



Process Equivalence Problems as Energy Games

Benjamin Bisping^(✉)

Technische Universität Berlin, Berlin, Germany
benjamin.bisping@tu-berlin.de
<https://bbisping.de>



Abstract. We characterize all common notions of behavioral equivalence by *one* 6-dimensional energy game, where energies bound capabilities of an attacker trying to tell processes apart. The defender-winning initial credits exhaustively determine which preorders and equivalences from the (strong) linear-time–branching-time spectrum relate processes.

The time complexity is exponential, which is optimal due to trace equivalence being covered. This complexity improves drastically on our previous approach for deciding groups of equivalences where exponential sets of distinguishing HML formulas are constructed on top of a super-exponential reachability game. In experiments using the VLTS benchmarks, the algorithm performs on par with the best similarity algorithm.

Keywords: Bisimulation · Energy games · Process equivalence spectrum

1 Introduction

Many verification tasks can be understood along the lines of “how equivalent” two models are. Figure 1 replicates a standard example, known for instance from the

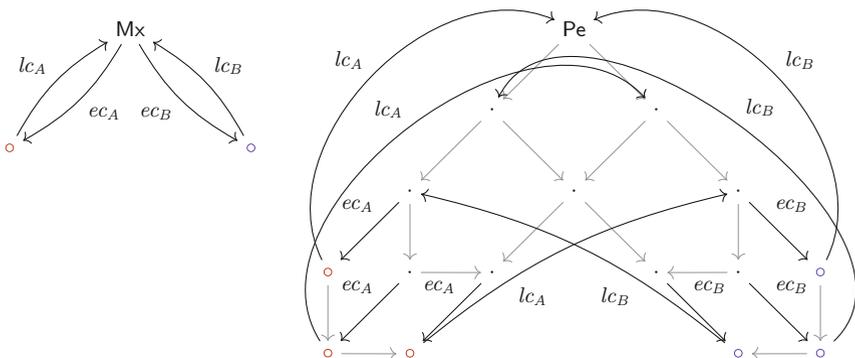


Fig. 1. A specification of mutual exclusion Mx, and Peterson’s protocol Pe.

textbook *Reactive Systems* [3]: A specification of mutual exclusion Mx as two alternating users A and B entering their critical section ec_A/ec_B and leaving lc_A/lc_B before the other may enter; and the transition system of Peterson’s [28] mutual exclusion algorithm Pe, minimized by weak bisimilarity, with internal steps \rightarrow due to the coordination that needs to happen. For Pe to faithfully implement mutual exclusion, it should behave somewhat similarly to Mx.

Semantics in concurrent models must take nondeterminism into account. Setting the degree to which nondeterminism counts induces equivalence notions with subtle differences: Pe and Mx *weakly simulate* each other, meaning that a tree of options from one process can be matched by a similar tree of the other. This implies that they have the same *weak traces*, that is, matching paths. However, they are not weakly *bi*-similar, which would require a higher degree of symmetry than mutual simulation, namely, matching absence of options. There are many more such notions. Van Glabbeek’s *linear-time-branching-time spectrum* [21] (cf. Fig. 3) brings order to the hierarchy of equivalences. But it is notoriously difficult to navigate. In our example, one might wonder: Are there notions relating the two *besides* mutual simulation?

Our recent algorithm for *linear-time-branching-time spectroscopy* by Bisping, Nestmann, and Jansen [7, 9] is capable of answering equivalence questions for finite-state systems by *deciding the spectrum of behavioral equivalences in one go*. In theory, that is. In practice, the algorithm of [7] runs out of memory when applied to the weak transition relation of even small examples like Pe. The reason for this is that saturating transition systems with the closure of weak steps adds a lot of nondeterminism. For instance, Pe may reach 10 different states by internal steps (\rightarrow^*). The spectroscopy algorithm of [7] builds a bisimulation game where the defender wins if the game starts at a pair of equivalent processes. To allow all attacks relevant for the spectrum, the [7]-game must consider partitionings of state sets reached through nondeterminism. There are 115,975 ways of partitioning 10 objects. As a consequence, the game graph of [7] comparing Pe and Mx has 266,973 game positions. On top of each position, [7] builds sets of distinguishing formulas of Hennessy–Milner modal logic (HML) [21, 24] with minimal expressiveness. These sets may grow exponentially. Game over!

Contributions. In this paper, we adapt the spectroscopy approach of [7, 9] to render small verification instances like Pe/Mx feasible. The key ingredients that will make the difference are: understanding the spectrum purely through *depth-properties of HML formulas*; using *multidimensional energy games* [15] instead of reachability games; and exploiting the considered spectrum to drastically *reduce the branching-degree* of the game as well as the height of the energy lattice. Figure 2 lays out the algorithm with pointers to key parts of this paper.

- Subsection 2.2 explains how the *linear-time-branching-time spectrum can be understood in terms of six dimensions of HML expressiveness*, and Subsect. 3.1 introduces a class of *declining energy games* fit for our task.
- In Subsect. 3.2, we describe the novel *spectroscopy energy game*, and, in Subsect. 3.3, *prove it to characterize all notions of equivalence definable by the six dimensions*.

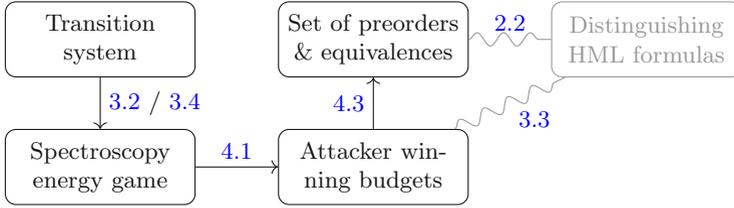


Fig. 2. Overview of the computations \rightarrow and correspondences \sim we will discuss.

- Subsection 3.4 shows that a *more clever game with only linear branching-factor* still covers the spectrum.
- Subsection 4.1 provides an algorithm to *compute winning initial energy levels for declining energy games with $\min_{\{.. \}}$* , which enables *decision of the whole considered spectrum in $2^{\mathcal{O}(|\mathcal{P}|)}$* for systems with $|\mathcal{P}|$ processes (Subsect. 4.2).
- In Subsect. 4.3, we add fine print on *how to obtain equivalences and distinguishing formulas* in the algorithm.
- Section 5 compares to [7] and [29] through *experiments with the widely used VLTS benchmark suite* [18]. The experiments also reveal insights about the suite itself.

2 Distinctions and Equivalences in Transition Systems

Two classic concepts of system analysis form the background of this paper: *Hennessy–Milner logic* (HML) interpreted over *transition systems* goes back to Hennessy and Milner [24] investigating observational equivalence in operational semantics. Van Glabbeek’s *linear-time–branching-time spectrum* [21] arranges all common notions of equivalence as a hierarchy of HML sublanguages.

2.1 Transition Systems and Hennessy–Milner Logic

Definition 1 (Labeled transition system). A labeled transition system is a tuple $\mathcal{S} = (\mathcal{P}, \Sigma, \rightarrow)$ where \mathcal{P} is the set of processes, Σ is the set of actions, and $\rightarrow \subseteq \mathcal{P} \times \Sigma \times \mathcal{P}$ is the transition relation.

By $\mathcal{I}(p)$ we denote the actions enabled initially for a process $p \in \mathcal{P}$, that is, $\mathcal{I}(p) := \{a \in \Sigma \mid \exists p'. p \xrightarrow{a} p'\}$. We lift the steps to sets with $P \xrightarrow{a} P'$ iff $P' = \{p' \mid \exists p \in P. p \xrightarrow{a} p'\}$.

Hennessy–Milner logic expresses *observations* that one may make on such a system. The set of formulas true of a process offers a denotation for its semantics.

Definition 2 (Hennessy–Milner logic). The syntax of Hennessy–Milner logic over a set Σ of actions, $\text{HML}[\Sigma]$, is defined by the grammar:

$$\begin{aligned} \varphi &::= \langle a \rangle \varphi && \text{with } a \in \Sigma \\ &| \bigwedge \{ \psi, \psi, \dots \} \\ \psi &::= \neg \varphi \mid \varphi. \end{aligned}$$

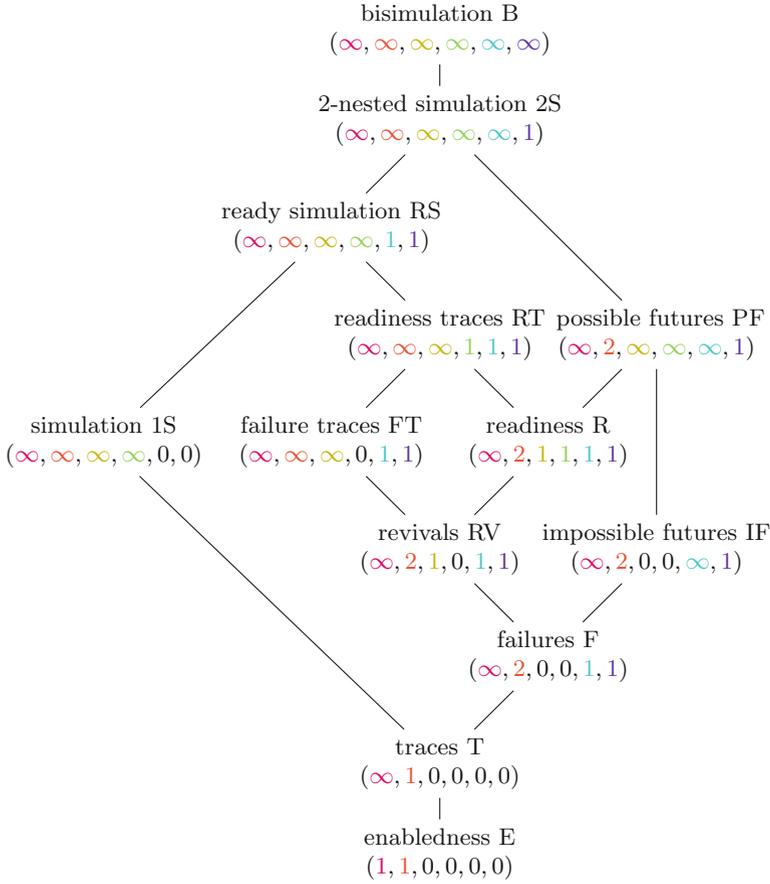


Fig. 3. Hierarchy of equivalences/preorders becoming finer towards the top.

Its semantics $\llbracket \cdot \rrbracket^{\mathcal{S}}$ over a transition system $\mathcal{S} = (\mathcal{P}, \Sigma, \rightarrow)$ is given as the set of processes where a formula “is true” by:

$$\begin{aligned} \llbracket \langle a \rangle \varphi \rrbracket^{\mathcal{S}} &:= \{p \in \mathcal{P} \mid \exists p' \in \llbracket \varphi \rrbracket^{\mathcal{S}}. p \xrightarrow{a} p'\} \\ \llbracket \bigwedge_{i \in I} \psi_i \rrbracket^{\mathcal{S}} &:= \bigcap_{i \in I} \{ \llbracket \psi_i \rrbracket^{\mathcal{S}} \mid i \in I \wedge \nexists \varphi. \psi_i = \neg \varphi \} \\ &\quad \setminus \bigcup \{ \llbracket \varphi \rrbracket^{\mathcal{S}} \mid \exists i \in I. \psi_i = \neg \varphi \}. \end{aligned}$$

HML basically extends propositional logic with a modal observation operation. Conjunctions then bound trees of future behavior. Positive conjuncts mean lower bounds, negative ones impose upper bounds. For the scope of this paper, finite bounds suffice, i.e., conjunctions are finite-width. The empty conjunction $\top := \bigwedge \emptyset$ is usually omitted in writing.

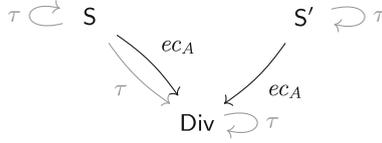


Fig. 4. Example system of internal decision $\xrightarrow{\tau}$ against an action $\xrightarrow{ec_A}$.

We use Hennessy–Milner logic to capture *differences* between program behaviors. Depending on how much of its expressiveness we use, different notions of equivalence are characterized.

Definition 3 (Distinguishing formulas and preordering languages). A formula $\varphi \in \text{HML}[\Sigma]$ is said to distinguish two processes $p, q \in \mathcal{P}$ iff $p \in \llbracket \varphi \rrbracket^S$ and $q \notin \llbracket \varphi \rrbracket^S$. A sublanguage of Hennessy–Milner logic, $\mathcal{O}_X \subseteq \text{HML}[\Sigma]$, either distinguishes two processes, $p \not\preceq_X q$, if it contains a distinguishing formula, or preorders them otherwise. If processes are preordered in both directions, $p \preceq_X q$ and $q \preceq_X p$, then they are considered X -equivalent, $p \sim_X q$.

Fig. 3 charts the *linear-time–branching-time spectrum*. If processes are preordered/equated by one notion of equivalence, they also are preordered/equated by every notion below. We will later formally characterize the notions through Proposition 1. For a thorough presentation, we point to [23]. For those familiar with the spectrum, the following example serves to refresh memories.

Example 1. Fig. 4 shows a tiny slice of the weak-step-saturated version of our initial example from Fig. 1 that is at the heart of why Pe and Mx are not bisimulation-equivalent. The difference between S and S' is that S can internally transition to Div (labeled $\xrightarrow{\tau}$) without ever performing an ec_A action. We can express this difference by the formula $\varphi_S := \langle \tau \rangle \wedge \{ \neg \langle ec_A \rangle \}$, meaning “after τ , ec_A may be impossible.” It is true for S, but not for S'. Knowing a distinguishing formula means that S and S' cannot be bisimilar by the Hennessy–Milner theorem. The formula φ_S is called a *failure* (or *refusal*) as it specifies a set of actions that are disabled after a trace. In the other direction of comparison, the negation $\varphi_{S'} := \wedge \{ \neg \langle \tau \rangle \wedge \{ \neg \langle ec_A \rangle \} \}$ distinguishes S' from S. The differences between the two processes cannot be expressed in HML without negation. Therefore the processes are simulation-equivalent, or *similar*, as similarity is characterized by the positive fragment of HML.

2.2 Price Spectra of Behavioral Equivalences

For algorithms exploring the linear-time–branching-time spectrum, it is convenient to have a representation of the spectrum in terms of numbers or “prices” of formulas as in [7]. We, here, use six dimensions to characterize the notions of equivalence depicted in Fig. 3. The numbers define the HML observation languages that characterize the very preorders/equivalences. Intuitively, the colorful

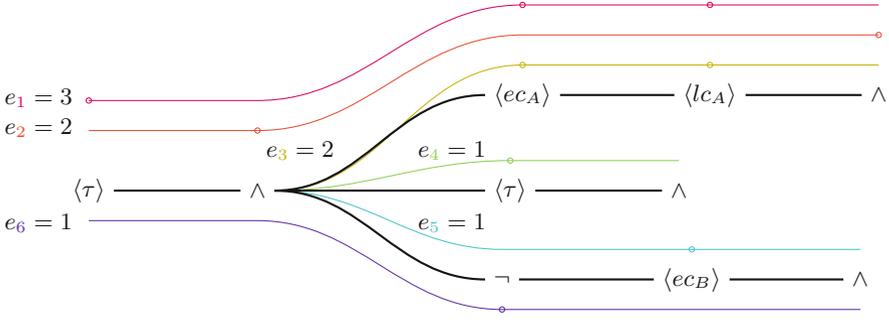


Fig. 5. Pricing e of formula $\langle \tau \rangle \wedge \{ \langle ec_A \rangle \langle lc_A \rangle \top, \langle \tau \rangle \top, \neg \langle ec_B \rangle \top \}$.

numbers mean: (1) Formula modal depth of observations. (2) Formula nesting depth of conjunctions. (3) Maximal modal depth of deepest positive clauses in conjunctions. (4) Maximal modal depth of the other positive clauses in conjunctions. (5) Maximal modal depth of negative clauses in conjunctions. (6) Formula nesting depth of negations. More formally:

Definition 4 (Energies). We denote as energies, \mathbf{En} , the set of N -dimensional vectors $(\mathbb{N})^N$, and as extended energies, \mathbf{En}_∞ , the set $(\mathbb{N} \cup \{\infty\})^N$.

We compare energies component-wise, i.e., $(e_1, \dots, e_N) \leq (f_1, \dots, f_N)$ iff $e_i \leq f_i$ for each i . Least upper bounds \sup are defined as usual as component-wise supremum, as are greatest lower bounds \inf .

Definition 5 (Formula prices). The expressiveness price $\text{expr}: \text{HML}[\Sigma] \rightarrow (\mathbb{N})^6$ of a formula interpreted as 6-dimensional energies is defined recursively by:

$$\text{expr}(\langle a \rangle \varphi) := \begin{pmatrix} 1 + \text{expr}_1(\varphi) \\ \text{expr}_2(\varphi) \\ \text{expr}_3(\varphi) \\ \text{expr}_4(\varphi) \\ \text{expr}_5(\varphi) \\ \text{expr}_6(\varphi) \end{pmatrix} \quad \text{expr}(\neg \varphi) := \begin{pmatrix} \text{expr}_1(\varphi) \\ \text{expr}_2(\varphi) \\ \text{expr}_3(\varphi) \\ \text{expr}_4(\varphi) \\ \text{expr}_5(\varphi) \\ 1 + \text{expr}_6(\varphi) \end{pmatrix}$$

$$\text{expr}\left(\bigwedge_{i \in I} \psi_i\right) := \sup \left(\left\{ \begin{pmatrix} 0 \\ 1 + \sup_{i \in I} \text{expr}_2(\psi_i) \\ \sup_{i \in Pos} \text{expr}_1(\psi_i) \\ \sup_{i \in Pos \setminus R} \text{expr}_1(\psi_i) \\ \sup_{i \in Neg} \text{expr}_1(\psi_i) \\ 0 \end{pmatrix} \right\} \cup \{ \text{expr}(\psi_i) \mid i \in I \} \right)$$

$$\begin{aligned} Neg &:= \{i \in I \mid \exists \varphi'_i. \psi_i = \neg \varphi'_i\} \\ Pos &:= I \setminus Neg \\ R &:= \begin{cases} \emptyset & \text{if } Pos = \emptyset \\ \{r\} & \text{for some } r \in Pos \text{ where } \text{expr}_1(\psi_r) \text{ maximal for } Pos. \end{cases} \end{aligned}$$

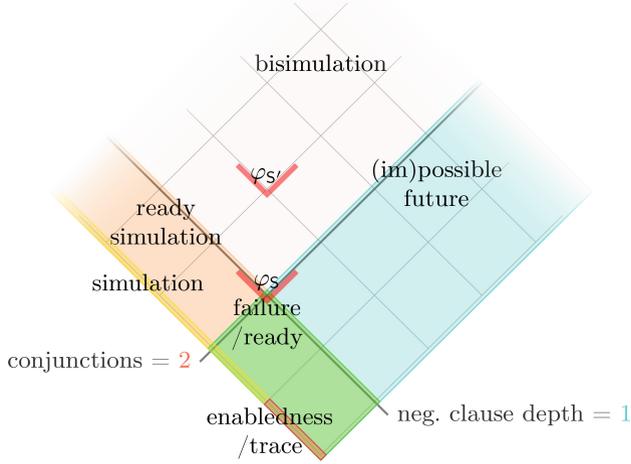


Fig. 6. Cut through the price lattice with dimensions 2 (conjunction nesting) and 5 (negated observation depth).

Figure 5 gives an example how the prices compound. The colors of the lines match those used for the dimensions and their updates in the other figures. Circles mark the points that are counted. The formula itself expresses a so-called *ready-trace* observation: We observe a trace $\tau \cdot ec_A \cdot lc_A$ and, along the way, may check what other options would have been enabled or disabled. Here, we check that τ is enabled and ec_B is disabled after the first τ -step. With the pricing, we can characterize all standard notions of equivalence:

Proposition 1. *On finite systems, the languages of formulas with prices below the coordinates given in Fig. 3 characterize the named notions of equivalence, that is, $p \preceq_X q$ with respect to equivalence X , iff no φ with $\text{expr}(\varphi) \leq e_X$ distinguishes p from q .*

Example 2. The formulas of Example 1 have prices: $\text{expr}(\langle \tau \rangle \wedge \{ \neg \langle ec_A \rangle \}) = (2, 2, 0, 0, 1, 1)$ for φ_S and $\text{expr}(\wedge \{ \neg \langle \tau \rangle \wedge \{ \neg \langle ec_A \rangle \} \}) = (2, 3, 0, 0, 2, 2)$ for $\varphi_{S'}$. The prices of the two are depicted as red marks in Fig. 6. This also visualizes how $\varphi_{S'}$ is a counterexample for bisimilarity and how φ_S is a counterexample for failure and finer preorders. Indeed the two preorders are coarsest ways of telling the processes apart. So, S and S' are equated by all preorders *below* the marks, i.e. similarity, $S \sim_{1S} S'$, and coarser preorders ($S \sim_T S'$, $S \sim_E S'$). This carries over to the initial example of Peterson’s mutex protocol from Fig. 1, where weak simulation, $Pe \sim_{1WS} Mx$, is the most precise equivalence. Practically, this means that the specification Mx has liveness properties not upheld by the implementation Px .

Remark 1. Definition 5 deviates from our previous formula pricing of [7] in a crucial way: We only collect the *maximal depths of positive clauses*, whereas [7]

tracks *numbers of deep and flat positive clauses* (where a flat clause is characterized by an observation depth of 1). Our change to a purely “depth-guided” spectrum will allow us to characterize the spectrum by an energy game and to eliminate the Bell-numbered blow up from the game’s branching-degree. The special treatment of the deepest positive branch is necessary to address revival, failure trace, and ready trace semantics, which are popular in the CSP community [17, 31].

3 An Energy Game of Distinguishing Capabilities

Conventional equivalence problems ask whether a pair of processes is related by a specific equivalence. These problems can be abstracted into a more general “spectroscopy problem” to determine the set of equivalences from a spectrum that relate two processes as in [7]. This section captures the spectrum of Fig. 3 by one rather simple energy game.

3.1 Energy Games

Multidimensional energy games are played on graphs labeled by vectors to be added to (or subtracted from) a vector of “energies” where one player must pay attention to the energies not being exhausted. We plan to encode the distinction capabilities of the semantic spectrum as energy levels in an energy game enriched by $\min_{\{.. \}}$ -operations that takes minima of components. This way, energy levels where the defender has a winning strategy will correspond to equivalences that hold. We will just need updates decrementing or maintaining energy levels.

Definition 6 (Energy updates). *The set of energy updates, \mathbf{Up} , contains vectors $(u_1, \dots, u_N) \in \mathbf{Up}$ where each component is of the form*

- $u_k \in \{-1, 0\}$, or
- $u_k = \min_D$ where $D \subseteq \{1, \dots, N\}$ and $k \in D$.

Applying an update to an energy, $\text{upd}(e, u)$, where $e = (e_1, \dots, e_N) \in \mathbf{En}$ (or \mathbf{En}_∞) and $u = (u_1, \dots, u_N) \in \mathbf{Up}$, yields a new energy vector e' where k th components $e'_k := e_k + u_k$ for $u_k \in \mathbb{Z}$ and $e'_k := \min_{d \in D} e_d$ for $u_k = \min_D$. Updates that would cause any component to become negative are illegal.

Definition 7 (Games). *An N -dimensional declining energy game $\mathcal{G}[g_0, e_0] = (G, G_d, \succrightarrow, w, g_0, e_0)$ is played on a directed graph uniquely labeled by energy updates consisting of*

- a set of game positions G , partitioned into
 - a set of defender positions $G_d \subseteq G$
 - a set of attacker positions $G_a := G \setminus G_d$,
- a relation of game moves $\succrightarrow \subseteq G \times G$,
- a weight function for the moves $w: (\succrightarrow) \rightarrow \mathbf{Up}$,
- an initial position $g_0 \in G$, and

– an initial energy budget vector $e_0 \in \mathbf{En}_\infty$.

The notation $g \xrightarrow{u} g'$ stands for $g \succrightarrow g'$ and $w(g, g') = u$.

Definition 8 (Plays, energies, and wins). We call the (finite or infinite) paths $\rho = g_0 g_1 \dots \in G^\infty$ with $g_i \xrightarrow{u_i} g_{i+1}$ plays of $\mathcal{G}[g_0, e_0]$.

The energy level of a play ρ at round i , $\mathbf{EL}_\rho(i)$, is recursively defined as $\mathbf{EL}_\rho(0) := e_0$ and otherwise as $\mathbf{EL}_\rho(i+1) := \text{upd}(\mathbf{EL}_\rho(i), u_i)$. If we omit the index, \mathbf{EL}_ρ , this refers to the final energy level of a finite run ρ , i.e., $\mathbf{EL}_\rho(|\rho| - 1)$.

Plays where energy levels become undefined (negative) are won by the defender. So are infinite plays. If a finite play is stuck (i.e., $g_0 \dots g_n \not\succrightarrow$), the stuck player loses: The defender wins if $g_n \in G_a$, and the attacker wins if $g_n \in G_d$.

Proposition 2. In this model, energy levels can only decline.

1. Updates may only decrease energies, $\text{upd}(e, u) \leq e$.
2. Energy level changes are monotonic: If $\mathbf{EL}_{\rho g} \leq \mathbf{EL}_{\sigma g}$ and $g \succrightarrow g'$ then $\mathbf{EL}_{\rho g g'} \leq \mathbf{EL}_{\sigma g g'}$.
3. If $e_0 \leq e'_0$ and $\mathcal{G}[g_0, e_0]$ has non-negative play ρ , then $\mathcal{G}[g_0, e'_0]$ also has non-negative play ρ .

Definition 9 (Strategies and winning budgets). An attacker strategy is a map from play prefixes ending in attacker positions to next game moves $s: (G^* \times G_a) \rightarrow G$ with $s(g_0 \dots g_a) \in (g_a \succrightarrow \cdot)$. Similarly, a defender strategy names moves starting in defender states. If all plays consistent with a strategy s ensure a player to win, s is called a winning strategy for this player. The player with a winning strategy for $\mathcal{G}[g_0, e_0]$ is said to win \mathcal{G} from position g_0 with initial energy budget e_0 . We call $\text{Win}_a(g) = \{e \mid \mathcal{G}[g, e] \text{ is won by the attacker}\}$ the attacker winning budgets.

Proposition 3. The attacker winning budgets at positions are upward-closed with respect to energy, that is, $e \in \text{Win}_a(g)$ and $e \leq e'$ implies $e' \in \text{Win}_a(g)$.

This means we can characterize the set of winning attacker budgets in terms of minimal winning budgets $\text{Win}_a^{\min}(g) = \text{Min}(\text{Win}_a(g))$, where $\text{Min}(S)$ selects minimal elements $\{e \in S \mid \nexists e' \in S. e' \leq e \wedge e' \neq e\}$. Clearly, Win_a^{\min} must be an antichain and thus finite due to the energies being well-partially-ordered [26]. Dually, we may consider the maximal energy levels winning for the defender, $\text{Win}_d^{\max}: G \rightarrow \mathbf{2}^{\mathbf{En}_\infty}$ where we need extended energies to bound won half-spaces.

3.2 The Spectroscopy Energy Game

Let us now look at the “spectroscopy energy game” at the center of our contribution. Figure 7 gives a graphical representation. The intuition is that the attacker shows how to construct formulas that distinguish a process p from every q in a set of processes Q . The energies limit the expressiveness of the formulas. The first dimension bounds for how many turns the attacker may challenge observations

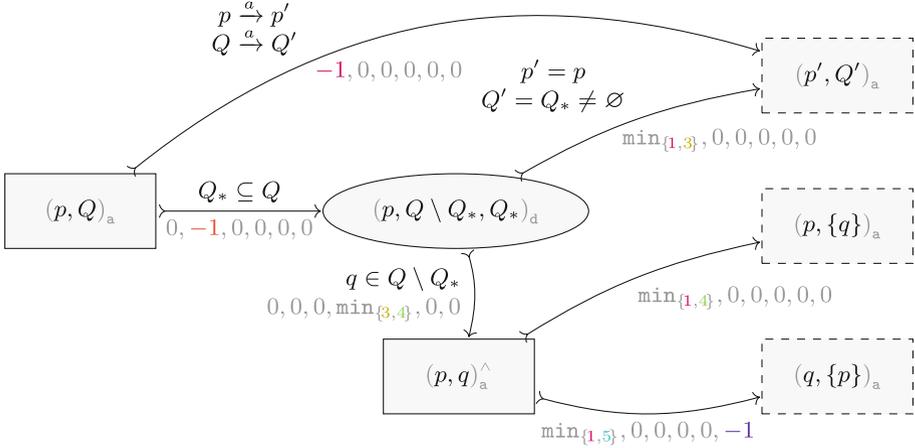


Fig. 7. Schematic spectroscopy game \mathcal{G}_Δ of Definition 10.

of actions. The second dimension limits how often they may use conjunctions to resolve nondeterminism. The third, fourth, and fifth dimensions limit how deeply observations may nest underneath a conjunction, where the fifth stands for negated clauses, the third for one of the deepest positive clauses and the fourth for the other positive clauses. The last dimension limits how often the attacker may use negations to enforce symmetry by swapping sides. The moves closely match productions in the grammar of Definition 2 and prices in Definition 5.

Definition 10. (Spectroscopy energy game). For a system $\mathcal{S} = (\mathcal{P}, \Sigma, \rightarrow)$, the 6-dimensional spectroscopy energy game $\mathcal{G}_\Delta^{\mathcal{S}}[g_0, e_0] = (G, G_d, \rightarrow, w, g_0, e_0)$ consists of

- attacker positions $(p, Q)_a \in G_a$,
- attacker clause positions $(p, q)_a^\wedge \in G_a$,
- defender conjunction positions $(p, Q, Q_*)_d \in G_d$,

where $p, q \in \mathcal{P}$ and $Q, Q_* \in 2^{\mathcal{P}}$, and six kinds of moves:

- observation moves $(p, Q)_a \xrightarrow{(-1, 0, 0, 0, 0, 0)} (p', Q')_a$ if $p \xrightarrow{a} p', Q \xrightarrow{a} Q'$,
- conj. challenges $(p, Q)_a \xrightarrow{(0, -1, 0, 0, 0, 0)} (p, Q \setminus Q_*, Q_*)_d$ if $Q_* \subseteq Q$,
- conj. revivals $(p, Q, Q_*)_d \xrightarrow{(\min_{\{1,3\}}, 0, 0, 0, 0, 0)} (p, Q_*)_a$ if $Q_* \neq \emptyset$,
- conj. answers $(p, Q, Q_*)_d \xrightarrow{(0, 0, 0, \min_{\{3,4\}}, 0, 0)} (p, q)_a^\wedge$ if $q \in Q$,
- positive decisions $(p, q)_a^\wedge \xrightarrow{(\min_{\{1,4\}}, 0, 0, 0, 0, 0)} (p, \{q\})_a$, and
- negative decisions $(p, q)_a^\wedge \xrightarrow{(\min_{\{1,5\}}, 0, 0, 0, 0, -1)} (q, \{p\})_a$ if $p \neq q$.

The spectroscopy energy game is a bisimulation game in the tradition of Stirling [33].

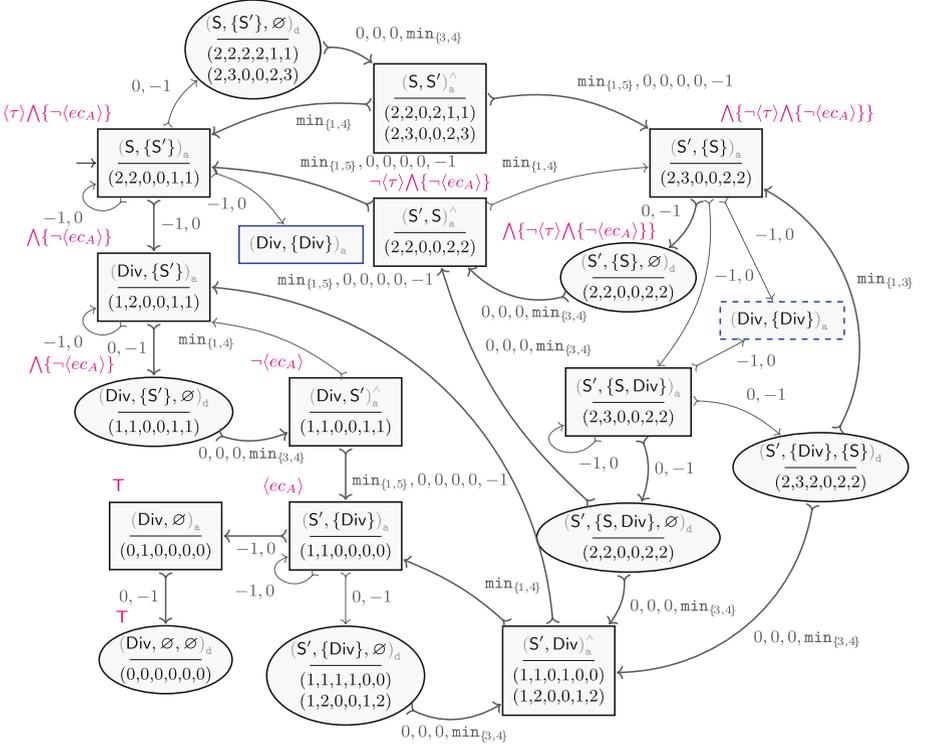


Fig. 8. Example 3 spectroscopy energy game, minimal attacker winning budgets, and distinguishing formulas/clauses. (In order to reduce visual load, only the first components of the updates are written next to the edges. The other components are 0.)

Lemma 1. (Bisimulation game, proof see [5]). p_0 and q_0 are bisimilar iff the defender wins $\mathcal{G}_\Delta((p_0, \{q_0\})_a, e_0)$ for every initial energy budget e_0 , i.e. if $(\infty, \infty, \infty, \infty, \infty, \infty) \in \text{Win}_d^{\max}((p_0, \{q_0\})_a)$.

In other words, if there are initial budgets winning for the attacker, then the compared processes can be told apart. For \mathcal{G}_Δ , the attacker “unknown initial credit problem” in energy games [34] coincides with the “apartness problem” [20] for processes.

Example 3. Figure 8 shows the spectroscopy energy game starting at $(S, \{S'\})_a$ from Example 1. The lower part of each node displays the node’s Win_a^{\min} . The magenta HML formulas illustrate distinctions relevant for the correctness argument of the following Subsect. 3.3. Section 4 will describe how to obtain attacker winning budgets and equivalences. The blue “symmetric” positions are definitely won by the defender—we omit the game graph below them. We also omit the move $(S', \{S, Div\})_a \xrightarrow{(0, -1, 0, 0, 0, 0)} (S', \{S\}, \{Div\})_d$ —it can be ignored as will be discussed in Subsect. 3.4.

3.3 Correctness: Tight Distinctions

We will check that winning budgets indeed characterize what equivalences hold by constructing price-minimal distinguishing formulas from attacker budgets.

Definition 11 (Strategy formulas). *Given the set of winning budgets Win_a , the set of attacker strategy formulas Strat for a position with given energy level e is defined inductively as follows:*

$$\begin{aligned}
& \langle b \rangle \varphi \in \text{Strat}((p, Q)_a, e) \text{ if } (p, Q)_a \xrightarrow{u} (p', Q')_a, e' = \text{upd}(e, u) \in \text{Win}_a((p', Q')_a), \\
& \quad p \xrightarrow{b} p', Q \xrightarrow{b} Q', \text{ and } \varphi \in \text{Strat}((p', Q')_a, e'), \\
& \varphi \in \text{Strat}((p, Q)_a, e) \text{ if } (p, Q)_a \xrightarrow{u} (p, Q, Q_*)_d, e' = \text{upd}(e, u) \in \text{Win}_a((p, Q, Q_*)_d), \\
& \quad \text{and } \varphi \in \text{Strat}((p, Q, Q_*)_d, e'), \\
& \bigwedge_{q \in Q} \psi_q \in \text{Strat}((p, Q, \emptyset)_d, e) \text{ if } (p, Q, \emptyset)_d \xrightarrow{u_q} (p, q)_a^\wedge, e_q = \text{upd}(e, u_q) \in \\
& \quad \text{Win}_a((p, q)_a^\wedge) \text{ and } \psi_q \in \text{Strat}((p, q)_a^\wedge, e_q) \text{ for each } q \in Q, \\
& \bigwedge_{q \in Q \cup \{*\}} \psi_q \in \text{Strat}((p, Q, Q_*)_d, e) \text{ if } (p, Q, Q_*)_d \xrightarrow{u_q} (p, q)_a^\wedge, e_q = \text{upd}(e, u_q) \in \\
& \quad \text{Win}_a((p, q)_a^\wedge) \text{ and } \psi_q \in \text{Strat}((p, q)_a^\wedge, e_q) \text{ for each } q \in Q, \text{ and if } (p, Q, Q_*)_d \xrightarrow{u_*} \\
& \quad (p, Q_*)_a, e_* = \text{upd}(e, u_*) \in \text{Win}_a((p, Q_*)_a), \text{ and } \psi_* \in \text{Strat}((p, Q_*)_a, e_*) \text{ is an} \\
& \quad \text{observation,} \\
& \varphi \in \text{Strat}((p, q)_a^\wedge, e) \text{ if } (p, q)_a^\wedge \xrightarrow{u} (p, \{q\})_a, e' = \text{upd}(e, u) \in \text{Win}_a((p, \{q\})_a) \\
& \quad \text{and } \varphi \in \text{Strat}((p, \{q\})_a, e') \text{ is an observation, and} \\
& \neg \varphi \in \text{Strat}((p, q)_a^\wedge, e) \text{ if } (p, q)_a^\wedge \xrightarrow{u} (q, \{p\})_a, e' = \text{upd}(e, u) \in \text{Win}_a((q, \{p\})_a) \\
& \quad \text{and } \varphi \in \text{Strat}((q, \{p\})_a, e') \text{ is an observation.}
\end{aligned}$$

Because of the game structure, we actually know the u needed in each line of the definition. It is $u = (-1, 0, 0, 0, 0)$ in the first case; $(0, -1, 0, 0, 0)$ in the second; $(0, 0, 0, \min_{\{3,4\}}, 0)$ in the third; $(0, 0, 0, \min_{\{3,4\}}, 0)$ and $(\min_{\{1,3\}}, 0, 0, 0, 0)$ in the fourth; $(\min_{\{1,4\}}, 0, 0, 0, 0)$ in the fifth; and $(\min_{\{1,5\}}, 0, 0, 0, 0, -1)$ in last case. $\text{Strat}((p, q)_a^\wedge, \cdot)$ can contain negative clauses, which form no proper formulas on their own.

Lemma 2 (Price soundness). $\varphi \in \text{Strat}((p, Q)_a, e)$ implies that $\text{expr}(\varphi) \leq e$ and that $\text{expr}(\varphi) \in \text{Win}_a((p, Q)_a)$.

Proof. By induction on the structure of φ with arbitrary p, Q, e , exploiting the alignment of the definitions of winning budgets and formula prices. Full proof in [5].

Lemma 3 (Price completeness). $e_0 \in \text{Win}_a((p_0, Q_0)_a)$ implies there are elements in $\text{Strat}((p_0, Q_0)_a, e_0)$.

Proof. By induction on the tree of winning plays consistent with some attacker winning strategy implied by $e_0 \in \text{Win}_a((p_0, Q_0)_a)$. Full proof in [5].

Lemma 4 (Distinction soundness). Every $\varphi \in \text{Strat}((p, Q)_a, e)$ distinguishes p from every $q \in Q$.

Proof. By induction on the structure of φ with arbitrary p, Q, e , exploiting that Strat can only construct formulas with the invariant that they are true for p and false for each $q \in Q$. Full proof in [5].

Lemma 5 (Distinction completeness). *If φ distinguishes p from every $q \in Q$, then $\text{expr}(\varphi) \in \text{Win}_a((p, Q)_a)$.*

Proof. By induction on the structure of φ with arbitrary p, Q , exploiting the alignment of game structure and HML semantics and the fact that expr cannot “overtake” inverse updates. Full proof in [5].

Theorem 1 (Correctness). *For any equivalence X with coordinate e_X , $p \preceq_X q$, precisely if all $e_{pq} \in \text{Win}_a^{\min}((p, \{q\})_a)$ are above or incomparable, $e_{pq} \not\leq e_X$.*

Proof. By contraposition, in both directions.

- Assume $p \not\preceq_X q$. This means some φ with $\text{expr}(\varphi) \leq e_X$ distinguishes p from q . By Lemma 5, $\text{expr}(\varphi) \in \text{Win}_a((p, \{q\})_a)$. Then either $\text{expr}(\varphi)$ or a lower price $e_{pq} \leq \text{expr}(\varphi)$ are minimal winning budgets, i.e., $e_{pq} \in \text{Win}_a^{\min}((p, \{q\})_a)$ and $e_{pq} \leq e_X$.
- Assume there is $e_{pq} \in \text{Win}_a^{\min}((p, \{q\})_a)$ with $e_{pq} \leq e_X$. By Lemma 3, there is $\varphi \in \text{Strat}((p, \{q\})_a, e_{pq})$. Due to Lemma 4, φ must be distinguishing for p and q , and due to Lemma 2, $\text{expr}(\varphi) \leq e_{pq} \leq e_X$.

The theorem basically means that by fixing an initial budget in \mathcal{G}_Δ , we can obtain a characteristic game for any notion from the spectrum.

3.4 Becoming More Clever by Looking One Step Ahead

The spectroscopy energy game \mathcal{G}_Δ of Definition 10 may branch exponentially with respect to $|Q|$ at conjunction challenges after $(p, Q)_a$. For the spectrum we are interested in, we can drastically limit the sensible attacker moves to four options by a little lookahead into the enabled actions $\mathcal{I}(q)$ of $q \in Q$ and $\mathcal{I}(p)$.

Definition 12 (Clever spectroscopy game). *The clever spectroscopy game, $\mathcal{G}_\blacktriangle$, is defined exactly like the previous spectroscopy energy game of Definition 10 with the restriction of the conjunction challenges*

$$(p, Q)_a \xrightarrow{(0, -1, 0, 0, 0)}_{\blacktriangle} (p, Q \setminus Q_*, Q_*)_d \quad \text{with } Q_* \subseteq Q,$$

to situations where $Q_* \in \{\emptyset, \{q \in Q \mid \mathcal{I}(q) \subseteq \mathcal{I}(p)\}, \{q \in Q \mid \mathcal{I}(p) \subseteq \mathcal{I}(q)\}, \{q \in Q \mid \mathcal{I}(p) = \mathcal{I}(q)\}\}$.

Theorem 2 (Correctness of cleverness). *Assume modal depth of positive clauses $e_4 \in \{0, 1, \infty\}$, $e_4 \leq e_3$, and that modal depth of negative clauses $e_5 > 1$ implies $e_3 = e_4$. Then, the attacker wins $\mathcal{G}_\blacktriangle[(p_0, Q_0)_a, e]$ precisely if they win $\mathcal{G}_\Delta[(p_0, Q_0)_a, e]$.*

Proof. The implication from the clever spectroscopy game $\mathcal{G}_\blacktriangle$ to the full spectroscopy game \mathcal{G}_Δ is trivial as the attacker moves in $\succrightarrow_{\blacktriangle}$ are a subset of those in \succrightarrow_{Δ} and the defender has the same moves in both games. For the other direction, we have to show that any move $(p, Q)_a \xrightarrow{(0, -1, 0, 0, 0)}_{\Delta} (p, Q \setminus Q_*, Q_*)_d$ winning at energy level e can be simulated by a winning move $(p, Q)_a \xrightarrow{(0, -1, 0, 0, 0)}_{\blacktriangle} (p, Q \setminus Q', Q')_d$. Full proof in [5].

4 Computing Equivalences

The previous section has shown that attacker winning budgets in the spectroscopy energy game characterize distinguishable processes and, dually, that the defender's wins characterize equivalences. We now examine how to actually compute the winning budgets of both players.

4.1 Computation of Attacker Winning Budgets

The winning budgets of the attacker (Definition 9) are characterized inductively:

- Where the defender is stuck, $g \in G_d$ and $g \not\rightsquigarrow$, the attacker wins with any budget, $(0, 0, 0, 0, 0) \in \text{Win}_a^{\min}(g)$.
- Where the defender has moves, $g \in G_d$ and $g \xrightarrow{u_i} g'_i$ (for some indexing $i \in I$ over all possible moves), the attacker wins if they have a budget equal or above to all budgets that might be necessary after the defender's move: If $\text{upd}(e, u_i) \in \text{Win}_a(g'_i)$ for all $i \in I$, then $e \in \text{Win}_a(g)$.
- Where the attacker moves, $g \in G_a$ and $g \xrightarrow{u} g'$, $\text{upd}(e, u) \in \text{Win}_a(g')$ implies $e \in \text{Win}_a(g)$.

By Proposition 3, it suffices to find the finite set of minimal winning budgets, Win_a^{\min} . Turning this into a computation is not as straightforward as in other energy game models. Due to the \min_D -updates, the energy update function $\text{upd}(\cdot, u)$ is neither injective nor surjective.

We must choose an inversion function upd^{-1} that picks minimal solutions and that minimally “casts up” inputs that are outside the image of $\text{upd}(\cdot, u)$, i.e., such that $\text{upd}^{-1}(e', u) = \inf\{e \mid e' \leq \text{upd}(e, u)\}$. We compute it as follows:

Definition 13 (Inverse update). *The inverse update function is defined as $\text{upd}^{-1}(e', u) := \sup(\{e\} \cup \{m(i) \mid \exists D. u_i = \min_D\})$ with $e_i = e'_i - u_i$ for all i where $u_i \in \{0, -1\}$ and $e_i = e'_i$ otherwise, and with $(m(i))_j = e'_i$ for $u_i = \min_D$ and $j \in D$, and $(m(i))_j = 0$ otherwise, for all i, j .*

Example 4. Let $u := (\min_{\{1,3\}}, \min_{\{1,2\}}, -1, -1)$. $(3, 4, 0, 1) \notin \text{img}(\text{upd}(\cdot, u))$, but:

$$\text{upd}^{-1}((3, 4, 0, 1), u) = \sup\{(3, 4, 1, 2), (3, 0, 3, 0), (4, 4, 0, 0)\} = (4, 4, 3, 2)$$

$$\text{upd}((4, 4, 3, 2), u) = (3, 4, 2, 1) \geq (3, 4, 0, 1)$$

$$\text{upd}^{-1}((3, 4, 2, 1), u) = \sup\{(3, 4, 3, 2), (3, 0, 3, 0), (4, 4, 0, 0)\} = (4, 4, 3, 2)$$

With upd^{-1} , we only need to find the Win_a^{\min} relation as a least fixed point of the inductive description. This is done by Algorithm 1. Every time a new way of winning a position for the attacker is discovered, this position is added to the `todo`. Initially, these are the positions where the defender is stuck. The update at an attacker position in Line 8 takes the inversely updated budgets (upd^{-1}) of successor positions to be tentative attacker winning budgets. At a defender position, the attacker only wins if they have winning budgets for all follow-up positions (Line 12). Any supremum of such budgets covering all follow-ups will be winning for the attacker (Line 13). At both updates, we only select the minima as a finite representation of the infinitely many attacker budgets.

```

1 def compute_winning_budgets( $\mathcal{G} = (G, G_d, \succrightarrow, w)$ ):
2   attacker_win :=  $[g \mapsto \{\} \mid g \in G]$ 
3   todo :=  $\{g \in G_d \mid g \not\succeq\}$ 
4   while todo  $\neq \emptyset$ :
5     g := some todo
6     todo := todo  $\setminus \{g\}$ 
7     if  $g \in G_a$  :
8       new_attacker_win :=  $\text{Min}(\text{attacker\_win}[g] \cup \{\text{upd}^{-1}(e', u) \mid$ 
9          $g \succrightarrow g' \wedge e' \in \text{attacker\_win}[g']\})$ 
10      else:
11        defender_post :=  $\{g' \mid g \succrightarrow g'\}$ 
12        options :=  $\{(g', \text{upd}^{-1}(e', u)) \mid g \succrightarrow g' \wedge e' \in \text{attacker\_win}[g']\}$ 
13        if  $\text{defender\_post} \subseteq \text{dom}(\text{options})$  :
14          new_attacker_win :=  $\text{Min}(\{\text{sup}_{g' \in \text{defender\_post}} \text{strat}(g') \mid$ 
15             $\text{strat} \in (G \rightarrow \mathbf{En}) \wedge \forall g'. \text{strat}(g') \in \text{options}(g')\})$ 
16          else:
17            new_attacker_win :=  $\emptyset$ 
18        if  $\text{new\_attacker\_win} \neq \text{attacker\_win}[g]$  :
19          attacker_win[g] := new_attacker_win
20          todo :=  $\text{todo} \cup \{g_p \mid \exists u. g_p \succrightarrow g\}$ 
21   Winamin := attacker_win
22   return Winamin
    
```

Algorithm 1: Algorithm for computing attacker winning budgets of declining energy game \mathcal{G} .

4.2 Complexity and How to Flatten It

For finite games, Algorithm 1 is sure to terminate in exponential time of game graph branching degree and dimensionality.

Lemma 6 (Winning budget complexity, proof see [5]). *For an N -dimensional declining energy game with \succrightarrow of branching degree o , Algorithm 1 terminates in $\mathcal{O}(|\succrightarrow| \cdot |G|^N \cdot (o + |G|^{(N-1) \cdot o}))$ time, using $\mathcal{O}(|G|^N)$ space for the output.*

Lemma 7 (Full spectroscopy complexity). *Time complexity of computing winning budgets for the full spectroscopy energy game \mathcal{G}_Δ is in $2^{\mathcal{O}(|\mathcal{P}| \cdot 2^{|\mathcal{P}|})}$.*

Proof. Out-degrees o in \mathcal{G}_Δ can be bounded in $\mathcal{O}(2^{|\mathcal{P}|})$, the whole game graph $|\succrightarrow_\Delta| \in \mathcal{O}(|\succrightarrow| \cdot 2^{|\mathcal{P}|} + |\mathcal{P}|^2 \cdot 3^{|\mathcal{P}|})$, and game positions $|G_\Delta| \in \mathcal{O}(|\mathcal{P}| \cdot 3^{|\mathcal{P}|})$. Insert with $N = 6$ in Lemma 6. Full proof in [5].

We thus have established the approach to be double-exponential. The complexity of the previous spectroscopy algorithm [7] has not been calculated. One must presume it to be equal or higher as the game graph has Bell-numbered branching degree and as the algorithm computes formulas, which entails more options than

the direct computation of energies. This is what lies behind the introduction’s observation that moderate nondeterminism already renders [7] unusable.

Our present energy game reformulation allows us to use two ingredients to do way better than double-exponentially when focussing on the common linear-time-branching-time spectrum:

First, Subsect. 3.4 has established that most of the partitionings in attacker conjunction moves can be disregarded by looking at the initial actions of processes.

Second, Fahrenberg et al. [15] have shown that considering just “capped” energies in a grid $\mathbf{En}_K = \{0, \dots, K\}^N$ can reduce complexity. Such a *flattening of the lattice* turns the space of possible energies into constant factor $(K + 1)^N$ (with $(K + 1)^{N-1}$ -sized antichains) independent of input size. For Algorithm 1, space complexity needed for `attacker_win` drops to $\mathcal{O}(|G|)$ and time complexity to $|\succrightarrow| \cdot 2^{\mathcal{O}(o)}$. If we are only interested in finitely many notions of equivalence as in the case of Fig. 3, we can always bound the energies to range to the maximal appearing number plus one. The last number represents all numbers outside the bound up to infinity.

Lemma 8 (Clever spectroscopy complexity). *Time complexity of computing winning budgets for the clever spectroscopy energy game $\mathcal{G}_\blacktriangle$ with capped energies is in $2^{\mathcal{O}(|\mathcal{P}|)}$.*

Proof. Out-degrees o in $\mathcal{G}_\blacktriangle$ can be bounded in $\mathcal{O}(|\mathcal{P}|)$, the whole game graph $|\succrightarrow_\blacktriangle| \in \mathcal{O}(|\dot{\rightarrow}| \cdot 2^{|\mathcal{P}|} + |\mathcal{P}|^2 \cdot 2^{|\mathcal{P}|})$, and game positions $|G_\blacktriangle| \in \mathcal{O}(|\mathcal{P}| \cdot 2^{|\mathcal{P}|})$. Inserting in the flattened version of Lemma 6 yields:

$$\begin{aligned} \mathcal{O}(|\succrightarrow_\blacktriangle| \cdot 2^{C_0 \cdot o}) &= \mathcal{O}((|\dot{\rightarrow}| \cdot 2^{|\mathcal{P}|} + |\mathcal{P}|^2 \cdot 2^{|\mathcal{P}|}) \cdot 2^{C_1 \cdot |\mathcal{P}|}) \\ &= \mathcal{O}((|\dot{\rightarrow}| + |\mathcal{P}|^2) \cdot 2^{C_2 \cdot |\mathcal{P}|}) \\ &= \mathcal{O}(|\dot{\rightarrow}| \cdot 2^{C_2 \cdot |\mathcal{P}|}). \end{aligned}$$

Deciding trace equivalence in nondeterministic systems is PSPACE-hard and will thus take at least exponential time. Therefore, the exponential time of the “clever” spectroscopy algorithm restricted to a finite spectrum is about as good as it may get, asymptotically speaking.

4.3 Equivalences and Distinguishing Formulas from Budgets

For completeness, let us briefly flesh out how to actually obtain equivalence information from the minimal attacker winning budgets $\text{Win}_\mathfrak{a}^{\min}((p, \{q\})_\mathfrak{a})$ we compute.

Definition 14. *For an antichain $Mn \subseteq \mathbf{En}$ characterizing an upper part of the energy space, the complement antichain $\overline{Mn} := \text{Min}(\mathbf{En}_\infty \cap (\{(\sup E') - (1, \dots, 1) \mid E' \subseteq Mn\} \cup \{e(i) \in \mathbf{En}_\infty \mid (e(i))_i = (\inf Mn)_i - 1 \wedge \forall j \neq i. (e(i))_j = \infty\}))$ has the complement energy space as its downset.*

$\text{Win}_d^{\max}((p, \{q\})_a) = \overline{\text{Win}_a^{\min}((p, \{q\})_a)}$ characterizes *all* preordering formula languages and thus equivalences defined in terms of expressiveness prices for p and q . This might contain multiple, incomparable, notions from the spectrum. Taking both directions, $\text{Win}_a^{\min}((p, \{q\})_a) \cup \text{Win}_a^{\min}((q, \{p\})_a)$, will thus characterize the finest intersection of equivalences to equate p and q .

If we just wonder which of the equivalences from the spectrum hold, we may establish this more directly by checking which of them are not dominated by attacker wins.

From the information, we can also easily build witness relations to certify that we return sound equivalence results. In particular, the pairs won with arbitrary attacker budgets, $\{(p, q) \mid (\infty, \infty, \infty, \infty, \infty, \infty) \in \text{Win}_d^{\max}((p, \{q\})_a)\}$ are a bisimulation. Similarly, the strategy formulas of Definition 9 can directly be computed to explain inequivalence.

If we use symbolic winning budgets capped as proposed at the end of Subsect. 4.2, the formula reconstruction will be harder and the $\overline{\text{Win}_a^{\min}((p, \{q\})_a)}$ might be below the maximal defender winning budgets if these exceed the bound. But this will not matter as long as we choose a cap beyond the natural numbers that characterize our spectrum.

5 Exploring Minimizations

Our algorithm can be used to analyze the equivalence structure of moderately-sized real-world transition systems. In this section, we take a brief look at its performance on the VLTS (“very large transition systems”) benchmark suite [18] and return to our initial Peterson example.

The energy spectroscopy algorithm has been added to the Linear-Time-Branching-Time Spectroscopy of [7] and can be tried on transition systems at <https://equiv.io/>.

Table 1 reports the results of running the implementation of [7] and this paper’s implementation in variants using the spectroscopy energy game \mathcal{G}_Δ and the clever spectroscopy energy game $\mathcal{G}_\blacktriangle$. We tested on the VLTS examples of up to 25,000 states and the Peterson example (Fig. 1). The table lists the \mathcal{P} -sizes of the input transition systems and of their bisimilarity quotient system \mathcal{P}/\sim_B . The spectroscopies have been performed on the bisimilarity quotient systems by constructing the game graph underneath positions comparing all pairs of enabledness-equivalent states. The middle three groups of columns list the resource usage for the three implementations using: the [7]-spectroscopy, the energy game \mathcal{G}_Δ , and the clever game $\mathcal{G}_\blacktriangle$. For each group, the first column reports traversed game size, and the second gives the time the spectroscopy took in seconds. Where the tests ran out of memory or took longer than five minutes (in the Java Virtual Machine with 8 GB heap space, at 4 GHz, single-threaded), the cells are left blank. The last three columns list the output sizes of state spaces reduced with respect to enabledness \sim_E , traces \sim_T , and simulation \sim_{IS} —as one would hope, all three algorithms returned the same results.

From the output, we learn that the VLTS examples, in a way, lack diversity: Bisimilarity \sim_B and trace equivalence \sim_T mostly coincide on the systems (third and penultimate column).

Concerning the algorithm itself, the experiments reveal that the computation time grows mostly linearly with the size of the game move graph. Our algorithm can deal with bigger examples than [7] (which fails at `peterson`, `vasy_10_56` and `cwi_1_2`, and takes more than 500s for `vasy_8_24`). Even where [7] has a smaller game graph (e.g. `cwi_3_14`), the exponential formula construction renders it slower. Also, the clever game graph $\succrightarrow_{\blacktriangle}$ indeed is much smaller than $\succrightarrow_{\triangle}$ for examples with a lot of nondeterminism such as `peterson`.

Table 1. Sample systems, sizes, and benchmark results.

system	\mathcal{P}	\mathcal{P}/\sim_B	[7]- \succrightarrow	t/s	$\succrightarrow_{\triangle}$	t/s	$\succrightarrow_{\blacktriangle}$	t/s	\mathcal{P}/\sim_E	\mathcal{P}/\sim_T	\mathcal{P}/\sim_{IS}
<code>peterson</code>	19	19			348,474	23.31	2,363	0.15	3	11	11
<code>vasy_0_1</code>	289	9	1,118	0.17	1,334	0.02	566	0.02	1	9	9
<code>vasy_1_4</code>	1,183	28	1,125	0.05	1,320	0.02	1,000	0.02	8	28	28
<code>vasy_5_9</code>	5,486	145	3,789	0.14	4,315	0.05	2,988	0.06	109	145	145
<code>vasy_8_24</code>	8,879	416	513,690	540.96	725,113	10.48	145,965	2.15	171	415	415
<code>vasy_8_38</code>	8,921	219	19,595	0.78	19,690	0.21	14,958	0.19	112	218	218
<code>vasy_10_56</code>	10,849	2,112					6,012,676	174.59	13	2,112	2,112
<code>vasy_18_73</code>	18,746	4,087									
<code>vasy_25_25</code>	25,217	25,217	100,866	1.15	0	0.32	0	0.33	25,217	25,217	25,217
<code>cwi_1_2</code>	1,952	1,132					22,723,369	384.13	9	1,132	1,132
<code>cwi_3_14</code>	3,996	62	14,761	2.48	25,666	0.28	18,350	0.3	2	62	62

Of those terminating, the heavily nondeterministic `cwi_1_2` is the most expensive example. As many coarse notions must record the nondeterministic options, this blowup is to be expected. If we compare to the best similarity algorithm by Ranzato and Tapparo [29], they report their algorithm SA to tackle `cwi_1_2` single-handedly. Like our implementation, the prototype of SA [29] ran out of memory while determining similarity for `vasy_18_73`. This is in spite of SA theoretically having optimal complexity and similarity being less complex (cubic) than trace equivalence, which we need to cover. The benchmarks in [29] failed at `vasy_10_56`, and `vasy_25_25`, which might be due to 2010's tighter memory requirements (they used 2 GB of RAM) or the degree to which bisimilarity and enabledness in the models is exploited.

6 Conclusion and Related Work

This paper has connected two strands of research in the field of system analysis: The strand of *equivalence games on transition systems* starting with Stirling's bisimulation game [7, 12, 32, 33] and the strand of *energy games for systems of bounded resources* [2, 10, 11, 14–16, 27, 30, 34].

The connection rests on the insight that levels of equivalence correspond to resources available to an attacker who tries to tell two systems apart. This parallel is present in recent work within the security domain [25] just as much as in the first thoughts on observable nondeterminism by Hennessy and Milner [24].

The paper has not examined the precise relationship of the games of Sect. 3 to the whole zoo of VASS, energy, mean-payoff, monotonic [1], and counter games. The spectroscopy energy game deviates slightly from common multi-energy games due to \min_D -updates and due to the attacker being energy-bound (instead of the defender). As the energies cannot be exhausted by defender moves, the game can also be interpreted as a VASS game [2, 10] where the attacker is stuck if they run out of energy. Our algorithm complexity matches that of general lower-bounded N -dimensional energy games [15]. Links between our declining energy games and other games from the literature might enable slight improvements of the algorithm. For instance, reachability in VASS games can turn polynomial [11].

In the strand of generalized game characterizations for equivalences [7, 12, 32], this paper extends applicability for real-world systems. The implementation performs on par with the most efficient similarity algorithm [29]. Given that among the hundreds of equivalence algorithms and tools most primarily address bisimilarity [19], a tool for coarser equivalences is a worthwhile addition. Although our previous algorithm [7] is able to solve the spectroscopy problem, its reliance on super-exponential partitions of the state space makes it ill-fit for transition systems with significant nondeterminism. In comparison, our new algorithm also needs one less layer of complexity because it determines equivalences without constructing distinguishing formulas.

These advances enable a spectroscopy of systems saturated by weak transitions. We can thus analyze weak equivalences such as in the example of Peterson’s mutex. For special weak equivalences without a strong counterpart such as branching bisimilarity [22], deeper changes to the modal logic are necessary [6].

The increased applicability has allowed us to exhaustively consider equivalences on the smaller systems of the widely-used VLTS suite [18]. The experiments reveal that the spectrum between trace equivalence and bisimilarity mostly collapses for the examined systems. It may often be reasonable to specify systems in such a way that the spectrum collapses. In a benchmark suite, however, a lack of semantic diversity can be problematic: For instance, otherwise sensible techniques like polynomial-time reductions [13] will not speed up language inclusion testing, and nuances of the weak equivalence spectrum [8] will falsely seem insignificant. One may also overlook errors and performance degradations that appear only for transition systems where equal traces do not imply equivalent branching behavior. We hope this blind spot does not affect the validity of any of the numerous studies relying on VLTS benchmarks.

Acknowledgments. This work benefited from discussion with Sebastian Wolf, with David N. Jansen, with members of the LFCS Edinburgh, and with the MTV research group at TU Berlin, as well as from reviewer comments.

Data Availability. Proofs and updates are to be found in the report version of this paper [5]. The Scala source is on GitHub: <https://github.com/benkeks/equivalence-fiddle/>. A webtool implementing the algorithm runs on <https://equiv.io/>. An artifact including the benchmarks is archived on Zenodo [4].

References

1. Abdulla, P.A., Bouajjani, A., D’orso, J.: Monotonic and downward closed games. *J. Log. Comput.* **18**(1), 153–169 (2008). <https://doi.org/10.1093/logcom/exm062>
2. Abdulla, P.A., Mayr, R., Sangnier, A., Sproston, J.: Solving parity games on integer vectors. In: D’Argenio, P.R., Melgratti, H. (eds.) *CONCUR 2013*. LNCS, vol. 8052, pp. 106–120. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40184-8_9
3. Aceto, L., Ingólfssdóttir, A., Larsen, K.G., Srba, J.: *Reactive Systems: Modelling, Specification and Verification*, chap. Modelling mutual exclusion algorithms, pp. 142–158. Cambridge University Press (2007). <https://doi.org/10.1017/CBO9780511814105.008>
4. Bisping, B.: Linear-time-branching-time spectroscopy v0.3.0 (2023). <https://doi.org/10.5281/zenodo.7870252>, archived on Zenodo
5. Bisping, B.: Process equivalence problems as energy games. Tech. rep., Technische Universität Berlin (2023). <https://doi.org/10.48550/arXiv.2303.08904>
6. Bisping, B., Jansen, D.N.: Linear-time-branching-time spectroscopy accounting for silent steps (2023). <https://doi.org/10.48550/arXiv.2305.17671>
7. Bisping, B., Jansen, D.N., Nestmann, U.: Deciding all behavioral equivalences at once: a game for linear-time-branching-time spectroscopy. *Logical Methods Comput. Sci.* **18**(3) (2022). [https://doi.org/10.46298/lmcs-18\(3:19\)2022](https://doi.org/10.46298/lmcs-18(3:19)2022)
8. Bisping, B., Nestmann, U.: Computing coupled similarity. In: Vojnar, T., Zhang, L. (eds.) *TACAS 2019*. LNCS, vol. 11427, pp. 244–261. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17462-0_14
9. Bisping, B., Nestmann, U.: A game for linear-time-branching-time spectroscopy. In: *TACAS 2021*. LNCS, vol. 12651, pp. 3–19. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72016-2_1
10. Brázdil, T., Jančar, P., Kučera, A.: Reachability games on extended vector addition systems with states. In: Abramsky, S., Gavioille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) *ICALP 2010*. LNCS, vol. 6199, pp. 478–489. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14162-1_40
11. Chaloupka, J.: Z-reachability problem for games on 2-dimensional vector addition systems with states is in P. In: Kučera, A., Potapov, I. (eds.) *RP 2010*. LNCS, vol. 6227, pp. 104–119. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15349-5_7
12. Chen, X., Deng, Y.: Game characterizations of process equivalences. In: Ramalingam, G. (ed.) *APLAS 2008*. LNCS, vol. 5356, pp. 107–121. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89330-1_8
13. Clemente, L., Mayr, R.: Efficient reduction of nondeterministic automata with application to language inclusion testing. *Logical Methods Comput. Sci.* **15**(1) (2019). [https://doi.org/10.23638/LMCS-15\(1:12\)2019](https://doi.org/10.23638/LMCS-15(1:12)2019)
14. Ehrenfeucht, A., Mycielski, J.: Positional strategies for mean payoff games. *Int. J. Game Theory* **8**(2), 109–113 (1979). <https://doi.org/10.1007/BF01768705>

15. Fahrenberg, U., Juhl, L., Larsen, K.G., Srba, J.: Energy games in multiweighted automata. In: Cerone, A., Pihlajasaari, P. (eds.) ICTAC 2011. LNCS, vol. 6916, pp. 95–115. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23283-1_9
16. Filiot, E., Hamel-de le Court, E.: Two-player boundedness counter games. In: Klin, B., Lasota, S., Muscholl, A. (eds.) 33rd International Conference on Concurrency Theory (CONCUR 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 243, pp. 21:1–21:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022). <https://doi.org/10.4230/LIPIcs.CONCUR.2022.21>
17. Fournet, C., Hoare, T., Rajamani, S.K., Rehof, J.: Stuck-free conformance. In: Alur, R., Peled, D.A. (eds.) CAV 2004. LNCS, vol. 3114, pp. 242–254. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27813-9_19
18. Garavel, H.: The VLTS benchmark suite (2017). <https://doi.org/10.18709/perscido.2017.11.ds100>, jointly created by CWI/SEN2 and INRIA/VASY as a CADP resource
19. Garavel, H., Lang, F.: Equivalence checking 40 years after: A review of bisimulation tools. In: Jansen, N., Stoelinga, M., van den Bos, P. (eds.) A Journey from Process Algebra via Timed Automata to Model Learning: Essays Dedicated to Frits Vaandrager on the Occasion of His 60th Birthday, pp. 213–265. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-15629-8_13
20. Geuvers, H., Jacobs, B.: Relating apartness and bisimulation. *Logical Methods Comput. Sci.* **17**(3) (2021). [https://doi.org/10.46298/LMCS-17\(3:15\)2021](https://doi.org/10.46298/LMCS-17(3:15)2021)
21. Glabbeek, R.J.: The linear time - branching time spectrum. In: Baeten, J.C.M., Klop, J.W. (eds.) CONCUR 1990. LNCS, vol. 458, pp. 278–297. Springer, Heidelberg (1990). <https://doi.org/10.1007/BFb0039066>
22. Glabbeek, R.J.: The linear time — Branching time spectrum II. In: Best, E. (ed.) CONCUR 1993. LNCS, vol. 715, pp. 66–81. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57208-2_6
23. van Glabbeek, R.J.: The linear time-branching time spectrum I: The semantics of concrete, sequential processes. In: Bergstra, J.A., Ponse, A., Smolka, S.A. (eds.) Handbook of Process Algebra, chap. 1, pp. 3–99. Elsevier, Amsterdam (2001). <https://doi.org/10.1016/B978-044482830-9/50019-9>
24. Hennessy, M., Milner, R.: On observing nondeterminism and concurrency. In: de Bakker, J., van Leeuwen, J. (eds.) ICALP 1980. LNCS, vol. 85, pp. 299–309. Springer, Heidelberg (1980). https://doi.org/10.1007/3-540-10003-2_79
25. Horne, R., Mauw, S.: Discovering ePassport Vulnerabilities using Bisimilarity. *Logical Methods Comput. Sci.* **17**(2) (2021). [https://doi.org/10.23638/LMCS-17\(2:24\)2021](https://doi.org/10.23638/LMCS-17(2:24)2021)
26. Kruskal, J.B.: The theory of well-quasi-ordering: A frequently discovered concept. *J. Combinatorial Theory, Ser. A* **13**(3), 297–305 (1972). [https://doi.org/10.1016/0097-3165\(72\)90063-5](https://doi.org/10.1016/0097-3165(72)90063-5)
27. Kupferman, O., Shamash Halevy, N.: Energy games with resource-bounded environments. In: Klin, B., Lasota, S., Muscholl, A. (eds.) 33rd International Conference on Concurrency Theory (CONCUR 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 243, pp. 19:1–19:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022). <https://doi.org/10.4230/LIPIcs.CONCUR.2022.19>
28. Peterson, G.L.: Myths about the mutual exclusion problem. *Inf. Process. Lett.* **12**, 115–116 (1981). [https://doi.org/10.1016/0020-0190\(81\)90106-X](https://doi.org/10.1016/0020-0190(81)90106-X)

29. Ranzato, F., Tapparo, F.: An efficient simulation algorithm based on abstract interpretation. *Inf. Comput.* **208**(1), 1–22 (2010). <https://doi.org/10.1016/j.ic.2009.06.002>
30. Reichert, J.: Reachability games with counters: Decidability and algorithms. Theses, École normale supérieure de Cachan - ENS Cachan (2015). <https://tel.archives-ouvertes.fr/tel-01314414>
31. Roscoe, A.W.: Revivals, stuckness and the hierarchy of CSP models. *J. Logic Algebraic Program.* **78**(3), 163–190 (2009). <https://doi.org/10.1016/j.jlap.2008.10.002>
32. Shukla, S.K., Hunt, H.B., Rosenkrantz, D.J.: HORNSAT, model checking, verification and games. In: Alur, R., Henzinger, T.A. (eds.) *CAV 1996*. LNCS, vol. 1102, pp. 99–110. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61474-5_61
33. Stirling, C.: Modal and temporal logics for processes. Department of Computer Science, University of Edinburgh, Tech. rep. (1993)
34. Velner, Y., Chatterjee, K., Doyen, L., Henzinger, T.A., Rabinovich, A., Raskin, J.F.: The complexity of multi-mean-payoff and multi-energy games. *Inf. Comput.* **241**, 177–196 (2015). <https://doi.org/10.1016/j.ic.2015.03.001>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

