A Temporal Logic for Modelling Activities of Daily Living

Kließ, Malte S.; Jonker, Catholijn M.; van Riemsdijk, M. Birna

DOI
10.4230/LIPIcs.TIME.2018.17

Publication date
2018

Document Version
Final published version

Published in
25th International Symposium on Temporal Representation and Reasoning, TIME 2018

A Temporal Logic for Modelling Activities of Daily Living

Malte S. Kließ
Interactive Intelligence, Delft University of Technology, Delft, The Netherlands
m.s.kliess@tudelft.nl
https://orcid.org/0000-0001-7219-309X

Catholijn M. Jonker
Interactive Intelligence, Delft University of Technology, Delft, The Netherlands and
Leiden Institute of Advanced Computer Science, Leiden University
C.M.Jonker@tudelft.nl and C.M.Jonker@liacs.leidenuniv.nl
https://orcid.org/0000-0003-4780-7461

M. Birna van Riemsdijk
Interactive Intelligence, Delft University of Technology, Delft, The Netherlands
M.B.vanRiemsdijk@tudelft.nl

Abstract

Behaviour support technology is aimed at assisting people in organizing their Activities of Daily Living (ADLs). Numerous frameworks have been developed for activity recognition and for generating specific types of support actions, such as reminders. The main goal of our research is to develop a *generic* formal framework for representing and reasoning about ADLs and their temporal relations. This framework should facilitate modelling and reasoning about 1) durative activities, 2) relations between higher-level activities and subactivities, 3) activity instances, and 4) activity duration. In this paper we present a temporal logic as an extension of the logic TPTL for specification of real-time systems. Our logic TPTL_{BIH} is defined over Behaviour Identification Hierarchies (BIHs) for representing ADL structure and typical activity duration. To model execution of ADLs, states of the temporal traces in TPTL_{BIH} comprise information about the start, stop and current execution of activities. We provide a number of constraints on these traces that we stipulate are desired for the accurate representation of ADL execution, and investigate corresponding validities in the logic. To evaluate the expressivity of the logic, we give a formal definition for the notion of Coherence for (complex) activities, by which we mean that an activity is done without interruption and in a timely fashion. We show that the definition is satisfiable in our framework. In this way the logic forms the basis for a generic monitoring and reasoning framework for ADLs.

2012 ACM Subject Classification Theory of computation → Modal and temporal logics

Keywords and phrases Temporal Logic, Reasoning, Durative Activities

Digital Object Identifier 10.4230/LIPIcs.TIME.2018.17

Funding This work is part of the research programme CoreSAEP, with project number 639.022.416, which is financed by the Netherlands Organisation for Scientific Research (NWO).

Acknowledgements We would like to thank the anonyomous referees for their insightful and constructive comments that helped improve the work presented in this paper.

25th International Symposium on Temporal Representation and Reasoning (TIME 2018).
Editors: Natasha Alechina, Kjetil Nørvåg, and Wojciech Penczek; Article No. 17; pp. 17:1–17:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Behaviour support technology [15, 9, 12, 18] is aimed at helping people organize their Activities of Daily Living (ADLs) and change their habits. For example, smart home and wearable technology is being developed to recognize people's activities and determine whether it constitutes abnormal behavior [17, 11]. This technology is developed with the aim of allowing elderly and people with special needs to continue living independently by providing support in cases of deviations from expected behavior.

While numerous such frameworks have been developed, they typically focus on specific types of behavior – such as forgetting to perform certain actions, and corresponding support actions – such as smart reminders [9, 12, 18, 13]. Perhaps as a consequence, the specification of what is the desired user behaviour is typically left implicit in these frameworks, intertwined with activity recognition and supportive actions.

In our work we aim to develop a *generic* formal framework for representing and reasoning about ADLs, in which the representation of desired behavior is explicit and decoupled from recognition of actual user behavior and corresponding support actions. These expressions of desired behaviour can originate from users themselves or from others in their social context, such as caregivers. Our approach [8, 23, 16, 22] is inspired by research on normative multiagent systems [2] in which computational models are developed for explicitly representing and reasoning about norms that govern societies of artificial autonomous agents at run-time. An important advantage of our approach is that it can form the foundation for developing more *comprehensive* supportive technology that can support a wide variety of desired types of behavior. Moreover, such explicit representation of desired behavior allows *reasoning* about possible conflicting desires as well as underlying motivations and *values* (see also [16]). Finally, it allows development of *generic supportive strategies* based on the representations of desired behavior and values.

Considering the inherent temporal nature of ADLs, we take temporal logic as a basis for representing desired behavior [10]. Specifically, the framework should facilitate modelling and reasoning about 1) durative activities, 2) relations between higher-level activities and subactivities, 3) activity instances, and 4) normal activity duration.

The motivation for this is that we would not consider actions as being done instantaneosly; thus our logic should not be limited to *when* an activity is done, but also include *for how long* it is being done. A hierarchy of activities will allow for a more fine-grained support: it might be the case that while a user requests support for a particular, complex activity, support is only really needed for a particular part of the activity, e.g. a user requiring support for the activity of *cooking dinner* may only need support when preparing the ingredients, but not for the actual cooking activity. The logic should be able to handle instances of activities, i.e. it should be able to distinguish between the more abstract description given by a user or caregiver, and the activity instance actually carried out. This allows us to check for any deviation from the recorded description of an activity, whether there is a deviation in the manner in which the activity is carried out or whether it takes an unusually long or short time. There are several temporal logic frameworks that allow expressing durative properties, e.g. [14]. All of our requirements considered, we believe that choosing TPTL as the basis for our logic is the most suitable starting point. In particular, we need a way to encode the exact duration that an instance of an activity takes to be carried out, as well as at which time activities are started and completed, respectively. That is, we desire a logic which enables us to express various types of temporal goals. Reasoning about how to handle the interaction between these goals, handling conflicts etc. is not the focus of this paper. Concerning these questions see e.g. [20, 19].

Once we have an understanding of which notions of temporality we want and need to capture for this type of technology, we can analyze which of many existing frameworks for representing and reasoning about norms and time we can build on, e.g. [6, 5, 3, 4, 7, 22].

A key notion for supporting people in their daily lives is that of *normal behaviour* [17, 11]: a support system should be able to detect when a user's activities deviate from a norm, and render support to the user in such situations. However, this notion is sometimes considered in very strict bounds: changing the order of subactivities in a specific activity is already considered *abnormal*. While this certainly applies, e.g. when a user is going to bed before taking off their clothes, or starting to eat before taking certain medication, there are situations in which this ordering may be too strict. When preparing a meal, for instance, we may have a record of the usual order in which the ingredients are prepared. For example, the user usually cuts the tomatoes, then washes the salad leaves, and finally peels and cooks potatoes, deviations from this particular order would not necessarily be seen as abnormal. Especially in the cooking domain, where a lot of activities can be carried out by multiple people in parallel to prepare a meal, we would want to allow more flexibility. In order to achieve this, we define the notion of *Coherence*, by which we mean that an activity is carried out without interruption and in a timely fashion, but which does not account for the particular order in which subactivities are carried out. While this means that there are sometimes extra conditions needed to make sure that e.g., the potatoes are peeled before they are cooked, our notion allows this increased flexibility compared to the much stricter notions of normal / abnormal behaviour.

The paper is organized as follows: in section 2 we introduce the Behaviour Identification Hierarchy $\mathfrak{A}$, which will form the basis for our logic. We also introduce the example of making a pizza for dinner, which we will use throughout the paper as a motivational example as well as demonstrating how the features of our logic work. In section 3, the temporal logic $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$ is defined and we discuss syntax and semantics as well as conditions on the semantic structures that correspond to properties we wish to hold in the logic. We will then introduce the notion of Coherence in section 4, and demonstrate that this is a notion that is satisfiable in our logic $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$.

2 A Basic Behaviour Identification Hierarchy

In this section we will introduce the Behaviour Identification Hierarchy (BIH) that forms the basis of our temporal logic. The BIH is intended to represent the user's recorded behaviour, desired or actual, that the agent will use for rendering support to the user. Before defining the BIH in detail, we will motivate our ideas by an example behaviour description. For this, we will follow the fictitious life of Pedro, who uses a supportive agent for assistance for certain daily tasks.

▶ **Example 1.** Pedro wants to have pizza for dinner and needs assistance for that. He describes his usual way of making a pizza as follows:

- He first prepares the dough and all ingredients he wants on the pizza. These include tomato sauce, cheese, and sometimes mushrooms and olives.
- For the preparation, he rolls out the dough, grates the cheese, washes and cuts the mushrooms, and drains and cuts the olives.
- He then spreads the tomato sauce on the rolled out dough, then decorates with mushrooms and olives, and finally puts the grated cheese on top.
- Finally, he puts the pizza in the oven for about 20 minutes.

Using this description, we obtain a basic abstraction structure for the activity of "*making pizza*":

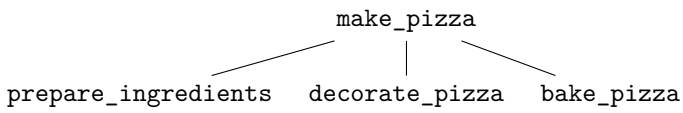

Figure 1

That is, we can refine the single activity of "*making pizza*" by separating it into three distinct subactivities, namely "*preparing ingredients*", "*decorating the pizza*" and "*baking the pizza*". Each of those can further be refined in turn (with the exception of *baking*, perhaps), with e.g. *washing* and *cutting mushrooms* part of *preparing ingredients*.

There is an implicit temporal structure given in the example: Pedro prepares the ingredients *before* he decorates the pizza with them, and he does so *before* baking the pizza. In the picture above one could read this as an ordering on the branches from left to right.

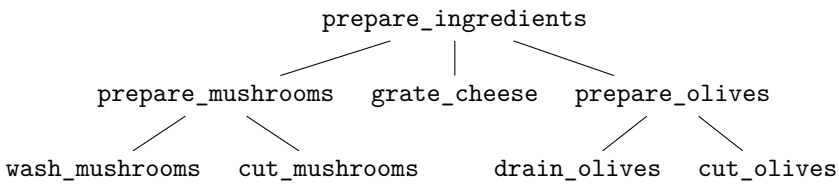

Figure 2

This implicit left-to-right temporal order does not always apply; in Figure 2 one could easily permute the ordering of the three children of `prepare_ingredients` on a temporal axis without changing the description of the activity: if all three child activities are done, the ingredients have been prepared. We even have some more flexibility: Pedro can do the activities described in the leaves of the tree in any order *provided* he washes the mushrooms before cutting them and drains the olives before cutting them. The ability to express temporal ordering will thus be a requirement for our logic. As we will opt for a logic based on TPTL, the *Until* operator will allow us to define the appropriate notion.

We will also need some attribute to record the length of activities; we already know that *baking* takes Pedro about 20 minutes, but it is unlikely that *washing and cutting mushrooms* is done instantaneously by Pedro. Even if Pedro does not care how long this is going to take him, an estimate should be given.

With this in mind we will define the Behaviour Identification Hierarchy as follows:

▶ **Definition 2** (Behaviour Identification Hierarchy)**.** An *activity name* is a string $\mathfrak{a}$. Let $\mathcal{A}$ be a (finite) collection of activity names and let T be a temporal domain. We define the Behaviour Identification Hierarchy $\mathfrak{A}$ as the quadruple $\langle \mathcal{A}, f_T, g_T, \mathit{passive} \rangle$, where $f_T, g_T : \mathcal{A} \to T$ are functions mapping each activity name to its (average) duration and (average) extra time, respectively. Let $\mathit{passive} : \mathcal{A} \to \{0, 1\}$ be an indicator function for passive activities, i.e. $\mathit{passive}(\mathfrak{a}) = 1$ if the activity $\mathfrak{a}$ is passive in the sense that it does not require active involvement by the user.[1]

[1] This can be a waiting "activity", e.g. baking a pizza does not require the user to attend to the oven during the whole time the activity takes place.

Furthermore we define a binary relation PartOf on $\mathcal{A}$, where $\mathfrak{a}$ PartOf $\mathfrak{b}$ iff activity $\mathfrak{a}$ is a part of activity $\mathfrak{b}$. PartOf is antisymmetric, irreflexive and such that for any activities $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$, if $\mathfrak{b}$ PartOf $\mathfrak{a}$ and $\mathfrak{b}$ PartOf $\mathfrak{c}$, then $\mathfrak{a} = \mathfrak{c}$.

The intuition behind this definition is as follows: The PartOf relation provides the tree structure: each activity has a unique parent activity in the tree, and cannot be a part of itself. The condition on unique parents has the effect that PartOf is not transitive. The main purpose is to ensure that we do not accidentally link two distinct subtrees:

Suppose we have an activity `prepare_mushrooms` in the tree as a sub-activity both for `make_pizza` and `make_pasta`. While the activity of preparing mushrooms would certainly look identical for an observer, we cannot use the same mushrooms for both pizza and pasta at the same time. Thus, any instance of preparing mushrooms will either be a sub-activity of an instance of making pizza, or a sub-activity of an instance of making pasta. We want the tree to reflect this.

Taking the transitive closure of PartOf will then enable us to state that `cut_mushrooms` as a sub-activity of `prepare_mushrooms` is also a sub-activity of `make_pizza` or `make_pasta`, respectively.

The functions f_T and g_T define the duration an activity usually takes the user. Here, f_T provides the average time, while g_T describes the time of a "grace period" a monitoring system should take into account before noticing that an activity takes too long. The reason for implementing this with a function, rather than a constant parameter, is that a user may have different "grace periods" for activities: for instance, mushroom cutting might be an easy activity that the user can do without hesitation, but switching the oven on and setting it to the correct temperature might be quite difficult for the user.

The definition above provides a *basic* BIH in the sense that we are not (yet) encoding any explicit temporal ordering. The links given by PartOf also do not distinguish between *optional* or *mandatory* parts. Furthermore, while the examples of such a BIH are all represented in the form of a tree structure, we did not in fact require a BIH to be a tree: the actions with the highest abstraction level, i.e. those that are not a part of another action, are not required to have a common root. Thus, a BIH can be seen as a forest of trees, each of which has as root a unique activity on this highest abstraction level.

3 A Temporal Logic

We will define a temporal logic for our notion of Behaviour Identification Hierarchy. Within this logic we can then express norms and norm compliance for the behaviours given in the BIH ([22]).

▶ **Definition 3.** We define the Temporal Logic $\text{TPTL}_{\text{BIH}}(\mathfrak{A})$ derived from TPTL (see e.g. [1]) as follows: Let $\mathfrak{A} = \langle \mathcal{A}, f_T, g_T, \mathit{passive} \rangle$ be a BIH. Let the temporal domain for $\text{TPTL}_{\text{BIH}}(\mathfrak{A})$ be $T = \mathbb{N}$.

- $\text{TPTL}_{\text{BIH}}(\mathfrak{A})$ has two sorts of variables: *temporal* variables, denoted with lower-case x, y, z, x_i, and *action* variables, denoted with upper-case X, Y, Z, X_i.
  Let V_T denote the set of temporal variables and V_A denote the set of action variables.
- For each $\mathfrak{a} \in \mathcal{A}$ there is a family of action constants $\gamma_1^{\mathfrak{a}}, \gamma_2^{\mathfrak{a}}, \gamma_3^{\mathfrak{a}}, \ldots$ in $\text{TPTL}_{\text{BIH}}(\mathfrak{A})$. There is a 1-1 function $\mathcal{F}$, mapping each such family $\{\gamma_n^{\mathfrak{a}} \mid n \in \mathbb{N}\}$ of constant symbols to its corresponding action $\mathfrak{a} \in \mathcal{A}$. $\mathcal{F}$ induces a partial order $\text{PartOf}_{\mathcal{F}}$ on the constant symbols by defining

$$\gamma_n^{\mathfrak{b}} \text{ PartOf}_{\mathcal{F}} \gamma_m^{\mathfrak{a}} \Leftrightarrow n = m \wedge \mathfrak{b} \text{ PartOf } \mathfrak{a} \wedge \gamma_n^{\mathfrak{b}} \in \mathcal{F}^{-1}(\mathfrak{b}) \wedge \gamma_m^{\mathfrak{a}} \in \mathcal{F}^{-1}(\mathfrak{a}).$$

We will omit the subscript $\mathcal{F}$ in $\mathrm{PartOf}_{\mathcal{F}}$ whenever the function $\mathcal{F}$ is understood from the context.

For each $t \in T$, there is a temporal constant c_t in $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$.

- $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$ contains predicate symbols `Start`, `Stop`, `Doing`, `Wait`, `Started`, `Stopped` and function symbols `Duration`, `Extratime` for actions, and symbols $+$ (addition), $\leq$ and $\equiv_d$ (equality modulo d) for each $d \in T, d > 0$ for temporal variables and constants. We will use $=$ as short-hand for $\equiv_1$.[2]
- The temporal terms, denoted π, π_i, are defined via

$$\pi ::= x + \pi \,|\, c_t + \pi \,|\, x \,|\, c_t.$$

Action terms are given by $\Upsilon ::= X | \gamma_n^{\mathfrak{a}}$.

- We define the atomic formulae p of $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$ by

$$\begin{aligned} p ::= & \texttt{Start}(\Upsilon) \,|\, \texttt{Stop}(\Upsilon) \,|\, \texttt{Doing}(\Upsilon) \,|\, \texttt{Wait}(\Upsilon) \,|\, \texttt{Duration}(\Upsilon) = \pi \,| \\ & \texttt{Extratime}(\Upsilon) = \pi \,|\, \texttt{Started}(\Upsilon, \pi) \,|\, \texttt{Stopped}(\Upsilon, \pi) \,|\, \Upsilon_1 \,\mathrm{PartOf}_{\mathcal{F}}\, \Upsilon_2 \,|\, \pi_1 \leq \pi_2 \,| \\ & \pi_1 \equiv_d \pi_2. \end{aligned}$$

Formulae ϕ of $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$ are then defined by

$$\phi ::= p \,|\, \phi \vee \psi \,|\, \neg\phi \,|\, \bigcirc \phi \,|\, \phi \mathcal{U} \psi \,|\, x.\phi \,|\, \forall X\, \phi,$$

where p is an atomic formula.

We obtain the usual logical operators, i.e. $\Diamond\phi$, $\Box\phi$ and ϕ *before* ψ, as abbreviations for $\top \mathcal{U} \phi$, $\neg\Diamond\neg\phi$ and $\neg(\neg\phi \mathcal{U} \psi)$ (see e.g. [21]).

- Finally, ϕ is an $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$-*sentence* if any temporal variable x occurring in ϕ is bound by an occurrence of the fix quantifier $x.\psi$, and any action variable X is bound by an occurrence of the universal quantifier $\forall X \psi$.

The intuition behind $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$ is the following: the activities of $\mathcal{A}$ should be viewed as the basis for propositions of the form $\texttt{Start}(\gamma_n^{\mathfrak{a}})$, $\texttt{Stop}(\gamma_n^{\mathfrak{a}})$, $\texttt{Doing}(\gamma_n^{\mathfrak{a}})$, etc. In particular, this allows us to treat the actions as *durative*: in the usual interpretation of LTL, actions would be treated as being done instantaneously between states, i.e. if $\mathfrak{a}$ is done in state σ_i, then it is completed in state σ_{i+1}. The attached times τ_i and τ_{i+1} would be able to record some notion of duration, but if other actions with shorter durations would be done simultaneously with $\mathfrak{a}$, we would need to have a number of consecutive states registering that $\mathfrak{a}$ is being done. The fact that $\mathfrak{a}$ then occurs in multiple states might be taken as multiple instances of $\mathfrak{a}$ being done, rather than just one action taking some longer time. Using `Doing`() here does not only allow for this interpretation to be avoided, it also allows the notion of "taking a break", i.e. the language is able to express that an action is interrupted for some time, and then continued.

The concrete language varies with different BIH $\mathfrak{A}$. Furthermore, our choice of taking countably many distinct instances for each activity is motivated by the idea that each instance of an activity can only be done once: if we baked pizza on Monday, 5th February 2018, then

[2] Note that both `Started` and `Stopped` are intended to work as predicates on *pairs* of the form $\langle \Upsilon, \pi \rangle$; while it certainly is permissible to have distinct action instances $\gamma_n^{\mathfrak{a}}$ associated with the same time c_t, conditions (C8) and (C9) in Definition 5 will ensure that each such instance will only be paired with precisely one such time. Thus both of these predicates could also be seen as functions. We opt to view them as predicates here to emphasize the rather static nature of record keeping for which they are intended.

we have an instance for the activity in any trace starting with or before this day, but we would certainly never bake the exact same pizza again in the future. Even if we chose to use the same recipe in the future to bake another pizza, we would still argue that it is not the exact same pizza that was baked in February 2018. To allow us to make this distinction one can see the index n of an activity constant $\gamma_n^{\mathfrak{a}}$ as some form of ID associated with a specific instance of an action, e.g. "the first time we bake this pizza", "the second time", etc.

While this means that the number of *distinct instances* of activities is infinite[3], the number of distinct *activities* is still finite as $\mathcal{A}$ is finite. We will now define trace semantics for $\text{TPTL}_{\text{BIH}}(\mathfrak{A})$. The general idea is as follows: a model of a collection of $\text{TPTL}_{\text{BIH}}(\mathfrak{A})$ -sentences is a sequence of timed states. Each state consists of a number of actions holding, namely those actions that are (actively) being carried out at the given time. The "timed" nature of these states is accomplished by pairing each state σ_i with a time τ_i.

▶ **Definition 4** (Trace semantics). A $\text{TPTL}_{\text{BIH}}(\mathfrak{A})$-structure $\mathfrak{M}$ is a pair $\mathfrak{M} = \langle \vec{\rho}, \mathcal{E} \rangle$, where

- $\vec{\rho}$ is a (countably infinite) sequence of pairs $\rho_i = \langle \sigma_i, \tau_i \rangle$, where

$$\sigma_i \in \mathcal{P}\left([\mathcal{A} \times \mathbb{N}] \cup \text{Start}(\mathcal{A} \times \mathbb{N}) \cup \text{Stop}(\mathcal{A} \times \mathbb{N}) \cup \text{Started}(\mathcal{A} \times \mathbb{N} \times T) \cup \text{Stopped}(\mathcal{A} \times \mathbb{N} \times T)\right)$$

  and $\tau_i \in T$, such that
  - $\text{Start}(\mathcal{A} \times \mathbb{N}) = \{\langle \text{Start}(\mathfrak{a}), i \rangle \mid \mathfrak{a} \in \mathcal{A}, i \in \mathbb{N}\}$,
  - $\text{Stop}(\mathcal{A} \times \mathbb{N}) = \{\langle \text{Stop}(\mathfrak{a}), i \rangle \mid \mathfrak{a} \in \mathcal{A}, i \in \mathbb{N}\}$,
  - $\text{Started}(\mathcal{A} \times \mathbb{N} \times T) = \{\langle \text{Started}(\mathfrak{a}), i, t \rangle \mid \mathfrak{a} \in \mathcal{A}, i \in \mathbb{N}, t \in T\}$,
  - $\text{Stopped}(\mathcal{A} \times \mathbb{N} \times T) = \{\langle \text{Stopped}(\mathfrak{a}), i, t \rangle \mid \mathfrak{a} \in \mathcal{A}, i \in \mathbb{N}, t \in T\}$,
  - the sequence $\vec{\tau} = \langle \tau_i \rangle$ is monotone, i.e. $i \leq j$ implies that $\tau_i \leq \tau_j$, and progressive, i.e. for each $t \in T$ there is some $j \in \mathbb{N}$ such that $\tau_j > t$,
- $\mathcal{E}_T : V_T \to T$ is an interpretation function for the temporal variables of $\text{TPTL}_{\text{BIH}}(\mathfrak{A})$; we extend $\mathcal{E}_T$ to temporal constants by letting $\mathcal{E}_T(c_T) = c_t^{\mathfrak{M}} \in T$ for each temporal constant c_t of $\text{TPTL}_{\text{BIH}}(\mathfrak{A})$.
- $\mathcal{E}_A : V_A \to \mathcal{A}$ is an interpretation function for the action variables of $\text{TPTL}_{\text{BIH}}(\mathfrak{A})$; we extent $\mathcal{E}_A$ to action constants by letting $\mathcal{E}_A(\gamma_n^{\mathfrak{a}}) = \mathfrak{a}$ for each action constant $\gamma_n^{\mathfrak{a}}$.
- For readability, and since both the domains and ranges of $\mathcal{E}_T$ and $\mathcal{E}_A$ are distinct, we can define a common interpretation function $\mathcal{E} = \mathcal{E}_T \cup \mathcal{E}_A$.

Let $\mathfrak{M}$ be a $\text{TPTL}_{\text{BIH}}(\mathfrak{A})$-structure and $i \in \mathbb{N}$. Then $\langle \mathfrak{M}, i \rangle \models \phi$ is defined as follows:

- $\langle \mathfrak{M}, i \rangle \models \texttt{Doing}(\gamma_n^{\mathfrak{a}})$ iff $\langle \mathfrak{a}, n \rangle \in \sigma_i$,
- $\langle \mathfrak{M}, i \rangle \models \texttt{Start}(\gamma_n^{\mathfrak{a}})$ iff $\langle \text{Start}(\mathfrak{a}), n \rangle \in \sigma_i$,
- $\langle \mathfrak{M}, i \rangle \models \texttt{Stop}(\gamma_n^{\mathfrak{a}})$ iff $\langle \text{Stop}(\mathfrak{a}), n \rangle \in \sigma_i$,
- $\langle \mathfrak{M}, i \rangle \models \texttt{Started}(\gamma_n^{\mathfrak{a}}, \pi)$ iff $\langle \text{Started}(\mathfrak{a}), n, t \rangle \in \sigma_i$ and $\mathcal{E}(\pi) = t$,
- $\langle \mathfrak{M}, i \rangle \models \texttt{Stopped}(\gamma_n^{\mathfrak{a}}, \pi)$ iff $\langle \text{Stopped}(\mathfrak{a}), n, t \rangle \in \sigma_i$ and $\mathcal{E}(\pi) = t$,
- $\langle \mathfrak{M}, i \rangle \models \texttt{Duration}(\gamma_n^{\mathfrak{a}}) = \pi$ iff $f_T(\mathfrak{a}) = t$ and $\mathcal{E}(\pi) = t$,
- $\langle \mathfrak{M}, i \rangle \models \texttt{Extratime}(\gamma_n^{\mathfrak{a}}) = \pi$ iff $g_T(\mathfrak{a}) = t$ and $\mathcal{E}(\pi) = t$,
- $\langle \mathfrak{M}, i \rangle \models \texttt{Wait}(\gamma_n^{\mathfrak{a}})$ iff $\textit{passive}(\mathfrak{a}) = 1$,
- $\langle \mathfrak{M}, i \rangle \models \gamma_n^{\mathfrak{a}} \,\text{PartOf}_{\mathcal{F}}\, \gamma_m^{\mathfrak{b}}$ iff $\mathfrak{a} \,\text{PartOf}\, \mathfrak{b}$ and $n = m$,
- $\langle \mathfrak{M}, i \rangle \models \pi_1 \leq \pi_2$ iff $\mathcal{E}(\pi_1) \leq \mathcal{E}(\pi_2)$.
- $\langle \mathfrak{M}, i \rangle \models \pi_1 \equiv_d \pi_2$ iff $\mathcal{E}(\pi_1) \equiv_d \mathcal{E}(\pi_2)$.

[3] In all practicality, this number will be finite as well, as human life is finite. Limiting the number of instance for any activity right from the start, however, seems to be too strict a requirement here.

- $\langle \mathfrak{M}, i\rangle \models \phi \vee \psi$ iff $\langle \mathfrak{M}, i\rangle \models \phi$ or $\langle \mathfrak{M}, i\rangle \models \psi$,
- $\langle \mathfrak{M}, i\rangle \models \neg\phi$ iff $\langle \mathfrak{M}, i\rangle \not\models \phi$,
- $\langle \mathfrak{M}, i\rangle \models \bigcirc\phi$ iff $\langle \mathfrak{M}, i+1\rangle \models \phi$,
- $\langle \mathfrak{M}, i\rangle \models \phi\mathcal{U}\psi$ iff there is $k \geq i$ such that $\langle \mathfrak{M}, k\rangle \models \psi$ and for all $i \leq l < k$, $\langle \mathfrak{M}, l\rangle \models \phi$,
- $\langle \mathfrak{M}, i\rangle \models x.\phi$ iff $\langle \mathfrak{M}', i\rangle \models \phi$ for that $\mathfrak{M}' = \langle \vec{\rho}, \mathcal{E}'\rangle$ such that $\mathcal{E}'_T(y) = \mathcal{E}_T(y)$ for all $y \neq x$ and $\mathcal{E}'_T(x) = \tau_i$,
- $\langle \mathfrak{M}, i\rangle \models \forall X\phi$ iff $\langle \mathfrak{M}', i\rangle \models \phi(X)$ for any $\mathfrak{M}' = \langle \rho, \mathcal{E}'\rangle$ such that $\mathcal{E}_A$ agrees with $\mathcal{E}'_A$ on all variables except for X.

If the truth value of $\langle \mathfrak{M}, i\rangle \models \phi$ is independent of i, then we shall drop i and simply write $\mathfrak{M} \models \phi$.

We will now define a few criteria that one might want to hold on the traces. For this, we let $\mathfrak{M}$ be an $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$-structure.

▶ **Definition 5.** We will define the following criteria for $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$ structures.

- (C0), (Finiteness of states) For each $i \in \mathbb{N}$, σ_i is finite.
- (C1), (No Stop without Start) If $\langle \mathrm{Stop}(\mathfrak{a}), n\rangle \in \sigma_i$, then there exists $k < i$ such that $\langle \mathrm{Start}(\mathfrak{a}), n\rangle \in \sigma_k$.
- (C2), (No Doing before Start) $\langle \mathrm{Start}(\mathfrak{a}), n\rangle \in \sigma_i$ implies $\forall k < i\, \langle \mathfrak{a}, n\rangle \notin \sigma_k$.
- (C3), (No Doing after Stop) $\langle \mathrm{Stop}(\mathfrak{a}), n\rangle \in \sigma_i$ implies $\forall k > i\, \langle \mathfrak{a}, n\rangle \notin \sigma_k$.
- (C4), (Logging Start) If $\langle \mathrm{Start}(\mathfrak{a}), n\rangle \in \sigma_i$, then for all $k > i$, $\langle \mathrm{Started}(\mathfrak{a}), n, \tau_i\rangle \in \sigma_k$.
- (C5), (Logging Stop) If $\langle \mathrm{Stop}(\mathfrak{a}), n\rangle$, then for all $k > i$, $\langle \mathrm{Stopped}(\mathfrak{a}), n, \tau_i\rangle \in \sigma_k$.
- (C6), (Doing at Start) If $\langle \mathrm{Start}(\mathfrak{a}), n\rangle \in \sigma_i$, then $\langle \mathfrak{a}, n\rangle \in \sigma_i$.
- (C7), (Not Doing at Stop) If $\langle \mathrm{Stop}(\mathfrak{a}), n\rangle \in \sigma_i$, then $\langle \mathfrak{a}, n\rangle \notin \sigma_i$.
- (C8), (Unique Start) If $\langle \mathrm{Start}(\mathfrak{a}), n\rangle \in \sigma_i$ and $\langle \mathrm{Start}(\mathfrak{a}), n\rangle \in \sigma_k$, then $i = k$.
- (C9), (Unique Stop) If $\langle \mathrm{Stop}(\mathfrak{a}), n\rangle \in \sigma_i$ and $\langle \mathrm{Stop}(\mathfrak{a}), n\rangle \in \sigma_k$, then $i = k$.

The conditions above are intended to correspond to properties we wish to hold in our logic: for instance, as discussed above, the states of our logic will only be finite, hence we have (C0) in the list of desired criteria.

The criteria (C1)-(C5) state some simple properties for `Start`() and `Stop`(): we actually want that `Start`() cannot be preceded by an instance of doing an activity, since this would contradict the intuition that one can only start an activity that is not already being executed. Similarly, we do not want any instance of doing an activity to occur after a `Stop`() is placed in the trace. We also want to ensure that any stop signal for an activity is preceded by a start signal. Furthermore, (C4) and (C5) ensure that we are logging all occurrences of `Start`() and `Stop`().

One could argue here that `Start`() could also be defined via properties of the language like $\mathcal{U}$, e.g. $\mathtt{Start}(\gamma_n^{\mathfrak{a}})$ could be defined via $\neg\mathtt{Doing}(\gamma_n^{\mathfrak{a}})\mathcal{U}\mathtt{Doing}(\gamma_n^{\mathfrak{a}})$. However, we want to allow that an activity can be paused, e.g. $\mathtt{Doing}(\gamma_n^{\mathfrak{a}})$ is not required to always hold on the interval determined by $\mathtt{Start}(\gamma_n^{\mathfrak{a}})$ and $\mathtt{Stop}(\gamma_n^{\mathfrak{a}})$. Without $\mathtt{Start}(\gamma_n^{\mathfrak{a}})$ as a primitive, we would need past time operators to determine whether a $\mathtt{Doing}(\gamma_n^{\mathfrak{a}})$ is actually the first one occurring, or whether the action has just been paused for a long time, and thus $\neg\mathtt{Doing}(\gamma_n^{\mathfrak{a}})\mathcal{U}\mathtt{Doing}(\gamma_n^{\mathfrak{a}})$ holds at a point i just because the most recent instance of $\mathtt{Doing}(\gamma_n^{\mathfrak{a}})$ has occurred at a point $k < i$.

With the criteria (C6) and (C7) we want to ensure that any `Start`() signal is indeed accompanied by actually doing an activity, while the `Stop`() does indeed mean that we are no longer seeing the activity being done.

The criteria (C8) and (C9) provide our interpretation of `Start`() and `Stop`(). We see the start of an activity as the initial point when it is begun, and the stop as the point of completion. Thus these signals are unique – we do not intend to use them in order to measure any breaks in the execution of an activity.

While it is an immediate consequence that (C8) follows from the conditions (C2) and (C6), this is not true for (C9); this condition does not follow from (C3) and (C7). The crucial point here is that the list of conditions does not guarantee that `Stop`() has to occur immediately after the last instance of doing an activity – therefore, having multiple `Stop`() signals occurring in the trace does not contradict any of the criteria (C0) - (C8).

The following proposition states that our list of conditions is indeed satisfyable.

▶ **Proposition 6.** *Given a BIH $\mathfrak{A}$, there exists a $TPTL_{\text{BIH}}(\mathfrak{A})$-structure $\mathfrak{M}$ satisfying the conditions (C0) - (C9).*

Sketch. Let $\mathfrak{A} = \langle \mathcal{A}, f_T, g_T, \mathit{passive} \rangle$ a BIH. We construct a structure as follows: Let $\vec{\tau}$ be an arbitrary, monotone and progressive sequence of T.

In a first step, add both $\langle \mathfrak{a}, 0 \rangle$ and $\langle \text{Start}(\mathfrak{a}), 0 \rangle$ to σ_0 for any $\mathfrak{a} \in \mathcal{A}$. Let σ_1 be the set containing the corresponding $\langle \text{Stop}(\mathfrak{a}), 0 \rangle$. Let all other σ_i be empty.

Now add, for any $\mathfrak{a} \in \mathcal{A}$, $\langle \text{Started}(\mathfrak{a}), 0, \tau_0 \rangle$ to all σ_i with $i > 0$, and similarly add $\langle \text{Stopped}(\mathfrak{a}), 0, \tau_1 \rangle$ to all σ_i with $i > 1$.

Note that since $\mathcal{A}$ is finite, each of the σ_i are finite.

It is easy to verify that the criteria (C1)-(C9) are satisfied as well. E.g., the fact that $\langle \text{Start}(\mathfrak{a}), 0 \rangle$ is not preceded by any occurrence of $\langle \mathfrak{a}, 0 \rangle$ is trivially satisfied. And since $\langle \mathfrak{a}, 0 \rangle$ is an element of σ_0 but not of σ_1, we have (C2), (C3) and (C6) holding. ◀

4 Coherence

In this section, we will define Coherence for an instance of an activity $\mathfrak{a}$. Informally, an activity is *coherently done* if it is done without distraction and without unexpected delays. In terms of Example 1, Pedro does not do any tasks not related to pizza making, and does not take too long breaks in between different subactivities, as well as during them. The only exception we may allow is the time when the pizza is baking in the oven: one could argue that since this is a *passive* activity, Pedro could use that time to attend to other activities, e.g. taking out the rubbish.

▶ **Definition 7** (Coherence)**.** We say that in a structure $\mathfrak{M}$, Coherence holds for an action instance $\gamma_n^{\mathfrak{a}}$ at point i, iff

$$\begin{aligned}
&\langle \mathfrak{M}, i \rangle \models \forall X \, \text{PartOf}_{\mathcal{F}} \, \gamma_n^{\mathfrak{a}} \, [\texttt{Start}(\gamma_n^{\mathfrak{a}}) \, \mathit{before} \, \texttt{Start}(X) \wedge \texttt{Stop}(X) \, \mathit{before} \, \texttt{Stop}(\gamma_n^{\mathfrak{a}})] \\
&\wedge \forall X \, [\neg X \, \text{PartOf}_{\mathcal{F}} \, \gamma_n^{\mathfrak{a}} \rightarrow (\neg \texttt{Start}(X) \vee \\
&\forall Y \, \text{PartOf}_{\mathcal{F}} \, \gamma_n^{\mathfrak{a}} \, [\texttt{Doing}(Y) \rightarrow (\texttt{Wait}(Y) \wedge \texttt{Stop}(X) \, \mathit{before} \, \texttt{Stop}(Y))])] \\
&\wedge \Diamond \, [\texttt{Started}(\gamma_n^{\mathfrak{a}}, \pi_1) \wedge \texttt{Stopped}(\gamma_n^{\mathfrak{a}}, \pi_2) \wedge \pi_2 \leq \pi_1 + \texttt{Duration}(\gamma_n^{\mathfrak{a}}) + \texttt{Extratime}(\gamma_n^{\mathfrak{a}})]
\end{aligned}$$

The intuition behind this definition is as follows: we consider an activity as being done coherently when any of its subactivities are performed only within the limits set by the activity itself; furthermore, unrelated activities are only performed if there is some time in which the user has to wait anyways; and finally, doing the activity does not take up more time than it should.

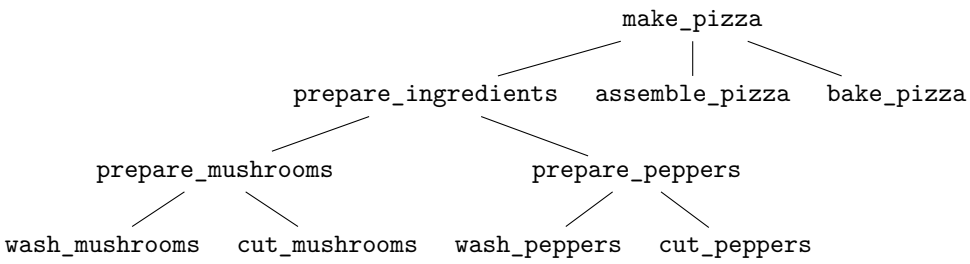

Figure 3

Notice that this definition means, Coherence holds only for activities that are *eventually completed*: we cannot refer to $\texttt{Stopped}(\gamma_n^a, \pi)$ before we actually see a $\texttt{Stop}(\gamma_n^{\mathfrak{a}})$ in the trace. The eventuality in the final conjunct thus implies that Coherence can only hold if the activity is eventually stopped. We would argue that this is indeed intended – the alternatives are that an activity is either abandoned or is carried on forever. Furthermore, this last conjunct implies that the collective duration of all subactivities of $\mathfrak{a}$ executed in this instance does not exceed the duration of $\mathfrak{a}$ in this instance. Since we want Coherence to have a chance of regularly being satisfied, we need the BIH $\mathfrak{A}$ to fulfill the further condition that for all $\mathfrak{a} \in \mathcal{A}$, $\sum_{\mathfrak{b}\,\mathrm{PartOf}\,\mathfrak{a}} f_T(\mathfrak{b}) + g_T(\mathfrak{b}) \leq f_T(\mathfrak{a}) + g_T(\mathfrak{a})$. This is a rational assumption, as one would not expect that an activity can be regularly done in a shorter timespan than it takes to do all of its parts if these are done consecutively.

As an example for Coherence, consider the tree shown in Figure 3.

With respect to the remark above regarding the condition on the durations for subactivities, we also add the following average durations (in minutes):

action	average duration
`make_pizza`	45
`bake_pizza`	20
`assemble_pizza`	5
`prepare_mushrooms`	8
`prepare_peppers`	8

Summing the times for the different parts of `make_pizza`, we see that doing everything effectively, there is a buffer of 4 minutes for making pizza.

Consider the following untimed initial trace $\vec{\sigma}$. For readability, we will omit the notation $\gamma_n^{\mathfrak{a}}$ and just use $\mathfrak{a}$, with $n = 0$ in this case. E.g., `make_pizza` in the table below should be read as $\gamma_0^{\texttt{make\_pizza}}$.

σ_0	`Start(make_pizza)`	`Start(wash_mushrooms)`
σ_1		`Stop(wash_mushrooms)`
σ_2		`Start(wash_peppers)`
σ_3		`Stop(wash_peppers)`
σ_4		`Start(cut_mushrooms)`
σ_5		`Stop(cut_mushrooms)`
σ_6		`Start(cut_peppers)`
σ_7		`Stop(cut_peppers)`
σ_8		`Start(assemble_pizza)`
σ_9		`Stop(assemble_pizza)`
σ_{10}		`Start(bake_pizza)`
σ_{11}	`Stop(make_pizza)`	`Stop(bake_pizza)`

Furthermore, add the logging statement `Started`(`make_pizza`, τ_0) to each σ_i with $i > 0$ and similarly add `Stopped`(`make_pizza`, τ_{11}) to each σ_i with $i > 12$. Proceed similarly for the logging events for each of the subactivities of `make_pizza`. We treat the τ_i here as placeholders for now, and replace them by the appropriate values once we have defined the concrete values below.

This initial segment can be extended to a trace $\vec{\sigma}$ arbitrarily to obtain a trace of states that satisfies a coherent run of making a pizza. To obtain an $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$-structure, we furthermore need some sequence of time points $\vec{\tau}$. The following initial sequence will then show that Coh(`make_pizza`) is satisfiable:

$$\tau_0 = 0, \tau_1 = 3, \tau_2 = 3, \tau_3 = 6, \tau_4 = 6, \tau_5 = 11, \tau_6 = 11, \tau_7 = 16,$$
$$\tau_8 = 16, \tau_9 = 21, \tau_{10} = 21, \tau_{11} = 41$$

Interpreting each time point τ_i as the minutes, then any extension of this initial sequence of τ_i to a time trace $\vec{\tau}$ will lead to an $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$-structure witnessing the satisfiability. Note that we can make this an acceptable $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$-structure by just adding to the σ_i the corresponding logging entries. Furthermore, it is easy to see now, that this does not make Coherence a theorem of acceptable $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$-structures: the validity of Coh(`make_pizza`) depends in this example on the values we pick for τ_i. Just picking different, larger values for the τ_i we can obtain an $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$-structure in which Coh(`make_pizza`) is false, simply because the condition for starting time and stopping time of `make_pizza` does not hold.

Intuitively, it seems that there are only two ways in which Coherence for a complex activity can fail: one would not expect to see the `Stop`(`make_pizza`) statement before e.g. `Start`(`bake_pizza`) happens. Thus if the time sequence matches the condition of Coherence for some activity $\mathfrak{a}$, one would expect that the only other way to make it still fail is if one introduces an unrelated activity, e.g. if in the pizza making example we have `Start`(`watch_TV`) in σ_0.

The strategy used above can be used to demonstrate that for any BIH $\mathfrak{A}$ satisfying the extra condition on the time functions, there exist $\mathrm{TPTL}_{\mathrm{BIH}}(\mathfrak{A})$ - structures $\mathfrak{M}$ satisfying Coherence for each $\mathfrak{a} \in \mathfrak{A}$:

▶ **Proposition 8.** *Let $\mathfrak{A}$ be a BIH satisfying the condition that for all $\mathfrak{a} \in \mathcal{A}$, $\sum_{\mathfrak{b}\,\mathrm{PartOf}\,\mathfrak{a}} f_T(\mathfrak{b}) + g_T(\mathfrak{b}) \leq f_T(\mathfrak{a}) + g_T(\mathfrak{a})$. Then there exists a $TPTL_{\mathrm{BIH}}(\mathfrak{A})$ - structure $\mathfrak{M}$ such that for any $\mathfrak{a} \in \mathfrak{A}$ and any instance $\gamma_n^{\mathfrak{a}}$ there is an index $i_{\mathfrak{a},n}$ such that*

$$\langle \mathfrak{M}, i_{\mathfrak{a},n} \rangle \models \mathrm{Coh}(\gamma_n^{\mathfrak{a}}).$$

Sketch. Assume an enumeration $\mathfrak{a}_1, \mathfrak{a}_2, \ldots, \mathfrak{a}_m$ of the top-level actions of $\mathcal{A}$, i.e. for each of those $\mathfrak{a}_i$, there is not $\mathfrak{b}$ such that $\mathfrak{a}_i\,\mathrm{PartOf}\,\mathfrak{b}$. Starting with $\mathfrak{a}_1$, we construct a trace satisfying $\mathrm{Coh}(\mathfrak{b})$ for any $\mathfrak{b} \in \mathcal{A}$ as follows.

Start off with any σ_i the empty set. Let $\tau_0 = 0$. Add to σ_0 the elements $\langle \mathrm{Start}(\mathfrak{a}_1), 0\rangle$ and $\langle \mathfrak{a}_1, 0\rangle$. Identify each $\mathfrak{b}_j\,\mathrm{PartOf}\,\mathfrak{a}_1$, and enumerate them as well, add $\langle \mathrm{Start}(\mathfrak{b}_1), 0\rangle$, $\langle \mathfrak{b}_1, 0\rangle$ to σ_0 as well. Continue until reaching a level in the tree for which there is no child, i.e. reach $\mathfrak{c}_1\,\mathrm{PartOf} \cdots \mathrm{PartOf}\,\mathfrak{b}_1\,\mathrm{PartOf}\,\mathfrak{a}_1$ such that there is no $\mathfrak{d}$ with $\mathfrak{d}\,\mathrm{PartOf}\,\mathfrak{c}_1$. This exists, since $\mathcal{A}$ is finite, and PartOf is well-founded.

Now add to σ_1 the element $\langle \mathrm{Stop}(\mathfrak{c}_1), 0\rangle$, for any of the higher level activities $\mathfrak{d}$ that were started add the elements $\langle \mathfrak{d}, 0\rangle$, and all the appropriate logging elements $\langle \mathrm{Started}(\mathfrak{d}), 0, 0\rangle$. Let $\tau_1 = f_T(\mathfrak{c}_1)$. At this point $\mathrm{Coh}(\gamma_0^{\mathfrak{c}_1})$ holds, as it does not have any subactivities and the difference $\tau_1 - \tau_0$ has the correct value.

Traversing the BIH tree, continue like this until all the subactivities are recorded as being started and stopped in a finite initial segment of $\vec{\sigma}$, with the τ_i having the appropriate values. Having reached σ_k in this way, add $\langle \text{Stopped}(\mathfrak{a}_1), 0\rangle$ to this σ_k and observe that $\text{Coh}(\gamma_0^{\mathfrak{a}_1})$ holds at this point.

Proceed like this for all the other $\mathfrak{a}_i$. Having reached a finite point l such that at σ_l, all the activity instances with index 0 have Coherence satisfied, we can just copy this initial segment, increase the indices of the instances by one and add τ_l to each of the τ_i, obtaining $\langle \vec{\sigma}, \vec{\tau}\rangle$ up to index $2l$ satisfying coherence for the first two instances for each activity in $\mathfrak{A}$. Continuing in this way, we obtain an infinite trace $\vec{\rho} = \langle \vec{\sigma}, \vec{\tau}\rangle$ satisfying $\text{Coh}(\gamma_n^{\mathfrak{a}})$ for each $\mathfrak{a} \in \mathcal{A}$ and each $n \in \mathbb{N}$. ◀

While it is certainly conceivable that Coherence could be used in a recursive manner, this is not given by the definition. For instance, considering Coh(`make_pizza`) in the example above, it is only required that `prepare_mushrooms` and `prepare_peppers` are executed within the activity of `make_pizza`, but not that coherence holds for both sub-activities. Depending on the user's need for support, one may add the requirement that Coh(`make_pizza`) $\wedge$ Coh(`prepare_mushrooms`) $\wedge$ Coh(`prepare_peppers`) holds, as the user feels incapable of mixing the latter two tasks. Thus Coherence need not propagate recursively to sub-activities in the BIH.

However, the situation is different when considering two distinct, unrelated activities. Suppose that the user needs to take out the trash before 6 pm as well as prepare the pizza for dinner at 6 pm. Taking the trash out at some time during the pizza preparation would violate Coherence for `making_pizza`. However, we would argue that the role of our logic is precisely to recognize such conflicts; an agent utilizing the logic would thus be able to recognize such a conflict, and can act upon that, e.g. by suggesting to take the trash out later, when the pizza is in the oven, or by merely reminding the user that some ingredients are not prepared when the user has finished the activity of taking the trash out. How a conflict like this is handled should be decided on a different level, e.g. through providing priority levels for activities in the BIH or other means. That is, we do not intend our logic $\text{TPTL}_{\text{BIH}}(\mathfrak{A})$ to handle conflicts like these.

5 Conclusion/Discussion

In this paper, we have presented $\text{TPTL}_{\text{BIH}}(\mathfrak{A})$ as a logic capable of formulating statements about Activities of Daily Living. The language can act as a framework for reasoning about and monitoring user-reported behaviour.

We see this paper as an important step towards building supportive systems that take a focus on the needs and well-being of the people intended to use them. Our approach does not only allow modelling different levels of activities, and by that allow for a more fine-grained support, but also provides the ability to let the cared-for person have an impact on the way they are supported: our logic allows that desired norms and behaviours are either formulated as general statements applying to all activities alike, or being encoded into the logic by means of the Behaviour Identification Hierarchy.

In particular our notion of Coherence together with the BIH allows a supportive agent to be flexibly attending to the user's needs: while an agent following our logic is able to register when actions are done in a timely manner, it does allow for individual difficulties, e.g. Pedro might struggle a lot more with remembering to take the pizza out of the oven before it is burnt than cutting the mushrooms, and thus the "grace periods" given by the function g_T in Pedro's BIH can vary accordingly.

The notion of Coherence, as defined in this paper, is closely related to the notion of normal/abnormal behaviour as mentioned in e.g. [17]. However, the notion of "normal behaviour", and thus its counterpart 'abnormal behaviour' is rather strictly formulated in the literature. While those descriptions of normal behaviour would imply the notion of Coherence, our approach is much more flexible: subactivities can be carried out in any order, so as long as the end result is succesfully achieved in due time, the monitored user can act in any way they want without the need of the supportive agent to intervene.

While the basic logic TPTL is known to be decidable, it is currently unkown whether our modifications of the base language change this fact. In particular since several generalizations of TPTL, allowing for dense or real time, are undecidable, the question whether our modifications have similar effects on the decidability becomes important.

Furthermore, the definition of a BIH as used in this paper is far from being practical in real-life situations: we currently do not allow any partiality: if putting mushrooms on the pizza is at least sometimes desired by the user, our BIH model makes that a certainty. It is clear that adding a notion of "sometimes" or "often" needs to be added to the current framework in order to more closely model actual human behaviour. As part of future work in this direction experiments are planned within the CoreSAEP research project. These will involve the question whether a more complex version of our BIH trees will indeed be useful to describe human habits.

References

1. Rajeev Alur and Thomas A. Henzinger. A really temporal logic. *J. ACM*, 41(1):181–203, 1994. doi:10.1145/174644.174651.
2. Giulia Andrighetto, Guido Governatori, Pablo Noriega, and Leendert van der Torre, editors. *Normative Multi-Agent Systems*, volume 4 of *Dagstuhl Follow-Ups*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013. doi:10.4230/DFU.Vol4.12111.i.
3. Jan Broersen, Frank Dignum, Virginia Dignum, and John-Jules Ch Meyer. Designing a deontic logic of deadlines. In *Proceedings Seventh International Workshop on Deontic Logic in Computer Science (DEON'04)*, volume 3065 of *LNCS*, pages 43–56. Springer-Verlag, 2004.
4. Stephen Cranefield. A rule language for modelling and monitoring social expectations in multi-agent systems. In *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems (ANIREM'05 and OOOP'05)*, volume 3913 of *LNCS*, pages 246–258, 2006.
5. F. Dignum and R. Kuiper. Specifying deadlines with dense time using deontic and temporal logic. *International Journal of Electronic Commerce*, 3(2):67–86, 1998.
6. Guido Governatori, Joris Hulstijn, Régis Riveret, and Antonino Rotolo. Characterising deadlines in temporal modal defeasible logic. In *Proceedings of the 20th Australian joint conference on Advances in artificial intelligence*, pages 486–496. Springer-Verlag, 2007.
7. Koen V. Hindriks and M. Birna van Riemsdijk. A real-time semantics for norms with deadlines. In *Proceedings of the twelfth international joint conference on autonomous agents and multiagent systems (AAMAS'13)*, pages 507–514. IFAAMAS, 2013.
8. Alex Kayal, Willem-Paul Brinkman, Hanna Zoon, Mark A. Neerincx, and M. Birna van Riemsdijk. A value-sensitive mobile social application for families and children. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 22nd Conference on User Modeling, Adaptation, and Personalization (UMAP'14)*, volume 1181. CEUR, 2014.
9. Thomas Kleinberger, Andreas Jedlitschka, Holger Storf, Silke Steinbach-Nordmann, and Stephan Prueckner. An approach to and evaluations of assisted living systems using ambient intelligence for emergency monitoring and prevention. In Constantine Stephanidis, editor, *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, pages 199–208, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

10 Malte S. Kließ and M. Birna van Riemsdijk. Requirements for a temporal logic of daily activities for supportive technology. In Matteo Baldoni, Cristina Baroglio, and Roberto Micalizio, editors, *Proceedings of the First Workshop on Computational Accountability and Responsibility in Multiagent Systems co-located with 20th International Conference on Principles and Practice of Multi-Agent Systems, CARe-MAS@PRIMA 2017, Nice, France, October 31st, 2017.*, volume 2051 of *CEUR Workshop Proceedings*, pages 43–51. CEUR-WS.org, 2017. URL: `http://ceur-ws.org/Vol-2051/paper4.pdf`.

11 Óscar D. Lara and Miguel A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192–1209, 2013.

12 T. Magherini, A. Fantechi, C. D. Nugent, and E. Vicario. Using temporal logic and model checking in automated recognition of human activities for ambient-assisted living. *IEEE Transactions on Human-Machine Systems*, 43(6):509–521, Nov 2013. `doi:10.1109/TSMC.2013.2283661`.

13 Ehsan Nazerfard, Parisa Rashidi, and Diane J. Cook. Using association rule mining to discover temporal relations of daily activities. In Bessam Abdulrazak, Sylvain Giroux, Bruno Bouchard, Hélène Pigot, and Mounir Mokhtari, editors, *Toward Useful Services for Elderly and People with Disabilities*, pages 49–56, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

14 Isabel Nunes, Jose Luiz Fiadeiro, and Wladyslaw M. Turski. A modal logic of durative actions. In Howard Barringer, Michael Fisher, Dov Gabbay, and Graham Gough, editors, *Advances in Temporal Logic*, pages 299–317. Springer Netherlands, Dordrecht, 2000. `doi:10.1007/978-94-015-9586-5_15`.

15 Harri Oinas-Kukkonen. Behavior change support systems: A research model and agenda. In Thomas Ploug, Per Hasle, and Harri Oinas-Kukkonen, editors, *Persuasive Technology: 5th International Conference, PERSUASIVE 2010, Copenhagen, Denmark, June 7-10, 2010. Proceedings*, pages 4–14. Springer, 2010.

16 Pietro Pasotti, M. Birna van Riemsdijk, and Catholijn M. Jonker. Representing human habits: towards a habit support agent. In *Proceedings of the 10th International workshop on Normative Multiagent Systems (NorMAS'16)*, LNCS. Springer, 2016. To appear.

17 M. Ros, M.P. Cuéllar, M. Delgado, and A. Vila. Online recognition of human activities and adaptation to habit changes by means of learning automata and fuzzy temporal windows. *Information Sciences*, 220:86–101, 2013. Online Fuzzy Machine Learning and Data Mining. `doi:10.1016/j.ins.2011.10.005`.

18 Leila S. Shafti, Pablo Alfonso Haya, Manuel García-Herranz, and Xavier Alamán. Personal ambient intelligent reminder for people with cognitive disabilities. In José Bravo, Ramón Hervás, and Marcela Rodríguez, editors, *Ambient Assisted Living and Home Care*, pages 383–390, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

19 John Thangarajah, James Harland, David Morley, and Neil Yorke-Smith. Operational behaviour for executing, suspending, and aborting goals in bdi agent systems. In *Proceedings of the 8th International Conference on Declarative Agent Languages and Technologies VIII*, DALT'10, pages 1–21, Berlin, Heidelberg, 2011. Springer-Verlag. URL: `http://dl.acm.org/citation.cfm?id=1996758.1996760`.

20 John Thangarajah, Lin Padgham, and Michael Winikoff. Detecting & exploiting positive goal interaction in intelligent agents. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 401–408, New York, NY, USA, 2003. ACM. `doi:10.1145/860575.860640`.

21 M. Birna van Riemsdijk, Louise Dennis, Michael Fisher, and Koen V. Hindriks. Agent reasoning for norm compliance: a semantic approach. In *Proceedings of the twelfth international joint conference on autonomous agents and multiagent systems (AAMAS'13)*, pages 499–506. IFAAMAS, 2013.

22 M. Birna van Riemsdijk, Louise Dennis, Michael Fisher, and Koen V. Hindriks. A semantic framework for socially adaptive agents: Towards strong norm compliance. In *Proceedings of the fourteenth international joint conference on autonomous agents and multiagent systems (AAMAS'15)*. IFAAMAS, 2015.

23 M. Birna van Riemsdijk, Catholijn M. Jonker, and Victor Lesser. Creating socially adaptive electronic partners: Interaction, reasoning and ethical challenges. In *Proceedings of the fourteenth international joint conference on autonomous agents and multiagent systems (AAMAS'15)*, pages 1201–1206. IFAAMAS, 2015.