The Influence of Durational Actions on Time Equivalences

Harald Fecher*

Christian-Albrechts-University Kiel, Germany
hf@informatik.uni-kiel.de

Abstract. The hierarchy of untimed equivalences is well understood for action-based systems. This is not the case for timed systems, where it is, for example, possible to detect concurrency by single timed action execution. To clarify the connection between equivalences in timed systems, a timed version of configuration structures is introduced together with timed equivalence notions adapted from untimed equivalences. There actions (events) have an occurrence time and a duration. The result of this paper is that all timed versions of the equivalences from [15] have the same relative discriminating power as in the untimed case, except that interleaving and step (for trace and bisimulation) equivalences coincide if systems are considered where every action must have a positive duration.

1 Introduction

Action-based formalisms are used to model systems at an abstract level. But nearly all those formalisms are too concrete in the sense that the same system can be described in many different ways. Therefore, equivalences are introduced to identify those descriptions that have the same observation. Since there are many notions of observations (e.g., linear/branching, interleaving/true concurrent, ...), different equivalences are investigated. In order to relate the level of abstraction of the equivalences, their discriminating power is examined. This is done, for example, for untimed equivalences in [15, 11, 13, 12].

Timed systems have much more observation possibilities, since observations concerning the occurrence time and the duration of actions are possible. This allows, for example, to detect some concurrency by action traces, in the case when ill-timed traces, i.e., traces where the occurrence time of actions may decrease inside a trace, are allowed [2, 1].

The observation that a certain degree of concurrency can be detected in timed systems leads to the question whether the inclusion relations of untimed equivalences are different for the timed versions, in particular, do some timed equivalence notions coincide? The advantage of the collapsing of equivalences is that the verification techniques, axiom systems and results existing for one of them can be used for the other. Furthermore, it can reduce the state spaces that have to be considered to verify the equivalences of processes; for example, when

* The work started when the author worked at the University of Mannheim, Germany.

single action execution (interleaving) bisimilarity coincides with step bisimilarity, it is enough to consider only the transition system consisting of single action executions and not also the step executions in order to verify that processes are step bisimilar.

Another question is whether the hierarchy of the timed equivalences are different when the considered system is restricted. For example do some equivalences coincide when every action has a positive duration and/or when no action may idle (i.e., an action can only be executed at the time when it becomes enabled). Systems where no action may idle, are called *urgent* systems in this paper.

The aim of this paper is to present answers to these questions. In order to have the possibility to compare equivalences, we have to define them on a common abstract domain. Configuration structures were used for this purpose in the untimed case [15]. We introduce a timed version of them, called durational configuration structures, where an event has also an occurrence time and a duration. These structures have to satisfy some well formed condition relating the causality and the occurrence time of events (an event that causes another event e has to occur earlier than e). We introduce timed equivalences, which are extensions of untimed equivalences, in durational configuration structures. These equivalences are compared with respect to their level of abstraction. We point out that the hierarchy of all considered equivalences are the same in the untimed and the timed case, except that interleaving and step (for trace and bisimulation) equivalences coincide if every event (action) must have a positive duration. In particular, the restriction to urgent systems does not have any influence on the equivalence hierarchy.

The structure of the paper is the following: In Section 2 durational configuration structures are investigated. The timed equivalences are defined on them in Section 3. In Section 4, we introduce a simple timed process algebra in order to obtain a more readable notation for our counterexamples. The equivalences are examined with respect to their discriminating power in Section 5. The paper is concluded in Section 6.

2 Durational Configuration Structures

Configuration structures [14, 16, 15] are a more general event-oriented model of untimed concurrent systems, which is more general than, e.g., event structures [28]. They usually consist of a family of finite sets (the configurations) and a labelling function which maps the elements that occur in a configuration to a set of action names. A configuration can be considered as a collection of events that can appear at some stage of the system run. The labelling function indicates which actions are observable when the events happen. Sometimes configuration structures also contain a termination predicate (a subset of configurations) [16, 15], which indicates terminated configurations. Since termination is not relevant for our approach, we do not consider termination predicates. It is straightforward to transform our result to termination sensitive equivalences.

In timed systems, it is possible to observe at which time the events (actions) happen and also how long their duration is, see for example [23, 9, 8]. We make an atomicity assumption, in the sense that the termination time of an event (action) is known as soon as this event starts to execute, i.e., the duration (and also non-termination) of events is known a priory[1]. The occurrence and the duration time information can not be encoded in a configuration structure by an additional global function, as the labelling function, since the same event may happen at different times (and with different durations) in different system runs. Therefore, we encode configurations as partial functions from the set of events to the set of their possible occurrence times together with their possible duration. In our case, these will be partial functions into $\mathbb{R}_0^{+\infty} = \mathbb{R}_0^+ \times (\mathbb{R}_0^+ \cup \{\infty\})$, where the duration ∞ denotes that the event will never terminate.

In the following, $f : M_1 \rightharpoonup M_2$ denotes that f is a *partial function* from M_1 to M_2. Partial functions are sometimes considered as sets, hence the partial function that is everywhere undefined is given by $\emptyset$. The *domain* of f, denoted by $\mathrm{dom}(f)$, is the set $\{m \in M_1 \mid f(m) \text{ is defined}\}$. The restriction of f to set $M_1' \subseteq M_1$ is $f \restriction M_1 = f \cap (M_1' \times M_2)$. Partial functions from M_1 to M_2 are ordered pointwise, denoted by $\sqsubseteq$, i.e., $f \sqsubseteq g$ iff $f \subseteq g$.

Definition 1. *A* labelled durational configuration structure over an alphabet $\mathcal{A}ct$ with respect to set E *is a tuple* $\mathcal{C} = (C, l)$ *such that*

$$C \subseteq \{f : E \rightharpoonup \mathbb{R}_0^+ \times \mathbb{R}_0^{+\infty} \mid |\mathrm{dom}(f)| < \infty\}$$
$$l : E \to \mathcal{A}ct.$$

An element f of C is called a durational configuration. Furthermore, π_t denotes the projection to the occurrence time and π_d denotes the projection to the duration, i.e., if $f(e) = (t, d)$ then $\pi_t(f(e)) = t$ (the absolute time when event e happens in f) and $\pi_d(f(e)) = d$ (the duration of event e in f). Hereafter, $\mathcal{C}$ is considered to be (C, l), E is assumed to be a fixed set of event names, and $\mathcal{A}ct$ be a fixed set of action names. Furthermore, if $M \subseteq E$ then $\overline{M}$ denotes the *complement* of M, i.e., $\overline{M} = \{e \in E \mid e \notin M\}$.

Remark 2. The untimed configuration structures of [15] can be modelled as durational configuration structures, where all events occur at time 0 and all events have duration 0, i.e., by assigning an untimed configuration structure $X \subseteq E$ to the durational configuration structure f_X, where $\mathrm{dom}(f_X) = X$ and $\forall e \in X : f_X(e) = (0, 0)$.

In the untimed case, a subset of configuration structures is characterized by the property that the causal dependencies in configurations can faithfully be represented by means of partial orders. It turned out that these are exactly the configuration structures associated with stable event structures [15]. In the following definition, we define the analogous property for durational configuration structures.

[1] This atomicity assumption has the consequence that the ST-equivalence [17] does not fit into our setting.

Definition 3. *A durational configuration structure $\mathcal{C}$ is*

- rooted *iff* $\emptyset \in C$,
- connected *iff* $\forall g \in C : g \neq \emptyset \Rightarrow \exists e \in \mathrm{dom}(g) : g \restriction \overline{\{e\}} \in C$,
- closed under bounded unions *iff* $\forall f, g, h \in C : (f \sqsubseteq h \wedge g \sqsubseteq h) \Rightarrow f \cup g \in C$,
- closed under bounded intersections *iff* $\forall f, g, h \in C : (f \sqsubseteq h \wedge g \sqsubseteq h) \Rightarrow f \cap g \in C$.

$\mathcal{C}$ is stable *iff it is rooted, connected, closed under bounded unions, and closed under bounded intersections.*

The causality relation on events describes which event occurrences are necessary for other event occurrences and the concurrency relation describes which event occurrences are independent from each other. These relations are defined with respect to configurations, i.e., the event dependencies may vary in different configurations (system runs). The relations are adapted to durational configuration structures as follows:

Definition 4. *Let $\mathcal{C}$ be a durational configuration and let $g \in C$.*
The causality relation $<_g^{\mathcal{C}} \subseteq \mathrm{dom}(g) \times \mathrm{dom}(g)$ *on g is given by $e' <_g^{\mathcal{C}} e$ iff $e' \leq_g^{\mathcal{C}} e$ and $e' \neq e$, where $e' \leq_g^{\mathcal{C}} e \iff \big(\forall f \in C : (f \sqsubseteq g \wedge e \in \mathrm{dom}(f)) \Rightarrow e' \in \mathrm{dom}(f)\big)$.*
The concurrency relation $\|_g^{\mathcal{C}} \subseteq \mathrm{dom}(g) \times \mathrm{dom}(g)$ *on g is given by $e' \|_g^{\mathcal{C}} e$ iff $\neg(e' <_g^{\mathcal{C}} e \vee e <_g^{\mathcal{C}} e')$.*

We write $<_g$ ($\|_g$) rather than $<_g^{\mathcal{C}}$ (respectively, $\|_g^{\mathcal{C}}$) if $\mathcal{C}$ is clear from the context. If an event e' is a causality of e, i.e., $e' <_g e$, then e' has to occur earlier than e. More precisely, e has to occur later than the termination of e'. This is formalized in the following definition, where also further time properties of configuration structures are specified.

Definition 5. *A durational configuration structure $\mathcal{C}$ is time stable iff $\mathcal{C}$ is stable and $\forall g \in C : e' <_g e \Rightarrow \pi_t(g(e')) + \pi_d(g(e')) \leq \pi_t(g(e))$. Let $\mathbb{C}$ be the set of all time stable durational configuration structures.*

- $\mathcal{C} \in \mathbb{C}$ has only durational action *iff* $\forall g \in C, e \in \mathrm{dom}(g) : \pi_d(g(e)) > 0$.
- $\mathcal{C} \in \mathbb{C}$ *is* urgent *iff* $\forall g \in C, e \in \mathrm{dom}(g) : \pi_t(g(e)) > 0 \Rightarrow \exists e' : e' <_g e \wedge \pi_t(g(e')) + \pi_d(g(e')) = \pi_t(g(e))$.

Let $\mathbb{C}^d$ be $\mathbb{C}$ restricted to those configuration structures that have only durational action and let $\mathbb{C}^u$ be $\mathbb{C}$ restricted to those configuration structures that are urgent. Finally, $\mathbb{C}^{du}$ denotes the set of all configuration structures that have only durational action and are urgent, i.e., $\mathbb{C}^{du} = \mathbb{C}^d \cap \mathbb{C}^u$.

If a durational configuration structure is urgent, then time must not pass between actions[2], i.e., time may only pass by action execution. Urgent systems are, for example, most generative systems, where the environment has no influence on the action execution. Durational configuration structures that have only

[2] In some approaches, e.g., [27, 21], urgency is only enforced on internal actions.

durational actions are generated by most physical processes, since an action has to consume time in real systems. In these systems actions are sometimes abstracted and are considered durationless. But such kind of abstractions are not always reasonable. Hence, we just motivated that the restriction to durational configuration structures that are urgent and/or have only durational actions is reasonable for applications.

Example 6. The durational configuration structure

$$\tilde{\mathcal{C}} = \big(\{\{(e_0,0,1)\},\{(e_1,0,2)\},\{(e_0,0,1),(e_1,0,2)\},\{(e_1,0,2),(e_2,2,1)\},\\ \{(e_0,0,1),(e_1,0,2),(e_2,2,1)\}\}\ ,\ \{(e_0,a),(e_1,b),(e_2,c)\}\big)$$

is time stable, has only durational action, and is urgent.

3 Timed Equivalences

In this section, we adapt equivalence notions of the untimed case to the timed case, where we concentrate on truly concurrent equivalences (i.e., those equivalences presented in [15]), since we rather expect a change in the hierarchical structure of truly concurrent equivalences. For simplicity, we only consider strong equivalences, i.e., we do not abstract from internal actions.

In order to define the equivalences, we introduce different kinds of action (not event) executions with respect to time. One concerns single action execution. Another one concerns simultaneous action execution (multisets over $\mathcal{A}ct$). A third one concerns the execution of sets of ordered actions, which are called pomsets. In all these executions, the occurrence time and the duration of the executed event is taken into account.

Definition 7. *Define $\mathcal{L} = \mathcal{A}ct \times \mathbb{R}_0^+ \times \mathbb{R}_0^{+\infty}$. The isomorphism class of partially ordered sets labelled by $\mathcal{L}$ are called* pomset*s over $\mathcal{L}$. The isomorphism class of a partially ordered set u is denoted by $[u]$. Let $\mathcal{P}$ be the set of all pomsets over $\mathcal{L}$ where the underlying sets have to be subsets of E.*

Suppose $l : E \rightharpoonup \mathcal{A}ct$ and $g : E \rightharpoonup \mathbb{R}_0^+ \times \mathbb{R}_0^{+\infty}$ such that $\mathrm{dom}(l) = \mathrm{dom}(g)$, then $l \times g : E \rightharpoonup \mathcal{L}$ is defined by $\mathrm{dom}(l \times g) = \mathrm{dom}(l)$ and $\forall e \in \mathrm{dom}(l) : (l \times g)(e) = (l(e), g(e))$.

Definition 8. *Suppose $\mathcal{C} \in \mathbb{C}$.*
Define the action execution relation $\longrightarrow_{\mathcal{C}} \subseteq C \times \mathcal{L} \times C$, where $f \xrightarrow[d]{(a,t)}_{\mathcal{C}} g$ abbreviates $(f,(a,t,d),g) \in \longrightarrow_{\mathcal{C}}$, by

$$f \xrightarrow[d]{(a,t)}_{\mathcal{C}} g \text{ iff } \exists e : e \notin \mathrm{dom}(f) \wedge g = f \cup \{(e,t,d)\} \wedge l(e) = a.$$

Define the step execution relation $\longmapsto_{\mathcal{C}} \subseteq C \times (\mathcal{L} \to \mathbb{N}) \times C$ by

$$f \overset{\gamma}{\longmapsto}_{\mathcal{C}} g \text{ iff } f \sqsubseteq g \wedge \exists H : H = \mathrm{dom}(g) \setminus \mathrm{dom}(f) \wedge (\forall e,e' \in H : e \|_g e') \wedge \\ \gamma(a,t,d) = |\{e \in H \mid l(e) = a \wedge g(e) = (t,d)\}|.$$

Define the pomset execution relation $\hookrightarrow_{\mathcal{C}} \subseteq C \times \mathcal{P} \times C$ *by*

$$f \overset{u}{\hookrightarrow}_{\mathcal{C}} g \text{ iff } f \sqsubseteq g \wedge \exists H : H = \mathrm{dom}(g) \setminus \mathrm{dom}(f) \wedge \\ u = [H, <_g \cap (H \times H), l \restriction H \times g \restriction H].$$

Example 9. Consider the durational configuration structures $\tilde{\mathcal{C}}$ from Example 6. Then

$$\begin{array}{ccc} & \{(e_1, 0, 2)\} & \\ \overset{(b,0)}{\underset{2}{\nearrow}}{}_{\tilde{\mathcal{C}}} & & \overset{\tilde{\gamma}}{\searrow}{}_{\tilde{\mathcal{C}}} \\ \emptyset & \overset{\tilde{u}}{\hookrightarrow}_{\tilde{\mathcal{C}}} & \{(e_0, 0, 1), (e_1, 0, 2), (e_2, 2, 1)\} \end{array}$$

where $\tilde{\gamma} = \{((a, 0, 1), 1), ((c, 2, 1), 1)\}$
and $\tilde{u} = [\{e_0, e_1, e_2\}, \{(e_1, e_2)\}, \{(e_0, a), (e_1, b), (e_2, c)\}]$.

Many equivalences are based on the trace or the bisimulation technique:

Definition 10. *Let* $\mathcal{C}, \mathcal{C}' \in \mathbb{C}$ *and* $\longrightarrow \subseteq C \times M \times C$ *and* $\longrightarrow' \subseteq C' \times M \times C'$ *for some set of labels* M. *An element* $(m_0, ..., m_{n-1}) \in M^*$ *is a* trace *of* $\mathcal{C}$ *with respect to* $\longrightarrow$ *if there exists* $f_0, ..., f_n \in C$ *with* $\emptyset = f_0$ *and* $\forall i < n : f_i \overset{m_i}{\longrightarrow} f_{i+1}$. *The set of all traces of* $\mathcal{C}$ *with respect to* $\longrightarrow$ *is denoted by* $\mathcal{T}_{(\mathcal{C}, \longrightarrow)}$.
$(\mathcal{C}, \longrightarrow)$ *and* $(\mathcal{C}', \longrightarrow')$ *are* trace equivalent *iff* $\mathcal{T}_{(\mathcal{C}, \longrightarrow)} = \mathcal{T}_{(\mathcal{C}', \longrightarrow')}$.

Definition 11. *Let* $\mathcal{C}, \mathcal{C}' \in \mathbb{C}$ *and* $\longrightarrow \subseteq C \times M \times C$ *and* $\longrightarrow' \subseteq C' \times M \times C'$ *for some set of labels* M. *A relation* $R \subseteq C \times C'$ *is called a* bisimulation *between* $(\mathcal{C}, \longrightarrow)$ *and* $(\mathcal{C}', \longrightarrow')$ *iff* $(\emptyset, \emptyset) \in R$ *and if* $(f, f') \in R$ *then*

- $f \overset{m}{\longrightarrow} g \Rightarrow \exists g' : f' \overset{m}{\longrightarrow} g' \wedge (g, g') \in R$,
- $f' \overset{m}{\longrightarrow} g' \Rightarrow \exists g : f \overset{m}{\longrightarrow} g \wedge (g, g') \in R$.

$(\mathcal{C}, \longrightarrow)$ *and* $(\mathcal{C}', \longrightarrow')$ *are* bisimilar *iff there is a bisimulation between* $(\mathcal{C}, \longrightarrow)$ *and* $(\mathcal{C}', \longrightarrow')$.

In the following, the *interleaving trace equivalence* [20], *interleaving bisimilarity* [22], *step trace equivalence* [24], *step bisimilarity* [24], *pomset trace equivalence* [7], *pomset bisimilarity* [7], *weak history-preserving equivalence* [10], *history-preserving equivalence* [25] and *hereditary history-preserving equivalence* [3] are adapted to time, i.e., defined on durational configuration structures.

Definition 12 (Interleaving trace). *The durational configuration structures* $\mathcal{C}$ *and* $\mathcal{C}'$ *are* interleaving trace equivalent, *denoted by* $\mathcal{C} \approx_{it} \mathcal{C}'$, *iff* $(\mathcal{C}, \longrightarrow_{\mathcal{C}})$ *and* $(\mathcal{C}', \longrightarrow_{\mathcal{C}'})$ *are trace equivalent.*

Definition 13 (Interleaving bisimilarity). *The durational configuration structures* $\mathcal{C}$ *and* $\mathcal{C}'$ *are* interleaving bisimilar, *denoted by* $\mathcal{C} \approx_{ib} \mathcal{C}'$, *iff* $(\mathcal{C}, \longrightarrow_{\mathcal{C}})$ *and* $(\mathcal{C}', \longrightarrow_{\mathcal{C}'})$ *are bisimilar.*

Definition 14 (Step trace). *The durational configuration structures* $\mathcal{C}$ *and* $\mathcal{C}'$ *are* step trace equivalent*, denoted by* $\mathcal{C} \approx_{st} \mathcal{C}'$*, iff* $(\mathcal{C}, \longmapsto_{\mathcal{C}})$ *and* $(\mathcal{C}', \longmapsto_{\mathcal{C}'})$ *are trace equivalent.*

Definition 15 (Step bisimilarity). *The durational configuration structures* $\mathcal{C}$ *and* $\mathcal{C}'$ *are* step bisimilar*, denoted by* $\mathcal{C} \approx_{sb} \mathcal{C}'$*, iff* $(\mathcal{C}, \longmapsto_{\mathcal{C}})$ *and* $(\mathcal{C}', \longmapsto_{\mathcal{C}'})$ *are bisimilar.*

Definition 16 (Pomset trace). *The durational configuration structures* $\mathcal{C}$ *and* $\mathcal{C}'$ *are* pomset trace equivalent*, denoted by* $\mathcal{C} \approx_{pt} \mathcal{C}'$*, iff* $(\mathcal{C}, \hookrightarrow_{\mathcal{C}})$ *and* $(\mathcal{C}', \hookrightarrow_{\mathcal{C}'})$ *are trace equivalent.*

It is easily seen that the definition of pomset trace equivalence can be restricted to single element traces, i.e.,

$\mathcal{C} \approx_{pt} \mathcal{C}'$ iff $\forall u \in \mathcal{P} : (\exists g : \emptyset \overset{u}{\hookrightarrow}_{\mathcal{C}} g) \Leftrightarrow (\exists g' : \emptyset \overset{u}{\hookrightarrow}_{\mathcal{C}'} g')$.

Definition 17 (Pomset bisimilarity). *The durational configuration structures* $\mathcal{C}$ *and* $\mathcal{C}'$ *are* pomset bisimilar*, denoted by* $\mathcal{C} \approx_{pb} \mathcal{C}'$*, iff* $(\mathcal{C}, \hookrightarrow_{\mathcal{C}})$ *and* $(\mathcal{C}', \hookrightarrow_{\mathcal{C}'})$ *are bisimilar.*

Definition 18 (Weak history-preserving). *The durational configuration structures* $\mathcal{C}$ *and* $\mathcal{C}'$ *are* weak history-preserving equivalent*, denoted by* $\mathcal{C} \approx_{wh} \mathcal{C}'$*, iff there is a bisimulation* R *between* $(\mathcal{C}, \longrightarrow_{\mathcal{C}})$ *and* $(\mathcal{C}', \longrightarrow_{\mathcal{C}'})$ *such that there is an isomorphism between the two partially ordered, labelled sets* $(\mathrm{dom}(f), <_f, l \upharpoonright \mathrm{dom}(f) \times f)$*, and* $(\mathrm{dom}(f'), <_{f'}, l \upharpoonright \mathrm{dom}(f') \times f')$ *for every* $(f, f') \in R$*.*

Definition 19 (History-preserving). *Let* $\mathcal{C}, \mathcal{C}' \in \mathbb{C}$*. A relation* $R \subseteq C \times C' \times (E \rightharpoonup E)$ *is called a* history-preserving bisimulation between $\mathcal{C}$ and $\mathcal{C}'$ *iff* $(\emptyset, \emptyset, \emptyset) \in R$ *and if* $(f, f', \iota) \in R$ *then*

- $\mathrm{dom}(\iota) = \mathrm{dom}(f)$ *and* ι *is an isomorphism between the two partially ordered, labelled sets* $(\mathrm{dom}(f), <_f, l \upharpoonright \mathrm{dom}(f) \times f)$*, and* $(\mathrm{dom}(f'), <_{f'}, l \upharpoonright \mathrm{dom}(f') \times f')$*,*
- $f \xrightarrow[d]{(a,t)}_{\mathcal{C}} g \Rightarrow \exists g', \kappa : f' \xrightarrow[d]{(a,t)}_{\mathcal{C}'} g' \wedge (g, g', \kappa) \in R \wedge \kappa \upharpoonright \mathrm{dom}(f) = \iota$,
- $f' \xrightarrow[d]{(a,t)}_{\mathcal{C}'} g' \Rightarrow \exists g, \kappa : f \xrightarrow[d]{(a,t)}_{\mathcal{C}} g \wedge (g, g', \kappa) \in R \wedge \kappa \upharpoonright \mathrm{dom}(f) = \iota$.

$\mathcal{C}$ *and* $\mathcal{C}'$ *are* history-preserving equivalent*, denoted by* $\mathcal{C} \approx_h \mathcal{C}'$*, iff there exists an history-preserving bisimulation between* $\mathcal{C}$ *and* $\mathcal{C}'$*.*

Definition 20 (Hereditary history-preserving). *The durational configuration structures* $\mathcal{C}$ *and* $\mathcal{C}'$ *are* hereditary history-preserving equivalent*, denoted by* $\mathcal{C} \approx_{hh} \mathcal{C}'$*, iff there is a history-preserving bisimulation* R *between* $\mathcal{C}$ *and* $\mathcal{C}'$ *such that* $\forall f, f', g, \iota, a, t, d : \big((f, g, \iota) \in R \wedge f' \xrightarrow[d]{(a,t)}_{\mathcal{C}} f\big) \Rightarrow (f', \iota(f'), \iota \upharpoonright \mathrm{dom}(f')) \in R$*, where* $\iota(f')_{(e')} = f'(\iota^{-1}(e'))$*.*

4 Process Algebra

In order to describe easily some durational configuration structures (e.g., to present counterexamples for the non-inclusion of equivalences), we introduce a simple event-based timed process algebra. The event-based approach is encoded in the syntax of our Process Algebra and becomes visible in the operational semantics. The process algebra expressions Expr are defined by the following BNF-grammar:

$$B ::= a_d \mid B + B \mid B; B \mid B \|_A B \mid B \backslash\backslash_d A \mid B[a \mapsto b] \mid d : B \mid \lceil B \rceil_i,$$

where $i \in \{1, 2\}$, $d \in \mathbb{R}_0^{+\infty}$, $a, b \in \mathcal{A}ct$, and $A \subseteq \mathcal{A}ct$. For reasons of simplicity, we do not consider recursion or further time-specific operators here, since these are not needed for our counterexamples given in the next section. Our process algebra can be straightforwardly extended with these constructs. A non-event-based process algebra, which only speak about actions and not events, can be straightforwardly deduced from our by removing the $\lceil \_ \rceil_i$ operators and neglecting the events in the operational semantics.

The expressions, which we sometimes call processes, have the following intuitive meaning: a_d is the process that executes a with duration d at time 0 (terminates at time d), $B_1 + B_2$ is the choice operator, $B_1; B_2$ is the sequential composition and $B_1 \|_A B_2$ is a parallel composition, where the processes have to synchronize on actions from A. An action may idle by waiting for synchronization. The restriction process $B \backslash\backslash_d A$ does not allow actions from A to be executed and termination may not happen before time point d. $B[a \mapsto b]$ is the relabelling operator. $d : B$ behaves like B except that every occurrence time is increased by d. In particular, $\infty : B$ cannot execute anything. Processes $\lceil B \rceil_i$ have the same behavior as B. They are only used for event renaming.

The event-based operational semantics of processes is given by a transition system $(\mathsf{Expr}, \longrightarrow)$, where $\longrightarrow \subseteq \mathsf{Expr} \times ((E \times \mathbb{R}_0^+ \times \mathbb{R}_0^{+\infty}) \times (\mathcal{A}ct \times \mathbb{R}_0^+)) \times (\mathsf{Expr} \cup \{\surd_d \mid d \in \mathbb{R}_0^{+\infty}\})$. The transition $B \xrightarrow[a,s]{(e,t,d)} B'$ indicates that B executes action a with identity e at time t and this action has duration d. Furthermore, this execution happens s times after the enabling of e. The transition $B \xrightarrow[a,s]{(e,t,d)} \surd_{d'}$ indicates that B terminates at time d' by execution of the action. This termination predicate is a timed version of the untimed one presented, for example, in [4]. The transition rules are presented in Figure 1. For technical reasons, we assume that $\bullet \in E$, $\star \notin E$ and for all $e, e' \in E$ we have $(e, e'), (\star, e), (e, \star) \in E$. These restriction are necessary to guarantee the uniqueness and the existence of events in process executions.

The durational configuration structure of process B, denoted by $\mathcal{C}_B$, is obtained by collecting all labels (neglecting the action name information) of a trace of B where the last component of the label is always zero (execution may not idle) to obtain a configuration of $\mathcal{C}_B$. The value on e of the labelling function of $\mathcal{C}_B$ is determined by the corresponding action label of a trace of B that contains e (where a fixed action name is used if such a trace does not exist).

$$\frac{}{a_d \xrightarrow[a,s]{(\bullet,s,d)} \surd_{d+s}}$$

$$\frac{B_1 \xrightarrow[a,s]{(e,t,d)} B'}{\begin{array}{l} B_1 + B_2 \xrightarrow[a,s]{((e,\star),t,d)} \lceil B' \rceil_1 \\ B_2 + B_1 \xrightarrow[a,s]{((\star,e),t,d)} \lceil B' \rceil_2 \end{array}}$$

$$\frac{B_1 \xrightarrow[a,s]{(e,t,d)} \surd_{d'}}{\begin{array}{l} B_1 + B_2 \xrightarrow[a,s]{((e,\star),t,d)} \surd_{d'} \\ B_2 + B_1 \xrightarrow[a,s]{((\star,e),t,d)} \surd_{d'} \end{array}}$$

$$\frac{B_1 \xrightarrow[a,s]{(e,t,d)} B_1'}{B_1; B_2 \xrightarrow[a,s]{((e,\star),t,d)} B_1'; B_2}$$

$$\frac{B_1 \xrightarrow[a,s]{(e,t,d)} \surd_{d'}}{B_1; B_2 \xrightarrow[a,s]{((e,\star),t,d)} d' : \lceil B_2 \rceil_2}$$

$$\frac{B_1 \xrightarrow[a,s]{(e,t,d)} B_1' \qquad a \notin A}{\begin{array}{l} B_1 \|_A B_2 \xrightarrow[a,s]{((e,\star),t,d)} B_1' \|_A B_2 \\ B_2 \|_A B_1 \xrightarrow[a,s]{((\star,e),t,d)} B_2 \|_A B_1' \end{array}}$$

$$\frac{B_1 \xrightarrow[a,s]{(e,t,d)} \surd_{d'} \qquad a \notin A}{\begin{array}{l} B_1 \|_A B_2 \xrightarrow[a,s]{((e,\star),t,d)} (\lceil B_2 \rceil_2) \backslash\backslash_{d'} A \\ B_2 \|_A B_1 \xrightarrow[a,s]{((\star,e),t,d)} (\lceil B_2 \rceil_1) \backslash\backslash_{d'} A \end{array}}$$

$$\frac{B_1 \xrightarrow[a,s_1]{(e_1,t,d)} B_1' \qquad B_2 \xrightarrow[a,s_2]{(e_2,t,d)} B_2' \qquad a \in A \qquad s = \min\{s_1, s_2\}}{B_1 \|_A B_2 \xrightarrow[a,s]{((e_1,e_2),t,d)} B_1' \|_A B_2'}$$

$$\frac{B_1 \xrightarrow[a,s_1]{(e_1,t,d)} \surd_{d'} \qquad B_2 \xrightarrow[a,s_2]{(e_2,t,d)} B_2' \qquad a \in A \qquad s = \min\{s_1, s_2\}}{\begin{array}{l} B_1 \|_A B_2 \xrightarrow[a,s]{((e_1,e_2),t,d)} (\lceil B_2' \rceil_2) \backslash\backslash_{d'} A \\ B_2 \|_A B_1 \xrightarrow[a,s]{((e_2,e_1),t,d)} (\lceil B_2' \rceil_1) \backslash\backslash_{d'} A \end{array}}$$

$$\frac{B_1 \xrightarrow[a,s_1]{(e_1,t,d)} \surd_{d_1} \qquad B_2 \xrightarrow[a,s_2]{(e_2,t,d)} \surd_{d_2} \qquad a \in A \qquad s = \min\{s_1, s_2\}}{B_1 \|_A B_2 \xrightarrow[a,s]{((e_1,e_2),t,d)} \surd_{\max\{d_1,d_2\}}}$$

$$\frac{B \xrightarrow[a,s]{(e,t,d)} B' \qquad a \notin A}{B \backslash\backslash_{d'} A \xrightarrow[a,s]{(e,t,d)} B' \backslash\backslash_{d'} A}$$

$$\frac{B \xrightarrow[a,s]{(e,t,d)} \surd_{d''} \qquad a \notin A}{B \backslash\backslash_{d'} A \xrightarrow[a,s]{(e,t,d)} \surd_{\max\{d',d''\}}}$$

$$\frac{B \xrightarrow[a,s]{(e,t,d)} B' \qquad a' \neq a}{\begin{array}{l} B[a \mapsto b] \xrightarrow[b,s]{(e,t,d)} B'[a \mapsto b] \\ B[a' \mapsto b] \xrightarrow[a,s]{(e,t,d)} B'[a' \mapsto b] \end{array}}$$

$$\frac{B \xrightarrow[a,s]{(e,t,d)} \surd_{d'} \qquad a' \neq a}{\begin{array}{l} B[a \mapsto b] \xrightarrow[b,s]{(e,t,d)} \surd_{d'} \\ B[a' \mapsto b] \xrightarrow[a,s]{(e,t,d)} \surd_{d'} \end{array}}$$

$$\frac{B \xrightarrow[a,s]{(e,t,d)} B' \qquad t' \neq \infty}{t' : B \xrightarrow[a,s]{(e,t+t',d)} t' : B'}$$

$$\frac{B \xrightarrow[a,s]{(e,t,d)} \surd_{d'} \qquad t' \neq \infty}{t' : B \xrightarrow[a,s]{(e,t+t',d)} \surd_{t'+d'}}$$

$$\frac{B \xrightarrow[a,s]{(e,t,d)} B'}{\begin{array}{l} \lceil B \rceil_1 \xrightarrow[a,s]{((e,\star),t,d)} \lceil B' \rceil_1 \\ \lceil B \rceil_2 \xrightarrow[a,s]{((\star,e),t,d)} \lceil B' \rceil_2 \end{array}}$$

$$\frac{B \xrightarrow[a,s]{(e,t,d)} \surd_{d'}}{\begin{array}{l} \lceil B \rceil_1 \xrightarrow[a,s]{((e,\star),t,d)} \surd_{d'} \\ \lceil B \rceil_2 \xrightarrow[a,s]{((\star,e),t,d)} \surd_{d'} \end{array}}$$

Fig. 1. Transition Rules

Example 21. Consider the process $\tilde{B} = a_1 \|_{\emptyset} (b_2; c_1)$. Then $\mathcal{C}_{\tilde{B}}$ is equal to the durational configuration structure $\tilde{\mathcal{C}}$ from Example 6 where $e_0 = (\bullet, \star)$, $e_1 = (\star, (\bullet, \star))$ and $e_2 = (\star, (\star, \bullet))$.

We present the following proposition without proof:

Proposition 22. *Let $B \in \mathsf{Expr}$. Then $\mathcal{C}_B \in \mathbb{C}$. Furthermore, if for every action a_d that occurs in B $d > 0$ holds, then $\mathcal{C}_B \in \mathbb{C}^d$. Moreover, if B does not contain subexpressions of the form $d : B'$ or $B \backslash\backslash_d A$, then $\mathcal{C}_B \in \mathbb{C}^u$.*

The above proposition illustrates, for example, that the concept of durational configuration structures is a reasonable one, since it can be naturally used for presenting a denotational semantics of timed process algebras.

5 Relations Between the Equivalences

The following theorem illustrates that there exists only a difference between the hierarchy of time equivalences versus the hierarchy of untimed equivalences, when only durational actions are allowed. In this case the step equivalences cannot be distinguished from the interleaving equivalences. All other inclusion of the equivalences are unaffected.

Theorem 23. *Let $\bowtie_{it}, \bowtie_{ib}, \bowtie_{st}, \bowtie_{sb}, \bowtie_{pt}, \bowtie_{pb}, \bowtie_{wh}, \bowtie_h, \bowtie_{hh}$ denote the corresponding equivalences restricted to $\mathbb{C}^u$, e.g., $\bowtie_{it} = \approx_{it} \cap \mathbb{C}^u \times \mathbb{C}^u$, let $\sim_{it}, \sim_{ib}, \sim_{st}, \sim_{sb}, \sim_{pt}, \sim_{pb}, \sim_{wh}, \sim_h, \sim_{hh}$ denote the corresponding equivalences restricted to $\mathbb{C}^d$, and let $\asymp_{it}, \asymp_{ib}, \asymp_{st}, \asymp_{sb}, \asymp_{pt}, \asymp_{pb}, \asymp_{wh}, \asymp_h, \asymp_{hh}$ denote the corresponding equivalences restricted to $\mathbb{C}^{du}$. Then all valid inclusion-relations between the equivalences are presented in Figure 2: If two equivalences are connected via a line, then the lower one identifies more elements than the upper one. Identical equivalences are separated by a comma.*

The rest of this section contains the proof of Theorem 23, where explicit counterexamples for the non-inclusions are presented.

5.1 Proof of the Inclusions

It is easily seen that $\approx_{hh} \subseteq \approx_h \subseteq \approx_{wh}$ and $\approx_{pb} \subseteq \approx_{sb} \subseteq \approx_{ib}$ and $\approx_{pt} \subseteq \approx_{st} \subseteq \approx_{it}$ and $\approx_{pb} \subseteq \approx_{pt}$ and $\approx_{sb} \subseteq \approx_{st}$ and $\approx_{ib} \subseteq \approx_{it}$. The proof of $\approx_h \subseteq \approx_{pb}$ can be carried out analogously to the proof of the untimed case [15] and the proof of $\approx_{wh} \subseteq \approx_{pt}$ and $\approx_{wh} \subseteq \approx_{sb}$ can also be carried out analogously to the untimed case [11].

In order to verify the inclusions $\sim_{it} \subseteq \sim_{st}$ and $\sim_{ib} \subseteq \sim_{sb}$, we introduce the following lemma. This lemma states that an ill-timed trace (a trace where time does not increase) can be replaced by a step execution only if durational actions are considered.

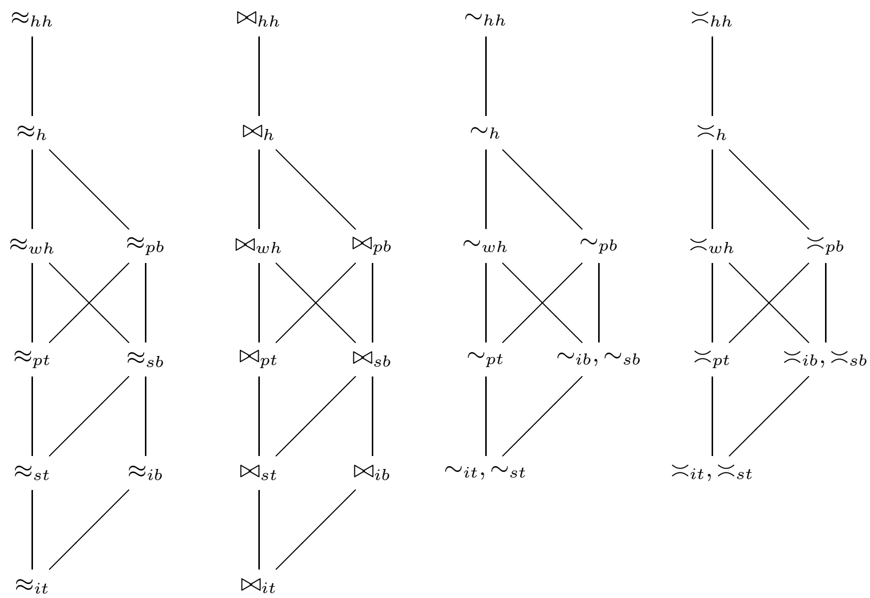


Fig. 2. Hierarchy of the Equivalences

Lemma 24. *Suppose* $\mathcal{C} \in \mathbb{C}^d$ *and* $f_1, f_2, f_3 \in C$ *such that* $f_1 \overset{\gamma}{\longmapsto}_{\mathcal{C}} f_2 \overset{(a,t)}{\underset{d}{\longrightarrow}}_{\mathcal{C}} f_3$ *and* $\forall (a', t', d') \in \mathcal{L} : \gamma(a', t', d') \neq 0 \Rightarrow t' \geq t$. *Then*

$$f_1 \overset{\gamma'}{\longmapsto}_{\mathcal{C}} f_3 \textit{ where } \gamma'(a', t', d') = \begin{cases} \gamma(a', t', d') + 1 & \textit{if } (a', t', d') = (a, t, d), \\ \gamma(a', t', d') & \textit{otherwise} . \end{cases}$$

Proof. The only non-trivial fact to show is that $e\|_{f_3}e'$ holds for every $e, e' \in (\mathrm{dom}(f_3) \setminus \mathrm{dom}(f_1))$. We proceed with the following case analysis:

$e, e' \in \mathrm{dom}(f_2) \setminus \mathrm{dom}(f_1)$: From $f_1 \overset{\gamma}{\longmapsto}_{\mathcal{C}} f_2$ we get $e\|_{f_2}e'$. This implies $e\|_{f_1}e'$, which is easily checked.

$e \in \mathrm{dom}(f_3) \setminus \mathrm{dom}(f_2)$: Without loss of generality $e' \in \mathrm{dom}(f_2) \setminus \mathrm{dom}(f_1)$. Suppose $e <_{f_3} e'$. Then by the definition of $<_{f_3}$ we get $e \in \mathrm{dom}(f_2)$, since $f_2 \sqsubseteq f_3 \wedge e' \in \mathrm{dom}(f_2)$. Contradiction.
Suppose $e' <_{f_3} e$ then $\pi_t(f_3(e')) + \pi_d(f_3(e')) \leq \pi_t(f_3(e))$, since $\mathcal{C}$ is time stable. Moreover, $\pi_t(f_3(e')) < \pi_t(f_3(e))$, since $\mathcal{C}$ has only durational action. Furthermore, $\gamma(l(e'), t', d') \neq 0$ where $(t', d') = f_3(e')$. Thus from the assumption, we get $\pi_t(f_3(e')) \geq t$. This is a contradiction, since $t = \pi_t(f_3(e))$. □

Lemma 25. $\sim_{it} \subseteq \sim_{st}$.

Proof. Suppose $\mathcal{C} \approx_{it} \mathcal{C}'$. Let $(\gamma_0, ..., \gamma_{n-1}) \in \mathcal{T}_{(\mathcal{C}, \longmapsto_{\mathcal{C}})}$. The other case follows analogously. By definition, there are $f_0, .., f_n \in C$ with $\forall i < n : f_i \overset{\gamma_i}{\longmapsto}_{\mathcal{C}} f_{i+1}$ and $\emptyset = f_0$. Take $q : \mathbb{N} \to \mathbb{N}$ and $e^i_1, ...e^i_{q(i)}$ for $i < n$ such that all e^i_j

are pairwise distinct and $\mathrm{dom}(f_{i+1}) \setminus \mathrm{dom}(f_i) = \{e^i_1, ..., e^i_{q(i)}\}$ and $\forall i < n$: $\forall j < q(i) - 1 : \pi_t(f_{i+1}(e^i_{j+1})) \geq \pi_t(f_{i+1}(e^i_{j+2}))$. Then it is easily checked that $(\alpha^0_1, ..., \alpha^0_{q(1)}, \alpha^1_1, ..., \alpha^{n-1}_{q(n-1)}) \in \mathcal{T}_{(\mathcal{C},\longrightarrow_{\mathcal{C}})}$, with $\alpha^i_j = (l(e^i_j), f_1(e^i_j))$. From $\mathcal{C} \approx_{it} \mathcal{C}'$ we obtain $(\alpha^0_1, ..., \alpha^{n-1}_{q(n-1)}) \in \mathcal{T}_{(\mathcal{C}',\longrightarrow_{\mathcal{C}'})}$. From Lemma 24 we obtain $(\gamma_0, ..., \gamma_{n-1}) \in \mathcal{T}_{(\mathcal{C}',\longmapsto_{\mathcal{C}'})}$, as required. □

Lemma 26. $\sim_{ib} \subseteq \sim_{sb}$.

Proof. Suppose $\mathcal{C} \approx_{ib} \mathcal{C}'$. Then there is a bisimulation R between $(\mathcal{C}, \longrightarrow_{\mathcal{C}})$ and $(\mathcal{C}', \longrightarrow_{\mathcal{C}'})$. In the following we show that R is also a step bisimulation between $\mathcal{C}$ and $\mathcal{C}'$. Let $(f, f') \in R$, then

$$\forall \gamma, g : f \overset{\gamma}{\longmapsto}_{\mathcal{C}} g \Rightarrow \exists g' : f' \overset{\gamma}{\longmapsto}_{\mathcal{C}} g' \wedge (g, g') \in R. \tag{1}$$

This is verified by induction on $|\sum_{\ell \in \mathcal{L}} \gamma(\ell)|$, which has to be finite, since only finite configurations are allowed. In the case $|\sum_{\ell \in \mathcal{L}} \gamma(\ell)| = 0$, we obtain $f = f'$ and therefore g' can be chosen to be g.

Suppose $|\sum_{\ell \in \mathcal{L}} \gamma(\ell)| > 0$. Choose (a, t, d) such that $\gamma(a, t, d) > 0$ and that it is minimal in the sense that $\forall (a', t', d') : \gamma(a', t', d') > 0 \Rightarrow t' \geq t$. It is easily checked that there is $\tilde{f}$ such that $f \overset{\gamma'}{\longmapsto}_{\mathcal{C}} \tilde{f} \xrightarrow[d]{(a,t)}_{\mathcal{C}} g$ with
$\gamma'(a', t', d') = \begin{cases} \gamma(a', t', d') - 1 & \text{if } (a', t', d') = (a, t, d), \\ \gamma(a', t', d') & \text{otherwise .} \end{cases}$

Then there exists $\tilde{f}'$ such that $f' \overset{\gamma'}{\longmapsto}_{\mathcal{C}} \tilde{f}' \wedge (\tilde{f}, \tilde{f}') \in R$ by induction. Since R is an interleaving bisimulation, there exists g' such that $\tilde{f}' \xrightarrow[d]{(a,t)}_{\mathcal{C}} g' \wedge (g, g') \in R$. Furthermore, $f' \overset{\gamma}{\longmapsto}_{\mathcal{C}} g'$ by Lemma 24.

Equation (1) also holds for the symmetrical case, thus R is also a bisimulation between $(\mathcal{C}, \longmapsto_{\mathcal{C}})$ and $(\mathcal{C}', \longmapsto_{\mathcal{C}'})$. Hence, $\mathcal{C} \sim_{sb} \mathcal{C}'$, as required. □

The other inclusions follow immediately from set theory, since $\mathbb{C}^{du} \subseteq \mathbb{C}^d \subseteq \mathbb{C}$ and $\mathbb{C}^{du} \subseteq \mathbb{C}^u \subseteq \mathbb{C}$.

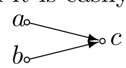
5.2 Proof of the Non-Inclusions

In the following examples, we illustrate some non-inclusions. The other non-inclusions can be derived from these counterexamples.

Example 27 ($\bowtie_{ib} \not\subseteq \bowtie_{st}$). Let $B_1 = a_0 \|_{\emptyset} b_0$ and $B'_1 = a_0; b_0 + b_0; a_0$. Obviously $\mathcal{C}_{B_1} \bowtie_{ib} \mathcal{C}_{B'_1}$ and $\mathcal{C}_{B_1} \not\bowtie_{st} \mathcal{C}_{B'_1}$.

Please note that it is essential in the counterexample presented in Example 27 that actions do not need to have a duration. In particular, the processes $\tilde{B}_1 = a_1 \|_{\emptyset} b_1$ and $\tilde{B}'_1 = a_1; b_1 + b_1; a_1$ are distinguishable in $\bowtie_{ib}$, since in $\tilde{B}'_1$ there is a trace where b happens at time 1, which cannot be the case in $\tilde{B}_1$ [2, 1].

Example 28 ($\asymp_{pt} \not\subseteq \asymp_{ib}$). Let $B_2 = a_1; (b_1 + c_1)$ and $B'_2 = a_1; b_1 + a_1; c_1$. Then it is easily seen that $\mathcal{C}_{B_2} \asymp_{pt} \mathcal{C}_{B'_2}$ and $\mathcal{C}_{B_2} \not\asymp_{ib} \mathcal{C}_{B'_2}$.

Example 29 ($\asymp_{sb} \not\subseteq \asymp_{pt}$). Let $B_3 = a_1\|_\emptyset(b_1;c_1) + (a_1;c_1)\|_\emptyset b_1$ and $B_3' = B_3 + (a_1\|_\emptyset b_1);c_1$. Then it is easily seen that $\mathcal{C}_{B_3} \asymp_{sb} \mathcal{C}_{B_3'}$. Furthermore, $\mathcal{C}_{B_3} \not\asymp_{pt} \mathcal{C}_{B_3'}$, since the pomset  can only be executed by B_3'.

Example 30 ($\asymp_{wh} \not\subseteq \asymp_{pb}$). The untimed counterexample from [15] is adapted to time as follows. Let $B_4 = \big((a_1;c_1)\|_{\{c\}}(a_1;b_1 + (a_1\|_\emptyset c_1))\big)[c \mapsto b]$ and $B_4' = ((a_1;b_1)\|_\emptyset(a_1;b_1 + a_1))\|_{\{b\}}b_1$. Then $\mathcal{C}_{B_4} \asymp_{wh} \mathcal{C}_{B_4'}$ and $\mathcal{C}_{B_4} \not\asymp_{pb} \mathcal{C}_{B_4'}$, which is argued similarly as in the untimed case [15].

Example 31 ($\asymp_{pb} \not\subseteq \asymp_{wh}$). Here, we modify the counterexample of [15] such that it applies to $\mathbb{C}^{du}$. Let $B_5 = (a_1\|_\emptyset a_1');(b_1 + c_1) + a_1\|_\emptyset(a_1';b_1) + (a_1;b_1)\|_\emptyset a_1' + (a_1\|_\emptyset a_1');b_1$ and $B_5' = (a_1\|_\emptyset a_1');(b_1 + c_1) + a_1\|_\emptyset(a_1';b_1) + (a_1;b_1)\|_\emptyset a_1'$.
Then $\mathcal{C}_{B_5} \asymp_{pb} \mathcal{C}_{B_5'}$, which can be seen as follows. The only non-obvious case is to match execution from $(a_1\|_\emptyset a_1');b_1$. Execution of $\{a\}$ is matched by $a_1\|_\emptyset(a_1';b_1)$, the execution of $\{a'\}$ and $\{a,a'\}$ are matched by $(a_1;b_1)\|_\emptyset a_1'$, and the execution of $\{a,a',b\}$ is matched by $(a_1\|_\emptyset a_1');(b_1 + c_1)$.
But $\mathcal{C}_{B_5} \not\asymp_{wh} \mathcal{C}_{B_5'}$, since the execution of a in $(a_1\|_\emptyset a_1');b_1$ has to be matched by $a_1\|_\emptyset(a_1';b_1)$ in order to be bisimilar. But then after the executions of a' and b the obtained configuration structures are not isomorphic, since in only one case a is a causality of b.

Example 32 ($\asymp_h \not\subseteq \asymp_{hh}$). The untimed counterexample from [15] can be immediately be used, since no sequential composition is used there. The counterexample is: $B_6 = (b_1\|_\emptyset(a_1 + c_1)) + (a_1\|_\emptyset b_1) + (a_1\|_\emptyset(b_1 + c_1))$ and $B_6' = (b_1\|_\emptyset(a_1 + c_1)) + (a_1\|_\emptyset(b_1 + c_1))$. See [15] for the arguments that $\mathcal{C}_{B_5} \asymp_h \mathcal{C}_{B_5'}$ and $\mathcal{C}_{B_5} \asymp_{hh} \mathcal{C}_{B_5'}$.

The other non-inclusions can be derived by the set theory from the above examples or from the fact that $\mathbb{C}^{du} \subseteq \mathbb{C}^{d} \subseteq \mathbb{C}$ and $\mathbb{C}^{du} \subseteq \mathbb{C}^{u} \subseteq \mathbb{C}$.

6 Conclusion

We have presented durational configuration structures, where events have an occurrence time and a duration. On these structures timed equivalences are introduced. We have shown that they have the same discriminating power except for durational configuration structures where every event has a positive duration. In this case interleaving and step equivalences coincide. Consequently, if one can restrict to systems where all actions have positive duration, then it is enough to consider only single action execution in order to show step equivalence, i.e., the considered transition steps are reduced, since no step execution have to be taken into account. Furthermore, we showed that no advantage results from restricting to urgent systems, since this has no influence on the hierarchy of the discriminating power.

There are many papers that introduces timed equivalences. Among them we want to mention [19], where ill-time sensitive timed bisimilarity is compared with non timed sensitive equivalences. In [26] timed configuration, where events have an occurrence time but no duration (which is the interesting aspect in our approach), are derived from a timed version of event structures. They introduce interleaving, step, and pomset trace/bisimulation equivalences. The discriminating power of these equivalences is examined on those timed configurations that are obtained from their timed event structures (and also from restricted kinds of their timed event structures).

In our paper, we only examine ill-time sensitive equivalences, i.e., the occurrence time of the events of an execution may be less than the occurrences time of a previous event. A future task is to place equivalence notions that do not allow negative time steps into the hierarchical structure of the discriminating power. Another task is to examine the equivalences where it is not a priory known how long the actions' duration will be. Such equivalences, for example the ST-equivalence [17], are especially of interest for reactive systems and/or action refinement [18]. Of course, it is also of interest to consider weak equivalences, which abstract from internal executions.

Acknowledgment. I thank Rob van Glabbeek for fruitful comments and suggestions. Furthermore, I thank Willem-Paul de Roever, Mila Majster-Cederbaum and the reviewers.

References

[1] L. Aceto and D. Murphy. On the ill-timed but well-caused. In Best [6], pages 97–111.

[2] L. Aceto and D. Murphy. Timing and causality in process algebra. *Acta Informatica*, 33:317–350, 1996.

[3] M. A. Bednarczyk. Hereditary history preserving bisimulation or what is the power of the future perfect in program logics. Technical report, Institute of Computer Science, Polish Academy of Science, 1991.

[4] J. A. Bergstra, W. Fokkink, and A. Ponse. Process algebra with recursive operations. In Bergstra et al. [5], pages 333–389.

[5] J. A. Bergstra, A. Ponse, and S. A. Smolka, editors. *Handbook of Process Algebra*. North-Holland, 2001.

[6] E. Best, editor. *CONCUR '93*, volume 715 of *LNCS*. Springer-Verlag, 1993.

[7] G. Boudol and I. Castellani. On the semantics of concurrency: Partial orders and transition systems. In H. Ehrig, R. Kowalski, G. Levi, and U. Montanari, editors, *TAPSOFT '87 (Volume 1)*, volume 249 of *LNCS*, pages 123–137. Springer-Verlag, 1987.

[8] X. J. Chen and F. Corradini. On the specification and verification of performance properties for a timed process algebra. In M. Johnson, editor, *Algebraic Methodology and Software Technology*, volume 1349 of *LNCS*, pages 123–137. Springer-Verlag, 1997.

[9] F. Corradini. Absolute versus relative time in process algebras. *Information and Computation*, 156:122–172, 2000.

[10] P. Degano, R. D. Nicola, and U. Montanari. Observational equivalences for concurrency models. In M. Wirsing, editor, *Formal Description of Programming Concepts – III, Proceedings of the* 3^{th} *IFIP WG 2.2 working conference,* Ebberup 1986, pages 105–129. North-Holland, 1987.

[11] H. Fecher. A completed hierarchy of true concurrent equivalences. *Information Processing Letters*, 89(5):261–265, 2004.

[12] R. v. Glabbeek. The linear time–branching time spectrum II: The semantics of sequential systems with silent moves (extended abstract). In Best [6], pages 66–81.

[13] R. v. Glabbeek. The linear time–branching time spectrum I. The semantics of concrete, sequential processes. In Bergstra et al. [5], pages 3–99.

[14] R. v. Glabbeek and U. Goltz. Refinement of actions in causality based models. In J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Stepwise Refinement of Distributed Systems. Models, Formalisms, Correctness*, volume 430 of *LNCS*, pages 267–300. Springer-Verlag, 1990.

[15] R. v. Glabbeek and U. Goltz. Refinement of actions and equivalence notions for concurrent systems. *Acta Informatica*, 37:229–327, 2001.

[16] R. v. Glabbeek and G. D. Plotkin. Configuration structures. In *Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science*, pages 199–209. IEEE Computer Society Press, 1995.

[17] R. v. Glabbeek and F. Vaandrager. Petri net models for algebraic theories of concurrency. In J. de Bakker, A. Nijman, and P. Treleaven, editors, *PARLE, Parallel Architectures and Languages Europe (Volume II)*, volume 259 of *LNCS*, pages 224–242. Springer-Verlag, 1987.

[18] R. Gorrieri and A. Rensink. Action refinement. In Bergstra et al. [5], pages 1047–1147.

[19] R. Gorrieri, M. Roccetti, and E. Stancampiano. A theory of processes with durational actions. *Theoretical Computer Science*, 140:73–94, 1995.

[20] C. A. R. Hoare. *Communications Sequential Processes*. International Series in Computer Science. Prentice Hall, 1985.

[21] J.-P. Katoen, R. Langerak, D. Latella, and E. Brinksma. On specifying real-time systems in a causality-based setting. In B. Jonsson and J. Parrow, editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 1135 of *LNCS*, pages 385–404. Springer-Verlag, 1996.

[22] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25:267–310, 1983.

[23] D. Murphy. Time and duration in noninterleaving concurrency. *Fundamenta Informaticae*, 19:403–416, 1993.

[24] L. Pomello. Some Equivalence Notions for Concurrent Systems. An Overview. In G. Rozenberg, editor, *Advances in Petri Nets 1985*, volume 222 of *LNCS*, pages 381–400. Springer-Verlag, 1986.

[25] A. Rabinovich and B. A. Trakhtenbrot. Behavior structures and nets. *Fundamenta Informaticae*, 11(4):357–404, 1988.

[26] I. B. Virbitskaite. Observational semantics for timed event structures. In D. Bjørner, M. Broy, and A. Zamulin, editors, *PSI 2001*, volume 2244 of *LNCS*, pages 214–224. Springer-Verlag, 2001.

[27] Y. Wang. CCS+time = an interleaving model for real time systems. In J. Leach Albert, B. Monien, and M. Rodríguez, editors, *Automata, Languages and Programming*, volume 510 of *LNCS*, pages 217–228. Springer-Verlag, 1991.

[28] G. Winskel. An introduction to event structures. In J. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, volume 354 of *LNCS*, pages 364–397. Springer-Verlag, 1989.