Duration Calculus

1. Symbols : $true, false, =, <, >, \leq, \geq, \quad f, g, \quad X, Y, Z, \quad x, y, z$

1.1 Predicate Symbols: $true, false, =, <, >, \leq, \geq$ $\qquad \mathbb{R}^n \to \mathbb{B}$

1.2 Function Symbols: $+, -$ $\qquad \mathbb{R}^n \to \mathbb{R}$

1.3 State variables and Domain values: State variable $X \to \mathcal{D}(X) = \{d_1, \ldots, d_n\}$

$\mathcal{I} : Obs \to (\text{Time} \to \mathcal{D})$

$\mathcal{I}(X) : \text{Time} \to \mathcal{D}(X)$

$\mathcal{I}(X)(t) \in \mathcal{D}(X)$ denotes value X has at time $t \in$ Time

1.4 Global Variables: x, y, z $\qquad \mathcal{V} : \text{GVar} \to \mathbb{R}$

2. State Assertions: $P ::= 0 \mid 1 \mid X = d \mid \neg P_1 \mid P_1 \wedge P_2$

2.1 Semnatics: $\mathcal{I}[\![P]\!] : \text{Time} \to \{0, 1\}$

$\underbrace{\mathcal{I}[\![X]\!]}_{\text{only if boolean } X}(t) = \mathcal{I}[\![X = 1]\!](t) = \mathcal{I}(X)(t) = X_{\mathcal{I}}(t)$

3.Terms: $\theta ::= x \mid \ell \mid \int P \mid f(\theta_1, \ldots, \theta_n)$ $\qquad \theta_1 + \theta_2 \widehat{=} +(\theta_1, \theta_2)$

Rigid Term: no l, nor $\int$ operators

3.1 Semantics $\mathcal{I}[\![\theta]\!] : \text{Val} \times \text{Intv} \to \mathbb{R}$

$\mathcal{I}[\![\theta]\!](\mathcal{V}, [b, e])$

4. Formulae: $F ::= p(\theta_1, \ldots, \theta_n) \mid \neg F_1 \mid F_1 \wedge F_2 \mid \forall x \bullet F_1 \mid F_1 ; F_2$

4.1 Semantics: $\mathcal{I}[\![F]\!] : \text{Val} \times \text{Intv} \to \{\text{tt}, \text{ff}\}$

5. Abbreviations:

$\lceil\rceil := l = 0$ $\qquad \lceil P \rceil := \int P = l \wedge l > 0$

$\lceil P \rceil^t := \lceil P \rceil \wedge l = t$ $\qquad \lceil P \rceil^{\leq t} := \lceil P \rceil \wedge l \leq t$

$\Diamond F := true; F; true$ $\qquad \Box F = \neg \Diamond \neg F$

6. Priority Groups: $\neg, \quad ;, \quad \wedge \vee, \quad \Longrightarrow \Longleftrightarrow, \quad \forall \exists$

7. Laws of the DC Integral operator: $\models ((\int P = r_1) ; (\int P = r_2)) \Rightarrow \int P = (r_1 + r_2)$

$\models \lceil \neg P \rceil \Rightarrow \int P = 0 \qquad \models \int P \leq l \qquad \models \lceil\rceil \Rightarrow \int P = 0$

3. Terms – Info

$\mathcal{V}(x) = 20 \quad \theta = x \cdot \int L \quad [b, e] = [0.5, 3.25]$

$\Rightarrow \mathcal{I}[\![\theta]\!](\mathcal{V}, [0.5, 3.25]) = \mathcal{I}[\![x \cdot \int L]\!](\mathcal{V}, [0.5, 3.25]) = \mathcal{I}[\![\hat{\cdot}(x, \int L)]\!](\mathcal{V}, [0.5, 3.25]) = \hat{\cdot}(\mathcal{I}[\![x]\!](\mathcal{V}, [0.5, 3.25]), \mathcal{I}[\![\int L]\!](\mathcal{V}, [0.5, 3.25])) = \hat{\cdot}(\mathcal{V}(x), \mathcal{I}[\![x]\!](\mathcal{V}, [0.5, 3.25])) = \hat{\cdot}\left(20, \int_{0.5}^{3.25} L_{\mathcal{I}}(t)\,\mathrm{d}t\right) = \hat{\cdot}(20, 1.25) = 20 \cdot 1.25 = 25$

Validity, Realizability, satisfyability

Holds: For **a certain** interpretation,evaluation, in a specific interval

$\mathcal{I}, \mathcal{V}, [b, e] \models F \Leftrightarrow \mathcal{I}[\![F]\!](\mathcal{V}, [b, e]) = tt$

Satisfies: For all interpretations and evaluations within a certain interval set

$\mathcal{I}, \mathcal{V} \models F \Leftrightarrow \forall [b, e] \in \text{Intervals} : \mathcal{I}, \mathcal{V}, [b, e] \models F$

Realises: For a certain Interpretation and for **all** intervals

$\mathcal{I} \models F \Leftrightarrow \forall \mathcal{V} \in \text{Valuations} : \mathcal{I}, \mathcal{V} \models F$

Valid:

$\models F \Leftrightarrow \forall \mathcal{I} : \mathcal{I} \models F$

Satisfiable $\Leftarrow$ Realisable $\Leftarrow$ Valid

Decidability Results for DC

	Discrete Time	Continuous Time
RDC	1. decidable	decidable
RDC $+ \ell = r$	decidable for $r \in \mathbb{N}$	undecidable for $r \in \mathbb{R}^+$
RDC $+ \int P_1 = \int P_2$	undecidable	undecidable
RDC $+ \ell = x, \forall x$	undecidable	2. undecidable
DC	undecidable	undecidable

	Discrete Time	Continuous Time
$\models^? (\lceil P \rceil; \lceil P \rceil) \Longrightarrow \lceil P \rceil$	Yes	Yes, smallest $e - b = 2$
$\models^? \lceil P \rceil \Longrightarrow (\lceil P \rceil; \lceil P \rceil)$	Yes	No, smallest $e - b = 1$

RDC in Discrete Time: $F ::= \lceil P \rceil \mid \neg F_1 \mid F_1 \vee F_2 \mid F_1 ; F_2,$

where P is a state assertion with boolean observables only

Idea: Give a procedure to construct a formula F,
a regular language $\mathcal{L}(F)$,
such that $\mathcal{I}[0, n] \models F$
iff $w \in \mathcal{L}(F)$, where w describes $\mathcal{I}$ on $[0, n]$.
Then F is satisfiable in discrete time iff $\mathcal{L}(F)$ is not empty
$\Rightarrow$ Emptyness problems are decidable for regular languages

$$\begin{aligned}\Sigma(F) = \{&(X \wedge Y \wedge Z), (X \wedge Y \wedge \neg Z), (X \wedge \neg Y \wedge Z),\\ &(X \wedge \neg Y \wedge \neg Z), (\neg X \wedge Y \wedge Z), (\neg X, \wedge Y \wedge \neg Z),\\ &(\neg X \wedge \neg Y \wedge Z), (\neg X \wedge \neg Y \wedge \neg Z)\}\end{aligned}$$

Disjunctive Normal Form (DNF):

$P = (X \wedge \neg Y) \Longleftrightarrow P = (X \wedge \neg Y \wedge Z) \vee (X \wedge \neg Y \wedge \neg Z)$

RDC $+ \ell = x, \forall x$ in Continuous Time:

$F ::= \lceil P \rceil \mid \neg F_1 \mid F_1 \vee F_2 \mid F_1 ; F_2 \mid \ell = 1 \mid \ell = x \mid \forall x \bullet F_1$

Idea: Reduce divergence of two counter-machines to realisability from 0.
$\mathcal{M} = (Q, q_0, q_{fin}, Prog)$
The (!) computation of $\mathcal{M}$ is a finite sequence of the form
$K_0 = (q_0, 0, 0) \vdash K_1 \vdash K_2 \ldots \vdash (q_{fin}, n_1, n_2)$
or an infinite sequence of the form
$K_0 = (q_0, 0, 0) \vdash K_1 \vdash K_2 \ldots$
A single configuration K of $\mathcal{M}$ can be encoded in an interval of length 4

Being an encoding interval can be characterised by a DC formula
Being an encoding of the run can be characterised by a DC Formula $\mathcal{F}(\mathcal{M})$
Then $\mathcal{M}$ diverges iff $\mathcal{F}(\mathcal{M}) \wedge \neg \Diamond \lceil q_{fin} \rceil$ is realisable from 0.

DC Standard Forms

Followed-By: $F \longrightarrow \lceil P \rceil \Longleftrightarrow \neg \Diamond (F; \lceil \neg P \rceil) \Longleftrightarrow \Box \neg (F; \lceil \neg P \rceil)$

$\forall x \bullet \Box ((F \wedge l = x) ; \ell > 0) \Longrightarrow ((F \wedge l = x) ; \lceil P \rceil ; true)$

(Timed) leads to: $F \xrightarrow{\theta} \lceil P \rceil :\Leftrightarrow (F \wedge \ell = \theta) \to \lceil P \rceil$

(Timed) up to: $F \xrightarrow{\leq\theta} \lceil P \rceil :\Leftrightarrow (F \wedge \ell \leq \theta) \to \lceil P \rceil$

Followed-by-initially: $F \underset{0}{\to} \lceil P \rceil :\Longleftrightarrow \neg (F; \lceil \neg P \rceil)$

(Timed) up-to-initally: $F \xrightarrow{\leq\theta}_0 \lceil P \rceil :\Longleftrightarrow (F \wedge l \leq \theta) \longrightarrow_0 \lceil P \rceil$

Initialisation: $\lceil\rceil \vee \lceil P \rceil ; true$

DC Implementables

1. **Initialisation:** $\lceil\rceil \vee \lceil \pi \rceil ; true$
   Initially, the control automaton is in phase π.
2. **Sequencing:** $\lceil \pi \rceil \longrightarrow \lceil \pi \vee \pi_1 \vee \ldots \vee \pi_n \rceil$
   When the control automaton is in π, it subsequently stays in π or moves to one of $\pi_1, \ldots, \pi_n$.
3. **Progress:** $\lceil \pi \rceil \xrightarrow{\theta} \lceil \neg \pi \rceil$
   After the control automaton stayed in phase π for θ time units, it subsequently leaves this phase, thus progress
4. **Synchronisation:** $\lceil \pi \wedge \varphi \rceil \xrightarrow{\theta} \lceil \neg \pi \rceil$
   After the control automation stayed for θ time units in phase π with the condition φ being true, it subsequently leaves this phase.
5. **Bounded stability:** $\lceil \neg \pi \rceil ; \lceil \pi \wedge \varphi \rceil \xrightarrow{\leq\theta} \lceil \pi \vee \pi_1 \vee \ldots \vee \pi_n \rceil$
   If the control automaton
6. **Unbound stability:** $\lceil \neg \pi \rceil ; \lceil \pi \wedge \varphi \rceil \longrightarrow \lceil \pi \vee \pi_1 \vee \ldots \vee \pi_n \rceil$
   If the control automaton
7. **Bounded inital stability:** $\lceil \pi \wedge \varphi \rceil \xrightarrow{\leq\theta}_0 \lceil \pi \vee \pi_1 \vee \ldots \vee \pi_n \rceil$
   If the control automaton
8. **Unbounded inital stability:** $\lceil \pi \wedge \varphi \rceil \longrightarrow_0 \lceil \pi \vee \pi_1 \vee \ldots \vee \pi_n \rceil$
   If the control automaton

Difference between ABC

A
B
C
D
E
F
HIER NOCH IRGENDWAS ZU DC IMPLEMENTABLES

Timed Automaton $\mathcal{A} = (L, B, X, I, E, \ell_{ini})$ Edges $E = (\ell, \alpha, \varphi, Y, \ell')$

Locations: L ={off, light, bright}
Alphabet: B = {press}
Clocks: X = {x}
Invariants: I = {off $\mapsto$ true, light $\mapsto$ true, bright $\mapsto$ true}
Edges: E = {(off, press?, true, {x}, light), (light, press?, $x > 0$, $\emptyset$, off),
(light, press?, x$\leq$3, $\emptyset$, bright), (bright, press?, true, $\emptyset$, off)}
Inital location: ℓ_{ini} = off

1. Clock Constraints: $\varphi ::= x \sim c \mid x - y \sim c \mid \varphi_1 \wedge \varphi_2$, $(\sim \in \{<, ><, \geq, \leq\})$

2. Clock Valuations: $\mathcal{V} : X \to$ Time, assigning each clock $x \in X$ the current time $\nu(x)$

2.1 Time Shift: $(\nu + t)(x) = \nu(x) + t$
$\mathcal{V} : \{x \mapsto 3.0\} \Rightarrow (\mathcal{V} + 0.27) = \nu(x) + 0.27 = 3.0 + 0.27 = 3.27$

2.2 Modification + Update: $(\mathcal{V}[Y := t]) = \begin{cases} t, & \text{if } x \in Y \\ \nu(x), & \text{otherwise} \end{cases}$ Reset $\Rightarrow t = 0$

3. Transitions:

3.1 Time or Delay Transition: $\langle \ell, \nu \rangle \xrightarrow{t} \langle \ell, \nu + t \rangle$ iff $\forall t \in$ Time : $\nu + t \models I(\ell)$
Some Time elapses respecting invariants, locations unchanged

3.2 Action or Discrete Transition: $\langle \ell, \nu \rangle \xrightarrow{\alpha} \langle \ell', \nu' \rangle$iff $\nu \models \varphi, \nu' = \nu[Y := 0], \nu' \models I(\ell')$
An Action occurs, location, clocks may change / reset, time does not elapse

3.3 Transition Sequences: $\langle \ell_0, \nu_0 \rangle \xrightarrow{\lambda_1} \langle \ell_1, \nu_1 \rangle \xrightarrow{\lambda_2} \ldots$ with $\langle \ell_0, \nu_0 \rangle \in C_{ini}$
$\lambda \in B \vee \lambda \in$ Time [-0.1cm] (finite or infinite)

4. Reachability: A configuration $\langle \ell, \nu \rangle$ is called reachable (in $\mathcal{A}$),
iff there is a transition sequence of the form:
$\langle \ell_0, \nu_0 \rangle \xrightarrow{\lambda_1} \langle \ell_1, \nu_1 \rangle \xrightarrow{\lambda_2} \langle \ell_2, \nu_2 \rangle \xrightarrow{\lambda_3} \ldots \xrightarrow{\lambda_n} \langle \ell_n, \nu_n \rangle = \langle \ell, \nu \rangle$

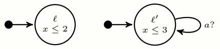
5. Time Stamped Configurations: $\langle \ell, \nu \rangle, t$ is a timed-stamped configuration

5.1 Time Stamped Delay Transition:
$\langle \ell, \nu \rangle, t \xrightarrow{t'} \langle \ell, \nu + t' \rangle, t + t'$ iff $t' \in$ Time and $\langle \ell, \nu \rangle, t \xrightarrow{t'} \langle \ell, \nu + t' \rangle$

5.2 Time Stamped Action Transition:
$\langle \ell, \nu \rangle, t \xrightarrow{\alpha} \langle \ell', \nu' \rangle, t$ iff $\alpha \in B_{?!}$ and $\langle \ell, \nu \rangle, t \xrightarrow{\alpha} \langle \ell', \nu' \rangle$

6. Computation Path: (=Sequence of time-stamped configurations (infinite/max finite))
$\xi = \langle \ell_0, \nu_0 \rangle, t_0 \xrightarrow{\lambda_1} \langle \ell_1, \nu_1 \rangle, t_1 \xrightarrow{\lambda_2} \ldots$ starting in $\langle \ell_0, \nu_0 \rangle$, with $\langle \ell_0, \nu_0 \rangle \in C_{ini}$

7. Time Locks and Zeno-Behaviour:



$\langle \ell, x = 0 \rangle, 0 \xrightarrow{2} \langle \ell, x = 2 \rangle, 2$
$\langle \ell, x = 0 \rangle, 0 \xrightarrow{0.1} \langle \ell, x = 0.1 \rangle, 0.1 \xrightarrow{0.01} \langle \ell, x = 0.11 \rangle, 0.11 \xrightarrow{0.001} \ldots$
A Configuration $\langle \ell, \nu \rangle$ is called timelock, iff no delay transition with $t > 0$ from $\langle \ell, \nu \rangle$ is possible (or any location change possible due to Invariant of location + guard of edge/transition)
$\langle \ell, x = 0 \rangle, 0 \xrightarrow{\frac{1}{2}} \langle \ell, x = \frac{1}{2} \rangle, \frac{1}{2} \xrightarrow{\frac{1}{4}} \ldots \xrightarrow{\frac{1}{2^n}} \langle \ell, x = \frac{2^n - 1}{2^n} \rangle, \frac{2^n - 1}{2^n}$

8. Real-Time Sequence: $t_0, t_1, t_2, \ldots, t_i \in$ Time for $i \in \mathbb{N}_0$
is called Real-Time Sequence iff it has the following properties:
8.1 Monotonicity: $\forall i \in \mathbb{N}_0 : t_i \leq t_{i+1}$
8.2 Non-Zeno Behaviour: $\forall t \in$ Time, $\exists i \in \mathbb{N}_0 : t < t_i$

9. Run: Starting in $\langle \ell, \nu \rangle, t_0$, a run is an infinite computation path, where ${t_i}_{i \in \mathbb{N}_0}$ is a Real-Time Sequence. ($\xi =$ run of $\mathcal{A}$ iff $\xi =$ computation path of $\mathcal{A}$)

Networks of Timed Automata

1. Handshake Edges: $(\ell_1, \alpha, \varphi_1, Y_1, \ell_1') \in E_1$ $(\ell_2, \alpha, \varphi_2, Y_2, \ell_2') \in E_2$

2. Asynchronous Edges: $((\ell_1, \ell_2), \tau, \varphi_1 \wedge \varphi_2, Y_1 \cup Y_2, (\ell_1', \ell_2')) \in E$
If $(\ell_1, \alpha, \varphi_1, Y_1, \ell_1') \in E_1$ then $\forall . \ell_2 \in E_2 \bullet ((\ell_1, \ell_2), \alpha, \varphi, Y_1, (\ell_1', \ell_2)) \in E$
If $(\ell_2, \alpha, \varphi_2, Y_2, \ell_2') \in E_2$ then $\forall . \ell_1 \in E_1 \bullet ((\ell_1, \ell_2), \alpha, \varphi, Y_2, (\ell_1, \ell_2')) \in E$

3. Channel Hiding: **(Introduces local channels)**
$(\ell, \alpha, \varphi, Y, \ell') \in E'$ iff $(\ell, \alpha, \varphi, Y, \ell') \in E \ \wedge \ \alpha \notin \{press!, press?\}$

4. Closed Networks: Hiding all channel transitions of given channel
$\Rightarrow$ Transitions are thus either internal actions τ or delay transitions (Hiding all channel yields a closed network)

5. Operational Semantics of Networks:

5.1 Local Transitions:
$\langle \vec{\ell}, \nu \rangle \xrightarrow{\alpha} \langle \vec{\ell}', \nu \rangle$
if there is an $i \in \{1, \ldots, n\}$ such that
$(\ell_i, \alpha, \varphi, Y, \ell_i) \in E$, $\alpha \in B_{?!}$
$\nu \models \varphi$
$\vec{\ell}_i' = \vec{\ell}[\ell_i := \ell_i']$
$\nu' = \nu[Y := 0]$
$\nu' \models I(\ell_i')$

5.3 Delay Transitions: $\langle \vec{\ell}, \nu \rangle \xrightarrow{t} \langle \vec{\ell}, \nu + t \rangle$
if $\forall t \in [0, t]$ $\nu + t' \models \bigwedge_{k=1}^{n} I_k(\ell_k)$

5.2 Synchronisation Transition:
$\langle \vec{\ell}, \nu \rangle \xrightarrow{\tau} \langle \vec{\ell}', \nu \rangle$
if there are $i, j \in \{1, \ldots, n\}, i \neq j$ and $b \in B_i \cup B_j$, such that
$(\ell_i, b!, \varphi_i, Y_i, \ell_i') \in E_i$
$(\ell_j, b?, \varphi_j, Y, \ell_j) \in E_j$
$\nu \models \varphi_i \wedge \varphi_j$
$\vec{\ell}_i' = \vec{\ell}[\ell_i := \ell_i'][\ell_j := \ell_j']$
$\nu' = \nu[Y_i \cup Y_j := 0]$
$\nu' \models I(\ell_i') \wedge I(\ell_j')$

Location Reachability / The Region Automata

Given: A timed automata $\mathcal{A}$ and one of its location ℓ
Question: Is ℓ reachable?
That is, if there is a transition sequence of the form
$\langle \ell_{ini}, \nu_0 \rangle \xrightarrow{\lambda_1} \langle \ell_1, \nu_1 \rangle \xrightarrow{\lambda_2} \langle \ell_2, \nu_2 \rangle \xrightarrow{\lambda_3} \ldots \xrightarrow{\lambda_n} \langle \ell_n, \nu_n \rangle$
with $\ell_n = \ell$ is the labelled transition system.

Note: Decidability is not soo obvious:
Clocks range over real numbers, thus infinitely many configurations
At each configurations uncountably many transitions $\xrightarrow{t}$ may originate

Recall: $\varphi := x \sim c \mid x - y \sim c \mid \varphi \wedge \varphi$ $x, y \in X$, $c \in \mathbb{Q}_0^+$, and $\sim \in \{<, >, \leq, \geq\}$

1. Observe clock constraints: Let $t_{\mathcal{A}}$ be the least common multiple of the denominators in $C(\mathcal{A})$. A location is reachable in $t_{\mathcal{A}} \cdot \mathcal{A}$ iff ℓ is reachable in $\mathcal{A}$

2. Time abstract Transition System $\mathcal{U}(\mathcal{A})$:
Let $\langle \ell, \nu \rangle, \langle \ell', \nu' \rangle \in$ Conf($\mathcal{A}$) be configurations of $\mathcal{A}$ and $\alpha \in B_{?!}$, an action,
then $\langle \ell, \nu \rangle \overset{\alpha}{\Rightarrow} \langle \ell', \nu' \rangle$ iff there exists $t \in$ Time, such that $\langle \ell, \nu \rangle \xrightarrow{t} \circ \xrightarrow{\alpha} \langle \ell', \nu' \rangle$

3. Region Automaton $\mathcal{R}(\mathcal{A})$:
Distinguish Clock valuations: if $c_x \geq 1$ then there are $(2c_x + 2)$ equivalence classes: $\{\{0\}, (0, 1), \{1\}, (1, 2), \{2\}, \ldots, \{c_x\}, (c_x, \infty)\}$.
If $\nu_1(x)$ and $\nu_2(x)$ are in the same equivalence class, then ν_1 and ν_2 are indistinguishable by $\mathcal{A}$.
$\mathcal{R}(\mathcal{A}) = \Big(\text{Conf}\,(\mathcal{R}(\mathcal{A}))(\,, B_{?!}, \Big\{\overset{\alpha}{\Rightarrow}_{\mathcal{R}(\mathcal{A})} \mid \alpha \in B_{?!}\Big\}, C_{ini}\Big)$
where Conf($\mathcal{R}(\mathcal{A})$) = $\{\langle \ell, [\nu] \rangle, \ell \in L, \nu : X \to \text{Time}, \nu \models I(\ell)\}$
$\alpha \in B_{?!} : \langle \ell, [\nu] \rangle \overset{\alpha}{\Rightarrow}_{\mathcal{R}(\mathcal{A})} \langle \ell', [\nu'] \rangle$iff $\langle \ell, \nu \rangle \overset{\alpha}{\Rightarrow} \langle \ell', \nu' \rangle \Big(\widehat{=} \langle \ell, \nu \rangle \overset{t}{\Rightarrow} \circ \overset{\alpha}{\Rightarrow} \langle \ell', \nu' \rangle\Big)$

4. The number of Regions: $2\,(c + 2)^{|X|} \cdot (4c + 3)^{\frac{1}{2}|X| \cdot (|X| - 1)}$ = upper bound number

Putting all together $\mathcal{A} = (L, B, X, I, E, \ell_{ini})$
There are finitely many locations in L (By definition)
There are finitely many regions
$\Rightarrow$ So Conf($\mathcal{R}(\mathcal{A})$) is finite (By construction)
It is decideable whether there exists a sequence
$\langle \ell_{ini}, [\nu_{ini}] \rangle \overset{\alpha}{\Rightarrow}_{\mathcal{R}(\mathcal{A})} \langle \ell_1, [\nu_1] \rangle \overset{\alpha}{\Rightarrow}_{\mathcal{R}(\mathcal{A})} \cdots \overset{\alpha}{\Rightarrow}_{\mathcal{R}(\mathcal{A})} \langle \ell_n, [\nu_n] \rangle$,
such that $\langle \ell_n, [\nu_n] \rangle = \langle \ell, [\nu] \rangle$

Note: We just observed that $\mathcal{R}(\mathcal{A})$ loses some information about the clock valuations that are possible in a region.

Region and Zones

A (clock) zone is a set $z \subseteq (X \to \text{Time})$ of valuations of clocks X such that there exists $\varphi \in \Phi(X)$ with $\nu \in z$ iff $\nu \models \varphi$.

1. **Let time elapse**
2. **Intersect with Invariant of ℓ**
3. **Intersect with guard**
4. **Reset clocks**
5. **Intersect with Invariant of ℓ'**

Pro's and Con's:
Zone-based: + avoids blowup by number of clocks and size of clock constraints through symbolic representation of clocks – confinde wrt. size of discrete state space
Region-based: + Less dependent on size of discrete state space – exponential in number of clocks

Extended Timed Automata $\mathcal{A} = (L, C, B, U, X, V, I, E, \ell_{ini})$ Edges $E = (\ell, \alpha, \varphi, \vec{r}, \ell')$

Comitted Locations: $C \subseteq L$
Urgend Channels: $U \subseteq B$
Set of Datavariables: $\mathcal{V}$
Urgend: Being in this location blocks time passing
Committed: Being in this location the next edge must change the committed location

1. Data Variables: When modelling controllers as timed automata, it is sometimes desirable to have (local and shared) non-clock variables. E.g, number of open doors

2. Urgent Locations: Enforce local immediate progress (In $t = 0$ time)

3. Committed Locations: Enforce atomic immediate progress (Direct) As long as data variables are finite the extension doesn't harm the decidability

Automatic Verification of DC Properties for Timed Automata

Testability

Timed Büchi Automata $\mathcal{A} = (\Sigma, S, S_0, X, E, F)$

Alphabet: Σ
Finite Set of States: S
Set of Start States: $S_0 \leq S$
Finite Set of Clocks: X
Set of Transitions: $E = (s, s', a, \lambda, \delta)$
Set of Acceptig States: F

1. Time sequence: $\tau = \tau_1, \tau_2, \ldots$ is an infinite sequence of time values $\tau_i \in \mathbb{R}_0^+$, satisfying the following constraints:
Monotonicity: τ increases strictly monotonically, i.e., $\tau_i < \tau_{i+1} \forall i \geq 1$
Progress: For every $t \in \mathbb{R}_0^+$, there is some $i \geq 1$ such that $\tau_i > t$

2. Timed Word: is a pair (σ, τ) over a an alphabet Σ, where $\sigma = \sigma_1, \sigma_2, \ldots \in \Sigma^\omega$ is an infinite word over Σ and τ is a time sequence

3. Timed Language: is a set of timed words over Σ over an alphabet Σ

4. Clock constraints: $\delta ::= x \leq c \mid c \leq x \mid \neg\delta \mid \delta_1 \wedge \delta_2$

5. (Accepting) TBA Run: A run r, denoted by $(\bar{s}, \bar{v})$ of a TBA over a timed word (σ, τ) is an infinite sequence $r : \langle s_o, \nu_0 \rangle \xrightarrow[\tau_1]{\sigma_1} \langle s_1, \nu_1 \rangle \xrightarrow[\tau_2]{\sigma_2} \ldots$ and is called (an) accepting (run) iff $\inf(r) \cap F \neq 0$ (Der Final-State wird ∞-oft besucht $\to \{s_2, s_3\} \cup s_2 \neq 0$)

6. Language of TBA For a TBA $\mathcal{A}$, the language $\mathcal{L}(\mathcal{A}$ of time wrds it accepts defined to be the set $\{(\sigma, \tau) | \mathcal{A}$ has an accepting run over $(\sigma, \tau)\}$. For short: $\mathcal{L}(\mathcal{A})$ is the language of $\mathcal{A}$.