Principles of Model Checking, C. Baier & J.-P. Katoen

Note that the resulting formulae are CTL formulae (or could be understood as such) provided Φ does not contain intervals different from $[0,\infty)$. ■

In order to verify whether $TA \models \Phi$ for TCTL formula Φ, the above result suggests equipping TA with a clock for each subformula of Φ of the form $\Psi \mathsf{U}^J \Psi'$ while replacing this subformula as indicated in Theorem 9.37. This yields $\text{TCTL}_\Diamond$ formula $\widehat{\Phi}$. As $\widehat{\Phi}$ does not contain timing parameters, and any clock constraint can be considered as an atomic proposition, in fact, $\widehat{\Phi}$ is a CTL formula! Verifying a timed CTL formula on a timed automaton TA thus reduces to checking a CTL formula on a TA extended with a clock whose sole purpose is to measure the elapse of time that is referred to in the formula.

9.3.2 Region Transition Systems

Consider timed automaton TA and $\text{TCTL}_\Diamond$ formula Φ. It is assumed that TA is equipped with an additional clock as explained in the previous section. The idea is impose an appropriate equivalence, denoted $\cong$, on the clock valuations—and implicitly on the states of $TS(TA)$ by letting $\langle \ell', \eta' \rangle \cong \langle \ell, \eta' \rangle$ if $\ell = \ell'$ and $\eta \cong \eta'$—such that:

(A) Equivalent clock valuations should satisfy the same clock constraints that occur in TA and Φ:

$$\eta \cong \eta' \;\Rightarrow\; (\eta \models g \quad \text{iff} \quad \eta' \models g \text{ for all } g \in ACC(TA) \cup ACC(\Phi))$$

where $ACC(TA)$ and $ACC(\Phi)$ denote the set of atomic clock constraints that occur in TA and Φ, respectively. These constraints are either of the form $x \leqslant c$ or $x < c$.

(B) Time-divergent paths emanating from equivalent states should be "equivalent". This property guarantees that equivalent states satisfy the same path formulae.

(C) The number of equivalence classes under $\cong$ is finite.

In the sequel we adopt the following notation for clock values.

Notation 9.39. Integral and Fractional Part of Real Numbers

Let $d \in \mathbb{R}$. The *integral part* of d is the largest integer that is at most d:

$$\lfloor d \rfloor \;=\; \max\{\, c \in \mathbb{N} \mid c \leqslant d \,\}.$$

The *fractional part* of d is defined by $frac(d) \;=\; d - \lfloor d \rfloor$. For example, $\lfloor 17.59267 \rfloor = 17$, $frac(17.59267) = 0.59267$, $\lfloor 85 \rfloor = 85$, and $frac(85) = 0$. ■

The definition of clock equivalence is based on three observations that successively lead to a refined notion of equivalence. Let us discuss these observations in detail.

First observation. Consider atomic clock constraint g, and let η be a clock valuation (both over the set C of clocks with $x \in C$). As g is an atomic clock constraint, g is either of the form $x < c$ or $x \leqslant c$ for $c \in \mathbb{N}$. We have that $\eta \models x < c$ whenever $\eta(x) < c$, or equivalently, $\lfloor \eta(x) \rfloor < c$. The fractional part of $\eta(x)$ in this case is not relevant. Similarly, $\eta \models x \leqslant c$ whenever either $\lfloor \eta(x) \rfloor < c$, or $\lfloor \eta(x) \rfloor = c$ and $frac(x) = 0$. Therefore, $\eta \models g$ only depends on the integral part $\lfloor \eta(x) \rfloor$, and the fact whether $frac(\eta(x)) = 0$. This leads to the initial suggestion that clock valuations η and η' are equivalent (denoted $\cong_1$) whenever

$$\lfloor \eta(x) \rfloor \; = \; \lfloor \eta'(x) \rfloor \quad \text{and} \quad frac(\eta(x)) = 0 \text{ iff } frac(\eta'(x)) = 0. \tag{9.1}$$

This constraint ensures that equivalent clock valuations satisfy the clock constraint g provided g only contains atomic clock constraints of the form $x < c$ or $x \leqslant c$. (In case one would restrict all atomic clock constraints to be strict, i.e., of the form $x < c$, the fractional parts would not be of importance and the second conjunct in the above equation may be omitted.) Note that it is crucial for this observation that only *natural number* constants are permitted in the clock constraints. This equivalence notion is rather simple, leads to a denumerable (but still infinite) number of equivalence classes, but is too coarse.

Example 9.40. A First Partitioning for Two Clocks

To exemplify the kind of equivalence classes that one obtains, consider the set of clocks $C = \{ x, y \}$. The quotient space for C obtained by suggestion (9.1) is depicted in Figure 9.18) where the equivalence classes are

- the corner points (q, p)
- the line segments $\{ (q, y) \mid p < y < p{+}1 \}$ and $\{ (x, p) \mid q < x < q{+}1 \}$, and
- the content of the squares $\{ (x, y) \mid q < x < q{+}1 \; \wedge \; p < y < p{+}1 \}$

where $p, q \in \mathbb{N}$ and $\{ (x, p) \mid q < x < q{+}1 \}$ is a shorthand for the set of all clock evaluations η with $\eta(x) \in \,]q, q{+}1[$ and $\eta(y) = p$. ■

Second observation. We demonstrate the fact that $\cong_1$ is too coarse by means of a small example. Consider location ℓ whose two outgoing transitions are guarded with $x \geqslant 2$ (action α) and $y > 1$ (action β), respectively; see also Figure 9.19. Let state $s = \langle \ell, \eta \rangle$ with $1 < \eta(x) < 2$ and $0 < \eta(y) < 1$. Both transitions are disabled, so the only possibility is to let time advance. The transition that is enabled next depends on the ordering of the

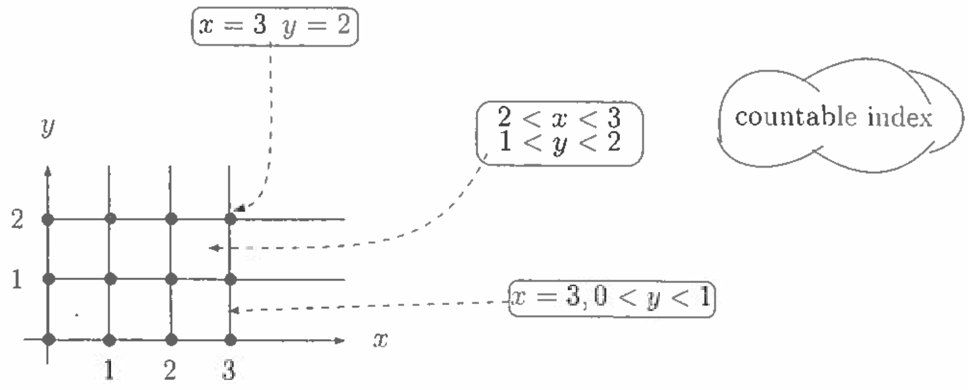


Figure 9.18: Initial partitioning for two clocks .

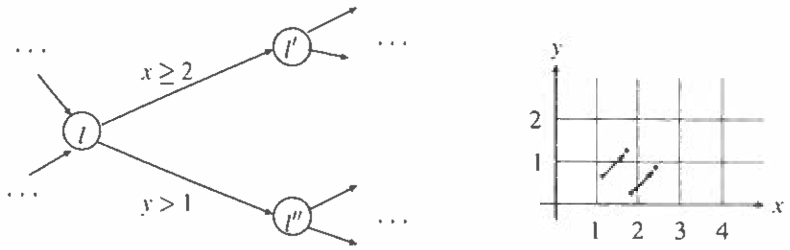


Figure 9.19: Fragment of timed automaton and time passage of two clock valuations.

fractional parts of the clocks x and y: if $frac(\eta(x)) < frac(\eta(y))$, then β is enabled before α; if $frac(\eta(x)) \geqslant frac(\eta(y))$, action α is enabled first. Time-divergent paths in s may thus start with α if $frac(\eta(x)) \geqslant frac(\eta(y))$, and with β otherwise. This is represented by the fact that delaying leads to distinct successor classes depending on the ordering of the fractional parts of clock, see Figure 9.19 (right part).

Thus, besides $\lfloor \eta(x) \rfloor$ and the fact whether $frac(\eta(x)) = 0$, apparently the *order of the fractional parts* of $\eta(x)$, $x \in C$ is important as well, i.e., whether for $x, y \in C$:

$$frac(\eta(x)) < frac(\eta(y)) \text{ or } frac(\eta(x)) > frac(\eta(y)) \text{ or } frac(\eta(x)) = frac(\eta(y)).$$

This suggests extending the initial proposal (9.1) for all $x, y \in C$ by

$$frac(\eta(x)) \leqslant frac(\eta(y)) \quad \text{if and only if} \quad frac(\eta'(x)) \leqslant frac(\eta'(y)), \tag{9.2}$$

i.e., $\eta_1 \cong_2 \eta_2$ iff $\eta_1 \cong_1 \eta_2$ and (9.2) holds. This strengthening will ensure that equivalent states $\langle \ell, \eta \rangle$ and $\langle \ell, \eta' \rangle$ have similar time-divergent paths.

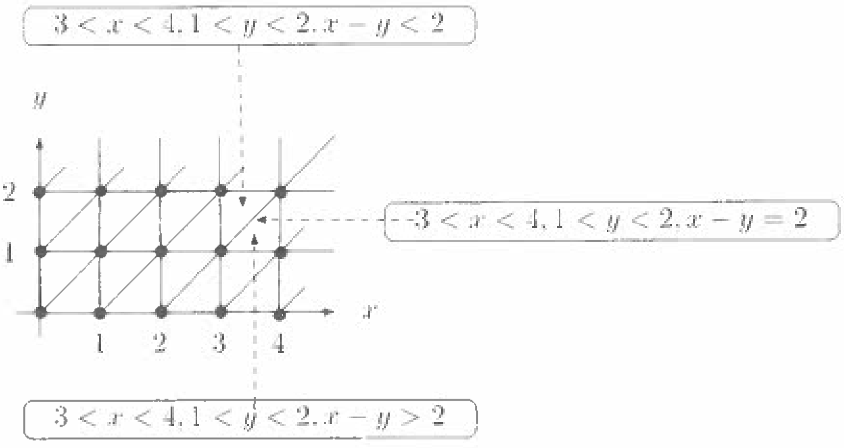


Figure 9.20: Refining the initial partitioning for two clocks.

Example 9.41. *A Second Partitioning for Two Clocks*

This observation suggests to decompose the squares $\{ (x, y) \mid q < x < q+1 \ \wedge \ p < y < p+1 \}$ into a line segment, an upper and lower triangle, i.e., the following three parts:

$$\{ (x, y) \mid q < x < q+1 \ \wedge \ p < y < p+1 \ \wedge \ x-y < q-p \},$$
$$\{ (x, y) \mid q < x < q+1 \ \wedge \ p < y < p+1 \ \wedge \ x-y > q-p \}, \text{ and}$$
$$\{ (x, y) \mid q < x < q+1 \ \wedge \ p < y < p+1 \ \wedge \ x-y = q-p \}.$$

Figure 9.20 illustrates the resulting partitioning for two clocks. ■

Final observation. The above constraints on clock equivalence yield a denumerable though not finite quotient. To obtain an equivalence with a *finite* quotient, we exploit the fact that in order to decide whether $TA \models \Phi$ only the clock constraints occurring in TA and Φ are relevant. As there are only finitely many clock constraints, we can determine for each clock $x \in C$ the maximal clock constraint, $c_x \in \mathbb{N}$, say, with which x is compared in some clock constraint in either TA (as guard or location invariant) or Φ. Since c_x is the largest constant with which clock x is compared it follows that if $\eta(x) > c_x$, the actual value of x is irrelevant. (Clock x that occurs neither in TA nor in Φ is superfluous and can be omitted: for these clocks we set $c_x = 0$.) As a consequence, the constraints (9.1) are only relevant if $\eta(x) \leqslant c_x$ and $\eta'(x) \leqslant c_x$, while for (9.2) in addition $\eta(y) \leqslant c_y$ and $\eta'(y) \leqslant c_y$.

The above considerations suggest the following notion of clock equivalence.

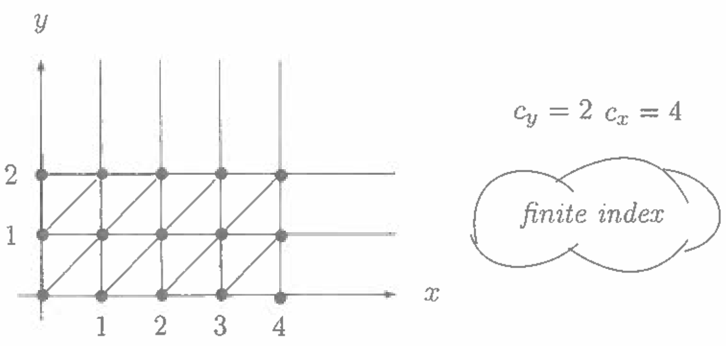


Figure 9.21: Third (and final) partitioning for two clocks (for $c_x = 4$ and $c_y = 2$).

Definition 9.42. Clock Equivalence $\cong$

Let TA be a timed automaton, Φ a TCTL$_\Diamond$ formula (both over set C of clocks), and c_x the largest constant with which $x \in C$ is compared with in either TA or Φ. Clock valuations $\eta, \eta' \in Eval(C)$ are *clock-equivalent*, denoted $\eta \cong \eta'$ if and only if either

- for any $x \in C$ it holds that $\eta(x) > c_x$ and $\eta'(x) > c_x$, or
- for any $x, y \in C$ with $\eta(x), \eta'(x) \leqslant c_x$ and $\eta(y), \eta'(y) \leqslant c_y$ all the following conditions hold:
  - $\lfloor \eta(x) \rfloor = \lfloor \eta'(x) \rfloor$ and $frac(\eta(x)) = 0$ iff $frac(\eta'(x)) = 0$,
  - $frac(\eta(x)) \leqslant frac(\eta(y))$ iff $frac(\eta'(x)) \leqslant frac(\eta'(y))$.

■

As the clock equivalence $\cong$ depends on TA and Φ, strictly speaking one should write $\cong_{TA,\Phi}$ instead of $\cong$. The dependency of $\cong$ on TA and Φ is limited to the largest constants c_x; that is to say, neither the structure of TA nor that of Φ is of relevance to clock equivalence. The equivalence $\cong$ is lifted to states of the transition system $TS(TA)$ as follows. For states $s_i = \langle \ell_i, \eta_i \rangle$, $i = 1, 2$, in $TS(TA)$:

$$s_1 \cong s_2 \quad \text{iff} \quad \ell_1 = \ell_2 \quad \text{and} \quad \eta_1 \cong \eta_2.$$

Equivalence classes under $\cong$ are called *clock regions*.

Definition 9.43. Clock and State Region

Let $\cong$ be a clock equivalence on C. The *clock region* of $\eta \in Eval(C)$, denoted $[\eta]$, is defined by

$$[\eta] \;=\; \{\, \eta' \in Eval(C) \mid \eta \cong \eta' \,\}.$$

The *state region* of $s = \langle \ell, \eta \rangle \in TS(TA)$, denoted $[s]$, is defined by

$$[s] \;=\; \langle \ell, [\eta] \rangle \;=\; \{\, \langle \ell, \eta' \rangle \mid \eta' \in [\eta] \,\}.$$

■

In the sequel, state and clock regions are often indicated as regions whenever it is clear from the context what is meant. Clock regions will be denoted by r, r', and so forth. We often use casual notations to denote clock regions or clock valuations. For a timed automaton with two clocks, x and y say,

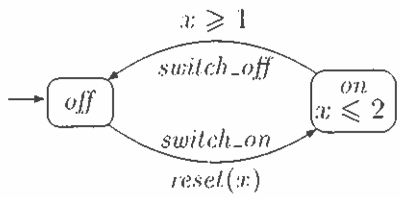
$$\{(x, y) \mid 1 < x < 2, 0 < y < 1, x - y < 1\}$$

denotes the clock region of all clock valuations $\eta \in Eval(\{\, x, y \,\})$ with

$$1 < \eta(x) < 2 \quad \text{and} \quad 0 < \eta(y) < 1 \quad \text{and} \quad frac(\eta(x)) < frac(\eta(y)).$$

Example 9.44. Light Switch

Consider the timed automaton over $C = \{\, x \,\}$ for the light switch and the $\mathrm{TCTL}_\Diamond$ formula $\Phi = \mathrm{true}$. It follows that the largest constant with which x is compared is $c_x = 2$; this is due to the location invariant $x \leqslant 2$.



We gradually construct the regions for this timed automaton by considering each of the constraints in Definition 9.42 separately. Clock valuations η, η' are equivalent if $\eta(x)$ and $\eta'(x)$ belong to the same equivalence class along the real line. (In general, for n clocks this amounts to considering an n-dimensional hyperspace on $\mathbb{R}_{\geqslant 0}$.)

1. The requirement that $\eta(x) > 2$ and $\eta'(x) > 2$ or $\eta(x) \leqslant 2$ and $\eta'(x) \leqslant 2$ yields the partitioning into the intervals $[0, 2]$ and $(2, \infty)$.