Reasoning with 'And Then' and 'While'*

(Extended Abstract)

Suman Roy
Honeywell Technology Solutions Lab. (HTSL)
151/1, Doraisanipalya, Bannerghatta Road, Bangalore 560 076, India
suman.roy@honeywell.com

Guido Sciavicco
Department of Information and Communications Engineering
University of Murcia
Murcia E-30100 - Campus de Espinardo, Spain
guido@um.es

Abstract

Interval-based temporal logics are natural frameworks for modeling a number of problems from various areas of computer science such as artificial intelligence, natural language processing, temporal databases and formal specification. Quite a few interval-based temporal logics became popular in recent years, such as Venema's CDT logic, Halpern and Shoham's HS logic, Moszkowski's ITL and its propositional version, and Goranko, Montanari, and Sciavicco's PNL. In this work we introduce a new propositional interval-based temporal logic called CW, which can be considered an extension of the propositional fragment of Moszkowski's ITL evaluated over different (parallel) lines, and which is particularly adapt for expressing natural language sentences. We study the logic CW and develop a (non-terminating) sound and complete deduction system based on tableaux for it.

1 Introduction

Modal logics of time, both at the propositional and the first-order level, have found a wide variety of applications in computer science. Such formalisms constitute a natural framework for reasoning about action and change, temporal qualitative reasoning, planning, and natural language processing; moreover, temporal logics have been successfully used in the areas of specification and automatic verification of programs, and temporal (and spatio-temporal) databases. But, while temporal logics based on points have been deeply studied, interval-based ones have received less attention in the literature. One reason of such a disparity is that interval temporal logics exhibit, in general, a bad computational behavior, due to their high expressive power and versatility; moreover, from the classical point of view, interval-based temporal languages correspond to fragments of first-order languages with *binary* predicates, while point-based ones correspond to fragments with *unary* predicates.

Earlier work on propositional interval-based temporal logics include Venemas's CDT [Ven91, GMSS06], Goranko, Montanari, and Sciavicco Propositional Neighborhood Logics [GMS03b], Moszkowski's PITL [Mos83], later studied by Bowman and Thompson [BT03], Rosner and Pnueli's point-based temporal logic DUXC with *chop* (also know as *and then*) [RP86], Barua, Roy and Zhou's first-order Neighborhood Logic [BRZ00], and Dutertre's first-order generalization of PITL [Dut95].

In both PITL and CDT the language includes the binary interval operator *chop*. Given two intervals $[d_0, d_1]$ and $[d_1, d_2]$ over a linearly ordered set, the *chop* operator allows one to express properties of both the two intervals and the interval $[d_0, d_2]$ (the sequencing of the two intervals) at the same time. As noticed in [Ven91] by Venema, and in [LR00], the *chop* operator presents interesting applications and analogies with natural language expressions. In [LR00] a new temporal operator called *while* has been introduced, and denoted by $||$; it allows one to consider different 'parallel' lines of evaluation of the same interval. For example, the formula $p||q$ is interpreted as $p \ \ while \ \ q$, where p and q are two propositional letters interpreted over

*This work does not necessarily reflect the research policies of HTSL, nor does it discourage such kind of work among its employees

14th International Symposium on Temporal Representation and Reasoning (TIME'07
0-7695-2836-8/07 $20.00 © 2007 **IEEE**

intervals. In this paper, we concentrate on the combined logic CW (Chop-While), which is a propositional interval-based temporal logic with different lines of evaluation, and featuring the operators C, $||$, and the modal constant π for point-intervals. CW is interpreted over a sequence of parallel time lines, each one of which can be viewed as a copy of the same linearly ordered set. As we shall see, the logic CW allows one to express natural languages sentences in a very easy way, and thus it facilitates immediate applications in understanding and reasoning about natural languages.

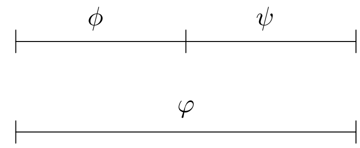

Figure 1. A pictorial representation of the formula $(\phi \frown \psi)||\varphi$

2 Syntax and Semantics of Chop-While

In this section we study a new interval logic called *Chop-While* (CW) which can be viewed as the generalization of C [Mos83, RS07, BT03, Dut95] to different (parallel) lines. This logic, with a slightly different semantics, has been originally introduced in [LR00].

The language of CW features a set of propositional letters $\mathcal{AP} = \{p, q, \ldots\}$, the classical operators $\neg, \vee$ (the remaining ones can be considered as abbreviations), and the binary modalities C, $||$, in addition to the modal constant π. Well-formed *formulas* can be obtained by the following abstract grammar:

$$\phi = \pi \mid p \mid \neg\phi \mid \phi \vee \psi \mid \phi C \psi \mid \phi || \psi,$$

where $p \in \mathcal{AP}$.

The models of CW are given in terms of parallel *evaluation* lines. We consider a set of linearly ordered set $\langle D_1, < \rangle$, $\langle D_2, < \rangle$, ..., $\langle D_n, < \rangle$. We call CW-*structure* the set $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$. The linearly ordered sets are pairwise isomorphic, and theirs elements are denoted here by $d_{i_1}, d_{j_1}, \ldots \in D_1, \ldots$, $d_{i_n}, d_{j_n}, \ldots \in D_n$, and so on; thus, for example, the element d_{3_4} is the third element of the fourth domain. For a given D_k, consider the set of all intervals that can be built on it, denote it by, $\mathbb{I}(D_k)$; let $\mathcal{I}(\mathcal{D}) = \bigcup_{h=1}^{n} \mathbb{I}(D_h)$. A CW-*model* is a tuple $M_{CW} = \langle \mathcal{D}, \mathcal{I}(\mathcal{D}), V_{CW} \rangle$, where V_{CW} is a valuation function such that, for each linearly ordered set $D_k \in \mathcal{D}$ and interval $[d_{i_k}, d_{j_k}]$, it assigns a truth value to each propositional variable $p \in AP$. In terms of classical modal logic, we can define Kripke-style models of CW as follows. Let A be the relation $A \subseteq \mathcal{I}(D) \times \mathcal{I}(D) \times \mathcal{I}(D)$, where $D \in \mathcal{D}$ is defined in such a way that for all intervals $[d_{i_k}, d_{j_k}], [d_{i'_k}, d_{j'_k}], [d_{i''_k}, d_{j''_k}] \in \mathcal{I}(D)$, the triple $([d_{i_k}, d_{j_k}], [d_{i'_k}, d_{j'_k}], [d_{i''_k}, d_{j''_k}]) \in A$ if and only if $d_{i_k} = d_{i''_k}$, $d_{j'_k} = d_{j''_k}$, and $d_{j_k} = d_{i'_k}$ (this is exactly Venema's *chop* relation [Ven91]), and let $\Pi \subseteq \mathcal{I}(D)$ be a predicate defined in such a way that $[d_{i_k}, d_{j_k}] \in \Pi$ if and only if $d_{i_k} = d_{j_k}$. We also define the relation $P \subset \mathcal{I}(D) \times \mathcal{I}(\mathcal{D})$ defined as $P([d_{i_h}, d_{j_h}], [d_{i_k}, d_{j_k}])$ if and only if $h \neq k$. Thus, a CW-model can be viewed as a tuple $M_{CW} = \langle \mathcal{I}(\mathcal{D}), A, P, \Pi, V_{CW} \rangle$. The intuitive picture of the *chop* operator is shown in Figure 1. The *truth* of a given CW-formula ϕ is given by the following clauses:

- $M_{CW}, [d_{i_k}, d_{j_k}] \Vdash \pi$ iff for each $D_h \in \mathcal{D}$, $[d_{i_h}, d_{j_h}] \in \Pi$;
- $M_{CW}, [d_{i_k}, d_{j_k}] \Vdash p$ iff $p \in V_{CW}([d_{i_k}, d_{j_k}])$;
- $M_{CW}, [d_{i_k}, d_{j_k}] \Vdash \neg\psi$ iff it is not the case that $M_{CW}, [d_{i_k}, d_{j_k}] \Vdash \psi$;
- $M_{CW}, [d_{i_k}, d_{j_k}] \Vdash \psi \vee \varphi$ iff $M_{CW}, [d_{i_k}, d_{j_k}] \Vdash \psi$ or $M_{CW}, [d_{i_k}, d_{j_k}] \Vdash \varphi$;
- $M_{CW}, [d_{i_k}, d_{j_k}] \Vdash \phi||\psi$ iff $M_{CW}, [d_{i_k}, d_{j_k}] \Vdash \phi$ and there exists some $D_h \in \mathcal{D}$ such that $P([d_{i_k}, d_{j_k}], [d_{i_h}, d_{j_h}])$ and $M_{CW}, [d_{i_h}, d_{j_h}] \Vdash \psi$;
- $M_{CW}, [d_{i_k}, d_{j_k}] \Vdash \phi C \psi$ iff there are two intervals $[d_{l_k}, d_{m_k}]$ and $[d'_{l_k}, d'_{m_k}]$ such that $A([d_{l_k}, d_{m_k}], [d'_{l_k}, d'_{m_k}], [d_{i_k}, d_{j_k}])$, $M, [d_{l_k}, d_{m_k}] \Vdash \phi$, and $M, [d'_{l_k}, d'_{m_k}] \Vdash \psi$.

Notice that, since intervals should behave the same way over all parallel lines, whenever we encounter a π over some interval $[d_{i_k}, d_{j_k}]$, *all* the intervals which are parallel to it (including $[d_{i_k}, d_{j_k}]$ itself) must be point-intervals. The concepts of *satisfiability, validity in a model*, and *validity* are defined in the usual way.

In the language of CW, we can define the *during* operator $\Diamond\phi$ as $\top C(\phi \vee (\top||\phi))C\top$, which says that ϕ holds somewhere within an interval, its dual operator (*throughout*) $\Box\phi = \neg\Diamond\neg\phi$, and the dual of the parallel operator: $p\ ||_i\ q = p \wedge \neg(\top||\neg q)$, which means that p is true on the current line and q is true on any other line, if any.

It is simple to see that the satisfiability problem for CW is not decidable. Indeed, in [Lod00] it has been shown that

Character's name	Character's reasoning
Cornelia	$(J \wedge S \wedge CO \wedge (J.drunk\ C\ S.shot.at\ C J.hysterical))\ C (J \wedge CO \wedge \neg S \wedge (J.hysterical\ C\ J.sedated)) \Vert (S \wedge S.hurt))\ C (S \wedge CO \wedge \neg J \wedge ((S.hurt\ C\ S.bullet\ C\ S.sedated) \Vert (J \wedge J.sedated))\ C ((\neg CO \wedge S \wedge S.sedated) \Vert (\neg CO \wedge J \wedge J.sedated)) \rightarrow \Box((J \wedge CO) \vee (J \wedge J.sedated)) \wedge \Box((S \wedge CO) \vee (S \wedge (S.hurt \vee S.sedated)))$
Poirot	$(J \wedge S \wedge CO \wedge (J.drunk\ C\ S.shot.at\ C\ J.hysterical))\ C((J \wedge CO \wedge \neg S \wedge (J.hysterical\ C\ J.sedated)) \Vert_i (S \wedge (\neg S.hurt\ C\ S.murder\ C\ S.shoots.leg\ C\ S.hurt))\ C((S \wedge CO \wedge \neg J \wedge (S.hurt\ C\ S.bullet\ C\ S.sedated)) \Vert_i (J \wedge J.sedated)) C\ ((\neg CO \wedge S \wedge S.sedated) \Vert_i (\neg CO \wedge J \wedge J.sedated))))$

Figure 2. Cornelia and Poirot's reasoning put in a logical formula.

the logic C (with *chop* and π only) is not decidable over dense linearly ordered sets. When C is interpreted in the class of all linearly ordered sets, the denseness property can be defined by a valid formula, and, since it is possible to express in C (a sort of) *universal modality*, it is easy to show that the satisfiability problem for C is not decidable. Thus, we have that also the satisfiability problem for CW is not decidable.

2.1 Using CW

In this section, we show how CW can be used to formalize a natural language expression by using a non-trivial example.

Example 1 *We borrow the next example from [LR00].*

> In the Agatha Christie novel *Death on the Nile* [Chr56], the two main suspects are Simon, husband of the murdered woman, and Jackie, his ex-girlfriend. On the evening of the murder, Cornelia, a witness, is with Simon and Jackie. Jackie gets drunk and shoots at Simon. He falls down. Simon asks Cornelia to bring a doctor, and to take the hysterical Jackie with her. Cornelia does so. The doctor gives Jackie a sedative and she goes to sleep. Cornelia brings the doctor where Simon is. The doctor pulls out the bullet from Simon's leg and gives him sedative, he goes to sleep. By the morning the murder has taken place. Cornelia reasons: while she was with Jackie (and not with Simon), he was shot and hurt, and later in the night, Simon was sedated, otherwise she was always with Simon. While she was with Simon (and not with Jackie), he was sedated, otherwise she was always with Jackie. So both Jackie and Simon have alibis. Hercule Poirot, the detective, notices that Simon was alone for the interval that Cornelia and Jackie went for the doctor. Under the assumption that he was hurt, he could not do the murder. But if he was not shot, he could do it. During the second interval that Cornelia was with Simon, the doctor removed the bullet from his leg, he must have been shot by then. That is still consistent if Jackie's shot did not hit him, his fall was faked, and while he was alone, he did the murder and then shot himself at the leg.

The logic CW is powerful enough to express both Cornelia's version of the facts ($\phi_{Cornelia}$), and Poirot's reasoning (ϕ_{Poirot}), as shown in Figure 2 above. Clearly, if the Poirot is right, it must be the case that the following formula is valid:

$$\phi_{Cornelia} \rightarrow \phi_{Poirot}.$$

3 Automatic Entailment in CW: a Tableaux Method

In this section we propose a tableau method for the logic CW. The method can be seen as an adaptation of the one used for PNL logics [GMS03a].

3.1 Basic Definitions

We assume the reader is familiar with the standard notions of *finite tree*, *root* of a tree, *successor* of a node $\mathbf{n}$, *leaf* node (i.e., a node with no successors), and *path* of nodes. During the construction of a tableaux for a given formula, we will consider a single branch (unless otherwise specified). We define the *height* of a node $\mathbf{n}$ as the maximum length (number of edges) of a path from $\mathbf{n}$ to a leaf. The expression $\mathbf{n} \prec \mathbf{n}'$ (resp. $\mathbf{n} \preceq \mathbf{n}'$) means that, for two nodes $\mathbf{n}, \mathbf{n}'$ belonging to the same branch, that the height of $\mathbf{n}$ is less than (resp. less than or equal to) the height of $\mathbf{n}'$.

We consider a collection $\langle C_1, < \rangle, \langle C_2, < \rangle, \ldots, \langle C_n, < \rangle$ of linearly ordered sets, and denote $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$.

14th International Symposium on Temporal Representation and Reasoning (TIME'07
0-7695-2836-8/07 $20.00 © 2007 IEEE

ϕ	Expansion rule for the node $\mathbf{n}$ and the branch B
$\neg\neg\psi$	expand the branch to $B \cdot \mathbf{n_0}$, with $\rho(\mathbf{n_0}) = ((\psi, [c_{i_k}, c_{j_k}]), \mathcal{C}_B, u)$
$\psi_0 \wedge \psi_1$	expand the branch to $B \cdot \mathbf{n_0} \cdot \mathbf{n_1}$, with $\rho(\mathbf{n_0}) = ((\psi_0, [c_{i_k}, c_{j_k}]), \mathcal{C}_B, u)$ and $\rho(\mathbf{n_1}) = ((\psi_1, [c_{i_k}, c_{j_k}]), \mathcal{C}_B, u)$
$\neg(\psi_0 \wedge \psi_1)$	expand the branch to $B \cdot \mathbf{n_0}\|\mathbf{n_1}$, with $\rho(\mathbf{n_0}) = ((\neg\psi_0, [c_{i_k}, c_{j_k}]), \mathcal{C}_B, u)$ and $\rho(\mathbf{n_1}) = ((\neg\psi_1, [c_{i_k}, c_{j_k}]), \mathcal{C}_B, u)$
$\neg(\psi_0 C \psi_1)$	take the least $c_{h_k} \in C_k$ ($c_{h_k} \in [c_{i_k}, c_{j_k}]$) which has not been yet used to expand the node $\mathbf{n}$ on B, and expand the branch to $B \cdot \mathbf{n_0}\|\mathbf{n_1}$, with $\rho(\mathbf{n_0}) = ((\neg\psi_0, [c_{i_k}, c_{h_k}]), \mathcal{C}_B, u)$ and $\rho(\mathbf{n_1}) = ((\neg\psi_1, [c_{h_k}, c_{j_k}]), \mathcal{C}_B, u)$
$\neg(\psi_0 \|\| \psi_1)$	take some evaluation line $C_l \in \mathcal{C}$ ($C_l \neq C_k$) such that the interval $[c_{i_l}, c_{j_l}]$ has not been yet used to expand the node $\mathbf{n}$ on B, and expand the branch to $B \cdot \mathbf{n_0}\|\mathbf{n_1}$, with $\rho(\mathbf{n_0}) = ((\neg\psi_0, [c_{i_k}, c_{j_k}]), \mathcal{C}_B, u)$ and $\rho(\mathbf{n_1}) = ((\neg\psi_1, [c_{i_l}, c_{j_l}]), \mathcal{C}_B, u)$
$\psi_0 C \psi_1$	expand the branch to $B \cdot (\mathbf{n_i} \cdot \mathbf{m_i})\| \ldots \|(\mathbf{n_j} \cdot \mathbf{m_j})\|(\mathbf{n'_i} \cdot \mathbf{m'_i})\| \ldots \|(\mathbf{n'_{j-1}} \cdot \mathbf{m'_{j-1}})$, where: (i) for all $c_{h_k} \in [c_{i_k}, c_{j_k}]$, $\rho(\mathbf{n_k}) = ((\psi_0, [c_{i_k}, c_{h_k}]), \mathcal{C}_B, u)$ and $\rho(\mathbf{m_k}) = ((\psi_1, [c_{h_k}, c_{j_k}]), \mathcal{C}_B, u)$, and (ii) for all $i \leq l \leq j-1$, let $\mathcal{C}'$ be the set if linearly ordered sets obtained by inserting a new element c between c_{l_z} and c_{l+1_z} (for all $1 \leq z \leq k$), $\rho(\mathbf{n'_k}) = ((\psi_0, [c_{i_k}, c]), \mathcal{C}', u)$, and $\rho(\mathbf{m'_k}) = ((\psi_1, [c, c_{j_k}]), \mathcal{C}', u)$;
$(\psi_0 \|\| \psi_1)$	expand the branch to $B \cdot \mathbf{n_0} \cdot (\mathbf{n_1}\| \ldots \|\mathbf{n_{n-1}}\|\mathbf{n'})$, where: (i) for all $1 \leq l \leq n, l \neq k$, $\rho(\mathbf{n_l}) = ((\psi_1, [c_{i_l}, c_{j_l}]), \mathcal{C}_B, u)$, (ii) $\rho(\mathbf{n_0}) = ((\psi_0, [c_{i_k}, c_{j_k}]), \mathcal{C}_B, u)$, and (iii) $\rho(\mathbf{n'}) = ((\psi_1, [c_{i_{n+1}}, c_{j_{n+1}}]), \mathcal{C}', u)$, where $\mathcal{C}' = \mathcal{C}_B \cup C_{n+1}$, and C_{n+1} is any linear domain isomorphic to C^*, where $C^* \in \mathcal{C}_B$ and $\|C^*\|$ is maximal.

Table 1. Branch-expansion rules

Each one of the linearly ordered sets is composed of the same elements (the sets C_i are pairwise isomorphic), which are denoted here as $c_{i_1}, c_{j_1}, \ldots \in C_1$, $c_{i_2}, c_{j_2}, \ldots \in C_2$, and so on.

Definition 2 *Let* $\mathbf{n}$ *be a node in a branch* B *of a tree* $\mathcal{T}$.

- *An* annotated formula *is a pair* $(\phi, [c_{i_k}, c_{j_k}])$*, where* $\phi \in$ *CW and* $c_{i_k}, c_{j_k} \in C_k, 1 \leq k \leq n$*, where* $|\mathcal{C}| = n$.
- *The* annotation $\rho(\mathbf{n})$ *is a triple* $((\phi, [c_{i_k}, c_{j_k}]), \mathcal{C}, u_{\mathbf{n},B_1}, \ldots, u_{\mathbf{n},B_q})$*, where* $(\phi, [c_{i_k}, c_{j_k}])$ *is an annotated formula and for each* $1 \leq r \leq q$, $u_{\mathbf{n},B_r} \in \{0, 1\}$ *is the* expansion flag, i.e., *it associates the values* 0 *or* 1 *with every branch* B_i *in* $\mathcal{T}$ *containing* $\mathbf{n}$.
- $\mathbf{n}$ *is said to be* active *if and only if its flag is 0 for at least one branch* B' *to which* $\mathbf{n}$ *belongs.*
- *An* annotated tree *is a tree in which every node has an annotation* $\rho(\mathbf{n})$.

For a node $\mathbf{n}$ and a branch B_r containing it, the flag is 0 when the node has not been expanded yet over B_r, and 1 otherwise, except for the universal cases $\neg C$ and $\neg||$, in which the flag always remains 0. For sake of simplicity, we will often assume an interval $[c_{i_k}, c_{j_k}]$, from the line C_k, to consist of the elements $c_{i_k} < c_{(i+1)_k} < \ldots < c_{j_k}$. For any branch B in a annotated tree, we denote by $\mathcal{C}_B$ the ordered set in the annotation of the leaf of B, and for any node $\mathbf{n}$ in a annotated tree, we denote by $\Phi(\mathbf{n})$ the formula in its annotation. If B is a branch, then $B \cdot \mathbf{n}$ denotes the result of the expansion of B with the node $\mathbf{n}$ (addition of an edge connecting the leaf of B to $\mathbf{n}$). Similarly, $B \cdot \mathbf{n_1} \mid \ldots \mid \mathbf{n_k}$ denotes the result of the expansion of B with k immediate successor nodes $\mathbf{n_1}, \ldots, \mathbf{n_k}$ (which produces k branches extending B). A tableau for CW will be defined as a distinguished annotated tree. It is worth noticing that $\mathcal{C}$ remains finite throughout the construction of the tableau.

3.2 Tableaux and Expansion Rule

Now we are ready to define the tableaux for CW and the expansion rule for them.

Definition 3 *Given an annotated tree* $\mathcal{T}$*, a branch* B *in* $\mathcal{T}$*, and a node* $\mathbf{n} \in B$ *such that* $\rho(\mathbf{n}) = ((\phi, [c_{i_k}, c_{j_k}]), \mathcal{C}, u)$*, with* $u_{\mathbf{n},B} = 0$*, the* branch-expansion rule *for* B *and* $\mathbf{n}$ *is defined as in Table 1. In all the considered cases,* $u_{\mathbf{n'},\mathbf{B'}} = 0$ *for all new pairs* $(\mathbf{n'}, B')$ *of nodes and branches, and* u *switches from* 0 *to* 1 *in all cases except for* $\neg C$ *and* $\neg||$.

Definition 4 *Let* B *a branch of a tableau* $\mathcal{T}$.

- *The branch-expansion rule is* applicable *to a node* $\mathbf{n}$ *on* B *if the node is active on* B *and the application of the rule generates at least one successor node with a new annotated formula;*

14th International Symposium on Temporal Representation and Reasoning (TIME'07
0-7695-2836-8/07 $20.00 © 2007 IEEE

- *A branch B is* closed *if some of the following conditions holds: (i) there are two nodes $\mathbf{n}, \mathbf{n}' \in B$ such that $\rho(\mathbf{n}) = ((\psi, [c_{i_k}, c_{j_k}]), \mathcal{C}, u)$ and $\rho(\mathbf{n}') = ((\neg\psi, [c_{i_k}, c_{j_k}]), \mathcal{C}', u')$ for some formula ψ and $c_{i_k}, c_{j_k} \in C \cap C'$; (ii) there is a node $\mathbf{n}$ such that $\rho(\mathbf{n}) = ((\pi, [c_{i_k}, c_{j_k}]), \mathcal{C}, u)$ and $c_{i_k} \neq c_{j_k}$; (iii) there is a node $\mathbf{n}$ such that $\rho(\mathbf{n}) = ((\neg\pi, [c_{i_k}, c_{j_k}]), \mathcal{C}, u)$ and $c_{i_k} = c_{j_k}$; (iv) there are two nodes $\mathbf{n}, \mathbf{n}' \in B$ such that $\rho(\mathbf{n}) = ((\psi, [c_{i_k}, c_{j_k}]), \mathcal{C}, u)$ and $\rho(\mathbf{n}') = ((\psi', [c_{i_h}, c_{j_h}]), \mathcal{C}', u')$ for some formulas ψ, ψ', and $c_{i_k} = c_{j_k}$ (resp., $c_{i_k} \neq c_{j_k}$) and $c_{i_h} \neq c_{j_h}$ (resp., $c_{i_h} = c_{j_h}$); otherwise, the branch is* open *(Clearly, a tableau for CW is* closed *if and only if every branch in it is closed, otherwise it is* open*);*

- *The* branch-expansion strategy *for a branch B in an annotated tree $\mathcal{T}$ consists of applying the branch-expansion rule to a branch B only if it is open, and, in such a case, applying it to the first active node one encounters moving from the root to the leaf of B to which the branch-expansion rule is applicable (if any);*

- *An* initial tableau *for a given formula $\phi \in$ CW is the finite annotated tree $\mathcal{T}$ composed of an empty root and two nodes $\mathbf{n_0}$ and $\mathbf{n_1}$ such that $\rho(\mathbf{n_0}) = ((\phi, [c_{0_1}, c_{0_1}]), \mathcal{C}, 0)$ and $\rho(\mathbf{n_1}) = ((\phi, [c_{0_1}, c_{1_1}]), \mathcal{C}', 0)$, where $\mathcal{C} = \{C_1\}$, $C_1 = \{c_{0_1}\}$, $\mathcal{C}' = \{C'_1\}$, $C'_1 = \{c_{0_1}, c_{1_1}\}$, and $c_{0_1} < c_{1_1}$. A* tableau *for a given formula $\phi \in$ CW is any finite annotated tree isomorphic to a finite annotated tree $\mathcal{T}$ obtained by expanding the initial tableau for ϕ through successive applications of the branch-expansion strategy to the existing branches.*

Theorem 5 *If $\phi \in$ CW and a tableau $\mathcal{T}$ for ϕ is closed, then ϕ is not satisfiable. Moreover, if ϕ is a valid CW-formula, then there is a closed tableau for $\neg\phi$.*

3.3 An Example

Finally, we consider a simple example taken from the context of natural language, and we analyze it by means of the tableaux-based method.

Example 6 *Guido works in Murcia (Spain), and Suman works in Bangalore (India). They have been working on the same paper. It is always true that when a paper is on work (draft), it is not a final version. During a given period of time, the Guido has been working on a draft version of the paper, while Suman has been working on a draft and got a final version of the paper. Is this situation consistent?*

In order to formalize the above situation, we will use the following proposition letters: the symbol w denotes that the paper is on work, dr denotes that the paper is still a draft, and fi that it is a final version. As we see in Figure 3, there is a closed tableau for the conjunction of the formulas that formalize the statement in the example, showing that the described situation is not consistent. For sake of simplicity, we have omitted some of the branches, and we used rules for $||_i$ and $\Box$ which do not appear in Definition 8, but can be easily deducted from the semantics of the operators.

Acknowledgement: The authors would like to thank Kamal Lodaya for his many useful suggestions and comments about this work (he has indeed worked on some of the ideas in this paper), and Ian Hodkinson for pointing out an important mistake in the first version of the paper. The first author is grateful to the management of HTSL for constant support and encouragement during this work. This research has been partially founded by the Spanish MEC project IDEATIO, no. TIN2006-15460-C04-01.

References

[BRZ00] R. Barua, S. Roy, and C. Zhou. Completeness of neighbourhood logic. *Journal of Logic and Computation*, 10(2):271–295, 2000.

[BT03] H. Bowman and S. Thompson. A decision procedure and complete axiomatization of finite interval temporal logic with projection. *Journal of Logic and Computation*, 13(2):195–239, 2003.

[Chr56] A. Christie. Death on the nile. Collins (London), 1956.

[Dut95] B. Dutertre. Complete proof systems for first order interval temporal logic. In *Proc. of the 10th International Symposium on Logic in Computer Science*, pages 36–43, 1995.

[GMS03a] V. Goranko, A. Montanari, and G. Sciavicco. A general tableau method for propositional interval temporal logic. In *Proc. of TABLEAUX 2003: 12th Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, volume 2796 of *LNAI*, pages 102–116, Rome, Italy, September 2003. Springer.

[GMS03b] V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood temporal logics. *Journal of Universal Computer Science*, 9(9):1137–1167, 2003.

$$((\Box(w \to dr)||_i \Box(w \to dr), [c_{0_1}, c_{1_1}]), \{\{c_{0_1} < c_{1_1}\}\}, 0)$$

$$((\Box(dr \to fi)||_i \Box(dr \to fi), [c_{0_1}, c_{1_1}]), \{\{c_{0_1} < c_{1_1}\}\}, 0)$$

$$((w||w\ C\ fi), [c_{0_1}, c_{1_1}]), \{\{c_{0_1} < c_{1_1}\}\}, 1)$$

$$(w, [c_{0_1}, c_{1_1}]), \{\{c_{0_1} < c_{1_1}\}\}, 1)$$

$$((w\ C\ fi, [c_{0_2}, c_{1_2}]), \{\{c_{0_1} < c_{1_1}\}, \{c_{0_2} < c_{1_2}\}\}, 1)$$

$$((w, [c_{0_2}, c_{2_2}]), \{\{c_{0_1} < c_{2_1} < c_{1_1}\}, \{c_{0_2} < c_{2_2} < c_{1_2}\}\}, 1)$$

$$((fi, [c_{2_2}, c_{1_2}]), \{\{c_{0_1} < c_{2_1} < c_{1_1}\}, \{c_{0_2} < c_{2_2} < c_{1_2}\}\}, 1)$$

$$((\neg w, [c_{2_2}, c_{1_2}], \{\{c_{0_1} < c_{2_1} < c_{1_1}\}, \{c_{0_2} < c_{2_2} < c_{1_2}\}\}, 1)$$

$$\times$$

$$(dr, [c_{2_2}, c_{1_2}], \{\{c_{0_1} < c_{2_1} < c_{1_1}\}, \{c_{0_2} < c_{2_2} < c_{1_2}\}\}, 1)$$

$$((\neg dr, [c_{2_2}, c_{1_2}]), \{\{c_{0_1} < c_{2_1} < c_{1_1}\}, \{c_{0_2} < c_{2_2} < c_{1_2}\}\}, 1)$$

$$\times$$

$$((\neg fi, [c_{2_2}, c_{1_2}]), \{\{c_{0_1} < c_{2_1} < c_{1_1}\}, \{c_{0_2} < c_{2_2} < c_{1_2}\}\}, 1)$$

$$\times$$

Figure 3. A closed tableau for Example 6.

[GMSS06] V. Goranko, A. Montanari, G. Sciavicco, and P. Sala. A general tableau method for propositional interval temporal logics: theory and implementation. *Journal of Applied Logic*, 4(3):305–330, 2006.

[Lod00] K. Lodaya. Sharpening the undecidability of interval temporal logic. In *Proc. of 6th Asian Computing Science Conference*, volume 1961 of *LNCS*, pages 290–298. Springer, 2000.

[LR00] K. Lodaya and S. Roy. Lines, a while, and intervals. In *Proc. of ESSLLI Workshop on Many-Dimensional Logical Systems, Birmingham*, pages 1–8, 2000.

[Mos83] B. Moszkowski. *Reasoning about digital circuits*. Tech. rep. stan-cs-83-970, Dept. of Computer Science, Stanford University, Stanford, CA, 1983.

[RP86] R. Rosner and A. Pnueli. A choppy logic. In *First Annual IEEE Symposium on Logic In Computer Science*, pages 306–314. IEEE Computer Society Press, 1986.

[RS07] S. Roy and G. Sciavicco. Completeness of chop. In H.W. Guesguen, G. Ligozat, and R.V. Rodriguez, editors, *Proc. of the IJCAI-07 Workshop on Spatial and Temporal Reasoning*, pages 90–95, 2007.

[Ven91] Y. Venema. A modal logic for chopping intervals. *Journal of Logic and Computation*, 1(4):453–476, 1991.

14th International Symposium on Temporal Representation and Reasoning (TIME'07
0-7695-2836-8/07 $20.00 © 2007 IEEE