Some notes on a BES library

Jeroen Keiren

15th September 2023

1 Boolean Equation Systems

Boolean Equation Systems (BESs) [**?**] are a class of equation system that can be employed to perform model checking of modal μ-calculus formulae. Basically, BESs are finite sequences of least and greatest fixpoint equations, where each right-hand side of an equation is a proposition in positive form. It has been shown [**?**] that solving a BES is equivalent to the model-checking problem. BESs are used for this purpose in *e.g.* the tool sets CADP [**?**] and mCRL2 [**?**]. Several algorithms for solving BESs exist, see [**?**, **?**]. Furthermore there are efficient algorithms for some special cases, see [**?**, **?**]. We formally introduce the theory required for understanding the results obtained in this paper.

Definition 1.1. We assume a set $\mathcal{X}$ of Boolean variables, with typical elements X, X_1, X_2, ... and a type $\mathbb{B}$ with elements true, false representing the Booleans. Furthermore we have fixpoint symbols μ for least fixpoint and ν for greatest fixpoint.

A Boolean Equation System is a system of fixpoint equations, inductively defined as follows:

- ϵ is the empty BES
- if $\mathcal{E}$ is a BES, then $(\sigma X = f)\mathcal{E}$ is also a BES, with $\sigma \in \{\mu, \nu\}$ a fixpoint symbol and f a formula over $\mathcal{X}$, defined by the following grammar:

$$f, g ::= c \mid X \mid \neg f \mid f \wedge g \mid f \vee g \mid f \Rightarrow g$$

  *** Do we also need to include $if(f, g, h)$ as expression; it is available in the deprecated BES header? Its use is in the construction of BDD structures for the comparison of formulae. Note that this could also be encoded as $(f \Rightarrow g) \wedge (\neg f \Rightarrow g)$ ***

  where $X \in \mathcal{X}$ is a proposition variable of type $\mathbb{B}$ and $c \in \{\mathsf{true}, \mathsf{false}\}$ is a Boolean constant. We refer to the set of formulae over $\mathcal{X}$ with $\mathcal{F}_{\mathcal{X}}$.

For any equation system $\mathcal{E}$, the set of *bound proposition variables*, $\mathsf{bnd}(\mathcal{E})$, is the set of variables occurring at the left-hand side of some equation in $\mathcal{E}$. The set of *occurring proposition variables*, $\mathsf{occ}(\mathcal{E})$, is the set of variables occurring at the right-hand side of some equation in $\mathcal{E}$.

$$\mathsf{bnd}(\epsilon) \triangleq \emptyset \qquad \mathsf{bnd}((\sigma X = f)\ \mathcal{E}) \triangleq \mathsf{bnd}(\mathcal{E}) \cup \{X\}$$

$$\mathsf{occ}(\epsilon) \triangleq \emptyset \qquad \mathsf{occ}((\sigma X = f)\ \mathcal{E}) \triangleq \mathsf{occ}(\mathcal{E}) \cup \mathsf{occ}(f)$$

where $\mathsf{occ}(f)$ is defined inductively as follows:

$$\mathsf{occ}(c) \triangleq \emptyset \qquad \mathsf{occ}(X) \triangleq \{X\}$$
$$\mathsf{occ}(\neg f) \triangleq \mathsf{occ}(f) \qquad \mathsf{occ}(f \Rightarrow g) \triangleq \mathsf{occ}(f) \cup \mathsf{occ}(g)$$
$$\mathsf{occ}(f \vee g) \triangleq \mathsf{occ}(f) \cup \mathsf{occ}(g) \qquad \mathsf{occ}(f \wedge g) \triangleq \mathsf{occ}(f) \cup \mathsf{occ}(g)$$

BESs $\mathcal{E}$ and $\mathcal{F}$ with $\mathsf{bnd}(\mathcal{E}) \cap \mathsf{bnd}(\mathcal{F}) = \emptyset$ are referred to as *non-conflicting* BESs.

As usual, we consider only equation systems $\mathcal{E}$ in which every proposition variable occurs at the left-hand side of at most one equation of $\mathcal{E}$. We define an ordering $\triangleleft$ on bound variables of an equation system $\mathcal{E}$, where $X \triangleleft X'$ indicates that the equation for X precedes the equation for X'.

Proposition formulae are interpreted in a context of an *environment* $\eta{:}\mathcal{X} \to \mathbb{B}$. For an arbitrary environment η, we write $\eta[X := b]$ for the environment η in which the proposition variable X has Boolean value b.

Finding a solution of a BES amounts to finding an assignment of true or false to each variable X_i such that all equations are satisfied. Furthermore if $\sigma_i = \mu$, then the assignment to X_i is as strong as possible, and if $\sigma_i = \nu$ it is as weak as possible, where the leftmost equation takes priority over equations that follow. The concept of a solution is formalised below.

Definition 1.2. Let $\eta{:}\mathcal{X} \to \mathbb{B}$ be an environment. The *interpretation* $[\![f]\!]\eta$ maps a proposition formula f to true or false:

$$[\![c]\!]\eta \triangleq c \qquad [\![X]\!]\eta \triangleq \eta(X)$$
$$[\![f \vee g]\!]\eta \triangleq [\![f]\!]\eta \vee [\![g]\!]\eta \qquad [\![f \wedge g]\!]\eta \triangleq [\![f]\!]\eta \wedge [\![g]\!]\eta$$

Let η be an environment. Let $b_\mu = \mathsf{false}$ and $b_\nu = \mathsf{true}$. The *solution* of a BES, given η, is inductively defined as follows:

$$\begin{aligned} [\![\epsilon]\!]\eta &= \eta \\ [\![(\sigma X = f)\mathcal{E}]\!]\eta &= [\![\mathcal{E}]\!]\eta[X := f([\![\mathcal{E}]\!]\eta[X := b_\sigma])] \end{aligned}$$

We also write $\eta(X)$ to denote the interpretation of X in environment η. In the sequel, when we refer to solving a BES we mean computing the solution of the BES.

We introduce the following terminology.

Definition 1.3. Let $\mathcal{E}$ be an equation system. Then

- $\mathcal{E}$ is *closed* whenever $\mathsf{occ}(\mathcal{E}) \subseteq \mathsf{bnd}(\mathcal{E})$;
- $\mathcal{E}$ is *solved* whenever $\mathsf{occ}(\mathcal{E}) = \emptyset$;

For closed BES $\mathcal{E}$, $[\![\mathcal{E}]\!]\eta = [\![\mathcal{E}]\!]\eta'$ for arbitrary environments η and η', hence we may omit the environment in this case. Also observe that according to the semantics $\wedge$ and $\vee$ are commutative and associative, hence we may write *e.g.* $\bigwedge_{i=0}^{j} f_i$ instead of $f_0 \wedge \ldots f_n$, for formulae f_i.

For a closed BES $\mathcal{E}$ we define the right hand side rhs of a propositional variable $X \in \mathsf{bnd}(\mathcal{E})$ as the *right hand side* of the defining equation of X in $\mathcal{E}$:

$$\mathsf{rhs}(X, (\sigma Y = f)\mathcal{E}) \triangleq \begin{cases} f & \text{if } X = Y \\ \mathsf{rhs}(X, \mathcal{E}) & \text{otherwise} \end{cases}$$

Besides assignments we can also define the notion of substitution on a BES.

Definition 1.4. A substitution $\sigma{:}\mathcal{X} \to \mathcal{F}_{\mathcal{X}}$ is a function assigning a boolean expression to a variable. We define application of a substitution σ to a boolean expression inductively as follows:

$$\sigma(c) \triangleq c \qquad \sigma(X) \triangleq \begin{cases} Y & \text{if } X := Y \text{ in } \sigma \\ X & \text{otherwise} \end{cases}$$

$$\sigma(\neg f) \triangleq \neg\sigma(f) \qquad \sigma(f \Rightarrow g) \triangleq \sigma(f) \Rightarrow \sigma(g)$$

$$\sigma(f \vee g) \triangleq \sigma(f) \vee \sigma(g) \qquad \sigma(f \wedge g) \triangleq \sigma(f) \wedge \sigma(g)$$

We now introduce some restricted BES formats which exhibit some interesting theoretic properties.

Definition 1.5. A closed BES $\mathcal{E}$ is in simple form (SF) if every equation in $\mathcal{E}$ is of the form $\sigma X = f$, $\sigma X = \bigwedge_{i=0}^{n} f_i$ or $\sigma X = \bigvee_{i=0}^{n} f_i$, where $n > 0$, and f is either a propositional variable, or one of the Boolean constants true or false.

That is, a closed BES is in simple form if every right hand side is either a single variable or Boolean constant, or it is a conjunction or a disjunction over propositional variables or Boolean constants. Conjunctions and disjunctions may not appear mixed in a single right hand side. Note that every closed BES can be transformed into simple form in polynomial time in such a way that the variables in the original BES are preserved, and variables that occur in both BESs have the same solution, see [**?**]. An equation can, for example, be transformed to simple form as follows. Given an equation $\sigma X = \bigwedge_{i=0}^{k} f_i$, and some f_j is disjunctive, replace this single equation by two equations $(\sigma X = \bigwedge_{i=0}^{j-1} \wedge X' \bigwedge_{i=j+1}^{k})(\sigma X' = f_j)$, where X' is fresh. The case for $\vee$ is analogous, and the transformation can be repeated until a BES in simple form is obtained.

We can also restrict a BES such that it does not contain Boolean constants. This is referred to as recursive form.

Definition 1.6. A closed BES $\mathcal{E}$ is in recursive form (RF) if the Boolean constants true and false do not occur in $\mathcal{E}$.

The transformation of a BES to a BES in RF can also be done in a solution preserving way, introducing auxiliary equations for Boolean constants true and false.

When we combine the notions of simple form and recursive form we obtain the concept of simple recursive form.

Definition 1.7. A closed BES $\mathcal{E}$ is in simple recursive form (SRF) if $\mathcal{E}$ is in simple form, and the Boolean constants true and false do not occur in $\mathcal{E}$.

The translation of a BES to SRF is simply the composition of the translations of a BES to SF and RF, and hence is also solution preserving.

Definition 1.8. A BES $\mathcal{E}$ is in conjunctive form if every equation in $\mathcal{E}$ is of the form $\sigma X = \bigwedge_{i=0}^{n} f_i$, with $n \geqslant 0$, and f_i a propositional variable or a Boolean constant.

That is, a BES in conjunctive form only contains conjuncts, single variables or Boolean constants as right hand sides. It has been shown [**?**] that given a BES $\mathcal{E}$ and an environment η there is a BES $\mathcal{E}'$ in conjunctive form such that $\mathcal{E}$ and $\mathcal{E}'$ have the same solutions in η.

A similar notion is a BES in disjunctive form, *i.e.* a BES that only contains disjuncts, single variables or Boolean constants as right hand sides.

Definition 1.9. A BES $\mathcal{E}$ is in disjunctive form if every equation in $\mathcal{E}$ is of the form $\sigma X = \bigvee_{i=0}^{n} f_i$, with $n \geqslant 0$, and f_i a propositional variable or Boolean constant.

A derived notion of a closed equation system $\mathcal{E}$ is its *dependency graph* $\mathcal{G}_{\mathcal{E}}$, which is defined as a structure $\langle V, \rightarrow \rangle$, where:

- $V = \mathsf{bnd}(\mathcal{E})$;
- $X \rightarrow Y$ iff there is some equation $\sigma X = f$ in $\mathcal{E}$ with $Y \in \mathsf{occ}(f)$;

We introduce the notion of rank of an equation, and some derived notions. These notions are an indication of the complexity of the BES, as well as a measure that occurs in the computational complexity of some of the algorithms for solving BESs.

Definition 1.10. Let $\mathcal{E}$ be an arbitrary equation system. The *rank* of some $X \in \mathsf{bnd}(\mathcal{E})$, denoted $\mathsf{rank}(X)$, is defined as $\mathsf{rank}(X) = \mathsf{rank}_{\nu,X}(\mathcal{E})$, where $\mathsf{rank}_{\nu,X}(\mathcal{E})$ is defined inductively as follows:

$$\mathsf{rank}_{\sigma,X}(\epsilon) = 0$$

$$\mathsf{rank}_{\sigma,X}((\sigma' Y = f)\mathcal{E}) = \begin{cases} 0 & \text{if } \sigma = \sigma' \text{ and } X = Y \\ \mathsf{rank}_{\sigma,X}(\mathcal{E}) & \text{if } \sigma = \sigma' \text{ and } X \neq Y \\ 1 + \mathsf{rank}_{\sigma',X}((\sigma' Y = f)\mathcal{E}) & \text{if } \sigma \neq \sigma' \end{cases}$$

Observe that $\mathsf{rank}(X)$ is odd iff X is defined in a least fixpoint equation.

The *alternation hierarchy* of an equation system can be thought of as the number of syntactic alternations of fixpoint signs occurring in the equation system. The alternation hierarchy $\mathsf{ah}(\mathcal{E})$ of an equation system $\mathcal{E}$ can be defined as the difference between the largest and the smallest rank occurring in $\mathcal{E}$, formally $\mathsf{ah}(\mathcal{E}) = \max\{\mathsf{rank}(X) \mid X \in \mathsf{bnd}(\mathcal{E})\} - \min\{\mathsf{rank}(X) \mid X \in \mathsf{bnd}(\mathcal{E})\}$.

Given an equation $(\sigma X = f)$ in SF, the function $\mathsf{op}(X)$ returns whether f is conjunctive ($\wedge$), disjunctive ($\vee$) or neither ($\perp$);

An alternative characterisation of the solution of a particular proposition variable X in an equation system $\mathcal{E}$ in SRF is obtained through the use of the dependency graph $\mathcal{G}_{\mathcal{E}}$. We first define the notion of a ν-dominated lasso.

Definition 1.11. Let $\mathcal{E}$ be a closed equation system, and let $\mathcal{G}_{\mathcal{E}}$ be its dependency graph. A *lasso* through $\mathcal{G}_{\mathcal{E}}$, starting in a node X, is a finite path $\langle X_0, X_1, \ldots, X_n \rangle$, satisfying $X_0 = X$, $X_n = X_j$ for some $j \leq n$, and for each $1 < i \leq n$, $X_{i-1} \rightarrow X_i$. A lasso is said to be *ν-dominated* if $\min\{\mathsf{rank}(X_i) \mid j \leq i \leq n\}$ is even; otherwise, it is *μ-dominated*.

The following lemma is loosely based on lemmata taken from Keinänen (see lemmata 40 and 41 in [**?**]).

Lemma 1.12. Let $\mathcal{E}$ be a closed equation system in SRF, and let $\mathcal{G}_\mathcal{E}$ be its dependency graph. Let $X \in \mathsf{bnd}(\mathcal{E})$. Then:

1. If $\mathcal{E}$ is *disjunctive*, then $[\![\mathcal{E}]\!](X) = \mathsf{true}$ iff some lasso starting in X in $\mathcal{G}_\mathcal{E}$ is ν-dominated;

2. If $\mathcal{E}$ is *conjunctive*, then $[\![\mathcal{E}]\!](X) = \mathsf{false}$ iff some lasso starting in X in $\mathcal{G}_\mathcal{E}$ is μ-dominated;

Proof We only consider the first statement; the proof of the second statement is analogous. Observe that when the proposition variable on the cycle of the lasso has an even lowest rank, it is a greatest fixpoint equation $\nu X' = f$, with $X' \triangleleft Y$ for all other equations $\sigma Y = g$ that are on the cycle. This follows from the fact that these have higher ranks. *Gauß elimination* [**?**] allows one to substitute g for Y in the equation for X', yielding $\nu X' = f[Y{:=}g]$. Since, ultimately, X' depends on X' again, this effectively enables one to rewrite $\nu X' = f$ to $\nu X' = f' \vee X'$. The solution to $\nu X' = f' \vee X'$ is easily seen to be $X' = \mathsf{true}$. Since all equations on the lasso are disjunctive, this solution ultimately propagates through the entire lasso, leading to $X = \mathsf{true}$.

Conversely, observe that there is an equation system $\mathcal{E}'$ consisting entirely of equations of the form $\sigma X' = X''$ (follows from Corollary 3.37 in [**?**]), with the additional property that $[\![\mathcal{E}]\!] = [\![\mathcal{E}']\!]$. In $\mathcal{E}'$, the answer to X can only be true if it depends at some point on some $\nu X' = X''$, where ultimately, X'' again depends on X', leading to a cycle in the dependency graph with even lowest rank. □

2 Some requirements for an implementation

2.1 Output formats

For BESs several output formats are available. First of all, we should support a boolean output format (probably ATerm based, like the file format of the PBES library). Furthermore, for compatibility with other tools it is useful to have output in CWI format (a textual format). Furthermore we know that BES in SRF coincide with parity games, hence for this class of equation systems it is useful to be able to output in the format used by the PGSolver toolset.

In the future it could become useful to write part of an already generated BES to disk (in order to save internal memory). Hence a design for a BES library should take this possibility into account.

2.2 Graph interface

As was described before, the notion of dependency graphs for BES in SRF has some nice properties. As an example it is straightforward to implement reduction modulo strong bisimulation (and also some weaker equivalences) on top of this interface (see [**?**] for more details). A more flexible notion of structure graph is also known [**?**], but at the moment it is not clear what the advantages of this notion are in an implementation.

2.3 Flexible storage

It is unclear how formulae in a BES should be stored exactly in an implementation. Sometimes *e.g.* storing in BDD format provides some advantages, whereas in other cases (like BES encoding bisimilarity of to LPSs), the effect is adverse. Hence the design should be sufficiently high-level as to leave a possibility for variation in the underlying format.

3 TODO

- Investigate requirements from bes_deprecated.h