([\![\mathcal{E},\sigma X;\mathcal{S}]\!](\eta)) \qquad \square
\end{aligned}$$

We have the following left-congruence result:

Lemma 3.9. *For* $\mathcal{E}_1, \mathcal{E}_2 \in \mathcal{E}qs$*,* $\mathcal{S}, \mathcal{S}_1, \mathcal{S}_2 \in \mathcal{S}pec$*, if* $\mathcal{E}_1 \overset{\mathcal{S}}{\equiv} \mathcal{E}_2$ *and* $[\![\mathcal{E}_1, \mathcal{S}_1]\!] = [\![\mathcal{E}_2, \mathcal{S}_2]\!]$*, then* $[\![\mathcal{E}_1, \mathcal{S};\mathcal{S}_1]\!] = [\![\mathcal{E}_2, \mathcal{S};\mathcal{S}_2]\!]$.

Proof. Induction on $\mathcal{S}$. The base case is trivial. For the induction step, assume $[\![\mathcal{E}_1, \mathcal{S}_1]\!] = [\![\mathcal{E}_2, \mathcal{S}_2]\!]$ and $\mathcal{E}_1 \overset{\sigma X;\mathcal{S}}{\equiv} \mathcal{E}_2$. Then also $\mathcal{E}_1 \overset{\mathcal{S}}{\equiv} \mathcal{E}_2$. So, for any $P \in U$:

$$\begin{aligned}
&\quad \text{(induction hypothesis)}\\
&\quad [\![\mathcal{E}_1,\mathcal{S};\mathcal{S}_1]\!](\eta[X := P]) = [\![\mathcal{E}_2,\mathcal{S};\mathcal{S}_2]\!](\eta[X := P])\\
&\Rightarrow \text{(define } F_i(P) := \mathcal{E}_{i,X}([\![\mathcal{E}_i,\mathcal{S};\mathcal{S}_i]\!](\eta[X := P]))\text{; note } \mathcal{E}_{1,X} = \mathcal{E}_{2,X}\text{)}\\
&\quad \eta[X := \sigma(F_1)] = \eta[X := \sigma(F_2)]\\
&\Rightarrow \text{(induction hypothesis)}\\
&\quad [\![\mathcal{E}_1,\mathcal{S};\mathcal{S}_1]\!](\eta[X := \sigma(F_1)]) = [\![\mathcal{E}_2,\mathcal{S};\mathcal{S}_2]\!](\eta[X := \sigma(F_2)])\\
&\Rightarrow \text{(definition)}\\
&\quad [\![\mathcal{E}_1,\sigma X;\mathcal{S};\mathcal{S}_1]\!](\eta) = [\![\mathcal{E}_2,\sigma X;\mathcal{S};\mathcal{S}_2]\!](\eta) \qquad \square
\end{aligned}$$

Remarkably, right-congruence doesn't hold in general. Corollary 5.5 will state a sufficient condition for right-congruence.

3.3. The Dependency Graph between Variables.

Since we introduced a semantic notion of equations, avoiding syntactic expressions, we also need a semantic notion of dependence between variables. Given $V_1, V_2 \subseteq \mathcal{X}$, we define that V_1 is *independent* of V_2 with respect to $\mathcal{E}$, notation $\mathit{indep}(\mathcal{E}, V_1, V_2)$, as follows:

$$\forall \eta_1, \eta_2.\, (\eta_1 \overset{\overline{V_2}}{\equiv} \eta_2) \Rightarrow (\mathcal{E}(\eta_1) \overset{V_1}{\equiv} \mathcal{E}(\eta_2)) \enspace .$$

That is: the solution of variables $X \in V_1$ is the same for all those η that differ at most on the values assigned to $Y \in V_2$. This is slightly more liberal than the usual syntactic requirement that Y *doesn't occur syntactically* in $\mathcal{E}_X$. We overload the definition of *indep* for individual variables and specifications (and any combination of those), *e.g.*, $\mathit{indep}(\mathcal{E}, \mathcal{S}_1, \mathcal{S}_2) = \mathit{indep}(\mathcal{E}, \mathit{dom}(\mathcal{S}_1), \mathit{dom}(\mathcal{S}_2))$ and $\mathit{indep}(\mathcal{E}, X, Y) = \mathit{indep}(\mathcal{E}, \{X\}, \{Y\})$.

This notion gives rise to the *variable dependency graph* of a FES $(\mathcal{E}, \mathcal{S})$. The variables in $\mathit{dom}(\mathcal{S})$ form the nodes of this graph; the edges $X \xrightarrow{\mathcal{E},\mathcal{S}} Y$ are defined as $\neg\mathit{indep}(\mathcal{E}, X, Y)$. We define X *depends (indirectly)* on Y (written $X \overset{\mathcal{E},\mathcal{S}}{\twoheadrightarrow} Y$) as the reflexive, transitive closure of $\xrightarrow{\mathcal{E},\mathcal{S}}$. In other words, there exists a path in the dependency graph from X to Y. We also use the notation $X \xrightarrow{\mathcal{E},\mathcal{S}}{}^{+} Y$ to denote the transitive closure of $\xrightarrow{\mathcal{E},\mathcal{S}}$, i.e. there is a *non-empty* path from X to Y in the dependency graph. We assume that $\mathit{indep}(\mathcal{E}, X, Y)$ is decidable for all $\mathcal{E}$, X and Y. Since $\mathit{dom}(\mathcal{S})$ is finite, this also makes $\xrightarrow{\mathcal{E},\mathcal{S}}\!\!\!\!\rightarrow$ and $\xrightarrow{\mathcal{E},\mathcal{S}}{}^{+}$ decidable.

4. Substituting in FES Equations

In this section, we define two substitution operations on the equations of a FES, and study under which conditions these operations preserve solutions. The first operation allows substituting variables by their definition. We show that this substitution preserves solutions in some new cases (cf. Section 7). The second operation replaces a variable in an equation by its solution.

4.1. Unfolding Definitions.

We define $\mathit{unfold}(\mathcal{E}, X, Y)$, where each occurrence of Y in the definition of X is replaced by the definition of Y, as follows:

$$\mathit{unfold}(\mathcal{E}, X, Y)(\eta) := \mathcal{E}(\eta)[X := \mathcal{E}_X(\eta[Y := \mathcal{E}_Y(\eta)])]$$

We will use the following observation several times. It basically states that unfolding Y in X doesn't affect other equations than that for X.

Lemma 4.1. *If $X \notin \mathit{dom}(\mathcal{S})$, then*
(1) $\mathit{unfold}(\mathcal{E}, X, Y) \overset{\mathcal{S}}{\equiv} \mathcal{E}$
(2) $[\![\mathit{unfold}(\mathcal{E}, X, Y), \mathcal{S}]\!] = [\![\mathcal{E}, \mathcal{S}]\!]$

Proof. (1) holds by definition of *unfold*, as it only modifies the value of $\mathcal{E}$ on variable X. Then (2) holds by Lemma 3.7.2. □

It is known (cf. Example 7.1) that in general one should not unfold Y in X, if Y precedes X in the specification. As a new result we show that we can substitute X in its own definition:

Lemma 4.2. *Let $\mathcal{E} \in \mathcal{E}qs$ be monotonic. Let $\mathcal{S} = \sigma X;\mathcal{S}_1$, and $X \notin \mathit{dom}(\mathcal{S}_1)$. Then $[\![\mathit{unfold}(\mathcal{E}, X, X), \mathcal{S}]\!] = [\![\mathcal{E}, \mathcal{S}]\!]$.*

Proof. For arbitrary η, define

$$F(P) := \mathcal{E}_X\Big([\![\mathcal{E}, \mathcal{S}_1]\!](\eta[X := P])\Big)$$
$$G(P) := \mathit{unfold}(\mathcal{E}, X, X)_X\Big([\![\mathcal{E}, \mathcal{S}_1]\!](\eta[X := P])\Big)$$
$$H(P, Q) := \mathcal{E}_X\Big([\![\mathcal{E}, \mathcal{S}_1]\!](\eta[X := P])[X := Q]\Big)$$

By Lemma 3.7.1 and $X \notin \mathit{dom}(\mathcal{S}_1)$, we obtain: $[\![\mathcal{E}, \mathcal{S}_1]\!](\eta[X := P])(X) = \eta[X := P](X) = P$, so

$$[\![\mathcal{E}, \mathcal{S}_1]\!](\eta[X := P])[X := P] \;=\; [\![\mathcal{E}, \mathcal{S}_1]\!](\eta[X := P]) \qquad (*)$$

Next, we prove that $\sigma(F) = \sigma(G)$:

$$\begin{aligned}
&\sigma P.\, G(P)\\
=\ &\text{(by definition of } \mathit{unfold}\text{)}\\
&\sigma P.\, \mathcal{E}_X\Big([\![\mathcal{E}, \mathcal{S}_1]\!](\eta[X := P])\Big[X := \mathcal{E}_X\big([\![\mathcal{E}, \mathcal{S}_1]\!](\eta[X := P])\big)\Big]\Big)\\
=\ &\text{(by * above)}
\end{aligned}$$

$$\begin{aligned}
&\sigma P.\, \mathcal{E}_X\Big([\![\mathcal{E},\mathcal{S}_1]\!](\eta[X := P])\Big[X := \mathcal{E}_X\big([\![\mathcal{E},\mathcal{S}_1]\!](\eta[X := P])[X := P]\big)\Big]\Big)\\
=\ &\sigma P.\, H(P, H(P,P))\\
=\ &\text{(unfold rule, Lemma 2.1.7)}\\
&\sigma P.\, H(P,P)\\
=\ &\text{(by * above)}\\
&\sigma P.\, F(P)
\end{aligned}$$

We can now finish the proof:

$$\begin{aligned}
&[\![\mathit{unfold}(\mathcal{E},X,X), \sigma X;\mathcal{S}_1]\!](\eta)\\
=\ &\text{(by definition and Lemma 4.1.2)}\\
&[\![\mathcal{E},\mathcal{S}_1]\!](\eta[X := \sigma(G)])\\
=\ &\text{(by the computation before)}\\
&[\![\mathcal{E},\mathcal{S}_1]\!](\eta[X := \sigma(F)])\\
=\ &\text{(by definition)}\\
&[\![\mathcal{E},\sigma X;\mathcal{S}_1]\!](\eta)
\end{aligned}$$

□

The full theorem allows to unfold Y in the equations for those X that precede that of Y, and in the equation of Y itself. So in particular, the case $X = Y$ is allowed.

Theorem 4.3. *Let $\mathcal{E} \in \mathcal{E}qs$ be monotonic. Let $\mathcal{S} = \mathcal{S}_1;\sigma Y;\mathcal{S}_2$ and $X \notin \mathit{dom}(\mathcal{S}_2)$. Then $[\![\mathit{unfold}(\mathcal{E},X,Y),\mathcal{S}]\!] = [\![\mathcal{E},\mathcal{S}]\!]$.*

Proof. The proof is by induction on $\mathcal{S}_1$. The base case is $\mathcal{S}_1 = \varepsilon$. If $X = Y$, we have to prove $[\![\mathit{unfold}(\mathcal{E},X,X),\sigma X;\mathcal{S}_2]\!] = [\![\mathcal{E},\sigma X;\mathcal{S}_2]\!]$, which is just Lemma 4.2. Otherwise, if $X \neq Y$, then $X \notin \mathit{dom}(\sigma Y;\mathcal{S}_2)$, so by Lemma 4.1 $[\![\mathit{unfold}(\mathcal{E},X,Y),\sigma Y;\mathcal{S}_2]\!] = [\![\mathcal{E},\sigma Y;\mathcal{S}_2]\!]$.

Next, for $\mathcal{S} = \rho Z;\mathcal{S}_1;\sigma Y;\mathcal{S}_2$, define $\mathcal{S}_3 := \mathcal{S}_1;\sigma Y;\mathcal{S}_2$, and assume the induction hypothesis, $[\![\mathit{unfold}(\mathcal{E},X,Y),\mathcal{S}_3]\!] = [\![\mathcal{E},\mathcal{S}_3]\!]$.

If $Z \neq X$, then $\mathcal{E} \overset{\{Z\}}{=} \mathit{unfold}(\mathcal{E},X,Y)$. From the induction hypothesis, it follows by congruence (Lemma 3.9) that $[\![\mathit{unfold}(\mathcal{E},X,Y),\rho Z;\mathcal{S}_3]\!] = [\![\mathcal{E},\rho Z;\mathcal{S}_3]\!]$.

If $Z = X$, we compute for arbitrary $\eta \in \mathcal{V}al$:

$$\begin{aligned}
&[\![\mathit{unfold}(\mathcal{E},X,Y),\sigma X;\mathcal{S}_3]\!](\eta)\\
=\ &\text{(by definition of the semantics)}\\
&[\![\mathit{unfold}(\mathcal{E},X,Y),\mathcal{S}_3]\!](\eta[X := \sigma(F)]),\ \text{where}\\
&\ F(P) := \mathit{unfold}(\mathcal{E},X,Y)_X([\![\mathit{unfold}(\mathcal{E},X,Y),\mathcal{S}_3]\!](\eta[X := P]))\\
=\ &\text{(by induction hypothesis)}\\
&[\![\mathcal{E},\mathcal{S}_3]\!](\eta[X := \sigma(F)]),\ \text{where}
\end{aligned}$$

$$\begin{array}{rl} F(P) := & unfold(\mathcal{E},X,Y)_X(\llbracket\mathcal{E},\mathcal{S}_3\rrbracket(\eta[X:=P])) \\ = & \text{(definition } unfold\text{)} \\ & \mathcal{E}_X\Big(\llbracket\mathcal{E},\mathcal{S}_3\rrbracket(\eta[X:=P])\Big[Y:=\mathcal{E}_Y(\llbracket\mathcal{E},\mathcal{S}_3\rrbracket(\eta[X:=P]))\Big]\Big) \\ = & \text{(Lemma 3.8; } Y\in dom(\mathcal{S}_3)\text{)} \\ & \mathcal{E}_X\Big(\llbracket\mathcal{E},\mathcal{S}_3\rrbracket(\eta[X:=P])\Big[Y:=\llbracket\mathcal{E},\mathcal{S}_3\rrbracket(\eta[X:=P])(Y)\Big]\Big) \\ = & \text{(congruence and extensionality: } \zeta[Y:=\zeta(Y)]=\zeta\text{)} \\ & \mathcal{E}_X(\llbracket\mathcal{E},\mathcal{S}_3\rrbracket(\eta[X:=P])) \end{array}$$

$$= \llbracket\mathcal{E},\sigma X;\mathcal{S}_3\rrbracket(\eta) \qquad \square$$

4.2. **Substituting a Partial Solution.** The following theorem is motivated in [Mad97] as follows. Assume we know by some means the solution a for a variable X in $(\mathcal{E},\mathcal{S})$. Then we can replace the definition of X by simply putting $X = a$. We simplify the proof in [Mad97], which is based on an infinite series of FESs. Instead, we just use induction on $\mathcal{S}$ and some properties of complete lattices.

Theorem 4.4. *Let $\mathcal{E}\in\mathcal{E}qs$ be monotonic and let $a := \llbracket\mathcal{E},\mathcal{S}\rrbracket(\eta)(X)$. Then*

$$\llbracket\mathcal{E},\mathcal{S}\rrbracket(\eta) = \llbracket\mathcal{E}[X\mapsto a],\mathcal{S}\rrbracket(\eta)$$

Proof. We prove the theorem by induction on $\mathcal{S}$. The base case is trivial, for $\llbracket\mathcal{E},\varepsilon\rrbracket(\eta) = \eta = \llbracket\mathcal{E}[X\mapsto a],\varepsilon\rrbracket(\eta)$. For the induction step $(\sigma Y;\mathcal{S})$, we need to define the functions $a,b\colon \mathcal{V}al\to U$ and $F,G\colon U\to U$ and $H\colon (U\times U)\to U$:

$$\begin{array}{rcl} a(\eta') & := & \llbracket\mathcal{E},\mathcal{S}\rrbracket(\eta')(X) \\ b(\eta) & := & \llbracket\mathcal{E},\sigma Y;\mathcal{S}\rrbracket(\eta)(X) \\ F(P) & := & \mathcal{E}_Y(\llbracket\mathcal{E},\mathcal{S}\rrbracket(\eta[Y:=P])) \\ G(P) & := & \mathcal{E}_Y(\llbracket\mathcal{E}[X\mapsto b(\eta)],\mathcal{S}\rrbracket) \\ H(P,Q) & := & \mathcal{E}_Y(\llbracket\mathcal{E}[X\mapsto a(\eta[Y:=P])],\mathcal{S}\rrbracket(\eta[Y:=Q])) \end{array}$$

Then from the induction hypothesis $\forall\eta'.\ \llbracket\mathcal{E},\mathcal{S}\rrbracket(\eta') = \llbracket\mathcal{E}[X\mapsto a(\eta')],\mathcal{S}\rrbracket(\eta')$, we must prove: $\forall\eta.\ \llbracket\mathcal{E},\sigma Y;\mathcal{S}\rrbracket(\eta) = \llbracket\mathcal{E}[X\mapsto b(\eta)],\sigma Y;\mathcal{S}\rrbracket(\eta)$.

We distinguish three cases:
If $X = Y$ and $X\notin dom(\mathcal{S})$, we compute:

$$\begin{array}{rl} & \llbracket\mathcal{E}[X\mapsto b(\eta)],\sigma X;\mathcal{S}\rrbracket(\eta) \\ = & \llbracket\mathcal{E}[X\mapsto b(\eta)],\mathcal{S}\rrbracket(\eta[X:=\sigma P.\,b(\eta)]) \\ = & \text{(by the constant rule, Lemma 2.1.2)} \\ & \llbracket\mathcal{E}[X\mapsto b(\eta)],\mathcal{S}\rrbracket(\eta[X:=b(\eta)]) \\ = & \text{(by Lemma 3.7.2 and } X\notin dom(\mathcal{S})\text{)} \\ & \llbracket\mathcal{E},\mathcal{S}\rrbracket(\eta[X:=b(\eta)]) \\ = & \text{(unfold definition of } \llbracket\ \rrbracket \text{ in } b(\eta)\text{.)} \\ & \llbracket\mathcal{E},\mathcal{S}\rrbracket(\eta[X:=\llbracket\mathcal{E},\mathcal{S}\rrbracket(\eta[X:=\sigma(F)])(X)]) \\ = & \text{(by Lemma 3.7.1 and } X\notin dom(\mathcal{S})\text{)} \\ & \llbracket\mathcal{E},\mathcal{S}\rrbracket(\eta[X:=\sigma(F)]) \\ = & \llbracket\mathcal{E},\sigma X;\mathcal{S}\rrbracket(\eta) \end{array}$$

If $X = Y$ and $X \in dom(\mathcal{S})$, then note that using Lemma 3.7.3, for any $\mathcal{E}'$ and appropriate F', we have:

$$[\![\mathcal{E}', \sigma X;\mathcal{S}]\!](\eta) = [\![\mathcal{E}', \mathcal{S}]\!](\eta[X := \sigma F']) = [\![\mathcal{E}', \mathcal{S}]\!](\eta)$$

So in particular, $b(\eta) = a(\eta)$, and we can apply the induction hypothesis: Hence

$$\begin{aligned}
& [\![\mathcal{E}[X \mapsto b(\eta)], \sigma X;\mathcal{S}]\!](\eta) \\
= & \text{ (by the equality above)} \\
& [\![\mathcal{E}[X \mapsto b(\eta)], \mathcal{S}]\!](\eta) \\
= & [\![\mathcal{E}[X \mapsto a(\eta)], \mathcal{S}]\!](\eta) \\
= & \text{ (induction hypothesis)} \\
& [\![\mathcal{E}, \mathcal{S}]\!](\eta) \\
= & \text{ (by the equality above)} \\
& [\![\mathcal{E}, \sigma X;\mathcal{S}]\!](\eta)
\end{aligned}$$

Finally, if $X \neq Y$, then we can compute:

$$\begin{aligned}
& [\![\mathcal{E}, \sigma Y;\mathcal{S}]\!](\eta) \\
= & [\![\mathcal{E}, \mathcal{S}]\!](\eta[Y := \sigma(F)]) \\
= & \text{ (induction hypothesis)} \\
& [\![\mathcal{E}[X \mapsto a(\eta[Y := \sigma(F)])], \mathcal{S}]\!](\eta[Y := \sigma(F)]) \\
= & \text{ (unfold definition of } [\![\ ]\!] \text{ in } b(\eta).) \\
& [\![\mathcal{E}[X \mapsto b(\eta)], \mathcal{S}]\!](\eta[Y := \sigma(F)]) \\
= & \text{ (will be proved below)} \\
& [\![\mathcal{E}[X \mapsto b(\eta)], \mathcal{S}]\!](\eta[Y := \sigma(G)]) \\
= & [\![\mathcal{E}[X \mapsto b(\eta)], \sigma Y;\mathcal{S}]\!](\eta)
\end{aligned}$$

We must still prove that $\sigma(F) = \sigma(G)$.

$$\begin{aligned}
& \sigma P.\, F(P) \\
= & \text{ (induction hypothesis)} \\
& \sigma P.\, \mathcal{E}_Y([\![\mathcal{E}[X \mapsto a(\eta[Y := P])], \mathcal{S}]\!](\eta[Y := P])) \\
= & \sigma P.\, H(P, P) \\
= & \text{ Lemma 2.1.8 (solve rule) on } \lambda P, Q.\, H(Q, P) \\
& \sigma P.\, H(\sigma P.\, H(P, P), P) \\
= & \text{ (} F(P) = H(P, P) \text{ as above)} \\
& \sigma P.\, H(\sigma P.\, F(P), P) \\
= & \text{ (unfold definition of } [\![\ ]\!] \text{ in } b \text{ in } G) \\
& \sigma P.\, G(P)
\end{aligned}$$

□

5. Swapping Variables in FES specifications

We now study swapping the order of variables in a specification. In general, this operation doesn't exactly preserve solutions. We first show that adjacent variables with the same sign may be swapped without changing the semantics (Section 5.1). Subsequently, we will prove that we can swap the order between blocks of equations, under certain independence criteria (Section 5.2). The main theorem of that section is new (cf. Section 7). Finally, we show that swapping a μ/ν sequence by the corresponding ν/μ in general leads to a greater or equal solution (Section 5.3).

5.1. **Swapping Equations with the same Sign.**

Theorem 5.1. *Assume $\mathcal{E} \in \mathcal{E}qs$ is monotonic. Then*

$$[\![\mathcal{E}, \mathcal{S}_1;\sigma X;\sigma Y;\mathcal{S}_2]\!] = [\![\mathcal{E}, \mathcal{S}_1;\sigma Y;\sigma X;\mathcal{S}_2]\!]$$

Proof. We first compute for arbitrary $\eta \in \mathcal{V}al$:

$$\begin{aligned}
&[\![\mathcal{E}, \sigma X;\sigma Y;\mathcal{S}_2]\!](\eta)\\
= {}&[\![\mathcal{E}, \sigma Y;\mathcal{S}_2]\!](\eta[X := \sigma(F_2)]), \text{ where}\\
&\quad F_2(P) = \mathcal{E}_X([\![\mathcal{E}, \sigma Y;\mathcal{S}_2]\!](\eta[X := P]))\\
= {}&[\![\mathcal{E}, \mathcal{S}_2]\!](\eta[X := \sigma(F_2),\, Y := \sigma(F_3)]), \text{ where}\\
&\quad F_2(P) = \mathcal{E}_X([\![\mathcal{E}, \sigma Y;\mathcal{S}_2]\!](\eta[X := P]))\\
&\quad F_3(Q) = \mathcal{E}_Y([\![\mathcal{E}, \mathcal{S}_2]\!](\eta[X := \sigma(F_2),\, Y := Q]))\\
= {}&\text{ (unfold definition of } [\![\ ]\!] \text{ in } F_2, \text{ introduce } F_1)\\
&[\![\mathcal{E}, \mathcal{S}_2]\!](\eta[X := \sigma(F_2),\, Y := \sigma(F_3)]), \text{ where}\\
&\quad F_1(P)(Q) = \mathcal{E}_Y([\![\mathcal{E}, \mathcal{S}_2]\!](\eta[X := P,\, Y := Q]))\\
&\quad F_2(P) = \mathcal{E}_X([\![\mathcal{E}, \mathcal{S}_2]\!](\eta[X := P,\, Y := \sigma(F_1(P))]))\\
&\quad F_3(Q) = \mathcal{E}_Y([\![\mathcal{E}, \mathcal{S}_2]\!](\eta[X := \sigma(F_2),\, Y := Q]))\\
= {}&A(\sigma(F_2), \sigma(F_3)), \text{ where}\\
&\quad A(P, Q) = [\![\mathcal{E}, \mathcal{S}_2]\!](\eta[X := P,\, Y := Q])\\
&\quad F_1(P)(Q) = \mathcal{E}_Y(A(P, Q))\\
&\quad F_2(P) = \mathcal{E}_X(A(P, \sigma(F_1(P))))\\
&\quad F_3(Q) = \mathcal{E}_Y(A(\sigma(F_2), Q))
\end{aligned}$$

Symmetrically, we get:

$$\begin{aligned}
&[\![\mathcal{E}, \sigma Y;\sigma X;\mathcal{S}_2]\!](\eta)\\
= {}&B(\sigma(G_2), \sigma(G_3)), \text{ where}\\
&\quad B(Q, P) = [\![\mathcal{E}, \mathcal{S}_2]\!](\eta[Y := Q,\, X := P])\\
&\quad G_1(Q)(P) = \mathcal{E}_X(B(Q, P))\\
&\quad G_2(Q) = \mathcal{E}_Y(B(Q, \sigma(G_1(Q))))\\
&\quad G_3(P) = \mathcal{E}_X(B(\sigma(G_2), P))
\end{aligned}$$

Note that the theorem is trivial when $X = Y$. So we may assume $X \neq Y$. Hence $A(P,Q) = B(Q,P)$, and we have:

$$\begin{aligned}
&\sigma(F_2)\\
&= \sigma P.\, \mathcal{E}_X(A(P, \sigma(F_1(P))))\\
&= \sigma P.\, \mathcal{E}_X(A(P, \sigma Q.\, \mathcal{E}_Y(A(P,Q))))\\
&= \quad \text{(Bekič rule, Lemma 2.1.9, with } H(p,q) := \mathcal{E}_X(A(p,q))\\
&\qquad \text{and } K(p,q) := \mathcal{E}_Y(A(q,p))\text{, which are monotonic, because}\\
&\qquad \mathcal{E} \text{ is by assumption, and } [\![\mathcal{E}, \mathcal{S}_2]\!] \text{ by Lemma 3.6.1)}\\
&\sigma P.\, \mathcal{E}_X(A(P, \sigma Q.\, \mathcal{E}_Y(A(\sigma P.\, \mathcal{E}_X(A(P,Q)), Q))))\\
&= \text{ (while } A(P,Q) = B(Q,P)\text{)}\\
&\sigma P.\, \mathcal{E}_X(B(\sigma Q.\, \mathcal{E}_Y(B(Q, \sigma P.\, \mathcal{E}_X(B(Q,P))))), P)\\
&= \sigma P.\, \mathcal{E}_X(B(\sigma Q.\, \mathcal{E}_Y(B(Q, \sigma(G_1(Q)))), P))\\
&= \sigma P.\, \mathcal{E}_X(B(\sigma(G_2), P))\\
&= \sigma(G_3)
\end{aligned}$$

We can now finish the proof:

$$\begin{aligned}
&\text{ (computation above, and full symmetry)}\\
&\sigma(F_2) = \sigma(G_3) \text{ and } \sigma(F_3) = \sigma(G_2)\\
\Rightarrow\ &\text{ (because } A(P,Q) = B(Q,P)\text{)}\\
&A(\sigma(F_2), \sigma(F_3)) = B(\sigma(G_2), \sigma(G_3))\\
\Rightarrow\ &[\![\mathcal{E}, \sigma X;\sigma Y;\mathcal{S}_2]\!](\eta) = [\![\mathcal{E}, \sigma Y;\sigma X;\mathcal{S}_2]\!](\eta)\\
\Rightarrow\ &\text{ (Lemma 3.9)}\\
&[\![\mathcal{E}, \mathcal{S}_1;\sigma X;\sigma Y;\mathcal{S}_2]\!](\eta) = [\![\mathcal{E}, \mathcal{S}_1;\sigma Y;\sigma X;\mathcal{S}_2]\!](\eta)
\end{aligned}$$ □

5.2. **Migrating Independent Blocks of Equations.** Our aim here is to investigate swapping blocks of equations that are independent. We first need two technical lemmas. The first lemma enables to commute updates to valuations with computing solutions:

Lemma 5.2. *If $X \notin dom(\mathcal{S})$ and $indep(\mathcal{E}, \mathcal{S}, X)$, then*

$$[\![\mathcal{E}, \mathcal{S}]\!](\eta[X := P]) = ([\![\mathcal{E}, \mathcal{S}]\!](\eta))[X := P]$$

Proof. Induction on $\mathcal{S}$. The base case is trivial:

$$[\![\mathcal{E}, \varepsilon]\!](\eta[X := P]) = \eta[X := P] = [\![\mathcal{E}, \varepsilon]\!](\eta)[X := P]$$

Case $\mathcal{S} = \sigma Y;\mathcal{S}_1$. Assume $X \notin dom(\mathcal{S})$ and $indep(\mathcal{E}, \mathcal{S}, X)$, then $X \neq Y$, and also $X \notin dom(\mathcal{S}_1)$ and $indep(\mathcal{E}, \mathcal{S}_1, X)$, so the induction hypothesis can be applied. Then

$$\begin{aligned}
&[\![\mathcal{E}, \sigma Y;\mathcal{S}_1]\!](\eta[X := P])\\
&= [\![\mathcal{E}, \mathcal{S}_1]\!](\eta[X := P,\, Y := \sigma(F)]), \text{ where}
\end{aligned}$$

$$
\begin{aligned}
F(Q) &:= \mathcal{E}_Y(\llbracket \mathcal{E}, \mathcal{S}_1 \rrbracket(\eta[X := P, Y := Q])) \\
&= \text{(induction hypothesis, and } X \neq Y) \\
&\quad \mathcal{E}_Y(\llbracket \mathcal{E}, \mathcal{S}_1 \rrbracket(\eta[Y := Q])[X := P]) \\
&= (Y \in \mathit{dom}(\mathcal{S}) \text{ is independent of } X) \\
&\quad \mathcal{E}_Y(\llbracket \mathcal{E}, \mathcal{S}_1 \rrbracket(\eta[Y := Q])) \\
&=: G(Q)
\end{aligned}
$$

$$
\begin{aligned}
&= \llbracket \mathcal{E}, \mathcal{S}_1 \rrbracket(\eta[X := P, Y := \sigma(G)]) \\
&= \text{(induction hypothesis, and } X \neq Y) \\
&\quad \llbracket \mathcal{E}, \mathcal{S}_1 \rrbracket(\eta[Y := \sigma(G)])[X := P] \\
&= \llbracket \mathcal{E}, \sigma Y ; \mathcal{S}_1 \rrbracket(\eta)[X := P]
\end{aligned}
$$

□

The next lemma states that independent specifications can be solved independently.

Lemma 5.3. *Let $\mathcal{E} \in \mathcal{E}qs$ and $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{S}pec$. If $\mathit{indep}(\mathcal{E}, \mathcal{S}_1, \mathcal{S}_2)$, then for all $\eta \in \mathcal{V}al$, $\llbracket \mathcal{E}, \mathcal{S}_1;\mathcal{S}_2 \rrbracket(\eta) = \llbracket \mathcal{E}, \mathcal{S}_2 \rrbracket(\llbracket \mathcal{E}, \mathcal{S}_1 \rrbracket(\eta))$.*

Proof. Induction on $\mathcal{S}_1$. In case $\mathcal{S}_1 = \varepsilon$ we obtain indeed:

$$\llbracket \mathcal{E}, \varepsilon;\mathcal{S}_2 \rrbracket(\eta) = \llbracket \mathcal{E}, \mathcal{S}_2 \rrbracket(\eta) = \llbracket \mathcal{E}, \mathcal{S}_2 \rrbracket(\llbracket \mathcal{E}, \varepsilon \rrbracket(\eta))$$

Next, consider $\mathcal{S}_1 = \sigma X;\mathcal{S}$. Assume $\mathit{indep}(\mathcal{E}, \mathcal{S}_1, \mathcal{S}_2)$, then it follows that $\mathit{indep}(\mathcal{E}, \mathcal{S}, \mathcal{S}_2)$, so we can use the induction hypothesis. Define:

$$
\begin{aligned}
F(P) &:= \mathcal{E}_X(\llbracket \mathcal{E}, \mathcal{S};\mathcal{S}_2 \rrbracket(\eta[X := P])) \\
G(P) &:= \mathcal{E}_X(\llbracket \mathcal{E}, \mathcal{S} \rrbracket(\eta[X := P]))
\end{aligned}
$$

In order to show that $F = G$, it suffices (because $\mathcal{E}_X$ is *independent* of $\mathcal{S}_2$) to show that for any $P \in U$ and $Y \notin \mathit{dom}(\mathcal{S}_2)$:

$$
\begin{aligned}
&\llbracket \mathcal{E}, \mathcal{S};\mathcal{S}_2 \rrbracket(\eta[X := P])(Y) \\
=\ &\text{(induction hypothesis)} \\
&\llbracket \mathcal{E}, \mathcal{S}_2 \rrbracket(\llbracket \mathcal{E}, \mathcal{S} \rrbracket(\eta[X := P]))(Y) \\
=\ &\text{(Lemma 3.7.1)} \\
&\llbracket \mathcal{E}, \mathcal{S} \rrbracket(\eta[X := P])(Y)
\end{aligned}
$$

Next, we finish the proof with the following calculation:

$$
\begin{aligned}
&\llbracket \mathcal{E}, \sigma X;\mathcal{S};\mathcal{S}_2 \rrbracket(\eta) \\
=\ &\llbracket \mathcal{E}, \mathcal{S};\mathcal{S}_2 \rrbracket(\eta[X := \sigma(F)]) \\
=\ &\text{(induction hypothesis)} \\
&\llbracket \mathcal{E}, \mathcal{S}_2 \rrbracket(\llbracket \mathcal{E}, \mathcal{S} \rrbracket(\eta[X := \sigma(F)])) \\
=\ &(F = G\text{, see above}) \\
&\llbracket \mathcal{E}, \mathcal{S}_2 \rrbracket(\llbracket \mathcal{E}, \mathcal{S} \rrbracket(\eta[X := \sigma(G)])) \\
=\ &\llbracket \mathcal{E}, \mathcal{S}_2 \rrbracket(\llbracket \mathcal{E}, \sigma X;\mathcal{S} \rrbracket(\eta))
\end{aligned}
$$

□

The next theorem shows that two disjoint blocks of equations can be swapped, provided one of them doesn't depend on the other. Note that a dependence in one direction is allowed, and that it doesn't matter in which direction by symmetry. Theorem 5.7 will generalize this by adding left- and right-contexts under certain conditions.

Theorem 5.4. *Let $disjoint(\mathcal{S}_1, \mathcal{S}_2)$ and $indep(\mathcal{E}, \mathcal{S}_1, \mathcal{S}_2)$. Then $[\![\mathcal{E}, \mathcal{S}_1;\mathcal{S}_2]\!] = [\![\mathcal{E}, \mathcal{S}_2;\mathcal{S}_1]\!]$.*

Proof. Induction on $\mathcal{S}_2$. The base case is trivial. For the induction step, let $\mathcal{S}_2 = \sigma X;\mathcal{S}$. If we assume $disjoint(\mathcal{S}_1, \mathcal{S}_2)$ and $indep(\mathcal{E}, \mathcal{S}_1, \mathcal{S}_2)$, then we also obtain $disjoint(\mathcal{S}_1, \mathcal{S})$ and $indep(\mathcal{E}, \mathcal{S}_1, \mathcal{S})$, so we may apply the induction hypothesis $[\![\mathcal{E}, \mathcal{S}_1;\mathcal{S}]\!] = [\![\mathcal{E}, \mathcal{S};\mathcal{S}_1]\!]$. Furthermore, from the same assumptions, we also get $X \notin dom(\mathcal{S}_1)$ and $indep(\mathcal{E}, \mathcal{S}_1, X)$. Let η be arbitrary.

$$\begin{aligned}
& [\![\mathcal{E}, \mathcal{S}_1;\sigma X;\mathcal{S}]\!](\eta) \\
= {} & \text{(Lemma 5.3)} \\
& [\![\mathcal{E}, \sigma X;\mathcal{S}]\!]([\![\mathcal{E}, \mathcal{S}_1]\!](\eta)) \\
= {} & [\![\mathcal{E}, \mathcal{S}]\!]([\![\mathcal{E}, \mathcal{S}_1]\!](\eta)[X := \sigma(F)]), \text{ where} \\
& \qquad F(P) := \mathcal{E}_X([\![\mathcal{E}, \mathcal{S}]\!]([\![\mathcal{E}, \mathcal{S}_1]\!](\eta)[X := P]) \\
& \qquad\qquad = \text{(Lemma 5.2)} \\
& \qquad\qquad\quad \mathcal{E}_X([\![\mathcal{E}, \mathcal{S}]\!]([\![\mathcal{E}, \mathcal{S}_1]\!](\eta[X := P]))) \\
& \qquad\qquad = \text{(Lemma 5.3)} \\
& \qquad\qquad\quad \mathcal{E}_X([\![\mathcal{E}, \mathcal{S}_1;\mathcal{S}]\!](\eta[X := P])) \\
& \qquad\qquad =: G(P) \\
= {} & [\![\mathcal{E}, \mathcal{S}]\!]([\![\mathcal{E}, \mathcal{S}_1]\!](\eta)[X := \sigma(G)]) \\
= {} & \text{(Lemma 5.2)} \\
& [\![\mathcal{E}, \mathcal{S}]\!]([\![\mathcal{E}, \mathcal{S}_1]\!](\eta[X := \sigma(G)])) \\
= {} & \text{(Lemma 5.3)} \\
& [\![\mathcal{E}, \mathcal{S}_1;\mathcal{S}]\!](\eta[X := \sigma(G)]) \\
= {} & [\![\mathcal{E}, \sigma X;\mathcal{S}_1;\mathcal{S}]\!](\eta) \\
= {} & \text{(Lemma 3.9 and induction hypothesis)} \\
= {} & [\![\mathcal{E}, \sigma X;\mathcal{S};\mathcal{S}_1]\!](\eta)
\end{aligned}$$
□

This migration theorem has several interesting corollaries. First, we get right-congruence for independent specifications.

Corollary 5.5. *Assume that $indep(\mathcal{E}_1, \mathcal{S}_1, \mathcal{S})$, $indep(\mathcal{E}_2, \mathcal{S}_2, \mathcal{S})$, $disjoint(\mathcal{S}, \mathcal{S}_1;\mathcal{S}_2)$ and that $\mathcal{E}_1 \overset{\mathcal{S}}{=} \mathcal{E}_2$. Then $[\![\mathcal{E}_1, \mathcal{S}_1]\!] = [\![\mathcal{E}_2, \mathcal{S}_2]\!]$ implies $[\![\mathcal{E}_1, \mathcal{S}_1;\mathcal{S}]\!] = [\![\mathcal{E}_2, \mathcal{S}_2;\mathcal{S}]\!]$.*

We also get the near-reverse of Lemma 5.3:

Corollary 5.6. *Let $indep(\mathcal{E}, \mathcal{S}_2, \mathcal{S}_1)$ and $disjoint(\mathcal{S}_1, \mathcal{S}_2)$. Then for all $\eta \in \mathcal{V}al$, we have $[\![\mathcal{E}, \mathcal{S}_1;\mathcal{S}_2]\!](\eta) = [\![\mathcal{E}, \mathcal{S}_1]\!]([\![\mathcal{E}, \mathcal{S}_2]\!](\eta))$.*

Proof. Under the given assumptions, we obtain from Theorem 5.4 (applied from right to left) and Lemma 5.3:

$$[\![\mathcal{E}, \mathcal{S}_1;\mathcal{S}_2]\!](\eta) = [\![\mathcal{E}, \mathcal{S}_2;\mathcal{S}_1]\!](\eta) = [\![\mathcal{E}, \mathcal{S}_1]\!]([\![\mathcal{E}, \mathcal{S}_2]\!](\eta))$$
□

Theorem 5.7. *Assume that $disjoint(\mathcal{S}_1, \mathcal{S}_2;\mathcal{S}_3)$. Also, assume that either $indep(\mathcal{E}, \mathcal{S}_1, \mathcal{S}_2;\mathcal{S}_3)$, or $indep(\mathcal{E}, \mathcal{S}_2;\mathcal{S}_3, \mathcal{S}_1)$. Then $[\![\mathcal{E}, \mathcal{S}_0;\mathcal{S}_1;\mathcal{S}_2;\mathcal{S}_3]\!] = [\![\mathcal{E}, \mathcal{S}_0;\mathcal{S}_2;\mathcal{S}_1;\mathcal{S}_3]\!]$.*

Proof. We also have *disjoint*$(\mathcal{S}_1, \mathcal{S}_3)$, and either *indep*$(\mathcal{E}, \mathcal{S}_1, \mathcal{S}_3)$ or *indep*$(\mathcal{E}, \mathcal{S}_3, \mathcal{S}_1)$. Using Lemma 3.9 and Theorem 5.4 twice (in either direction) we get:

$$[\![\mathcal{E}, \mathcal{S}_0;\mathcal{S}_1;\mathcal{S}_2;\mathcal{S}_3]\!] = [\![\mathcal{E}, \mathcal{S}_0;\mathcal{S}_2;\mathcal{S}_3;\mathcal{S}_1]\!] = [\![\mathcal{E}, \mathcal{S}_0;\mathcal{S}_2;\mathcal{S}_1;\mathcal{S}_3]\!] \qquad \square$$

5.3. **Inequalities by Swapping or Changing Signs.** In this section, we will prove a few inequalities. Theorem 5.9 shows the consequence of swapping variables with a different sign; Theorem 5.10 shows the effect of changing the sign of a variable. But first, it will be shown that $\leq$ is a left congruence:

Lemma 5.8. *Let $\mathcal{E}_1, \mathcal{E}_2 \in \mathcal{E}qs$ and $\mathcal{S}, \mathcal{S}_1, \mathcal{S}_2 \in \mathcal{S}pec$. If $\mathcal{E}_1$ is monotonic, $\mathcal{E}_1 \overset{\mathcal{S}}{\equiv} \mathcal{E}_2$, and $[\![\mathcal{E}_1, \mathcal{S}_1]\!] \leq [\![\mathcal{E}_2, \mathcal{S}_2]\!]$, then $[\![\mathcal{E}_1, \mathcal{S};\mathcal{S}_1]\!] \leq [\![\mathcal{E}_2, \mathcal{S};\mathcal{S}_2]\!]$.*

Proof. Induction on $\mathcal{S}$. The base case is trivial.

$$\begin{aligned}
&[\![\mathcal{E}_1, \sigma X;\mathcal{S};\mathcal{S}_1]\!](\eta)\\
=\ &[\![\mathcal{E}_1, \mathcal{S};\mathcal{S}_1]\!](\eta[X := \sigma(F)]), where\\
&\quad F(P) := \mathcal{E}_{1,X}([\![\mathcal{E}_1, \mathcal{S};\mathcal{S}_1]\!](\eta[X := P]))\\
&\qquad\quad \leq \text{ (by induction hypothesis and } \mathcal{E}_1 \text{ monotonic)}\\
&\qquad\quad\ \ \mathcal{E}_{1,X}([\![\mathcal{E}_2, \mathcal{S};\mathcal{S}_2]\!](\eta[X := P]))\\
&\qquad\quad = \mathcal{E}_{2,X}([\![\mathcal{E}_2, \mathcal{S};\mathcal{S}_2]\!](\eta[X := P]))\\
&\qquad\quad =: G(P)\\
\leq\ &\text{(Using Lemma 3.6.1)}\\
&[\![\mathcal{E}_1, \mathcal{S};\mathcal{S}_1]\!](\eta[X := \sigma(G)])\\
\leq\ &\text{(by induction hypothesis)}\\
&[\![\mathcal{E}_2, \mathcal{S};\mathcal{S}_2]\!](\eta[X := \sigma(G)])\\
=\ &[\![\mathcal{E}_2, \sigma X;\mathcal{S};\mathcal{S}_2]\!](\eta)
\end{aligned} \qquad \square$$

Note that, by duality, the above lemma may also be applied if $\mathcal{E}_2$ is monotonic instead of $\mathcal{E}_1$. Moreover, note that (only) for monotonic $\mathcal{E}_1$, Lemma 3.9 would follow from Lemma 5.8.

Theorem 5.9. *Assume $\mathcal{E} \in \mathcal{E}qs$ is monotonic and $X \neq Y$. Then*

$$[\![\mathcal{E}, \mathcal{S}_1;\mu X;\nu Y;\mathcal{S}_2]\!] \leq [\![\mathcal{E}, \mathcal{S}_1;\nu Y;\mu X;\mathcal{S}_2]\!]$$

Proof. As in Theorem 5.1, and using $X \neq Y$, we obtain:

$$\begin{aligned}
[\![\mathcal{E}, \mu X;\nu Y;\mathcal{S}_2]\!](\eta) &= A(\mu(F_2), \nu(F_3)), \text{ where}\\
A(P, Q) &= [\![\mathcal{E}, \mathcal{S}_2]\!](\eta[X := P, Y := Q])\\
F_1(P)(Q) &= \mathcal{E}_Y(A(P, Q))\\
F_2(P) &= \mathcal{E}_X(A(P, \nu(F_1(P))))\\
F_3(Q) &= \mathcal{E}_Y(A(\mu(F_2), Q))
\end{aligned}$$

$$\begin{aligned}
[\![\mathcal{E}, \nu Y;\mu X;\mathcal{S}_2]\!](\eta) &= A(\mu(G_3), \nu(G_2)), \text{ where}\\
G_1(Q)(P) &= \mathcal{E}_X(A(P, Q))\\
G_2(Q) &= \mathcal{E}_Y(A(\mu(G_1(Q)), Q))\\
G_3(P) &= \mathcal{E}_X(A(P, \nu(G_2)))
\end{aligned}$$

By Lemma 2.2.4(a), $\mu(F_2) \leq \mu(G_3)$, and by Lemma 2.2.4(b), $\nu(F_3) \leq \nu(G_2)$, whence it follows that $[\![\mathcal{E}, \mu X;\nu Y;\mathcal{S}_2]\!](\eta) \leq [\![\mathcal{E}, \nu Y;\mu X;\mathcal{S}_2]\!](\eta)$. The theorem then follows by Lemma 5.8. □

We end this section with another inequality:

Theorem 5.10. *If $\mathcal{E}$ is monotonic, then $[\![\mathcal{E}, \mathcal{S}_1;\mu X;\mathcal{S}_2]\!] \leq [\![\mathcal{E}, \mathcal{S}_1;\nu X;\mathcal{S}_2]\!]$.*

Proof. Let η be an arbitrary valuation, and define $F : U \to U$ by $F(P) := \mathcal{E}_X([\![\mathcal{E}, \mathcal{S}_2]\!](\eta[X := P]))$. We then have:

$$
\begin{aligned}
& \text{(Theorem 2.2.1)}\\
& \mu P.\, F(P) \leq \nu P.\, F(P)\\
\Rightarrow\ & \text{(Monotonicity, Theorem 3.6.1)}\\
& [\![\mathcal{E}, \mathcal{S}_2]\!](\eta[X := \mu P.\, F(P)]) \leq [\![\mathcal{E}, \mathcal{S}_2]\!](\eta[X := \nu P.\, F(P)])\\
\Rightarrow\ & \text{(Definition semantics)}\\
& [\![\mathcal{E}, \mu X;\mathcal{S}]\!](\eta) \leq [\![\mathcal{E}, \nu X;\mathcal{S}_2]\!](\eta)\\
\Rightarrow\ & (\eta \text{ was arbitrary})\\
& [\![\mathcal{E}, \mu X;\mathcal{S}]\!] \leq [\![\mathcal{E}, \nu X;\mathcal{S}_2]\!]\\
\Rightarrow\ & \text{(Congruence, Theorem 5.8)}\\
& [\![\mathcal{E}, \mathcal{S}_1;\mu X;\mathcal{S}_2]\!] \leq [\![\mathcal{E}, \mathcal{S}_1;\nu X;\mathcal{S}_2]\!]
\end{aligned}
$$

□

In the next section, we will see sufficient conditions under which the inequality signs of these theorems can be turned into equalities. These conditions will be phrased in terms of the dependency graph.

6. Indirect Dependencies and Loops

In Section 5.2, we studied direct dependencies between variables. Basically, a direct dependency of X on Y means that Y occurs in the definition of X. We will now study the effect of indirect dependencies, written $X \xrightarrow{\mathcal{E},\mathcal{S}} Y$ (cf. the definitions in Section 3.3)

Given a specification $\mathcal{S}$ and a computable predicate f on variables, we define $\mathit{split}_f(\mathcal{S}) = (\mathcal{S}_1, \mathcal{S}_2)$, where $\mathcal{S}_1$ is the sublist of $\mathcal{S}$ with those X for which $f(X)$ does *not* hold and $\mathcal{S}_2$ is the sublist of $\mathcal{S}$ with those X for which $f(X)$ *does* hold. Notice that, within $\mathcal{S}_1$ and $\mathcal{S}_2$, variables keep their order from $\mathcal{S}$.

The following basic facts follow directly from the definition of *split*.

Lemma 6.1. *Let $\mathit{split}_f(\mathcal{S}) = (\mathcal{S}_1, \mathcal{S}_2)$, then $\mathit{dom}(\mathcal{S}) = \mathit{dom}(\mathcal{S}_1) \cup \mathit{dom}(\mathcal{S}_2)$ and $\mathit{disjoint}(\mathcal{S}_1, \mathcal{S}_2)$.*

We first show how the equations in a FES may be rearranged if the specification is split in such a way that certain independence conditions are fulfilled.

Lemma 6.2. *Let f be a predicate and $\mathcal{S}$ a specification such that $\mathit{split}_f(\mathcal{S}) = (\mathcal{S}_1, \mathcal{S}_2)$ and $\mathit{indep}(\mathcal{E}, \mathcal{S}_2, \mathcal{S}_1)$. Then $[\![\mathcal{E}, \mathcal{S}]\!] = [\![\mathcal{E}, \mathcal{S}_1;\mathcal{S}_2]\!]$.*

Proof. We perform induction on $\mathcal{S}$. The base case is trivial. Let $\mathit{split}_f(\mathcal{S}) = (\mathcal{S}_1, \mathcal{S}_2)$ and assume as induction hypothesis that $\mathit{indep}(\mathcal{E}, \mathcal{S}_2, \mathcal{S}_1)$ implies $[\![\mathcal{E}, \mathcal{S}]\!] = [\![\mathcal{E}, \mathcal{S}_1;\mathcal{S}_2]\!]$. For the induction step, we consider the specification $\sigma Y;\mathcal{S}$ and distinguish two cases.

If $f(Y)$ does not hold, then we have $\mathit{split}_f(\sigma Y;\mathcal{S}) = (\sigma Y;\mathcal{S}_1, \mathcal{S}_2)$. Accordingly, we assume $\mathit{indep}(\mathcal{E}, \sigma Y;\mathcal{S}_2, \mathcal{S}_1)$, which implies $\mathit{indep}(\mathcal{E}, \mathcal{S}_2, \mathcal{S}_1)$. We can thus apply the induction hypothesis and congruence (Lemma 3.9) to obtain $[\![\mathcal{E}, \sigma Y;\mathcal{S}]\!] = [\![\mathcal{E}, \sigma Y;\mathcal{S}_1;\mathcal{S}_2]\!]$.

Otherwise, if $f(Y)$ holds, we obtain $\mathit{split}_f(\sigma Y;\mathcal{S}) = (\mathcal{S}_1, \sigma Y;\mathcal{S}_2)$. Now we assume that $\mathit{indep}(\mathcal{E}, \mathcal{S}_2, \sigma Y;\mathcal{S}_1)$, which again implies $\mathit{indep}(\mathcal{E}, \mathcal{S}_2, \mathcal{S}_1)$. We also have $\mathit{disjoint}(\mathcal{S}_1, \sigma Y;\mathcal{S}_2)$ (Lemma 6.1), so we may apply Theorem 5.7 below:

$$\begin{aligned}
&[\![\mathcal{E}, \sigma Y;\mathcal{S}]\!] \\
=\ &\text{(by induction hypothesis and Lemma 3.9)} \\
&[\![\mathcal{E}, \sigma Y;\mathcal{S}_1;\mathcal{S}_2]\!] \\
=\ &\text{(by Theorem 5.7)} \\
&[\![\mathcal{E}, \mathcal{S}_1;\sigma Y;\mathcal{S}_2]\!]
\end{aligned}$$

□

We simplify notation and write $\mathit{split}_{X,\mathcal{E}}(\mathcal{S})$ for $\mathit{split}_{\mathit{dep}^X_{\mathcal{E},\mathcal{S}}}(\mathcal{S})$, defining the predicate $\mathit{dep}^X_{\mathcal{E},\mathcal{S}}(Y) = X \xrightarrow{\mathcal{E},\mathcal{S}}\!\!\!\!\!\rightarrow Y$. If $\mathit{indep}(\mathcal{E}, X, Y)$ is computable (which we assume henceforward), then $X \xrightarrow{\mathcal{E},\mathcal{S}}\!\!\!\!\!\rightarrow Y$ is also computable since $\mathcal{S}$ is finite. Intuitively, if $\mathit{split}_{X,\mathcal{E}}(\mathcal{S}) = (\mathcal{S}_1, \mathcal{S}_2)$, then $\mathcal{S}_1$ is the sublist of $\mathcal{S}$ with those Y on which X *does not* depend indirectly; and $\mathcal{S}_2$ is the sublist of $\mathcal{S}$ with those Z on which X *does* depend indirectly. Notice that, if $X \notin \mathit{dom}(\mathcal{S})$, then $\mathit{split}_{X,\mathcal{E}}(\mathcal{S}) = (\mathcal{S}, \varepsilon)$.

We have the following lemma about splitting a specification based on the dependencies of X:

Lemma 6.3. *If* $\mathit{split}_{X,\mathcal{E}}(\mathcal{S}) = (\mathcal{S}_1, \mathcal{S}_2)$*, then* $\mathit{indep}(\mathcal{E}, \mathcal{S}_2, \mathcal{S}_1)$.

Proof. Assume that some $Z \in \mathit{dom}(\mathcal{S}_2)$ would use some $Y \in \mathit{dom}(\mathcal{S}_1)$ in its definition in $\mathcal{E}$. Then $X \xrightarrow{\mathcal{E},\mathcal{S}}\!\!\!\!\!\rightarrow Z \xrightarrow{\mathcal{E},\mathcal{S}} Y$, so $X \xrightarrow{\mathcal{E},\mathcal{S}}\!\!\!\!\!\rightarrow Y$, and Y would be in $\mathcal{S}_2$ and not in $\mathcal{S}_1$. □

The key theorem of this section states that the equations in a FES can be rearranged, such that all equations that X depends on precede all other equations, or vice versa. This is useful, because those parts can be solved independently, using Lemma 5.3. By repeatedly picking a variable in a terminal strongly connected component of the remaining variable dependency graph, one can thus solve all SCCs one by one. This idea already appeared in [Jur00] for parity games, although it does not always provide performance benefits in practice [FL09]. The theorem may also be used to reduce the number of fixpoint alternations in a FES.

Theorem 6.4. *Let* $\mathit{split}_{X,\mathcal{E}}(\mathcal{S}) = (\mathcal{S}_1, \mathcal{S}_2)$*. Then* $[\![\mathcal{E}, \mathcal{S}]\!] = [\![\mathcal{E}, \mathcal{S}_1;\mathcal{S}_2]\!] = [\![\mathcal{E}, \mathcal{S}_2;\mathcal{S}_1]\!]$.

Proof. The first equality follows from Lemmas 6.2 and 6.3. From Theorem 5.4 and Lemmas 6.1 and 6.3 it follows that also $[\![\mathcal{E}, \mathcal{S}_1;\mathcal{S}_2]\!] = [\![\mathcal{E}, \mathcal{S}_2;\mathcal{S}_1]\!]$. □

Based on this reordering principle, we can prove three more interesting results, which we will do in the next subsections.

6.1. Swapping Signs and Dependency Loops.

The first result (Theorem 6.7) states that the sign of a variable X is only relevant if it depends on itself, *i.e.*, X is on a cycle in $\xrightarrow{\mathcal{E},\mathcal{S}}{}^{+}$ (recall that $\xrightarrow{\mathcal{E},\mathcal{S}}{}^{+}$ indicates a non-empty path in the variable dependency graph). We first need a couple of auxiliary lemmas:

Lemma 6.5. *If $X \notin dom(\mathcal{S})$ and we have $indep(\mathcal{E}, \mathcal{S}, X)$ as well as $indep(\mathcal{E}, X, X)$, then $[\![\mathcal{E}, \mu X; \mathcal{S}]\!] = [\![\mathcal{E}, \nu X; \mathcal{S}]\!]$.*

Proof. For $\sigma \in \{\mu, \nu\}$ and arbitrary valuation η, we have:

$$\begin{aligned}
& [\![\mathcal{E}, \sigma X; \mathcal{S}]\!](\eta) \\
= \; & \text{(by definition of semantics)} \\
& [\![\mathcal{E}, \mathcal{S}]\!](\eta[X := \sigma(F)]), \text{ where} \\
& \quad F(P) := \mathcal{E}_X([\![\mathcal{E}, \mathcal{S}]\!](\eta[X := P])) \\
& \qquad = \; \text{(Lemma 5.2, and } X \notin dom(\mathcal{S}) \text{ and } indep(\mathcal{E}, \mathcal{S}, X)\text{)} \\
& \qquad \quad \mathcal{E}_X(([\![\mathcal{E}, \mathcal{S}]\!]\eta)[X := P]) \\
& \qquad = \; \text{(by definition of } indep(\mathcal{E}, X, X)\text{)} \\
& \qquad \quad \mathcal{E}_X([\![\mathcal{E}, \mathcal{S}]\!]\eta) \\
= \; & \text{(constant rule, Lemma 2.1.2)} \\
& [\![\mathcal{E}, \mathcal{S}]\!](\eta[X := \mathcal{E}_X([\![\mathcal{E}, \mathcal{S}]\!]\eta)])
\end{aligned}$$

So indeed $[\![\mathcal{E}, \mu X; \mathcal{S}]\!] = [\![\mathcal{E}, \nu X; \mathcal{S}]\!]$. ☐

Lemma 6.6. *If not $X \xrightarrow{\mathcal{E}, \mu X; \mathcal{S}}{}^{+} X$, and $X \notin dom(\mathcal{S})$, then $[\![\mathcal{E}, \mu X; \mathcal{S}]\!] = [\![\mathcal{E}, \nu X; \mathcal{S}]\!]$.*

Proof. Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be such that $split_{X,\mathcal{E}}(\sigma X; \mathcal{S}) = (\mathcal{S}_1, \sigma X; \mathcal{S}_2)$, for $\sigma \in \{\mu, \nu\}$. Note that if not $indep(\mathcal{E}, \mu X; \mathcal{S}_2, X)$, then for some $Y \in dom(\mu X; \mathcal{S}_2)$, by definition of *split*, $X \overset{\mathcal{E}, \mu X; \mathcal{S}}{\twoheadrightarrow} Y \xrightarrow{\mathcal{E}} X$, which contradicts the assumption not $X \xrightarrow{\mathcal{E}, \mu X; \mathcal{S}}{}^{+} X$. From $dom(\mathcal{S}_2) \subseteq dom(\mathcal{S})$, we obtain $X \notin dom(\mathcal{S}_2)$. Hence, $X \notin dom(\mathcal{S}_2)$ and $indep(\mathcal{E}, \mu X; \mathcal{S}_2, X)$, so Lemma 6.5 applies. Together with Theorem 6.4 and Lemma 3.9, we then compute:

$$[\![\mathcal{E}, \mu X; \mathcal{S}]\!] = [\![\mathcal{E}, \mathcal{S}_1; \mu X; \mathcal{S}_2]\!] = [\![\mathcal{E}, \mathcal{S}_1; \nu X; \mathcal{S}_2]\!] = [\![\mathcal{E}, \nu X; \mathcal{S}]\!]$$ ☐

Intuitively, the sign of X is only relevant if X is the most relevant variable (i.e. leftmost in the specification) on some loop in the dependency graph. So in the full theorem, we can restrict to dependencies through variables right from X:

Theorem 6.7. *Assume that not $X \xrightarrow{\mathcal{E}, \mu X; \mathcal{S}_2}{}^{+} X$, and $X \notin dom(\mathcal{S}_2)$. Then*

$$[\![\mathcal{E}, \mathcal{S}_1; \mu X; \mathcal{S}_2]\!] = [\![\mathcal{E}, \mathcal{S}_1; \nu X; \mathcal{S}_2]\!]$$

Proof. By Lemma 6.6, $[\![\mathcal{E}, \mu X; \mathcal{S}_2]\!] = [\![\mathcal{E}, \nu X; \mathcal{S}_2]\!]$. The result follows by congruence, Lemma 3.9. ☐

6.2. **Reordering Variables and Dependency Loops.**

The second result (Theorem 6.9) allows to swap any two neighbouring variables that don't occur on a loop in the dependency graph.

Lemma 6.8. *Let not $X \overset{\mathcal{E}, \sigma X; \rho Y; \mathcal{S}}{\twoheadrightarrow} Y$. Then $[\![\mathcal{E}, \sigma X; \rho Y; \mathcal{S}]\!] = [\![\mathcal{E}, \rho Y; \sigma X; \mathcal{S}]\!]$.*

Proof. Note that for some $\mathcal{S}_1$ and $\mathcal{S}_2$, we have

$$split_{X,\mathcal{E}}(\sigma X; \rho Y; \mathcal{S}) = (\rho Y; \mathcal{S}_1, \sigma X; \mathcal{S}_2) = split_{X,\mathcal{E}}(\rho Y; \sigma X; \mathcal{S})$$

Hence, by applying Theorem 6.4 twice, we obtain:

$$[\![\mathcal{E}, \sigma X; \rho Y; \mathcal{S}]\!] = [\![\mathcal{E}, \rho Y; \mathcal{S}_1; \sigma X; \mathcal{S}_2]\!] = [\![\mathcal{E}, \rho Y; \sigma X; \mathcal{S}]\!]$$ ☐

Again, we can strengthen this, by observing that X and Y can be swapped, when there is no loop that has either X or Y as its most relevant variable in the specification:

Theorem 6.9. *Assume that not $X \xrightarrow{\mathcal{E},\sigma X;\rho Y;\mathcal{S}_2} Y$. Then we have*

$$[\![\mathcal{E}, \mathcal{S}_1;\sigma X;\rho Y;\mathcal{S}_2]\!] = [\![\mathcal{E}, \mathcal{S}_1;\rho Y;\sigma X;\mathcal{S}_2]\!]$$

Proof. By Lemma 6.8, $[\![\mathcal{E}, \sigma X;\rho Y;\mathcal{S}_2]\!] = [\![\mathcal{E}, \rho Y;\sigma X;\mathcal{S}_2]\!]$. The result then follows by congruence, Lemma 3.9. ☐

Note that this result strengthens Theorem 5.1 (the signs may now be different), Theorem 5.7 (we can have mutual dependencies on $\mathcal{S}$, as long as no loop is introduced) and Theorem 5.9 (we have here equality rather than inequality).

6.3. Forward Substitution and Dependency Loops.

The final result strengthens Theorem 4.3 by allowing unfolding of Y in the definition of X, even if Y precedes X in the specification, provided Y doesn't depend on X:

Theorem 6.10. *Let $\mathcal{E} \in \mathcal{E}qs$ be monotonic. Let $\mathcal{S} = \mathcal{S}_1, \sigma Y, \mathcal{S}_2$. Assume that not $Y \xrightarrow{\mathcal{E},\mathcal{S}} X$. Then $[\![\mathcal{E}, \mathcal{S}]\!] = [\![unfold(\mathcal{E}, X, Y), \mathcal{S}]\!]$.*

Proof. Assume not $Y \xrightarrow{\mathcal{E},\mathcal{S}} X$. Then the following two observations hold:
(1) for all $Z \in \mathcal{X}$, $Y \xrightarrow{\mathcal{E},\mathcal{S}} Z \iff Y \xrightarrow{unfold(\mathcal{E},X,Y),\mathcal{S}} Z$
(2) $split_{Y,\mathcal{E}}(\mathcal{S}) = split_{Y,unfold(\mathcal{E},X,Y)}(\mathcal{S})$
The first item holds, because $unfold(\mathcal{E}, X, Y)$ only modifies the definition of X, but Y doesn't refer to it. The second then follows from the definition of *split*.

Let $(L_1, L_2) := split_{Y,\mathcal{E}}(\mathcal{S})$. Then, as $Y \xrightarrow{\mathcal{E},\mathcal{S}} Y$, we have $L_2 = L_3;\sigma Y;L_4$. Note that $X \notin dom(L_4)$, for we would then have $Y \xrightarrow{\mathcal{E},\mathcal{S}} X$, contradicting the assumptions. Then we can compute:

$$\begin{aligned}
& [\![\mathcal{E}, \mathcal{S}]\!] \\
= \ & \text{(Theorem 6.4)} \\
& [\![\mathcal{E}, L_1;L_3;\sigma Y;L_4]\!] \\
= \ & \text{(Theorem 4.3)} \\
& [\![unfold(\mathcal{E}, X, Y), L_1;L_3;\sigma Y;L_4]\!] \\
= \ & \text{(Theorem 6.4, observation (2) above)} \\
& [\![unfold(\mathcal{E}, X, Y), \mathcal{S}]\!]
\end{aligned}$$

☐

7. Summary – Examples – Related Work

Table 1 summarizes our main results. We will discuss their relevance and compare them to previous work in Section 7.1-7.3. Table 2 contains some other useful facts on FES, discussed in Section 7.4.

7.1. **Substituting Definitions and Solutions.** Theorem 4.3 in this form is new. It generalizes [Mad97, Lemma 6.3] (for BES only) and [GW05a, Lemma 18] (for PBES only) to FES. Another generalization is that we allow that $X = Y$. That is, besides unfolding the Y's in the definition of some X *preceding* Y, one can even unfold Y in its own definition. The proof for this case is more involved (cf. Lemma 4.2). For BES this is useless, but for PBES this is useful, and already used in [OW10, PWW11] to unfold PBESs to BESs. The technique of unfolding PBESs is perhaps the most commonly applied method of solving PBESs [FAAKS24, KFG20, PWW11], although symbolic approaches do exist [KNIU19, NWG20].

Theorem 6.10 is a new result, generalizing the case where $indep(\mathcal{E}, Y, X)$ for all X (i.e. Y is in solved form, [Mad97, GW05a]). In general, one cannot unfold Y in the definition of X, when Y precedes X. However, if there is no dependency path from Y to X, then a forward substitution is allowed. The proof is based on clever reordering of equations. The following example shows that this condition is necessary:

Example 7.1. Consider the following two Boolean Equation Systems:

B_1	B_2
$\nu Y = X$ $\mu X = Y$	$\nu Y = X$ $\mu X = X$

Unfolding Y in the definition of X in B_1 yields B_2. However, B_1 has the solution $(\top, \top)$, while the solution of B_2 is $(\bot, \bot)$. The reader can check this with the method described in Example 7.2.

Theorem 4.4 allows to substitute a partial solution in a FES. It occurs already in [Mad97, Lemma 3.19]. However, our proof is more direct. Mader suggests that a direct inductive argument is not possible, and proves the theorem by contradiction, constructing an infinite set of equation systems. We show that with an appropriate induction loading, the theorem can be reduced to another lemma in fixpoint calculus (Lemma 2.1.8).

The substitution theorems form the basis for solving BES and PBES by Gauss-elimination. They are called the global steps. Besides global steps, one needs local steps, to eliminate X from the right hand side of its own definition. For BES, a local step is trivial, because (only) in the Boolean lattice we have $\mu X.f(X) = f(\bot)$ and $\nu X.f(X) = f(\top)$. Local solution for PBES is much harder, and studied in [GW05a, OW10]. We stress that our results show that the global steps hold in any FES. However, effective local solution is specific to the underlying complete lattice.

Example 7.2. The following example shows the solution of a BES by Gauss elimination. Basically, one first substitutes definitions backwards using Theorem 4.3 (along the way, we use the identity $Y \vee (Y \wedge X) \equiv Y$):

$$\boxed{\begin{array}{lcl} \mu X & = & Y \vee Z \\ \nu Y & = & Z \\ \mu Z & = & Y \wedge X \end{array}} \rightarrow \boxed{\begin{array}{lcl} \mu X & = & Y \\ \nu Y & = & Y \wedge X \\ \mu Z & = & Y \wedge X \end{array}} \rightarrow \boxed{\begin{array}{lcl} \mu X & = & Y \wedge X \\ \nu Y & = & Y \wedge X \\ \mu Z & = & Y \wedge X \end{array}}$$

Next, one obtains $X = \bot$ by a local elimination step in the first equation, using that $Y \wedge \bot \equiv \bot$. This solution can then be substituted forward by Theorem 6.10, to obtain the full solution $(\bot, \bot, \bot)$. In general, steps 1 and 2 must be mixed.

The next example shows a PBES where unfolding X in its own definition makes sense.

Thm	**(In)Equality**	**Conditions**
Reordering Variables		
5.1	$[\![\mathcal{E},\mathcal{S}_1;\sigma X;\sigma Y;\mathcal{S}_2]\!] = [\![\mathcal{E},\mathcal{S}_1;\sigma Y;\sigma X;\mathcal{S}_2]\!]$	- $\mathcal{E}$ is monotonic
5.9	$[\![\mathcal{E},\mathcal{S}_1;\mu X;\nu Y;\mathcal{S}_2]\!] \leq [\![\mathcal{E},\mathcal{S}_1;\nu Y;\mu X;\mathcal{S}_2]\!]$	- $\mathcal{E}$ is monotonic - $X \neq Y$
6.9	$[\![\mathcal{E},\mathcal{S}_1;\sigma X;\rho Y;\mathcal{S}_2]\!] = [\![\mathcal{E},\mathcal{S}_1;\rho Y;\sigma X;\mathcal{S}_2]\!]$	- $X \not\twoheadrightarrow Y$ in $(\mathcal{E},\sigma X;\rho Y;\mathcal{S}_2)$
5.7	$[\![\mathcal{E},\mathcal{S}_0;\mathcal{S}_1;\mathcal{S}_2;\mathcal{S}_3]\!] = [\![\mathcal{E},\mathcal{S}_0;\mathcal{S}_2;\mathcal{S}_1;\mathcal{S}_3]\!]$	- $\mathit{disjoint}(\mathcal{S}_1,\mathcal{S}_2;\mathcal{S}_3)$ - either $\mathit{indep}(\mathcal{E},\mathcal{S}_1,\mathcal{S}_2;\mathcal{S}_3)$, or $\mathit{indep}(\mathcal{E},\mathcal{S}_2;\mathcal{S}_3,\mathcal{S}_1)$
Substituting Definitions and Solutions		
4.3	$[\![\mathit{unfold}(\mathcal{E},X,Y),\mathcal{S}]\!] = [\![\mathcal{E},\mathcal{S}]\!]$ ($X = Y$ is allowed)	- $\mathcal{E}$ is monotonic - $\mathcal{S} = \mathcal{S}_1;\sigma Y;\mathcal{S}_2$ - $X \notin \mathit{dom}(\mathcal{S}_2)$
6.10	$[\![\mathit{unfold}(\mathcal{E},X,Y),\mathcal{S}]\!] = [\![\mathcal{E},\mathcal{S}]\!]$	- $\mathcal{E}$ is monotonic - $\mathcal{S} = \mathcal{S}_1;\sigma Y;\mathcal{S}_2$ - $Y \not\twoheadrightarrow X$ in $(\mathcal{E},\mathcal{S})$
4.4	$[\![\mathcal{E},\mathcal{S}]\!](\eta) = [\![\mathcal{E}[X \mapsto A],\mathcal{S}]\!](\eta)$	- $\mathcal{E}$ is monotonic - $A = [\![\mathcal{E},\mathcal{S}]\!](\eta)(X)$
Swapping Signs		
5.10	$[\![\mathcal{E},\mathcal{S}_1;\mu X;\mathcal{S}_2]\!] \leq [\![\mathcal{E},\mathcal{S}_1;\nu X;\mathcal{S}_2]\!]$	- $\mathcal{E}$ is monotonic
6.7	$[\![\mathcal{E},\mathcal{S}_1;\mu X;\mathcal{S}_2]\!] = [\![\mathcal{E},\mathcal{S}_1;\nu X;\mathcal{S}_2]\!]$	- $X \notin \mathit{dom}(\mathcal{S}_2)$ - $X \not\to^{+} X$ in $(\mathcal{E},\mu X;\mathcal{S}_2)$

Table 1: Main results for arbitrary FES

Example 7.3. Applying Theorem 4.3 to unfold X in its own definition, we get:

$$\boxed{\begin{array}{rcl} \nu Y &=& X(\top) \\ \mu X(b) &=& (b \wedge Y) \vee X(\neg b) \end{array}} \rightarrow \boxed{\begin{array}{rcl} \nu Y &=& X(\top) \\ \mu X(b) &=& (b \wedge Y) \vee ((\neg b \wedge Y) \vee X(\neg\neg b)) \\ &\equiv& Y \vee X(b) \end{array}}$$

Applying Theorem 4.3 again, to unfold X in Y yields $\nu Y = Y \vee X(\top)$, hence by local resolution $Y = \top$, hence $X(b) = \top$.

7.2. **Reordering Variables.** Theorem 5.1 indicates that two adjacent variables with the same sign may be interchanged. This theorem occurs already in [Mad97, Lemma 3.21]. For PBES it is repeated in [GW05a, Lemma 21]. However, [Mad97, GW05a] don't give a full proof, but refer to Bekič Lemma. In our proof, we show exactly how Theorem 4.3 reduces to our version of Bekič Rule (Lemma 2.1.9). In other works [Sei96, KNIU19], adjacent variables with the same sign are grouped in unordered *blocks*. No claim is made about the correctness of such a definition.

Theorem 5.9 shows the inequality that arises when interchanging adjacent variables with different sign. It occurs already in [Mad97, Lemma 3.23], but our proof is different. Our proof depends on a probably new inequality in fixpoint calculus, which we coin Bekič Inequality (Lemma 2.2.4).

Theorem 6.9 is a new result. It states that in the special case that X and Y are not on the same dependency loop, they can be interchanged without modifying the solution.

This generalizes [GW05a, Lemma 19], which requires that the right-hand side of Y in $\mathcal{E}$ is a constant, *i.e.*, $\mathit{indep}(\mathcal{E}, Y, Z)$ for all Z.

Finally, Theorem 5.7 in this form is new. It allows to swap whole blocks of equations. Mader [Mad97, Lemma 3.22] claims a similar result, under the condition that both $\mathit{indep}(\mathcal{E}, \mathcal{S}_1, \mathcal{S}_2)$ and $\mathit{indep}(\mathcal{E}, \mathcal{S}_2, \mathcal{S}_1)$. However, [GW05a] show a counter example to this. The repair in [GW05a, Lemma 22] requires that $\mathcal{S}_3$ is empty. We show a stronger result: if $\mathcal{S}_3$ is empty, only one of the requirements $\mathit{indep}(\mathcal{S}_2, \mathcal{S}_1)$ or $\mathit{indep}(\mathcal{S}_2, \mathcal{S}_1)$ is needed.

Notably, our result even applies to nonempty $\mathcal{S}_3$, provided we have $\mathit{indep}(\mathcal{S}_1, \mathcal{S}_2;\mathcal{S}_3)$ (or its reverse), i.e. $\mathcal{S}_1$ is also independent on the variables in $\mathcal{S}_3$. Note that we allow arbitrary (dependent) alternations within $\mathcal{S}_1$ and $\mathcal{S}_2$, and even $\mathcal{S}_2$ might depend on $\mathcal{S}_1$. We lifted two other unnecessary restrictions: surprisingly, this result doesn't require monotonicity of $\mathcal{E}$. Also, the results in [Mad97, GW05a] are for individual equations only, while we can swap whole blocks at the same time.

We now show an application of swapping blocks to reduce the number of μ/ν-alternations.

Example 7.4. Consider the following four Boolean Equation Systems:

B_3	B_4	B_5	B_6
$\mu X = Y$	$\mu X = Y$	$\nu Z = W$	$\mu X = Y$
$\mu Y = X$	$\nu Z = W$	$\mu X = Y$	$\mu Y = X$
$\nu Z = W$	$\mu Y = X$	$\mu Y = X$	$\mu W = Z$
$\mu W = Z$	$\mu W = Z$	$\mu W = Z$	$\nu Z = W$

For these BES, the dependency graph between the variables consists of two loops, $X \leftrightarrow Y$ and $Z \leftrightarrow W$. In particular, we have $\mathit{indep}(\{X, Y\}, \{Z, W\})$. We want to transform B_3 to B_5, because it has fewer alternations. Theorem 5.1 cannot be applied, because the sign of Z is different from all the others.

Using Theorem 5.7 on individual equations, one can show that $[\![B_3]\!] = [\![B_4]\!]$, because $\mathit{indep}(Y, \{Z, W\})$. However, one cannot derive $[\![B_4]\!] = [\![B_5]\!]$ using Theorem 5.7, because neither $\mathit{indep}(X, \{Z, Y, W\})$, nor $\mathit{indep}(\{Z, Y, W\}, X)$ holds. However, one can prove $B_3 = B_5$ directly with Theorem 5.7, by swapping block $[X, Y]$ with Z, because indeed we have $\mathit{indep}(\{X, Y\}, \{Z, W\})$. Alternatively, one can observe that $X \not\twoheadrightarrow Z$, and apply Theorem 6.9 to deduce that $B_4 = B_5$ directly.

All theorems fail to prove the equivalence of B_{3-5} with B_6. However, Theorem 5.9 guarantees that $[\![B_6]\!] \leq [\![B_3]\!]$. As a matter of fact, the solution of B_3, B_4 and B_5 is $(X = \bot, Y = \bot, Z = \top, W = \top)$, while the solution of B_6 is $(X = \bot, Y = \bot, Z = \bot, W = \bot)$. The reader may verify this by Gauss Elimination, cf. Example 7.2. This shows that the reordering theorems cannot easily be strengthened.

7.3. **Swapping Signs.** The inequality of Theorem 5.10 is well known and appears for instance in [Mad97, Lemma 3.24]. Theorem 6.7 is new. It shows that the sign of variable X is only relevant when X is the most relevant variable on a dependency loop.

Example 7.5. Now consider the next three BESs which only differ in their fixpoint signs:

B_7	B_8	B_9
$\mu X = Y$	$\mu X = Y$	$\mu X = Y$
$\nu Y = X \vee Z$	$\nu Y = X \vee Z$	$\mu Y = X \vee Z$
$\mu Z = Z \wedge W$	$\nu Z = Z \wedge W$	$\mu Z = Z \wedge W$
$\nu W = X \wedge \bot$	$\nu W = X \wedge \bot$	$\mu W = X \wedge \bot$

Lem	**Result**	**Condition**
3.8	$[\![\mathcal{E},\mathcal{S}]\!](\eta)(X) = \mathcal{E}_X([\![\mathcal{E},\mathcal{S}]\!](\eta))$	- $\mathcal{E}$ monotonic - $\mathcal{X} \in \mathit{dom}(\mathcal{S})$
5.3	$[\![\mathcal{E},\mathcal{S}_1;\mathcal{S}_2]\!](\eta) = [\![\mathcal{E},\mathcal{S}_2]\!]([\![\mathcal{E},\mathcal{S}_1]\!](\eta))$	- $\mathit{indep}(\mathcal{E},\mathcal{S}_1,\mathcal{S}_2)$
5.6	$[\![\mathcal{E},\mathcal{S}_1;\mathcal{S}_2]\!](\eta) = [\![\mathcal{E},\mathcal{S}_1]\!]([\![\mathcal{E},\mathcal{S}_2]\!](\eta))$	- $\mathit{indep}(\mathcal{E},\mathcal{S}_2,\mathcal{S}_1)$ - $\mathit{disjoint}(\mathcal{S}_1,\mathcal{S}_2)$
5.8	$[\![\mathcal{E},\mathcal{S}_1]\!] \leq [\![\mathcal{E},\mathcal{S}_2]\!]$ implies $[\![\mathcal{E},\mathcal{S};\mathcal{S}_1]\!] \leq [\![\mathcal{E},\mathcal{S};\mathcal{S}_2]\!]$	- $\mathcal{E}$ monotonic
3.9	$[\![\mathcal{E},\mathcal{S}_1]\!] = [\![\mathcal{E},\mathcal{S}_2]\!]$ implies $[\![\mathcal{E},\mathcal{S};\mathcal{S}_1]\!] = [\![\mathcal{E},\mathcal{S};\mathcal{S}_2]\!]$	
5.5	$[\![\mathcal{E},\mathcal{S}_1]\!] = [\![\mathcal{E},\mathcal{S}_2]\!]$ implies $[\![\mathcal{E},\mathcal{S}_1;\mathcal{S}]\!] = [\![\mathcal{E},\mathcal{S}_2;\mathcal{S}]\!]$	- $\mathit{indep}(\mathcal{E},\mathcal{S}_1,\mathcal{S})$ - $\mathit{indep}(\mathcal{E},\mathcal{S}_2,\mathcal{S})$ - $\mathit{disjoint}(\mathcal{S},\mathcal{S}_1;\mathcal{S}_2)$

Table 2: Some useful lemmas for arbitrary FES

Like before, want to manipulate B_7 to reduce the number of fixpoint alternations. We identify two possibilities. First, we may flip the sign of Z, obtaining B_8, which has the solution $(\top, \top, \bot, \bot)$. By Theorem 5.10, it holds that $[\![B_7]\!] \leq [\![B_8]\!]$ and so we conclude that also $[\![B_7]\!](Z) = [\![B_7]\!](W) = \bot$. The other option is to flip the sign of Y and W, yielding B_9. Since $Y \not\to^+ Y$ in Y, Z, W, Theorem 6.7 gives that $[\![B_7]\!] = [\![B_9]\!]$.

7.4. **Other Results.** Along the way, we proved (and proof checked) several lemmas on FES that may be interesting on their own. For quick reference, we summarize these in Table 2. Here η denotes an arbitrary valuation. All these lemmas occur in some form in the literature. Lemma 3.8 states that the semantics indeed returns a solution, and follows from [Mad97, Lemma 3.5]. Lemma 5.3 corresponds to [Mad97, Lemma 3.10] (which is not proved there) and [GW05a, Lemma 7]. Actually, [Mad97] has Lemma 5.6, which is equivalent according to our Theorem 5.7. Lemma 3.9 and 5.8 are from [Mad97, Lemma 3.14] as well, and Lemma 5.5 follows directly from Lemma 5.3. We included it here to stress that right congruence doesn't hold in general.

Finally, we needed some basic results on fixpoints in complete lattices, cf. Lemma 2.1 and 2.2. The existence and definition of least and greatest fixpoints is due to Knaster (on sets) and Tarski (on complete lattices) [Tar55], see [LNS82] for a historical account. We (re)proved a number of identities (Lemma 2.1) and inequalities (Lemma 2.2) on fixpoint expressions. Most of these results are known. Lemma 2.1.1-6 can for instance be found in [Bac02]. Rule 9 (Bekič Equality) can be found in e.g. [dB80, Bek84], but stated in a different form, involving simultaneous fixpoints. We have not found in the literature the inequality in Lemma 2.2.4, which resembles Bekič equality on terms with mixed minimal and maximal fixpoints.

8. Formalisation in Coq & PVS

We have formalised all of the above theory in both Coq [Ber08, S$^+$23] and PVS [OS08]. A replication artefact containing these proofs is available at [NvdP24]. The formalized definitions and proofs follow the definitions and proof steps in this paper quite closely. Here, we highlight the main difference between the two formalisations.

In Coq, we captured the concepts of complete lattices and monotonic functions in typeclasses. For these, we defined several typeclass instances, for example the product lattice and composition of monotonic functions. In many cases, Coq is able to perform automatic typeclass resolution, saving us from manually proving monotonicity of complex functions, for example those in Lemma 2. Furthermore, Coq supports user-defined notation, allowing us to closely follow the notation used in the paper. The proofs for showing decidability of $X \xrightarrow{\mathcal{E},\mathcal{S}}\mathrel{\mkern-14mu}\rightarrow Y$ are extensive, something that is not reflected in the paper.

Our PVS definitions and proofs were originally developed under PVS version 4.2, but could be ported to version 7.1 with minimal effort. Contrary to Coq, PVS is built on classical logic and thus allows the law of excluded middle (for all propositions P, it holds $P \vee \neg P$). We thus do not need to supply proofs for decidability of $X \xrightarrow{\mathcal{E},\mathcal{S}}\mathrel{\mkern-14mu}\rightarrow Y$. This also means that we do not rely on finiteness of $\mathcal{S}$, and thus the definition of $\twoheadrightarrow$ only depends on $\mathcal{E}$ and the domain $\mathcal{E}$ is restricted where necessary, *e.g.*, in Theorem 6.7. This simplifies the proof of Lemma 6.2: it can operate on $\mathit{split}_{X,\mathcal{E}}$ directly.

9. Conclusion

We provided several equalities and inequalities involving a range of operations on fixpoint equation systems (FES). We refer to Table 1 and 2 (Section 7) for a summary of the theorems. Lemmas 2.1 and 2.2 provide a useful overview on equalities and inequalities for nested fixed points in complete lattices.

We provided self-contained and detailed proofs of all results and mechanised these proofs in two proof assistants, Coq and PVS.

By the generic nature of FES, these results carry over to other formalisms such as Boolean equation systems (BES), parity games (and variations thereof), and parameterised (first-order) Boolean equation systems (PBES).

Acknowledgment

Large part of the research was carried out at the Centrum voor Wiskunde en Informatica.

References

[And94] Henrik Reif Andersen. Model checking and Boolean graphs. *Theoretical Computer Science*, 126(1):3–30, April 1994. `doi:10.1016/0304-3975(94)90266-6`.

[AV95] Henrik Reif Andersen and Bart Vergauwen. Efficient checking of behavioural relations and modal assertions using fixed-point inversion. In P. Wolper, editor, *CAV 1995*, volume 939 of *LNCS*, pages 142–154. Springer, 1995. `doi:10.1007/3-540-60045-0_47`.

[Bac02] Roland Backhouse. Galois connections and fixed point calculus. In R. Backhouse, R. Crole, and J. Gibbons, editors, *Algebraic and coalgebraic methods in the mathematics of program construction*, volume 2297 of *LNCS*, pages 89–150. Springer, 2002. `doi:10.1007/3-540-47797-7_4`.

[Bek84] Hans Bekič. Definable operation in general algebras, and the theory of automata and flowcharts. In C.B. Jones, editor, *Programming Languages and Their Definition*, volume 177 of *LNCS*, pages 30–55. Springer, 1984. `doi:10.1007/BFb0048939`.

[Ber08] Yves Bertot. A Short Presentation of Coq. In Otmane Ait Mohamed, César Muñoz, and Sofiène Tahar, editors, *TPHOLs 2008*, volume 5170 of *LNCS*, pages 12–16. Springer, 2008. `doi:10.1007/978-3-540-71067-7_3`.

[BGK$^+$19] Olav Bunte, Jan Friso Groote, Jeroen J. A. Keiren, Maurice Laveaux, Thomas Neele, Erik P. de Vink, Wieger Wesselink, Anton Wijs, and Tim A. C. Willemse. The mCRL2 Toolset for Analysing Concurrent Systems: Improvements in Expressivity and Usability. In *TACAS 2019*, volume 11428 of *LNCS*, pages 21–39, 2019. `doi:10.1007/978-3-030-17465-1_2`.

[BKP20] Paolo Baldan, Barbara König, and Tommaso Padoan. Abstraction, Up-To Techniques and Games for Systems of Fixpoint Equations. In Igor Konnov and Laura Kovács, editors, *CONCUR 2020*, LIPIcs, pages 25:1–25:20, 2020. `doi:10.4230/LIPIcs.CONCUR.2020.25`.

[CD12] Krishnendu Chatterjee and Laurent Doyen. Energy parity games. *Theoretical Computer Science*, 458:49–60, November 2012. `doi:10.1016/j.tcs.2012.07.038`.

[CJK$^+$17] Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasipolynomial time. In *STOC 2017*, pages 252–263. ACM, June 2017. `doi:10.1145/3055399.3055409`.

[CPvdPW07] Taolue Chen, Bas Ploeger, Jaco van de Pol, and Tim A. C. Willemse. Equivalence Checking for Infinite Systems using Parameterized Boolean Equation Systems. In *CONCUR 2007*, volume 4703 of *LNCS*, pages 120–135, 2007. `doi:10.1007/978-3-540-74407-8_9`.

[CS93] Rance Cleaveland and Bernhard Steffen. A linear-time model-checking algorithm for the alternation-free modal mu-calculus. *Formal Methods in System Design*, 2:121–147, 1993. `doi:10.1007/BF01383878`.

[dB80] J.W. de Bakker. *Mathematical Theory of Program Correctness*. Prentice-Hall, 1980.

[EGLS19] Søren Enevoldsen, Kim Guldstrand Larsen, and Jiří Srba. Abstract Dependency Graphs and Their Application to Model Checking. In *TACAS 2019*, volume 11427 of *LNCS*, pages 316–333. Springer, 2019. `doi:10.1007/978-3-030-17462-0_18`.

[EJ91] E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy. In *FOCS 1991*, pages 368–377, 1991. `doi:10.1109/sfcs.1991.185392`.

[FAAKS24] Mahmudul Faisal Al Ameen, Naoki Kobayashi, and Ryosuke Sato. Asynchronous unfold/fold transformation for fixpoint logic. *Science of Computer Programming*, 231:103014, January 2024. `doi:10.1016/j.scico.2023.103014`.

[FL09] Oliver Friedmann and Martin Lange. Solving Parity Games in Practice. In *ATVA 2009*, volume 5799 of *LNCS*, pages 182–196. Springer, 2009. `doi:10.1007/978-3-642-04761-9_15`.

[GK04] Jan Friso Groote and Misa Keinänen. Solving disjunctive/conjunctive boolean equation systems with alternating fixed points. In K. Jensen and A. Podelski, editors, *TACAS 2004*, volume 2988 of *LNCS*, pages 436–450. Springer, 2004. `doi:10.1007/978-3-540-24730-2_33`.

[GLMS13] Hubert Garavel, Frédéric Lang, Radu Mateescu, and Wendelin Serwe. CADP 2011: A Toolbox for the Construction and Analysis of Distributed Processes. *International Journal on Software Tools for Technology Transfer*, 15(2):89–107, 2013. ISBN: 978-3-540-73367-6. `doi:10.1007/978-3-540-73368-3_18`.

[GM99] Jan Friso Groote and Radu Mateescu. Verification of temporal properties of processes in a setting with data. In A.M. Haeberer, editor, *AMAST 1998*, volume 1548 of *LNCS*, pages 74–90. Springer, 1999. `doi:10.1007/3-540-49253-4_8`.

[GW05a] Jan Friso Groote and Tim A.C. Willemse. Parameterised boolean equation systems. *Theoretical Computer Science*, 343:332–369, 2005. `doi:10.1016/j.tcs.2005.06.016`.

[GW05b] J.F. Groote and T.A.C. Willemse. Model-checking processes with data. *Science of Computer Programming*, 56(3):251–273, 2005. `doi:10.1016/j.scico.2004.08.002`.

[GW23] Jan Friso Groote and Tim A. C. Willemse. Real Equation Systems with Alternating Fixed-Points. In *CONCUR 2023*, volume 279 of *LIPIcs*, pages 28:1–28:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. `doi:10.4230/LIPICS.CONCUR.2023.28`.

[HS21] Daniel Hausmann and Lutz Schröder. Quasipolynomial Computation of Nested Fixpoints. In Jan Friso Groote and Kim Guldstrand Larsen, editors, *TACAS 2021*, volume 12651 of *LNCS*, pages 38–56. Springer, 2021. `doi:10.1007/978-3-030-72016-2_3`.

[JMT22] Marcin Jurdziński, Rémi Morvan, and K. S. Thejaswini. Universal Algorithms for Parity Games and Nested Fixpoints. In *Principles of Systems Design*, volume 13660 of *LNCS*, pages 252–271. 2022. `doi:10.1007/978-3-031-22337-2_12`.

[Jur00] Marcin Jurdziński. Small Progress Measures for Solving Parity Games. In *STACS 2000*, volume 1770 of *LNCS*, pages 290–301. Springer, 2000. `doi:10.1007/3-540-46541-3_24`.

[KFG20] Naoki Kobayashi, Grigory Fedyukovich, and Aarti Gupta. Fold/Unfold Transformations for Fixpoint Logic. In *TACAS 2020*, volume 12079 of *LNCS*, pages 195–214, 2020. `doi:10.1007/978-3-030-45237-7_12`.

[KNIU19] Naoki Kobayashi, Takeshi Nishikawa, Atsushi Igarashi, and Hiroshi Unno. Temporal Verification of Programs via First-Order Fixpoint Logic. In Bor-Yuh Evan Chang, editor, *SAS 2019*, volume 11822 of *LNCS*, pages 413–436. Springer, 2019. `doi:10.1007/978-3-030-32304-2_20`.

[LNS82] J.-L. Lassez, V. L. Nguyen, and E. A. Sonenberg. Fixed point theorems and semantics: A folk tale. *Information Processing Letters*, 14(3):112–116, May 1982. `doi:10.1016/0020-0190(82)90065-5`.

[Mad97] Angelika Mader. *Verification of Modal Properties Using Boolean Equation Systems*. PhD thesis, Technische Universität München, 1997.

[Mat98] Radu Mateescu. *Vérification des propriétés temporelles des programmes parallèles*. PhD thesis, Institut National Polytechnique de Grenoble - INPG, April 1998.

[Mat06] Radu Mateescu. CAESAR_SOLVE: A generic library for on-the-fly resolution of alternation-free boolean equation systems. *International Journal on Software Tools for Technology Transfer*, 8(1):37–56, February 2006. `doi:10.1007/s10009-005-0194-9`.

[MS03] Radu Mateescu and Mihaela Sighireanu. Efficient on-the-fly model-checking for regular alternation-free mu-calculus. *Science of Computer Programming*, 46(3):255–281, March 2003. `doi:10.1016/S0167-6423(02)00094-1`.

[Nee22] Thomas Neele. (Re)moving Quantifiers to Simplify Parameterised Boolean Equation Systems. In *ARQNL 2022*, volume 3326 of *CEUR Workshop Proceedings*, pages 64–80. CEUR-WS.org, 2022.

[NvdP24] Thomas Neele and Jaco van de Pol. Replication package with proofs for the paper "Operations on Fixpoint Equation Systems", February 2024. `doi:10.5281/zenodo.10640564`.

[NWG20] Thomas Neele, Tim A. C. Willemse, and Jan Friso Groote. Finding Compact Proofs for Infinite-Data Parameterised Boolean Equation Systems. *Science of Computer Programming*, 188:102389, 2020. `doi:10.1016/j.scico.2019.102389`.

[NWWV22] Thomas Neele, Tim A. C. Willemse, Wieger Wesselink, and Antti Valmari. Partial-order reduction for parity games and parameterised Boolean equation systems. *International Journal on Software Tools for Technology Transfer*, 24(5):735–756, October 2022. `doi:10.1007/s10009-022-00672-0`.

[OS08] Sam Owre and Natarajan Shankar. A Brief Overview of PVS. In Otmane Ait Mohamed, César Muñoz, and Sofiène Tahar, editors, *TPHOLs 2008*, volume 5170 of *LNCS*, pages 22–27. Springer, 2008. `doi:10.1007/978-3-540-71067-7_5`.

[OW10] Simona Orzan and Tim A. C. Willemse. Invariants for Parameterised Boolean Equation Systems. *Theoretical Computer Science*, 411(11-13):1338–1371, 2010. `doi:10.1016/j.tcs.2009.11.001`.

[PWW11] B. Ploeger, J. W. Wesselink, and T. A. C. Willemse. Verification of reactive systems via instantiation of Parameterised Boolean Equation Systems. *Information and Computation*, 209(4):637–663, 2011. `doi:10.1016/j.ic.2010.11.025`.

[S^+23] Matthieu Sozeau et al. The Coq Proof Assistant, June 2023. `doi:10.5281/zenodo.8161141`.

[Sei96] Helmut Seidl. Fast and simple nested fixpoints. *Information Processing Letters*, 59(6):303–308, September 1996. `doi:10.1016/0020-0190(96)00130-5`.

[Tar55] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309, June 1955. `doi:10.2140/pjm.1955.5.285`.

[TC02] Li Tan and R. Cleaveland. Evidence-based model checking. In E. Brinksma and K.G. Larsen, editors, *CAV 2002*, volume 2404 of *LNCS*, pages 455–470. Springer, 2002. `doi:10.1007/3-540-45657-0_37`.

[vD18] Tom van Dijk. Oink: An Implementation and Evaluation of Modern Parity Game Solvers. In Marieke Huisman and Dirk Beyer, editors, *TACAS 2018*, volume 10805 of *LNCS*, pages 291–308. Springer, 2018. `doi:10.1007/978-3-319-89960-2_16`.

[ZC05] Dezhuang Zhang and Rance Cleaveland. Fast generic model-checking for data-based systems. In F. Wang, editor, *FORTE 2005*, volume 3731 of *LNCS*, pages 83–97. Springer, 2005. `doi:10.1007/11562436_8`.