Universität Augsburg

Bisimulation on Speed: A Unified Approach

Gerald Lüttgen and Walter Vogler

Report 2004–15 December 2004

Institut für Informatik
D-86135 Augsburg

Bisimulation on Speed: A Unified Approach

Gerald Lüttgen[1] and Walter Vogler[2]

[1] Department of Computer Science, University of York, York YO10 5DD, U.K.,
luettgen@cs.york.ac.uk
[2] Institut für Informatik, Universität Augsburg, D–86135 Augsburg, Germany,
vogler@informatik.uni-augsburg.de

Abstract. Two process–algebraic approaches have been developed for comparing two bisimulation–equivalent processes with respect to speed: the one of Moller/Tofts equips actions with lower time bounds, while the other by Lüttgen/Vogler considers upper time bounds instead.
This paper sheds new light onto both approaches by testifying to their close relationship and brings the research into bisimulation–based faster–than preorders to a close. We introduce a general, very intuitive concept of "faster–than", which is formalised by a notion of *amortised faster–than preorder*. When closing this preorder under all contexts, exactly the two faster–than preorders investigated by Moller/Tofts and Lüttgen/Vogler arise. For processes incorporating both lower and upper time bounds we also show that the largest precongruence contained in the amortised faster–than preorder is not a proper preorder but a timed bisimulation. In the light of this result we systematically investigate under which circumstances the amortised faster–than preorder degrades to an equivalence.

Keywords. Asynchronous systems, timed process algebra, time bounds, faster–than relation, amortised faster–than preorder, bisimulation.

Correspondence. Gerald Lüttgen

Address:	Dept. of Computer Science, The University of York Heslington, York YO10 5DD, UK
E–mail:	luettgen@cs.york.ac.uk
Phone:	+44 190 443–4774
Fax:	+44 190 443–2767

1 Introduction

Process algebras provide a popular framework for modelling and analysing the communication behaviour of asynchronous systems [7]. Various extensions of classical process algebras, such as Milner's *Calculus of Communicating Systems* (CCS) [18], are also well established in the literature, including *timed process algebras* [6]. Timed process algebras add constructs for modelling timeouts and delays of actions, and thus enable one to reason not only about the communication, or functional, behaviour of processes but also about their timing behaviour. Despite the vast literature on timed process algebra, most of which has concentrated on capturing behaviour in terms of process equivalence and refinement, there is relatively little work on relating functionally equivalent processes with respect to speed. This is surprising since designers of distributed algorithms are very interested which one out of several possible solutions to a given problem is the most time efficient one. Indeed, time efficiency is not something that can only be decided once an algorithm is implemented — often *lower* and/or *upper time bounds* on the algorithm's actions are known at design time [17].

Within timed process algebra, the idea of "*faster–than*" was first addressed by Moller and Tofts [20] who studied an extension of CCS, called $\mathrm{TACS}^{\mathrm{lt}}$ in this paper, that allows for specifying lower time bounds of actions. They proposed the *MT–preorder* which refines bisimulation [18] and has recently been put on firm theoretical grounds via a full–abstraction result established by us in [15]. Previously, we had also investigated an analogous approach to extending CCS with upper time bounds of actions, which resulted in the calculus $\mathrm{TACS}^{\mathrm{ut}}$ and the *LV–preorder* [16]; this preorder was also justified intuitively by a full–abstraction result. That latter work complements research in various Petri–net [14, 23] and process–algebra [9] frameworks that are equipped with a testing semantics [11] rather than a bisimulation semantics. The main shortcoming of our previous research is that the reference preorders for the two *full–abstraction results* — though similar in spirit — are quite different in detail and indeed somewhat tuned towards the desired outcomes. In addition, we have not explored, and neither have others in the literature, the consequences of combining both lower and upper time bounds in a single setting.

Contributions. This paper presents a unified approach to studying faster–than preorders for asynchronous processes. It unifies the previously known results on faster–than preorders in two ways. Firstly, it proposes a very natural reference preorder for relating two processes with respect to speed: the *amortised faster–than preorder.* This preorder formalises the intuition that the faster process must execute each action no later than the slower process does, while both processes must be functionally equivalent in the sense of strong bisimulation [18]; here, "no later" refers to absolute time as measured from the system start, as opposed to relative time which is used in our operational semantics and describes the passing of time between actions. Although the amortised faster–than relation is more abstract than the reference preorders considered in [15, 16], we show that both the MT–preorder and the LV–preorder remain fully–abstract in $\mathrm{TACS}^{\mathrm{lt}}$ and $\mathrm{TACS}^{\mathrm{ut}}$, respectively.

Secondly, this paper characterises the largest precongruence contained in the amortised faster–than preorder when combining the calculi $\mathrm{TACS}^{\mathrm{lt}}$ and $\mathrm{TACS}^{\mathrm{ut}}$, so as to being able to specify *both* lower *and* upper time bounds of actions. This is an important open problem in the literature, and it turns out that the resulting precongruence is not a proper preorder but a variant of *timed bisimulation* [19]. The concluding part of this paper systematically investigates under which circumstances a proper preorder is obtained, and when exactly the amortised faster–than preorder degrades to an equivalence. For example, we get a positive result as in [16] when we extend $\mathrm{TACS}^{\mathrm{ut}}$ by actions that may be delayed arbitrarily long; such *lazy* actions are useful for, e.g., modelling system errors that are not bound to occur within some fixed time interval.

The full–abstraction results of this paper complete the picture of faster–than preorders within bisimulation–based process algebras. On the one hand can the various published faster–than preorders be traced back to the same notion of "faster–than", which is rooted in the concept of *amortisation.* On the other hand does the amortisation approach highlight the limits for defining a useful faster–than preorder that fully supports *compositionality.*

Organisation. The next section presents our process–algebraic framework of *Timed Asynchronous Communicating Systems* (TACS), of which both $\mathrm{TACS}^{\mathrm{lt}}$ [15, 20] and $\mathrm{TACS}^{\mathrm{ut}}$ [16] are sub–calculi. Sec. 3 then introduces the amortised faster–than preorder and generalises the full–abstraction results of [15] and [16]. For the full TACS calculus, Sec. 4 shows that the amortised faster–than preorder degrades to a congruence rather than a precongruence, when closed under all contexts, while Sec. 5 sheds further light onto the borderline between precongruence and congruence results. Finally, Secs. 6 and 7 discuss related work and present our conclusions, respectively.

2 Timed Asynchronous Communicating Systems

Our process algebra TACS combines the timed process algebras $\mathrm{TACS}^{\mathrm{lt}}$ [15] and $\mathrm{TACS}^{\mathrm{ut}}$ [16], both of which extend Milner's CCS [18] by permitting the specification of *lower* and respectively *upper time bounds* for the execution of actions and processes. These time bounds will be used in the next sections for comparing processes with respect to speed. Syntactically, TACS includes two types of actions: *lazy* actions α and *urgent* actions $\underline{\alpha}$; the idea is that the former can idle arbitrarily, while the latter have to be performed immediately. It also includes one clock prefixing operator "$\sigma.$", called *must–clock prefix*, for specifying minimum delays and another "$\underline{\sigma}.$", called *can–clock prefix*, for specifying maximum delays. Semantically and as in CCS, an action a or $\underline{a}$ communicates with the complements $\overline{a}$ or $\underline{\overline{a}}$, irrespective of whether either action is urgent. This communication results in an urgent internal action, if both participating actions are urgent, and a lazy internal action otherwise. Moreover, TACS adopts a concept of global, discrete time that behaves as follows: process $\sigma.P$ *must wait* for *at least* one time unit before it can start executing process P (lower time bound), while process $\underline{\sigma}.P$ *can wait* for *at most* one time unit (upper time bound); thus, $\underline{\sigma}$ can be understood as a potential time step. Upper time bounds are technically enforced by the concept of *maximal progress* [13], such that time can only pass if no urgent internal computation can be performed.

Syntax. The syntax of TACS is identical to CCS, except that we include the two clock–prefixing operators and distinguish between lazy and urgent actions, as discussed above. Formally, let Λ be a countably infinite set of lazy actions not including the distinguished unobservable, *internal* action τ. With every $a \in \Lambda$ we associate a *complementary action* $\overline{a}$, and define $\overline{\Lambda} =_{\mathrm{df}} \{\overline{a} \,|\, a \in \Lambda\}$. Each lazy action $a \in \Lambda$ ($\overline{a} \in \overline{\Lambda}$, τ) has an associated urgent variant, i.e., an action $\underline{a}$ ($\underline{\overline{a}}$, $\underline{\tau}$). We define $\underline{\Lambda} =_{\mathrm{df}} \{\underline{a} \,|\, a \in \Lambda\}$ and $\underline{\overline{\Lambda}} =_{\mathrm{df}} \{\underline{\overline{a}} \,|\, a \in \Lambda\}$, and take $\mathcal{A}$ ($\underline{\mathcal{A}}$) to denote the set $\Lambda \cup \overline{\Lambda} \cup \{\tau\}$ ($\underline{\Lambda} \cup \underline{\overline{\Lambda}} \cup \{\underline{\tau}\}$). Complementation is lifted to $\Lambda \cup \overline{\Lambda}$ ($\underline{\Lambda} \cup \underline{\overline{\Lambda}}$) by defining $\overline{\overline{a}} =_{\mathrm{df}} a$ ($\underline{\overline{\overline{a}}} =_{\mathrm{df}} \underline{a}$). We let $a, b, \ldots$ ($\underline{a}, \underline{b}, \ldots$) range over $\Lambda \cup \overline{\Lambda}$ ($\underline{\Lambda} \cup \underline{\overline{\Lambda}}$) and $\alpha, \beta, \ldots$ ($\underline{\alpha}, \underline{\beta}, \ldots$) over $\mathcal{A}$ ($\underline{\mathcal{A}}$). The syntax of TACS is defined as follows:

$$P \ ::= \ \mathbf{0} \mid x \mid \alpha.P \mid \underline{\alpha}.P \mid \sigma.P \mid \underline{\sigma}.P \mid P + P \mid P|P \mid P \setminus L \mid P[f] \mid \mu x.P\,,$$

where x is a *variable* taken from a countably infinite set $\mathcal{V}$ of variables, $L \subseteq \mathcal{A} \setminus \{\tau\}$ is a *restriction set*, and $f : \mathcal{A} \to \mathcal{A}$ is a *finite relabelling*. A finite relabelling satisfies the properties $f(\tau) = \tau$, $f(\overline{a}) = \overline{f(a)}$, and $|\{\alpha \,|\, f(\alpha) \neq \alpha\}| < \infty$. The set of all terms is abbreviated by $\widehat{\mathcal{P}}$, and we define $\overline{L} =_{\mathrm{df}} \{\overline{a} \,|\, a \in L\}$. We use the standard definitions for the semantic *sort* $\mathsf{sort}(P) \subseteq \Lambda \cup \overline{\Lambda}$ of some term P, *open* and *closed* terms, and *contexts* (terms with a "hole"). A variable is called *guarded* in a term if each occurrence of the variable is within the scope of an action– or σ–prefix. Moreover, we require for terms of the form $\mu x.P$ that x is guarded in P. Note that, since $\underline{\sigma}$ only denotes a potential time step, $\underline{\sigma}.P$ can perform the actions of P immediately, whence $\underline{\sigma}$ does not count as a guard. We refer to closed and guarded terms as *processes*, with the set of all processes written as $\mathcal{P}$, and let $\equiv$ stand for syntactic equality.

Semantics. The *operational semantics* of a TACS term $P \in \widehat{\mathcal{P}}$ is given by a labelled transition system and an urgent action set. The labelled transition system has the form $\langle \widehat{\mathcal{P}}, \mathcal{A} \cup \{\sigma\}, \longrightarrow, P \rangle$, where $\widehat{\mathcal{P}}$ is the set of states, $\mathcal{A} \cup \{\sigma\}$ the alphabet, $\longrightarrow \subseteq \widehat{\mathcal{P}} \times (\mathcal{A} \cup \{\sigma\}) \times \widehat{\mathcal{P}}$ the transition relation, and P the start state. Transitions labelled with an action α are called *action transitions* that, like in CCS, are either internal activities or local handshake communications in which two processes may synchronise to take a joint state change together. Transitions labelled with the clock symbol σ are called *clock transitions* representing a recurrent global synchronisation that encodes the progress of time. Note that the transition relation is labelled by ordinary (lazy) actions only. Urgency is dealt with in an orthogonal fashion by a predicate on processes, the *urgent action set*. The urgent action set of some term P is defined by the rules shown in Table 2 and contains exactly the urgent actions in which P can initially engage. Note that the communication of two complementary actions results in an *urgent* silent action only if the two participating actions are urgent.

According to our operational rules, the *action–prefix* terms $\alpha.P$ and $\underline{\alpha}.P$ may engage in action α and then behave like P. The processes $\alpha.P$ ($\alpha \in \mathcal{A}$) and $\underline{a}.P$ ($a \in \Lambda \cup \overline{\Lambda}$) may also *idle*, i.e., engage in a clock transition to themselves, as process $\mathbf{0}$ does; the rationale is that even an urgent communication action may have to wait for a communication partner; see also below. The *must–clock prefix* term $\sigma.P$ can only engage in a clock transition to P; thus, σ stands for a delay of exactly one time unit, and it can be used to define lower time bounds, since P may perform further time steps due to clock prefixes, lazy actions or waiting for a communication. The *may–clock prefix* term $\underline{\sigma}.P$ can additionally perform any action transition that P can engage in; in this sense, $\underline{\sigma}$ represents a delay of at most one time unit and can be used to define arbitrary upper time bounds.

Table 1. Operational semantics for TACS (action transitions)

Act	$\dfrac{-}{\alpha.P \xrightarrow{\alpha} P}$	uAct	$\dfrac{-}{\underline{\alpha}.P \xrightarrow{\alpha} P}$	uPre	$\dfrac{P \xrightarrow{\alpha} P'}{\underline{\sigma}.P \xrightarrow{\alpha} P'}$
Sum1	$\dfrac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}$	Sum2	$\dfrac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'}$	Rec	$\dfrac{P \xrightarrow{\alpha} P'}{\mu x.P \xrightarrow{\alpha} P'[\mu x.P/x]}$
Com1	$\dfrac{P \xrightarrow{\alpha} P'}{P\|Q \xrightarrow{\alpha} P'\|Q}$	Com2	$\dfrac{Q \xrightarrow{\alpha} Q'}{P\|Q \xrightarrow{\alpha} P\|Q'}$	Com3	$\dfrac{P \xrightarrow{a} P' \quad Q \xrightarrow{\overline{a}} Q'}{P\|Q \xrightarrow{\tau} P'\|Q'}$
Rel	$\dfrac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]}$	Res	$\dfrac{P \xrightarrow{\alpha} P'}{P \setminus L \xrightarrow{\alpha} P' \setminus L}\ \alpha \notin L \cup \overline{L}$		

Table 2. Urgent action sets

$\mathcal{U}(\alpha.P) =_{\mathrm{df}} \emptyset$	$\mathcal{U}(\underline{\alpha}.P) =_{\mathrm{df}} \{\alpha\}$	$\mathcal{U}(\mathbf{0}) =_{\mathrm{df}} \emptyset$
$\mathcal{U}(\sigma.P) =_{\mathrm{df}} \emptyset$	$\mathcal{U}(\underline{\sigma}.P) =_{\mathrm{df}} \emptyset$	$\mathcal{U}(x) =_{\mathrm{df}} \emptyset$
$\mathcal{U}(P \setminus L) =_{\mathrm{df}} \mathcal{U}(P) \setminus (L \cup \overline{L})$	$\mathcal{U}(P[f]) =_{\mathrm{df}} \{f(\alpha) \mid \alpha \in \mathcal{U}(P)\}$	$\mathcal{U}(\mu x.P) =_{\mathrm{df}} \mathcal{U}(P)$
$\mathcal{U}(P+Q) =_{\mathrm{df}} \mathcal{U}(P) \cup \mathcal{U}(Q)$	$\mathcal{U}(P\|Q) =_{\mathrm{df}} \mathcal{U}(P) \cup \mathcal{U}(Q) \cup \{\tau \mid \mathcal{U}(P) \cap \overline{\mathcal{U}(Q)} \neq \emptyset\}$	

Table 3. Operational semantics for TACS (clock transitions)

tNil	$\dfrac{-}{\mathbf{0} \xrightarrow{\sigma} \mathbf{0}}$	tAct	$\dfrac{-}{\alpha.P \xrightarrow{\sigma} \alpha.P}$	tuAct	$\dfrac{-}{\underline{a}.P \xrightarrow{\sigma} \underline{a}.P}$
tPre	$\dfrac{-}{\sigma.P \xrightarrow{\sigma} P}$	tuPre	$\dfrac{-}{\underline{\sigma}.P \xrightarrow{\sigma} P}$	tRec	$\dfrac{P \xrightarrow{\sigma} P'}{\mu x.P \xrightarrow{\sigma} P'[\mu x.P/x]}$
tSum	$\dfrac{P \xrightarrow{\sigma} P' \quad Q \xrightarrow{\sigma} Q'}{P+Q \xrightarrow{\sigma} P'+Q'}$	tCom	$\dfrac{P \xrightarrow{\sigma} P' \quad Q \xrightarrow{\sigma} Q'}{P\|Q \xrightarrow{\sigma} P'\|Q'}\ \tau \notin \mathcal{U}(P\|Q)$		
tRel	$\dfrac{P \xrightarrow{\sigma} P'}{P[f] \xrightarrow{\sigma} P'[f]}$	tRes	$\dfrac{P \xrightarrow{\sigma} P'}{P \setminus L \xrightarrow{\sigma} P' \setminus L}$		

The term $P|Q$ stands for the *parallel composition* of P and Q according to an interleaving semantics with synchronised communication on complementary actions resulting in the internal action τ. Time has to proceed equally on both sides of the operator. The side condition of Rule (tCom) ensures that $P|Q$ can only progress on σ, if it cannot engage in any urgent internal computation, in accordance with our notion of maximal progress. Thus, due to the urgency of the actions, $\underline{a}.P \,|\, \underline{\overline{a}}.Q$ cannot perform a time step. On the other hand, $\underline{a}.P \,|\, \underline{b}.Q$ or $\underline{a}.P \,|\, \overline{a}.Q$ can, since communication is not possible or can at least be delayed; thus, $\underline{a}$ is urgent but also *patient*. Note that predicates within structural operational rules, such as $\tau \notin \mathcal{U}(P|Q)$ in Rule (tCom), are well understood [22].

The *summation operator* $+$ denotes nondeterministic choice such that $P+Q$ may behave like P or Q. Again, time has to proceed equally on both sides of summation, whence $P+Q$ can engage in a clock transition and delay the nondeterministic choice if and only if both P and Q can. The *restriction operator* $\setminus L$ prohibits the execution of actions in $L \cup \overline{L}$ and, thus, permits the scoping of actions. $P[f]$ behaves exactly as P where actions are renamed by the *relabelling* f. Finally, $\mu x.\, P$ denotes *recursion*, i.e., $\mu x.\, P$ behaves as a distinguished solution of the equation $x = P$.

The rules for action transitions are the same as for CCS, with the exception of the rule for the new may–clock prefix and the rule for recursion; however, the latter is equivalent to the standard CCS rule over guarded terms [5]. It is important to note that both faster–than settings previously investigated by us in [15, 16] can be found within TACS. The sub–calculus obtained when considering only lazy actions (urgent actions) and only must–clock prefixing (can–clock prefixing) is exactly the calculus TACS^{lt} (TACS^{ut}) studied in [15] ([16]). For improving readability we also write $\mathcal{P}^{\text{lt}}$ ($\mathcal{P}^{\text{ut}}$) for the set of processes in TACS^{lt} (TACS^{ut}).

The operational semantics for TACS possesses several important properties [13]. Firstly, it is *time–deterministic*, i.e., progress of time does not resolve choices. Formally, $P \stackrel{\sigma}{\longrightarrow} P'$ and $P \stackrel{\sigma}{\longrightarrow} P''$ implies $P' \equiv P''$, for all $P, P', P'' \in \widehat{\mathcal{P}}$, which can easily be proved by induction on the structure of P. This property is very intuitive, as only actions can resolve choices, and also technically convenient. Secondly, by our variant of *maximal progress*, a guarded term P can engage in a clock transition exactly if it cannot engage in an urgent internal transition. Formally, $P \stackrel{\sigma}{\longrightarrow}$ if and only if $\tau \notin \mathcal{U}(P)$, for all guarded terms P. In particular, processes in TACS^{lt} satisfy *laziness*: they can always engage in a clock transition. Last, but not least, it is noted that the sort $\mathsf{sort}(P)$ of any process P is finite. This is because we only allow *finite* relabellings.

3 Generalised Full–Abstraction Results

This section presents our unified approach to "faster–than" by introducing a very simple and intuitive preorder, the *amortised faster–than preorder*, which captures the essence of faster–than within a bisimulation–based setting, as discussed below. Using this preorder as a reference preorder, we show that the LV–

preorder [16] and the MT–preorder [20] are fully–abstract within the TACS$^{\mathrm{ut}}$ and TACS$^{\mathrm{lt}}$ sub–calculi of TACS, respectively.

Definition 1 (Amortised faster–than preorder). A family $(\mathcal{R}_i)_{i\in\mathbb{N}}$ of relations over $\mathcal{P}$, indexed by natural numbers (including 0), is a *family of amortised faster–than relations* if, for all $i \in \mathbb{N}$, $\langle P, Q\rangle \in \mathcal{R}_i$, and $\alpha \in \mathcal{A}$:

1. $P \xrightarrow{\alpha} P'$ implies $\exists Q', k, l.\, Q \xrightarrow{\sigma}{}^{k} \xrightarrow{\alpha} \xrightarrow{\sigma}{}^{l} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}_{i+k+l}$.
2. $Q \xrightarrow{\alpha} Q'$ implies $\exists P', k, l.\ k{+}l{\leq}i,\ P \xrightarrow{\sigma}{}^{k} \xrightarrow{\alpha} \xrightarrow{\sigma}{}^{l} P'$, and $\langle P', Q'\rangle \in \mathcal{R}_{i-k-l}$.
3. $P \xrightarrow{\sigma} P'$ implies $\exists Q', k{\geq}1{-}i.\ Q \xrightarrow{\sigma}{}^{k} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}_{i-1+k}$.
4. $Q \xrightarrow{\sigma} Q'$ implies $\exists P',\ k{\leq}i{+}1.\ P \xrightarrow{\sigma}{}^{k} P'$ and $\langle P', Q'\rangle \in \mathcal{R}_{i+1-k}$.

We write $P \mathrel{\sqsupset\!\!\sim}_i Q$ if $\langle P, Q\rangle \in \mathcal{R}_i$ for some family $(\mathcal{R}_i)_{i\in\mathbb{N}}$ of amortised faster–than relations, and call $\mathrel{\sqsupset\!\!\sim}_0$ the *amortised faster–than preorder.*

Here, $\xrightarrow{\sigma}{}^{k}$ stands for k consecutive clock transitions. It is easy to show that $\mathrel{\sqsupset\!\!\sim}_0$ is indeed a preorder. While reflexivity is obvious, transitivity follows immediately from the property $\mathrel{\sqsupset\!\!\sim}_i \circ \mathrel{\sqsupset\!\!\sim}_j \;\subseteq\; \mathrel{\sqsupset\!\!\sim}_{i+j}$, for any $i, j \in \mathbb{N}$. Furthermore, $(\mathrel{\sqsupset\!\!\sim}_i)_{i\in\mathbb{N}}$ is the (componentwise) largest family of amortised faster–than relations.

Intuitively, the above definition reflects our intuition that processes that perform delays later along execution paths are faster than functionally equivalent ones that perform delays earlier; this is because the former processes are executing actions at earlier absolute times (as measured from the start of the processes) than the latter ones. As a simple example, consider the processes $P =_{\mathrm{df}} a.b.\sigma.\sigma.c.\mathbf{0}$ and $Q =_{\mathrm{df}} \sigma.a.\sigma.b.c.\mathbf{0}$. Roughly speaking, in process P, actions a, b are executed at absolute time 0 and action c at absolute time 2. In process Q, analogously, action a is executed at absolute time 1 and actions b, c at absolute time 2. Hence, every action in P is executed earlier than, or at the same absolute time as in Q, whence P is strictly faster than Q. This idea is formalised in the above definition as follows: Q is permitted to match an a from P by σa; the additional time step is saved as a credit by increasing the index of $\mathcal{R}$ such that P can perform this time step when needed, i.e., after its b. Thus, in Def. 1, an action or clock transition is matched by allowing the matching process fewer or more clock transitions as far as this is allowed by the available credit; the difference in the number of clock transitions is added to or subtracted from the credit. In this sense, our definition is a canonical translation of the idea of amortisation.

The remainder of this paper is concerned with the characterisation of the largest precongruence contained in $\mathrel{\sqsupset\!\!\sim}_0$, for various sub–calculi of TACS, in particular TACS$^{\mathrm{ut}}$ and TACS$^{\mathrm{lt}}$. We will also discuss below, which variants of $\mathrel{\sqsupset\!\!\sim}_0$ have been used for TACS$^{\mathrm{ut}}$ and TACS$^{\mathrm{lt}}$ in [15, 16], and for notational convenience we will write $\mathrel{\sqsupset\!\!\sim}_i^{\mathrm{ut}}$ and $\mathrel{\sqsupset\!\!\sim}_i^{\mathrm{lt}}$ when restricting $\mathrel{\sqsupset\!\!\sim}_i$ to processes in TACS$^{\mathrm{ut}}$ and TACS$^{\mathrm{lt}}$, respectively. The technical development of our characterisations will rely on the following well–known result from universal algebra.

Theorem 2 (Universal Algebra). *For every preorder X over* TACS *processes, there exists a largest precongruence X^c in X satisfying*

$$X^c = \{\langle P, Q\rangle \mid \langle C[P], C[Q]\rangle \in X \textit{ for all contexts } C[\_]\} \,.$$

If Y is a further TACS *preorder such that $X^c \subseteq Y \subseteq X$, then $X^c = Y^c$.*

3.1 The LV–Preorder is Fully Abstract in TACS$^{\mathrm{ut}}$

TACS$^{\mathrm{ut}}$ is the sub–calculus of TACS that emerges when restricting ourselves to urgent actions $\underline{\alpha}$ and can–clock prefixing $\underline{\sigma}$ only, i.e., disregarding lazy actions and must–clock prefixing. We start off by recalling some definitions and a key result of [16].

Definition 3 (LV–preorder [16]). A relation $\mathcal{R}$ over $\mathcal{P}^{\mathrm{ut}}$ is an *LV–relation* if, for all $\langle P, Q\rangle \in \mathcal{R}$ and $\alpha \in \mathcal{A}$:

1. $P \xrightarrow{\alpha} P'$ implies $\exists Q'.\, Q \xrightarrow{\alpha} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.
2. $Q \xrightarrow{\alpha} Q'$ implies $\exists P'.\, P \xrightarrow{\alpha} P'$ and $\langle P', Q'\rangle \in \mathcal{R}$.
3. $P \xrightarrow{\sigma} P'$ implies $\mathcal{U}(Q) \subseteq \mathcal{U}(P)$ and $\exists Q'.\, Q \xrightarrow{\sigma} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.

We write $P \sqsupseteq\!\!\!\!\sim_{\mathrm{lv}} Q$ if $\langle P, Q\rangle \in \mathcal{R}$ for some LV–relation $\mathcal{R}$, and call $\sqsupseteq\!\!\!\!\sim_{\mathrm{lv}}$ the LV–preorder.

This definition is of an elegant simplicity, since an LV–relation essentially combines bisimulation on actions with simulation on clock steps; the condition on the inclusion of urgent sets had to be added to obtain a precongruence for parallel composition.

We also introduced in [16] an amortised variant of the LV–preorder which, in contrast to the amortised faster–than preorder of Def. 1, does not allow for leading and trailing clock transitions when matching action transitions — just as for the LV–preorder. Also, for matching clock transitions, the increase or decrease of the credit is restricted.

Definition 4 (Amortised LV–preorder [16]). A family $(\mathcal{R}_i)_{i\in\mathbb{N}}$ of relations over $\mathcal{P}^{\mathrm{ut}}$ is a *family of amortised LV–relations* if, for all $i \in \mathbb{N}$, $\langle P, Q\rangle \in \mathcal{R}_i$, and $\alpha \in \mathcal{A}$:

1. $P \xrightarrow{\alpha} P'$ implies $\exists Q'.\, Q \xrightarrow{\alpha} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}_i$.
2. $Q \xrightarrow{\alpha} Q'$ implies $\exists P'.\, P \xrightarrow{\alpha} P'$ and $\langle P', Q'\rangle \in \mathcal{R}_i$.
3. $P \xrightarrow{\sigma} P'$ implies (a) $\exists Q'.\, Q \xrightarrow{\sigma} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}_i$, or
   (b) $i{>}0$ and $\langle P', Q\rangle \in \mathcal{R}_{i-1}$.
4. $Q \xrightarrow{\sigma} Q'$ implies (a) $\exists P'.\, P \xrightarrow{\sigma} P'$ and $\langle P', Q'\rangle \in \mathcal{R}_i$, or
   (b) $\langle P, Q'\rangle \in \mathcal{R}_{i+1}$.

We write $P \sqsupseteq\!\!\!\!\sim_i^{\mathrm{lv}} Q$ if $\langle P, Q\rangle \in \mathcal{R}_i$ for some family $(\mathcal{R}_i)_{i\in\mathbb{N}}$ of amortised LV–relations, and call $\sqsupseteq\!\!\!\!\sim_0^{\mathrm{lv}}$ the *amortised LV–preorder*.

An important result of [16] that relates the above preorders is the following.

Theorem 5 (Full abstraction [16]).
The LV–preorder $\sqsupseteq_{lv}$ *is the* largest precongruence *contained in* $\sqsupseteq_0^{lv}$.

The next theorem is the main result of this section and, because of $\sqsupseteq_0^{\mathrm{lv}} \subseteq \sqsupseteq_0^{\mathrm{ut}}$, generalises the above theorem.

Theorem 6 (Generalised full abstraction in TACS$^{\mathbf{ut}}$).
The LV–preorder $\sqsupseteq_{lv}$ *is the* largest precongruence *contained in* $\sqsupseteq_0^{ut}$.

Proof. According to Thms. 2 and 5 it is sufficient to establish $(\sqsupseteq_0^{\mathrm{ut}})^{\mathrm{c}} \subseteq \sqsupseteq_0^{\mathrm{lv}} \subseteq \sqsupseteq_0^{\mathrm{ut}}$. The inclusion $\sqsupseteq_0^{\mathrm{lv}} \subseteq \sqsupseteq_0^{\mathrm{ut}}$ is obvious from the definition of both preorders. For proving $(\sqsupseteq_0^{\mathrm{ut}})^{\mathrm{c}} \subseteq \sqsupseteq_0^{\mathrm{lv}}$ we show that $\sqsupseteq_i^{\mathrm{aux}} =_{\mathrm{df}} \{\langle P, Q\rangle \,|\, C_{PQ}[P] \sqsupseteq_i^{\mathrm{ut}} C_{PQ}[Q]\}$, for $i \in \mathbb{N}$, $C_{PQ}[\_] =_{\mathrm{df}} \_ \,|\, \mu x.\, \underline{\tau}.(\underline{\sigma}.\underline{\tau}.x + \underline{d}.\mathbf{0})$ and a 'fresh' action $\underline{d}$ that is not in the sorts of P and Q, defines a family of amortised LV–relations. Note that, obviously, $(\sqsupseteq_0^{\mathrm{ut}})^{\mathrm{c}} \subseteq \sqsupseteq_0^{\mathrm{aux}}$.

Let $P \sqsupseteq_i^{\mathrm{aux}} Q$, for some $i \in \mathbb{N}$. We have to check the four conditions of Def. 4:

1. $\underline{P \xrightarrow{\alpha} P'}$:
   Hence, $C_{PQ}[P] \xrightarrow{\alpha} C_{PQ}[P']$. Since $C_{PQ}[P] \sqsupseteq_i^{\mathrm{ut}} C_{PQ}[Q]$, there exist $\widehat{Q}', k, l$ such that $C_{PQ}[Q] \xrightarrow{\sigma}{}^k \xrightarrow{\alpha} \xrightarrow{\sigma}{}^l \widehat{Q}'$ and $C_{PQ}[P'] \sqsupseteq_{i+k+l}^{\mathrm{ut}} \widehat{Q}'$. We observe that $C_{PQ}[Q]$ always offers an initial urgent $\underline{\tau}$, i.e., $\tau \in \mathcal{U}(C_{PQ}[Q])$, and that — to deal with the case $\alpha = \tau$ — the τ–derivative of the context enables the distinguished urgent action $\underline{d}$, which is not offered by $C_{PQ}[P']$; we conclude that $k{=}l{=}0$ and $\widehat{Q}' \equiv C_{PQ}[Q']$ for some Q' with $Q \xrightarrow{\alpha} Q'$. In addition we obviously have $\mathsf{sort}(P') \subseteq \mathsf{sort}(P)$ and $\mathsf{sort}(Q') \subseteq \mathsf{sort}(Q)$, which means by construction of the context $C_{PQ}[\_]$ that $C_{PQ}[P'] \sqsupseteq_i^{\mathrm{ut}} C_{PQ}[Q']$ implies $C_{P'Q'}[P'] \sqsupseteq_i^{\mathrm{ut}} C_{P'Q'}[Q']$. In summary we have established the existence of a Q' such that $Q \xrightarrow{\alpha} Q'$ and $P' \sqsupseteq_i^{\mathrm{aux}} Q'$.
2. $\underline{Q \xrightarrow{\alpha} Q'}$:
   Hence, $C_{PQ}[Q] \xrightarrow{\alpha} C_{PQ}[Q']$. Because of $C_{PQ}[P] \sqsupseteq_i^{\mathrm{ut}} C_{PQ}[Q]$ we know of the existence of $\widehat{P}', k, l$ such that $k{+}l{\le}i$, $C_{PQ}[P] \xrightarrow{\sigma}{}^k \xrightarrow{\alpha} \xrightarrow{\sigma}{}^l \widehat{P}'$, and $\widehat{P}' \sqsupseteq_{i-k-l}^{\mathrm{ut}} C_{PQ}[Q']$. Again, due to the τ–derivative in the context enabling the distinguished action $\underline{d}$ and since $\tau \in \mathcal{U}(C_{PQ}[\_])$, we have $k{=}l{=}0$ and $\widehat{P}' \equiv C_{PQ}[P']$ for a P' with $P \xrightarrow{\alpha} P'$. As above we infer $C_{P'Q'}[P'] \sqsupseteq_i^{\mathrm{ut}} C_{P'Q'}[Q']$. Summarising, there exists some P' satisfying $P \xrightarrow{\alpha} P'$ and $P' \sqsupseteq_i^{\mathrm{aux}} Q'$.
3. $\underline{P \xrightarrow{\sigma} P'}$:
   Hence, $C_{PQ}[P]$ can engage in the following three–step sequence of transitions: $C_{PQ}[P] \xrightarrow{\tau} P \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\sigma} P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\tau} C_{PQ}[P']$, where $H_{PQ} =_{\mathrm{df}} \mu x.\, \underline{\tau}.(\underline{\sigma}.\underline{\tau}.x + \underline{d}.\mathbf{0})$.
   Starting with the premise $C_{PQ}[P] \sqsupseteq_i^{\mathrm{ut}} C_{PQ}[Q]$ and the first step of $C_{PQ}[P]$ above, as well as considering the urgent $\underline{\tau}$–actions in $C_{PQ}[\_]$ and $\underline{d}$ being a distinguished action, we find ourselves in one of the following two cases:

(a) $C_{PQ}[Q] \xrightarrow{\tau}\xrightarrow{\sigma} Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$ for some Q' such that $Q \xrightarrow{\sigma} Q'$ and $P \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \sqsupseteq\!\!\!\sim^{\text{ut}}_{i+1} Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$

(b) $C_{PQ}[Q] \xrightarrow{\tau} Q \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$ and, moreover, $P \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \sqsupseteq\!\!\!\sim^{\text{ut}}_{i} Q \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$.

We consider each case in turn.

(a) The second step of $C_{PQ}[P]$, i.e., the clock transition, must be trivially matched by $Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$ since $\tau \in \mathcal{U}(Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}))$. Hence, $P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$ must have used one credit when performing its clock transition and $P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \sqsupseteq\!\!\!\sim^{\text{ut}}_{i} Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$.

(b) The second step of $C_{PQ}[P]$ can only be matched by either

  i. a single clock transition $Q \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\sigma} Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$, for some Q' with $Q \xrightarrow{\sigma} Q'$, and $P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \sqsupseteq\!\!\!\sim^{\text{ut}}_{i} Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$; note that further clock transitions are impossible since the first one makes the second urgent $\underline{\tau}$–action of the context available; or

  ii. consuming one credit (only applicable if $i{>}0$), i.e., $P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \sqsupseteq\!\!\!\sim^{\text{ut}}_{i-1} Q \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$.

In Cases (3a) and (3(b)i), the third step of $C_{PQ}[P]$ above can only be matched by $Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\tau} C_{PQ}[Q']$ such that $C_{PQ}[P'] \sqsupseteq\!\!\!\sim^{\text{ut}}_{i} C_{PQ}[Q']$, because of the distinguished $\underline{d}$–action and the $\underline{\tau}$–actions of the context.

In Case (3(b)ii), the third step of $C_{PQ}[P]$ above implies, due to the distinguished $\underline{d}$–action and the urgent $\underline{\tau}$–actions offered by the context, that either

1. $Q \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\sigma} Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\tau} C_{PQ}[Q']$ for some Q' such that $Q \xrightarrow{\sigma} Q'$ and $C_{PQ}[P'] \sqsupseteq\!\!\!\sim^{\text{ut}}_{i} C_{PQ}[Q']$; or
2. $Q \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\tau} C_{PQ}[Q]$ and $C_{PQ}[P'] \sqsupseteq\!\!\!\sim^{\text{ut}}_{i-1} C_{PQ}[Q]$. The latter implies $C_{P'Q}[P'] \sqsupseteq\!\!\!\sim^{\text{ut}}_{i-1} C_{P'Q}[Q]$ since $\mathsf{sort}(P') \subseteq \mathsf{sort}(P)$.

Summarising, for Cases (3a), (3(b)i) and (3(b)ii.1), we have established the existence of a Q' satisfying $Q \xrightarrow{\sigma} Q'$ and $P' \sqsupseteq\!\!\!\sim^{\text{aux}}_{i} Q'$. For Case (3(b)ii.2) we have $i{>}0$ and $P' \sqsupseteq\!\!\!\sim^{\text{aux}}_{i-1} Q$.

4. $\underline{Q \xrightarrow{\sigma} Q'}$:

Hence, $C_{PQ}[Q]$ can engage in the following three–step sequence of transitions: $C_{PQ}[Q] \xrightarrow{\tau} Q \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\sigma} Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\tau} C_{PQ}[Q']$, where H_{PQ} is defined as above. Considering the first step of this sequence and the premise $C_{PQ}[P] \sqsupseteq\!\!\!\sim^{\text{ut}}_{i} C_{PQ}[Q]$, we find ourselves in one of the following two cases, again due to the $\underline{\tau}$– and $\underline{d}$–actions of the context:

(a) $C_{PQ}[P] \xrightarrow{\tau} P \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$, for which $P \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \sqsupseteq\!\!\!\sim^{\text{ut}}_{i} Q \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$.

(b) $C_{PQ}[P]$ consumes one additional credit after τ (only applicable if $i{>}0$), i.e. $C_{PQ}[P] \xrightarrow{\tau} P \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\sigma} P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$ as well as $P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \sqsupseteq\!\!\!\sim^{\text{ut}}_{i-1} Q \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$, where P' is such that $P \xrightarrow{\sigma} P'$.

We consider each case in turn.

(a) For matching the second step of the above three–step sequence, there exist two possibilities:

i. $P \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$ does nothing and gains one credit, which leads to $P \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \mathrel{\underset{\sim}{\sqsupset}}^{\mathrm{ut}}_{i+1} Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$. The third step of $C_{PQ}[Q]$ is then matched by either

1. $P \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\sigma} P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\tau} C_{PQ}[P']$ and $C_{PQ}[P'] \mathrel{\underset{\sim}{\sqsupset}}^{\mathrm{ut}}_{i} C_{PQ}[Q']$, where P' is such that $P \xrightarrow{\sigma} P'$; note that the τ–step can only be performed by the context as action $\underline{d}$ is distinguished; or
2. $P \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\tau} C_{PQ}[P]$ and $C_{PQ}[P] \mathrel{\underset{\sim}{\sqsupset}}^{\mathrm{ut}}_{i+1} C_{PQ}[Q']$; we get $C_{PQ'}[P] \mathrel{\underset{\sim}{\sqsupset}}^{\mathrm{ut}}_{i+1} C_{PQ'}[Q']$ because of $\mathsf{sort}(Q') \subseteq \mathsf{sort}(Q)$.

ii. $P \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\sigma} P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$, where $P \xrightarrow{\sigma} P'$ such that $P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \mathrel{\underset{\sim}{\sqsupset}}^{\mathrm{ut}}_{i} Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$. Note that $P \,|\, (\underline{\sigma}.\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$ cannot engage in more than one clock transition, due to the availability of an urgent $\underline{\tau}$ in the context after the first clock transition. The third step of $C_{PQ}[Q]$ can only be matched by $P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0}) \xrightarrow{\tau} C_{PQ}[P']$, because of the urgent $\underline{\tau}$–actions and the distinguished action $\underline{d}$ in the context.

Summarising, in Cases (4(a)i.1) and (4(a)ii) we have shown the existence of some P' with $P \xrightarrow{\sigma} P'$ and $P' \mathrel{\underset{\sim}{\sqsupset}}^{\mathrm{aux}}_{i} Q'$. In Case (4(a)i.2) we have established $P \mathrel{\underset{\sim}{\sqsupset}}^{\mathrm{aux}}_{i+1} Q'$.

(b) Since $P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}.\mathbf{0})$ cannot perform any time step, it must match the second step of $C_{PQ}[Q]$ by doing nothing; thus, we find ourselves in Case (4(a)ii) again, which we had just settled. □

3.2 The MT–Preorder is Fully Abstract in TACS$^{\mathrm{lt}}$

We turn our attention to the TACS sub–calculus TACS$^{\mathrm{lt}}$ in which only lazy actions α and the must–clock prefix σ are available, but not urgent actions and the can–clock prefix. Although a σ–prefix corresponds to exactly one time unit, these prefixes specify lower time bounds for actions in this fragment, since actions can always be delayed arbitrarily. We first recall the faster–than preorder introduced by Moller and Tofts in [20], to which we refer as *Moller–Tofts preorder*, or MT–preorder for short.

Definition 7 (MT–preorder [20]). A relation $\mathcal{R}$ over $\mathcal{P}^{\mathrm{lt}}$ is an *MT–relation* if, for all $\langle P, Q\rangle \in \mathcal{R}$ and $\alpha \in \mathcal{A}$:

1. $P \xrightarrow{\alpha} P'$ implies $\exists Q', k, P''.\, Q \xrightarrow{\sigma}{}^{k} \xrightarrow{\alpha} Q'$, $P' \xrightarrow{\sigma}{}^{k} P''$, and $\langle P'', Q'\rangle \in \mathcal{R}$.
2. $Q \xrightarrow{\alpha} Q'$ implies $\exists P'.\, P \xrightarrow{\alpha} P'$ and $\langle P', Q'\rangle \in \mathcal{R}$.
3. $P \xrightarrow{\sigma} P'$ implies $\exists Q'.\, Q \xrightarrow{\sigma} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.
4. $Q \xrightarrow{\sigma} Q'$ implies $\exists P'.\, P \xrightarrow{\sigma} P'$ and $\langle P', Q'\rangle \in \mathcal{R}$.

We write $P \mathrel{\underset{\sim}{\sqsupset}}_{\mathrm{mt}} Q$ if $\langle P, Q\rangle \in \mathcal{R}$ for some MT–relation $\mathcal{R}$, and call $\mathrel{\underset{\sim}{\sqsupset}}_{\mathrm{mt}}$ the *MT–preorder*.

It is easy to see that $\sqsupseteq\!\!\!\sim_{\mathrm{mt}}$ is indeed a preorder and that it is the largest MT–relation. We have also proved in [15] that $\sqsupseteq\!\!\!\sim_{\mathrm{mt}}$ is a precongruence for all TACS$^{\mathrm{lt}}$ operators. The only difficult and non–standard part of that proof concerned compositionality regarding parallel composition and was based on the following *commutation lemma*.

Lemma 8 (Commutation lemma [15]). *Let* $P,\ P' \in \mathcal{P}^{lt}$ *and* $w \in (\mathcal{A} \cup \{\sigma\})^*$. *If* $P \xrightarrow{w} \xrightarrow{\sigma}{}^{k} P'$, *for* $k \in \mathbb{N}$, *then* $\exists P''.\, P \xrightarrow{\sigma}{}^{k} \xrightarrow{w} P''$ *and* $P' \sqsupseteq\!\!\!\sim_{mt} P''$.

This lemma holds as well within the slightly more general setting of Sec. 5.2, in which also can–clock prefixes are allowed. We also introduced in [15] an amortised variant of the MT–preorder, which is however less abstract than the amortised faster–than preorder of Def. 1.

Definition 9 (Amortised MT–preorder [15]). A family $(\mathcal{R}_i)_{i \in \mathbb{N}}$ of relations over $\mathcal{P}^{\mathrm{lt}}$ is a *family of amortised MT–relations* if, for all $i \in \mathbb{N}$, $\langle P, Q \rangle \in \mathcal{R}_i$, and $\alpha \in \mathcal{A}$:

1. $P \xrightarrow{\alpha} P'$ implies $\exists Q', k.\, Q \xrightarrow{\sigma}{}^{k} \xrightarrow{\alpha} Q'$ and $\langle P', Q' \rangle \in \mathcal{R}_{i+k}$.
2. $Q \xrightarrow{\alpha} Q'$ implies $\exists P', k \leq i.\ P \xrightarrow{\sigma}{}^{k} \xrightarrow{\alpha} P'$ and $\langle P', Q' \rangle \in \mathcal{R}_{i-k}$.
3. $P \xrightarrow{\sigma} P'$ implies $\exists Q', k \geq 0.\ k \geq 1-i,\ Q \xrightarrow{\sigma}{}^{k} Q'$, and $\langle P', Q' \rangle \in \mathcal{R}_{i-1+k}$.
4. $Q \xrightarrow{\sigma} Q'$ implies $\exists P',\ k \geq 0.\ k \leq i+1,\ P \xrightarrow{\sigma}{}^{k} P'$, and $\langle P', Q' \rangle \in \mathcal{R}_{i+1-k}$.

We write $P \sqsupseteq\!\!\!\sim_i^{\mathrm{mt}} Q$ if $\langle P, Q \rangle \in \mathcal{R}_i$ for some family $(\mathcal{R}_i)_{i \in \mathbb{N}}$ of amortised MT–relations, and call $\sqsupseteq\!\!\!\sim_0^{\mathrm{mt}}$ the *amortised MT–preorder*.

When comparing Defs. 9 and 1, it is obvious that $\sqsupseteq\!\!\!\sim_0^{\mathrm{mt}} \subseteq \sqsupseteq\!\!\!\sim_0^{\mathrm{lt}}$. While Conds. (3) and (4) coincide in Defs. 9 and 1, Conds. (1) and (2) do not allow clock transitions to trail the matching α–transition — just as it is the case in Cond. (1) in Def. 7. We recall the following full–abstraction result from [15].

Theorem 10 (Full abstraction [15]).
The MT–preorder $\sqsupseteq\!\!\!\sim_{mt}$ *is the* largest precongruence *contained in* $\sqsupseteq\!\!\!\sim_0^{mt}$.

We generalise this full–abstraction result here by replacing $\sqsupseteq\!\!\!\sim_0^{\mathrm{mt}}$ by $\sqsupseteq\!\!\!\sim_0^{\mathrm{lt}}$.

Theorem 11 (Generalised full abstraction in TACS$^{\mathbf{lt}}$).
The MT–preorder $\sqsupseteq\!\!\!\sim_{mt}$ *is the* largest precongruence *contained in* $\sqsupseteq\!\!\!\sim_0^{lt}$.

The proof of this theorem requires the following proposition, which closes the gap between Conds. (1) of Defs. 9 and 1.

Proposition 12. *In a setting with lazy actions only, Cond. (1) of our definition of* $\sqsupseteq\!\!\!\sim_i$ *(cf. Def. 1) can be replaced by*

(1') $\quad P \xrightarrow{\alpha} P'$ *implies* $\exists Q', k.\, Q \xrightarrow{\sigma}{}^{k} \xrightarrow{\alpha} Q'$ *and* $\langle P', Q' \rangle \in \mathcal{R}_{i+k}$.

without changing the preorder, i.e., $\sqsupseteq\!\!\!\!\sim_0 = \sqsupseteq\!\!\!\!\sim'_0$, when referring to the family of faster–than relations using Cond. (1') instead of Cond. (1) as $(\sqsupseteq\!\!\!\!\sim'_i)_{i\in\mathbb{N}}$.

Proof. The inclusion $\sqsupseteq\!\!\!\!\sim'_0 \subseteq \sqsupseteq\!\!\!\!\sim_0$ is obvious, as Cond. (1) is less stringent than Cond. (1'). For establishing the other inclusion we show that $(\sqsupseteq\!\!\!\!\sim_i)_{i\in\mathbb{N}}$ is an amortised faster–than family in the sense of Cond. (1'). It suffices to consider the case $P \sqsupseteq\!\!\!\!\sim_i Q$ and $P \xrightarrow{\alpha} P'$ for some P' and α, as Conds. (2)–(4) are the same for both faster–than families.

In this case, the definition of $\sqsupseteq\!\!\!\!\sim_i$ yields the existence of $Q', \widehat{Q}', k, l$ such that $Q \xrightarrow{\sigma}{}^k \widehat{Q}' \xrightarrow{\alpha}\xrightarrow{\sigma}{}^l Q'$ and $P' \sqsupseteq\!\!\!\!\sim_{i+k+l} Q'$. The commutation lemma, Lemma 8, then provides a Q'' satisfying $\widehat{Q}' \xrightarrow{\sigma}{}^l \xrightarrow{\alpha} Q''$ and $Q' \sqsupseteq\!\!\!\!\sim_{\mathrm{mt}} Q''$. Since $\sqsupseteq\!\!\!\!\sim_{\mathrm{mt}} = (\sqsupseteq\!\!\!\!\sim_0^{\mathrm{mt}})^{\mathrm{c}} \subseteq \sqsupseteq\!\!\!\!\sim_0^{\mathrm{mt}}$, by Thm. 10, and $\sqsupseteq\!\!\!\!\sim_0^{\mathrm{mt}} \subseteq \sqsupseteq\!\!\!\!\sim_0$, we have $Q' \sqsupseteq\!\!\!\!\sim_0 Q''$. Further, by the property $\sqsupseteq\!\!\!\!\sim_m \circ \sqsupseteq\!\!\!\!\sim_n \subseteq \sqsupseteq\!\!\!\!\sim_{m+n}$ for any $m, n \in \mathbb{N}$, we conclude from $P' \sqsupseteq\!\!\!\!\sim_{i+k+l} Q' \sqsupseteq\!\!\!\!\sim_0 Q''$ that $P' \sqsupseteq\!\!\!\!\sim_{i+k+l} Q''$. Summarising we have established the existence of a Q'' such that $Q \xrightarrow{\sigma}{}^{k+l} \xrightarrow{\alpha} Q''$ and $P' \sqsupseteq\!\!\!\!\sim_{i+(k+l)} Q''$, as desired. □

For the purposes of this section we only consider $\sqsupseteq\!\!\!\!\sim'_0$ on processes in $\mathrm{TACS}^{\mathrm{lt}}$. We are now able to prove Thm. 11.

Proof. [of Thm. 11] Because of Thm. 10 and Prop. 12, it suffices to show $(\sqsupseteq\!\!\!\!\sim_0^{\mathrm{mt}})^{\mathrm{c}} = (\sqsupseteq\!\!\!\!\sim'_0)^{\mathrm{c}}$. According to Thm. 2 this can be done by establishing $(\sqsupseteq\!\!\!\!\sim'_0)^{\mathrm{c}} \subseteq \sqsupseteq\!\!\!\!\sim_0^{\mathrm{mt}} \subseteq \sqsupseteq\!\!\!\!\sim'_0$. The inclusion $\sqsupseteq\!\!\!\!\sim_0^{\mathrm{mt}} \subseteq \sqsupseteq\!\!\!\!\sim'_0$ is obvious since Cond. (2) of $\sqsupseteq\!\!\!\!\sim_i^{\mathrm{mt}}$ is stronger than Cond. (2) of $\sqsupseteq\!\!\!\!\sim'_i$, for any $i \in \mathbb{N}$. The other inclusion $(\sqsupseteq\!\!\!\!\sim'_0)^{\mathrm{c}} \subseteq \sqsupseteq\!\!\!\!\sim_0^{\mathrm{mt}}$ follows from the fact that $\sqsupseteq\!\!\!\!\sim_i^{\mathrm{aux}} =_{\mathrm{df}} \{\langle P, Q\rangle \mid C_{PQ}[P] \sqsupseteq\!\!\!\!\sim'_i C_{PQ}[Q] \mid \Pi^i d.\mathbf{0}\}$ is a family of amortised MT–relations in the sense of Def. 9. Here, $C_{PQ}[\_] =_{\mathrm{df}} \_ \mid \mu x.\, \sigma.(d.\mathbf{0} \mid x)$, with d being a distinguished action not in the sorts of P and Q. Moreover, $\Pi^i d.\mathbf{0}$ denotes i replications of the parallel component $d.\mathbf{0}$; for notational convenience we will identify some process $\_ \mid \mathbf{0}$ with $\_$ in the remainder of this paper. Again, $(\sqsupseteq\!\!\!\!\sim'_0)^{\mathrm{c}} \subseteq \sqsupseteq\!\!\!\!\sim_0^{\mathrm{aux}}$ is obvious.

We now prove that $(\sqsupseteq\!\!\!\!\sim_i^{\mathrm{aux}})_{i\in\mathbb{N}}$ is indeed a family of amortised MT–relations. Let $P \sqsupseteq\!\!\!\!\sim_i^{\mathrm{aux}} Q$ be arbitrary; we have to check the four conditions of Def. 9:

1. $\underline{P \xrightarrow{\alpha} P'}$:
   Hence, $C_{PQ}[P] \xrightarrow{\alpha} C_{PQ}[P']$. Since $P \sqsupseteq\!\!\!\!\sim_i^{\mathrm{aux}} Q$ we know of some $\widehat{Q}', k$ such that $C_{PQ}[Q] \mid \Pi^i d.\mathbf{0} \xrightarrow{\sigma}{}^k \xrightarrow{\alpha} \widehat{Q}'$ and $C_{PQ}[P'] \sqsupseteq\!\!\!\!\sim'_{i+k} \widehat{Q}'$. By the context's construction, this implies $\widehat{Q}' \equiv C_{PQ}[Q'] \mid \Pi^{i+k} d.\mathbf{0}$, for a Q' with $Q \xrightarrow{\sigma}{}^k \xrightarrow{\alpha} Q'$. Because of $\mathsf{sort}(P') \subseteq \mathsf{sort}(P)$ and $\mathsf{sort}(Q') \subseteq \mathsf{sort}(Q)$, it also follows that $P' \sqsupseteq\!\!\!\!\sim_{i+k}^{\mathrm{aux}} Q'$.
2. $\underline{Q \xrightarrow{\alpha} Q'}$:
   Hence, $C_{PQ}[Q] \mid \Pi^i d.\mathbf{0} \xrightarrow{d}{}^i C_{PQ}[Q] \xrightarrow{\alpha} C_{PQ}[Q']$. Since (i) $P \sqsupseteq\!\!\!\!\sim_i^{\mathrm{aux}} Q$, (ii) d is a distinguished action, and (iii) $C_{PQ}[P]$ has only i credits available, $C_{PQ}[P]$ can only match the i d–transitions by $C_{PQ}[P] (\xrightarrow{\sigma}\xrightarrow{d})^i C_{PQ}[P'']$ (essentially), where $P \xrightarrow{\sigma}{}^i P''$ and $C_{PQ}[P''] \sqsupseteq\!\!\!\!\sim'_0 C_{PQ}[Q]$.

Since $C_{PQ}[P'']$ has no credits available, the α–transition above must be matched by an α–step of P'' without any preceding or trailing clock transitions, i.e., $C_{PQ}[P''] \xrightarrow{\alpha} C_{PQ}[P']$ for some P' such that $P'' \xrightarrow{\alpha} P'$ and $C_{PQ}[P'] \mathrel{\sqsupset\!\!\!\sim}'_0 C_{PQ}[Q']$. As above we may conclude $P \xrightarrow{\sigma}{}^{i} \xrightarrow{\alpha} P'$, for some P' with $P' \mathrel{\sqsupset\!\!\!\sim}^{\mathrm{aux}}_0 Q'$.

3. $\underline{P \xrightarrow{\sigma} P'}$:
   Hence, $C_{PQ}[P] \xrightarrow{\sigma} C_{PQ}[P'] \,|\, d.\mathbf{0} \xrightarrow{d} C_{PQ}[P']$. Because of $P \mathrel{\sqsupset\!\!\!\sim}^{\mathrm{aux}}_i Q$, the clock transition must be matched by $C_{PQ}[Q] \,|\, \Pi^i d.\mathbf{0} \xrightarrow{\sigma}{}^{k} \widehat{Q}'$ for some $\widehat{Q}'$ and $k{\geq}1{-}i$ such that $C_{PQ}[P'] \,|\, d.\mathbf{0} \mathrel{\sqsupset\!\!\!\sim}'_{i-1+k} \widehat{Q}'$. Due to the construction of the context, $\widehat{Q}' \equiv C_{PQ}[Q''] \,|\, \Pi^{i+k} d.\mathbf{0}$, for some Q'' with $Q \xrightarrow{\sigma}{}^{k} Q''$.
   For matching the d–transition above we know of the existence of $\widehat{Q}'', l$ such that $\widehat{Q}' \xrightarrow{\sigma}{}^{l} \xrightarrow{d} \widehat{Q}''$ and $C_{PQ}[P'] \mathrel{\sqsupset\!\!\!\sim}'_{i-1+k+l} \widehat{Q}''$. Considering the definition of the context, this implies $C_{PQ}[Q''] \,|\, \Pi^{i+k} d.\mathbf{0} \xrightarrow{\sigma}{}^{l} C_{PQ}[Q'] \,|\, \Pi^{i+k+l} d.\mathbf{0} \xrightarrow{d} C_{PQ}[Q'] \,|\, \Pi^{i+k+l-1} d.\mathbf{0}$, for some process Q' satisfying $Q \xrightarrow{\sigma}{}^{k} Q'' \xrightarrow{\sigma}{}^{l} Q'$ and $P' \mathrel{\sqsupset\!\!\!\sim}^{\mathrm{aux}}_{i-1+(k+l)} Q'$. Moreover, $k{+}l \geq 1{-}i$ since $k \geq 1{-}i$.
4. $\underline{Q \xrightarrow{\sigma} Q'}$:
   Hence, $C_{PQ}[Q] \,|\, \Pi^i d.\mathbf{0} \xrightarrow{d}{}^{i} C_{PQ}[Q] \xrightarrow{\sigma} C_{PQ}[Q'] \,|\, d.\mathbf{0} \xrightarrow{d} C_{PQ}[Q']$. As we have $P \mathrel{\sqsupset\!\!\!\sim}^{\mathrm{aux}}_i Q$, the i d–transitions must (essentially) be matched by $C_{PQ}[P] (\xrightarrow{\sigma}\xrightarrow{d})^i C_{PQ}[P'']$, for some P'' which satisfies $P \xrightarrow{\sigma}{}^{i} P''$ and $C_{PQ}[P''] \mathrel{\sqsupset\!\!\!\sim}'_0 C_{PQ}[Q]$. Note that d is a distinguished action and that $C_{PQ}[P]$ has only i credits available. The clock transition above can potentially be matched in two ways:
   (a) $C_{PQ}[P''] \xrightarrow{\sigma} C_{PQ}[P'] \,|\, d.\mathbf{0}$, for a process P' such that $P'' \xrightarrow{\sigma} P'$ and $C_{PQ}[P'] \,|\, d.\mathbf{0} \mathrel{\sqsupset\!\!\!\sim}'_0 C_{PQ}[Q'] \,|\, d.\mathbf{0}$. Due to the lack of credits, the final d–transition above must be matched by $C_{PQ}[P'] \,|\, d.\mathbf{0} \xrightarrow{d} C_{PQ}[P']$ such that $C_{PQ}[P'] \mathrel{\sqsupset\!\!\!\sim}'_0 C_{PQ}[Q']$.
   (b) $C_{PQ}[P''] \mathrel{\sqsupset\!\!\!\sim}'_1 C_{PQ}[Q'] \,|\, d.\mathbf{0}$, i.e., the left–hand side decides to do nothing and thus gain one credit. This credit must be spent immediately when matching the final d–transition above, since d is a distinguished action. Hence, $C_{PQ}[P''] \xrightarrow{\sigma} C_{PQ}[P'] \,|\, d.\mathbf{0} \xrightarrow{d} C_{PQ}[P']$, for some P' with $P'' \xrightarrow{\sigma} P'$ and $C_{PQ}[P'] \mathrel{\sqsupset\!\!\!\sim}'_0 C_{PQ}[Q']$. Note that $C_{PQ}[P'']$ cannot engage in more than one clock transition since it has only a single credit available.

   Summarising, we have shown in both cases the existence of some P' such that $P \xrightarrow{\sigma}{}^{i} \xrightarrow{\sigma} P'$ and, because of $\mathsf{sort}(P') \subseteq \mathsf{sort}(P)$ and $\mathsf{sort}(Q') \subseteq \mathsf{sort}(Q)$, $P' \mathrel{\sqsupset\!\!\!\sim}^{\mathrm{aux}} Q'$. □

Thms. 6 and 11 testify not only to the elegance of the amortised faster–than preorder as a very intuitive faster–than preorder, but also as a unified starting point to approaching faster–than relations on processes.

4 Full Abstraction in TACS

Having identified the largest precongruence contained in the amortised preorder for the sub–calculi TACS^{ut} and TACS^{lt} of TACS, it is natural to investigate the same issue for the full calculus.

For a calculus with must–clock prefixing and urgent actions, Moller and Tofts informally argued in [20] that a precongruence relating bisimulation–equivalent processes cannot satisfy a property one would, at first sight, expect from a faster–than preorder, namely that omitting a must–clock prefix should result in a faster process. This intuition can be backed up by a more general result within our setting, which includes must–clock prefixing and urgent actions, too. Our result is not just based on a specific property; instead, we have a semantic definition of an intuitive faster–than as the coarsest precongruence refining the amortised faster–than preorder, and we will show that this precongruence degrades to a congruence, rather than a proper precongruence. This congruence turns out to be a variant of *timed bisimulation*, whence we start off by recalling the standard definition of timed bisimulation [6, 18] first.

Definition 13 (Timed bisimulation). A relation $\mathcal{R}$ over $\mathcal{P}$ is a *timed bisimulation relation* if, for all $\langle P, Q\rangle \in \mathcal{R}$ and $\alpha \in \mathcal{A}$:

1. $P \xrightarrow{\alpha} P'$ implies $\exists Q'.\, Q \xrightarrow{\alpha} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.
2. $P \xrightarrow{\sigma} P'$ implies $\exists Q'.\, Q \xrightarrow{\sigma} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.
3. $Q \xrightarrow{\alpha} Q'$ implies $\exists P'.\, P \xrightarrow{\alpha} P'$ and $\langle P', Q'\rangle \in \mathcal{R}$.
4. $Q \xrightarrow{\sigma} Q'$ implies $\exists P'.\, P \xrightarrow{\sigma} P'$ and $\langle P', Q'\rangle \in \mathcal{R}$.

We write $P \sim_{\text{t}} Q$ if $\langle P, Q\rangle \in \mathcal{R}$ for some timed bisimulation relation $\mathcal{R}$, and call $\sim_{\text{t}}$ *timed bisimulation.*

It is obvious that timed bisimulation $\sim_{\text{t}}$ is an equivalence and that it refines the amortised faster–than preorder $\sqsupset\!\!\!\sim_0$. However, $\sim_{\text{t}}$ is not a congruence for TACS since it is not compositional for parallel composition. To see this, consider the processes $\underline{a}.\mathbf{0}+\underline{b}.\mathbf{0} \sqsupset\!\!\!\sim_0 \sigma.\underline{a}.\mathbf{0}+\underline{b}.\mathbf{0}$. When putting them in parallel with process $\overline{\underline{b}}.\mathbf{0}$ the relation $\sqsupset\!\!\!\sim_0$ is no longer preserved since $(\underline{a}.\mathbf{0}+\underline{b}.\mathbf{0}) \,|\, \overline{\underline{b}}.\mathbf{0}$ can engage in an a–transition while $(\sigma.\underline{a}.\mathbf{0}+\underline{b}.\mathbf{0}) \,|\, \overline{\underline{b}}.\mathbf{0}$ cannot, as the clock transition that would enable action $\underline{a}$ is preempted by the urgent communication on $\underline{b}$. We thus have to refine timed bisimulation and take initial urgent action sets into account.

Definition 14 (Urgent timed bisimulation). A relation $\mathcal{R}$ over $\mathcal{P}$ is an *urgent timed bisimulation relation* if, for all $\langle P, Q\rangle \in \mathcal{R}$ and $\alpha \in \mathcal{A}$:

1. $P \xrightarrow{\alpha} P'$ implies $\exists Q'.\, Q \xrightarrow{\alpha} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.
2. $P \xrightarrow{\sigma} P'$ implies $\mathcal{U}(Q) \subseteq \mathcal{U}(P)$ and $\exists Q'.\, Q \xrightarrow{\sigma} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.
3. $Q \xrightarrow{\alpha} Q'$ implies $\exists P'.\, P \xrightarrow{\alpha} P'$ and $\langle P', Q'\rangle \in \mathcal{R}$.
4. $Q \xrightarrow{\sigma} Q'$ implies $\mathcal{U}(P) \subseteq \mathcal{U}(Q)$ and $\exists P'.\, P \xrightarrow{\sigma} P'$ and $\langle P', Q'\rangle \in \mathcal{R}$.

We write $P \simeq_{\text{t}} Q$ if $\langle P, Q\rangle \in \mathcal{R}$ for some urgent timed bisimulation relation $\mathcal{R}$, and call $\simeq_{\text{t}}$ *urgent timed bisimulation.*

We have used set inclusion in Conds. (2) and (4) above in analogy to Def. 3. It is important to note the following: if $P \stackrel{\sigma}{\longrightarrow} P'$, then $Q \stackrel{\sigma}{\longrightarrow} Q'$ by Cond. (2), so that Cond. (4) becomes applicable. Therefore, we could just as well require equality of urgent sets in Conds. (2) and (4).

Urgent timed bisimulation is the desired refinement of timed bisimulation, as the following theorem shows.

Theorem 15 (Full abstraction).
Urgent timed bisimulation $\simeq_t$ is the largest congruence *contained in $\sim_t$.*

Proof. The proof follows line–by–line a similar proof in our previous work (cf. Theorem 19 in [16]), where we showed such a statement for a notion of *faster–than precongruence* (the LV–preorder of Def. 3) and *faster–than preorder*. Their definitions coincide with Defs. 13 and 14, respectively, except that they leave out Cond. (4). However Cond. (4) is fully symmetric to Cond. (3) and thus poses no problem for adopting the proof of [16].

We first convince ourselves that $\simeq_{\mathrm{t}}$ is indeed a congruence. All operators of TACS are as in the setting of [16], with exception of the must–clock and lazy–action prefix operators, for which we need to show that $P \simeq_{\mathrm{t}} Q$ implies $\sigma.P \simeq_{\mathrm{t}} \sigma.Q$ and $\alpha.P \simeq_{\mathrm{t}} \alpha.Q$. This is obvious, however, since, e.g., in the first case, the initial clock transition of $\sigma.P$ can be matched by the initial clock transition of $\sigma.Q$, and since no action transitions can be performed.

Establishing that $\simeq_{\mathrm{t}}$ is the *largest* congruence contained in $\sim_{\mathrm{t}}$ now follows exactly the lines of [16]. In a nutshell, because $\simeq_{\mathrm{t}}$ is a congruence contained in $\sim_{\mathrm{t}}$, we have $\simeq_{\mathrm{t}} \subseteq \sim_{\mathrm{t}}^{\mathrm{c}}$. It remains to show that $P \simeq_{\mathrm{t}} Q$, for processes $P, Q \in \mathcal{P}$, whenever $C[P] \sim_{\mathrm{t}} C[Q]$ for all TACS contexts $C[\_]$. To do so, it suffices to consider the relation

$$\sim_{\mathrm{t}}^{\mathrm{aux}} =_{\mathrm{df}} \{\langle P, Q\rangle \mid C_{\mathcal{L}}[P] \sim_{\mathrm{t}} C_{\mathcal{L}}[Q] \text{ for some finite } \mathcal{L} \supseteq \mathsf{sort}(P) \cup \mathsf{sort}(Q)\} .$$

Here, $C_{\mathcal{L}}[x] =_{\mathrm{df}} x \mid H_{\mathcal{L}}$ and $H_{\mathcal{L}} =_{\mathrm{df}} \mu x.(\underline{e}.\mathbf{0} + \sum\{\underline{\tau}.(\sum_{d \in L} \underline{d}.\mathbf{0} + \underline{d_L}.x) \mid L \subseteq \overline{\mathcal{L}}\})$. Note that $H_{\mathcal{L}}$ is well–defined due to the finiteness of $\mathcal{L}$. The actions $\underline{e}$ and $\underline{d_L}$ and their complements are taken to be 'fresh' actions not in the sorts of P and Q. The proof now proceeds as in [16] by establishing that $\sim_{\mathrm{t}}^{\mathrm{aux}}$ is an urgent timed bisimulation relation. □

We can now state and prove the main result of this section.

Theorem 16 (Full abstraction in TACS).
Urgent timed bisimulation $\simeq_t$ is the largest (pre–)congruence *contained in $\sqsupseteq\!\!\!\!\sim_0$.*

Proof. By Thms. 15 and 2, it is sufficient to show that $\sqsupset\!\!\!\sim_0^{\mathrm{c}} \subseteq \sim_{\mathrm{t}} \subseteq \sqsupset\!\!\!\sim_0$. Since the inclusion $\sim_{\mathrm{t}} \subseteq \sqsupset\!\!\!\sim_0$ immediately follows from Defs. 13 and 1, it remains to show that $\sqsupset\!\!\!\sim_0^{\mathrm{c}} \subseteq \sim_{\mathrm{t}}$. To do so, we prove that $\simeq^{\mathrm{aux}} =_{\mathrm{df}} \{\langle P, Q\rangle \mid C_{PQ}[P] \sqsupset\!\!\!\sim_0 C_{PQ}[Q]\}$, where $C_{PQ}[\_] =_{\mathrm{df}} (\_ \mid H_{PQ}) \setminus \mathsf{sort}(PQ)$, $H_{PQ} =_{\mathrm{df}} \mu x. \sum_{\underline{a} \in \mathsf{sort}(PQ)} \overline{\underline{a}}.(\underline{\tau}.x + \underline{d_a}.x) + \underline{\tau}.(\underline{d}.x + \sigma.\underline{\tau}.x)$, and $\mathsf{sort}(PQ) =_{\mathrm{df}} \mathsf{sort}(P) \cup \mathsf{sort}(Q)$ with $\underline{d_a}, \underline{d} \notin \mathsf{sort}(PQ)$, is a timed bisimulation relation. (Again, $(\sqsupset\!\!\!\sim_0)^{\mathrm{c}} \subseteq \simeq^{\mathrm{aux}}$ is obvious.) Let $P \simeq^{\mathrm{aux}} Q$ be arbitrary; we consider the following cases:

1. $\underline{P \xrightarrow{a} P'}$:
   Hence, $C_{PQ}[P] \xrightarrow{\tau} (P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}_{\underline{a}}.H_{PQ})) \setminus \mathsf{sort}(PQ) \xrightarrow{\tau} C_{PQ}[P']$. Since $C_{PQ}[P] \underset{\sim}{\sqsupset}_0 C_{PQ}[Q]$, there exist $\widehat{Q}'', k, l$ such that $C_{PQ}[Q] \xrightarrow{\sigma}{}^{k} \xrightarrow{\tau} \xrightarrow{\sigma}{}^{l} \widehat{Q}''$ and $(P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}_{\underline{a}}.H_{PQ})) \setminus \mathsf{sort}(PQ) \underset{\sim}{\sqsupset}_{0+k+l} \widehat{Q}''$. Due to the placement of urgent $\underline{\tau}$'s in the context and the fact that $\underline{d}_{\underline{a}}$ is a distinguished action, we conclude $k{=}l{=}0$ and $Q \xrightarrow{a} Q'$, for some Q' with $\widehat{Q}'' \equiv (Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}_{\underline{a}}.H_{PQ})) \setminus \mathsf{sort}(PQ)$.
   For matching the τ–transition of $(P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d}_{\underline{a}}.H_{PQ})) \setminus \mathsf{sort}(PQ)$ above we similarly know of the existence of $\widehat{Q}', k', l'$ such that $\widehat{Q}'' \xrightarrow{\sigma}{}^{k'} \xrightarrow{\tau} \xrightarrow{\sigma}{}^{l'} \widehat{Q}'$ and $C_{PQ}[P'] \underset{\sim}{\sqsupset}_0 \widehat{Q}'$. Again, because of the placement of urgent $\underline{\tau}$'s in the context and the fact that $\underline{d}_{\underline{a}}$ is a distinguished action, we may infer $k'{=}l'{=}0$ and $\widehat{Q}' \equiv C_{PQ}[Q']$.
   Further, $C_{PQ}[P'] \underset{\sim}{\sqsupset}_0 C_{PQ}[Q']$ implies $C_{P'Q'}[P'] \underset{\sim}{\sqsupset}_0 C_{P'Q'}[Q']$, when considering $\mathsf{sort}(P') \subseteq \mathsf{sort}(P)$, $\mathsf{sort}(Q') \subseteq \mathsf{sort}(Q)$, and the construction of the context. Note that additional summands in H_{PQ} cannot influence transitions due to the restriction. Hence, we have shown the existence of a Q' satisfying $Q \xrightarrow{a} Q'$ and $P' \simeq^{\mathrm{aux}} Q'$.
2. $\underline{P \xrightarrow{\tau} P'}$:
   Hence, $C_{PQ}[P] \xrightarrow{\tau} C_{PQ}[P']$. Since the context has an urgent $\underline{\tau}$ enabled and since $\underline{d}$ and the $\underline{d}_{\underline{a}}$ are distinguished actions, the premise $P \simeq^{\mathrm{aux}} Q$ implies that $C_{PQ}[Q] \xrightarrow{\tau} C_{PQ}[Q']$ and $C_{PQ}[P'] \underset{\sim}{\sqsupset}_0 C_{PQ}[Q']$, for some Q' with $Q \xrightarrow{\tau} Q'$. As above, $C_{PQ}[P'] \underset{\sim}{\sqsupset}_0 C_{PQ}[Q']$ implies $P' \simeq^{\mathrm{aux}} Q'$.
3. $\underline{P \xrightarrow{\sigma} P'}$:
   Hence, $C_{PQ}[P] \xrightarrow{\tau} (P \,|\, (\underline{d}.H_{PQ} + \sigma.\underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ) \xrightarrow{\sigma} (P' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ) \xrightarrow{\tau} C_{PQ}[P']$ by our operational rules.
   Because of $C_{PQ}[P] \underset{\sim}{\sqsupset}_0 C_{PQ}[Q]$ we know of the existence of $\widehat{Q}'', k, l$ satisfying $C_{PQ}[Q] \xrightarrow{\sigma}{}^{k} \xrightarrow{\tau} \xrightarrow{\sigma}{}^{l} \widehat{Q}''$ and $(P \,|\, (\underline{d}.H_{PQ} + \sigma.\underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ) \underset{\sim}{\sqsupset}_{0+k+l} \widehat{Q}''$. Taking into account the urgent $\underline{\tau}$'s initially enabled by the context and after the context's σ–prefix (and action $\underline{d}$), we may infer $k = 0$ and $l \in \{0, 1\}$, respectively. For matching the above second step, i.e., the clock–transition, we distinguish the cases $l = 1$ and $l = 0$.
   (a) *Case* $l = 1$*:* Here, $Q \xrightarrow{\sigma} Q'$ and $\widehat{Q}'' \equiv (Q' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ)$ for some Q'.
       While $\widehat{Q}''$ cannot match the second step of the left hand–side, i.e., the clock transition, there is one credit available. Hence, $(P' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ) \underset{\sim}{\sqsupset}_0 (Q' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ)$.
   (b) *Case* $l = 0$*:* Here, $\widehat{Q}'' \equiv (Q \,|\, (\underline{d}.H_{PQ} + \sigma.\underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ)$.
       The matching of the clock transition of the left–hand side implies the existence of $\widehat{Q}'''$ and l''' such that $\widehat{Q}'' \xrightarrow{\sigma}{}^{l'''} \widehat{Q}'''$ and $(P' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ) \underset{\sim}{\sqsupset}_{l'''} \widehat{Q}'''$. Since the context has an urgent $\underline{\tau}$–action enabled after performing its σ–prefix we may further conclude $l''' = 1$. Hence, $Q \xrightarrow{\sigma} Q'$ and $\widehat{Q}''' \equiv (Q' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ)$ for a suit-

able Q' and again $(P' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ) \sqsupseteq\!\!\!\sim_0 (Q' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ)$.

When matching the third step above, i.e., the second τ–transition, and since $(P' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ) \sqsupseteq\!\!\!\sim_0 (Q' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ)$, we may infer $\widehat{Q}', k', l'$ such that $(Q' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ) \stackrel{\sigma}{\longrightarrow}^{k'} \stackrel{\tau}{\longrightarrow} \stackrel{\sigma}{\longrightarrow}^{l'} \widehat{Q}'$ and $C_{PQ}[P'] \sqsupseteq\!\!\!\sim_0 \widehat{Q}'$. Note that the placement of urgent $\underline{\tau}$'s in the context necessarily implies $k'{=}l'{=}0$, whence $\widehat{Q}' \equiv C_{PQ}[Q']$ due to action $\underline{d}$. Summarising, we have established the existence of a Q' such that $Q \stackrel{\sigma}{\longrightarrow} Q'$ and $P' \simeq^{\mathrm{aux}} Q'$.

4. $\underline{Q \stackrel{a}{\longrightarrow} Q'}$:
   Hence, $C_{PQ}[Q] \stackrel{\tau}{\longrightarrow} (Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d_a}.H_{PQ})) \setminus \mathsf{sort}(PQ) \stackrel{\tau}{\longrightarrow} C_{PQ}[Q']$. Since $C_{PQ}[P] \sqsupseteq\!\!\!\sim_0 C_{PQ}[Q]$ and thus no credits are available to $C_{PQ}[P]$, we have $C_{PQ}[P] \stackrel{\tau}{\longrightarrow} \widehat{P}''$ and $\widehat{P}'' \sqsupseteq\!\!\!\sim_0 (Q' \,|\, (\underline{\tau}.H_{PQ} + \underline{d_a}.H_{PQ})) \setminus \mathsf{sort}(PQ)$. Because Q's move to Q' enables the distinguished action $\underline{d_a}$ within the context, we may further infer $\widehat{P}'' \equiv (P' \,|\, (\underline{\tau}.H_{PQ} + \underline{d_a}.H_{PQ})) \setminus \mathsf{sort}(PQ)$, for some P' with $P \stackrel{a}{\longrightarrow} P'$.
   The second step of the right–hand side, i.e., the τ–transition, can only be matched as follows: $\widehat{P}'' \stackrel{\tau}{\longrightarrow} \widehat{P}'$ and $\widehat{P}' \sqsupseteq\!\!\!\sim_0 C_{PQ}[Q']$, where $\widehat{P}' \equiv C_{PQ}[P']$. This is due to the fact that no credits are available to $\widehat{P}''$ and that $\underline{d_a}$ is a distinguished action. Summarising, we have established the existence of a P' such that $P \stackrel{a}{\longrightarrow} P'$ and $P' \simeq^{\mathrm{aux}} Q'$.
5. $\underline{Q \stackrel{\tau}{\longrightarrow} Q'}$:
   Hence, $C_{PQ}[Q] \stackrel{\tau}{\longrightarrow} C_{PQ}[Q']$. Because $C_{PQ}[P] \sqsupseteq\!\!\!\sim_0 C_{PQ}[Q]$, i.e., no credits are available to $C_{PQ}[P]$, and because $\underline{d}, \underline{d_a}$ are distinguished actions, we know of the existence of some $\widehat{P}'$ satisfying $C_{PQ}[P] \stackrel{\tau}{\longrightarrow} \widehat{P}'$, $\widehat{P}' \sqsupseteq\!\!\!\sim_0 C_{PQ}[Q']$, and $\widehat{P}' \equiv C_{PQ}[P']$ for some P' such that $P \stackrel{\tau}{\longrightarrow} P'$. Thus, $P' \simeq^{\mathrm{aux}} Q'$, too.
6. $\underline{Q \stackrel{\sigma}{\longrightarrow} Q'}$:
   Hence, $C_{PQ}[Q] \stackrel{\tau}{\longrightarrow} (Q \,|\, (\underline{d}.H_{PQ} + \sigma.\underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ) \stackrel{\sigma}{\longrightarrow} (Q' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ) \stackrel{\tau}{\longrightarrow} C_{PQ}[Q']$.
   Since $C_{PQ}[P] \sqsupseteq\!\!\!\sim_0 C_{PQ}[Q]$, i.e., $C_{PQ}[P]$ has no credits available, we know of the existence of a $\widehat{P}'''$ such that $C_{PQ}[P] \stackrel{\tau}{\longrightarrow} \widehat{P}'''$ and $\widehat{P}''' \sqsupseteq\!\!\!\sim_0 (Q \,|\, (\underline{d}.H_{PQ} + \sigma.\underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ)$. Moreover, $\widehat{P}''' \equiv (P \,|\, (\underline{d}.H_{PQ} + \sigma.\underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ)$ since $\underline{d}$ is a distinguished action.
   For matching the second step of $C_{PQ}[Q]$ above, i.e., the clock transition, the following two possibilities arise:
   (a) The left–hand side process does nothing and gains one credit, whence $\widehat{P''} =_{\mathrm{df}} (P \,|\, (\underline{d}.H_{PQ} + \sigma.\underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ) \sqsupseteq\!\!\!\sim_1 (Q' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ)$.
   When matching the third step of $C_{PQ}[Q]$ above and noting that $\underline{d}$ is a distinguished action and that one credit is available, we obtain some $\widehat{P}'$ satisfying $\widehat{P}'' \stackrel{\sigma}{\longrightarrow} \stackrel{\tau}{\longrightarrow} \widehat{P}'$, $\widehat{P}' \sqsupseteq\!\!\!\sim_0 C_{PQ}[Q]$, and $\widehat{P}' \equiv C_{PQ}[P']$, for a P' such that $P \stackrel{\sigma}{\longrightarrow} P'$.

(b) The left–hand side engages in a clock transition, too, whence $\widehat{P}''' \xrightarrow{\sigma} \widehat{P}'' \equiv (P' \,|\, (\underline{d}.H_{PQ} + \underline{\tau}.H_{PQ})) \setminus \mathsf{sort}(PQ)$, for a P' such that $P \xrightarrow{\sigma} P'$. Matching the third step of $C_{PQ}[Q]$ above and observing that the left–hand side has still no credit available, there must exist a $\widehat{P}'$ satisfying $\widehat{P}'' \xrightarrow{\tau} \widehat{P}'$, $\widehat{P}' \sqsupseteq\!\!\!\!\sim_0 C_{PQ}[Q]$, and $\widehat{P}' \equiv C_{PQ}[P']$, for some P' with $P \xrightarrow{\sigma} P'$.

Summarising, we have established in both cases the existence of a P' such that $P \xrightarrow{\sigma} P'$ and $P' \simeq^{\text{aux}} Q'$. □

Hence, not all faster–than settings on the basis of the amortised faster–than preorder admit a faster–than precongruence that is a proper precongruence. As shown for the full TACS calculus, it is possible that the obtained fully–abstract precongruence degrades to a congruence.

5 Discussion

The aim of this section is to investigate when exactly the amortised faster–than preorder, when closed under all contexts, collapses from a proper precongruence to a congruence.

We have shown in the TACS sub–calculus with only must–clock prefixing and lazy actions (cf. Sec. 3.1) and in the sub–calculus with only can–clock prefixing and urgent actions (cf. Sec. 3.2) that indeed proper precongruences are obtained: the MT–preorder and the LV–preorder, respectively. However, when combining both clock prefixes as well as lazy and urgent actions, then the result is a congruence: urgent timed bisimulation (cf. Sec. 4). We desire to explore where exactly this borderline lies, by charaterising the largest precongruence contained in the amortised faster–than preorder for other combinations of can–/must–clock prefixes as well as urgent/lazy actions. While some of the resulting settings might not appear natural, others are clearly practically relevant, and this will be pointed out when analysing each combination in turn.

5.1 Can–Clock Prefixing and Urgent+Lazy Actions

Here we find ourselves in the sub–calculus TACS^{ut} investigated in Sec. 3.1, where additionally lazy actions may be present. Lazy actions might be used for modelling the potential of errors: many errors in practice can occur at any moment and thus cannot be associated with maximal delays.

Corollary 17 (Full–abstraction in the can/urgent+lazy setting). *The LV–preorder* $\sqsupseteq\!\!\!\!\sim_{lv}$ *is the* largest precongruence *contained in* $\sqsupseteq\!\!\!\!\sim_0$*, when considering* TACS *processes with can–clock prefixes only.*

Hence, Thm. 6 of Sec. 3.1 remains valid in the presence of lazy actions. This can be seen by checking the proof of Thm. 6 as well as all the proofs of [16] on which it depends.

5.2 Must– and Can–Clock Prefixing and Lazy Actions

The setting here is the one of TACS$^{\mathrm{lt}}$, where can–clock prefixes are added. This does not change the result we obtained for the TACS$^{\mathrm{lt}}$ setting (cf. Thm. 11 in Sec. 3.2), when extending the definition of the MT–preorder $\sqsupseteq_{\mathrm{mt}}$ (cf. Def. 7) from processes in $\mathcal{P}^{\mathrm{lt}}$ to the class of processes considered here.

Theorem 18 (Full abstraction in the must+can/lazy setting). *The MT–preorder* $\sqsupseteq_{mt}$ *is the* largest precongruence *contained in* $\sqsupseteq_0$*, when considering* TACS *processes with lazy actions only.*

This statement can be deduced by inspecting the proofs of Sec. 3.2, i.e., the proof of Thm. 11 and the proofs of the underlying statements adopted from [15], in the presence of $\underline{\sigma}$–prefixes. The only parts that are not straightforward concern checking whether the MT–preorder $\sqsupseteq_{\mathrm{mt}}$ is also compositional for can–clock prefixes and whether the commutation lemma, Lemma 8, still holds. To do so we first need to adapt the syntactic faster–than preorder $\succ$ of [15] by adding the clause $P \succ \underline{\sigma}.P$.

Definition 19 (Syntactic Faster–Than Preorder). The relation $\succ \subseteq \widehat{\mathcal{P}} \times \widehat{\mathcal{P}}$ is defined as the smallest relation satisfying the following properties, for all $P, P', Q, Q' \in \widehat{\mathcal{P}}$.

$$\begin{array}{rll}
\text{Always:} & (1)\ P \succ P & (2)\ \ (a)\ P \succ \sigma.P \text{ and } (b)\ P \succ \underline{\sigma}.P \\
P' \succ P,\ Q' \succ Q: & (3)\ P'|Q' \succ P|Q & (4)\ \ P' + Q' \succ P + Q \\
 & (5)\ P' \setminus L \succ P \setminus L & (6)\ \ P'[f] \succ P[f] \\
P' \succ P,\ x \text{ guarded}: & (7)\ P'[\mu x.\, P/x] \succ \mu x.\, P &
\end{array}$$

This syntactic faster–than relation possesses the following important property which is adopted from Lemma 5(2) of the full version of [15] and also used in the next section.

Lemma 20. *For any* P, P'*, if* $P \xrightarrow{\sigma} P'$ *then* $P' \succ P$*.*

The proof of this lemma is by a straightforward induction on the structure of P. Also the other parts of Lemma 5 of the full version of [15] hold under the modified syntactic faster–than preorder, in particular $P' \succ P$ implies $P' \sqsupseteq_{\mathrm{mt}} P$ for *processes* P', P in the TACS fragment we consider in this subsection. For the proof of Lemma 5 it is important, that these processes satisfy the *laziness property*, i.e., each of them can perform a time step. We can now prove that the MT–preorder is compositional for can–clock prefixes, in the TACS sub–calculus that is restricted to lazy actions only.

Lemma 21. *Let* P, Q *be* TACS *processes with lazy actions only. Then* $P \sqsupseteq_{mt} Q$ *implies* $\underline{\sigma}.P \sqsupseteq_{mt} \underline{\sigma}.Q$*.*

Proof. The only nontrivial case concerns $\underline{\sigma}.P \xrightarrow{\alpha} P'$ for some action α and process P. By our operational rules we know that this can only be the case if $P \xrightarrow{\alpha} P'$. Since $P \mathrel{\underset{\sim}{\sqsupset}}_{\mathrm{mt}} Q$, there exists some Q', P'', k such that $Q \xrightarrow{\sigma}{}^{k} \xrightarrow{\alpha} Q'$, $P' \xrightarrow{\sigma}{}^{k} P''$ and $P'' \mathrel{\underset{\sim}{\sqsupset}}_{\mathrm{mt}} Q'$. Hence, $\underline{\sigma}.Q \xrightarrow{\sigma}{}^{k+1} \xrightarrow{\alpha} Q'$. Further, due to the laziness property, there exists a process P''' such that $P'' \xrightarrow{\sigma} P'''$. As seen above (cf. Lemma 20 and the property that $R' \succ R$ implies $R' \mathrel{\underset{\sim}{\sqsupset}}_{\mathrm{mt}} R$ for any processes R', R), this implies $P''' \mathrel{\underset{\sim}{\sqsupset}}_{\mathrm{mt}} P''$. Hence, $P' \xrightarrow{\sigma}{}^{k+1} P'''$ and, by transitivity, $P''' \mathrel{\underset{\sim}{\sqsupset}}_{\mathrm{mt}} Q'$. □

Moreover, since the correctness of the commutation lemma is only based on Lemma 5 of the full version of [15], the laziness property as well as the time–determinism property, the commutation lemma obviously remains valid even in the presence of can–clock prefixing.

5.3 Can–Clock Prefixing and Lazy Actions

This combination is one that does not appear to be intuitive. If every action can delay its execution, additional potential delays specified by can–clock prefixes seem irrelevant and can be omitted (cf. Prop. 22). Further, if every delay specified by a clock prefix can indeed be omitted, then it appears that delays are not relevant at all and may thus be safely ignored (cf. Thm. 24).

Proposition 22. *$P \sim_t \underline{\sigma}.P$ for all* TACS *processes P with can–clock prefixes and lazy actions only.*

Proof. Since $P \succ \underline{\sigma}.P$ according to Def. 19(2b), it is sufficient to show that $\succ$, when restricted to processes, is a timed bisimulation relation. This is done by induction on the length of inference over $\succ$. The only interesting case concerns $P \succ \underline{\sigma}.P$:

1. $P \xrightarrow{\alpha} P'$ implies $\underline{\sigma}.P \xrightarrow{\alpha} P'$ by Rule (uPre) and $P' \succ P'$ by Def. 19(1).
2. $\underline{\sigma}.P \xrightarrow{\alpha} P'$ implies $P \xrightarrow{\alpha} P'$ by Rule (uPre) and $P' \succ P'$ by Def. 19(1).
3. $P \xrightarrow{\sigma} P'$ implies $\underline{\sigma}.P \xrightarrow{\sigma} P$ by Rule (tuPre) and $P' \succ P$ by Lemma 20.
4. $\underline{\sigma}.P \xrightarrow{\sigma} P$. Because of the laziness property, there exists some P' with $P \xrightarrow{\sigma} P'$. By applying Lemma 20 we obtain $P' \succ P$.

All other cases only involve a straightforward application of the induction hypothesis. □

Because of the irrelevance of timed behaviour, timed bisimulation $\sim_{\mathrm{t}}$ coincides with standard bisimulation $\sim$ [18] — where clock transitions are ignored — in the setting considered in this section.

Lemma 23. $\sim\, = \,\sim_t$ *on* TACS *processes P with can–clock prefixes and lazy actions only.*

Proof. The proof of the non–trivial inclusion "$P \sim Q$ implies $P \sim_{\mathrm{t}} Q$" is straightforward when first stripping the processes P, Q off their σ–prefixes while preserving timed bisimulation, and thus standard bisimulation, according to Prop. 22. $\square$

As expected, the amortised faster–than preorder, when closed under all contexts, degrades to standard bisimulation in this setting.

Theorem 24 (Full abstraction in the can/lazy setting).
Standard bisimulation $\sim$ *is the* largest precongruence *contained in* $\sqsupset\!\!\!\!\sim_0$*, when considering* TACS *processes with can–clock prefixes and lazy actions only.*

Proof. The inclusion $\sim \subseteq \sqsupset\!\!\!\!\sim_0^{\mathrm{c}}$ is obvious since $\sim = \sim_{\mathrm{t}} \subseteq \sqsupset\!\!\!\!\sim_0$ and since $\sim = \sim_{\mathrm{t}}$ is a (pre–)congruence. To prove the inverse inclusion $\sqsupset\!\!\!\!\sim_0^{\mathrm{c}} \subseteq \sim$ we establish the stronger statement $\sqsupset\!\!\!\!\sim_0 \subseteq \sim$.

Let $P \sqsupset\!\!\!\!\sim_0 Q$. Because of Prop. 22 we can remove all $\underline{\sigma}$–prefixes of P and Q to obtain $\widehat{P} \sim_{\mathrm{t}} P$ and $\widehat{Q} \sim_{\mathrm{t}} Q$, respectively, implying $\widehat{P} \sqsupset\!\!\!\!\sim_0 \widehat{Q}$. Now, $\widehat{P} \sim_{\mathrm{t}} \widehat{Q}$ follows since $\bigcup_{i \in \mathbb{N}} \sqsupset\!\!\!\!\sim_i$ is a timed bisimulation relation on processes without can–clock prefixes and only lazy actions; this property is straightforward since time steps are always possible and do not change process terms other than unfolding recursion.

Summarising, we have $P \sim_{\mathrm{t}} \widehat{P} \sim_{\mathrm{t}} \widehat{Q} \sim_{\mathrm{t}} Q$, i.e., $P \sim_{\mathrm{t}} Q$. Lemma 23 now yields $P \sim Q$, as desired. $\square$

To conclude, it should be noted that Prop. 22 does not hold in the presence of must–clock prefixes. For example, $\underline{\sigma}.\sigma.a.\mathbf{0} \xrightarrow{\sigma} \sigma.a.\mathbf{0}$ and $\sigma.a.\mathbf{0} \xrightarrow{\sigma} a.\mathbf{0}$, but obviously $\sigma.a.\mathbf{0} \not\sim a.\mathbf{0}$.

5.4 Must–Clock Prefixing and Urgent Actions, & More

For the full algebra TACS, we have shown in Sec. 4 that the largest precongruence contained in the amortised faster–than preorder is urgent timed bisimulation (cf. Thm. 16). Full TACS combines must– and can–clock prefixing with lazy and urgent actions. When leaving out either lazy actions, or can–clock prefixes, or both, the result remains valid, as can be checked by inspecting the proofs of Sec. 4. Essentially, the reason is that the context constructed within this proof uses neither lazy actions nor can–clock prefixes.

Most interesting is the case when we are left with must–clock prefixing and urgent actions only. This setting coincides with the one of Hennessy and Regan's well–known *Timed Process Language* [13], TPL, in terms of both syntax and operational semantics, when leaving out TPL's timeout operator; we refer to this calculus as TPL$^-$. It is important to note that, for TPL$^-$, urgent timed bisimulation is the same as timed bisimulation; this is because all actions are urgent, and the bisimulation conditions on actions imply that equivalent processes have the same initial (urgent) actions.

However, adding either can–clock prefixing or lazy actions to TPL$^-$ leads to a more expressive calculus than TPL$^-$. For example, the process $\underline{\sigma}.\underline{\tau}.P$ in the setting must+can–clock prefixing and urgent actions can engage in both a clock transition and a τ–transition, and the same applies to process $\tau.P$. This semantic behaviour is incompatible with the maximal–progress property in TPL$^-$, and indeed in full TPL, bearing in mind that every action is urgent.

6 Related Work

Relatively little work has been published on theories that relate processes with respect to speed. This is somewhat surprising, given the wealth of literature on timed process algebras [6] and the importance of reasoning about time efficiency in system design [17].

Early research on process efficiency compares untimed CCS–like terms by counting internal actions either within a testing–based [21] or a bisimulation–based [3, 4] setting. Due to interleaving, e.g., $(\tau.a.\mathbf{0} \,|\, \tau.\overline{a}.b.\mathbf{0}) \setminus \{a\}$ is considered to be as efficient as $\tau.\tau.\tau.b.\mathbf{0}$, whereas $(\sigma.a.\mathbf{0} \,|\, \sigma.\overline{a}.b.\mathbf{0}) \setminus \{a\}$ $((\underline{\sigma}.\underline{a}.\mathbf{0} \,|\, \underline{\sigma}.\underline{\overline{a}}.\underline{b}.\mathbf{0}) \setminus \{\underline{a}\})$ is strictly faster than $\sigma.\sigma.\tau.b.\mathbf{0}$ $(\underline{\sigma}.\underline{\sigma}.\underline{\tau}.\underline{b}.\mathbf{0})$ in our setting.

The most closely related research to ours is obviously the one by Moller and Tofts on processes equipped with lower time bounds [20] and our own on processes equipped with upper time bounds [16]. The work of Moller and Tofts has recently been revisited by us [15] and completed by adding an axiomatisation for finite processes, a full–abstraction result, and a "weak" variant of the MT–preorder that abstracts from the unobservable action τ. Our work on upper time bounds [16] features similar results for the LV–preorder. In both papers [15, 16], the chosen reference preorders for the full–abstraction results are less abstract than the amortised faster–than preorder advocated here. Although a couple of these reference preorders borrowed some idea of amortisation (cf. Defs. 4 and 9), they were somewhat tweaked to fit the LV–preorder and the MT–preorder, respectively. Thus, Thms. 6 and 11 are indeed significant generalisations of the corresponding theorems in [16] and in [15] (cf. Thms. 5 and 10), respectively.

Most other published work on faster–than relations has focused on settings with upper time bounds and on preorders based on De Nicola and Hennessy's testing theory [11]. Initially, research was conducted within the setting of Petri nets [23, 24], and later for the Theoretical–CSP–style process algebra PAFAS [9]. An attractive feature when adopting testing semantics is a fundamental result stating that the considered faster–than testing preorder based on continuous–time semantics coincides with the analogous testing preorder based on discrete–time semantics [24]. It remains to be seen whether a similar result holds for our bisimulation–based approach.

Last, but not least, Corradini et al. [10] have introduced the *ill–timed–but–well–caused* approach for relating processes with respect to speed [2, 12]. This approach allows system components to attach local time stamps to actions. However, as a byproduct of interleaving semantics, local time stamps may decrease within action sequences exhibited by concurrent processes. The presence of these "ill–timed" runs makes it difficult to relate the faster–than preorder of Corradini et al. to ours.

7 Conclusions and Future Work

In this paper we proposed a general amortised faster–than preorder for unifying bisimulation–based process theories [15, 16, 20] that relate asynchronous processes with respect to speed. Our amortised preorder ensures that a faster process must execute each action no later than the related slower process does, while both processes must be functionally equivalent in the sense of strong bisimulation [18].

Since the amortised faster–than preorder is normally not closed under all system contexts, we characterised the largest precongruences contained in it for a range of settings. The chosen range is spanned by a two–dimensional space, with one axis indicating whether only must–clock prefixes, only can–clock prefixes, or both are permitted, and the other axis determining whether only lazy actions, only urgent actions, or both kinds of actions are available. In this space, the settings of Moller/Tofts [20], which is concerned with lower time bounds, and of Lüttgen/Vogler [16], which is concerned with upper time bounds, can be recognised as "must/lazy" and "can/urgent" combinations, respectively. Since all reference preorders chosen in [15, 16] are less abstract than the amortised faster–than preorder, the results of this paper strengthen the ones obtained for both the Moller/Tofts and the Lüttgen/Vogler approach. The following table summarises our findings for each combination of clock prefix and action type, i.e., each entry identifies the behavioural relation that characterises the largest precongruence contained in the amortised faster–than preorder.

	Lazy	*Urgent*	*Lazy+Urgent*
Must	MT–preorder	Timed bisimulation	Urgent timed bisimulation
Can	Bisimulation	LV–preorder	LV–preorder
Must+Can	MT–preorder	Urgent timed bisimulation	Urgent timed bisimulation

The table shows that the amortised faster–than relation degrades to timed bisimulation as soon as must–clock prefixes and urgent actions come together. In this case, which includes the established process algebra TPL [13], one may express time intervals by equipping actions with both lower and upper time bounds. Moreover, when extending the Moller/Tofts approach by can–clock prefixing or the Lüttgen/Vogler approach by lazy actions, the MT–preorder and the LV–preorder, respectively, remain fully–abstract.

Future work shall investigate decision procedures for the MT–preorder and the LV–preorder, respectively, in order for them to be implemented in automated verification tools, such as the *Concurrency Workbench NC* [8]. This is of particular interest since bisimulation semantics lends itself to more efficient algorithms than testing semantics [1], bearing in mind that most related work on faster–than relations had focused on testing–based, rather than bisimulation–based, preorders.

References

[1] S. Abramsky. Observation equivalence as a testing equivalence. *TCS*, 53:225–241, 1987.

[2] L. Aceto and D. Murphy. Timing and causality in process algebra. *Acta Inform.*, 33(4):317–350, 1996.

[3] S. Arun-Kumar and M.C.B. Hennessy. An efficiency preorder for processes. *Acta Inform.*, 29(8):737–760, 1992.

[4] S. Arun-Kumar and V. Natarajan. Conformance: A precongruence close to bisimilarity. In *STRICT '95*, Workshops in Comp., pp. 55–68. Springer-Verlag, 1995.

[5] E. Badouel and P. Darondeau. On guarded recursion. *TCS*, 82(2):403–408, 1991.

[6] J.C.M. Baeten and C.A. Middelburg. *Process Algebra with Timing*. Monographs in Theoret. Comp. Sc. Springer-Verlag, 2002.

[7] J.A. Bergstra, A. Ponse, and S.A. Smolka, eds. *Handbook of Process Algebra*. Elsevier Science, 2001.

[8] R. Cleaveland and S. Sims. The NCSU Concurrency Workbench. In *CAV '96*, vol. 1102 of *LNCS*, pp. 394–397. Springer-Verlag, 1996.

[9] F. Corradini, M. Di Berardini, and W. Vogler. PAFAS at work: Comparing the worst-case efficiency of three buffer implementations. In *APAQS 2001*, pp. 231–240. IEEE Computer Society Press, 2001.

[10] F. Corradini, R. Gorrieri, and M. Roccetti. Performance preorder and competitive equivalence. *Acta Inform.*, 34(11):805–835, 1997.

[11] R. De Nicola and M.C.B. Hennessy. Testing equivalences for processes. *TCS*, 34(1-2):83–133, 1984.

[12] R. Gorrieri, M. Roccetti, and E. Stancampiano. A theory of processes with durational actions. *TCS*, 140(1):73–94, 1995.

[13] M.C.B. Hennessy and T. Regan. A process algebra for timed systems. *Inform. and Comp.*, 117(2):221–239, 1995.

[14] L. Jenner and W. Vogler. Fast asynchronous systems in dense time. *TCS*, 254(1-2):379–422, 2001.

[15] G. Lüttgen and W. Vogler. Bisimulation on speed: Lower time bounds. In *FOSSACS 2004*, vol. 2987 of *LNCS*, pp. 333–347. Springer-Verlag, 2004. Full version available as TR-2004-1 at Universität Augsburg, Germany.

[16] G. Lüttgen and W. Vogler. Bisimulation on speed: Worst–case efficiency. *Inform. and Comp.*, 191(2):105–144, 2004.

[17] N. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, 1996.

[18] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[19] F. Moller and C. Tofts. A temporal calculus of communicating systems. In *CONCUR '90*, vol. 458 of *LNCS*, pp. 401–415. Springer-Verlag, 1990.

[20] F. Moller and C. Tofts. Relating processes with respect to speed. In *CONCUR '91*, vol. 527 of *LNCS*, pp. 424–438. Springer-Verlag, 1991.

[21] V. Natarajan and R. Cleaveland. An algebraic theory of process efficiency. In *LICS '96*, pp. 63–72. IEEE Computer Society Press, 1996.

[22] C. Verhoef. A congruence theorem for structured operational semantics with predicates and negative premises. *Nordic J. of Comp.*, 2(2):274–302, 1995.

[23] W. Vogler. Efficiency of asynchronous systems, read arcs, and the MUTEX-problem. *TCS*, 275(1–2):589–631, 2002.

[24] W. Vogler. Faster asynchronous systems. *Inform. and Comp.*, 184(2):311–342, 2003.