We are now ready to state the main language-related property of the region automaton construction.

**Proposition 2.** Let \( A \) be a timed automaton with set of clocks \( X \) and set of constraints \( C \). We assume we can construct a set of regions \( R \) for \( X \) and \( C \). Then,

\[
\text{Untime}(L(A)) = L(\Gamma_R(A))
\]

where \( L(\Gamma_R(A)) \) is the (untimed) language accepted by the finite automaton \( \Gamma_R(A) \) and \( \text{Untime}((a_1,t_1)\ldots(a_p,t_p)) = a_1\ldots a_p \) assigns to a timed word (and by extension to a timed language) a finite word (and by extension a classical language).

**Proof.** A standard induction on the length of runs allows to build runs in \( A \) and paths in \( \Gamma_R(A) \) which coincide in the following sense: there is a run \((\ell_0,v_0) \xrightarrow{d_1,a_1} \ldots \xrightarrow{d_p,a_p} (\ell_p,v_p)\) in \( A \) iff there is a path \((\ell_0,[v_0]_R) \xrightarrow{a_1} \ldots \xrightarrow{a_p} (\ell_p,[v_p]_R)\). This implies the expected result. □

**Remark 5.** In terms of behavioural equivalence, the finite automaton \( \Gamma_R(A) \) is the quotient of the timed transition system \( \tau_m^A \) w.r.t. the time-abstract bisimulation \( \cong_R \).

**Example 7.** We consider the following small timed automaton, and we would like to know whether location \( \ell_4 \) is reachable from the initial state \((\ell_0,0\{x,y\})\).

A possible set of regions for that automaton has been given in Example 5, and the corresponding region automaton is the following finite automaton.

Applying Proposition 2, we have that location \( \ell_4 \) is reachable in the original timed automaton iff the state \((\ell_4,R_1)\) is reachable in the region automaton (actually it should be iff any of the \((\ell_4,R_i)\) is reachable, but due to the constraint labelling the transition from \( \ell_1 \) to \( \ell_4 \), we can focus on \((\ell_4,R_1)\)). We see that it is not the case, and hence that there is no run which starts in the initial state \((\ell_0,0\{x,y\})\) and ends in \( \ell_4 \). □

For every timed automaton \( A \) for which we can **effectively** construct a finite set of regions \( R \) (satisfying conditions ①, ② and ③), we can transfer the checking of reachability properties in \( A \) to the finite automaton \( \Gamma_R(A) \). It remains to see how we can effectively build sets of regions for timed automata.
4.3 Effective construction of sets of regions

In the previous section, we have presented an abstract construction which allows to reduce for instance the model-checking of reachability properties in timed automata to the model-checking of reachability properties in finite automata (under the condition that there is a finite set of regions for the set of constraints used in the timed automaton). However, we did not explain how we construct a set of regions for timed automata, which is the basis to the whole construction.

In this section, we fix a finite set of clocks $X$.

Regions for sets of diagonal-free constraints. Let $M \in \mathbb{N}$ be an integer. We define a set of regions for $X$ and the set of $M$-bounded diagonal-free clock constraints $C_{df}^M(X)$. A natural partition would be to take the partition induced by the set of constraints itself, see Figure 3(a) for an illustration with two clocks. But this is actually not fine enough because compatibility condition (3) is not satisfied (as illustrated on the figure by the two gray valuations). A correct partition is then the one of Figure 3(b).

![Region construction for set of diagonal-free constraints](image)

Fig. 3: Region construction for set of diagonal-free constraints $C_{df}^M(\{x,y\})$

We formalize this idea and we define the equivalence relation $\equiv_{df}^{X,M}$ over $T^X$. We will give three different (equivalent) definitions.

(i) The first definition formalizes the 2-dimensional intuition that we have seen earlier. Let $v$ and $v'$ be two valuations of $T^X$, we say that $v \equiv_{df}^{X,M} v'$ if all three following conditions are satisfied:

(a) $v(x) > M$ iff $v'(x) > M$ for each $x \in X$,

(b) if $v(x) \leq M$, then $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$ and $\{v(x)\} = 0$ iff $\{v'(x)\} = 0$ for each $x \in X$,

9 $\lfloor \cdot \rfloor$ (resp. $\{ \cdot \}$) represents the integral (resp. fractional) part.
(c) if \( v(x) \leq M \) and \( v(y) \leq M \), then \( \{v(x)\} \leq \{v(y)\} \) iff \( \{v'(x)\} \leq \{v'(y)\} \) for all \( x, y \in X \).

The relation \( \equiv_{df}^{X,M} \) is an equivalence relation of finite index, and it naturally induces a finite partition \( \mathcal{R}_{df}^{M}(X) \) of \( \mathbb{T}^X \) (defined as the set of equivalence classes of \( \mathbb{T}^X / \equiv_{df}^{X,M} \)).

The construction for two clocks is precisely that illustrated on Figure 3(b).

(ii) We give a second description of the regions in \( \mathcal{R}_{df}^{M}(X) \), which makes it easier to give an upper bound on the number of regions in \( \mathcal{R}_{df}^{M}(X) \). An interval of \( \mathbb{T} \) with integral bounds is said \( M \)-\emph{simple} if it is of one of the following forms: \((c, c+1)\) with \( 0 \leq c < M \), or \([c, c]\) with \( 0 \leq c \leq M \), or \((M, +\infty)\). It is said bounded if it is one of the two first forms, and singular in the second form. Each region of \( \mathcal{R}_{df}^{M}(X) \) can then be characterized uniquely by:

- an \( M \)-simple interval \( I_x \) for every clock \( x \in X \), and
- a preorder \( \prec \) on the set of clocks

\[
Z_{(I_x)_{x \in X}} = \{ x \in X \mid I_x \text{ bounded and non-singular} \}.
\]

Intuitively the interval \( I_x \) is the interval to which \( x \) belongs, and the preorder is given by the preorder on the fractional parts of all clocks bounded by \( M \) with non-integral values.

Assume that region \( R \) is given by \((I_x)_{x \in X}\) and \( \prec \); then:

\[
v \in R \text{ iff } \forall x \in X, v(x) \in I_x \text{ and } \forall x, y \in Z_{(I_x)_{x \in X}}, (x \prec y \Leftrightarrow \{v(x)\} \leq \{v(y)\})
\]

(iii) We give a third description of the regions, which give an interesting one-dimensional understanding of the regions. It reuses elements of the characterization above. Each region of \( \mathcal{R}_{df}^{M}(X) \) can be characterized uniquely by:

- the set \( X_\infty = \{ x \in X \mid I_x = (M, +\infty) \} \)
- the set \( X_0 = \{ x \in X \setminus X_\infty \mid I_x = [c, c] \text{ for some } c \leq M \} \)
- a partition \((X_i)_{1 \leq i \leq p}\) of \( X \setminus (X_0 \cup X_\infty) \) such that:
  - for every \( 1 \leq i \leq p \), \( X_i \neq \emptyset \)
  - for \( x \in X \setminus (X_0 \cup X_\infty) \), writing \( i(x) \) for the unique index such that \( x \in X_{i(x)} \), for every \( x, y \in X \setminus (X_0 \cup X_\infty) \):
    \[
x \prec y \Leftrightarrow i(x) \leq i(y)
    \]

That is: clocks in the same \( X_i \) have the same fractional part, whereas fractional parts of clocks in two different \( X_i \)'s have their fractional parts ordered accordingly.

- for every \( x \in X \setminus X_\infty \), \( c_x \) is an integer bounded by \( M \), and if \( x \notin X_0 \), \( c_x < M \).

Note: sets \( X_0 \) and \( X_\infty \) can be empty (contrary to the other \( X_i \)'s).

The generic representation of such a region is as follows (it represents the interval \([0, 1]\) and shows the repartition of the clocks within that interval (according to their fractional part)):
Assume that region \( R \) is given by \( X_0, (X_i)_{1 \leq i \leq p}, X_\infty \) and \( (c_x)_{x \in X \setminus X_\infty} \); for every \( x \in X \setminus X_\infty \), write \( i(x) \in \{0,1,\ldots,p\} \) such that \( x \in X_i \). Then:

\[
\forall x \in X \setminus X_\infty, \quad v(x) = c_x + \gamma_{i(x)}
\]

For instance, the lightgray region depicted in Figure 3(b) has the following second characterization:

\[
\begin{cases}
  x \in (1,2) \\
  y \in (0,1) \\
  x < y, \ y \not< x \ (\text{meaning } \{x\} < \{y\})
\end{cases}
\]

and the following third characterization:

\[
\begin{align*}
  X_0 &= X_\infty = \emptyset \\
  X_1 &= \{x\}, \ X_2 = \{y\} \\
  c_x &= 1, \ c_y = 0
\end{align*}
\]

Exercise 1. Prove the equivalence of these definitions.

The following lemma states the correctness of the above partition.

**Proposition 3.** The partition \( R_M^{\mathcal{C}_M}(X) \) is a finite set of regions for clocks \( X \) and \( M \)-bounded diagonal-free clock constraints \( \mathcal{C}_M^M(X) \).

**Proof.** Conditions ① and ③ are rather easy to check, we thus omit the details. The case of condition ② requires more careful developments.

We will base the proof on the third characterization of the regions, and first define a “next successor” operation on the regions: the region \( r_\infty \) defined by \( X_\infty = X \) has no next successor; given a region \( r \) characterized by \( (X_i)_{0 \leq i \leq p, i=\infty} \) and \( (c_x)_{x \in X \setminus X_\infty} \), its next successor \( \text{succ}(r) \), characterized by \( (X'_i)_{0 \leq i \leq p, i=\infty} \) and \( (c'_x)_{x \in X \setminus X_\infty} \), is defined as follows:

- if \( X_0 \neq \emptyset \) and \( \{x \in X_0 \mid c_x < M\} \neq \emptyset \),

\[
\begin{align*}
  X'_0 &= \emptyset \\
  X'_\infty &= X_\infty \cup \{x \in X_0 \mid c_x = M\} \\
  X'_1 &= \{x \in X_0 \mid c_x < M\} \\
  X'_{i+1} &= X_i, \ \forall 1 \leq i \leq p \\
  c'_x &= c_x, \ \forall x \in X \setminus X'_\infty
\end{align*}
\]
if $X_0 \neq \emptyset$ and $\{x \in X_0 \mid c_x < M\} = \emptyset$,

\[
\begin{cases}
X'_0 = \emptyset \\
X'_\infty = X_\infty \cup X_0 \\
X'_i = X_i, \forall 1 \leq i \leq p \\
c'_x = c_x, \forall x \in X \setminus X'_\infty
\end{cases}
\]

- if $X_0 = \emptyset$,

\[
\begin{cases}
X'_\infty = X_\infty \\
X'_0 = X_p \\
X'_i = X_i, \forall 1 \leq i < p \\
c'_x = c_x, \forall x \in \bigcup_{1 \leq i \leq p} X_i \\
c'_x = c_x + 1, \forall x \in X'_{0}
\end{cases}
\]

Indeed, time elapsing on that representation is obtained by circular translation of the sets of clocks, with an absorbing set $X_\infty$ when clocks go above the maximal constant.

Find below a (rough) representation of the two first cases, and then of the third case:

Formally:

**Lemma 4.** Let $r \neq r_\infty$ be a region such that $\text{succ}(r)$ is defined. For every $v \in r$, there exists $t \in \mathbb{T}$ such that $v + t \in \text{succ}(r)$, and for all $0 \leq t' \leq t$, $v + t' \in r \cup \text{succ}(r)$.

**Proof.** Let $v \in r$, and let $\gamma_i$ be the fractional part of the clocks in $X_i$, assuming $\gamma_0 = 0$: for every $x \in X \setminus X_\infty$, $v(x) = c_x + \gamma_i(x)$.

If we are in one of the two first cases above, we let $t = \frac{1-\gamma}{2}$, and set $v' = v + t$. It is easy to check that $v' \in \text{succ}(r)$, and that is also the case for all $v + t'$ with $0 \leq t' \leq t$. 

20
Then, assume we are in the third case, and let $t = 1 - \gamma_p$. Setting $v' = v + t$, we get the expected result as well.

Let $r$ be a region. We define $\text{succ}^0(r) = r$, and $\text{succ}^{i+1}(r) = \text{succ}(\text{succ}^i(r))$ if it is defined. Then $\text{succ}^*(r) = \bigcup_{i \geq 0} \text{succ}^i(r)$. We then prove the following lemma:

**Lemma 5.** Let $r$ be a region and let $v \in r$. For every $t \in \mathbb{T}$, $v + t \in \text{succ}^*(r)$.

*Proof.* We show the result by induction on the number of regions included in $\text{succ}^*(r)$. The basis case is when $\text{succ}^*(r)$ only includes $r$, which can only happen when $r = r_\infty$: this is obvious since for every $v \in r_\infty$, for every $t \in \mathbb{T}$, $v + r \in r_\infty$.

Pick $n \in \mathbb{N}_{\geq 1}$, and assume we have proven the result for every region $r$ such that $\text{succ}^*(r)$ contains at most $n$ regions. Pick now a region $r$ such that $\text{succ}^*(r)$ contains $n + 1$ regions. Towards a contradiction assume that the expected property does not hold for $r$, and choose $v \in r$ and $t \in \mathbb{T}$ such that $v + t \notin \text{succ}^*(r)$. Notice first that $r \neq r_\infty$, hence $\text{succ}(r)$ is well-defined. Let $t_1$ be such that $v + t_1 \in \text{succ}(r)$. Because of the previous lemma, it must be the case that $t > t_1$. Letting $r_1 = \text{succ}(r)$, $v_1 = v + t_1 \in r_1$, with delay $t - t_1$, we can apply the induction hypothesis, and obtain a contradiction.

We are now ready to prove property 2. Let $r$ be a region. Applying Lemma 4, we show by induction no $i$ that for every $i \in \mathbb{N}$, for every $v \in r$, there is $t_i$ such that $v + t_i \in \text{succ}^i(r)$. Lemma 5 shows the completeness, since for every $v \in r$, for every $t \in \mathbb{T}$, there is $i$ such that $v + t \in \text{succ}^i(r)$.

Hence, condition 2 is satisfied by the partition $\mathcal{R}_{df}^M(X)$.

**Lemma 6.** The number of regions in $\mathcal{R}_{df}^M(X)$ is bounded by $(M + 1)^{|X|}.|X|!.2^{|X|+2}$ where $|X|$ is the cardinal of $X$.

*Proof.* We use the third characterization of regions to get the expected upper bound. In this representation, we can assume that $c_x = M$ whenever $x \in X_\infty$. The factor $(M + 1)^{|X|}$ is then for the choice of each constant $c_x$ (for $x \in X$). Now, to understand the rest of the formula, let us write down the list of the clocks by starting with $X_0$, then $X_1, \ldots, X_p$ and finally $X_\infty$. There are $|X|$ such lists. Finally we distinguish all first elements of each set (plus two Booleans for checking whether $X_0$ or $X_\infty$ is empty or not), yielding the factor $2^{|X|+2}$.

Note: this bound is obviously not optimal; indeed, the order in which we list the various sets $X_i$ is not important.

Note that a tighter bound can be obtained, but the computation is much more involved, see [Kop96].\(^{10}\)

\(^{10}\) Thanks to Luca Aceto who pointed out this reference.
Regions for sets of general constraints. Let $M \in \mathbb{N}$ be an integer. The aim is to define a set of regions for $X$ and the set of $M$-bounded clock constraints $C^M(X)$. The partition we have defined in the previous paragraph is no more compatible with the set of constraints $C^M(X)$, we thus need to refine it.

We define an equivalence relation $\equiv^{X,M}$ over $T^X$ as follows: let $v$ and $v'$ be two valuations of $T^X$, we say that $v \equiv^{X,M} v'$ if all two following conditions hold:

- $v \equiv^{X,M} v'$, and
- $v \models (x - y \sim c)$ iff $v' \models (x - y \sim c)$ for every $(x - y \sim c) \in C^M(X)$.

This equivalence relation refines $\equiv_{df}^{X,M}$ in that two equivalent valuations satisfy in addition the same diagonal constraints (bounded by $M$). This new partition is denoted $R^M(X)$ and is illustrated for two clocks on Figure 4.

![Fig. 4: Set of regions $R^2(X)$ for 2-bounded clock constraints with two clocks](image)

We omit the proof of the following proposition, which states the correctness of the refinement.

**Proposition 4.** The partition $R^M(X)$ is a set of regions for $X$ and the set of $M$-bounded clock constraints $C^M(X)$.

As previously, each region of $R^M(X)$ can be characterized by:

- an $M$-simple interval $I_x$ for every clock $x \in X$, and
- an $M$-simple interval, or minus an $M$-simple interval, $J_{x,y}$ for all clocks $x, y \in X$.

Intuitively the interval $I_x$ gives the constraint on clock $x$, and $J_{x,y}$ gives the constraint on the difference $x - y$. The order between fractional parts of two clocks $x$ and $y$ can now be inferred from the intervals $I_x, I_y$ and $J_{x,y}$.

As a direct consequence of this characterization, we get the following upper bound on the number of regions.

**Lemma 7.** The number of regions in $R^M(X)$ is bounded by $(4M + 3)(|X| + 1)^2$ where $|X|$ is the cardinal of $X$.

**Remark 6.** Note that $R^M(X)$ is also a set of regions for the set of constraints $C^M_{df}(X)$.
Remark 7. Note that sets of regions we have described could be made smaller: there is no need to have the same maximal constant for all clocks, one maximal constant for each clock could be used. However, for the purpose of these notes, there is no need for such a refinement.

4.4 An example of region automaton [AD94]

Consider the following timed automaton:

![Diagram of a timed automaton]

It has two clocks and maximal constant 1. The set of regions can be described as follows:

The corresponding region automaton is therefore:

![Diagram of the region automaton]

Many things can be read on the region automaton... e.g. Zeno cycles.

4.5 Some exercices

Exercise 2. We assume we can update clocks with more involve operations than resets to zero. For this exercise, we write $U$ for a finite set of functions $u_p : T^X \rightarrow T^X$. Those updates
can be put on transitions in place of resets, and when firing the transition, the valuation
is updated according to this function.

How should we strengthen conditions 1, 2 and 3 in the construction of a finite set of
regions to take into account those new updates?

Can you build a (finite) set of regions in the following cases? Explain.

– diagonal-free clock constraints, and updates

\[ U = \{ x := 0, x := x + 1 \mid x \in X \} \]

where, writing up for ‘\( x := x + 1 \)’ up\( (v) \) is the valuation assigning \( v(x) + 1 \) to \( x \) and
\( v(y) \) to \( y \neq x \);

– general clock constraints, and updates

\[ U = \{ x := 0, x := x + 1 \mid x \in X \} \]

– diagonal-free clock constraints, and updates

\[ U = \{ x := 0, x := x - 1 \mid x \in X \} \]

– general clock constraints, and updates

\[ U = \{ x := 0, x := x - 1 \mid x \in X \} \]

\[ \rightleftharpoons \]

Exercise 3. Prove that for any timed automaton \( A \), we can construct a diagonal-free timed
automaton \( B \) that recognizes the same timed language. What is the size of \( B \)? Do you think
we can avoid that blowup?

Exercise 4. If we add linear constraints to the model, i.e., constraints of the form \( \sum_{x \in X} \alpha_x x \sim c \) with \( \alpha_x \in \mathbb{Z} \) for every \( x \in X \) and \( c \in \mathbb{Z} \), prove that the reachability problem becomes
undecidable.

5 Complexity issues

5.1 Application to the reachability problem

Let \( A \) be a timed automaton with set of clocks \( X \). Let \( M \) be the maximal constant involved
in one of the constraints of \( A \), the set \( R^M(X) \) (or even \( R_{\text{df}}^M(X) \) in case \( A \) is a diagonal-
free timed automaton) is a set of regions for \( A \). Hence the region abstraction and all the
developments made in the previous section can be used. The following result, due to Alur
and Dill [AD90,AD94], is the core of the verification of timed systems.
Theorem 2. The reachability problem is decidable for timed automata. It is a PSPACE-complete problem (for both diagonal-free and general timed automata).

Although this theorem has been first stated in [AD90,AD94], the proof we present here is taken from [AL02].

Proof. To prove PSPACE membership, we use the region automaton construction, and check reachability properties in this finite abstraction (see Proposition 2). Applying Lemmas 6 and 7, we know that the size of this finite automaton is exponential in the size of the original timed automaton (in both diagonal-free and general cases). Moreover, using the characterizations of the regions we have given we know that each state of the finite automaton can be stored in polynomial space, and, given a state, we can guess a successor in polynomial space. Hence, we apply the classical NLOGSPACE algorithm for checking reachability properties in finite automata, and get a PSPACE algorithm for checking reachability properties in the original timed automaton.

The PSPACE-hardness can be proved by reducing the termination of a (non-deterministic) linearly bounded Turing machine (LBTM for short) on some given input to the reachability problem in timed automata. We present the proof for general timed automata, and then explain informally how it can be extended to diagonal-free timed automata. The complete reduction can be found in the appendices of [AL02].

Let $\mathcal{M}$ be a LBTM and $w_0$ an input for $\mathcal{M}$. We write $N$ for a (linear) bound on the length of the tape which is used when $\mathcal{M}$ executes on $w_0$. We encode the behaviour of $\mathcal{M}$ on $w_0$ as the behaviour of a timed automaton. The encoding is as follows: assuming the alphabet of $\mathcal{M}$ is $\{a, b\}$ (this can be done w.l.o.g.), and writing $\#$ for the blank symbol, the content of cell $C_i$ of the tape of the LBTM is encoded by a constraint on two clocks $x_i$ and $y_i$. Cell $C_i$ contains a symbol $a$ when the constraint $x_i = y_j$ holds, and cell $C_j$ contains a symbol $b$ when the constraint $x_j < y_j$ holds. If the cell is empty (or equivalently contains the blank symbol $\#$), then the constraint $x_k > y_k$ holds. Note that these three constraints are invariant by time elapsing. This is illustrated on Figure 5.

We assume that the set of states of $\mathcal{M}$ is $Q$, its initial state is $q_0$, and its halting state is $q_F$. We construct a timed automaton $\mathcal{A} = (L, L_0, L_F, X, \Sigma, T)$ as follows:

- $L = (Q \times \{1, \ldots, N\}) \cup \{\text{init}\}$;
- $L_0 = \{\text{init}\}$;
- \( L_F = \{(q_F, i) \mid i \in \{1, \ldots, N\}\} \);
- \( X = \{x_i, y_i \mid i \in \{1, \ldots, N\}\} \cup \{u\}; \)
- \( \Sigma = \{a\}; \)
- For every rule of \( M \), we will have several transitions in \( T \). We define the constraint \( g_{a,i} \) by \((u > 0 \land x_i < y_i)\), the constraint \( g_{b,i} \) by \((u > 0 \land x_i = y_i)\), and the constraint \( g_{\#_i} \) by \((u > 0 \land x_i > y_i)\). Those constraints express that cell \( i \) contains an \( a \), respectively an \( b \) and a blank character. We define the resetting sets \( Y_{a,i} = \{u, x_i, y_i\} \) and \( Y_{b,i} = \{u, x_i\} \). Those are sets of clocks to be reset for expressing the fact that we write an \( a \), respectively an \( b \), in cell \( i \).
  - Consider a rule \((q, \text{Read}_a, \text{Write}_\beta, \text{Right}, q')^{11} \) in \( M \). For every \( i \in \{1, \ldots, N - 1\}\), there is a transition \((q, i) \xrightarrow{a, Y_{\beta,i}} (q', i + 1)\) in \( T \).
  - Consider a move \((q, \text{Read}_a, \text{Write}_\beta, \text{Left}, q')\) in \( M \). For every \( i \in \{2, \ldots, N\}\), there is a transition \((q, i) \xrightarrow{a, Y_{\beta,i}} (q', i - 1)\) in \( T \).
  There is an extra transition from state \text{init} to initialize the input word \( w_0 \) on the tape: \( \text{init} \xrightarrow{\emptyset, Y_{\text{init}}} (q_0, 1) \) where \( Y_{\text{init}} = \{u\} \cup \{x_i, y_i \mid w_0(i) = a\} \cup \{x_i \mid w_0(i) = b\} \cup \{y_i \mid i > |w_0|\} \) (\( w_0(i) \) denotes the \( i \)-th letter of \( w_0 \)).

We claim that there is a halting computation in \( M \) iff \( L(A) \neq \emptyset \), and we let the careful reader get convinced of this equivalence. As the halting problem for LBTMs is \text{PSPACE}-hard, we get the expected lower bound.

See next exercise for the case of diagonal-free clock constraints.

\textit{Remark 8}. In [CY92], a proof of \text{PSPACE}-hardness is given for diagonal-free timed automata with only three clocks, it is rather technical, hence we have chosen not to present it here.

\textit{Exercise 5}. Think of a \text{PSPACE} lower bound reduction for diagonal-free timed automata.

\textit{Proof (Solution to this exercise)}. We use a proof which is close to the previous one. We can nevertheless not use diagonal constraints, which were very convenient for encoding the content of the cells (since diagonal constraints are invariant by time elapsing).

Let \( N \) be the bound on the tape of \( M \) when simulating on input word \( w_0 \). We assume the alphabet is \( \{a, b\} \) and we encode the content of \( j \)-th cell \( C_j \) using a clock \( x_j \) with the following convention: when we enter a module, cell \( C_j \) contains an \( a \) whenever \( x_j < 1 \) and it contains a \( b \) whenever \( x_j > 2 \). The simulation of a transition \((q, \text{Read}_a, \text{Write}_\beta, \text{Right}, q')\) is given on the figure below (we assume \( i + 1 \leq N \), so that this transition is meaningful). Guard \( g_{a,i} \) is \( x_i < 4, u < 3 \) whereas guard \( g_{b,i} \) is \( x_i > 4, u < 3 \). Set \( Y_{a,i} \) is \( \{x_i\} \) and set \( Y_{b,i} \) is \( \emptyset \).

\[\begin{array}{cccc}
& u := 0 & 2 < u < 3 & x_1 < 4, u < 3 \\
q_i & x_1 := 0 & x_2 := 0 & x_2 < 4, u < 3 \\
& x_1 > 4, u < 3 & x_2 > 4, u < 3 & \cdots \\
& \cdots & g_{a,i}, Y_{b,i} & \cdots \\
& \cdots & x_N \leq 0 & \emptyset, q', i + 1 \\
& x_N > 4, u < 3 & x_N \leq 0 & \emptyset, q', i + 1 \\
& x_N > 4, u < 3 & x_N > 4, u < 3 & \cdots \\
\end{array}\]

\(^{11}\) This rule reads as follows: from state \( q \), if we read an \( a \) in the current cell of the tape, then we write a \( \beta \) onto the current cell, move the head of the tape to the right and go to state \( q' \).
The case of a transition \((q, \text{Read}_\alpha, \text{Write}_\beta, \text{Right}, q')\) is very similar, but requires \(i - 1 \geq 0\), and then goes to \((q', i - 1)\) instead of \((q', i + 1)\).

5.2 The case of ‘simply-timed’ timed automata

The previous complexity result holds for timed automata with three clocks or more (the proof of three clocks is rather involved, though). For some simpler systems, for instance for systems with a single clock, this result can be improved. Of course, the same set of regions cannot be used, because even though the number of clocks is one, the number of regions given in Lemma 6 remains exponential, due to the binary encoding of constants in the timed automaton. However we can choose a smaller and rougher set of regions which yields the following result, due to [LMS04].

**Proposition 5.** The reachability problem for single-clock timed automata is NLOGSPACE-complete.

**Proof.** NLOGSPACE-hardness follows from that of reachability in finite graphs [HU79].

The NLOGSPACE membership can be obtained using a rougher set of regions than that presented in Section 4.3. Given a finite set \(\mathcal{C}\) of constraints over a single clock \(x\), we define the set of constants \(\mathcal{C} = \{c \in \mathbb{N} \mid \exists (x \gg c) \in \mathcal{C}\} \cup \{0\}\), and we assume that this set is ordered: \(\mathcal{C} = \{c_0 < c_1 < c_2 < \cdots < c_p\}\). We define the partition \(\mathcal{R}_\mathcal{C}\) as the (finite) set of intervals of one of the forms: (i) \(\{c_i\}\) with \(0 \leq i \leq p\), or (ii) \((c_i, c_{i+1})\) with \(0 \leq i < p\), or (iii) \((c_p, +\infty)\). This is not hard to prove that \(\mathcal{R}_\mathcal{C}\) is a set of regions for \(\{x\}\) and \(\mathcal{C}\). The size of \(\mathcal{R}_\mathcal{C}\) is polynomial in the size of \(\mathcal{C}\), which yields a polynomial-size region automaton, hence the expected result.

The case of two-clock timed automata has recently been solved in the literature [FJ15] and proved to be PSPACE-complete. For a long time, this was known to be NP-hard only. The proof is long and technical, therefore it is omitted here.

**Exercise 6.** Prove an NP lower bound for the reachability problem in two-clocks timed automata.