Journal of Universal Computer Science, vol. 5, no. 11 (1999), 743-764
submitted: 3/9/99, accepted: 23/11/99, appeared: 28/11/99 © Springer Pub. Co.

A Decision Method for Duration Calculus

Nathalie Chetcuti-Sperandio
(Institut de Recherche en Informatique de Toulouse, France
chetcuti@irit.fr)

Luis Fariñas del Cerro
(Institut de Recherche en Informatique de Toulouse, France
farinas@irit.fr)

Abstract: The Duration Calculus is an interval logic introduced for designing real-time systems. This calculus is able to capture important real-time problems like the specification of the behaviours of schedulers and classical examples like a gas burner. From a practical point of view an important challenge becomes to define automated proof procedures for this calculus. Since the propositional calculus is undecidable, we are interested then into isolating decidable fragments of this calculus. A first fragment was given and its decidability proved via regular languages.
In this paper we isolate another fragment and we define a tableau method which gives a natural procedure to decide whether a given formula is satisfiable.
Key Words: Duration Calculus, automated deduction, tableau method.
Category: D.2.1, F.4.1, I.2.3, I.2.4

1 Introduction

The Duration Calculus [Zhou, Hoare and Ravn 91] is an interval logic introduced for designing real-time systems. This calculus is able to capture important real-time problems like the specification of the behaviours of schedulers and classical examples like a gas burner. From a practical point of view an important challenge becomes to define automated proof procedures for this calculus. Since the propositional calculus is undecidable, we are interested then into isolating decidable fragments of this calculus. A first fragment was given [Zhou, Hansen and Sestoft 93] and its decidability proved via regular languages. In this paper we isolate another fragment and we define a tableau method which gives a natural procedure to decide whether a given formula is satisfiable. This fragment is not comparable with the one of [Zhou, Hansen and Sestoft 93] as both fragments strictly overlap each other without including or being included in the other one.
This paper is organized as follows: firstly we introduce propositional Duration Calculus, then the fragment we isolated and finally the tableau method we defined before concluding.

2 Propositional Duration Calculus

Duration Calculus [Zhou, Hoare and Ravn 91] is a temporal logic based on Interval Temporal Logic [Moszkowski 85] with an additional notion of *duration* in

a state, *i.e.* the duration for which a system stays in a particular state.
This section introduces Duration Calculus but the reader interested by a thorough presentation of Duration Calculus can refer to [Hansen and Zhou 97] for the logical aspects.

2.1 Syntax

Three types of expressions are considered:

1. *state expressions*, built from *state variables* and classical logic operators – the states of a system are modelled as state expressions, state variables being basic states –,
2. *terms*, defined as usual with a peculiar operator $\int$ – $\int \sigma$ being the duration for which the system is in state σ for a given temporal interval –,
3. *formulae* which are boolean formulae with a special operator called *chop* such that $\phi_1 ; \phi_2$ is true in a given interval if this interval can be split into two subintervals, ϕ_1 being true in the former and ϕ_2 in the latter.

2.2 Abbreviations

Apart from classical abbreviations, standard modal operators can be defined in this calculus:

- $\Diamond\phi \equiv \top ; \phi ; \top$
  ϕ is true for some subinterval of the current interval
- $\Box\phi \equiv \neg\Diamond(\neg\phi)$
  ϕ is true for every subinterval of the current interval

and more typical abbreviations:

- $\ell \equiv \int 1$
  ℓ is the length of the current interval
- $\lceil\rceil \equiv \ell = 0$
  the current interval is a point interval
- $\lceil\sigma\rceil \equiv (\int \sigma = \ell) \wedge \ell > 0$
  state expression σ is 1 almost everywhere on the current interval and this one is not a point interval.

2.3 Semantics

Time is modelled by the set of the real numbers. For each b, $e \in \mathbb{T}\mathrm{ime}$ such that $b \leq e$, the closed interval $[b, e]$ denotes the set $\{t \in \mathbb{T}\mathrm{ime} : b \leq t \leq e\}$ and $\mathbb{I}\mathrm{nt}$ will denote the set of intervals.

State expressions A state expression is interpreted as a boolean function over temporal points: $\mathbb{T}\mathrm{ime} \longrightarrow \{0, 1\}$. State variables are finitely varied, *i.e.* they may have only a finite number of discontinuities on a bounded interval, state expressions being thus Riemann-integrable on any bounded interval.

Terms A term is interpreted as a real-valued function: $\mathbb{I}\mathrm{nt} \longrightarrow \mathbb{R}$, giving for any temporal interval the duration of a given state expression.

Formulae A formula is interpreted as a boolean function over intervals: $\mathbb{I}\mathrm{nt} \longrightarrow \{false, true\}$.

Let $\mathcal{I}$ be an interpretation and let $[b, e]$ be a $\mathbb{T}$ime interval. Semantically, $M = (\mathcal{I}, [b, e])$ satisfies ϕ (or M is a model of ϕ), denoted $M \models \phi$, if $\mathcal{I}[\![\phi]\!]([b, e]) = true$. Likewise, $(\mathcal{I}, [b, e]) \not\models \phi \equiv \mathcal{I}[\![\phi]\!]([b, e]) = false$.

2.4 Proof System and Decidability

Michael R. Hansen and Zhou Chaochen give a sound and relatively complete axiomatization of Duration Calculus in [Hansen and Zhou 92].
Some subsets have been identified to be decidable while the others are undecidable [Zhou, Hansen and Sestoft 93].
In a decidable subset of Duration Calculus, a regular language $\mathcal{L}(\phi)$ can be generated from a formula ϕ. $\mathcal{L}(\phi)$ represents a set of strings corresponding to interpretations satisfying ϕ, such that ϕ is satisfiable if, and only if, $\mathcal{L}(\phi)$ is not empty. The emptyness of a regular language being decidable, so is the satisfiability of a formula.
A decision algorithm for checking the validity of formulae of a decidable Duration Calculus subset was implemented in [Skakkebæk and Sestoft 94].

In what follows, a new fragment of Duration Calculus is presented and a decidable tableau-based method is given.

3 Duration Calculus: an Interval Fragment

We will consider only a fragment of Duration Calculus; nevertheless this fragment is a decidable subset of Duration Calculus.
In this section we introduce the syntax and the semantics of our fragment then we show how to express a simplified version of a gas burner in our fragment.

3.1 Syntax

State expressions State expressions are defined by induction in the following way:

1. state variables are state expressions,
2. if s is a state expression then $\neg s$ is a state expression,
3. if σ_1 and σ_2 are state expressions then $\sigma_1 \wedge \sigma_2$ is a state expression.

Terms Terms are defined in the following way:

1. $\int 1$ is a term, denoted by ℓ,
2. if σ is a state expression then $\int \sigma$ is a term,
3. if k is a function symbol of arity 0 then k is a term, called *constant*.

A constant is interpreted as a real number.

Formulae Now, formulae are defined, still by induction, in the following way:

1. $\perp$ and $\top$ are formulae,
2. if k is a constant then $\ell \ \mathsf{S} \ k$, where $\mathsf{S} \in \{<, \leq, =, \neq, \geq, >\}$, is a formula,
3. if σ is a state expression then $\int \sigma = 0$ and $\int \sigma = \ell$ are formulae,
4. if ϕ_1 and ϕ_2 are formulae then $\phi_1 \vee \phi_2$, $\phi_1 \wedge \phi_2$ and $\phi_1 ; \phi_2$ are formulae.

The binary predicate symbols $\{<, \leq, =, \neq, \geq, >\}$ are interpreted as binary boolean functions over the set of the real numbers: $\mathbb{R}^2 \longrightarrow \{true, false\}$.

3.2 Semantics

An interpretation $\mathcal{I}$ associates with each state variable, a boolean function over $\mathbb{T}$ime: $\mathbb{T}\text{ime} \longrightarrow \{0, 1\}$.

State expressions The semantics of a state expression in interpretation $\mathcal{I}$ is a function: $\mathbb{T}\text{ime} \longrightarrow \{0, 1\}$, defined by induction in the following way:

- if s is a state variable then $\mathcal{I}[\![s]\!](t) = \mathcal{I}(s)(t)$,
- if σ is a state expression then $\mathcal{I}[\![\neg\sigma]\!](t) = 1 - \mathcal{I}[\![\sigma]\!](t)$,
- if σ_1 and σ_2 are state expressions then $\mathcal{I}[\![\sigma_1 \wedge \sigma_2]\!](t) = \mathcal{I}[\![\sigma_1]\!](t) \times \mathcal{I}[\![\sigma_2]\!](t)$.

Terms The semantics of a term in interpretation $\mathcal{I}$ is a function: $\mathbb{I}\text{nt} \longrightarrow \mathbb{R}$ defined in the following way:

- $\mathcal{I}[\![\ell]\!]([b, e]) = \int_b^e 1dt$,
- if σ is a state expression then $\mathcal{I}[\![\int \sigma]\!]([b, e]) = \int_b^e \mathcal{I}[\![\sigma]\!](t)$,
- $\mathcal{I}[\![k]\!]([b, e]) = k$.

Formulae The semantics of formulae in interpretation $\mathcal{I}$ is a function: $\mathbb{I}\text{nt} \longrightarrow \{false, true\}$ defined by induction in the following way:

- $\mathcal{I}[\![\perp]\!]([b, e]) = false$,
- $\mathcal{I}[\![\top]\!]([b, e]) = true$,
- if k is a constant and S is a binary predicate symbol such that $\mathsf{S} \in \{<, \leq, =, \neq, \geq, >\}$ then $\mathcal{I}[\![\ell \ \mathsf{S} \ k]\!]([b, e]) = true$ if, and only if, $\mathcal{I}[\![\ell]\!]([b, e]) \ \mathsf{S} \ k$,
- if σ is a state expression then $\mathcal{I}[\![\int \sigma = 0]\!]([b, e]) = true$ if, and only if, $\mathcal{I}[\![\int \sigma]\!]([b, e]) = 0$,

- if σ is a state expression then $\mathcal{I}[\![\int\sigma = \ell]\!]([b,e]) = true$ if, and only if, $\mathcal{I}[\![\int\sigma]\!]([b,e]) = \mathcal{I}[\![\ell]\!]([b,e])$,
- if ϕ_1 and ϕ_2 are formulae then $\mathcal{I}[\![\phi_1 \vee \phi_2]\!]([b,e]) = true$ if, and only if, $\mathcal{I}[\![\phi_1]\!]([b,e]) = true$ or $\mathcal{I}[\![\phi_2]\!]([b,e]) = true$,
- if ϕ_1 and ϕ_2 are formulae then $\mathcal{I}[\![\phi_1 \wedge \phi_2]\!]([b,e]) = true$ if, and only if, $\mathcal{I}[\![\phi_1]\!]([b,e]) = true$ and $\mathcal{I}[\![\phi_2]\!]([b,e]) = true$,
- if ϕ_1 and ϕ_2 are formulae then $\mathcal{I}[\![\phi_1 ; \phi_2]\!]([b,e]) = true$ if, and only if, $\exists m \in [b,e]$ such that $\mathcal{I}[\![\phi_1]\!]([b,m]) = true$ and $\mathcal{I}[\![\phi_2]\!]([m,e]) = true$.

Note 1. – In formulae, state expressions only appear within a $\int$.
- Let σ_1 and σ_2 be two state expressions and let σ_1 be σ_2 except for a finite number of points then $\forall [b,e] \in \mathbb{I}\mathrm{nt}$, $\int_b^e \sigma_1(t)dt = \int_b^e \sigma_2(t)dt$. Therefore for any state expression we can drop a finite number of discontinuity points without altering the truth value of formulae.

3.3 Example: a Gas Burner

The classical example on which Duration Calculus relies is the gas burner [Ravn, Rischel and Hansen 93]; in this section we show how to express the requirements for a simplified gas burner system in our fragment.
There are three requirements for a gas burner:

1. Gas must never leak for more than 4 seconds in any period of 30 seconds at most.
2. Heat request off must after 60 seconds result in the flame being off.
3. Heat request on must after 60 seconds result in the gas burning unless an ignition or flame failure occurred. An ignition failure occurs if the gas does not ignite within 0.5 second and a flame failure occurs if the gas stops burning while still flowing.

We shall consider the second as a time unit.
First we need state variables to express the gas burner state:

- $HeatRequest$, valued 1 when there is a heat request to the gas burner,
- Gas, valued 1 when the gas is flowing,
- $Ignition$, valued 1 when the ignition transformer tries to ignite the gas,
- and $Flame$, valued 1 when the gas is burning.

As far as the first requirement ($Req1$) is concerned, the gas is leaking when it is flowing and not burning; we consider that for any period of 30 seconds at most the gas may leak only once and for 4 seconds at most:

$$\ell > 30 \vee ((\int(\neg Gas \vee Flame) = \ell); \\ (\int(Gas \wedge \neg Flame) = \ell \wedge \ell \leq 4); (\int(\neg Gas \vee Flame) = \ell))$$

As far as the second requirement ($Req2$) is concerned, there are three possible cases:

- the time period is less than 60 seconds: $\ell \leq 60$,
- the period is composed of a time interval of 60 seconds at most followed by a time interval during which the gas is not burning:
  $\ell \leq 60; \lceil \neg Flame \rceil$
- Heat request is not off on the whole period; again we consider that heat request may be off only once in a given period:

$$(\lceil HeatRequest \rceil; (\int \neg HeatRequest = \ell); (\int HeatRequest = \ell)) \\ \vee ((\int HeatRequest = \ell); (\int \neg HeatRequest = \ell); \lceil HeatRequest \rceil)$$

Consequently the second requirement can be formalized as:

$$(\ell \leq 60) \vee (\ell \leq 60; \lceil \neg Flame \rceil) \\ \vee ((\lceil HeatRequest \rceil; (\int \neg HeatRequest = \ell); (\int HeatRequest = \ell)) \\ \vee ((\int HeatRequest = \ell); (\int \neg HeatRequest = \ell); \lceil HeatRequest \rceil))$$

As far as the third requirement ($Req3$) is concerned, there are five possible cases:

- the time period is less than 60 seconds: $\ell \leq 60$,
- the period is composed of a time interval of 60 seconds at most followed by a time interval during which the gas is burning: $\ell \leq 60; \lceil Flame \rceil$
- Heat request is not on on the whole period; again we consider that heat request may be on only once on a given period:

$$(\lceil \neg HeatRequest \rceil; (\int HeatRequest = \ell); (\int \neg HeatRequest = \ell)) \\ \vee ((\int \neg HeatRequest = \ell); (\int HeatRequest = \ell); \lceil \neg HeatRequest \rceil)$$

- an ignition failure can occur:
  $\top; (\lceil Gas \wedge Ignition \rceil \wedge ((\ell > 0.5 \wedge \int \neg Flame = \ell); \int Flame = \ell)); \top$
- a flame failure can occur:
  $\top; (\lceil Gas \rceil \wedge (\top; \lceil Flame \rceil; \lceil \neg Flame \rceil; \top)); \top$

Consequently the third requirement can be formalized as:

$$(\ell \leq 60) \vee (\ell \leq 60; \lceil Flame \rceil \\ \vee (\lceil \neg HeatRequest \rceil; (\int HeatRequest = \ell); (\int \neg HeatRequest = \ell))) \\ \vee (((\int \neg HeatRequest = \ell); (\int HeatRequest = \ell); \lceil \neg HeatRequest \rceil)) \\ \vee (\top; (\lceil Gas \wedge Ignition \rceil \wedge ((\ell > 0.5 \wedge \int \neg Flame = \ell); \int Flame = \ell)); \top) \\ \vee (\top; (\lceil Gas \rceil \wedge (\top; \lceil Flame \rceil; \lceil \neg Flame \rceil; \top)); \top)$$

Finally the gas burner must satisfy the three requirements:

$$Req1 \wedge Req2 \wedge Req3$$

4 Tableau Method

In this section we define a tableau method characterized by a set of extension rules. A step-by-step procedure applies these rules to a tree initially possessing an only node, called root, to which a single formula is attached; the tree thus constructed is called a *tableau*. If the original formula is consistent, there exists a model associated with its tableau satisfying it.

4.1 Tableau Method

Extension Rules We shall call *bounded* formula (respectively bounded state expression) an interval-stamped formula (respectively interval-stamped state expression), *e.g.* $F_{[b,e]}$.

An *extension rule* constructs a new tree T_{i+1} from a tree T_i by adding sons to a terminal node of T_i.
It is made up of a father and one or several sons, each node containing patterns of bounded formulae, bounded state expressions or interval bound constraints. If a childless node n contains bounded formulae, bounded state expressions or interval bound constraints matching all the rule father's ones then the rule applies to n.

Extension Rules for Duration Calculus

– stop rules:
 - stop_1:

$$\begin{array}{c} S, \bot_{[b',e']} \\ | \\ \mathsf{stop} \end{array}$$

 - stop_2:

$$\begin{array}{c} S, x_{[b',e']}, \neg x_{[b',e']} \\ | \\ \mathsf{stop} \end{array}$$

 - stop_3:

$$\begin{array}{c} S, b' < e', b' = e' \\ | \\ \mathsf{stop} \end{array}$$

– building rules:

- $\top$:

$$\begin{array}{c} S, \top_{[b',e']} \\ | \\ S, b' \leq e' \end{array}$$

- double negation: σ is a state expression.

$$\begin{array}{c} S, (\neg\neg\sigma)_{[b',e']} \\ | \\ S, \sigma_{[b',e']} \end{array}$$

- negation and: σ_1 and σ_2 are state expressions.

$$\begin{array}{ccc} & S, \neg(\sigma_1 \wedge \sigma_2)_{[b',e']} & \\ / & & \backslash \\ S, \neg\sigma_{1[b',e']} & & S, \neg\sigma_{2[b',e']} \end{array}$$

- interval length S: k is a constant.

$$\begin{array}{c} S, (\ell \;\mathsf{S}\; k)_{[b',e']} \\ | \\ S, b' \leq e', (e' - b') \;\mathsf{S}\; k \end{array}$$

where $\mathsf{S} \in \{<, \leq, =, \neq, \geq, >\}$

- null length: σ is a state expression.

$$\begin{array}{c} S, (\int \sigma = 0)_{[b',e']} \\ | \\ S, \neg\sigma_{[b',e']}, b' \leq e' \end{array}$$

- maximal length: σ is a state expression.

$$\begin{array}{c} S, (\int \sigma = \ell)_{[b',e']} \\ | \\ S, \sigma_{[b',e']}, b' \leq e' \end{array}$$

- or: x and y are formulae.

$$\begin{array}{ccc} & S, (x \vee y)_{[b',e']} & \\ / & & \backslash \\ S, x_{[b',e']} & & S, y_{[b',e']} \end{array}$$

- and:

$$S, (x \wedge y)_{[b',e']}$$

$$S, x_{[b',e']}, y_{[b',e']}$$

- subinterval sequence:

$$S, (x; y)_{[b',e']}$$

$$S, x_{[b',b_1]}, y_{[b_1,e']}$$

b_1 new interval bound

- interval union:

$$S, x_{[b',b_1]}, x_{[b_1,e']}$$

$$S, x_{[b',e']}$$

- interval disjunction:

$$S, x_{[b_1,e_1]}, \neg x_{[b_2,e_2]}$$

$$S, x_{[b_1,e_1]}, \neg x_{[b_2,e_2]}, e_1 \leq b_2 \qquad\qquad S, x_{[b_1,e_1]}, \neg x_{[b_2,e_2]}, e_2 \leq b_1$$

Rule interval disjunction does not apply if constraints $e_1 \leq b_2$ or $e_2 \leq b_1$ appear in the father.

Tableaux

Definition 1 (Tableau). *Let F be a formula and $\{T_0, T_1, \ldots, T_i, T_{i+1}, \ldots\}$ be a series of trees such that*

- *T_0 is a single node containing solely $F_{[b,e]}$, where b and e are generic interval bounds,*
- *T_{i+1} is obtained by applying an extension rule to T_i.*

If this series has a limit (note that any applicable rule is eventually applied), it is called a tableau *for formula F.*

Fact 4.11 *Any constraint belonging to a node also belongs to its* stop*-free descendants (if it has any).*

Firstly, stop rules are not concerned as they generate sons containing stop. Secondly, on the one hand each rule carries the bounded formulae, bounded state expressions and constraints not involved in its application from the father to the son(s), on the other hand no rule makes any constraint disappear.

Theorem 1 (Existence of a Tableau). *For every formula F, there exists a tableau for F.*

Proof. First the series of trees constructed to find a tableau for F is never empty since the first tree T_0 is a single node containing the only formula $F_{[b,e]}$, where b and e are generic interval bounds.
Secondly as each element of the series is built by applying an extension rule to its predecessor and as the application of an extension rule to a node creates two sons at most, it suffices to prove that each extension rule applies finitely often in a given branch to prove that the series of trees is finite, yielding thus a tableau.

- Rules stop_1, stop_2 and stop_3 stop the construction of the current branch, so they apply only once to a given node.
- Rules double negation, negation and, or, and and subinterval sequence generate bounded subformulae of the bounded formula to which they apply in the sons of the current node. Formulae being finite, the number of their subformulae is also finite, so rules double negation, negation and, or, and and subinterval sequence apply finitely often.
- Rule $\top$ just generates a constraint in the son of the current node, so it applies only once to a given node and a given bounded formula.
- Rule interval union applies to two bounded formulae and generates a single one in the son of the current node. The number of formulae being finite (as formulae are finite), rule interval union applies finitely often.
- Rule interval length S just generates two constraints in the son of the current node so it applies only once to a given node and a given bounded formula. The number of formulae being finite, this rule applies only finitely often.
- Rule null length generates the negation of a bounded state subexpression and a constraint. Formulae being finite, the number of their state subexpressions is also finite, hence rule null length applies finitely often.
- Rule maximal length generates a bounded state subexpression and a constraint. Formulae being finite, the number of their state subexpressions is also finite, thus rule maximal length applies finitely often.
- Rule interval disjunction carries both bounded state expressions to which it applies and generates a $b \leq e$-like constraint in both sons of the current node. Rule interval disjunction can not apply twice to the same pair of state expressions because of the application condition on rule interval disjunction and because of fact 4.11. Furthermore formulae being finite, the number of state expressions, hence the number of pairs of state expressions, is finite so rule interval disjunction applies finitely often.

Definition 2. – *A node is* preconsistent *if it is* stop*-free.*
- *A terminal node is* open *if it is preconsistent and the set of its constraints is consistent, otherwise it is* closed.
- *A tableau branch is* open *if its terminal node is open, otherwise it is* closed.
- *A tableau is open if one of its branches is open, otherwise it is closed.*

4.2 Definitions and Preliminary Lemmas

T-satisfiability

Definition 3 (Model). *A model for bounded expressions and constraints is a triplet* $(\mathtt{V}, \mathcal{I}, [b, e])$ *such that*

- $\mathtt{V} : \{interval\ bounds\} \rightarrow \mathbb{T}\mathrm{ime}$,
- $\mathcal{I} : \{state\ variables\} \longrightarrow (\mathbb{T}\mathrm{ime} \longrightarrow \{0, 1\})$,
- $[b, e]$ *is a* $\mathbb{T}\mathrm{ime}$*-interval.*

Definition 4 (T-satisfiability). T-satisfiability *is defined on bounded expressions and constraints in the following way:*

- *Let* $F_{[b', e']}$ *be a bounded formula. A model* $M = (\mathtt{V}, \mathcal{I}, [b, e])$ *t-satisfies* $F_{[b', e']}$ *if* $[\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b, e]$ *and* $(\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(e')]) \models F$,
- *Let* $x_{[b', e']}$ *be a bounded state expression. A model* $M = (\mathtt{V}, \mathcal{I}, [b, e])$ *t-satisfies* $x_{[b', e']}$ *if* $[\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b, e]$ *and* $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e')), \mathcal{I}(x)(t) = 1$,
- *Let* $b'\ \mathsf{R}\ e'$ *be an interval bound constraint with* $\mathsf{R} \in \{<, =, \leq\}$*. A model* $M = (\mathtt{V}, \mathcal{I}, [b, e])$ *t-satisfies* $b'\ \mathsf{R}\ e'$ *if* $\mathtt{V}(b') \in [b, e]$, $\mathtt{V}(e') \in [b, e]$ *and the constraint* $\mathtt{V}(b')\ \mathsf{R}\ \mathtt{V}(e')$ *is satisfied,*
- *Let* $(e' - b')\ \mathsf{S}\ k$ *be a quantitative interval bound constraint with* $\mathsf{S} \in \{<, \leq, =, \neq, \geq, >\}$ *and* $k \in \mathbb{R}$*. A model* $M = (\mathtt{V}, \mathcal{I}, [b, e])$ *t-satisfies* $(e' - b')\ \mathsf{S}\ k$ *if* $\mathtt{V}(b') \in [b, e]$, $\mathtt{V}(e') \in [b, e]$ *and the quantitative constraint* $(\mathtt{V}(e') - \mathtt{V}(b'))\ \mathsf{S}\ k$ *is satisfied.*

Lemma 1. *For all interval bounds* b' *and* e'*, bounded formula* $F_{[b', e']}$ *is t-satisfiable if, and only if, formula* F *is satisfiable.*

Proof.

- ($\Rightarrow$) Let b' and e' be interval bounds. $F_{[b', e']}$ is t-satisfiable
  $\Leftrightarrow \exists(\mathtt{V}, \mathcal{I}, [b, e]) \ :\ [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b, e]$ and $(\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(e')]) \models F$
  $\Rightarrow \exists(\mathcal{I}'', [b'', e'']) \ :\ (\mathcal{I}'', [b'', e'']) \models F$ (just take $\mathcal{I}'' = \mathcal{I}$, $b'' = \mathtt{V}(b')$ and $e'' = \mathtt{V}(e')$)
  $\Leftrightarrow F$ is satisfiable.
- ($\Leftarrow$) F is satisfiable
  $\Leftrightarrow \exists(\mathcal{I}, [b, e]) \ :\ (\mathcal{I}, [b, e]) \models F$
  $\Rightarrow \forall b', e' \in \{interval\ bounds\}, \exists(\mathtt{V}, \mathcal{I}'', [b'', e''])$: $[\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b'', e'']$ and $(\mathcal{I}'', [\mathtt{V}(b'), \mathtt{V}(e')]) \models F$ (just take $\mathtt{V}(b') = b = b''$, $\mathtt{V}(e') = e = e''$ and $\mathcal{I}'' = \mathcal{I}$)
  $\Leftrightarrow$ for all interval bounds b' and e', $F_{[b', e']}$ is t-satisfiable.

T-satisfiability Preservation

Lemma 2. *For every rule, the father's bounded formulae, bounded state expressions and constraints are t-satisfiable if, and only if, the bounded formulae, bounded state expressions and constraints in one of the sons are t-satisfiable.*

Proof. see Appendix A.

Tableau Fundamental Propriety

Definition 5. *$\succ$ is a partial order relation defined on tree nodes by:*

- *$n' \succ n$ if n' is a descendant of n,*
- *$n' \succcurlyeq n$ if n' is a descendant of n or is the same node as n.*

Fact 4.21 *If a bounded state expression $x_{[b,e]}$ belongs to a node then any bounded state subexpression coming from $x_{[b,e]}$ is $[b,e]$-stamped.*

Looking at the rules applied to bounded state expressions, *i.e.* rules double negation, interval length S, null length, maximal length, and, interval union and interval disjunction, it is obvious.

Definition 6. *Let* $\mathtt{n}$ *be a node and b, b' interval bounds. b and b' are* connected in $\mathtt{n}$, *denoted by $b \bowtie b' \in \mathtt{n}$, if there exist $b_1, \ldots, b_m$ interval bounds such that the set of constraints $\{b \mathsf{R}_1 b_1, b_1 \mathsf{R}_2 b_2, \ldots, b_m \mathsf{R}_{m+1} b' : \mathsf{R}_i \in \{<, \leq, =\}\}$ belongs to* $\mathtt{n}$.

Fact 4.22 – *The binary relation $\bowtie$ is transitive.*
– *If $b \bowtie b'$ then $b \leq b'$.*

Lemma 3. *Let $F_{[b,e]}$ be a bounded formula or bounded state expression in a preconsistent node* $\mathtt{n}$. *In every preconsistent terminal node such that* $\mathtt{n}' \succcurlyeq \mathtt{n}$, *$b \bowtie e$.*

Proof. see Appendix A.

Lemma 4. *As far as rule* interval disjunction *is concerned, its application while not in the stated conditions is useless, i.e. it does not bring any new information.*

Proof. see Appendix A.

Interval Bound Constraint Resolution The interval bound constraints of a given terminal node $\mathtt{n}$ can be represented as a Simple Temporal Problem with inequations (STP$^{\neq}$ [Gerevini and Cristani 97]):

- the set of variables is the set of the interval bounds appearing in $\mathtt{n}$,
- each constraint is a pair of which the second element is the set of excluded points:
  - each $(e - b \neq k)$-like interval bound constraint is translated as the excluded point k,

  and of which the first element is a binary constraint composed of a single interval:
  the constraint between two variables b and e is the intersection of all the constraints involving b and e given that
  - each $(b \mathsf{R}\ e)$-like interval bound constraint is translated as the constraint
    - $(0, +\infty)$ if R is $<$,
    - $[0, +\infty)$ if R is $\leq$,

  * or $[0, 0]$ if R is $=$

  between b and e,
- and each $((e - b)\ \mathsf{S}\ k)$-like interval bound constraint is translated as the constraint
  * $(-\infty, k)$ if S is $<$,
  * $(-\infty, k]$ if S is $\leq$,
  * $[k, k]$ if S is $=$,
  * $[k, +\infty)$ if S is $\geq$,
  * or $(k, +\infty)$ if S is $>$

  between b and e.

Computing a STP$^{\neq}$ consistency is in $\mathcal{O}(n^3 + k)$ time and $\mathcal{O}(n^2 + k)$ space, where n is the number of variables and k is the number of inequations. A solution (if there is any) can be found in $\mathcal{O}(n^3 + k)$ time too [Gerevini and Cristani 97].

4.3 Model Associated with an Open Tableau

Definition 7. *Let* $\mathtt{n}$ *be a consistent terminal node of some open tableau then the model* $(\mathtt{V}, \mathcal{I}, [b, e])$ *associated with* $\mathtt{n}$ *is constructed in the following way :*

- *Let* $\mathtt{B} = \{b_1, \ldots, b_m\}$ *be the set of interval bounds occurring in the constraints of* $\mathtt{n}$.
  $\mathtt{V} : \mathtt{B} \to \mathbb{T}\mathrm{ime}$ *associates with every bound of* $\mathtt{B}$ *a* $\mathbb{T}\mathrm{ime}$ *value such that the order on* $\{\mathtt{V}(b_1), \ldots, \mathtt{V}(b_m)\}$ *complies with the bound constraints of* $\mathtt{n}$*, i.e. if* $b_i\ \mathsf{R}\ b_j \in \mathtt{n}$ *then* $\mathtt{V}(b_i)\ \mathsf{R}\ \mathtt{V}(b_j)$*, where* $\mathsf{R} \in \{<, \leq, =\}$ *and if* $(b_j - b_i)\ \mathsf{S}\ k \in \mathtt{n}$ *then* $(\mathtt{V}(b_j) - \mathtt{V}(b_i))\ \mathsf{S}\ k$*, where* $\mathsf{S} \in \{<, \leq, =, \neq, \geq, >\}$ *and* $k \in \mathbb{R}$.
  Let $b = min_{i \in \{1, \ldots, m\}}(\mathtt{V}(b_i))$ *and* $e = max_{i \in \{1, \ldots, m\}}(\mathtt{V}(b_i))$.
- *Let* $\{x_1, \ldots, x_p\}$ *be the set of state variables occurring in node* $\mathtt{n}$.
  $\forall j \in \{1, \ldots, p\}\ \mathcal{I}(x_j) \in \mathbb{T}\mathrm{ime} \to \{0, 1\}$ *such that*
  if ${x_j}_{[b', e']}$ *belongs to node* $\mathtt{n}$ *then* $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e')),\ \mathcal{I}(x_j)(t) = 1$
  and $\forall t \notin \bigcup\{[\mathtt{V}(b'), \mathtt{V}(e')) : {x_j}_{[b', e']} \in \mathtt{n}\},\ \mathcal{I}(x_j)(t) = 0$.

4.4 Soundness

Theorem 2 (Soundness). *If a tableau for a formula* F *is open then* F *is satisfiable.*

Proof. Let $\mathtt{n}$ be a consistent terminal node of an open tableau for F and let $(\mathtt{V}, \mathcal{I}, [b, e])$ be the model associated with $\mathtt{n}$.
First, let us prove that model $(\mathtt{V}, \mathcal{I}, [b, e])$ t-satisfies every bounded state expression and every interval bound constraint of node $\mathtt{n}$:

- if $b_i\ \mathsf{R}\ b_j \in \mathtt{n}$ with $\mathsf{R} \in \{<, \leq, =\}$ then, by definition 7, $\mathtt{V}(b_i) \in [b, e]$, $\mathtt{V}(b_j) \in [b, e]$ and $\mathtt{V}(b_i)\ \mathsf{R}\ \mathtt{V}(b_j)$.
- if $(b_j - b_i)\ \mathsf{S}\ k \in \mathtt{n}$ with $\mathsf{S} \in \{<, \leq, =, \neq, \geq, >\}$ and $k \in \mathbb{R}$ then, by definition 7, $\mathtt{V}(b_i) \in [b, e]$, $\mathtt{V}(b_j) \in [b, e]$ and $(\mathtt{V}(b_j) - \mathtt{V}(b_i))\ \mathsf{S}\ k$.

– if $x_{[b',e']} \in \mathtt{n}$ then, by lemma 3, as $\mathtt{n}$ is a consistent terminal node $b' \bowtie e'$ thus $b' \in \mathtt{B}$, $e' \in \mathtt{B}$ and $b' \leq e'$ (by fact 4.22) so, by definition 7, $\mathtt{V}(b') \in [b,e]$, $\mathtt{V}(e') \in [b,e]$ and $\mathtt{V}(b') \leq \mathtt{V}(e')$.
  Owing to the construction of $(\mathtt{V}, \mathcal{I}, [b,e])$, $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $\mathcal{I}(x)(t) = 1$ hence $(\mathtt{V}, \mathcal{I}, [b,e])$ t-satisfies $x_{[b',e']}$.
– if $\neg x_{[b',e']} \in \mathtt{n}$ then, by lemma 3, since $\mathtt{n}$ is a consistent terminal node $b' \bowtie e'$ thus $b' \in \mathtt{B}$, $e' \in \mathtt{B}$ and $b' \leq e'$ (by fact 4.22) so $\mathtt{V}(b') \in [b,e]$, $\mathtt{V}(e') \in [b,e]$ and $\mathtt{V}(b') \leq \mathtt{V}(e')$.
  Rule interval disjunction applied to all $x_{[b'',e'']} \in \mathtt{n}$
  $\Rightarrow$ for all $x_{[b'',e'']} \in \mathtt{n}$, $e'' \leq b'$ or $e' \leq b''$
  $\Rightarrow$ for all $x_{[b'',e'']} \in \mathtt{n}$, $b'' \in \mathtt{B}$, $e'' \in \mathtt{B}$ and ($\mathtt{V}(e'') \leq \mathtt{V}(b')$ or $\mathtt{V}(e') \leq \mathtt{V}(b'')$)
  $\Rightarrow$ for all $x_{[b'',e'']} \in \mathtt{n}$, $[\mathtt{V}(b'), \mathtt{V}(e')) \cap [\mathtt{V}(b''), \mathtt{V}(e'')) = \emptyset$
  $\Rightarrow [\mathtt{V}(b'), \mathtt{V}(e')) \cap \bigcup\{[\mathtt{V}(b''), \mathtt{V}(e'')) : x_{[b'',e'']} \in \mathtt{n}\} = \emptyset$
  $\Rightarrow \forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $t \notin \bigcup\{[\mathtt{V}(b''), \mathtt{V}(e'')) : x_{[b'',e'']} \in \mathtt{n}\}$.
  Moreover, by definition 7, $\forall t \notin \bigcup\{[\mathtt{V}(b''), \mathtt{V}(e''))$: $x_{[b'',e'']} \in \mathtt{n}\}$, $\mathcal{I}(x)(t) = 0$, so $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $\mathcal{I}(x)(t) = 0$, thus $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $\mathcal{I}(\neg x)(t) = 1$. Hence $(\mathtt{V}, \mathcal{I}, [b,e])$ t-satisfies $\neg x_{[b',e']}$.

Furthermore, by lemma 2, for every rule, the father's bounded formulae, bounded state expressions and interval bound constraints are t-satisfiable if the bounded formulae, bounded state expressions and interval bound constraints in one of the sons are t-satisfiable. Consequently $(\mathtt{V}, \mathcal{I}, [b,e])$ t-satisfies $F_{[\beta,\epsilon]}$.
Accordingly, by lemma 1, F is satisfiable.

4.5 Completeness

Theorem 3 (Completeness). *If F is a satisfiable formula then there exists an open tableau for F.*

Proof. Let $F_{[\beta,\epsilon]}$ be the root bounded formula. If F is satisfiable then $F_{[\beta,\epsilon]}$ is t-satisfiable (lemma 1). According to lemma 2, for every rule if the father's bounded formulae are t-satisfiable then the bounded formulae of one of the sons are t-satisfiable too. This entails the existence of an open tableau for F.

4.6 Decidability and Complexity

The ending of a tableau construction is provided by the proof of theorem 1. Consequently this Duration Calculus fragment is decidable.
As far as complexity is concerned, since the consistency problem of a $\mathrm{STP}^{\neq}$ (Simple Temporal Problem with inequations) is polynomial the overall complexity of the method will be the one associated with the tableau construction: our current work is to characterize this complexity.

5 Conclusion

In this note a deduction method for a fragment of the Duration Calculus was presented. In this fragment we can express quantitative constraints, *i.e.* constraints concerning the duration of actions, which is essential to reason in the

field of real-time systems.
Work in progress concerns on the one hand the study of the complexity of our method, on the other hand the implementation of this method. The results presented in this paper concerning the tableau construction characterize the theoretical feasibility of the decision procedure. In order to apply this proof procedure to realistic specifications, the introduction of strategies is necessary (see for example the section "Comparison" in [TABLEAUX'98,TABLEAUX'99]) ; this work is currently under way.

References

[Allen 83] Allen, J. F.: "Maintaining Knowledge about Temporal Intervals"; Communications of the ACM, 26, 11 (1983), 832–843.

[Gerevini and Cristani 97] Gerevini, A., Cristani, M.: "On Finding a Solution in Temporal Constraint Satisfaction Problem"; Proc. IJCAI'97, Morgan Kaufmann, 2 (1997), 1460–1465.

[Hansen and Zhou 92] Hansen, M. R., Zhou, C.: "Semantics and Completeness of Duration Calculus"; In W.-P. de Roever, J. W. de Bakker, C. Huizing, and G. Rozenberg, editors, *Real-Time: Theory in Practice*, REX Workshop, volume 600 of Lecture Notes in Computer Science, Springer-Verlag (1992), 209–225.

[Hansen and Zhou 97] Hansen, M. R., Zhou, C.: "Duration Calculus : Logical Foundations"; Formal Aspects of Computing, BCS, Springer, 9, 3 (1997), 283–330.

[Moszkowski 85] Moszkowski, B.: "A Temporal Logic for Multilevel Reasoning about Hardware"; Computer, IEEE Computer Society, 18, 2 (1985), 10–19.

[Ravn, Rischel and Hansen 93] Anders P. Ravn, A. P., Rischel, H. and Hansen, K. M.: "Specifying and Verifying Requirements of Real-Time Systems"; IEEE Transactions on Software Engineering, Special Issue on Software for Critical Systems, 19, 1 (1993), 41-55.

[Skakkebæk and Sestoft 94] Skakkebæk, J. U., Sestoft, P.: "Checking Validity of Duration Calculus Formulas"; Technical Report ID/DTH JUS 3/1, Department of Computer Science, Technical University of Denmark, 2800 Lyngby, Denmark (1994), also appeared as electronic version, available at ftp://ftp.it.dtu.dk/pub/ProCoS/Jens.U.Skakkebaek/IDDTH-JUS-3-1.ps.Z.

[TABLEAUX'98] de Swart, H., editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, TABLEAUX'98, volume 1397 of Lecture Notes in Artificial Intelligence, Springer-Verlag (1998).

[TABLEAUX'99] Murray, N.V., editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, TABLEAUX'99, volume 1617 of Lecture Notes in Artificial Intelligence, Springer-Verlag (1999).

[van Beek 90] van Beek, P.: "Exact and Approximate Reasoning about Qualitative Temporal Relations"; PhD thesis, University of Waterloo, Canada (1990).

[Zhou, Hoare and Ravn 91] Zhou, C., Hoare, C.A.R., Ravn, A. P.: "A calculus of durations"; information processing letters, North–Holland, 40, 5 (1991), 269–276.

[Zhou, Hansen and Sestoft 93] Zhou, C., Hansen, M. R., Sestoft, P.: "Decidability and Undecidability Results for Duration Calculus"; In P. Enjalbert, A. Finkel, and K. W. Wagner, editors, STACS 93, 10th Annual Symposium on Theoretical Aspects of Computer Science, volume 665 of Lecture Notes in Computer Science, Springer-Verlag (1993), 58–68.

A Proofs of lemmas

Lemma 2 (page 11). *For every rule, the father's bounded formulae, bounded state expressions and constraints are t-satisfiable if, and only if, the bounded formulae, bounded state expressions and constraints in one of the sons are t-satisfiable.*

Proof. First, let us prove that for all stop rules, the father's bounded formulae are not t-satisfiable.

- stop_1:
  Let $M = (\mathtt{V}, \mathcal{I}, [b, e])$ be a model of the father's bounded formulae:
  M t-satisfies $\perp_{[b',e']}$
  $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b, e]$ and $(\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(e')]) \models \perp \Rightarrow$ impossible.
- stop_2:
  Let $M = (\mathtt{V}, \mathcal{I}, [b, e])$ be a model of the father's bounded formulae:
  x is a state expression.
  M t-satisfies $x_{[b',e']}$ and M t-satisfies $\neg x_{[b',e']} \Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b, e]$ and $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $\mathcal{I}(x)(t) = 1$ and $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $\mathcal{I}(\neg x)(t) = 1$
  $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b, e]$ and $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $\mathcal{I}(x)(t) = 1$ and $\mathcal{I}(\neg x)(t) = 1$
  $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b, e]$ and $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $\mathcal{I}(x)(t) = 1$ and $\mathcal{I}(x)(t) = 0$
  $\Rightarrow$ impossible.
- stop_3:
  Let $M = (\mathtt{V}, \mathcal{I}, [b, e])$ be a model of the father's bounded formulae then $\mathtt{V}(b') \in [b, e]$, $\mathtt{V}(e') \in [b, e]$, $\mathtt{V}(b') < \mathtt{V}(e')$ and $\mathtt{V}(b') = \mathtt{V}(e')$ which is impossible.

Let us prove now that for every rule (but stop_1, stop_2 and stop_3) a model M t-satisfies the father's bounded formulae, bounded state expressions and constraints if, and only if, M t-satisfies the bounded formulae, bounded state expressions and interval bound constraints in one of the child nodes.

- $\top$:
  M t-satisfies $\top_{[b',e']}$
  $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(e')]) \models \top$
  $\Leftrightarrow \mathtt{V}(b') \in [b,e]$, $\mathtt{V}(e') \in [b,e]$ and $\mathtt{V}(b') \leq \mathtt{V}(e')$
  $\Leftrightarrow M$ t-satisfies $b' \leq e'$.
- double negation:
  M t-satisfies $(\neg\neg\sigma)_{[b',e']}$
  $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e'))$, $\mathcal{I}(\neg\neg\sigma)(t) = 1$
  $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e'))$, $\mathcal{I}(\neg\sigma)(t) = 1 - \mathcal{I}(\neg\neg\sigma)(t) = 0$
  $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e'))$, $\mathcal{I}(\sigma)(t) = 1-(1-\mathcal{I}(\neg\neg\sigma)(t)) = 1$
  $\Leftrightarrow M$ t-satisfies $\sigma_{[b',e']}$
- negation and:
  M t-satisfies $\neg(\sigma_1 \wedge \sigma_2)_{[b',e']}$
  $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e'))$, $\mathcal{I}(\neg(\sigma_1 \wedge \sigma_2))(t) = 1$
  $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e'))$, $\mathcal{I}(\sigma_1 \wedge \sigma_2)(t) = 1 - \mathcal{I}(\neg(\sigma_1 \wedge \sigma_2))(t) = 0$
  $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e'))$, $\mathcal{I}(\sigma_1)(t) \times \mathcal{I}(\sigma_2)(t) = 0$
  $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $(\forall t \in [\mathtt{V}(b'),\mathtt{V}(e')), \mathcal{I}(\sigma_1)(t) = 0$ or $\mathcal{I}(\sigma_2))(t) = 0)$
  $\Leftrightarrow ([\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e')), \mathcal{I}(\sigma_1)(t) = 0)$ or $([\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e'))$, $\mathcal{I}(\sigma_2))(t) = 0)$
  $\Leftrightarrow ([\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e')), \mathcal{I}(\neg\sigma_1)(t) = 1)$ or $([\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e')), \mathcal{I}(\neg\sigma_2))(t) = 1)$
  $\Leftrightarrow M$ t-satisfies $\neg{\sigma_1}_{[b',e']}$ or M t-satisfies $\neg{\sigma_2}_{[b',e']}$
- interval length S:
  M t-satisfies $(\ell \ \mathsf{S}\ k)_{[b',e']}$ where $\mathsf{S} \in \{<,\leq,=,\neq,\geq,>\}$
  $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(e')]) \models \ell \ \mathsf{S}\ k$
  $\Leftrightarrow \mathtt{V}(b') \in [b,e]$, $\mathtt{V}(e') \in [b,e]$, $\mathtt{V}(b') \leq \mathtt{V}(e')$ and $(\mathtt{V}(e') - \mathtt{V}(b'))\ \mathsf{S}\ k$
  $\Leftrightarrow M$ t-satisfies $b' \leq e'$ and M t-satisfies $(e' - b')\ \mathsf{S}\ k$.
- null length:
  - M t-satisfies $(\int \sigma = 0)_{[b',e']}$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(e')]) \models \int \sigma = 0$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\mathcal{I}[\![\int \sigma]\!]([\mathtt{V}(b'),\mathtt{V}(e')]) = 0$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\int_{\mathtt{V}(b')}^{\mathtt{V}(e')} \mathcal{I}[\![\sigma]\!](t)dt = 0$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e'))$, $\mathcal{I}[\![\sigma]\!](t) = 0$ except for a finite number of points
    $\Rightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and by note 1 $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e'))$, $\mathcal{I}[\![\sigma]\!](t) = 0$
    $\Leftrightarrow \mathtt{V}(b') \in [\mathtt{V}(b'),\mathtt{V}(e')]$, $\mathtt{V}(e') \in [\mathtt{V}(b'),\mathtt{V}(e')]$, $\mathtt{V}(b') \leq \mathtt{V}(e')$, $[\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e'))$, $\mathcal{I}[\![\neg\sigma]\!](t) = 1$
    $\Leftrightarrow M$ t-satisfies $\neg\sigma_{[b',e']}$ and M t-satisfies $b' \leq e'$
  - M t-satisfies $\neg\sigma_{[b',e']}$ and $b' \leq e'$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$, $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e'))$, $\mathcal{I}[\![\neg\sigma]\!](t) = 1$, $\mathtt{V}(b') \in [\mathtt{V}(b'),\mathtt{V}(e')]$, $\mathtt{V}(e') \in [\mathtt{V}(b'),\mathtt{V}(e')]$ and $\mathtt{V}(b') \leq \mathtt{V}(e')$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e'))$, $\mathcal{I}[\![\sigma]\!](t) = 0$
    $\Rightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\int_{\mathtt{V}(b')}^{\mathtt{V}(e')} \mathcal{I}[\![\sigma]\!] = 0$

$\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\mathcal{I}[\![\int \sigma]\!]([\mathtt{V}(b'), \mathtt{V}(e')]) = 0$
$\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(e')]) \models \int \sigma = 0$
$\Leftrightarrow M$ t-satisfies $(\int \sigma = 0)_{[b',e']}$

– maximal length:
  - M t-satisfies $(\int \sigma = \ell)_{[b',e']}$
    $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(e')]) \models \int \sigma = \ell$
    $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\mathcal{I}[\![\int \sigma]\!]([b,e]) = \mathcal{I}[\![\ell]\!]([b,e])$
    $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\int_{\mathtt{V}(b')}^{\mathtt{V}(e')} \mathcal{I}[\![\sigma]\!](t)dt = \int_{\mathtt{V}(b')}^{\mathtt{V}(e')} 1dt$
    $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\int_{\mathtt{V}(b')}^{\mathtt{V}(e')} \mathcal{I}[\![\sigma]\!](t)dt = \mathtt{V}(e') - \mathtt{V}(b')$
    $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $\mathcal{I}[\![\sigma]\!](t) = 1$ except for a finite number of points
    $\Rightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and by note 1 $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $\mathcal{I}[\![\sigma]\!](t) = 1$
    $\Leftrightarrow \mathtt{V}(b') \in [\mathtt{V}(b'), \mathtt{V}(e')]$, $\mathtt{V}(e') \in [\mathtt{V}(b'), \mathtt{V}(e')]$, $\mathtt{V}(b') \leq \mathtt{V}(e')$, $[\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $\mathcal{I}[\![\sigma]\!](t) = 1$
    $\Leftrightarrow M$ t-satisfies $\sigma_{[b',e']}$ and M t-satisfies $b' \leq e'$
  - M t-satisfies $\sigma_{[b',e']}$ and $b' \leq e'$
    $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$, $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $\mathcal{I}[\![\sigma]\!](t) = 1$, $\mathtt{V}(b') \in [\mathtt{V}(b'), \mathtt{V}(e')]$, $\mathtt{V}(e') \in [\mathtt{V}(b'), \mathtt{V}(e')]$ and $\mathtt{V}(b') \leq \mathtt{V}(e')$
    $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $\mathcal{I}[\![\sigma]\!](t) = 1$
    $\Rightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\int_{\mathtt{V}(b')}^{\mathtt{V}(e')} \mathcal{I}[\![\sigma]\!](t)dt = \int_{\mathtt{V}(b')}^{\mathtt{V}(e')} 1dt$
    $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\int_{\mathtt{V}(b')}^{\mathtt{V}(e')} \mathcal{I}[\![\sigma]\!](t)dt = \int_{\mathtt{V}(b')}^{\mathtt{V}(e')} \mathcal{I}[\![1]\!](t)dt$
    $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\mathcal{I}[\![\int \sigma]\!]([\mathtt{V}(b'), \mathtt{V}(e')]) = \mathcal{I}[\![\int 1]\!]([\mathtt{V}(b'), \mathtt{V}(e')])$
    $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\mathcal{I}[\![\int \sigma]\!]([\mathtt{V}(b'), \mathtt{V}(e')]) = \mathcal{I}[\![\ell]\!]([\mathtt{V}(b'), \mathtt{V}(e')])$
    $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(e')]) \models \int \sigma = \ell$
    $\Leftrightarrow M$ t-satisfies $(\int \sigma = \ell)_{[b',e']}$

– or:
  x and y are formulae.
  M t-satisfies $(x \vee y)_{[b',e']}$
  $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(e')]) \models (x \vee y)$
  $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $((\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(e')]) \models x$ or $(\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(e')]) \models y)$
  $\Leftrightarrow ([\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(e')]) \models x)$ or $([\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(e')]) \models y)$
  $\Leftrightarrow M$ t-satisfies $x_{[b',e']}$ or M t-satisfies $y_{[b',e']}$

– and:
  if x and y are state expressions then
  M t-satisfies $(x \wedge y)_{[b',e']}$
  $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e')), \mathcal{I}(x \wedge y)(t) = 1$
  $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e'))$, $(\mathcal{I}(x)(t) = 1$ and $\mathcal{I}(y)(t) = 1)$
  $\Leftrightarrow ([\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e')), \mathcal{I}(x)(t) = 1)$ and $([\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'), \mathtt{V}(e')), \mathcal{I}(y)(t) = 1)$
  $\Leftrightarrow M$ t-satisfies $x_{[b',e']}$ and M t-satisfies $y_{[b',e']}$
  if x and y are formulae then
  M t-satisfies $(x \wedge y)_{[b',e']}$
  $\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(e')]) \models (x \wedge y)$

$\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$, $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(e')]) \models x$ and $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(e')]) \models y$
$\Leftrightarrow ([\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(e')]) \models x)$ and $([\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(e')]) \models y)$
$\Leftrightarrow M$ t-satisfies $x_{[b',e']}$ and M t-satisfies $y_{[b',e']}$

– subinterval sequence:
  - M t-satisfies $(x;y)_{[b',e']}$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(e')]) \models (x;y)$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and there exists $b_2 \in [\mathtt{V}(b'),\mathtt{V}(e')]$ such that $(\mathcal{I},[\mathtt{V}(b'),b_2]) \models x$ and $(\mathcal{I},[b_2,\mathtt{V}(e')]) \models y$
    $\Rightarrow$ let b_1 be a new interval bound such that $\mathtt{V}(b_1) = b_2$, $[\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$, $\mathtt{V}(b_1) \in [\mathtt{V}(b'),\mathtt{V}(e')]$, $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(b_1)]) \models x$ and $(\mathcal{I},[\mathtt{V}(b_1),\mathtt{V}(e')]) \models y$
    $\Leftrightarrow ([\mathtt{V}(b'),\mathtt{V}(b_1)] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(b_1)]) \models x)$ and $([\mathtt{V}(b_1),\mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b_1),\mathtt{V}(e')]) \models y)$
    $\Leftrightarrow M$ t-satisfies $x_{[b',b_1]}$ and M t-satisfies $y_{[b_1,e']}$
  - M t-satisfies $x_{[b',b_1]}$ and M t-satisfies $y_{[b_1,e']}$
    $\Leftrightarrow ([\mathtt{V}(b'),\mathtt{V}(b_1)] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(b_1)]) \models x)$ and $([\mathtt{V}(b_1),\mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b_1),\mathtt{V}(e')]) \models y)$
    $\Rightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and there exists $b_2 \in [\mathtt{V}(b'),\mathtt{V}(e')]$ such that $(\mathcal{I},[\mathtt{V}(b'),b_2]) \models x$ and $(\mathcal{I},[b_2,\mathtt{V}(e')]) \models y$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(e')]) \models (x;y)$
    $\Leftrightarrow M$ t-satisfies $(x;y)_{[b',e']}$

– interval union:
  if x is a state expression then
  - M t-satisfies $x_{[b',b_1]}$ and M t-satisfies $x_{[b_1,e']}$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(b_1)] \subseteq [b,e]$, $\forall t \in [\mathtt{V}(b'),\mathtt{V}(b_1)), \mathcal{I}(x)(t) = 1$, $[\mathtt{V}(b_1),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b_1),\mathtt{V}(e')), \mathcal{I}(x)(t) = 1$
    $\Rightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e')), \mathcal{I}(x)(t) = 1$
    $\Leftrightarrow M$ t-satisfies $x_{[b',e']}$
  - M t-satisfies $x_{[b',e']}$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b'),\mathtt{V}(e')), \mathcal{I}(x)(t) = 1$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall b_2 \in [\mathtt{V}(b'),\mathtt{V}(e')]$, $\forall t \in [\mathtt{V}(b'),b_2)$, $\mathcal{I}(x)(t) = 1$ and $\forall t \in [b_2,\mathtt{V}(e')), \mathcal{I}(x)(t) = 1$ (the temporal domain is dense)
    $\Rightarrow$ let b_1 be an interval bound such that $\mathtt{V}(b_1) \in [\mathtt{V}(b'),\mathtt{V}(e')]$, $[\mathtt{V}(b'),\mathtt{V}(b_1)] \subseteq [b,e]$, $\forall t \in [\mathtt{V}(b'),\mathtt{V}(b_1)), \mathcal{I}(x)(t) = 1$, $[\mathtt{V}(b_1),\mathtt{V}(e')] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b_1),\mathtt{V}(e')), \mathcal{I}(x)(t) = 1$
    $\Leftrightarrow M$ t-satisfies $x_{[b',b_1]}$ and M t-satisfies $x_{[b_1,e']}$

  if x is a formula then
  - M t-satisfies $x_{[b',b_1]}$ and M t-satisfies $x_{[b_1,e']}$
    $\Leftrightarrow [\mathtt{V}(b_1),\mathtt{V}(e')] \subseteq [b,e]$, $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(b_1)]) \models x$, $[\mathtt{V}(b_1),\mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b_1),\mathtt{V}(e')]) \models x$
    $\Rightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(e')]) \models x;x$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(e')]) \models x$
    $\Leftrightarrow M$ t-satisfies $x_{[b',e']}$
  - M t-satisfies $x_{[b',e']}$
    $\Leftrightarrow [\mathtt{V}(b'),\mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I},[\mathtt{V}(b'),\mathtt{V}(e')]) \models x$

$\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(e')]) \models x; x$ (the temporal domain is dense)
$\Leftrightarrow [\mathtt{V}(b'), \mathtt{V}(e')] \subseteq [b,e]$ et $\forall b_2 \in [\mathtt{V}(b'), \mathtt{V}(e')]$, $(\mathcal{I}, [\mathtt{V}(b'), b_2]) \models x$ and $(\mathcal{I}, [b_2, \mathtt{V}(e')]) \models x$
$\Rightarrow$ let b_1 be an interval bound such that $\mathtt{V}(b_1) \in [\mathtt{V}(b'), \mathtt{V}(e')]$, $[\mathtt{V}(b_1), \mathtt{V}(e')] \subseteq [b,e]$, $(\mathcal{I}, [\mathtt{V}(b'), \mathtt{V}(b_1)]) \models x$, $[\mathtt{V}(b_1), \mathtt{V}(e')] \subseteq [b,e]$ and $(\mathcal{I}, [\mathtt{V}(b_1), \mathtt{V}(e')]) \models x$
$\Leftrightarrow M$ t-satisfies $x_{[b',b_1]}$ and M t-satisfies $x_{[b_1,e']}$

– interval disjunction:
Additional hypothesis: $e_1 \leq b_2$ does not belong to the father and neither does $e_2 \leq b_1$.
M t-satisfies $x_{[b_1,e_1]}$ and M t-satisfies $\neg x_{[b_2,e_2]}$
$\Leftrightarrow [\mathtt{V}(b_1), \mathtt{V}(e_1)] \subseteq [b,e]$, $\forall t \in [\mathtt{V}(b_1), \mathtt{V}(e_1)), \mathcal{I}(x)(t) = 1$, $[\mathtt{V}(b_2), \mathtt{V}(e_2)] \subseteq [b,e]$ and $\forall t \in [\mathtt{V}(b_2), \mathtt{V}(e_2)], \mathcal{I}(\neg x)(t) = 1$
$\Leftrightarrow [\mathtt{V}(b_1), \mathtt{V}(e_1)] \subseteq [b,e]$, $[\mathtt{V}(b_2), \mathtt{V}(e_2)] \subseteq [b,e]$, $\forall t \in [\mathtt{V}(b_1), \mathtt{V}(e_1))$, $\mathcal{I}(x)(t) = 1$ and $\forall t \in [\mathtt{V}(b_2), \mathtt{V}(e_2)]$, $\mathcal{I}(x)(t) = 0$
$\Leftrightarrow$ ($[\mathtt{V}(b_1), \mathtt{V}(e_1)] \subseteq [b,e]$, $\forall t \in [\mathtt{V}(b_1), \mathtt{V}(e_1))$, $\mathcal{I}(x)(t) = 1$ and $[\mathtt{V}(b_2), \mathtt{V}(e_2)] \subseteq [b,e]$, $\forall t \in [\mathtt{V}(b_2), \mathtt{V}(e_2)]$, $\mathcal{I}(x)(t) = 0$ and $\mathtt{V}(e_1) \in [b,e]$, $\mathtt{V}(b_2) \in [b,e]$, $\mathtt{V}(e_1) \leq \mathtt{V}(b_2)$) or ($[\mathtt{V}(b_1), \mathtt{V}(e_1)] \subseteq [b,e]$, $\forall t \in [\mathtt{V}(b_1), \mathtt{V}(e_1)]$, $\mathcal{I}(x)(t) = 1$ and $[\mathtt{V}(b_2), \mathtt{V}(e_2)] \subseteq [b,e]$, $\forall t \in [\mathtt{V}(b_2), \mathtt{V}(e_2)]$, $\mathcal{I}(x)(t) = 0$ and $\mathtt{V}(e_1) \in [b,e]$, $\mathtt{V}(b_2) \in [b,e]$, $\mathtt{V}(e_2) \leq \mathtt{V}(b_1)$)
$\Leftrightarrow$ (M t-satisfies $x_{[b_1,e_1]}$ and M t-satisfies $\neg x_{[b_2,e_2]}$ and M t-satisfies $e_1 \leq b_2$) or (M t-satisfies $x_{[b_1,e_1]}$ and M t-satisfies $\neg x_{[b_2,e_2]}$ and M t-satisfies $e_2 \leq b_1$)

Lemma 3 (page 12). *Let $F_{[b,e]}$ be a bounded formula or bounded state expression in a preconsistent node* $\mathtt{n}$. *In every preconsistent terminal node such that* $\mathtt{n}' \succcurlyeq \mathtt{n}$, $b \bowtie e$.

Proof. – If F is a formula, we prove the lemma by structural induction on F:

- if $F = \bot$ then applying rule stop_1 generates stop in the only son, the requirements of the lemma are not met: it does not apply to $\bot$.
- base case: $F = \top$, $\ell \mathsf{S} k$, $\int \sigma = 0$ or $\int \sigma = \ell$, where $\mathsf{S} \in \{<, \leq, =, \neq, \geq, >\}$, k is a real number and σ is a state expression.
  - $*$ if $F = \top$ then applying rule $\top$ generates constraint $b \leq e$ in the only son of $\mathtt{n}$. By fact 4.11, this constraint propagates to the preconsistent terminal descendants of $\mathtt{n}$, if $\mathtt{n}$ is not terminal itself. Thus the lemma is true for $\top$.
  - $*$ if $F = (\ell \mathsf{S} k)$, where $\mathsf{S} \in \{<, \leq, =, \neq, \geq, >\}$ and $k \in \mathbb{R}$, then applying rule interval length generates constraints $b \leq e$ and $(e - b) \mathsf{S} k$ in the only son of $\mathtt{n}$. By fact 4.11, these constraints propagate to the preconsistent terminal descendants of $\mathtt{n}$, if $\mathtt{n}$ is not terminal itself. Thus the lemma is true for $\ell \mathsf{S} k$, where $\mathsf{S} \in \{<, \leq, =, \neq, \geq, >\}$ and k is a real number.
  - $*$ if $F = (\int \sigma = 0)$, where σ is a state expression, then applying rule null length generates the constraint $b \leq e$ in the only son of $\mathtt{n}$. By fact 4.11,

that constraint propagates to the preconsistent terminal descendants of $\mathtt{n}$, if $\mathtt{n}$ is not terminal itself. Thus the lemma is true for $\int \sigma = 0$, where σ is a state expression.

* if $F = (\int \sigma = \ell)$, where σ is a state expression, then applying rule maximal length generates the constraint $b \leq e$ in the only son of $\mathtt{n}$. By fact 4.11, that constraint propagates to the preconsistent terminal descendants of $\mathtt{n}$, if $\mathtt{n}$ is not terminal itself. Thus the lemma is true for $\int \sigma = \ell$, where σ is a state expression.

- induction hypothesis: the lemma is true for all subformulae of F; let us prove that it is true then for F if $F = F_1 \vee F_2$, $F_1 \wedge F_2$ ou $F_1 ; F_2$.
  * if $F = F_1 \vee F_2$ then applying rule or generates respectively ${F_1}_{[b,e]}$ and ${F_2}_{[b,e]}$ in the two sons of $\mathtt{n}$,$\mathtt{n}_1'$ and $\mathtt{n}_2'$. By induction hypothesis the lemma is true for ${F_1}_{[b,e]}$ and ${F_2}_{[b,e]}$ hence in every preconsistent terminal node $\mathtt{n}_1''$ such that $\mathtt{n}_1'' \succ \mathtt{n}_1{}'$ – so $\mathtt{n}_1'' \succ \mathtt{n}$ –, $b \bowtie e$ and in every preconsistent terminal node $\mathtt{n}_2''$ such that $\mathtt{n}_2'' \succ \mathtt{n}_2'$ – so $\mathtt{n}_2'' \succ \mathtt{n}$ –, $b \bowtie e$. Thus the lemma is true for $F_1 \vee F_2$.
  * if $F = F_1 \wedge F_2$ then applying rule and generates bounded formulae ${F_1}_{[b,e]}$ and ${F_2}_{[b,e]}$ in the only son of $\mathtt{n}$, $\mathtt{n}'$. By induction hypothesis the lemma is true for ${F_1}_{[b,e]}$ and ${F_2}_{[b,e]}$ hence in every preconsistent terminal node $\mathtt{n}''$ such that $\mathtt{n}'' \succ \mathtt{n}'$ – so $\mathtt{n}'' \succ \mathtt{n}$ –, $b \bowtie e$. Thus the lemma is true for $F_1 \wedge F_2$.
  * if $F = F_1 ; F_2$ then applying rule subinterval sequence generates bounded formulae ${F_1}_{[b,b']}$ and ${F_2}_{[b',e]}$, where b' is a new interval bound, in the only son of $\mathtt{n}$, $\mathtt{n}'$. By induction hypothesis the lemma is true for ${F_1}_{[b,b']}$ and ${F_2}_{[b',e]}$ hence in every preconsistent terminal node $\mathtt{n}''$ such that $\mathtt{n}'' \succ \mathtt{n}'$ – so $\mathtt{n}'' \succ \mathtt{n}$ –, $b \bowtie b'$ and $b' \bowtie e$. By fact 4.22, $b \bowtie e \in \mathtt{n}''$. Thus the lemma is true for $F_1 ; F_2$.

Therefore the lemma is true for every formula.

– Suppose now that F is a state expression.
The root contains a formula F' so any state expression belonging to a descendant comes from a subformula of F', more precisely from a $(\int \sigma = k)$-like subformula of F' because rules null length and maximal length are the only ones to generate a state expression from a formula. Given that from a bounded formula $(\int \sigma = 0)_{[b',e']}$ rule null length generates a bounded state expression $\neg\sigma_{[b',e']}$ and from a bounded formula $(\int \sigma = \ell)_{[b',e']}$ rule maximal length generates a bounded state expression $\sigma_{[b',e']}$ and given fact 4.21, if state expression $F_{[b,e]}$ comes from $(\int \sigma = 0)_{[b',e']}$ or $(\int \sigma = \ell)_{[b',e']}$ then actually b is b' and e is e'. Moreover applying rules null length or maximal length to $(\int \sigma = k)_{[b',e']}$ generates the constraint $b' \leq e'$ and by fact 4.11, this constraint propagates to n and to its preconsistent terminal descendants.
Thus the lemma is true for every state expression.

Lemma 4 (page 12). *As far as rule interval disjunction is concerned, its application while not in the stated conditions is useless, i.e. it does not bring any new information.*

Proof. Suppose that a terminal node n contains $x_{[b_1,e_1]}$ and $\neg x_{[b_2,e_2]}$ but that the application requirement of rule interval disjunction is not satisfied.

- case $e_1 \leq b_2$ belongs to n and $e_2 \leq b_1$ does not: the application of rule interval disjunction carries all the bounded formulae, bounded state expressions and constraints of n, among them $e_1 \leq b_2$, to both sons of n and generates constraints $e_1 \leq b_2$ in node n_1 and $e_2 \leq b_1$ in node n_2, n_1 and n_2 being the sons of n.
  On the one hand n_1 does not contain any new information, on the other hand all the bounded formulae, bounded state expressions and constraints of n_1 belong to n_2 consequently any model of n_2 is a model of n_1 and if n_1 is not satisfiable then so is n_2. As we are interested in the open branches of the tableau, n_2 is useless. Since n_1 is identical to n and n_2 is "less interesting" than n_1, the application of rule interval disjunction in that case is useless.
- case $e_2 \leq b_1$ belongs to n and $e_1 \leq b_2$ does not: similar to previous case.
- case $e_1 \leq b_2$ and $e_2 \leq b_1$ belong to n: the application of rule interval disjunction carries all the bounded formulae, bounded state expressions and constraints of n, among them $e_1 \leq b_2$ and $e_2 \leq b_1$, to both sons of n and generates constraints $e_1 \leq b_2$ or $e_2 \leq b_1$ in the sons of n. These ones being already present in both sons of n, the application of rule interval disjunction in that case does not bring any new information.