Playing with Repetitions in Data Words

Anirban Majumdar

M.Sc. Thesis

Supervisor: M Praveen

Chennai Mathematical Institute

February 20, 2019

Acknowledgements

I would like to thank my supervisor, Prof. M Praveen, for introducing me to the logic of repeating values and for all the useful discussions that we have had.

I also wish to thank Prof. B Srivathsan who kindled my interest in the field of Automata Theory and Verification.

Finally, I would like to thank Prof. Madhavan Mukund, Prof. Aiswarya Cyriac, Prof. Amaldev Manuel, Prof. Mandayam Srivas and Prof. Narayan Kumar for the exciting courses in Logic, Automata Theory and Verification that I took under them.

Abstract

We study two-player games which build words over infinite alphabets, and we study the problem of checking the existence of winning strategies. These games are played by two players, who take turns in choosing valuations for variables ranging over an infinite data domain, thus generating multi-attributed data words. The winner of the game is specified by formulas in the Logic of Repeating Values, which can reason about repetitions of data values in infinite data words. It is known that checking whether one of the players has a winning strategy is undecidable in general and becomes decidable under some restrictions. Existing works do not address the case where the logic of repeating values can use nested formulas. We study this scenario and determine that decidability depends on whether or not nested formulas can refer to the future.

Contents

Chapter 1

Introduction

Model checking is a method for formally verifying finite or infinite state systems. Given a model of a system, in verification, we encounter the following problem: we check whether the model satisfies a given specification. For example, we could check whether a system satisfies safety requirement *i.e.* absence of deadlocks etc. For this we need a precise and unambiguous statement of the specification to be examined. To this end, specifications about the system are often expressed as formulas in temporal logic (LTL) and the task is then to check whether the formula is satisfiable in the given structure. This is known as *model checking problem.*

In some situations, checking satisfiability is not of much help. Suppose we want to specify two properties of a coffee machine. The first property is that when a *coffee button* is pressed, coffee should be produced in the next step. The second property is that when a *stop button* is pressed, no coffee should be produced in the next step. We want to specify the conjunction of these two conditions. This has a satisfying assignment: neither button is ever pressed and no coffee is ever produced. But this does not give any intuition of the conflict of the two conditions. This situation can be tackled if we consider a two player game between, say environment and system. Suppose the value of *coffee button* and *stop button* be chosen by environment and the value of *coffee produced* be chosen by system. Then we would like to find if system can play in such a way such that it can win (*i.e.* both the conditions are satisfied) irrespective how environment plays. This is known as *realizability problem.* In this example, we can see that system cannot ensure that (we say, system does not have any winning strategy). The realizability problem on traces has been extensively studied for Boolean variables under LTL definable properties, starting from [12]. In this thesis, we study realizability problem for specifications in an extension of LTL.

Data words are words over an infinite alphabet (on the contrary to the ones accepted by finite automata or Büchi automata where the alphabet is finite). Data words appear in many contexts, for example as abstractions of timed words, runs of counter automata etc. Characterisation of languages of data words have been studied; both automata structure [11, 13] and logics [2, 5, 8, 6] have been proposed for such languages. We want to express properties like whether a data value assigned to a variable at a certain position is repeated locally (eg, in the previous position) or remotely (*e.g.*, in any position in future). To this end, *Logic of Repeating Values* (LRV) has been introduced [3] which is an extension of linear temporal logic (LTL) with the above tests.

The satisfiability problem for LRV has been studied in [3, 4]. There it has been proved that this problem is inter-reducible with the reachability problem for Vector Addition Systems with States (VASS). In [7], the authors introduce two player games on such structures. In this, some data values are chosen by system player and some by environment player. The reactive system has to satisfy a specified property, given as a logical formula over data words. Now they consider the *realizability problem.* Recall, the *realizability* problem asks whether it is possible that there exists a system that always satisfies the specified property, irrespective of what the environment does.

In [4, 7], the authors also consider some fragments of LRV. In [7], the authors show that if we consider the fragment of LRV where no future obligations, inequality constraints and nested formulas are allowed (in notation, LRV[$\top$, $\approx$, $\leftarrow$], formally defined later), then the above mentioned realizability problem is undecidable. Then they define single-sided game where environment has only boolean variables and system has data variables. They also show that in the above fragment the problem becomes decidable if we consider such single-sided games. They also show that instead of past obligations, if we allow only future obligations (LRV[$\top$, $\approx$, $\rightarrow$]), then even the single-sided games are undecidable.

In this thesis, we try to extend some results from[7] for some other fragments of LRV. More specifically, in [7], the authors do not show any result for the fragment where nested formulas are allowed (*i.e.* , the logic LRV[$\approx$, $\leftarrow$]). In this thesis, we show that on this fragment, the realizability problem for single sided games are undecidable. But again, if we restrict the nested formulas to some special kind, then the problem is reducible to single sided VASS game.

Thesis Outline: In chapter 2, we present the preliminaries which we will need throughout the thesis. We define the main logic (LRV), its different fragments we are interested in. We also present basic definitions of the VASS games, and counter machines which we shall use for our decidability and undecidability results. Then we move to proving our main results in this thesis. In chapter 3, as described above, first we show that the realizability problem on LRV$[\Psi, \approx, \leftarrow]$ (where Ψ is defined in the chapter 3 itself - it is basically a restriction for nested formulas) is decidable; this result is proved using the same idea used in section 5 of [7]. And then we move to proving the undecidability of the realizability problem for LRV$[\approx, \leftarrow]$ in section 2 of chapter 3, which is also similar to the proof used in section 6 of [7]. One can read [7] to get a better understanding on the proof techniques; but for completeness, we briefly explain here the relevant two sections from that paper and add it in the appendix.

Chapter 2

Preliminaries

In this chapter, we first formally define the syntax and semantics of the logic of repeating values and its fragments [4, 7]. Then we shall define the game of repeating values which is our main point of interest. Then we shall quickly mention counter machines and parity games on integer vectors which we will need for proving the decidability and undecidability results in the following chapters. Finally, we shall give an outline of the problems we are interested in and the results from the literature and also some extensions of those which we shall prove in the next chapters.

Let $\mathbb{Z}$ denote the set of integers and $\mathbb{N}$ the set of non-negative integers. For an alphabet Σ, Σ^* denotes the set of finite words over Σ, Σ^+ denotes the set of non-empty finite words over Σ and by Σ^ω we denote the set of infinite words over Σ. For a finite word $u \in \Sigma^*$, $|u|$ denotes the length of u. For any set S, we denote by $\mathcal{P}(S)$ the set of all subsets of S (*i.e.* the powerset of S) and by $\mathcal{P}^+(S)$ the set of all non-empty subsets of S.

2.1 Logic of repeating values

This logic is an extension of the classical propositional linear temporal logic (LTL) which can also reason about repetitions of data values from an infinite domain. Let $BVARS = \{q, p, \ldots\}$ be a countably infinite set of boolean variables taking values from $\{\top, \bot\}$ and $DVARS = \{x, y, \ldots\}$ be a countably infinite set of data variables taking values from an infinite domain $\mathbb{D}$. The boolean variables could be simulated by data variables but for convenience we keep them separate. We denote by LRV the logic whose formulas are defined as follows:

$$\begin{aligned}\varphi ::= {} & q \mid x \approx \mathsf{X}^j y \mid x \approx \langle \varphi ? \rangle y \mid x \not\approx \langle \varphi ? \rangle y \mid x \approx \langle \varphi ? \rangle^{-1} y \\ & \mid x \not\approx \langle \varphi ? \rangle^{-1} y \mid \varphi \wedge \varphi \mid \neg \varphi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \mathsf{X}^{-1}\varphi \\ & \mid \varphi \mathsf{S} \varphi \ , \text{ where } q \in \mathit{BVARS},\ x, y \in \mathit{DVARS},\ j \in \mathbb{Z}\end{aligned}$$

A *valuation* is the union of a mapping from BVARS to $\{\top, \bot\}$ and a mapping from DVARS to $\mathbb{D}$. A *model* is a finite or infinite sequence σ of valuations and $\sigma(i)$ denotes the i^{th} valuation in σ, where $i \in \mathbb{N} \setminus \{0\}$. For any model σ and position $i \in \mathbb{N} \setminus \{0\}$, the satisfaction relation $\models$ is defined inductively as follows.

$$\begin{aligned}
\sigma, i \models q &: \sigma(i)(q) = \top \\
\sigma, i \models x \approx \mathsf{X}^j y &\text{ iff } 1 \leq i + j \leq |\sigma|,\ \sigma(i)(x) = \sigma(i+j)(y) \\
\sigma, i \models x \approx \langle \varphi ? \rangle y &\text{ iff } \exists j > i \text{ s.t. } \sigma(i)(x) = \sigma(j)(y),\ \sigma, j \models \varphi \\
\sigma, i \models x \not\approx \langle \varphi ? \rangle y &\text{ iff } \exists j > i \text{ s.t. } \sigma(i)(x) \neq \sigma(j)(y),\ \sigma, j \models \varphi \\
\sigma, i \models x \approx \langle \varphi ? \rangle^{-1} y &\text{ iff } \exists j < i \text{ s.t. } \sigma(i)(x) = \sigma(j)(y),\ \sigma, j \models \varphi \\
\sigma, i \models x \not\approx \langle \varphi ? \rangle^{-1} y &\text{ iff } \exists j < i \text{ s.t. } \sigma(i)(x) \neq \sigma(j)(y),\ \sigma, j \models \varphi \\
\sigma, i \models \varphi \wedge \varphi' &\text{ iff } \sigma, i \models \varphi \text{ and } \sigma, i \models \varphi' \\
\sigma, i \models \neg \varphi &\text{ iff } \sigma, i \not\models \varphi \\
\sigma, i \models \mathsf{X}\varphi &\text{ iff } \sigma, i+1 \models \varphi \\
\sigma, i \models \mathsf{X}^{-1}\varphi &\text{ iff } i > 1 \text{ and } \sigma, i-1 \models \varphi \\
\sigma, i \models \varphi \mathsf{U} \varphi' &\text{ iff } \exists i \leq j \text{ such that } \sigma, j \models \varphi' \text{ and for every } i \leq l < j, \text{ we have } \sigma, l \models \varphi \\
\sigma, i \models \varphi \mathsf{S} \varphi' &\text{ iff } \exists j \leq i \text{ such that } \sigma, j \models \varphi' \text{ and for every } j < l \leq i, \text{ we have } \sigma, l \models \varphi
\end{aligned}$$

for $q \in \mathit{BVARS}$, $x, y \in \mathit{DVARS}$ and $j \in \mathbb{Z}$.

Note that the temporal operators next (X), previous (X^{-1}), until (U) and since (S) and the boolean connectives are defined in the usual way. We write $\sigma \models \varphi$ if $\sigma, 1 \models \varphi$.

Intuitively, the formula $x \approx \mathsf{X}^j y$ tests that the data value mapped to the variable x at the current position repeats in the variable y after j positions. We use the notation $\mathsf{X}^i x \approx \mathsf{X}^j y$ as an abbreviation for the formula $\mathsf{X}^i(x \approx \mathsf{X}^{j-i} y)$ (without loss of generality assume $i \leq j$). The formula $x \approx \langle \varphi ? \rangle y$ tests that the data value mapped to x at the current position repeats in y at a future position that satisfies the nested formula φ. Similarly the formula $x \not\approx \langle \varphi ? \rangle y$ tests that the data value mapped to x at the current position is not the same as the data value of y at a future position that satisfies the nested formula φ. The formulas $x \approx \langle \varphi ? \rangle^{-1} y$ and $x \not\approx \langle \varphi ? \rangle^{-1} y$ are similar but test for repetitions of data values in past positions.

Symbol	Meaning
$\top$	φ has to be $\top$ in $x \approx \langle\varphi?\rangle y$ (no nested formulas)
$\approx$	disequality constraints ($x \not\approx \langle\varphi?\rangle y$ or $x \not\approx \langle\varphi?\rangle^{-1} y$) are not allowed
$\rightarrow$	past obligations ($x \approx \langle\varphi?\rangle^{-1} y$ or $x \not\approx \langle\varphi?\rangle^{-1} y$) are not allowed
$\leftarrow$	future obligations ($x \approx \langle\varphi?\rangle y$ or $x \not\approx \langle\varphi?\rangle y$) are not allowed

Table 2.1: Restrictions of LRV

Now we shall define some fragments of the logic. We append symbols to LRV for denoting syntactic restrictions as shown in the Table 2.1. For example, LRV$[\top, \approx, \rightarrow]$ denotes the fragment of LRV in which nested formulas, disequality constraints and past obligations are not allowed. For simplicity in notation, we replace $\langle\top?\rangle$ with $\Diamond$ in formulas. E.g., we write $x \approx \langle\top?\rangle y$ as $x \approx \Diamond y$.

2.2 Game of repeating values

The game of repeating values [7] is played between two players, called environment and system. Informally, the set of variables are partitioned into two sets, one for each player. The game starts with a move of the environment player. In the first round, environment chooses a valuation for its variables. In the next round, system player responds by choosing a valuation for its variables. In the next round, again environment chooses a valuation and so on. This way environment and system keeps on playing forever one after another and builds an infinite model. The winning condition is given by a LRV formula; system wins if the infinite model satisfies the formula.

The following example illustrates the utility of this game. Consider a scenario in which the system is trying to schedule tasks on processors. The number of tasks can be unbounded and task identifiers can be data values. Suppose the variable init carries identifiers of tasks that are initialized and proc carries identifiers of tasks that are processed. Then the formula $G\,(\text{proc} \approx \Diamond^{-1}\text{init})$ specifies that all tasks that are processed must have been initialized beforehand. Let the variable log carry identifiers of tasks that have been processed and are being logged into an audit table. The formula $G\,(\text{proc} \approx \mathsf{X}\ \text{log})$ specifies that all processed tasks are logged into the audit table in the next step. Suppose there is a Boolean variable lf belonging to the environment that denotes whether the logger is working or not. The formula $G\,(\neg\text{lf} \ \Rightarrow\ \neg(\text{log} \approx \mathsf{X}^{-1}\ \text{proc}))$ specifies that if lf is false (denoting that the logger is not working), then the logger can not put the task that was processed in the previous step into the audit table in this step. The combination of the last two specifications is not realizable by any system since as soon as the system processes a task, the environment can make the logger non-functional in the next step. This can be algorithmically determined by the fact that for the conjunction of the last two formulas, there is no winning strategy for system in the game of repeating values.

Now let us formally define the game of repeating values. The set $BVARS$ (resp. $DVARS$) is partitioned as $BVARS^e$ (resp. $DVARS^e$), $BVARS^s$ (resp. $DVARS^s$), owned by environment and system respectively. Let $B\Upsilon^e$ (resp. $D\Upsilon^e$, $B\Upsilon^s$, $D\Upsilon^s$) be the set of all mappings $bv^e : BVARS^e \to \{\top, \bot\}$ (resp., $dv^e : DVARS^e \to \mathbb{D}$, $bv^s : BVARS^s \to \{\top, \bot\}$, $dv^s : DVARS^s \to \mathbb{D}$). Given two mappings $v_1 : V_1 \to \mathbb{D} \cup \{\top, \bot\}$, $v_2 : V_2 \to \mathbb{D} \cup \{\top, \bot\}$ for disjoint sets of variables V_1, V_2, we denote by $v = v_1 \oplus v_2$ the mapping defined as $v(x_1) = v_1(x_1)$ for all $x_1 \in V_1$ and $v(x_2) = v_2(x_2)$ for all $x_2 \in V_2$. Let Υ^e (resp., Υ^s) be the set of mappings $\{bv^e \oplus dv^e \mid bv^e \in B\Upsilon^e, dv^e \in D\Upsilon^e\}$ (resp. $\{bv^s \oplus dv^s \mid bv^s \in B\Upsilon^s, dv^s \in D\Upsilon^s\}$).

The first round of a game of repeating values is begun by environment choosing a mapping $v_1^e \in \Upsilon^e$, to which system responds by choosing a mapping $v_1^s \in \Upsilon^s$. Then environment continues with the next round by choosing a mapping from Υ^e and so on. The game continues forever and results in an infinite model $\sigma = (v_1^e \oplus v_1^s)(v_2^e \oplus v_2^s) \cdots$. The winning condition is given by a LRV formula φ — system wins iff $\sigma \models \varphi$.

Now let us define strategy of a player. Let Υ be the set of all valuations. For any model σ and $i > 0$, let $\sigma \upharpoonright i$ denote the valuation sequence $\sigma(1) \cdots \sigma(i)$, and $\sigma \upharpoonright 0$ denote the empty sequence. A strategy for environment is a mapping $te : \Upsilon^* \to \Upsilon^e$. A strategy for system is a mapping $ts : \Upsilon^* \cdot \Upsilon^e \to \Upsilon^s$. We say that environment plays according to a strategy te if the resulting model $(v_1^e \oplus v_1^s)(v_2^e \oplus v_2^s) \cdots$ is such that $v_i^e = te(\sigma \upharpoonright (i-1))$ for all positions $i \in \mathbb{N} \setminus \{0\}$. System plays according to a strategy ts if the resulting model is such that $v_i^s = ts(\sigma \upharpoonright (i-1) \cdot v_i^e)$ for all positions $i \in \mathbb{N} \setminus \{0\}$. A strategy ts for system is winning if system wins all games that she plays according to ts, irrespective of the strategy used by environment.

We say a game is single-sided if the environment player has only boolean variables while the system player has data variables. Later we shall show some decidability and undecidability results on such games of repeating values.

2.3 2-Counter Machine

A 2-counter machine is described by a 6-tuple $(Q, \delta, q_{init}, q_{fin}, c_1, c_2)$, where Q is a finite set of states, q_{init} is the initial state, q_{fin} is the final state, c_1, c_2 are two counters and δ is a finite set of transitions which are of the form (q_1, a, q_2), where $q_1, q_2 \in Q$ and a is one of the following actions: (1) '$c_i - -$',

(2) 'c_i++', or (3) '$c_i = 0?$' for $i \in \{1,2\}$. Intuitively, transitions can either increase or decrease a counter or test whether the counter value is 0. A configuration of the 2-counter machine is a triple (q, n_1, n_2) where $q \in Q$ and $n_1, n_2 \in \mathbb{N}$ denote the current counter values. The transition relation $\rightarrow$ on configurations is defined as follows. We have $(q, n_1, n_2) \rightarrow (q', n_1', n_2')$ iff one of the following is true:

(1) $(q, c_i++, q') \in \delta$ for $i \in \{1,2\}$ and $n_i' = n_i + 1$, $n_{3-i}' = n_{3-i}$;

(2) $(q, c_i--, q') \in \delta$ for $i \in \{1,2\}$ and $n_i > 0$, $n_i' = n_i - 1$, $n_{3-i}' = n_{3-i}$;

(3) $(q, c_i = 0?, q') \in \delta$ for $i \in \{1,2\}$ and $n_i = 0$, $(n_1', n_2') = (n_1, n_2)$.

A *run* of a counter machine is a (possibly infinite) sequence of configurations $s_0, s_1, \ldots$ such that $s_0 \rightarrow s_1 \rightarrow s_2 \ldots$. We say that a counter machine is *deterministic* if for every configuration (q, n_1, n_2) there exists at most one configuration (q', n_1', n_2') so that $(q, n_1, n_2) \rightarrow (q', n_1', n_2')$. Given a 2-counter machine, the **reachability problem** is to determine if there exists a sequence of transitions of the 2-counter machine starting from the configuration $(q_{init}, 0, 0)$ and ending at the configuration (q_{fin}, n_1, n_2) for some $n_1, n_2 \in \mathbb{N}$. It is known that the reachability problem for 2-counter machines is undecidable [10]. For our undecidability results we will use the above fact for a deterministic 2-counter machine. To simplify our undecidability results we further assume, without any loss of generality, that there exists a transition $\hat{t} = (q_{fin}, c_1++, q_{fin}) \in \delta$.

Theorem 1. *[10] The reachability problem for 2-counter machine is undecidable.*

2.4 Lossy Counter Machine

A n-counter machine is defined [9] in almost the same way as above except the fact that now we have n counters $c_1, c_2, \ldots, c_n$. In this case, a configuration is a tuple of the form $(q, m_1, m_2, \ldots, m_n)$ where $q \in Q$ and $m_1, m_2, \ldots, m_n \in \mathbb{N}$ denote the current counter values.

Now let us define a transition relation $\xrightarrow{s}$ on the configurations of a counter machine as follows. We have $(q, m_1, m_2, \ldots, m_n) \xrightarrow{s} (q', m_1', m_2', \ldots, m_n')$ iff one of the following is true:

(1) $(q, m_1, m_2, \ldots, m_n) = (q', m_1', m_2', \ldots, m_n')$;

(2) $q = q'$ and $\sum_{i=1}^{n} m_i > \sum_{i=1}^{n} m_i'$.

Intuitively, the relation $\xrightarrow{s}$ says that either the configuration remains unchanged or the sum of all the counter values decreases. We say a relation $\xrightarrow{l}$ is a lossiness relation iff $\xrightarrow{l} \subseteq \xrightarrow{s}$.

A lossy counter machine (LCM) is described by a counter machine and a lossiness relation $\xrightarrow{l}$. We define the lossy transition relation $\Rightarrow$ on configurations of a LCM as follows. We have $\gamma_1 \Rightarrow \gamma_2$ iff $\exists\ \gamma_3, \gamma_4$ such that $\gamma_1 \xrightarrow{l} \gamma_3 \rightarrow \gamma_4 \xrightarrow{l} \gamma_2$. A *run*, defined as before, is a (possibly infinite) sequence of configurations $s_0, s_1, \ldots$ such that $s_0 \Rightarrow s_1 \Rightarrow s_2 \ldots$.

For our undecidability result, we shall consider $\xrightarrow{l}$ to be **reset lossiness** defined as follows. Whenever there is a zero test for a counter, that counter can suddenly become zero. Formally, $(q, m_1, m_2, \ldots, m_n) \xrightarrow{l} (q', m_1', m_2', \ldots, m_n')$ iff one of the following is true:

(1) $(q, m_1, m_2, \ldots, m_n) = (q', m_1', m_2', \ldots, m_n')$;

(2) $q = q', m_i' = 0$ for some i , $m_j' = m_j$ for all $j \neq i$ and $\exists\ q''$ such that $(q, c_i = 0?, q'') \in \delta$.

We define two restrictions of a n-counter machine. We say a counter machine is *input-bounded* if from any configuration, the sum of the counter values of every reachable configuration is bounded by (*i.e.* less than or equal to) the sum of the counter values in the initial configuration. We say a counter machine is *strongly-cyclic* if in every run from any configuration, the initial state q_{init} is visited infinitely often.

Now consider the following problem. Given a strongly-cyclic, input-bounded LCM M with 5 counters and initial state q_{init}, the problem is to determine if there exist initial counter values $m_1, \ldots, m_5 \in \mathbb{N}$ and a state $q \in Q$ such that there is an infinite run starting at the configuration $(q, m_1, \ldots, m_5)$. Let us call this the *existential infinite run problem.* This problem is known to be undecidable. We use this result for proving our undecidability result in a later chapter.

Theorem 2. *[9] The existential infinite run problem for a strongly-cyclic, input-bounded LCM is undecidable.*

2.5 Parity games on integer vectors

We define the parity games on Vector Addition Systems with States (VASS) [1]. The game is played between two players, environment and system. A VASS game is a tuple $(Q = Q^e \uplus Q^s, C, T, \pi)$ where Q is a finite set of states,

Q^e are states belonging to environment player and Q^s are states belonging to system player, C is a finite set of counters, T is a finite set of transitions and $\pi : Q \to \mathbb{N} \setminus \{0\}$ is a colouring function that assigns a natural number to each state. A transition in T is of the form (q, op, q') where $q, q' \in Q$ are called the source and target states respectively and op is an operation of the form $x++$ (increment a counter), $x--$ (decrement a counter) or nop (counter values remain unchanged), where $x \in C$ is a counter. We say that a transition of a VASS game belongs to a player if its source state belongs to that player. We call a VASS game *single-sided* if every environment transition is of the form (q, nop, q'). We also assume that there is no dead-lock, *i.e.* from every state, we have at least one outgoing transition.

A configuration of the VASS game is a tuple $(q, \vec{n})$ where $q \in Q$ is a state and $\vec{n} \in \mathbb{N}^C$ is a valuation for the counters. A play of the VASS game begins at a designated initial configuration. The owner of the state of the current configuration (say $(q, \vec{n})$) plays one of the outgoing transitions (say (q, op, q')) and changes the configuration to $(q', \vec{n'})$, where $\vec{n'}$ is obtained from $\vec{n}$ iff one of the following is true:

(1) op is $x++$; $\vec{n'}(x) = \vec{n}(x) + 1$ and $\vec{n'}(y) = \vec{n}(y)$;

(2) op is $x--$; $\vec{n'}(x) = \vec{n}(x) - 1$ and $\vec{n'}(y) = \vec{n}(y)$;

(3) op is nop; $\vec{n'} = \vec{n}$.

where y is any counter distinct from x. We denote this update as $(q, \vec{n}) \xrightarrow{(q,op,q')} (q', \vec{n'})$. The play is then continued similarly by the owner of the state of the next configuration. The counter values should be non-negative throughout the play. From any configuration, a transition that wants to decrement a counter is enabled only when that counter has a positive value before the transition. We say a VASS game single-sided if environment cannot change the value of the counters.

The game continues forever and results in an infinite sequence of configurations $(q_0, \vec{n_0})(q_1, \vec{n_1}) \cdots$. The play is winning for System if the maximum colour occurring infinitely often in $\pi(q_0)\pi(q_1)\pi(q_2) \cdots$ is even. Without loss of generality we can assume that from any configuration, at least one transition is enabled (otherwise, we can add extra states and transitions to create a loop ensuring that the owner of the deadlocked configuration loses). For our purpose, we shall use a generalized form of transitions $q \xrightarrow{\vec{u}} q'$ where $\vec{u} \in \mathbb{Z}^C$, to indicate that each counter c should be updated by adding $\vec{u}(c)$. Such VASS games can be effectively translated into ones of the form defined above preserving winning regions.

A strategy se for environment in a VASS game is a mapping $se : (Q \times \mathbb{N}^C)^* \cdot (Q^e \times \mathbb{N}^C) \to T$ such that for any $\gamma \in (Q \times \mathbb{N}^C)^*$, $q^e \in Q^e$ and $\vec{n} \in \mathbb{N}^C$,

$se(\gamma \cdot (q^e, \vec{n}))$ is a transition that is enabled whose source state is q^e. Similarly we define strategy ss for system player ($ss : (Q \times \mathbb{N}^C)^* \cdot (Q^s \times \mathbb{N}^C) \to T$). We say that environment plays according to a strategy se if the resulting sequence of configurations $(q_0, \vec{n_0})(q_1, \vec{n_1}) \cdots$ is such that for all $i \in \mathbb{N}$, $q_i \in Q^e$ implies $(q_i, \vec{n_i}) \xrightarrow{se((q_0,\vec{n}_0)(q_1,\vec{n}_1)\cdots(q_i,\vec{n}_i))} (q_{i+1}, \vec{n}_{i+1})$. The notion is defined similarly for the to system player. We say a strategy ss for system is winning if system wins all the plays that it plays according to ss, irrespective of the strategy used by environment.

Given a single-sided VASS game and an initial configuration, it is decidable to check whether system has a winning strategy from that configuration in the game [1]. We shall use this decidability result in our proofs.

Theorem 3. *[1] The existential winning strategy problem for a single-sided VASS game is decidable.*

Chapter 3

Realizability of LRV With Nested Formulas

In [7, Section 5] (See A.1), authors prove that single-sided LRV$[\top, \approx, \leftarrow]$ games are decidable. Now we want to examine whether this result remains true if we also allow nested formulas in the logic. It turns out that it becomes undecidable. But if the nested formulas are only of some special form, then it remains decidable. In this chapter, we prove these two results.

Consider the following syntax which is a restriction of the main logic:

$$\psi ::= q \mid x \approx \mathsf{X}^j y \mid x \approx \langle \psi ? \rangle^{-1} y \mid \psi \wedge \psi \mid \neg \psi,$$
$$\text{where } q \in \mathit{BVARS},\ x, y \in \mathit{DVARS},\ j \leq 0.$$

Let us denote by Ψ the set of all the formulas that can be generated from this grammar. Let us denote by LRV$[\Psi, \approx, \leftarrow]$ the restriction of the main logic where the nested formulas are from the set Ψ *i.e.* for any formula of the form $x \approx \langle \psi ? \rangle^{-1} y$, we have $\psi \in \Psi$.

3.1 Decidability of single-sided LRV$[\Psi, \approx, \leftarrow]$ games

The proof is similar to the proof we saw for single-sided LRV$[\top, \approx, \leftarrow]$ games in section A.1. Here also we build symbolic model using frames, but the constraints of a frame and the properties are bit different in this case.

Suppose we are given a formula φ. Let Φ be the set $\{\psi \mid \exists\ x, y \in \mathit{DVARS}^{\varphi}$ s.t. $x \approx \langle \psi ? \rangle^{-1} y$ is a sub-formula in $\varphi\}$ to keep track of all the variables taking same data value in different positions. We also add $\top$ to the set Φ. Let $\hat{\Phi} := \Phi \cup \{\top\}$. Let $\Phi_{\mathcal{AT}}$ denotes the set of atomic constraints from the closure (under sub-formulas) of Φ. Note that the closure of a set of formulas is the union of the closure of the formulas. And the closure of a formula φ (denoted by $cl(\varphi)$) is defined in the usual way:

$cl(\varphi)$ is the smallest set satisfying the following properties:

- $\varphi \in cl(\varphi)$
- $x \approx \langle \varphi_1? \rangle^{-1} y \in cl(\varphi) \Rightarrow \varphi_1 \in cl(\varphi)$.
- $\varphi_1 \wedge \varphi_2 \in cl(\varphi) \Rightarrow \varphi_1 \in cl(\varphi)$ and $\varphi_2 \in cl(\varphi)$.
- $\neg\varphi_1 \in cl(\varphi) \Rightarrow \varphi_1 \in cl(\varphi)$.

For example, take $\varphi := x \approx \langle (\psi_1 \wedge \psi_2)? \rangle^{-1} y$ where $\psi_1 = x_1 \approx \langle \psi_3? \rangle^{-1} y_1$; $\psi_2 = x_2 \approx \mathsf{X}^{-2} y_2$; $\psi_3 = x_3 \approx \Diamond^{-1} y_3$.

$\Phi = \{\psi_1 \wedge \psi_2, \psi_3, \top\}$

$\hat{\Phi} = \{\psi_1 \wedge \psi_2, \psi_3, \top\}$

$\Phi_{\mathcal{AT}} = \{\psi_1, \psi_2, \psi_3, \top\}$

Now we define a Repetition Pattern corresponding to a variable x (we shall denote it by I_x) to be a subset of $\mathcal{P}(DVARS^{\varphi}) \times \mathcal{P}(\Phi_{\mathcal{AT}})$. We denote by RP the set of all such subsets; *i.e.* $RP = \mathcal{P}(\mathcal{P}(DVARS^{\varphi}) \times \mathcal{P}(\Phi_{\mathcal{AT}}))$.

The intuition is that a Repetition Pattern I_x stores past positions where the data value of x at the current position has appeared and the formulas satisfied in those positions. This is explained later in a formal way.

Given a formula in LRV[Ψ, $\approx$, $\leftarrow$], we first replace all the sub-formulas of the form $x \approx \mathsf{X}^{-j} y$ with $\mathsf{X}^{-j}(y \approx \mathsf{X}^{j} x)$ if $j > 0$ for simplicity. For a formula φ obtained after such replacements, let l be the maximum i such that a term of the form $\mathsf{X}^i x$ appears in φ. We call l the 'X-length' of φ. Let $BVARS^{\varphi}$ and $DVARS^{\varphi}$ denote the set of boolean and data variables respectively used in φ. Let Ω_l^{φ} be the set of constraints of the form $\mathsf{X}^i q$, $\mathsf{X}^i x \approx \mathsf{X}^j y$ or $\mathsf{X}^i(x \approx \langle \psi? \rangle^{-1} y)$, where $q \in BVARS^{\varphi}$, $x, y \in DVARS^{\varphi}$ and $i, j \in \{0, \ldots, l\}$ and $\psi \in \hat{\Phi}$.

For $e \in \{0, \ldots, l\}$, an (e, φ)-frame is a (e+2)-tuple such that the first component is a set of constraints $fr \subseteq \Omega_l^{\varphi}$, and each of the other $(e + 1)$ components is a function from $DVARS^{\varphi}$ to RP. In notation, $fr \in \mathcal{P}(\Omega_l^{\varphi}) \times (RP^{DVARS^{\varphi}})^{(e+1)}$. If a formula $\mathsf{X}^i q \in fr[1]$, then we say that q is true at level i of fr. Now define the set

$$\mathcal{G}_i^{fr} := \{\varphi' \in \{\mathcal{B}(\Phi_{\mathcal{AT}}) \cup \top\} \cap \hat{\Phi} \mid \{\mathsf{X}^i(\varphi'') \in fr[1], \text{ for any } \varphi''\} \models_{\mathcal{B}} \varphi'\}$$

where $\mathcal{B}(S)$ denotes the set of all possible boolean combinations of the formulas in S and $\models_{\mathcal{B}}$ stands for boolean satisfaction. Intuitively this is the set of nested formulas which are true at level i in fr.

Past obligations and equivalence classes are defined in the same way as before. Given a formula φ in LRV$[\Psi, \approx, \leftarrow]$ of X-length l, for $e \in \{0, \dots, l\}$, an (e, φ)-frame fr, $i \in \{0, \dots, e\}$ and a variable x, the equivalence class of x at level i in fr is $[(x, i)]_{fr} = \{y \in DVARS^{\varphi} \mid \mathsf{X}^i x \approx \mathsf{X}^i y \in fr[1]\}$. The set of past obligations of the variable x at level i in fr is the set $\mathsf{PO}_{fr}(x, i) = \{(y, \psi) \in DVARS^{\varphi} \times \hat{\Phi} \mid \mathsf{X}^i(x \approx \langle \psi ? \rangle^{-1} y) \in fr[1]\}$.

We say that a set of past obligations $\mathsf{PO}_{fr}(w, i) = O_w$ (say) is *matched* to a $\mathsf{Repetition\ Pattern}$ $I_w \in RP$ if there exists a mapping $g : O \to I$ such that:

- $\forall (x, \psi) \in O_w$, $x \in g((x, \psi))[1]$
- $\forall (x, \psi) \in O_w$, $g((x, \psi))[2] \models_{\mathcal{B}} \psi$, and
- $\bigcup_{x:(x,\psi) \in O_w} \{x\} = \bigcup_{J \in I_w} J[1]$.

Intuitively, the first condition says that the data value of w at the current position repeats in past at x. The second condition says that the corresponding past position should satisfy ψ. The last condition here says that the set of variables in O should be the same as the set of variables occurring in the first components of I.

The frames should satisfy certain conditions. These are similar to those in A.1 with some modifications and some extra conditions. The conditions an (e, φ)-frame fr should satisfy are the following:

(F0) For all constraints $\mathsf{X}^i q, \mathsf{X}^i x \approx \mathsf{X}^j y, \mathsf{X}^i(x \approx \langle \psi ? \rangle^{-1} y) \in fr[1]$, $i, j \in \{0, \dots, e\}$.

(F1) For all $i \in \{0, \dots, e\}$ and $x \in DVARS^{\varphi}$, $\mathsf{X}^i x \approx \mathsf{X}^i x \in fr[1]$.

(F2) For all $i, j \in \{0, \dots, e\}$ and $x, y \in DVARS^{\varphi}$, $\mathsf{X}^i x \approx \mathsf{X}^j y \in fr[1]$ iff $\mathsf{X}^j y \approx \mathsf{X}^i x \in fr[1]$.

(F3) For all $i, j, j' \in \{0, \dots, e\}$ and $x, y, z \in DVARS^{\varphi}$, if $\{\mathsf{X}^i x \approx \mathsf{X}^j y, \mathsf{X}^j y \approx \mathsf{X}^{j'} z\} \subseteq fr[1]$, then $\mathsf{X}^i x \approx \mathsf{X}^{j'} z \in fr[1]$.

(F4) For all $i, j \in \{0, \dots, e\}$ and $x, y \in DVARS^{\varphi}$ such that $\mathsf{X}^i x \approx \mathsf{X}^j y \in fr[1]$:

- if $i = j$, then for every $z \in DVARS^{\varphi}$ and every $\psi \in \hat{\Phi}$, we have $\mathsf{X}^i(x \approx \langle \psi ? \rangle^{-1} z) \in fr[1]$ iff $\mathsf{X}^j(y \approx \langle \psi ? \rangle^{-1} z) \in fr[1]$.
- if $i < j$, then $\mathsf{X}^j(y \approx \langle \psi ? \rangle^{-1} x) \in fr[1]$ $\forall \psi \in \mathcal{G}_i^{fr}$ and for every $z \in DVARS^{\varphi}$ and every $\hat{\psi} \in \hat{\Phi}$, $\mathsf{X}^j(y \approx \langle \hat{\psi} ? \rangle^{-1} z) \in fr[1]$ iff either $\mathsf{X}^i(x \approx \langle \hat{\psi} ? \rangle^{-1} z) \in fr[1]$ or there exists $i \le j' < j$ with $\mathsf{X}^j y \approx \mathsf{X}^{j'} z \in fr[1]$ and $\hat{\psi} \in \mathcal{G}_{j'}^{fr}$.

(F5) For all $i, j \in \{0, \ldots, e\}$ and $x, y \in DVARS^{\varphi}$ such that $\mathsf{X}^i x \approx \mathsf{X}^j y \in fr[1]$, let $I_x := fr[i+2](x)$ and $I_y := fr[j+2](y)$, then:

- if $i = j$, then $I_x = I_y$.
- if $i < j$, and there is no j' such that $i < j' < j$ satisfying $\mathsf{X}^i x \approx \mathsf{X}^{j'} z \in fr[1]$ for any z, then $I_y = I_x \cup \{([(x,i)]_{fr}, \{\psi \mid \mathsf{X}^i(\psi) \in fr[1] \text{ and } \psi \in \Phi_{\mathcal{AT}}\})\}$.

(F6) For all i, and for all $x \in DVARS^{\varphi}$, $\mathsf{PO}_{fr}(x,i)$ should be matched with $fr[i+2](x)$, where the notion of matching is defined below.

For example, suppose at some level i of a frame the set of past obligations, $\mathsf{PO}_{fr}(w,i) = \{(x,\varphi_1),(y,\varphi_2 \wedge \varphi_3),(z,\top)\}$ for some $\varphi_1, \varphi_2 \wedge \varphi_3, \top \in \hat{\Phi}$. Then this can be matched with a Repetition Pattern $I_x = \{(\{x\},\{\varphi_1\}),(\{y\},\{\varphi_2,\varphi_3\}),(\{z\},\{\top\})\}$ where $\varphi_1, \varphi_2, \varphi_3, \top \in \Phi_{\mathcal{AT}}$.

Similar to the previous chapter, we define one-step consistency between two frames. The conditions are almost the same with some modifications and some extra conditions. A pair of (l,φ)-frames (fr, fr') is said to be one-step consistent iff the following conditions are satisfied

(O1) For all $\mathsf{X}^i x \approx \mathsf{X}^j y \in \Omega_l^{\varphi}$ with $i, j > 0$, we have $\mathsf{X}^i x \approx \mathsf{X}^j y \in fr[1]$ iff $\mathsf{X}^{i-1} x \approx \mathsf{X}^{j-1} y \in fr'[1]$,

(O2) For all $\mathsf{X}^i(x \approx \langle\psi?\rangle^{-1} y) \in \Omega_l^{\varphi}$ with $i > 0$, we have $\mathsf{X}^i(x \approx \langle\psi?\rangle^{-1} y) \in fr[1]$ iff $\mathsf{X}^{i-1}(x \approx \langle\psi?\rangle^{-1} y) \in fr'[1]$,

(O3) For all $\mathsf{X}^i q \in \Omega_l^{\varphi}$ with $i > 0$, we have $\mathsf{X}^i q \in fr[1]$ iff $\mathsf{X}^{i-1} q \in fr'[1]$,

(O4) For all $i > 0$, $fr[i+2] = fr'[i+1]$.

Similarly, for $e \in \{0, \ldots, l-1\}$, an (e,φ) frame fr and an $(e+1,\varphi)$ frame fr', the pair (fr, fr') is said to be one step consistent iff $fr \subseteq fr'$ and for every constraint in fr' of the form $\mathsf{X}^i x \approx \mathsf{X}^j y$, $\mathsf{X}^i q$ or $\mathsf{X}^i(x \approx \langle\psi?\rangle^{-1} y)$ with $i, j \in \{0, \ldots, e\}$, the same constraint also belongs to fr.

The symbolic models and the symbolic satisfaction relation are defined in the same way as in the previous chapter except in this case the definition will only depend on the first components of the frames. An (infinite) (l,φ)-symbolic model ρ is an infinite sequence of (l,φ)-frames such that for all $i \in \mathbb{N} \setminus \{0\}$, the pair $(\rho(i), \rho(i+1))$ is one-step consistent. We say $\rho, i \models_{symb} \varphi'$ where φ' is a sub-formula of φ if $(\rho, i)[1] \models_{symb} \varphi'$ and the later is defined in the same way as in section A.1. We say that a concrete model σ realizes a symbolic model ρ if for every $i \in \mathbb{N} \setminus \{0\}$, $\rho(i)[1] = \{\varphi' \in \Omega_l^{\varphi} \mid \sigma, i \models \varphi'\}$.In this case also we can prove the same result as Lemma 11:

Lemma 4 (symbolic vs. concrete models). *Suppose φ is a* LRV$[\Psi, \approx, \leftarrow]$ *formula of* X*-length l, ρ is a (l, φ)-symbolic model and σ is a concrete model realizing ρ. Then ρ symbolically satisfies φ iff σ satisfies φ.*

We define forward reference and backward reference in the same way. For $e \in \{0, \ldots, l\}$, an (e, φ)-frame fr, $i \in \{0, \ldots, e\}$ and a variable x, we say that there is a forward reference (resp. backward reference) from (x, i) in fr if $\mathsf{X}^i x \approx \mathsf{X}^{i+j} y \in \mathit{fr}[1]$ (resp. $\mathsf{X}^i x \approx \mathsf{X}^{i-j} y \in \mathit{fr}[1]$) for some $j > 0$ and $y \in \mathit{DVARS}^{\varphi}$.

Now we define points of increment and decrement of $\mathsf{Repetition\ Pattern}$s as follows: for $e \in \{0, \ldots, l\}$, and a variable x,

- For an (l, φ)-frame fr, if there is no forward reference from $(x, 0)$, then $[(x, 0)]_{\mathit{fr}}$ is a *point of increment* for the $\mathsf{Repetition\ Pattern}$ $I_x \cup \{([(x, 0)]_{\mathit{fr}}, \mathit{fr}[2])\}$ and the $\mathsf{Repetition\ Pattern}$ is incremented once for each equivalence class $[(x, 0)]_{\mathit{fr}}$.
- For an (e, φ)-frame fr, if there is no backward reference from (x, e), then $[(x, e)]_{\mathit{fr}}$ is a *point of decrement* for the $\mathsf{Repetition\ Pattern}$ I_x and the $\mathsf{Repetition\ Pattern}$ is decremented once for each equivalence class $[(x, e)]_{\mathit{fr}}$.

We denote by $\mathit{inc}(\mathit{fr})$ the vector indexed by non-empty elements of RP, where each coordinate contains the number of points of increment in fr for the corresponding $\mathsf{Repetition\ Pattern}$. Similarly, we have the vector $\mathit{dec}(\mathit{fr})$ for points of decrement.

Given a LRV$[\Psi, \approx, \leftarrow]$ formula φ in which $\mathit{DVARS}^{\mathsf{e}} = \emptyset = \mathit{BVARS}^{\mathsf{s}}$, we construct a single-sided VASS game as follows. The construction is almost the same as in the construction in Section A.1.

Let the X-length of φ is l, FR be the set of all (e, φ)-frames for all $e \in \{0, \ldots, l\}$, A_{φ} be the deterministic parity automaton whose language is the set of all symbolic models which symbolically satisfies φ with initial state $q^{\varphi}_{\mathit{init}}$. $\mathsf{Environment}$ states: $\{-1, 0, \ldots, l\} \times Q^{\varphi} \times (\mathsf{FR} \cup \{\perp\})$ and system states: $\{-1, 0, \ldots, l\} \times Q^{\varphi} \times (\mathsf{FR} \cup \{\perp\}) \times \mathcal{P}(\mathit{BVARS}^{\varphi})$. Transitions of the VASS game are defined as follows:

- $(e, q, \mathit{fr}) \xrightarrow{\vec{0}} (e, q, \mathit{fr}, V)$ for every $e \in \{-1, 0, \ldots, l\}$, $q \in Q^{\varphi}$, $\mathit{fr} \in \mathsf{FR} \cup \{\perp\}$ and $V \subseteq \mathit{BVARS}^{\varphi}$.
- $(e, q^{\varphi}_{\mathit{init}}, \mathit{fr}, V) \xrightarrow{\mathit{inc}(\mathit{fr}) - \mathit{dec}(\mathit{fr}')} (e+1, q^{\varphi}_{\mathit{init}}, \mathit{fr}')$ for every $V \subseteq \mathit{BVARS}^{\varphi}$, $e \in \{-1, 0, \ldots, l-2\}$, (e, φ)-frame fr and $(e+1, \varphi)$-frame fr', where the pair $(\mathit{fr}, \mathit{fr}')$ is one-step consistent and $\{p \in \mathit{BVARS}^{\varphi} \mid \mathsf{X}^{e+1} p \in \mathit{fr}'[1]\} = V$.

- $(e, q, fr, V) \xrightarrow{inc(fr)-dec(fr')} (\lceil e+1 \rceil l, q', fr')$, for every $e \in \{l-1, l\}$, (e, φ)-frame fr, $V \subseteq BVARS^{\varphi}$, $q, q' \in Q^{\varphi}$ and $(\lceil e+1 \rceil l, \varphi)$-frame fr', where the pair (fr, fr') is one-step consistent, $\{p \in BVARS^{\varphi} \mid \mathsf{X}^{\lceil e+1 \rceil l} p \in fr'[1]\} = V$ and $q \xrightarrow{fr'} q'$ is a transition in A^{φ}.

We will have a counter for each non-empty Repetition Pattern. We can prove the equivalence of the winning strategies in the same manner.

Theorem 5 (repeating values to VASS). *Let φ be a* LRV[Ψ, $\approx$, $\leftarrow$] *formula with $DVARS^{e} = BVARS^{s} = \emptyset$. Then* system *has a winning strategy in the corresponding single-sided* LRV[Ψ, $\approx$, $\leftarrow$] *game iff she has a winning strategy in the single-sided VASS game constructed above.*

Proof. Result follows from the following two lemmas.

Lemma 6. *If* system *has a winning strategy in the single-sided* LRV[Ψ, $\approx$, $\leftarrow$] *then it has a winning strategy in the single-sided VASS game constructed above.*

Proof. We shall define a map $\mu : (\mathcal{P}(BVARS^{\varphi}))^{*} \to \mathsf{FR} \cup \{\bot\}$. For every sequence $\chi \in (\mathcal{P}(BVARS^{\varphi}))^{*}$, we will define $\mu(\chi)$ and a concrete model of length $|\chi|$, by induction on $|\chi|$.

When $|\chi| = 0$, the concrete model is the empty sequence and the frame is $\bot$.

Now suppose, $\chi = \chi' \cdot V$ and σ is the concrete model defined for χ'. Let $v^{e} : BVARS^{e} \to \{\top, \bot\}$ be such that $v^{e}(p) = \top$ iff $p \in V$. The system player's winning strategy ts in the single-sided LRV[Ψ, $\approx$, $\leftarrow$] game will give a valuation $ts(\sigma \cdot v^{e}) = v^{s} : DVARS^{s} \to \mathbb{D}$. Then the finite concrete model corresponding to χ to be $\sigma \cdot (v^{e} \oplus v^{s})$ and $\mu(\chi)$ to be the frame fr' such that $fr'[1] = \{\varphi' \in \Omega_{l}^{\varphi} \mid \sigma \cdot (v^{e} \oplus v^{s}), |\sigma| + 1 - \lceil |\sigma| \rceil l \models \varphi'\}$. Suppose fr' is an (e, φ) frame; then for each $i \in \{0, \ldots, e-1\}$ and $x \in DVARS^{\varphi}$, $fr'[i+2](x)$ can be achieved by the previous frame due to one-step consistency and $fr'[e+2](x)$ is defined to be a Repetition Pattern that contains all the tuples $(X, Y) \in \mathcal{P}(DVARS^{\varphi}) \times \mathcal{P}(\Phi_{\mathcal{AT}})$ satisfying the following conditions:

- There should be one such tuple corresponding to each position in the past that repeats the data value of x at the current position.
- For each such past position, the first component will contain the equivalence class of variables taking the data value. And the second component will contain all the formulas from $\Phi_{\mathcal{AT}}$ that are true at that position.

Note that, since the atomic constraints captured in the frame only depend on the past, hence the frame can easily be constructed just looking at the concrete model built so far. If we consider the LRV with any nested formula, at this step we could not have built the frame just looking at the past since the nested formula could also depend on the future positions.

Let us denote by ρ the symbolic model $fr_{l+1}fr_{l+2}\cdots$. Same way as in Lemma 13, we can conclude $\rho \models_{symb} \varphi$ and hence the system player's strategy in the VASS game satisfies the parity winning condition.

Now we will show that the counter (recall that counters are Repetition Patterns in this case) value remains non-negative always. Initially all counters are 0. In the first $l+1$ frames, any counter cannot decrement since it would have a backward reference. Now it is enough to show that we can assign a unique point of increment to each point of decrement for every counter in any frame fr_i for $i > l+1$.

Suppose, for some $x \in DVARS^{\varphi}$, $[(x,l)]_{fr_i}$ is a point of decrement for I_x. Choose the last position before i such that $\sigma(i')(x') = \sigma(i)(x)$ and associate with $[(x,l)]_{fr_i}$ the class $[(x',0)]_{fr_{i'+l}}$. Now have to show that this is unique. Suppose for contradiction, it is also associated with $[(y,l)]_{fr_j}$. Then $\sigma(j)(y) = \sigma(i)(x)$. If $i = j$, then $[(y,l)]_{fr_j} = [(x,l)]_{fr_i}$. Now suppose $j < i$. then $j \in \{1,\ldots,i-l-1\}$ (otherwise it would have a backward reference which contradicts the fact that I_x has a point of decrement); now if we compute j' for $[(y,l)]_{fr_j}$ like we computed i' for $[(x,l)]_{fr_i}$, then we get $j' < i'$ (since $j \leq i'$). Similarly if we assume $j > i$, then $i \in \{1,\ldots,i-j-1\}$, hence $i \leq j'$ implying $i' < j'$; hence $[(x',0)]_{fr_{j'+l}}$ we associate with $[(y,l)]_{fr_j}$ would be different from $[(x',0)]_{fr_{i'+l}}$. □

Lemma 7. *If system has a winning strategy in the single-sided VASS game constructed above, then it has a winning strategy in the single-sided* LRV$[\Psi$, $\approx$, $\leftarrow]$.

Proof. From system player's strategy ss in VASS game, we construct its winning strategy in the LRV game. For every $\sigma \in \Upsilon^*$ and every $\upsilon^e \in \Upsilon^e$, we will define $ts(\sigma \cdot \upsilon^e) : DVARS^{\varphi} \to \mathbb{D}$ and a sequence of configurations $\chi \cdot ((e,q,fr), \vec{n}_{inc} - \vec{n}_{dec})$ in $(Q \times \mathbb{N}^C)^* \cdot (Q^e \times \mathbb{N}^C)$ of length $2|\sigma|+3$ such that for every counter $I_x \in RP$, $\vec{n}_{inc}(I_x)$ (resp. $\vec{n}_{dec}(I_x)$) is the sum of the number of points of increment (resp., points of decrement) for I_x in all the frames occurring in χ. By *frames occurring in* χ, we refer to frames fr such that there are consecutive configurations $((e,q,fr),\vec{n})((e,q,fr,V),\vec{n})$ in χ. By $\Pi_{\mathsf{FR}}(\chi)(i)$, we refer to ith such occurrence of a frame in χ. We induct on $|\sigma|$. Let $\{d_0, d_1, \ldots\} \subseteq \mathbb{D}$ be a countably infinite set of data values.

When $|\sigma| = 0$, let V be defined s.t. $p \in V$ iff $\upsilon^e(p) = \top$. Let $(-1, q^{\varphi}_{init}, \bot, V) \xrightarrow{\vec{0}-dec(fr_1)} (0, q, fr_1)$ be the transition chosen by ss. Since it is winning, hence $dec(fr_1)$ is

0. We define some unused data value to each equivalence class in fr_1. And define the sequence of configurations to be $((-1, q^{\varphi}_{init}, \perp), \vec{0}) \cdot ((-1, q^{\varphi}_{init}, \perp, V), \vec{0}) \cdot ((0, q, fr_1), -dec(fr_1))$.

For the induction step, suppose $\sigma \cdot v^e = \sigma' \cdot (v^e_1 \oplus v^s_1) \cdot v^e$ and $\chi' \cdot ((e, q, fr), \vec{n})$ is the sequence of configurations given by the induction hypothesis for $\sigma' \cdot v^e_1$. We have $\{\varphi' \in \Omega^{\varphi}_l \mid \sigma' \cdot (v^e_1 \oplus v^s_1), |\sigma'| + 1 - e \models \varphi'\} = fr[1]$ and $fr[e+2](x)$ contains all the tuples $(X, Y) \in \mathcal{P}(DVARS^{\varphi}) \times \mathcal{P}(\Phi_{\mathcal{AT}})$ satisfying the condition that there exists $j \leq |\sigma|$, such that for all $y \in X$, $(\sigma \cdot (v^e \oplus v^s))(|\sigma| + 1)(x) = \sigma(j)(y)$ and for all $\psi \in Y$, $\sigma, j \models \psi$.(otherwise, it would mean that the system player has already deviated from the strategy defined so far; in that case we choose arbitrarily). Note that the other coordinates of fr are fixed due to one-step consistency of frames. Let V be defined s.t. $p \in V$ iff $v^e(p) = \top$. Let $(e, q, fr, V) \xrightarrow{inc(fr) - dec(fr')} (\lceil e + 1 \rceil l, q', fr')$ be the transition chosen by ss. We define the sequence of configurations as $\chi' \cdot ((e, q, fr), \vec{n}) \cdot ((e, q, fr, V)\vec{n}) \cdot ((\lceil e + 1 \rceil l, q', fr'), \vec{n} + inc(fr) - dec(fr'))$. Since ss is winning for system, hence $\vec{n} + inc(fr) - dec(fr') \geq 0$. To define $ts(\sigma \cdot v^e)$, to each equivalence class $[(x, \lceil e + 1 \rceil l)]_{fr'}$, assign the data value d' as defined below:

1. If x has a backward reference at level l in fr', then choose the data value to be the data value at that position.

2. If there is no backward reference and no past obligations also, then assign a new data value.

3. Otherwise, it is a point of decrement for some I_x. Match it with a point of increment in a frame that has not been paired off before (it is possible because $(\vec{n} + inc(fr))(I_x) \geq dec(fr')(I_x))$). Suppose we pair off I_x at level $(e + 1)$ in fr' with a point of increment $[(x, 0)]$ in the frame $fr_i = \Pi_{\mathsf{FR}}(\chi' \cdot ((e, q, fr), \vec{n}) \cdot ((e, q, fr, V), \vec{n}))(i)$. Then we define d' to be $\sigma' \cdot (v^e_1 \oplus v^s_1)(i)(x)$.

Suppose ss results in the model $\sigma = (v^e_1 \oplus v^s_1) \cdot (v^e_2 \oplus v^s_2) \cdots$. It is clear from the construction that there is a sequence of configurations

$$\begin{gathered} ((-1, q^{\varphi}_{init}, \perp), \vec{0})((-1, q^{\varphi}_{init}, \perp, V_1), \vec{0}) \\ ((0, q^{\varphi}_{init}, fr_1), \vec{n}_1)((0, q^{\varphi}_{init}, fr_1, V_2), \vec{n}_1) \\ ((1, q^{\varphi}_{init}, fr_2), \vec{n}_2) \cdots ((l, q, fr_{l+1}), \vec{n}_{l+1}) \\ ((l, q, fr_{l+1}, V_{l+2}), \vec{n}_{l+1})((l, q', fr_{l+2}), \vec{n}_{l+2}) \cdots \end{gathered}$$

in the single-sided VASS game such that the concrete model σ realizes the symbolic model $fr_{l+1} fr_{l+2} \cdots$. Since ss is winning for the system player, the sequence of configurations above satisfy the parity condition of the single-sided VASS game, so $fr_{l+1} fr_{l+2} \cdots$ symbolically satisfies φ. From Lemma 4, we conclude that σ satisfies φ. $\square$

This concludes the proof of Theorem 5. □

3.2 Undecidability of single-sided LRV$[\approx, \leftarrow]$ games

In the previous section, we have shown that if the nested formulas are of some restricted kind, particularly generated by the syntax defined in the beginning of this chapter, then the existence of winning strategy for single-sided LRV games with past and equality constraints is reducible to single-sided VASS game. But in general, if we allow any form of nested formula in the LRV game with past, then even the single-sided games become undecidable.

Theorem 8. *The existence of winning strategy for single-sided* LRV$[\approx, \leftarrow]$ *games is undecidable, even when* environment *has 1 Boolean variable and* system *has 5 data variables (and some boolean variables).*

Proof. We prove this result by showing a reduction from the undecidability problem for a lossy counter machine defined in Theorem 2. System makes use of *labels* to encode the sequence of transitions of a witnessing run of the LCM. The undecidability result is true even when system has 5 data variables $x_1, \ldots, x_5$ (in addition to a number of Boolean variables which encode the labels); and environment has just one Boolean variable b.

Let's name the counters of the LCM $c_1, \ldots, c_5$. The variables $x_1, \ldots, x_5$ are used to encode the counters $c_1, \ldots, c_5$ respectively, and the variable b is used to ensure that there are no 'illegal' transitions — namely, no decrements of a zero-valued counter.

Given a LCM M with 5 counters, we first include 5 extra states to the model. Suppose Q was the set of states with initial state q_{init} and set of transitions δ. Then construct a new LCM M' with set of states $Q' = Q \uplus \{p_i \mid i = 1, \ldots, 5\}$ (without loss of generality assume that none of the p_i's appears in Q). M' will have initial state p_1 and the transition relation δ' will include transitions from δ, in addition we will have the following transitions:

- $(p_i, c_i++, p_i) \in \delta'$ for $i \in \{1, \ldots, 5\}$
- $(p_i, \epsilon, p_{i+1}) \in \delta'$ for $i \in \{1, \ldots, 4\}$
- $(p_5, \epsilon, q) \in \delta'$ for all $q \in Q$

These additional transitions will help the system player to choose an appropriate initial configuration of the LCM. We will discuss this in more details later when we will prove the equivalence between the existence of winning strategyof the system player in the LRV game and the existence of an infinite run in M'.

Now we define a LRV[$\approx$, $\leftarrow$] formula to force environment and system to simulate runs of the LCM M' as follows. Suppose σ is the concrete model built during a game. The value of counter c_j (for any j) before the i^{th} transition (encoded in the $(i+1)^{\text{st}}$ position) is the number of positions $k < i+1$ satisfying the following three conditions: i) the position k should have the label of a c_j++ transition, ii) $\sigma(k)(x) \notin \{\sigma(k')(x) \mid k+1 < k' < i+1\}$ and iii) there should not be any zero testing transition for counter c_j between positions k and $i+1$. Intuitively, if $(i+1)$ is the current position, the value of c_j is the number of positions from the last zero testing transition for c_j that have the label of a c_j++ transition and whose data value is not yet matched by a position with the label of a c_j-- transition.

In this reduction we assume that system plays first and environment plays next at each round, since it is easier to understand (the reduction also holds for the game where turns are inverted by shifting environment behavior by one position). At each round, system chooses the transition of the LCM M' and sets the value of its variables to complete the simulation. To this end, system is bound by the following set of rules:

1. The transition (p_5, ϵ, q) should have been taken at some point of time for some $q \in Q$. Let $p^j_{t_5}$ be a boolean variable belonging to the system player corresponding to such a transition for j^{th} state in Q. Then the following formula captures this situation.

$$\varphi_1 \equiv \bigvee_{j:q_j\in Q} \mathsf{F}(p^j_{t_5})$$

2. The transitions taken by system should be compatible. Let for any transition t, p_t be the corresponding boolean variable belonging to the system player.

$$\varphi_2 \equiv \mathsf{G}(\bigwedge_{t \text{ any transition}} (p_t \Rightarrow (\bigvee_{t':(t,t') \text{ compatible}} \mathsf{X}p_{t'})))$$

3. At a $c_i{+}{+}$ transition, the variable x_i should take a new data value from the domain which has not been used so far.

$$\varphi^i_3 \equiv \mathsf{G}((\bigvee_{t \text{ increments } c_i} p_t) \Rightarrow \neg(x_i \approx \Diamond^{-1} x_i))$$

$$\text{And } \varphi_3 = \bigwedge_{i=1}^{5} \varphi^i_3$$

4. At a zero testing transition $c_i = 0?$, the variable x_i should take a new data value (without loss of generality).

$$\varphi_4^i \equiv \mathsf{G}((\bigvee_{t \text{ is } c_i=0?} p_t) \Rightarrow \neg(x_i \approx \Diamond^{-1} x_i))$$

$$\text{And } \varphi_4 = \bigwedge_{i=1}^{5} \varphi_4^i$$

5. At a decrementing transition $c_i - -$, the data value at the variable x_i should repeat in the past at a $c_i + +$ transition.

$$\varphi_5^i \equiv \mathsf{G}((\bigvee_{t \text{ decrements } c_i} p_t) \Rightarrow (x_i \approx \langle \tau_{c_i++}? \rangle^{-1} x_i))$$

$$\text{And } \varphi_5 = \bigwedge_{i=1}^{5} \varphi_5^i$$

where τ_{c_i++} checks that the current position is an incrementing transition for counter c_i.

6. Any data value can repeat only atmost twice. This is equivalent to saying that it is not true that any data value occurs three times.

$$\varphi_6^i \equiv \mathsf{G}(\neg(x_i \approx \langle (x_i \approx \Diamond^{-1} x_i)? \rangle^{-1} x_i))$$

$$\text{And } \varphi_6 = \bigwedge_{i=1}^{5} \varphi_6^i$$

From these rules, it follows that every $c_i + +$ can be matched to at most one future $c_i - -$. Now note that, in incrementing transitions, system cannot cheat. Also, there is no possibility of cheating in the zero testing transitions by the system player since we are considering the lossiness relation to be reset lossiness; hence whenever there is a zero test for a counter, that counter can become zero immediately. But this coding can fail in the following way: there could be some $c_i - -$ transition and its matching $c_i + +$ transition such that there exists a zero test $c_i = 0?$ in-between. The variable b helps capturing the occurrence of this.

In all the rounds, environment always plays $\top$, except if it detects that this above situation has arisen, in which case it plays $\bot$. In that case system has to satisfy a formula that will ensure that if there was indeed the situation, system will lose, or if environment was just 'bluffing', then system will win.

Avoiding illegal decrements: Suppose, at some point of the simulation of the LCM M', system decides to play a $c_i - -$ transition such that in-between this transition and its matching incrementing transition $c_i + +$, there is a zero testing transition $c_i = 0?$. This is a form of cheating by system and environment should respond accordingly. Also assume that this is the first cheating that has occurred so far. Environment, who so far has been playing only $\top$, decides then to mark this position with $\bot$; and for the remaining of the play environment plays only $\top$ (even if more illegal transitions are performed in the sequel). So, for this situation environment's best strategy has a value sequence from $\top^*\bot\top^\omega$, and the following property characterizes environment's denouncement of an illegal decrement.

Property 1: b becomes $\bot$ at a $c_k - -$ position and stops being $\bot$ immediately after.

The following formula π_1 can test Property 1.

$$\pi_1 = b\mathsf{U}(\neg b \wedge \tau_{c_i--} \wedge \mathsf{X}(\mathsf{G}b))$$

Now if environment declares a cheating, then system has to justify it. In other words, system has to satisfy the follwoing formula along with the ones listed above:

(7) If at some position b turns to $\bot$ (let the corresponding transition be $c_i - -$), then the corresponding counter value at that position should repeat in past at some incrementing position, but there should not be any zero test in-between.

$$\varphi_7^i \equiv \pi_1 \Rightarrow \mathsf{G}(\neg b \Rightarrow x_i \approx \langle \tau_{c_i++} \wedge (\neg \tau_{c_i=0?}\mathsf{U}\neg b)?\rangle^{-1} x_i)$$

$$\text{And } \varphi_7 = \bigwedge_{i=1}^{5} \varphi_7^i$$

Note that, each of the formulas $\varphi_3, \ldots, \varphi_7$ are conjuncts of five disjoint formulas, for five different counters.

Now we define the winning condition for system player in the LRV game to be

$$\varphi = \bigwedge_{i=1}^{7} \varphi_i$$

It follows that system has a winning strategy for the game with input φ if and only if there is a positive answer to the existential infinite run problem for the lossy counter machine.

Correctness. First suppose that the LCM M has an infinite run starting from some initial configuration $(q, n_1, \ldots, n_5)$. In that case, the system player makes the counter values as required using the self loops of the states p_i, after some point it reaches q and from q, system's strategy would be to play according to the infinite run of M. This way φ_1 holds. φ_2 also becomes true since system actually follows the transitions of M. With respect to the data values on $c_1, \ldots, c_5$, system will respect the rues from (3)-(6). That will satisfy $\varphi_3, \ldots, \varphi_6$.

If the environment declares a cheating (plays a $\bot$), the transitions of M makes sure that from the last zero test for that counter, the increments can be matched with decrements which makes the formula φ_7 true.

Now suppose there is no infinite run of the LCM starting from any initial configuration, then each play of system satisfying $\varphi_1, \ldots, \varphi_6$ must have an illegal transition of the kind stated above (at a decrementing transition, the value of the corresponding variable should have repeated in past but in between a zero test occurs). At such first illegal transition, environment plays a $\bot$ and plays $\top$ in rest of the play. In this case, the antecedent of φ_7 will be true but the consequent will be false; thus φ_7 becomes false making system uncapable of finding a winning strategy.

This concludes the proof of Theorem 8. □

The following result follows immediately.

Corollary 9. *The existence of winning strategy for single-sided* LRV[$\leftarrow$] *games is undecidable, even when environment has 1 Boolean variable and system has 5 data variables (and some boolean variables).*

Chapter 4

Conclusion

In this thesis, we have extended the work done in section 5 of [7] and have shown that the decidability result does not hold for the realizability problem even for single-sided games if we allow nested formulas in past obligations.

As mentioned in the conclusion of [7], one possible future direction for future work would be to look for restriction of LRV games other than single-sidedness to get decidability for the realizability problem in section 3.1 (and also in section A.1).

For the decidable cases, we do not know how the structures of winning strategies will be. That is a possible direction we can look into. For example whether the strategy needs any memory or not; if yes then is it finitely representable etc.

The result we prove in section 3.1 is for a restriction of the nested formulas where we show that the realizability problem is decidable whereas in A.2 we have shown that with unrestricted nested formulas, it is undecidable. We do not know whether this boundary is tight *i.e.* the problem for nested formulas of special form which is a bit more extension of what we assume in section 3.1.

The undecidability result shown in sectionA.2 is based on a reduction from existential infinite run problem of a lossy counter machine. In this reduction we show that given a LCM, there exists an initial configuration from which there exists an infinite run iff system has a winning strategy in the game. Therefore, the game we consider here are infinite duration games. It would be interesting to investigate similar result for finite duration games also. It would be nice if we find a different undecidability proof where this assumption of infiniteness is not necessary or it might be decidable for finite games.

Bibliography

[1] P. A. Abdulla, R. Mayr, A. Sangnier, and J. Sproston. Solving parity games on integer vectors. In *CONCUR 2013*, volume 8052 of *LNCS*, pages 106–120. Springer Berlin Heidelberg, 2013.

[2] M. Bojańczyk, C. David, A. Muscholl, Th. Schwentick, and L. Segoufin. Two-variable logic on data words. *ACM Transactions on Computational Logic*, 12(4):27, 2011.

[3] S. Demri, D. D'Souza, and R. Gascon. Temporal logics of repeating values. *Journal of Logic and Computation*, 22(5):1059–1096, 2012.

[4] S. Demri, D. Figueira, and M. Praveen. Reasoning about Data Repetitions with Counter Systems. *Logical Methods in Computer Science*, Volume 12, Issue 3:1–55, Aug 2016.

[5] S. Demri and R. Lazić. LTL with the freeze quantifier and register automata. *ACM Transactions on Computational Logic*, 10(3), 2009.

[6] D. Figueira. A decidable two-way logic on data words. In *LICS'11*, pages 365–374. IEEE, 2011.

[7] Diego Figueira and M. Praveen. Playing with repetitions in data words using energy games. In *LICS*, 2018.

[8] A. Kara, Th. Schwentick, and Th. Zeume. Temporal logics on words with multiple data values. In *FST&TCS'10*, pages 481–492. LZI, 2010.

[9] Richard Mayr. Lossy counter machines. 1998.

[10] M. L. Minsky. Recursive unsolvability of post's problem of 'tag' and other topics in the theory of turing machines. *Annals of Mathematics*, 74:437–455, 1961.

[11] F. Neven, T. Schwentick, and V. Vianu. Finite state machines for strings over infinite alphabets. volume 5, pages 403–435, 2004.

[12] A. Pnueli and R. Rosner. On the synthesis of an asynchronous reactive module. *Automata, Languages and Programming*, pages 652–671, 1989.

[13] L. Segoufin. Automata and logics for words and trees over an infinite alphabet. In *CSL'06*, volume 4207 of *Lecture Notes in Computer Science*, pages 41–57. Springer, 2006.

Appendix A

A.1 Decidability of single-sided LRV$[\top, \leftarrow]$ games

In this section, we shall briefly present the proof from Section 5 of [7]. More precisely, in this section, we shall show that the single-sided LRV$[\top, \leftarrow]$ game is decidable. To prove this, we shall first show that given a formula φ in LRV$[\top, \leftarrow]$, we can transform it to a formula φ' in LRV$[\top, \approx, \leftarrow]$ such that the system player has a winning strategy with winning condition φ iff it has a winning strategy with winning condition φ' and then prove the decidability for LRV$[\top, \approx, \leftarrow]$ games.

Proposition 10. *The winning strategy existence problem for* LRV$[\top, \leftarrow]$ *is polynomial-time reducible into the problem on* LRV$[\top, \approx, \leftarrow]$.

Proof. It is proved in the same way as for the satisfiability problem in [4, Proposition 4]. Note that

- $\neg(x \not\approx \Diamond^{-1} y)$ is equivalent to $\neg\mathsf{X}^{-1}\top \vee (x \approx \mathsf{X}^{-1} y \wedge \mathsf{G}^{-1}(\neg\mathsf{X}^{-1}\top \vee y \approx \mathsf{X}^{-1} y))$ (note that the later is in LRV$[\top, \approx, \leftarrow]$);
- $x \not\approx \Diamond^{-1} y$ can be translated into $\neg(x \approx x_{\approx\Diamond^{-1}y}) \wedge x_{\approx\Diamond^{-1}y} \approx \Diamond^{-1} y$ where $x_{\approx\Diamond^{-1}y}$ is a new variable belonging to the same player as of x (note that the later is in LRV$[\top, \approx, \leftarrow]$).

Given any formula φ in negation normal form (*i.e.* , negation is only applied to boolean variables and data tests) in LRV$[\top, \leftarrow]$, construct the formula φ' using the replacements listed above. Note that φ' is in LRV$[\top, \approx , \leftarrow]$. It is straightforward to show that there is a winning strategy for the system player in the game with winning condition φ if and only if it has a winning strategy for the game with winning condition φ'. $\square$

Now we move towards proving the fact that single-sided LRV$[\top, \approx, \leftarrow]$ game is decidable and this will imply that single-sided LRV$[\top, \leftarrow]$ game is decidable because of Proposition 10. The main concept we use for proving decidability is a symbolic representation of models, introduced in [3]. The building blocks of the symbolic representation are *frames*. We shall show that a single-sided LRV[$\top, \approx, \leftarrow$] games can be efficiently reduced to a single-sided VASS game such that the system player has a winning strategy in one iff it has a winning strategy in the other. And because of Fact 3, the later is decidable; and hence also the former.

Given a formula in LRV[$\top$, $\approx$, $\leftarrow$], we first replace all the sub-formulas of the form $x \approx \mathsf{X}^{-j}y$ with $\mathsf{X}^{-j}(y \approx \mathsf{X}^{j}x)$ if $j > 0$ for simplicity. For a formula φ obtained after such replacements, let l be the maximum i such that a term of the form $\mathsf{X}^{i}x$ appears in φ. We call l the 'X-length' of φ. Let $BVARS^{\varphi}$ and $DVARS^{\varphi}$ denote the set of boolean and data variables respectively used in φ. Let Ω_l^{φ} be the set of constraints of the form $\mathsf{X}^{i}q$, $\mathsf{X}^{i}x \approx \mathsf{X}^{j}y$ or $\mathsf{X}^{i}(x \approx \Diamond^{-1}y)$, where $q \in BVARS^{\varphi}$, $x, y \in DVARS^{\varphi}$ and $i, j \in \{0, \dots, l\}$. These are the atomic constraints. Now we define frames. Intuitively, for any $e \in \{0, \dots, l\}$, an (e, φ)-frame consists of all possible formulas from Ω_l^{φ} which are true in the next e-many positions.

Formally, for $e \in \{0, \dots, l\}$, an (e, φ)-frame is a set of constraints $fr \subseteq \Omega_l^{\varphi}$ that satisfies the following conditions:

(F0) For all constraints $\mathsf{X}^{i}q, \mathsf{X}^{i}x \approx \mathsf{X}^{j}y, \mathsf{X}^{i}(x \approx \Diamond^{-1}y) \in fr$, $i, j \in \{0, \dots, e\}$.

(F1) For all $i \in \{0, \dots, e\}$ and $x \in DVARS^{\varphi}$, $\mathsf{X}^{i}x \approx \mathsf{X}^{i}x \in fr$.

(F2) For all $i, j \in \{0, \dots, e\}$ and $x, y \in DVARS^{\varphi}$, $\mathsf{X}^{i}x \approx \mathsf{X}^{j}y \in fr$ iff $\mathsf{X}^{j}y \approx \mathsf{X}^{i}x \in fr$.

(F3) For all $i, j, j' \in \{0, \dots, e\}$ and $x, y, z \in DVARS^{\varphi}$, if $\{\mathsf{X}^{i}x \approx \mathsf{X}^{j}y, \mathsf{X}^{j}y \approx \mathsf{X}^{j'}z\} \subseteq fr$, then $\mathsf{X}^{i}x \approx \mathsf{X}^{j'}z \in fr$.

(F4) For all $i, j \in \{0, \dots, e\}$ and $x, y \in DVARS^{\varphi}$ such that $\mathsf{X}^{i}x \approx \mathsf{X}^{j}y \in fr$:

- if $i = j$, then for every $z \in DVARS^{\varphi}$ we have $\mathsf{X}^{i}(x \approx \Diamond^{-1}z) \in fr$ iff $\mathsf{X}^{j}(y \approx \Diamond^{-1}z) \in fr$.
- if $i < j$, then $\mathsf{X}^{j}(y \approx \Diamond^{-1}x) \in fr$ and for any $z \in DVARS^{\varphi}$, $\mathsf{X}^{j}(y \approx \Diamond^{-1}z) \in fr$ iff either $\mathsf{X}^{i}(x \approx \Diamond^{-1}z) \in fr$ or there exists $i \le j' < j$ with $\mathsf{X}^{j}y \approx \mathsf{X}^{j'}z \in fr$.

The condiditon (F0) says that an (e, φ)-frame constrain at most $(e + 1)$ contiguous valuations. (F1)-(F3) says that the equality constraints in a frame form an equivalence relation. The condition (F4) ensures that the past obligations are consistent among the variables: if x-value repeats in past in variable z and also if value of x is same as the data value of some y at the current position, then we also add the formula that y-value repeats in past in z.

Now we define one-step consistency between two frames. A pair of (l, φ)-frames $(\mathit{fr}, \mathit{fr}')$ is said to be one-step consistent iff the following conditions are satisfied:

(O1) For all $\mathsf{X}^i x \approx \mathsf{X}^j y \in \Omega_l^\varphi$ with $i, j > 0$, we have $\mathsf{X}^i x \approx \mathsf{X}^j y \in \mathit{fr}$ iff $\mathsf{X}^{i-1} x \approx \mathsf{X}^{j-1} y \in \mathit{fr}'$,

(O2) For all $\mathsf{X}^i(x \approx \Diamond^{-1} y) \in \Omega_l^\varphi$ with $i > 0$, we have $\mathsf{X}^i(x \approx \Diamond^{-1} y) \in \mathit{fr}$ iff $\mathsf{X}^{i-1}(x \approx \Diamond^{-1} y) \in \mathit{fr}'$ and

(O3) For all $\mathsf{X}^i q \in \Omega_l^\varphi$ with $i > 0$, we have $\mathsf{X}^i q \in \mathit{fr}$ iff $\mathsf{X}^{i-1} q \in \mathit{fr}'$.

For $e \in \{0, \ldots, l-1\}$, an (e, φ) frame fr and an $(e+1, \varphi)$ frame fr', the pair $(\mathit{fr}, \mathit{fr}')$ is said to be one step consistent iff $\mathit{fr} \subseteq \mathit{fr}'$ and for every constraint in fr' of the form $\mathsf{X}^i x \approx \mathsf{X}^j y$, $\mathsf{X}^i q$ or $\mathsf{X}^i(x \approx \Diamond^{-1} y)$ with $i, j \in \{0, \ldots, e\}$, the same constraint also belongs to fr.

Now we define symbolic models. An (infinite) (l, φ)-symbolic model ρ is an infinite sequence of (l, φ)-frames such that for all $i \in \mathbb{N} \setminus \{0\}$, the pair $(\rho(i), \rho(i+1))$ is one-step consistent according to the above definition. Let us define the symbolic satisfaction relation $\rho, i \models_{\mathit{symb}} \varphi'$ where φ' is a sub-formula of φ. The relation $\models_{\mathit{symb}}$ is defined in the same way as $\models$ for LRV, except that for every element φ' of Ω_l^φ, we have $\rho, i \models_{\mathit{symb}} \varphi'$ whenever $\varphi' \in \rho(i)$. We say that a concrete model σ realizes a symbolic model ρ if for every $i \in \mathbb{N} \setminus \{0\}$, $\rho(i) = \{\varphi' \in \Omega_l^\varphi \mid \sigma, i \models \varphi'\}$. The next result follows easily from definitions.

Lemma 11 (symbolic vs. concrete models)**.** *Suppose φ is a* LRV$[\top, \approx, \leftarrow]$ *formula of* X*-length l, ρ is a (l, φ)-symbolic model and σ is a concrete model realizing ρ. Then ρ symbolically satisfies φ iff σ satisfies φ.*

Now we define some useful notations we shall use throughout this chapter. Given a formula φ in LRV$[\top, \approx, \leftarrow]$ of X-length l, for $e \in \{0, \ldots, l\}$, an (e, φ)-frame fr, $i \in \{0, \ldots, e\}$ and a variable x, the set of past obligations of the variable x at level i in fr (denoted by $\mathsf{PO}_{\mathit{fr}}(x, i)$) is the set of past obligations from x at level i those are present in fr. The equivalence class of x at level i in fr (denoted by $[(x, i)]_{\mathit{fr}}$) is defined to be the set of all the variables which have the data value as of x at level i. In notation,

- $\mathsf{PO}_{\mathit{fr}}(x, i) = \{y \in \mathit{DVARS}^\varphi \mid \mathsf{X}^i(x \approx \Diamond^{-1} y) \in \mathit{fr}\}$.
- $[(x, i)]_{\mathit{fr}} = \{y \in \mathit{DVARS}^\varphi \mid \mathsf{X}^i x \approx \mathsf{X}^i y \in \mathit{fr}\}$.

For $e \in \{0, \ldots, l\}$, an (e, φ)-frame fr, $i \in \{0, \ldots, e\}$ and a variable x,

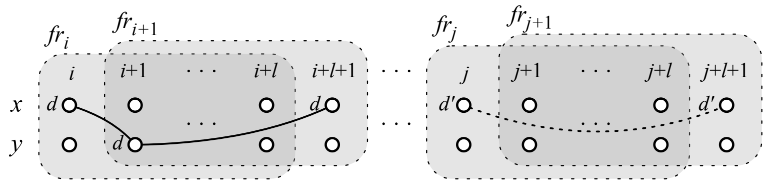
- We say there is a forward reference from (x, i) in fr if $\mathsf{X}^i x \approx \mathsf{X}^{i+j} y \in fr$.
- We say there is a backward reference from (x, i) in fr if $\mathsf{X}^i x \approx \mathsf{X}^{i-j} y \in fr$.

In some cases, there is no way to capture the data repetitions because in a frame we only keep track of at most l contiguous positions. For that, we maintain a counter corresponding to each subset of data variables to keep track of the number of such remote data repetitions. Let $X \subseteq DVARS^{\varphi}$ be a subset of data variables. Then

- A *point of decrement* for counter X in an (e, φ)-frame fr is an equivalence class of the form $[(x, e)]_{fr}$ such that there is no backward reference from (x, e) in fr and $\mathsf{PO}_{fr}(x, e) = X$.
- A *point of increment* for X in an (l, φ)-frame fr is an equivalence class of the form $[(x, 0)]_{fr}$ such that there is no forward reference from $(x, 0)$ in fr and $[(x, 0)]_{fr} \cup \mathsf{PO}_{fr}(x, 0) = X$.

We define points of increment only for (l, φ)-frames because (e, φ)-frames for $e < l$ do not contain complete information about constraints in the next l positions. We denote by $inc(fr)$ the vector indexed by non-empty subsets of $DVARS^{\varphi}$, where each coordinate contains the number of points of increment in fr for the corresponding subset of variables. Similarly, we have the vector $dec(fr)$ for points of decrement.

Here we give two examples of two different possible situations.



In the left column here, $\sigma(i)(x) = \sigma(i+1)(y) = \sigma(i+l+1)(x) = d$, where l is the X-length. The constraint $x \approx \mathsf{X}y$ in fr_i above is a forward reference from $(x, 0)$ in fr_i, while the constraint $\mathsf{X}^l x \approx y$ is a backward reference from (x, l) in fr_{i+1}. This way we capture that data value of x at positions i and $i + l + 1$ are equal in this case.

But if we consider the right column, there we have $\sigma(j)(x) = \sigma(j+l+1)(y)$. We cannot capture this kind of situations (the two positions are too far apart, neither are there any intermediate positions with the same data value) like in the left column. In this, we find the use of introducing counters. Here, the equivalence class $[(x, l)]_{fr_{j+1}}$ in the frame fr_{j+1} is a point of decrement for $\{x\}$.

And the equivalence class $[(x,0)]_{fr_j}$ in the frame fr_j is a point of increment for $\{x\}$.

Given a LRV[$\top$, $\approx$, $\leftarrow$] formula φ in which $DVARS^{e} = \emptyset = BVARS^{s}$, we construct a single-sided VASS game as follows. Fix the X-length of φ, let it be l. Let FR be the set of all (e,φ)-frames for all $e \in \{0,\ldots,l\}$. Note that the symbolic satisfiability defined above is just like LTL satisfiability. Hence we can construct an automaton accepting the symbolic models which symbolically satisfies φ. Let A^{φ} be a deterministic parity automaton that accepts a symbolic model iff it symbolically satisfies φ. Let Q^{φ} be the set of states of A^{φ} and q^{φ}_{init} be the initial state .

The single-sided VASS game will have set of counters $\mathcal{P}^{+}(DVARS^{\varphi})$, set of environment states $\{-1,0,\ldots,l\} \times Q^{\varphi} \times (\mathsf{FR} \cup \{\bot\})$ and set of system states $\{-1,0,\ldots,l\} \times Q^{\varphi} \times (\mathsf{FR} \cup \{\bot\}) \times \mathcal{P}(BVARS^{\varphi})$. The colour of a state will be the colour of its Q^{φ} component. We assume $\bot$ to be the only $(-1,\varphi)$-frame and $(\bot, fr')$ be one-step consistent for every 0-frame fr'. Initially all the counters have value 0. The initial state is $(-1, q^{\varphi}_{init}, \bot)$. Let $\lceil\cdot\rceil l$ denotes the mapping that is identity on $\{-1,0,\ldots,l-1\}$ and maps all others to l. The transitions of the VASS game are listed below:

- $(e,q,fr) \xrightarrow{\vec{0}} (e,q,fr,V)$ for every $e \in \{-1,0,\ldots,l\}$, $q \in Q^{\varphi}$, $fr \in \mathsf{FR} \cup \{\bot\}$ and $V \subseteq BVARS^{\varphi}$.
- $(e, q^{\varphi}_{init}, fr, V) \xrightarrow{inc(fr)-dec(fr')} (e+1, q^{\varphi}_{init}, fr')$ for every $V \subseteq BVARS^{\varphi}$, $e \in \{-1,0,\ldots,l-2\}$, (e,φ)-frame fr and $(e+1,\varphi)$-frame fr', where the pair (fr,fr') is one-step consistent and $\{p \in BVARS^{\varphi} \mid \mathsf{X}^{e+1}p \in fr'\} = V$.
- $(e,q,fr,V) \xrightarrow{inc(fr)-dec(fr')} (\lceil e+1\rceil l, q', fr')$ for every $e \in \{l-1, l\}$, (e,φ)-frame fr, $V \subseteq BVARS^{\varphi}$, $q,q' \in Q^{\varphi}$ and $(\lceil e+1\rceil l, \varphi)$-frame fr', where the pair (fr,fr') is one-step consistent, $\{p \in BVARS^{\varphi} \mid \mathsf{X}^{\lceil e+1\rceil l}p \in fr'\} = V$ and $q \xrightarrow{fr'} q'$ is a transition in A^{φ}.

The first kind of transitions are chosen by the environment player and note that the counter values do not change (ensures that game is single-sided). In this, the environment player chooses a set of variables to make them true in the next round. In the second and third kind of transitions, the condition $\{p \in BVARS^{\varphi} \mid \mathsf{X}^{\lceil e+1\rceil l}p \in fr'\} = V$ makes sure that the frame fr' chosen by system should be compatible with the subset V chosen by environment in the previous step. In the third kind of transition, the condition $q \xrightarrow{fr'} q'$ ensures that the symbolic model is accepted by A^{φ} and hence symbolically satisfies φ. The update vector $inc(fr) - dec(fr')$ ensures that symbolic models are realizable, which we shall show formally next.

Theorem 12 (repeating values to VASS). *Let φ be a* LRV[$\top$, $\approx$, $\leftarrow$] *formula with $DVARS^e = BVARS^s = \emptyset$. Then* system *has a winning strategy in the corresponding single-sided* LRV[$\top$, $\approx$, $\leftarrow$] *game iff it has a winning strategy in the single-sided VASS game constructed above.*

Proof. Result follows from the following two lemmas.

Lemma 13. *If* system *has a winning strategy in the single-sided* LRV[$\top$, $\approx$, $\leftarrow$] *then it has a winning strategy in the single-sided VASS game constructed above.*

Proof. We shall define a map $\mu : (\mathcal{P}(BVARS^\varphi))^* \to \mathsf{FR} \cup \{\bot\}$; it is then straight-forward to construct a strategy for system in the VASS game. For every sequence $\chi \in (\mathcal{P}(BVARS^\varphi))^*$, we will define $\mu(\chi)$ and a concrete model of length $|\chi|$, by induction on $|\chi|$. For the base case $|\chi| = 0$, the concrete model is the empty sequence and the frame is $\bot$.

For the induction step, suppose $\chi = \chi' \cdot V$ and σ is the concrete model defined for χ' by induction hypothesis. Let $v^e : BVARS^e \to \{\top, \bot\}$ be such that $v^e(p) = \top$ iff $p \in V$. The system player's winning strategy ts in the single-sided LRV[$\top$, $\approx$, $\leftarrow$] game will give a valuation $ts(\sigma \cdot v^e) = v^s : DVARS^s \to \mathbb{D}$. We define the finite concrete model corresponding to χ to be $\sigma \cdot (v^e \oplus v^s)$ and $\mu(\chi)$ to be the frame $fr' = \{\varphi' \in \Omega_l^\varphi \mid \sigma \cdot (v^e \oplus v^s), |\sigma| + 1 - \lceil |\sigma| \rceil l \models \varphi'\}$.

Let ss be the system player's strategy that we get from χ in the VASS game, resulting in the sequence of states

$$(-1, q^\varphi_{init}, \bot)(-1, q^\varphi_{init}, \bot, V_1)(0, q^\varphi_{init}, fr_1)(0, q^\varphi_{init}, fr_1, V_2)$$
$$(1, q^\varphi_{init}, fr_2) \cdots (l, q, fr_{l+1})(l, q, fr_{l+1}, V_{l+2})(l, q', fr_{l+2}) \cdots$$

The sequence $fr_{l+1} fr_{l+2} \cdots$ is an (l, φ)-symbolic model; call it ρ. From the construction, ρ is realized by a concrete model σ, which is the result of the system player's winning strategy ts. So $\sigma, 1 \models \varphi$ and by Lemma 11, $\rho \models_{symb} \varphi$. Also, since the strategy follows the transitions A_φ, it satisfies the parity condition. Now it remains to show that the counter value remains non-negative always.

Initially all counters are 0. In the first $l + 1$ frames, any counter cannot decrement since it would have a backward reference. Now it is enough to show that we can assign a unique point of increment to each point of decrement for every counter in any frame fr_i for $i > l + 1$. Suppose, for some $x \in DVARS^\varphi$, $[(x, l)]_{fr_i}$ is a point of decrement for X. Choose the last position before i such that $\sigma(i')(x') = \sigma(i)(x)$ and associate with $[(x, l)]_{fr_i}$ the class $[(x', 0)]_{fr_{i'+l}}$. Now have to show that this is unique. Suppose for contradiction, it is also associated with $[(y, l)]_{fr_j}$, which is a point of decrement for X in fr_j. Then $\sigma(j)(y) = \sigma(i)(x)$. If $i = j$, then $[(y, l)]_{fr_j} = [(x, l)]_{fr_i}$. Now suppose $j < i$.

then $j \in \{1, \ldots, i-l-1\}$ (otherwise it would have a backward reference which contradicts the fact that X has a point of decrement); now if we compute j' for $[(y,l)]_{fr_j}$ like we computed i' for $[(x,l)]_{fr_i}$, then we get $j' < i'$ (since $j \leq i'$). Similarly if we assume $j > i$, then $i \in \{1, \ldots, i-j-1\}$, hence $i \leq j'$ implying $i' < j'$; hence $[(y',0)]_{fr_{j'+l}}$ we associate with $[(y,l)]_{fr_j}$ would be different from $[(x',0)]_{fr_{i'+l}}$.

□

Lemma 14. *If* system *has a winning strategy in the single-sided VASS game constructed above, then it has a winning strategy in the single-sided* LRV$[\top, \approx, \leftarrow]$.

Proof. Let ss be the system's strategy in the VASS game. For every $\sigma \in \Upsilon^*$ and every $v^e \in \Upsilon^e$, we will define $ts(\sigma \cdot v^e) : DVARS^{\varphi} \to \mathbb{D}$ and a sequence of configurations $\chi \cdot ((e,q,fr), \vec{n}_{inc} - \vec{n}_{dec})$ in $(Q \times \mathbb{N}^C)^* \cdot (Q^e \times \mathbb{N}^C)$ of length $2|\sigma|+3$ such that for every counter $X \in \mathcal{P}^+(DVARS^{\varphi})$, $\vec{n}_{inc}(X)$ (resp. $\vec{n}_{dec}(X)$) is the sum of the number of points of increment (resp., points of decrement) for X in all the frames occurring in χ. By *frames occurring in* χ, we refer to frames fr such that there are consecutive configurations $((e,q,fr),\vec{n})((e,q,fr,V),\vec{n})$ in χ. By $\Pi_{\mathsf{FR}}(\chi)(i)$, we refer to i^{th} such occurrence of a frame in χ. We induct on $|\sigma|$. Let $\{d_0, d_1, \ldots\} \subseteq \mathbb{D}$ be a countably infinite set of data values.

When $|\sigma| = 0$, let V be defined s.t. $p \in V$ iff $v^e(p) = \top$. Let $(-1, q^{\varphi}_{init}, \perp, V) \xrightarrow{\vec{0} - dec(fr_1)} (0, q, fr_1)$ be the transition chosen by ss. Since it is winning, hence $dec(fr_1)$ is 0. We define some unused data value to each equivalence class in fr_1. And define the sequence of configurations to be $((-1, q^{\varphi}_{init}, \perp), \vec{0}) \cdot ((-1, q^{\varphi}_{init}, \perp, V), \vec{0}) \cdot ((0, q, fr_1), -dec(fr_1))$.

For the induction step, suppose $\sigma \cdot v^e = \sigma' \cdot (v^e_1 \oplus v^s_1) \cdot v^e$ and $\chi' \cdot ((e,q,fr),\vec{n})$ is the sequence of configurations given by the induction hypothesis for $\sigma' \cdot v^e_1$. We have $\{\varphi' \in \Omega^{\varphi}_l \mid \sigma' \cdot (v^e_1 \oplus v^s_1), |\sigma'| + 1 - e \models \varphi'\} = fr$ (otherwise, it would mean that the system player has already deviated from the strategy defined so far; in that case we choose arbitrarily). Let V be defined s.t. $p \in V$ iff $v^e(p) = \top$. Let $(e,q,fr,V) \xrightarrow{inc(fr) - dec(fr')} (\lceil e+1 \rceil l, q', fr')$ be the transition chosen by ss. We define the sequence of configurations as $\chi' \cdot ((e,q,fr),\vec{n}) \cdot ((e,q,fr,V)\vec{n}) \cdot ((\lceil e+1 \rceil l, q', fr'), \vec{n} + inc(fr) - dec(fr'))$. Since ss is winning for system, hence $\vec{n} + inc(fr) - dec(fr') \geq 0$. To define $ts(\sigma \cdot v^e)$, to each equivalence class $[(x, \lceil e+1 \rceil l)]_{fr'}$, assign the data value d' as defined below:

1. If x has a backward reference at level l in fr', then choose the data value to be the data value at that position.
2. If there is no backward reference and no past obligations also, then assign a new data value.

3. Otherwise, it is a point of decrement for some X. Match it with a point of increment in a frame that has not been paired off before (it is possible because $(\vec{n} + \mathit{inc}(\mathit{fr}))(X) \geq \mathit{dec}(\mathit{fr}')(X))$. Suppose we pair off $[(x, \lceil e+1 \rceil l)]_{\mathit{fr}'}$ at level $(e+1)$ in fr' with a point of increment $[(y, 0)]$ in the frame $\mathit{fr}_i = \Pi_{\mathsf{FR}}(\chi' \cdot ((e, q, \mathit{fr}), \vec{n}) \cdot ((e, q, \mathit{fr}, V), \vec{n}))(i)$. Then we define d' to be $\sigma' \cdot (\upsilon_1^e \oplus \upsilon_1^s)(i)(y)$.

Suppose ss results in the model $\sigma = (\upsilon_1^e \oplus \upsilon_1^s) \cdot (\upsilon_2^e \oplus \upsilon_2^s) \cdots$. It is clear from the construction that there is a sequence of configurations

$$\begin{gathered}((-1, q_{\mathit{init}}^{\varphi}, \perp), \vec{0})((-1, q_{\mathit{init}}^{\varphi}, \perp, V_1), \vec{0})\\ ((0, q_{\mathit{init}}^{\varphi}, \mathit{fr}_1), \vec{n}_1)((0, q_{\mathit{init}}^{\varphi}, \mathit{fr}_1, V_2), \vec{n}_1)\\ ((1, q_{\mathit{init}}^{\varphi}, \mathit{fr}_2), \vec{n}_2) \cdots ((l, q, \mathit{fr}_{l+1}), \vec{n}_{l+1})\\ ((l, q, \mathit{fr}_{l+1}, V_{l+2}), \vec{n}_{l+1})((l, q', \mathit{fr}_{l+2}), \vec{n}_{l+2}) \cdots\end{gathered}$$

in the single-sided VASS game such that the concrete model σ realizes the symbolic model $\mathit{fr}_{l+1}\mathit{fr}_{l+2}\cdots$. Since ss is winning for the system player, the sequence of configurations above satisfy the parity condition of the single-sided VASS game, so $\mathit{fr}_{l+1}\mathit{fr}_{l+2}\cdots$ symbolically satisfies φ. From Lemma 11, we conclude that σ satisfies φ. □

This concludes the proof of Theorem 12. □

A.2 Undecidability of single-sided LRV[$\top$, $\approx$, $\rightarrow$] games

In the previous section, we have shown that the existence of winning strategy for single-sided LRV games with past and equality constraints and with no nested formula is decidable. But in this secetion, we shall see that if instead of past, if we allow only future obligations then this problem becomes undecidable [7, Section 6].

Theorem 15. *The existence of winning strategy for single-sided* LRV[$\top, \approx, \rightarrow$] *games is undecidable, even when* environment *has 1 Boolean variable and* system *has 3 data variables (and some boolean variables).*

Proof. We prove this result by showing a reduction from the undecidability problem for a 2-counter machine M defined in Theorem 1. System makes use of *labels* to encode the sequence of transitions of a witnessing run of the CM. The undecidability result is true even when system has 3 data variables x, y, z (in addition to a number of Boolean variables which encode the labels); and

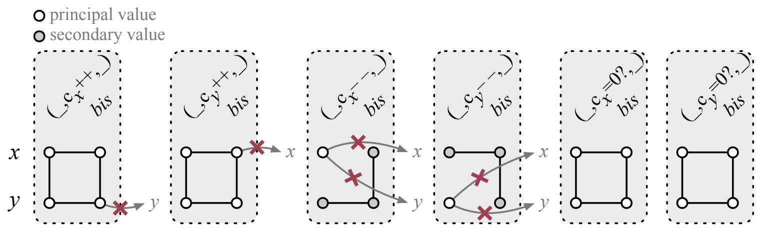


Figure A.1: Properties of data values of variables

environment has just one Boolean variable b. For convenience, we name the counters of the 2-counter machines c_x and c_y instead of c_1 and c_2. Variables x, y are used to encode the counters c_x and c_y as before, and variables z, b are used to ensure that there are no 'illegal' transitions — namely, no decrements of a zero-valued counter, and no tests for zero for a non-zero-valued counter.

Each transition in the run of the 2-counter machine will be encoded using *two* consecutive positions of the game. In this encoding, transitions are interspersed with a special *bis* label, and thus a witnessing reachability run $t_1 \, t_2 \cdots t_n \in \delta^*$ is encoded as $t_1 \mathit{bis} \, t_2 \, \mathit{bis} \cdots t_n \, \mathit{bis} \, (\hat{t} \, \mathit{bis})^\omega \in (\delta \cup \{\mathit{bis}\})^\omega$. We will see why we need two positions per transition.

Now we define a LRV$[\top, \approx, \rightarrow]$ formula to force environment and system to simulate runs of the 2-counter machine M as follows. Suppose σ is the concrete model built during a game. The value of counter c_x before the i^{th} transition (encoded in the $2i^{th}$ and $(2i+1)^{st}$ positions) is the number of positions $j < 2i$ satisfying the following two conditions: i) the position j should have the label of a c_x++ transition and ii) $\sigma(j)(x) \notin \{\sigma(j')(x) \mid j+1 < j' < 2i\}$. Same for counter c_y. Intuitively, at some position, the value of a counter is the number of incrementing positions of that counter before the current position whose data value is not yet matched with any position.

In this reduction we assume that system plays first and environment plays next at each round, since it is easier to understand (the reduction also holds for the game where turns are inverted by shifting environment behavior by one position). At each round, if the last label played was a transitionsystem will play a label *bis*. Otherwise, it will choose the next transition of the 2-counter machine to simulate. To follow the encoding above, system will follow the rules described pictorially in Figure A.1.

The properties of data values are clear from the figure itself. For example, the first (leftmost) rule in the figure says that whenever there is a c_x++ transition label, then all four values for x and y in both positions (*i.e.* , the transition position and the next *bis* position) must have the same data value d (which we call 'principal'), which does not occur in the future under variable y. The presence of two positions per transition ensures that (though we don't use past obligations,) if at a position has the label of a c_x++ transition and the variable x has the data value d and the data value d repeats in the future, it will be only once and at a position that has the label of a c_x-- transition.

The LRV formula for the rules depicted in the figure is the following:

$$\varphi_{x,y} = \mathsf{G}(\bigwedge_{a \in A} \tau_a \Rightarrow \zeta_a)$$

where $A = \{c_x++, c_x--, c_x = 0?, c_y++, c_y--, c_y = 0?\}$ and

$$\tau_a = \bigvee_{q,q' \in Q, (q,a,q') \in \delta} \lambda_{(q,a,q')},$$

where $\lambda_{(q,a,q')}$ tests that we are standing on a position labelled with transition (q, a, q') (in particular not a *bis* position). And ζ_a encodes the rules as already described. That is,

$$\begin{aligned}
\zeta_{c_x++} &= x \approx y \wedge x \approx \mathsf{X}x \wedge y \approx \mathsf{X}y \wedge \neg\mathsf{X}(x \approx \Diamond y),\\
\zeta_{c_x--} &= \neg x \approx y \wedge \neg x \approx \Diamond x \wedge \neg x \approx \Diamond y \wedge\\
&\quad\; y \approx \mathsf{X}y \wedge y \approx \mathsf{X}x,\\
\zeta_{c_x=0?} &= x \approx y \wedge x \approx \mathsf{X}x \wedge y \approx \mathsf{X}y,
\end{aligned}$$

and similarly for the rules on c_y.

But there can be two ways in which this coding can fail: a) there could be invalid tests $c_k = 0?$, and b) there could be some c_k-- with no previous matching c_k++ (for $k \in \{x, y\}$). Variables z and b play a crucial role in the game whenever any of these two cases occurs as explained next.

(a) Avoiding illegal tests for zero: Suppose there is some preceding c_k++ transition for which either: (a1) there is no matching c_k-- transition; this situation is avoided by the following formula: $\mu = \mathsf{G}(\tau_{(c_k++)} \wedge \mathsf{F}\tau_{(c_k=0?)} \Rightarrow k \approx \Diamond k)$ or (a2) there is a matching c_k-- transition but it occurs after the $c_k = 0?$ transition. For (a2), suppose this is the *first* illegal transition that has occurred so far. Then environment will choose b to be $\bot$ at this position and will continue playing $\bot$ till the next matching c_k-- transition (if it is never reached, it is situation a1); in rest of the places b value will be $\top$. So, for this situation environment's best strategy has a value sequence from $\top^*\bot^*\top^\omega$, and the following property characterizes environment's denouncement of an illegal zero test. In the following, we analyze the case assuming $k = x$; the final formula will be a conjunct of two such formulas each for $k = x$ and $k = y$.

- A formula π_1 can test the property: b becomes $\bot$ at a $c_x = 0?$ position and stops being $\bot$ at a c_x-- position thereafter.

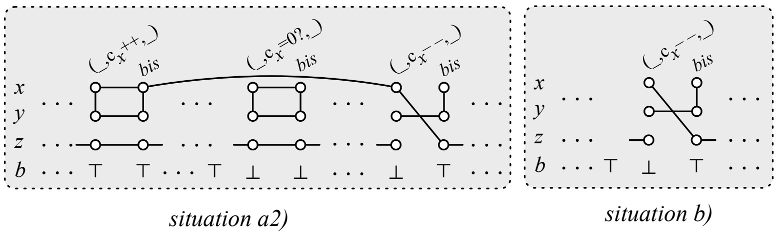


Figure A.2: Depiction of best strategies in both situations.

$$\pi_1 = b\mathsf{U}(\neg b \wedge \tau_{c_x=0?} \wedge \mathsf{X}(\neg b \mathsf{U} \tau_{c_x--}))$$

- A formula φ_1 constrains that z takes the same data value which is different than all the values of all other variables until the last $\perp$ is played by environment, from the next position z takes the current value of x forever.

$$\begin{aligned}\varphi_1 = {} & \neg(z \approx \Diamond x \vee z \approx \Diamond y) \wedge \\ & (z \approx \mathsf{X}z)\mathsf{U}\big(\neg(z \approx \mathsf{X}z) \wedge \neg b \wedge \mathsf{X}b \wedge x \approx \mathsf{X}z \wedge \mathsf{XG}(z \approx \mathsf{X}z)\big)\end{aligned}$$

- Further, environment can cheat in his denouncement by linking a $c_x = 0?$ transition with a future $c_x --$ with a matching $c_x ++$ that falls in-between zero test and the decrement.A formula π_1' can test this: there exists a $c_x ++$ with $\perp$ whose principal value matches that of a future z-value.
$$\pi_1' = \mathsf{F}(\tau_{c_x++} \wedge \neg b \wedge x \approx \Diamond z)$$

- If environment's denouncement was correct, the following formula ensures that there exists a $c_x ++$ transition whose principal value corresponds to the z-value of some future position.

$$\varphi_1' = \mathsf{F}(\tau_{c_x++} \wedge x \approx \Diamond z).$$

Thus, the encoding for this situation is expressed with the formula $\psi_1 = \psi_1^x \wedge \psi_1^y$ where $\psi_1^x = \mu \wedge ((\pi_1 \wedge \neg\pi_1') \Rightarrow (\varphi_1 \wedge \neg\varphi_1'))$ and same way we define ψ_1^y.

This situation is depicted in Figure A.2.

(b) Avoiding illegal decrements.

Supoose there is a decrementing transition $c_k --$ but it has no matching $c_k ++$ transition having the same value. To expose this kind of cheating by system (assume this is the first illegal transition made), environment plays $\perp$ at the current decrementing position and $\top$ in others. So, for this situation environment's best strategy has a value sequence from $\top^*\bot\top^\omega$, and the following property characterizes environment's denouncement of an illegal decrement.

Again we do it for $k = x$; the final formula will be a conjunct of two such formulas each for $k = x$ and $k = y$.

- Formula π_2 tests that b becomes $\bot$ at a $c_x --$ position and remains $\top$ at a $c_x --$ position thereafter.

$$\pi_2 = b\mathsf{U}(\neg b \wedge \tau_{c_x--} \wedge \mathsf{X}(\mathsf{G}b))$$

- Like φ_1, a formula can constrain the z value.

$$\begin{aligned}\varphi_2 = \neg(z \approx \Diamond x \vee z \approx \Diamond y) \wedge \\ (z \approx \mathsf{X}z)\mathsf{U}\big(\neg(z \approx \mathsf{X}z) \wedge \neg b \wedge x \approx \mathsf{X}z \wedge \mathsf{XG}(z \approx \mathsf{X}z)\big)\end{aligned}$$

- A formula φ_2' tests that in this situation there must be some $c_x ++$ position with a data value equal to variable z of a future position:

$$\varphi_2' = \mathsf{F}(\tau_{c_x++} \wedge x \approx \Diamond z).$$

Thus, the encoding for this situation is expressed with the formula $\psi_2 = \psi_2^x \wedge \psi_2^y$ where $\psi_2^x = \pi_2 \Rightarrow \varphi_2 \wedge \varphi_2'$ and same way we define ψ_2^y.

This situation is also depicted in Figure A.2.

The final formula to test is then of the form $\varphi = \varphi_{lab} \wedge \varphi_{x,y} \wedge \psi_1 \wedge \psi_2$, where φ_{lab} ensures the finite-automata behavior of labels, and in particular that a final state can be reached, and $\varphi_{x,y}$, ψ_1 and ψ_2 are described above.

Correctness. Suppose first that the 2-counter machine has an accepting run $(q_{init}, I_1, q_1) \cdots (q_{n-1}, I_n, q_n)$ with $q_n = q_f$. System's strategy is then to play (the encoding of) the labels

$$(q_0, I_1, q_1)\ bis \cdots (q_{n-1}, I_n, q_n)\ bis\, (\hat{t}\ bis)^\omega.$$

Therefore the formula φ_1 is satisfied. For assigning the data values to the variables, system will follow the rules depicted in Figure A.1 satisfying $\varphi_{x,y}$.

Also system will assign to z such a value which is different from all the data values in x and y until b becomes $\bot$. At the next position after the last $\bot$, it will assign the value of x at the position of last $\bot$ and keeps forever. If it is due to an illegal zero test (a2), $\varphi_1 \wedge \neg\varphi_1'$ is true making ψ_1 true, in the other case (b), $\varphi_2 \wedge \varphi_2'$ will be true making ψ_2 true; all other cases of b are also winning for system because antacedent will become false. Therefore this raises to a winning strategy for system player in the single-sided LRV[$\top$, $\approx$, $\rightarrow$] game.

Now suppose there is no accepting run for the counter machine from the initial configuration. Then for any play of system satisfying φ_{lab} and $\varphi_{x,y}$ must have an illegal transition of one kind where environment will play $\bot$. If it is at a decrementing transition, environment will again play $\top$ forever; if it was a zero test, *i.e.* of type (a), then it keeps playing $\bot$ until the corresponding c_k-- matching a witness to a witnessing c_k++ before the $c_k = 0?$ is reached (plays $\top$ afterwards). In either case, the antecedent of ψ_1 or ψ_2 will be true but the consequent will be false making one of them false. Hence]sys does not have any winning strategy in the LRV[$\top$, $\approx$, $\rightarrow$] game.

This concludes the proof of Theorem 15. $\square$