Monodic temporal logic with quantified propositional variables

This item was submitted to Loughborough University's Institutional Repository by the/an author.


Additional Information:

- This article was published in the Journal of Logic and Computation [© The Author. Published by Oxford University Press] and the definitive version is available at: http://dx.doi.org/10.1093/logcom/exr004

Metadata Record: https://dspace.lboro.ac.uk/2134/14855

Version: Accepted for publication

Publisher: © The Author. Published by Oxford University Press

Please cite the published version.
This item was submitted to Loughborough’s Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
Monodic Temporal Logic with Quantified Propositional Variables

Walter Hussak*

February 2011

Abstract

We extend the monodic fragment of first-order linear temporal logic to include right-linear grammar operators and quantification of propositional variables. Unlike propositional temporal logic, the use of grammar operators in first-order temporal logic is not equivalent to general propositional quantification, as the latter admit satisfiable formulae without countable models. We consider the decision problem for fragments where propositional quantification occurs outside of quantification of individual variables and temporal (grammar) operators. We show that if externally quantified propositions inside temporal operators occur within positive occurrences of universal quantifiers for individual variables then validity for all propositional prefix classes is recursively enumerable, and decidable in the two-variable case. Without this condition we show that, even with very severe restrictions on the first-order part of the logic, no non-trivial prefix class is recursively enumerable.

Keywords: Monodic Temporal Logic, Quasimodels, Decidability, Recursive Enumerability.

1 Introduction

Propositional linear temporal logic can be made more expressive by introducing propositional quantification [18], [17]. The resulting logic QPTL is decidable, but with non-elementary complexity. Other ways of achieving the expressive power of QPTL include the use of right-linear grammar operators as in ETL [21], the linear μ-calculus [2], and the use of various kinds of automata as temporal connectives [20], [15]. These alternative methods have often been preferred to propositional quantification because of elementary decision problems, and have given rise to extensions in a wide range of logics from Extended CTL* to Automata Logic [5]. Recently, there has been an increase in interest in

*Department of Computer Science, Loughborough University, Loughborough, LE11 3TU, UK. W.Hussak@lboro.ac.uk
QPTL with complete proof systems being given in [14] and [9]. Propositional
quantification has also been studied in branching temporal logic [8].

In this paper we extend both the use of grammar operators and propositional
quantification to first-order linear temporal logic and establish basic results on
decidability/recursive enumerability. For the latter to hold, the underlying first-
order temporal logic being extended has to be decidable/recursively enumerable,
and so we work with the monodic fragment [12]. We show that unsatisfiability
is recursively enumerable for the monodic fragment where propositional quan-
tification occurs outside of quantification of individual variables and grammar
operators, inside of which externally quantified propositions are within positive
occurrences of existential quantifiers for individual variables in the overall for-
mula. Satisfiability is decidable for the case where the number of individual
variables is restricted to two.

This paper is structured as follows. In section 2, we define the monodic
fragment of first-order temporal logic and monodic fragments with grammar
operators and propositional quantification. The structures used for proving
decidability of monodic temporal logics are ‘quasimodels’ [12]. In section 3, we
give an expanded form of quasimodel corresponding to a collection of models
of a monodic formula differing on the assignments of propositions. The main
decidability/recursive enumerability results are given in section 4. We make
some concluding remarks in section 5.

2 Languages and models

The alphabet of the temporal language consists of lists of propositional sym-
bols $P_0, P_1, \ldots$ some of which are designated as quantifiable, predicate symbols
$p_0, p_1, \ldots$ (but not equality), individual variables $x_0, x_1, \ldots$ individual constants
c_0, c_1, \ldots, the booleans $\neg, \land, \top, \bot$, the quantifiers $\exists x$ and $\exists P$ for individual and
propositional variables respectively, and temporal operators $G^i(\psi_1, \ldots, \psi_m)$ cor-
responding to right-linear grammars $G^i$. Each $G^i$ with non-terminals $V_0, \ldots, V_j$
and terminals $v_1, \ldots, v_m$, generates words by means of a finite set of produc-
tions $V_h \rightarrow v_1 V_h', V_h' \rightarrow v_2$ and $G^W(\psi_1, \psi_2)$ by the pro-
ductions $V_0 \rightarrow v_1 V_0, V_0 \rightarrow v_2$. Formulae $\psi$ are defined by:

$$
\psi \ ::= \ P \mid p(t_1, \ldots, t_n) \mid \neg \psi_1 \mid \psi_1 \land \psi_2 \mid \top \mid \bot \mid \exists x \psi \mid \exists P \psi \mid G^u(\psi_1, \ldots, \psi_m)
$$

where $P_i$ is quantifiable, $p$ is an n-ary predicate symbol and any of the $t_i$’s in
$p(t_1, \ldots, t_n)$ is a constant or individual variable. We use the standard abbrevi-
ations for $\lor$, $\Rightarrow$ and $\forall$, and

$$
\bigcirc \psi = G^O(\top, \psi), \psi_1 U \psi_2 = G^W(\psi_1 \land \neg \psi_2, \neg \psi_1 \land \neg \psi_2), \Diamond \psi = T U \psi, \square \psi = \neg \Diamond \neg \psi
$$

and refer, loosely, to ‘predicates’ $p(x_1, \ldots, x_n)$. A formula denoted by, for ex-
ample, $\psi(x_1, \ldots, x_n)$, indicates that its set of free variables is a subset of the
set \{x_1, \ldots, x_n\}. The formula which results from substituting all occurrences of \(x\) by \(y\) in \(\psi\), is denoted \(\psi[x/y]\). We write \(\psi_1 \leq \psi_2\) to denote that \(\psi_1\) is a subformula of \(\psi_2\). The symbols \(\bot\) and \(\top\) are also used loosely to denote truth values. The scopes of occurrences of \(\exists x\) and \(G^i\) in \(\exists x\psi\) and \(G^i(\psi_1, \ldots, \psi_m)\) are \(\psi\) and \((\psi_1, \ldots, \psi_m)\) respectively.

A structure (or model) \(I\) over domain \(D\) assigns values of appropriate type to propositional symbols, predicate symbols, and individual constants over some domain \(D\)

\[
I = \langle D, P^0_1, \ldots, P^0_n, \ldots, c^0_1, \ldots \rangle
\]

If \(u\) assigns values to some subset of symbols, then the structure \(I \upharpoonright u\) denotes the structure which assigns values for these symbols according to \(u\) and other symbols according to \(I\). A temporal structure (or model) \(\mathcal{M}\) is a sequence of structures \((I(n) : n \in \mathbb{N})\) over the same domain \(D\) such that, for every individual constant \(c\) and \(n_1, n_2 \in \mathbb{N}, c^{I(n_1)} = c^{I(n_2)}\). The truth relation \((\mathcal{M}, n) \models^a \psi\), where \(a\) is an assignment of values to individual variables, is defined inductively on the construction of \(\psi\) as follows:

- \((\mathcal{M}, n) \models^a P\) iff \(P^{I(n)} = \top\)
- \((\mathcal{M}, n) \models^a P(t_1, \ldots, t_n)\) iff \(P^{I(n)}(a_1, \ldots, a_n) = \top\), where \(a_i = a(t_i)\) if \(t_i\) is a variable, and \(a_i = t_i^{I(n)}\) if \(t_i\) is a constant
- \((\mathcal{M}, n) \models^a \neg \psi_1\) iff \((\mathcal{M}, n) \not\models^a \psi_1\) does not hold
- \((\mathcal{M}, n) \models^a \psi_1 \land \psi_2\) iff \((\mathcal{M}, n) \models^a \psi_1\) and \((\mathcal{M}, n) \models^a \psi_2\)
- \((\mathcal{M}, n) \models^a \exists x \psi\) iff \((\mathcal{M}, n) \models^b \psi\) for some assignment \(b\) that may differ from \(a\) only on \(x\)
- \((\mathcal{M}, n) \models^a \exists P \psi\) iff for all \(n' \in \mathbb{N}\) there is an assignment \(u_{n'}\) of a truth value to \(P\) such that, if \(\mathcal{M}' = (I(n') \upharpoonright u_{n'} : n' \in \mathbb{N})\), we have \((\mathcal{M}', n) \models^a \psi\)
- \((\mathcal{M}, n) \models^a G^i(\psi_1, \ldots, \psi_m)\) iff there is a finite or infinite word \(w_n, w_{n+1}, \ldots\) generated by \(G^i\) such that, for all \(n' \geq n\), if \(w_{n'}\) is defined and \(w_{n'} = v_j\), then \((\mathcal{M}, n') \models^a \psi_j\)

A temporal structure \(\mathcal{M}\) satisfies a formula \(\phi\) with no free individual variables iff \((\mathcal{M}, 0) \models^a \phi\) for some assignment \(a\), and \(\phi\) is valid iff all \(\mathcal{M}\) satisfy \(\phi\). Satisfiability/validity for a subset \(L\) of the temporal language is decidable/recursively enumerable iff the set of satisfiable/valid formulae in \(L\) is recursive/recursively enumerable.

A formula \(\phi\) is monodic if any subformula of \(\phi\) of the form \(G^i(\psi_1, \ldots, \psi_m)\) has at most one free individual variable. The language EMTL is the set of monodic formulae with no propositional quantification and the language QEMTL is the set of closed (for the individual variables) monodic formulae where propositional quantification occurs only outside of quantification of individual variables and grammar operators. By first-order logic we mean the set of formulae without either grammar operators or propositional quantification. We denote by
ψ the set of matrices in (1) such that, whenever curs free in every trix
They also have a set of the domain of a first-order model
that may contain quantifiers for individual variables, these are only prefix classes for
formulae (1) called EMTL
ples of P
involving a
(2) says that the quantifier that binds the free variable in any
temporal model (ψ) from each realizable state candidate. This set of runs
have, at each moment in time n, a 'state candidate' which is a collection of possible sets,
indexed by elements t in some finite set T, of the formulae ψ in subϕ
such that ψ is true at moment in time n, for some element a (substituting for x)
of the domain of a first-order model J(n) that 'realizes' the state candidate.
They also have a set R of 'runs', which are sequences comprising a choice of t
from each realizable state candidate. This set of runs R forms the domain of a
temporal model (I(n) : n ∈ N) for ϕ by defining, for each n ∈ N, a surjection

Every formula in QEMTL can be put into a prenex normal form

Q1P1 . . . QkPk, ϕ(P1, . . . ,Pk)

where each Qi is either ∃ or ∀, each Pi a propositional variable, and ϕ has no propositional quantifiers. Here, Q1P1 . . . QkPk is the prefix and ϕ the mar-
trix. A variable x is said to be live in a subformula ψ of ϕ if there is some G′(ψ1, . . . ,ψm) ≤ ψ in which some Pj(1 ≤ j ≤ k) occurs, and such that x oc-
curs free in every ψ′ with G′(ψ1, . . . ,ψm) ≤ ψ′ ≤ ψ. The language EMTL3pos
is the set of matrices in (1) such that, whenever ∃xψ ≤ φ and x is live in ψ, then

|{ψ′ : ∃xψ ≤ ¬ψ′ ≤ ϕ}| is even

(2)

Then, QEMTL3pos is the set of formulae (1) such that ϕ ∈ EMTL3pos. Note that
(2) says that the quantifier that binds the free variable in any G′(ψ1, . . . ,ψm)
involving a Pj(1 ≤ j ≤ k), is always a positive occurrence of ∃ in ϕ. Examples of EMTL3pos
formulae will be seen in section 4. We define subclasses of
formulae (1) called prefix classes by specifying allowable prefixes. As ϕ in (1)
may contain quantifiers for individual variables, these are only prefix classes for
propositional quantification. We denote by Π the set \{P1, . . . ,Pk\}, by surϕ
the set of propositions, constants, predicates and surrogates of subformulae of
ϕ, and by surϕ the set of propositions Pι ∈ Π and surrogates of subformulae of
ϕ containing at least one Pi ∈ Π. A structure J assigning values of appropri-
type to symbols in surϕ surϕ ϕ may be expanded to a structure J′ that
assigns values to symbols in surϕ ϕ. In that case J′ is called a Π-expansion of

J.

3 Quasimodels

The decidability of monodic fragments of first-order temporal logic is usually
established using 'quasimodels' [12]. Quasimodels for a formula ϕ have, at each
moment in time n, a 'state candidate' which is a collection of possible sets,
indexed by elements t in some finite set T, of the formulae ψ in subϕ
such that ψ is true at moment in time n, for some element a (substituting for x)
of the domain of a first-order model J(n) that 'realizes' the state candidate.
They also have a set R of 'runs', which are sequences comprising a choice of t
from each realizable state candidate. This set of runs R forms the domain of a
temporal model (I(n) : n ∈ N) for ϕ by defining, for each n ∈ N, a surjection
\(\Delta_n\) from \(R\) to the domain \(D\) of \(J(n)\) so that a run \(r\) is interpreted in \(I(n)\) in the same way as its image under \(\Delta_n\) in \(D\) is interpreted in \(J(n)\). The effect of \(\Delta_n\) is to ‘duplicate’ the points of the \(J(n)\) so that different duplicates can produce different required behaviours at different moments in time, and hence give rise to a temporal model for \(\phi\). A test for satisfiability of \(\phi\) is constructed by encoding the existence of a quasimodel into one of the (decidable) second-order theories of natural numbers with successor functions and monadic predicates \(S1S\) or \(S2S\). The exact nature of quasimodels depends on the linear [12] or branching [13], [3] temporal logic of interest. Our aim in this section is to determine when an EMTL\textsuperscript{pos} formula \(\phi\) has a model for all (temporal) assignments \(v\) in some set of assignments \(T\) where the assignments to \(P_i \in \Pi\) vary. This is achieved by extending the basic quasimodel approach.

We define quasimodels for which realizable state candidates have an extra index \(u\) in some other finite set \(U\), to cater for different subsets of \(sub_x\phi\) being true at the same moment in time \(n\) and element of the domain, but for different assignments over time for the propositions \(P_i \in \Pi\). This leads to a set of first-order models at \(n\), \((J_u(n) : u \in U)\), for state candidates. These new realizable state candidates are defined below by Definitions 3.8 and 3.9 and quasimodels and the surjections \(\Delta_n\) in some other finite set \(U\) so that a run \(r\) is interpreted in \(I(n)\) (for state candidates). These new realizable state candidates are defined below by Definitions 3.8 and 3.9 and quasimodels and the surjections \(\Delta_n\) respectively.

The main result, linking the existence of our (new) quasimodels and models for \(\phi\), for all assignments \(v \in T\), is Theorem 3.13. It relies on an induction on subformulae \(\psi\) of \(\phi\), proving that \(\neg \exists x \psi\) is true in \(I_{u_{\text{run}}}(n)\), for an assignment of free individual variables in \(\psi\) to runs \(r \in R\), iff \(\psi\) is true in the temporal model at \(n\) for the same assignment of free individual variables and for assignments to \(P_i \in \Pi\) according to \(\psi\). A difficulty arises in overcoming occurrences of \(\forall x\), i.e. occurrences of \(\neg \exists x\), in the induction, as runs \(r \in R\) that are in \(R_u\), behave well in \(I_{u_{\text{run}}}(n)\), but runs in another \(R_{u'} (u' \neq u)\), which are in the domain of the model and have to be taken into account by the universal quantifier, may not behave so well in \(I_{u'_{\text{run}}}(n)\). Fortunately, as \(\phi\) belongs to EMTL\textsuperscript{pos}, there is no \(P_i \in \Pi\) in the scope of any \(\forall x\), and so interpretations for those scopes will be the same for \(v\) as for any other \(u'\). In the induction, the live individual variables in \(\psi\) (that are in the scope of an existential quantifier in \(\phi\) that contains \(P_i \in \Pi\)), are assigned (by ‘realizing’ assignments) to runs \(r \in R_u\) that satisfy existentially quantified subformulae, as required, in \(J_{u_{\text{run}}}(n)\) and therefore \(I_{u_{\text{run}}}(n)\). Free individual variables bound by other quantifiers can have any assignments as they behave identically in all \(v \in T\). Realizing assignments are given in Definition 3.5.

The other parts of this section that we have not mentioned are Lemma 3.11, which concerns countable models, and the two following results from classical
first-order logic that we use.

**Lemma 3.1** Let \( J = \langle D, P^J_0, \ldots, p^J_l, \ldots, c^J_0, \ldots \rangle \) be a structure of first-order logic. Suppose that
\[
\Delta : D' \rightarrow D
\]
is a surjection and that the structure \( I = \langle D', P^I_0, \ldots, p^I_l, \ldots, c^I_0, \ldots \rangle \) is such that, for all \( l \geq 0 \), and for all assignments \( a \) of individual variables to elements in \( D' \), if the assignment \( a^J \) is defined by \( a^J(x) = \Delta(a(x)) \), we have that \( P^I_l = P^J_l \), \( I \models^a p_l(x_1, \ldots, x_n) \) iff \( J \models^{a^J} p_l(x_1, \ldots, x_n) \), and \( \Delta(c^I_l) = c^J_l \). Then, as equality is absent, for any formula \( \psi \) and assignment \( a \),
\[
I \models^a \psi \text{ iff } J \models^{a^J} \psi
\]
We will say that \( \Delta \) induces the first-order structure \( I \) over \( D' \) from \( J \).

**Lemma 3.2** Satisfiability for the class of first-order logic formulae not necessarily in prenex normal form, which use at most two individual variables and any number of constants, is decidable (see [4]).

For the remainder of this paper, we shall fix \( \phi \) to be an \( \text{EMTL}_3^\text{pos} \) formula which is the matrix of a formula as in (1) with no free individual variables.

**Definition 3.3 (state candidate)** Let \( C \) be the set of individual constants in \( \phi \), \( U \) be a set of cardinality \( 2^{2|\text{sub}_x \phi|} \cdot 2^{|\text{sub}_x \phi||C|} \) and \( T \) a set of cardinality \( 2^{|\text{sub}_x \phi||T|} + |C| \). Suppose that \( T \) and \( C \) are functions with signatures:
\[
T : T \times U \rightarrow \mathcal{P}(\text{sub}_x \phi), \quad C : C \times U \rightarrow \mathcal{P}(\text{sub}_x \phi)
\]
where \( \mathcal{P}(\text{sub}_x \phi) \) is the powerset of \( \text{sub}_x \phi \). Then, the pair \( \langle T, C \rangle \) is called a state candidate for \( \phi \). As \( T, C, U \) and \( \text{sub}_x \phi \) are finite, there is an effective bound on the number of possible state candidates for \( \phi \). The state candidates of interest are the ‘realizable’ state candidates.

**Definition 3.4 (realizable state candidate)** For a structure \( J \) and an element \( a \) of its domain \( D \), the type \( tp_J(a) \) is defined to be:
\[
\text{tp}_J(a) = \{ \psi \in \text{sub}_x \phi : J \models^{\psi}(a) \}
\]
Let \( J \) be a structure assigning values to all symbols in \( \text{surr} \phi - \text{surr} p \phi \) over a domain \( D \) and let \( \mathcal{J} = \{ J_u : u \in U \} \) be a set of \( \Pi \)-expansions of \( J \). Then, \( \mathcal{J} \) realizes the state candidate \( \langle T, C \rangle \) with base \( J \) iff:
(i) For all \( t \in T \), there is \( a \in D \) such that, for all \( u \in U \), \( T(t, u) = \text{tp}_{J_u}(a) \).
(ii) For all \( a \in D \), there is \( t \in T \) such that, for all \( u \in U \), \( \text{tp}_{J_u}(a) = T(t, u) \).
(iii) For all \( c \in C \), if \( c^l = a \) where \( a \in D \), then, for all \( u \in U \), \( \text{tp}_{J_u}(a) = C(c, u) \).
A state candidate is realizable if it is realized by some such \( J \). Intuitively, the \( u \in U \) give different structures to match different assignments to \( P_i \in \Pi \) over time. Lemma 3.6 below provides a test for realizability. In the proof of the main result, Theorem 3.13, we shall make use of the following assignment for live variables.

**Definition 3.5 (realizing assignment)** Suppose that \( J \) realizes \( \langle T, C \rangle \) as in Definition 3.4. Let \( \exists x \psi \) be a subformula of \( \phi \) such that \( x \) is live in \( \psi \), \( a \) be an assignment to individual variables, and \( u \in U \) be such that

\[
J_u \models a \exists x \psi
\]

Then, clearly, there exists an assignment \( r^a_u, u, \psi \), called a realizing assignment, such that

\[
r^a_u, u, \psi (y) = a(y) \quad (y \neq x)
\]

and

\[
J_u \models r^a_u, u, \psi
\]

(Strictly speaking, we should write \( r^a_u, u, \psi, J \) as the realizing assignment depends on \( J \), but \( J \) will be clear from the context.)

**Lemma 3.6** A state candidate \( \langle T, C \rangle \) for \( \phi \) is realizable iff the conjunction \( \theta \) of the following three first-order logic formulae \( \theta_1, \theta_2, \theta_3 \) is satisfiable:

\[
\theta_1 = \bigwedge_{t \in T} \exists x \bigwedge_{u \in U} ( \bigwedge_{\psi(x) \in T(t,u)} \overline{\psi}(x)[\ldots] \land \bigwedge_{\psi(x) \notin T(t,u)} \neg \overline{\psi}(x)[\ldots] )
\]

\[
\theta_2 = \forall x \bigvee_{t \in T} \bigwedge_{u \in U} ( \bigwedge_{\psi(x) \in T(t,u)} \overline{\psi}(x)[\ldots] \land \bigwedge_{\psi(x) \notin T(t,u)} \neg \overline{\psi}(x)[\ldots] )
\]

\[
\theta_3 = \bigwedge_{c \in C} \bigwedge_{u \in U} ( \bigwedge_{\psi(c) \in C(c,u)} \overline{\psi}(c)[\ldots] \land \bigwedge_{\psi(c) \notin C(c,u)} \neg \overline{\psi}(c)[\ldots] )
\]

where \( [\ldots] \) substitutes occurrences of \( P_i \in \Pi \) by propositions \( P_{i,u} \) and predicates \( p_\psi \in \text{surr}_P \phi \) by predicates \( p_{\psi,u} \) of the same arity. Furthermore, if \( \phi \) has at most two individual variables (possibly reused with different quantifiers) then realizability is decidable.

**Proof** For the decidable case, if \( \phi \) has at most two individual variables, the conjunction \( \theta_1 \land \theta_2 \land \theta_3 \) can be written using no more than two individual variables. Decidability follows from Lemma 3.2. \( \blacksquare \)

**Definition 3.7 (state function)** A state function \( f \) associates with each \( n \in \mathbb{N} \) a realizable state candidate \( f(n) = \langle T_n, C_n \rangle \).

**Definition 3.8** An individual assignment \( v^{P_i} \) for \( P_i \in \Pi \) \((1 \leq i \leq k)\), is a subset of \( \mathbb{N} \). For each \( n \in \mathbb{N} \), we write \( v^{P_i}(n) = \top \) iff \( n \in v^{P_i} \). A set of assignments for \( \Pi \) over time \( \mathcal{T} \) is a set of functions \( v \in \mathcal{Y} \) such that

\[
v : \mathbb{N} \to \mathcal{P}(\Pi)
\]
giving, for each \( P_i \in \Pi \), the individual assignment \( v^{P_i} \in \mathcal{P}(N) \) defined by 
\[ n \in v^{P_i} \iff P_i \in v(n) \]
For each \( n \in N \), we also use the notation \( v_n \) in the manner:
\[ v_n(P_i) = \top \iff P_i \in v(n) \quad (P_i \in \Pi) \]
For the remainder of this section, we fix some non-empty \( \Upsilon \).

**Definition 3.9 (runs)** A family of runs for \((\phi, \Upsilon)\), with respect to the state function \( f \), consists of pairwise disjoint sets \( R_v(v \in \Upsilon) \) and a map
\[ R : R \times N \rightarrow T \times U \]
where \( R = \bigcup_{v \in \Upsilon} R_v \), with function \( R^T : R \times N \rightarrow T \) giving the \( t \) elements of \( R \), and, for all \( v \in \Upsilon \) and \( n \in N \), an \( u_{vn} \in U \), called the \( u \) value corresponding to \( R_v \) at \( n \), satisfying
\[ R(r, n) = (R^T(r, n), u_{vn}) \quad (r \in R_v, n \in N) \quad (5) \]
such that the following properties hold:

(i) For all \( v \in \Upsilon \), \( n \in N \) and \( t \in T \), there exists \( r \in R_v \) such that
\[ R^T(r, n) = t \]

(ii) For all \( c \in C \) and \( v \in \Upsilon \), there is a \( r_c \in R_v \) such that, for all \( n \in N \),
\[ T_v(R(r, n)) = C_v(c, u_{vn}) \]

(iii) For all \( r \in R \), \( n \in N \) and \( G'(\psi_1, \ldots, \psi_m) \in \text{sub}_x \phi \), \( G'(\psi_1, \ldots, \psi_m) \in T_v(R(r, n)) \) iff there is a word \( w_n w_{n+1} \ldots \) generated by the corresponding grammar \( G' \), such that, for \( n' \geq n \) and \( 1 \leq j \leq m \), if \( w_{n'} \) is defined and \( w_{n'} = v_j \) then \( \psi_j \in T_v(R(r, n')) \).

(iv) For all \( n \in N \), \( v \in \Upsilon \) and \( r \in R_v \), \( \Pi \cap T_v(R(r, n)) = v(n) \).
As with runs, quasimodels are defined with respect to \( \Upsilon \).

**Definition 3.10 (quasimodel)** A quasimodel for \((\phi, \Upsilon)\) is a pair \( \langle f, R \rangle \) where:

(i) \( f \) is a state function

(ii) \( R \) is a family of runs for \((\phi, \Upsilon)\) with respect to \( f \) as in Definition 3.9, such that \( \phi \in T_0(R(r, 0)) \) for all \( r \in R_v \) and \( v \in \Upsilon \).

The following is clear from Definition 3.10.

**Lemma 3.11** If there exists a quasimodel for \((\phi, \Upsilon)\) and \( \Upsilon \) is countable, then there is a quasimodel \( \langle f, R \rangle \) where \( R \) is a function \( R : R \times N \rightarrow T \) for some countable set \( R \).
Models are constructed from quasimodels in Theorem 3.13 below by taking the runs in $R$ as the domain, and inducing structures $I(n)$ from the first-order model $J(n)$ of the realizable state candidate at $n$, by means of a surjection $\Delta_n$ from $R$ to the domain $D$ of the $J$'s. Indeed, surjections $\Delta_{vn}$ from $R_v$ to $D$ are defined and, to ensure that the new domain has at least as many elements as $D$, the larger set $R \times D$ rather than $R$ is used for the domain of the model.

**Lemma 3.12** Let $\langle f, R \rangle$ be a quasimodel for $\langle \phi, \Upsilon \rangle$. Suppose that, for each $n \in \mathbb{N}$, $f(n) = \langle I_n, C_n \rangle$ is realized by $J(n) = \{ J_u(n) : u \in U \}$, with base $J(n)$, (common to all $n$) domain $D$ and types $tp_{J_u(n)}(a)$. Then, for each $v \in \Upsilon$ and $n \in \mathbb{N}$, there exists a surjection

$$\Delta_{vn} : R_v \times D \to D$$

such that, for all $r, r_c \in R_v$ and $a \in D$, if $u_{vn}$ is the $u$ value corresponding to $R_v$ at $n$ as in (5) of Definition 3.9,

(i) $T_n(R(r, n), u_{vn}) = tp_{J_{u_{vn}}(n)}(\Delta_{vn}(r, a))$

(ii) $\Delta_{vn}(r_c, c^{J(n)}) = c^{J(n)}$

**Proof** Fix $v, n$. By (5) and Definition 3.9(i),

$$\{ T_n(R(r, n), u_{vn}) : r \in R_v \} = \{ T_n(t, u_{vn}) : t \in T \}$$

and, by Definition 3.4(i) and (ii),

$$\{ T_n(t, u_{vn}) : t \in T \} = \{ tp_{J_{u_{vn}}(n)}(a) : a \in D \}$$

From (6) and (7), it follows that there exists a surjection $\Delta_{vn} : R_v \times D \to D$ such that, for all $r \in R_v$ and $a \in D$,

$$T_n(R(r, n), u_{vn}) = tp_{J_{u_{vn}}(n)}(\Delta_{vn}(r, a))$$

and

$$\Delta_{vn}(r, a) = a \text{ if } T_n(R(r, n), u_{vn}) = tp_{J_{u_{vn}}(n)}(a)$$

For (ii), if $c \in C$, by (5), Definition 3.9(ii) and Definition 3.4(iii),

$$T_n(R(r_c, n), u_{vn}) = T_n(R(r_c, n)) = C_n(c, u_{vn}) = tp_{J_{u_{vn}}(n)}(c^{J(n)})$$

By (8) and (9), $\Delta_{vn}(r_c, c^{J(n)}) = c^{J(n)}$.

The correspondence between models that satisfy $\phi$ for all assignments in $\Upsilon$ and quasimodels for $\langle \phi, \Upsilon \rangle$ is given in the following theorem.

**Theorem 3.13** Let $\phi \in EMTL^{\exists \text{pos}}$. Then:

(i) There is a temporal structure $\mathfrak{M} = \langle I(n) : n \in \mathbb{N} \rangle$ such that the temporal structure $\mathfrak{M}_v = \langle I(n) \upharpoonright v_n : n \in \mathbb{N} \rangle$ satisfies $\phi$ for all $v \in \Upsilon$, iff there exists a quasimodel for $\langle \phi, \Upsilon \rangle$. 

9
(ii) If $\Upsilon$ is countable and $\mathfrak{M}$ exists as in (i), then $\mathfrak{M}$ can be chosen to have a countable domain.

Proof. The proof given here is a sketch with some simplifications; full details are in Appendix 1. Let $\mathfrak{M}$ and $\mathfrak{M}_v$ be temporal structures as in (i) over domain $D$. For all $n \in \mathbb{N}$ and $v \in \Upsilon$, let $J_v(n)$ be the expansion of $I(n) \upharpoonright v$ to surrogates $p_\psi \in \text{surr}\phi$ given by:

$$p_\psi^{J_v(n)}(a) = \top \iff (\mathfrak{M}_v, n) \models \psi(a) \quad (a \in D) \quad (10)$$

so that, for all $n \in \mathbb{N}$ and $\psi \subseteq \phi$, and all assignments $a$ into $D$,

$$J_v(n) \models^a \psi \iff (\mathfrak{M}_v, n) \models^a \psi \quad (11)$$

For a quasimodel, we need a finite family of structures $\{J_u : u \in U\}$. We obtain this by finding, for each $n \in \mathbb{N}$, $|U|$ `representatives' $\{J_{v_1}^n, \ldots, J_{v_{|U|}}^n\}$, where $v_1^n, \ldots, v_{|U|}^n \in \Upsilon$, of the structures $\{J_v(n) : v \in \Upsilon\}$, and $|T|$ representatives $\{a_1^n, \ldots, a_{|T|}^n\}$ of the elements of $D$. The representatives are chosen to be such that, for each $v \in \Upsilon$, there exists $u_{vn} \in \{v_1^n, \ldots, v_{|U|}^n\}$ such that:

$$\{\{\psi \in \text{sub}_x \phi : J_v(n) \models \psi(a)\} : a \in D\} =$$

$$\{\{\psi \in \text{sub}_x \phi : J_{u_{vn}}(n) \models \psi(a)\} : a \in D\} \quad (12)$$

and, for all $c \in C$,

$$\{\psi \in \text{sub}_x \phi : J_v(n) \models \psi(c^{J(n)})\} = \{\psi \in \text{sub}_x \phi : J_{u_{vn}}(n) \models \psi(c^{J(n)})\}$$

and such that, for each $a \in D$, there exists $a^n_c \in \{a_1^n, \ldots, a_{|T|}^n\} \supseteq \{c^{J(n)} : c \in C\}$ such that, for all $u_{vn} \in \{v_1^n, \ldots, v_{|U|}^n\}$,

$$\{\psi \in \text{sub}_x \phi : J_{u_{vn}}(n) \models \psi(a)\} = \{\psi \in \text{sub}_x \phi : J_{u_{vn}}(n) \models \psi(a^n_c)\} \quad (13)$$

We then let $\text{elem}T^n : \{a_1^n, \ldots, a_{|T|}^n\} \to T$ and $\text{elem}U^n : \{v_1^n, \ldots, v_{|U|}^n\} \to U$ be bijections and define a realizable state candidate $\langle T_u, C_u \rangle$ by:

$$T_u(\text{elem}T^n(a_c^n), \text{elem}U^n(v_{j^n})) = \{\psi \in \text{sub}_x \phi : J_{v_{j^n}}(n) \models \psi(a_c^n)\}$$

$$C_u(c, \text{elem}U^n(v_{j^n})) = \{\psi \in \text{sub}_x \phi : J_{v_{j^n}}(n) \models \psi(c^{J(n)})\}$$

$(1 \leq i \leq |T|, 1 \leq j \leq |U|, c \in C)$. A family of runs $R = \bigcup_{v \in T} R_v$, where $R_v = \{r_{a,u} : a \in D\}$, is defined by:

$$R(r_{a,u}, n) = (\text{elem}T^n(a_c^n), \text{elem}U^n(u_{vn}))$$

where, from (12) and (13), $a_c^n$ is chosen to be such that $T_u(\{\psi \in \text{sub}_x \phi : J_{u_{vn}}(n) \models \psi(a_c^n)\}) = \{\psi \in \text{sub}_x \phi : J_v(n) \models \psi(a)\} \quad (14)$
Given (14), it is a routine though lengthy task to show that \( \langle f, R \rangle \), where \( f \) associates with each \( n \in \mathbb{N} \) the realizable state candidate \( \langle \mathcal{T}_n, \mathcal{C}_n \rangle \), is a quasimodel for \( \langle \phi, \Upsilon \rangle \). The full proof is given in Appendix 1.

Conversely, suppose that \( \langle f, R \rangle \) is a quasimodel for \( \langle \phi, \Upsilon \rangle \) where \( f(n) \) is the realizable state candidate \( \langle \mathcal{T}_n, \mathcal{C}_n \rangle \). There are only finitely many possibilities for the \( \langle \mathcal{T}_n, \mathcal{C}_n \rangle \) and each distinct \( \langle \mathcal{T}_n, \mathcal{C}_n \rangle \) can be realized over a countable domain by classical model theory. It follows, by Lemma 3.1, that there is a single countable domain \( D \) such that, for each \( n \in \mathbb{N} \), \( \langle \mathcal{T}_n, \mathcal{C}_n \rangle \) is realized by a set of structures over \( D \), \( \mathcal{J}(n) = \{ J_n(n) : u \in U \} \), with base \( J(n) \) and types \( tp_{J_n(n)}(a) (a \in D, u \in U) \). As equality is absent it also follows by Lemma 3.1 that each \( J(n) \) can be chosen to be such that, if \( c, d \in C \) are distinct, then \( c^{J(n)} \neq d^{J(n)} \). Then, we can arrange for the \( J(n) \)'s to be such that for all \( c \in C, n, n' \in \mathbb{N} \)

\[
c^{J(n)} = c^{J(n')} \tag{15}
\]

As the \( R_v (v \in \Upsilon) \) that make up \( R \) are pairwise disjoint, we define, consistently, surjections \( \Delta_n : R \times D \rightarrow D (n \in \mathbb{N}) \) such that:

\[
\Delta_n(r, a) = \Delta_{\nu n}(r, a) \quad (r \in R_v, a \in D), \quad \Delta_n(r, c^{J(n)}) = c^{J(n)} \quad (r_c \in R_v, c \in C)
\]

where the \( \Delta_{\nu n} \) satisfy all the conditions of Lemma 3.12. By Lemmas 3.1 and 3.12(ii), for each \( n \in \mathbb{N} \), \( \Delta_n \) induces a set of structures \( \mathcal{I}(n) = \{ I_n(n) : u \in U \} \) with base \( I(n) \) over domain \( D' = R \times D \) from \( \mathcal{J}(n) \) such that, for \( n, n' \in \mathbb{N} \), \( c^{J(n)} = (r_c, c^{J(n)}) = (r_c, c^{J(n')}) \) (by (15)) = \( c^{J(n')} \), where \( r_c \in R_v \) for some (any) choice of \( v \in \Upsilon \). Then, \( \mathcal{I}(n) \) realizes \( \langle \mathcal{T}_n, \mathcal{C}_n \rangle \). Clearly, \( \mathcal{M} = \langle I(n) : n \in \mathbb{N} \rangle \) and therefore, for \( v \in \Upsilon \), \( \mathcal{M}_v = \langle I(n) \upharpoonright \nu_n : n \in \mathbb{N} \rangle \) are temporal structures. We need to show that \( \mathcal{M}_v \) satisfies \( \phi \), i.e. for all \( v \in \Upsilon \),

\[
(\mathcal{M}_v, 0) \models \phi \tag{16}
\]

This is proved by induction on subformulæ \( \psi \) of \( \phi \) by showing that, for all \( n \in \mathbb{N} \) and \( v \in \Upsilon \), if \( u_{\nu n} \) is the \( u \) value corresponding to \( R_v \) at \( n \) as in (5) and Lemma 3.12, and \( a \) is chosen to be an assignment in \( D' \) to individual variables of the form

\[
a(x) = (r, a) \quad \text{where } r \in R_v \text{ if } x \text{ is live in } \psi, \tag{17}
\]

then

\[
I_{u_{\nu n}}(n) \models^a \zeta \psi \bar{\psi} \quad \text{implies} \quad (\mathcal{M}_v, n) \models^a \zeta \psi \tag{18}
\]

where \( \zeta \psi \) is either \( \neg \psi \) or \( \psi \) according to whether the number of enclosing \( \neg \)'s is odd or even, i.e.

\[
\zeta = \begin{cases} 
\text{empty string}, & \text{if } |\{ \psi' : \psi \leq \psi' \leq \neg \psi' \leq \phi \}| \text{ is even} \\
\neg, & \text{otherwise}
\end{cases}
\]

From (18), if \( \psi = \phi \), \( \psi \) is closed and \( \zeta \psi \) is the empty string and so, for any assignment \( a \), we have that

\[
I_{u_{\nu n}}(n) \models^a \bar{\phi} \quad \text{implies} \quad (\mathcal{M}_v, n) \models^a \phi \tag{19}
\]
Then (16) follows from (19) as, by Definition 3.10(ii) and Lemma 3.12(i), for any \( a \in D, r \in R_\nu \) and \( \nu \in \Upsilon, \)

\[ \phi \in T_\sigma(R(r, 0)) = tp_{J_\nu\gamma_0}(0)(\Delta_{\nu\theta}(r, a)) \]

and so, by the definition of \( tp \) (Definition 3.4), as \( \phi \) has no free variables, 

\[ J_\nu\gamma_0(0) \models \phi \]

and, as \( I_\nu\gamma_0(0) \) is induced by \( J_\nu\gamma_0(0) \), \( I_\nu\gamma_0(0) \models \phi \) by (19). The main part of the induction reassigns a live variable \( x \) according to the realizing assignment \( a, u_{\nu, a, \phi} \) (Definition 3.5) to prove the \( \exists x \psi \) case. The details are in the Appendix 1.

To prove (ii) we note that, if \( \Upsilon \) is countable then, by Lemma 3.11, the quasimodel \( \langle f, R \rangle \) in (i) can be chosen so that \( R : R \times N \to T \) has countable \( R \). Then, the temporal structure \( (I(n) : n \in N) \) constructed above has domain \( D' = R \times D \) which is countable as \( R \) and \( D \) are countable.

\[ \Box \]

4 Decidability and recursive enumerability

The main result in Section 3, Theorem 3.13, gives an association between models for (propositionally unquantified) \( EMTL^{\exists \text{pos}} \) formulae \( \phi \), for all assignments \( \nu \) to \( \Pi \in \Pi \) in a set of assignments \( \Upsilon \), and quasimodels for \( \langle \phi, \Upsilon \rangle \). By Skolemization (Lemma 4.4 below), a \( QEMTL^{\exists \text{pos}} \) formula \( Q_1P_1 \ldots Q_kP_k \phi \) has a model iff \( \phi \) has a model for all assignments \( \nu \) in a set of assignments \( \Upsilon_\sigma \) generated by a Skolem function for the prefix \( Q_1P_1 \ldots Q_kP_k \). In this section, we encode the existence of quasimodels for \( \langle \phi, \Upsilon \rangle \) into S1S, by means of which a test in S1S for satisfiability of \( QEMTL^{\exists \text{pos}} \) can be constructed (Theorem 4.5). The main decidability/recursive enumerability result follows in Theorem 4.6. Expressiveness of \( QEMTL^{\exists \text{pos}} \) is considered in Theorem 4.7, and Theorem 4.8 shows that the decidability/recursive enumerability result in Theorem 4.6 cannot be improved significantly.

We arrive at the encoding in stages. Let the ‘predicate’

\[ \xi_\phi(f, \nu, <<u_n>>) \]

be parameterized by functions \( f : N \to \Sigma \), where \( \Sigma \) is the set of all realizable state candidates for \( \phi \) (with respect to \( \Pi = \{P_1, \ldots, P_k\} \)), assignments \( \nu \) to \( P_i \in \Pi \) over time, and sequences \( <<u_n : n \in N>> \) (abbreviated \( <<u_n>> \)) of elements of \( U \). For the function \( f \), let \( f(m)^L \) and \( f(m)^C \) denote the \( L \) and \( C \) part of the state candidate \( f(m) \) respectively \( (m \in N) \). Let \( \xi_\phi(f, \nu, <<u_n>>) \) assert that, for all \( n \in N, t \in T \) and \( c \in C \), there exist functions

\[ \rho_{n,t} : N \to T \quad \text{and} \quad \rho^c : N \to T \]

such that (compare with corresponding conditions of Definition 3.9(i), (ii), (iii) and (iv), and Definition 3.10(ii)):

\[ (\rho_{3,9(i)}) \quad \rho_{n,t}(n) = t, \]

\[ (\rho_{3,9(ii)}) \quad \text{the sequence} \langle f(m)^L(\rho_{n,t}(m), u_m) : m \in N \rangle \text{of subsets of} \quad \text{sub}_x \phi \]

\[ \Box \]
respects grammars,
\((p_{3.9(iii)})\) the sequence \(<f(m)\overline{\lambda}(\rho_{n,t}(m), u_m) : m \in \mathbb{N}>\) of subsets of \(sub_{x}\phi\)
agrees with \(v\),
\((p_{3.10(iii)})\) \(\rho_{n,t}(0)\) contains \(\phi\) at 0, \(i.e.\) \(f(0)\overline{\lambda}(\rho_{n,t}(0), u_0)\) contains \(\phi\),
\((p_{3.9(ii)})\) conditions \(p_{3.9(iii)}\) and \(p_{3.10(iii)}\) are satisfied when \(\rho_{n,t}\)
is replaced by \(\rho^c\) and, for all \(m \in \mathbb{N}\),
\[f(m)\overline{\lambda}(\rho^c(m), u_m) = f(m)\overline{\lambda}(c, u_m)\]

**Lemma 4.1** A function \(f : \mathbb{N} \rightarrow \Sigma\) is the state function of a quasimodel \(⟨f, \mathcal{R}⟩\)
for \((\phi, \Upsilon)\) iff, for all \(v \in \Upsilon\), there exists \(<u_n>\) such that \(\rho_{\phi}(f, v, <u_n>)\) is true.

**Proof** The proof is given in Appendix 2.

We encode \(\rho_{\phi}(f, v, <u_n>)\) into SIS as follows. A function \(f : \mathbb{N} \rightarrow \Sigma\) is given
by unary predicates \(P_s(z)\) \((s \in \Sigma)\), where exactly one of the \(P_s(z)\)'s is true at
each \(z \in \mathbb{N}\), \(i.e.\)
\[f^{\text{SIS}} = \forall z \left( \bigvee_{s \in \Sigma} P_s(z) \land \bigwedge_{s, s' \in \Sigma, s \neq s'} \neg (P_s(z) \land P_{s'}(z)) \right)\]

An assignment \(v\) is given by predicates \(P_1(z), \ldots, P_k(z)\) corresponding to \(P_1, \ldots, P_k\). A sequence of \(U\) values \(<u_n>\) is given by predicates \(R_u(z)\) \((u \in U)\)
each \(z \in \mathbb{N}\) which is true at each \(z \in \mathbb{N}\). We also need to refer to \(T\) values at
\(z \in \mathbb{N}\) in order to define \(\rho\) maps. Assuming \(T \cap U = \emptyset\), we have
\[\chi_t(z) = R_t(z) \land \bigwedge_{t' \in T, t' \neq t} \neg R_{t'}(z), \quad \chi_u(z) = R_u(z) \land \bigwedge_{u' \in U, u' \neq u} \neg R_{u'}(z),\]
\[\tau_T = \forall z \bigvee_{t \in T} \chi_t(z), \quad \tau_U = \forall z \bigvee_{u \in U} \chi_u(z)\]

Grammars are defined as follows. For a grammar \(G'\), with non-terminals \(V_0, \ldots, V_l\), terminals \(v_1, \ldots, v_m\) and productions \(\text{Prod}'\), define predicates \(V_0(z), \ldots, V_l(z)\)
to correspond to the non-terminals and \(W_1(z), \ldots, W_m(z)\) the terminals. Word
generation by \(G'\) at \(z\) is represented by the formula:
\[\gamma'(z) = V_0(z) \land \bigwedge_{0 \leq h \leq l} \bigvee_{V_h \rightarrow v_j V_h \in \text{Prod}'}[V_h(z') \Rightarrow \bigvee_{V_h \rightarrow v_j V_h \in \text{Prod}'} (W_j(z') \land V_h(z' + 1)) \land \bigwedge_{1 \leq j' \leq m} \neg W_{j'}(z' + 1)] \land \bigwedge_{0 \leq h, h' \leq l, h \neq h', 1 \leq j, j' \leq m, j \neq j'\} \bigwedge_{0 \leq h} \bigwedge_{0 \leq h} \bigwedge_{1 \leq j \leq m} \neg W_j(z') \Rightarrow \bigwedge_{1 \leq j \leq m} \neg W_j(z'')]\]

13
The properties \( (p_{3.9(i)}) \), \( (p_{3.9(iii)}) \), \( (p_{3.9(iv)}) \), \( (p_{3.10(i)}) \) and \( (p_{3.9(ii)}) \) are given below by \( p_{3.9(i)} \), \( p_{3.9(iii)} \), \( p_{3.9(iv)} \), \( p_{3.10(i)} \) and \( p_{3.9(ii)} \) respectively, where \( p_{3.9(iii)} \land p_{3.9(iv)} \) is abbreviated by \( p \). The \( p' \) of \( (p_{3.9(i)}) \) is defined by the \( R_t \) (\( t \in T \)) corresponding to \( c \) in \( p_{3.9(ii)} \) below.

\[
p_{3.9(i)} = \forall z \wedge \bigwedge_{i \in T} \exists \bigwedge_{i \in T} (\tau t \land \chi_i(z) \land \rho)
\]

\[
p_{3.9(ii)} = \wedge \exists \bigwedge_{i \in T} (\tau t \land \rho) \land \forall z \bigwedge_{i \in T} \wedge \land \wedge \exists \bigwedge_{i \in T}
\]

\[
p_{3.9(iii)} = \forall z \bigwedge_{i \in T} \exists \bigwedge_{i \in T} (\tau t \land \rho) \land \forall z \bigwedge_{i \in T} \wedge \land \wedge \exists \bigwedge_{i \in T}
\]

\[
p_{3.9(iv)} = \forall z \bigwedge_{i \in T} \exists \bigwedge_{i \in T} (\tau t \land \rho) \land \forall z \bigwedge_{i \in T} \wedge \land \wedge \exists \bigwedge_{i \in T}
\]

\[
p_{3.10(i)} = \bigwedge_{i \in T} \exists \bigwedge_{i \in T} (\tau t \land \rho) \land \forall z \bigwedge_{i \in T} \wedge \land \wedge \exists \bigwedge_{i \in T}
\]

The predicate \( \xi(f, v, \less u_n \rangle) \) in free variables \( f \), \( v \) and \( \less u_n \rangle \) is given in SIS by a formula

\[
\xi^{SIS}_{f}(P_{s}, P_{1}, \ldots, P_{k}, R_{u} : u \in U, \less u_n \rangle)
\]

in free variables \( P_{s} \) \((s \in \Sigma, P_{1}, \ldots, P_{k}, R_{u} : u \in U) \), and is defined by

\[
\xi^{SIS}_{f}(P_{s}, P_{1}, \ldots, P_{k}, R_{u} : u \in U) = f^{SIS} \land p_{3.9(i)} \land p_{3.9(ii)} \land \tau U
\]

We have the following restatement of Lemma 4.1 in terms of \( \xi^{SIS}_{f} \).

**Lemma 4.2**  
Unary predicates \( P_{s}(z) \) \((s \in \Sigma) \) define a state function of a quasimodel for \((\phi, \Upsilon) \) iff, for all values \( v \in \Upsilon \) for \( P_{1}, \ldots, P_{k} \),

\[
\exists \bigwedge_{u \in U} \exists R_{u} : z \bigwedge_{u \in U} \xi^{SIS}_{f}(P_{s} : s \in \Sigma, P_{1}, \ldots, P_{k}, R_{u} : u \in U)
\]

is true.

The \( \bigcup \binom{Q_{1}P_{1} \ldots Q_{k}P_{k}}{3pos} \) prefix of \( QEMTL^{3pos} \) formulae is dealt with by a Skolem function.
Definition 4.3 A Skolem function for a quantifier prefix $Q_1P_1\ldots Q_kP_k$ for QEMTL (respectively S1S), where $P_1,\ldots,P_k \in \Pi$ (respectively $P_1(z),\ldots,P_k(z)$ are unary predicates in S1S), is a map $\sigma$ that gives a value in $\mathcal{P}(\mathbb{N})$ for each $P_j$ with $Q_j = \exists$, given as argument a sequence of values (each in $\mathcal{P}(\mathbb{N})$) for the $P_i$ such that $i < j$ and $Q_i = \forall$. For such a $\sigma$, we write $\Upsilon_\sigma$ for the set of all assignments to $P_1,\ldots,P_k$, where values for $P_i$ with $Q_i = \forall$ are arbitrary, and values for $P_j$ with $Q_j = \exists$ are given by $\sigma$.

The classical result which relates prefixes and Skolem functions, translates to the following result here:

Lemma 4.4 Given a temporal structure $\mathfrak{M}$ or a structure $\mathfrak{M}^{SIS}$ for S1S over $\mathbb{N}$, and a QEMTL or S1S formula $Q_1P_1\ldots Q_kP_k.\phi$, we have that:

(i) $(\mathfrak{M},0) \models Q_1P_1\ldots Q_kP_k.\phi$ iff $(\mathfrak{M},v,0) \models \phi$ for all $v \in \Upsilon_\sigma$

(ii) $\mathfrak{M}^{SIS} \models Q_1P_1\ldots Q_kP_k.\phi$ iff $\mathfrak{M}^{SIS},v \models \phi$ for all $v \in \Upsilon_\sigma$

for some Skolem function $\sigma$ for $Q_1P_1\ldots Q_kP_k$.

By Lemma 4.4, satisfiability of a prefixed expression $Q_1P_1\ldots Q_kP_k.\phi$ reduces to satisfiability of $\phi$ for a set of values $\Upsilon_\sigma$ defined by a Skolem function $\sigma$. In view of Lemma 4.2, a test for satisfiability of $Q_1P_1\ldots Q_kP_k.\phi$ can be encoded into S1S.

Theorem 4.5 A QEMTL$^{\exists\text{pos}}$ formula $Q_1P_1\ldots Q_kP_k.\phi$ is satisfiable iff the S1S formula

$$\exists \prod_{s \in \Sigma} Q_1P_1\ldots Q_kP_k \exists \prod_{u \in U} R_u \ldots \zeta^{SIS}_\phi (P_s : s \in \Sigma, P_1,\ldots,P_k, R_u : u \in U)$$

(20)

is satisfiable.

Proof A function $f$, defined in S1S by unary predicates $P_s(s \in \Sigma)$, is a state function for a quasimodel $(f,P)$ for $(\phi,\Upsilon_\sigma)$ for some Skolem function $\sigma$ for $Q_1P_1\ldots Q_kP_k$, if, by Lemma 4.2,

$$\exists \prod_{u \in U} R_u \ldots \zeta^{SIS}_\phi (P_s : s \in \Sigma, P_1,\ldots,P_k, R_u : u \in U)$$

is true for all values $v \in \Upsilon_\sigma$ for $P_1,\ldots,P_k$ for some Skolem function $\sigma$, if, by Lemma 4.4(ii),

$$Q_1P_1\ldots Q_kP_k \exists \prod_{u \in U} R_u \ldots \zeta^{SIS}_\phi (P_s : s \in \Sigma, P_1,\ldots,P_k, R_u : u \in U)$$

(21)

is true (for the chosen $P_s$’s). By Lemma 4.4(i), $Q_1P_1\ldots Q_kP_k.\phi$ is satisfiable iff there is a Skolem function $\sigma$ for $Q_1P_1\ldots Q_kP_k$ such that $\phi$ is satisfiable for all
values \( v \in \mathcal{Y}_\sigma \) for \( P_1, \ldots, P_k \). By Theorem 3.13(i), \( \phi \) is satisfiable for all \( v \in \mathcal{Y}_\sigma \) iff there exists a quasimodel \( \langle f, R \rangle \) for \((\phi, \mathcal{Y}_\sigma)\). Therefore, \( Q_1 P_1 \ldots Q_k P_k \phi \) is satisfiable iff there exists a state function \( f \) such that for some Skolem function \( \sigma \) there is a quasimodel \( \langle f, R \rangle \) for \((\phi, \mathcal{Y}_\sigma)\). From (21), this is the case iff

\[
\exists \mathcal{S}_1 \exists P_s \ldots \exists \mathcal{S}_k \exists P_k \exists R_u \ldots \exists \mathcal{S}_\phi \left( \begin{array}{c}
P_s : s \in \mathcal{S}, P_1, \ldots, P_k, R_u : u \in U
\end{array} \right)
\]

is satisfiable.

**Theorem 4.6** Validity for \( QEMTL \) formulae in which live variables occur within positive occurrences of universal quantifiers, i.e. formulae of the form:

\[ Q_1 P_1 \ldots Q_k P_k . \neg \phi(P_1, \ldots, P_k) \]

where \( \phi(P_1, \ldots, P_k) \in EMTL^{\exists_{pos}} \), is decidable when \( \phi \) is restricted to have at most two individual variables, and recursively enumerable otherwise.

**Proof** This corresponds to satisfiability of formulae \( Q_1 P_1 \ldots Q_k P_k . \phi(P_1, \ldots, P_k) \) where \( \phi(P_1, \ldots, P_k) \in EMTL^{\exists_{pos}} \). Such a formula is satisfiable iff the S1S formula (20) of Theorem 4.5 is satisfiable. In the 2-variable case, the set of all realizable state candidates \( \Sigma \) can be computed by Lemmas 3.2 and 3.6, and satisfiability of (20) can be decided as S1S is decidable [6]. In the general case, (20) can be evaluated for all possible sets \( \Sigma \) of (not necessarily realizable) state candidates. By Theorem 4.5, if \( Q_1 P_1 \ldots Q_k P_k . \phi \) is not satisfiable, no set \( \Sigma \) of realizable state candidates will satisfy (20). State candidates in any set \( \Sigma \) satisfying (20) can be tested for realizability in first-order logic as per Lemma 3.6. If \( Q_1 P_1 \ldots Q_k P_k . \phi \) is not satisfiable, as unsatisfiability for first-order logic is recursively enumerable, such a \( \Sigma \) will eventually be shown not to be realizable. Thus, unsatisfiability of \( Q_1 P_1 \ldots Q_k P_k . \phi \) is recursively enumerable.

**Theorem 4.7** The language \( QEMTL^{\exists_{pos}} \) is more expressive than \( EMTL \).

**Proof** It is easy to see that \( QEMTL^{\exists_{pos}} \) is more expressive than \( EMTL \) as it admits formulae such as

\[ \forall P \exists x \Box p(x) \Leftrightarrow P \]

which has only uncountable models, whereas all satisfiable \( EMTL \) formulae have a countable model by Theorem 3.13(ii). An example of a \( QEMTL^{\exists_{pos}} \) formula that has countable models but cannot be expressed in \( EMTL \) is the formula

\[
\forall P_1 \forall P_2 (\Box (P_1 \Box P_2) \land \bigwedge_{i=1,2} \neg P_i \mathcal{U} (P_i \land \Box \neg P_i)) \Rightarrow
\]

\[
\exists x (p(x) \land P_1 \land \Box \Box (\neg p(x) \land P_2)) \lor (\neg p(x) \land P_1 \land \Box \Box (p(x) \land P_2))
\]

This expresses the negation of the 2-sorted temporal logic formula

\[
\exists t_1 \exists t_2 (t_1 < t_2 \land \forall x (p(t_1, x) \Leftrightarrow p(t_2, x))
\]
Theorem 4.8 Let $L$ be the set of formulae of the form:
\[ \exists P_1 \ldots \exists P_k. \phi(P_1, \ldots, P_k) \]
where $\phi(P_1, \ldots, P_k)$ is in EMTL and uses just one individual variable and predicate symbol. Then, the set of valid formulae in $L$ is not recursively enumerable.

Proof We make use of the encoding of the $\Sigma_1^1$-complete recurring tiling problem for $\mathbb{N} \times \mathbb{N}$ [10] into temporal logic with monadic predicates given in [12]. The recurrent tiling problem is to determine whether there is a tiling of $\mathbb{N} \times \mathbb{N}$ by a given set of tiles $\{t_0, \ldots, t_n\}$, where each $t_i$ has fixed orientation and coloured edges $right(t_i)$, $left(t_i)$, $up(t_i)$ and $down(t_i)$, such that $t_0$ occurs infinitely often in the first row. Arguing as in [12] we can show, by associating each $t_i$ with a unary predicate $p_i$, that there is a recurrent tiling of $\mathbb{N} \times \mathbb{N}$ iff the conjunction of the following formulae is satisfiable:

\[
\theta_1 = \exists x \square \diamond (C_0 \land p(x)) \\
\theta_2 = \forall x \exists y (\Box (C_{n+1} \land p(x) \land \Box p(y))) \\
\theta_3 = \forall x \forall y (\Box (C_{n+1} \land p(x) \land \Box p(y)) \Rightarrow \Box (C_{n+1} \land p(x) \land \Box p(y))) \\
\theta_4 = \forall x (\neg \Box (C_{n+2} \land \Box (\neg p(x) [\mathcal{U} C_{n+1}]) \land \Box (\neg p(x) \land \neg C_{n+1} \land \neg C_{n+2} \\
\land \Box (\neg C_{n+1} [\mathcal{U} (p(x) \land \neg C_{n+1})]))) \\
\theta_5 = \forall 0 \leq i \leq n \Box \forall x \exists y (C_0 \land \Box p(x) \land \Box (C_{n+1} \land p(x) \land \Box p(y)) \\
\Rightarrow \forall_{up(t_i) = down(t_i)} \Box p(y)) \\
\theta_6 = \forall 0 \leq i \leq n \Box \forall x (C_i \land p(x) \Rightarrow \forall_{right(t_i) = left(t_i)} \Box (\neg C_{n+i+3} + j + p(x))) \\
\theta_7 = C_{n+1} \land \Box C_{n+2} \land \bigwedge_{0 \leq i \leq n} \Box (C_i \land \Box (\neg C_{n+1} \land \neg C_{n+1})) \\
\land \bigwedge_{0 \leq i \leq n} \bigwedge_{j \leq n+2 \leq n+2} \Box (C_i \land C_j)
\]

The first six formulae correspond to those in Theorem 2 of [12]. To economize on predicate symbols, $\theta_7$ partitions time into ‘periods’ of $n+3$ moments

\[C_{n+1}, C_{n+2}, C_0, \ldots, C_n, C_{n+1}, C_{n+2}, C_0, \ldots, C_n, C_{n+1}, \ldots\]

The predicates $Q_1(x)$ and $Q_2(y)$ in [12] being true at the same moment of time correspond to $p(x)$ and $p(y)$ being true here at consecutive $C_{n+1}$ and $C_{n+2}$. The predicate $P_i(x)$ ($0 \leq i \leq n$) being true in [12] is represented by $p(x)$ being true at $C_i$.

The formulae $\theta_1$, $\theta_4$, $\theta_6$ and $\theta_7$ belong to EMTL. It suffices to translate $\theta_2$, $\theta_3$ and $\theta_5$ into negations of formulae in $L$. These translate thus:

\[
\theta_2 = \forall x \Box (C_{n+1} \land p(x) \land \exists x \Box p(x)) \\
\theta_3 = \forall S_1 \forall S_2 (2x \Box (S_1 \leftrightarrow C_{n+1} \land p(x))) \land \exists x \Box (S_2 \equiv C_{n+2} \land p(x)) \Rightarrow \Box (C_{n+1} \land S_1 \land \Box S_2) \\
\theta_4 = \forall C_{n+1} \land S_1 \land \Box S_2 \Rightarrow \Box (C_{n+1} \land S_1 \land \Box S_2)) \\
\theta_5 = \forall S_{10} \ldots \forall S_{1n} \forall S_{20} \ldots \forall S_{2n} \forall S_1 \forall S_2 ( \exists x \bigwedge_{0 \leq i \leq n} \Box (S_i \equiv C_i \land p(x)) \land
\]

which is shown to be not expressible in first-order ETL in [1] and therefore is not expressible in its monodic fragment EMTL.
\[(S_1 \Leftrightarrow C_{n+1} \land p(x))) \land \exists x \bigwedge_{0 \leq i \leq n} \Box((S_{2i} \Leftrightarrow C_i \land p(x)) \land (S_{2i} \Leftrightarrow C_{n+1} \land p(x))) \Rightarrow \\
\bigwedge_{0 \leq i \leq n} \Box(C_0 \land \Box S_{1i} \land \Diamond(S_{1i} \land \Box S_{2i}) \Rightarrow \bigvee_{\text{up}(t_j) = \text{down}(t_j)} \Box S_{2j}) \big]

The reason for undecidability above in Theorem 4.8 is that, in the absence of restrictions on \(\phi\), propositional quantification allows non-monodic formulae to be simulated monodically. Only the trivial \(\forall P_1 \ldots \forall P_k\) prefix class is recursively enumerable for unrestricted monodic \(\phi\).

5 Conclusions

In [7] monodic temporal logic is extended to include least fixed points such that scopes of individual variable quantifiers do not contain free fixed point variables. The correspondence between models and quasimodels is an adaptation of that in [12] and a similar method would suffice for \(EMTL\) here, though not \(QEMTL_{\exists \text{pos}}\). As with \(EMTL\) here, the existence of models implies the existence of countable models obtained from suitably chosen quasimodels. The work of [19] considers quantification of flexible variables in first-order temporal logic. Decidability of satisfiability is shown for a fragment which also has a restriction to positive occurrences of existential quantifiers, however the result assumes that predicates are rigid.

We have dealt with satisfiability over arbitrary domains. A similar encoding to that given in Theorem 4.8 here, of formulae (1)-(8) of Theorem 3 in [12], would show that validity over finite domains, for propositional quantification at the outer level, is not recursively enumerable without further restrictions. The problems of finding fragments admitting propositional quantification for which validity over finite domains is recursively enumerable, or fragments where propositional quantification occurs within temporal operators and/or individual variable quantification for which validity over arbitrary domains is recursively enumerable, are left open. Also, we have not considered past-time operators or flows of time other than the natural numbers, though some of these cases may be proved by easy additions to the work here.

The aim in this paper has been to demonstrate decidability or recursive enumerability in the most convenient manner. It is possible that the bounds given here (in particular that for \(|T|\)) can be reduced. However, the complexity of \(QEMTL_{\exists \text{pos}}\) is non-elementary as it contains \(QPTL\). A more favourable complexity analysis of monadic \(EMTL\) may be possible by methods similar to those in [11] and [16].

Acknowledgements

I have to thank the anonymous referee for the sacrifice of time in reading several revisions of this paper, and for the many suggestions that have improved the
presentation, accuracy and indeed the main result of the paper.

References


Appendix 1 - proof of Theorem 3.13(i)

Only if

We prove the case where, if $c_i \neq c_j \in C$, for all $v \in \Upsilon$ and $n \in \mathbb{N}$,

$$|\Upsilon| \geq |U|, |D| \geq |T|$$

and $c_i^{J_v(n)} \neq c_j^{J_v(n)}$ (22)

Put $\mathcal{J}_n = \{J_v(n) : v \in \Upsilon\}$. Now, $\text{sub}_x \phi$ comprises $\kappa$ subformulæ, say, $\psi_1, \ldots, \psi_\kappa$ of $\phi$. Define the equivalence relation $\approx_n^\Upsilon$ on $\Upsilon$ by $v_1 \approx_n^\Upsilon v_2$ ($v_1, v_2 \in \Upsilon$) iff

$$\forall a \in D \exists a' \in D \bigwedge_{1 \leq i \leq \kappa} J_{v_1}(n) \models \overline{\psi_i}(a) \Leftrightarrow J_{v_2}(n) \models \overline{\psi_i}(a')$$

(23)

and

$$\forall a' \in D \exists a \in D \bigwedge_{1 \leq i \leq \kappa} J_{v_1}(n) \models \overline{\psi_i}(a') \Leftrightarrow J_{v_2}(n) \models \overline{\psi_i}(a)$$

(24)

and

$$\forall c \in C \bigwedge_{1 \leq i \leq \kappa} J_{v_1}(n) \models \overline{\psi_i}(c^{J_{v_1}(n)}) \Leftrightarrow J_{v_2}(n) \models \overline{\psi_i}(c^{J_{v_2}(n)})$$

(25)
As \( c^{J_{\nu_1}}(n) = c^{J_{\nu_2}}(n) \), an upper bound for the number of equivalence classes of
\( \approx^n_1 \) is \( 2^{2^n} \cdot 2^{\kappa[C]} = 2^{2^{[\text{sub}_x \phi]}} \cdot 2^{[\text{sub}_x \phi][C]} = |U| \) (by Definition 3.3). If, for all \( v \in \Upsilon \), 
\( [v] = \approx^n_1 \) denotes the equivalence class containing \( v \), then as, by (22), \( |\Upsilon| \geq |U| \),
there exist distinct \( v^n_1, \ldots, v^n_{|U|} \in \Upsilon \) such that all equivalence classes are listed in
\[
[v^n_1] \approx^n_1, \ldots, [v^n_{|U|}] \approx^n_1
\]
Now define the equivalence relation \( \approx^n_1 \) on \( D \) by \( a \approx^n_1 a' \) \((a, a' \in D) \) iff
\[
\bigwedge_{1 \leq i \leq n} \bigwedge_{1 \leq j \leq |U|} J_{\nu_i}(n) \models \exists \psi_i(a) \iff J_{\nu_j}(n) \models \exists \psi_j(a') \tag{26}
\]
Here, there are at most \( 2^{2^n} \) equivalence classes and so, by (22), as \( D \) has at least \(|T| = 2^{[\text{sub}_x \phi][|U|]} + |C| = 2^{2^n} + |C| \) elements, there exist distinct
\( a^n_1, \ldots, a^n_{2^{[|U|] + |C|}}, c_1^{J(n)}, \ldots, c_{|C|}^{J(n)} \in D \) such that all equivalence classes are listed in
\[
[a^n_1] \approx^n_1, \ldots, [a^n_{2^{[|U|] + |C|}}] \approx^n_1, [c_1^{J(n)}] \approx^n_1, \ldots, [c_{|C|}^{J(n)}] \approx^n_1
\]
where \( C = \{c_1, \ldots, c_{|C|}\} \) and \( c_i^{J(n)} = c_i^{J(n)} \) \((1 \leq i \leq |C|) \) is the common value in
\( D \) assigned to \( c_i \) by all \( J_{\nu}(n) \in J_n \) for all \( n \in \mathbb{N} \). Henceforth, we will not
distinguish between the individual constant \( c_i \) and its value \( c_i^{J(n)} \).

First of all, we construct a realizable state candidate \( \langle T_n, C_n \rangle \). For each
\( n \in \mathbb{N} \), let
\[
elem T^n : \{a^n_1, \ldots, a^n_{2^{[|U|] + |C|}}, c_1, \ldots, c_{|C|}\} \rightarrow T
\]
and
\[
elem U^n : \{v^n_1, \ldots, v^n_{|U|}\} \rightarrow U
\]
be bijections. Define
\[
\mathcal{T}_n : T \times U \rightarrow \mathcal{P}(\text{sub}_x \phi),
\]
\[
\mathcal{T}_n(elem T^n(a), elem U^n(v)) = \{\psi \in \text{sub}_x \phi : J_{\nu}(n) \models \exists \psi(a)\} \tag{27}
\]
\[
\mathcal{C}_n : C \times U \rightarrow \mathcal{P}(\text{sub}_x \phi),
\]
\[
\mathcal{C}_n(c, elem U^n(v)) = \{\psi \in \text{sub}_x \phi : J_{\nu}(n) \models \exists \psi(c)\} \tag{28}
\]
From now on, we will drop the \( elem T^n \) and \( elem U^n \) and will identify \( elem T^n(a) \) and
\( elem U^n(v) \) with \( a \) and \( v \) respectively. Clearly, \( \langle T_n, C_n \rangle \) is a state candidate.
We show that \( \langle T_n, C_n \rangle \) is realized by the finite family of structures
\[
J_n^F = \{J_{\nu}(n) : v \in U\}
\]
over domain \( D \). For conditions (i) and (ii) of Definition 3.4, suppose that \((t \models a \in T \) and \( a' \in D \) are such that \( a \approx^n_1 a' \), and that \( v \in U \). Then, for all
\( \psi \in \text{sub}_x \phi \),
\[
\psi \in \mathcal{T}_n(a, v)
\]
iff, by (27),
\[
J_{\nu}(n) \models \exists \psi(a)
\]
21
iff, by (26), as $a \approx_n^1 a'$,

$$J_v(n) \models \overline{\psi}(a')$$

iff, by the definition of $tp$,

$$\psi \in tp_{J_v(n)}(a')$$

Thus, $\sum_{x}(a, v) = tp_{J_v(n)}(a')$ if $a \approx_n^1 a'$. As $\approx_n^1$ is an equivalence relation, given $(t \equiv) a \in T$, $a' \in D$ such that $a \approx_n^1 a'$ can be found (e.g. $a' = a$), and, given $a' \in D$, $(t \equiv) a \in T$ such that $a \approx_n^1 a'$ can be found. Conditions (i) and (ii) of Definition 3.4 follow from this. For condition (iii) of Definition 3.4, let $c \in C$ and $a \in D$ be such that $c J_n(a)$, and let $\psi \in u$. Then, for all $\psi \in sub_x \phi$,

$$\psi \in C_n(c, v)$$

iff, by (28),

$$J_v(n) \models \overline{\psi}(c J_n(a))$$

iff, by the definition of $tp$,

$$\psi \in tp_{J_v(n)}(a)$$

We now construct a quasimodel $\langle f, R \rangle$ for $(\phi, \Upsilon)$. The state function $f$ associates with each $n \in \mathbb{N}$ the realizable state candidate $f(n) = \langle \sum_{x}, C_n \rangle$ defined above. For each $a \in D$, $v \in \Upsilon$ and $1 \leq i \leq |T|$, let $r_{a, v, i}$ be a (distinct) run identifier and put

$$R_v = \{r_{a, v, i} : a \in D, 1 \leq i \leq |T|\}, \quad R = \bigcup_{v \in \Upsilon} R_v$$

We have added an $i$ subscript in $r_{a, v, i}$, to allow for sufficiently many run identifiers for each $a$ and $v$, so that each $a'' \in T$ can be mapped to by $R T$ from some $r_{a, v, i}$ as required by Definition 3.9(i) (see (35) below). As, for each $n \in \mathbb{N}$, $\approx_n^2$ is an equivalence relation, then given $v \in \Upsilon$, there exists $u_{vn} \in U$ ($= \{v_{1}^{n}, \ldots, v_{|U|}^{n}\}$) such that

$$u_{vn} \approx_n^2 v \quad (29)$$

Given $a \in D$, by (23) there exists $a' \in D$ such that

$$J_v(n) \models \overline{\psi}(a) \iff J_{u_{vn}}(n) \models \overline{\psi}(a') \quad (\psi \in sub_x \phi) \quad (30)$$

By (26) and the definition of $T$ in terms of $\approx_n^1$, there exists $a'' \in T$ such that

$$J_{u_{vn}}(n) \models \overline{\psi}(a') \iff J_{u_{vn}}(n) \models \overline{\psi}(a'') \quad (\psi \in sub_x \phi) \quad (31)$$

Therefore, by (29), (30) and (31), given $v \in \Upsilon$ there exists $u_{vn} \in U$ such that

$$\forall a \in D \exists a'' \in T. \quad J_v(n) \models \overline{\psi}(a) \iff J_{u_{vn}}(n) \models \overline{\psi}(a'') \quad (\psi \in sub_x \phi) \quad (32)$$
Moreover, by (24),
\[ \forall b' \in T \exists b \in D. \ J_v(n) \models \overline{\psi}(b) \iff J_{a:v:n}(n) \models \overline{\psi}(b') \ (\psi \in \text{sub}_a \phi) \]  
(33)

By (32), there exists a function \( R^T : R \times N \to T \) such that \( R^T(r_{a:v,i}, n) \) is the \( a'' \) corresponding to \( a \) in (32); precisely
\[ J_v(n) \models \overline{\psi}(a) \iff J_{a:v:n}(n) \models \overline{\psi}(R^T(r_{a:v,i}, n)) \ (\psi \in \text{sub}_a \phi) \]  
(34)

As, for each \( \nu \in \Upsilon \) and \( a \in D \), there are \(|T|\) many \( r_{a:v,i} \)'s as \( i \) ranges from 1 to \(|T|\), by (33) there exists a function \( R^T \) satisfying (34) such that every \( b' \in T \) has a \( r_{b:v,i} \) mapped to it; precisely
\[ \forall b' \in T \exists b \in D. \bigvee_{1 \leq i \leq |T|} R^T(r_{b:v,i}, n) = b' \]  
(35)

Furthermore, by (25), there exists \( R^T \) satisfying (34) and (35) such that
\[ R^T(r_{c:v,1}, n) = c_i \]  
(36)

We put
\[ R(r_{a:v,i}, n) = (R^T(r_{a:v,i}), u_{vn}) \ (r_{a:v,i} \in R_v, n \in N) \]  
(37)

and show that \( \langle f, R \rangle \) is a quasimodel for \( (\phi, \Upsilon) \). This requires showing that Definition 3.9(i)-(iv) and Definition 3.10(i) and (ii) hold. Definition 3.9(i) follows from (35). For Definition 3.9(ii), if \( c \in C \) and \( \nu \in \Upsilon \), put \( r_c = r_{c:v,1} (\in R_v) \).

Then, for all \( n \in N \),
\[ T_n(R(r_{c}, n)) = T_n(R^T(r_{c:v,1}, n), u_{vn}) \ (\text{by (37)}) \]
\[ C_n(c, u_{vn}) \ (\text{by (27) and (28)}) \]

For Definition 3.9(iii), we have that (the two \( i \)'s below are not related)
\[ \mathcal{G}^i(\psi_1, \ldots, \psi_m) \in T_n(R(r_{a:v,i}, n)) \]
if by (37),
\[ \mathcal{G}^i(\psi_1, \ldots, \psi_m) \in T_n(R^T(r_{a:v,i}, n), u_{vn}) \]
if, by (27), if \( x \) is the free variable in \( \mathcal{G}^i(\psi_1, \ldots, \psi_m) \),
\[ J_{a:v:n}(n) \models \overline{\mathcal{G}^i(\psi_1, \ldots, \psi_m)}[x/R^T(r_{a:v,i}, n)] \]
if, by (34), as \( \mathcal{G}^i(\psi_1, \ldots, \psi_m) \in \text{sub}_x \phi \),
\[ J_v(n) \models \overline{\mathcal{G}^i(\psi_1, \ldots, \psi_m)}[x/a] \]
if, by (11),
\[ \mathfrak{M}_v, n \models \mathcal{G}^i(\psi_1, \ldots, \psi_m)[x/a] \]

23
iff there is a word $w_nw_{n+1}\ldots$ generated by the corresponding grammar $G^i$, such that, for all $n' \geq n$ and $1 \leq j \leq m$, if $w_{n'} = v_j$ then

$$(\mathfrak{M}, n') \models \psi_j(a)$$

iff, by (11),

$$J_\nu(n') \models \overline{\psi}_j(a)$$

iff, by (34),

$$J_{u_{vn'}}(n') \models \overline{\psi}_j(R_T(r_{a,v,i}, n'))$$

iff, by (27),

$$\psi_j \in T_{n'}(R_T(r_{a,v,i}, n'), u_{vn'})$$

iff, by (37),

$$\psi_j \in T_{n'}(R(r_{a,v,i}, n'))$$

as required. For Definition 3.9(iv), we have that

$$P \in \Pi \cap T_n(R(r_{a,v,i}, n))$$

iff, by (37),

$$P \in \Pi \cap T_n(R_T(r_{a,v,i}, n), u_{vn})$$

iff, by (27),

$$J_{u_{vn}}(n) \models P$$

iff by (34), as $P$ is one of the $\psi$’s,

$$J_\nu(n) \models P$$

iff, by the definition of $J_\nu(n)$ and Definition 3.8,

$$P \in \nu(n)$$

We have defined $f$ to be a state function, so Definition 3.10(i) holds. As Definition 3.9(i)-(iv) have been shown to hold above, it follows that $R$ is a family of runs for $(\phi, \Upsilon)$ with respect to $f$. It remains to show that Definition 3.10(ii) holds, i.e. $\phi \in T_0(R(r_{a,v,i}, 0))$. As $(I(n) \uparrow \nu_n : n \in \mathbb{N})$ satisfies $\phi$, we have that

$$(\mathfrak{M}, 0) \models \phi$$

which implies, by (11), that

$$J_\nu(0) \models \overline{\phi}$$

which implies, by (34), as $\phi$ is closed and thus belongs to $\text{sub}_x \phi$,

$$J_{w,0}(0) \models \overline{\phi}$$

which implies, by (27), that

$$\phi \in T_0(R_T(r_{a,v,i}, 0), u_{w,0})$$
which implies, by (37), that
\[ \phi \in T_0(\mathcal{R}(r_{a,\nu,1}, 0)) \]

If

We prove (18) by induction.

Case \( \psi = P_i \) where \( P_i \notin \Pi \) or \( \psi = p(x_1, \ldots, x_m) \)

By the definition of \( \models^a \) and as \( \bar{\psi} \) contains no surrogates
\[ I_{u_{\nu,n}}(n) \models^a \bar{\psi} \text{ iff } (\mathfrak{M}_{\nu, n}) \models^a \psi \]
from which (18) follows.

Case \( \psi = P_i \) where \( P_i \in \Pi \)

\begin{equation}
P_i \in \mathcal{T}_\nu(\mathcal{R}(r, n)) = tp_{J_{u_{\nu,n}}(n)}(\Delta_{\nu,n}(r, a)) \text{ iff } J_{u_{\nu,n}}(n) \models^a P_i, \tag{38}
\end{equation}

by Lemma 3.12(i) and Definition 3.4. By Definition 3.9(iv) and the definition of \( \mathfrak{M}_{\nu} \),
\begin{equation}
P_i \in \mathcal{T}_\nu(\mathcal{R}(r, n)) \text{ iff } P_i \in v(n) \text{ iff } (\mathfrak{M}_{\nu, n}) \models^a P_i \tag{39}
\end{equation}
As \( I_{u_{\nu,n}}(n) \) is induced from \( J_{u_{\nu,n}}(n) \), (38) and (39) yield
\[ I_{u_{\nu,n}}(n) \models^a P_i \text{ iff } J_{u_{\nu,n}}(n) \models^a P_i \text{ iff } (\mathfrak{M}_{\nu, n}) \models^a P_i \]
from which (18) follows.

Case \( \psi = \neg \psi_1 \)

In this case, \( \zeta_\psi = \text{empty string} \) iff \( \zeta_{\psi_1} = \neg \).

Subcase I: \( \zeta_{\psi_1} = \neg \). By induction,
\[ I_{u_{\nu,n}}(n) \models^a \zeta_{\psi_1} \overline{\psi_1} \text{ implies } (\mathfrak{M}_{\nu, n}) \models^a \zeta_{\psi_1} \psi_1 \]
and so
\[ I_{u_{\nu,n}}(n) \models^a \neg \overline{\psi_1} \text{ implies } (\mathfrak{M}_{\nu, n}) \models^a \neg \psi_1 \]
and so
\[ I_{u_{\nu,n}}(n) \models^a \psi \text{ implies } (\mathfrak{M}_{\nu, n}) \models^a \psi \]
and so, as \( \zeta_\psi = \text{empty string} \),
\[ I_{u_{\nu,n}}(n) \models^a \zeta_{\psi} \overline{\psi} \text{ implies } (\mathfrak{M}_{\nu, n}) \models^a \zeta_{\psi} \psi \]
Subcase 2: $\zeta\psi_1 = \text{empty string}$. By induction,

$$I_{\nu\psi_1}(n) \models^a \zeta\psi_1 \bar{\psi}_1 \text{ implies } (M_{\nu'}, n) \models^a \zeta\psi_1 \psi_1$$

and so

$$I_{\nu\psi_1}(n) \models^a \bar{\psi}_1 \text{ implies } (M_{\nu'}, n) \models^a \psi_1$$

and so

$$I_{\nu\psi_1}(n) \models^a \neg\neg\bar{\psi}_1 \text{ implies } (M_{\nu'}, n) \models^a \neg\neg\psi_1$$

and so

$$I_{\nu\psi_1}(n) \models^a \neg\bar{\psi} \text{ implies } (M_{\nu'}, n) \models^a \neg\psi$$

and so, as $\zeta\psi = \neg$,

$$I_{\nu\psi_1}(n) \models^a \zeta\bar{\psi} \text{ implies } (M_{\nu'}, n) \models^a \zeta\psi$$

Case $\psi = \psi_1 \land \psi_2$

In this case, $\zeta\psi = \zeta\psi_1 = \zeta\psi_2$. By induction,

$$I_{\nu\psi_1}(n) \models^a \zeta\psi_1 \bar{\psi}_1 \text{ implies } (M_{\nu'}, n) \models^a \zeta\psi_1 \psi_1 \quad (40)$$

and

$$I_{\nu\psi_1}(n) \models^a \zeta\psi_2 \bar{\psi}_2 \text{ implies } (M_{\nu'}, n) \models^a \zeta\psi_2 \psi_2 \quad (41)$$

Subcase 1: $\zeta\psi = \zeta\psi_1 = \zeta\psi_2 = \text{empty string}$.

$$I_{\nu\psi_1}(n) \models^a \zeta\psi \bar{\psi} \text{ implies } (M_{\nu'}, n) \models^a \zeta\psi$$

follows immediately from (40) and (41).

Subcase 2: $\zeta\psi = \zeta\psi_1 = \zeta\psi_2 = \neg$. We have that

$$I_{\nu\psi_1}(n) \models^a \neg(\bar{\psi}_1 \land \bar{\psi}_2) \text{ iff } I_{\nu\psi_1}(n) \models^a \neg\bar{\psi}_1 \text{ or } I_{\nu\psi_1}(n) \models^a \neg\bar{\psi}_2 \quad (42)$$

Hence, by (40), (41) and (42), it follows that

$$I_{\nu\psi_1}(n) \models^a \neg(\bar{\psi}_1 \land \bar{\psi}_2) \text{ implies } (M_{\nu'}, n) \models^a \neg(\psi_1 \land \psi_2)$$

i.e.

$$I_{\nu\psi_1}(n) \models^a \zeta\psi_1 \land \psi_2 \text{ implies } (M_{\nu'}, n) \models^a \zeta\psi_1 \land \psi_2$$

Case $\psi = \exists x\psi_1$

Here, $\zeta\psi = \zeta\psi_1$.  

26
Subcase 1: \( \zeta_\psi = \neg \)
\[
I_{u,v}(n) \models^a \zeta_\psi \neg \psi, \text{ i.e. } I_{u,v}(n) \models^a \neg \exists x.\neg \psi_1
\]
implies, for all assignments \( b \) differing from \( a \) only on \( x \),
\[
I_{u,v}(n) \models^b \neg \neg \psi_1
\]
implies, as \( \zeta_\psi = \neg \) and so the number of enclosing \( \neg \)'s of \( \exists x.\psi_1 \) is odd and thus \( x \) cannot be live in \( \psi_1 \), all such \( b \) satisfy (17) trivially and therefore, by induction, for all such \( b \),
\[
(M_v, n) \models^b \zeta_\psi_1 \psi_1 \ (= \neg \psi_1)
\]
iff
\[
(M_v, n) \models^a \neg \exists x.\psi_1
\]
iff
\[
(M_v, n) \models^a \zeta_\psi \psi
\]

Subcase 2: \( \zeta_\psi = \text{empty string} \)
\[
I_{u,v}(n) \models^a \zeta_\psi \neg \psi, \text{ i.e. } I_{u,v}(n) \models^a \exists x.\neg \psi_1
\]
implies, putting \( a'(y) = \Delta_n(a(y)) \) for all \( y \), as \( \Delta_n \) induces \( I_{u,v} \),
\[
J_{u,v}(n) \models^a \exists x.\neg \psi_1
\]
implies (we give the more difficult case when \( x \) is live in \( \psi_1 \) as, if \( x \) is not live, there is no requirement that \( r \in R_v \) for \( b(x) = (r, a) \) below), by (4) of Definition 3.5,
\[
J_{u,v}(n) \models^{\Delta_n, a', u,v, \psi_1} \exists x.\neg \psi_1 \tag{43}
\]
implies, as \( \Delta_{v,n} \) is a surjection, choosing \( r \in R_v \) and \( a \in D \) such that
\[
\Delta_n(r, a) = \Delta_{v,n}(r, a) = \Delta_{a', u,v, \psi_1}(x)
\]
and putting
\[
b(y) = a(y) \ (y \neq x), \quad b(x) = (r, a) \tag{44}
\]
we have (from (43), as \( \Delta_{a', u,v, \psi_1}(y) = a'(y) \) (by (3) of Definition 3.5) wherever \( b(y) = a(y) \), and \( \Delta_n \) induces \( I_{u,v} \) from \( J_{u,v} \)), by Lemma 3.1,
\[
I_{u,v}(n) \models^b \neg \psi_1
\]
implies, by induction (condition (18) can be applied as, by (44), \( x \) is reassigned to \( (r, a) \) for some \( r \in R_v \) and therefore (17) still holds),
\[
(M_v, n) \models^b \psi_1
\]
implies
\[
(M_v, n) \models^a \exists x.\psi_1
\]
i.e.

\[(\mathfrak{M}_u, n) \models^a \zeta_{\psi} \psi\]

Case \(\psi = G^i\)

Put \(\psi(x) = G^i(\psi_1(x), \ldots, \psi_m(x))\). Note that \(\zeta_\psi = \zeta_{\psi_1} = \ldots = \zeta_{\psi_m}\).

Subcase 1: \(\zeta_\psi = \text{empty string}\)

\[I_{u_{\psi n}}(n) \models^a \zeta_{\psi} \overline{\psi}, \text{ i.e } I_{u_{\psi n}}(n) \models^a \overline{\psi(x)}\]

implies, for some \(a \in D, v' \in \Upsilon\) and \(r \in R_{v'}\) such that \(a(x) = (r, a)\),

\[I_{u_{\psi n}}(n) \models p_\psi((r, a))\]

implies, as \(\Delta_n\) induces \(I_{u_{\psi n}}(n)\),

\[J_{u_{\psi n}}(n) \models p_\psi(\Delta_n(r, a))\]

implies, as \(p_\psi \in \text{surr } \phi - \text{surr } P\phi\) if \(x\) is not live in \(\psi\) and \(v' = v\) (by (17)) if \(x\) is live in \(\psi\),

\[J_{u_{v' n}}(n) \models p_\psi(\Delta_n(r, a))\]

implies, as \(\Delta_n(r, a) = \Delta_{v' n}(r, a)\),

\[J_{u_{v' n}}(n) \models p_\psi(\Delta_{v' n}(r, a))\]

implies, by the definition of \(\text{tp}_{J_{u_{v' n}}(n)}(\Delta_{v' n}(r, a))\) (Definition 3.4) and Lemma 3.12(i),

\[\psi(x) = G^i(\psi_1(x), \ldots, \psi_m(x)) \in \text{tp}_{J_{u_{v' n}}(n)}(\Delta_{v' n}(r, a)) = T_n(R(r, n))\]

implies, by Definition 3.9(iii), there is a word \(w_n w_{n+1} \ldots\) generated by \(G^i\) such that, for all \(n' \geq n\), if \(w_{n'} = \psi_j\) then

\[\psi_j(x) \in T_n(R(r, n'))\]

implies, by Lemma 3.12(i),

\[\psi_j(x) \in T_n(R(r, n')) = \text{tp}_{J_{u_{v' n}}(n')}(\Delta_{v' n'}(r, a))\]

implies, by the definition of \(\text{tp}_{J_{u_{v' n'}}(n')}(\Delta_{v' n'}(r, a))\),

\[J_{u_{v' n'}}(n') \models \overline{\psi_j(\Delta_{v' n'}(r, a))}\]

implies as, if \(x\) is not live in \(\psi\), then \(\psi\) and therefore its subformula \(\psi_j\) do not contain any \(P \in \Pi\) and so \(\overline{\psi_j}\) only contains symbols in \(\text{surr } \phi - \text{surr } P\phi\) (and thus, by an easy induction, we have that, for all \(\xi \leq \psi_j, n \in \mathbb{N}\), assignments \(a'\)
to elements of $D$, and $u, u' \in U$, $J_u(n) \models^{a_f} \xi$ iff $J_{u'} \models^{a'_f} \xi$ and, if $x$ is live in $\psi$, then $u' = v$,

$$J_{u_{n'}}(n') \models \overline{\psi}_j(\Delta_{n'}'(r, a))$$

implies, as $\Delta_{n'}'(r, a) = \Delta_{n'}(r, a)$ and $\Delta_{n'}$ induces $I_{u_{n'}}(n')$,

$$I_{u_{n'}}(n') \models \overline{\psi}_j((r, a))$$

implies there is a word $w_n w_{n+1} \ldots$ generated by $G^i$ such that, for $n' \geq n$ and $w_{n'} = v_j$,

$$I_{u_{n'}}(n') \models^{a} \psi_j(x)$$

implies, by induction (if $x$ is live in $\psi_j$ then $x$ is plainly live in $\psi$ and so by hypothesis $a(x) \in R_{a'} \times D$ and the inductive hypothesis can be applied) as $\zeta_{\psi_j} = \zeta_{\psi} = \emptyset$, there is a word $w_n w_{n+1} \ldots$ generated by $G^j$ such that, for $n' \geq n$ and $w_{n'} = v_j$

$$(M_{w}, n') \models^{a} \psi_j(x)$$

implies

$$(M_{w}, n) \models^{a} G^i(\psi_1(x), \ldots, \psi_m(x)) (= \zeta_{\psi})$$

Subcase 2: $\zeta_{\psi} = \neg$

Given that $\zeta_{\psi} = \neg$, we need to prove that

$$I_{u_{n}}(n) \models^{a} \neg \psi \implies (M_{w}, n) \models^{a} \neg \psi$$

i.e.

$$(M_{w}, n) \models^{a} \psi \implies I_{u_{n}}(n) \models^{a} \neg \psi$$

Letting $a \in D$, $v' \in \Upsilon$ and $r \in R_{a'}$ be such that $a(x) = (r, a)$,

$$(M_{w}, n) \models^{a} \psi$$

i.e.

$$(M_{w}, n) \models^{a} G^i(\psi_1, \ldots, \psi_m)$$

implies there is a word $w_n w_{n+1} \ldots$ generated by $G^i$ such that, for $n' \geq n$ and $w_{n'} = v_j$,

$$(M_{w}, n') \models^{a} \psi_j(x)$$

implies (as $\zeta_{\psi_j} = \zeta_{\psi} = \neg$ and therefore, by induction, $I_{u_{n'}}(n') \models^{a} \neg \psi_j(x)$ implies $(M_{w}, n') \models^{a} \neg \psi_j(x)$, i.e. $(M_{w}, n') \models^{a} \psi_j(x)$ implies $I_{u_{n'}}(n') \models^{a} \psi_j(x)$)

$$I_{u_{n'}}(n') \models^{a} \overline{\psi}_j((r, a))$$

implies

$$I_{u_{n'}}(n') \models \overline{\psi}_j((r, a))$$
implies, as $\Delta_{n'}(r, a) = \Delta_{n'}(r, a)$ and $\Delta_{n'}$ induces $I_{u_{n'}}(n')$,

$$J_{u_{n'}}(n') \models \overline{\psi}_j(\Delta_{n'}(r, a))$$

implies, as $\overline{\psi}_j$ only contains symbols in $surr\phi - surr\rho \phi$ if $x$ is not live in $\psi$, and $\nu' = \nu$ if $x$ is live in $\psi$,

$$J_{u_{n'}}(n') \models \overline{\psi}_j(\Delta_{n'}(r, a))$$

implies, by the definition of $tp_{J_{u_{n'}}(n')}(\Delta_{n'}(r, a))$,

$$\psi_j(x) \in tp_{J_{u_{n'}}(n')}(\Delta_{n'}(r, a))$$

implies, by Lemma 3.12(i) (there is a word generated by $G'$ such that, for $n' \geq n$, if $w_{n'} = v_j$ then),

$$\psi_j \in tp_{J_{u_{n'}}(n')}(\Delta_{n'}(r, a)) = T_{n'}(R(n'))$$

implies, by Definition 3.9(iii) and Lemma 3.12(i),

$$\psi(x) = G'(\psi_1, \ldots, \psi_m) \in T_n(R(r, n)) = tp_{J_{u_{n'}}(n)}(\Delta_{n'}(r, a))$$

implies, by the definition of $tp_{J_{u_{n'}}(n)}(\Delta_{n'}(r, a))$,

$$J_{u_{n'}}(n) \models p_\psi(\Delta_{n'}(r, a))$$

implies, as $\Delta_{n}(r, a) = \Delta_{n'}(r, a)$,

$$J_{u_{n}}(n) \models p_\psi(\Delta_{n}(r, a))$$

implies, as $p_\psi \in surr\phi - surr\rho \phi$ if $x$ is not live in $\psi$, and $\nu' = \nu$ if $x$ is live in $\psi$,

$$J_{u_{n}}(n) \models p_\psi(\Delta_{n}(r, a))$$

implies, as $\Delta_{n}$ induces $I_{u_{n}}(n)$,

$$I_{u_{n}}(n) \models p_\psi((r, a))$$

implies

$$I_{u_{n}}(n) \models \overline{\psi}$$

\[\blacksquare\]

**Appendix 2 - proof of Lemma 4.1**

**Only if**

Let $(f, R)$ be a quasimodel for $(\phi, \mathcal{Y})$ as in Definitions 3.9 and 3.10. Then, by Definition 3.9, for each $v \in \mathcal{Y}$, there is a sequence

$$<u_n : n \in \mathbb{N}> = <u_{v_n} : n \in \mathbb{N}>$$
of elements of \( U \), and a function \( R^T : R_v \times \mathbb{N} \to T \) satisfying (5), Definition 3.9(i), (iii) and (iv), and Definition 3.10(ii). Also, for all \( c \in C \), there is a \( r_c \in R_v \), satisfying Definition 3.9(ii). For all \( n \in \mathbb{N} \) and \( t \in T \), consider \( \rho_{n,t} : \mathbb{N} \to T \) defined by

\[
\rho_{n,t}(m) = R^T(r, m) \quad (m \in \mathbb{N})
\]

where \( r \) is as in Definition 3.9(i), so that \( \rho_{n,t}(n) = R^T(r, n) = t \) and therefore \((p_{3.9(i)})\) is satisfied. By Definition 3.9(iii) and (iv) and Definition 3.10(ii), the sequence

\[
<f(m)\mathcal{L}(\rho_{n,t}(m), u_m) : m \in \mathbb{N} > = <T_m(R^T(r, m), u_{vm}) : m \in \mathbb{N} >
\]

satisfies \((p_{3.9(iii)})\), \((p_{3.9(iv)})\) and \((p_{3.10(ii)})\) respectively. Also, for all \( c \in C \), there exists \( \rho^c : \mathbb{N} \to T \) such that, for all \( m \in \mathbb{N} \),

\[
\rho^c(m) = R^T(r_c, m)
\]

and so

\[
f(m)\mathcal{L}(\rho^c(m), u_m) = T_m(R^T(r_c, m), u_{vm}) = T_m(R(r_c, m)) \quad \text{(by (5))}
\]

As \((p_{3.9(iii)})\), \((p_{3.9(iv)})\) and \((p_{3.10(ii)})\) can be shown to be satisfied for \( \rho^c \) as above, it follows that \((p_{3.9(ii)})\) is satisfied.

If

Conversely, suppose that, for all \( v \in \mathcal{V} \), there exists \( <u_n> \) such that, for the given \( f \), \( \xi_\phi(f, v, <u_n>) \) holds. With this \( f \), we construct a quasimodel \( \langle f, R \rangle \) for \((\phi, \mathcal{Y})\). Let \( v \in \mathcal{Y} \), and let \( <u_n> \) be such that \( \xi_\phi(f, v, <u_n>) \) holds. Put \( u_{vn} = u_n \ (n \in \mathbb{N}) \) and

\[
R_v = (\mathbb{N} \times T) \cup C
\]

For all \( r \in R_v \) and \( m \in \mathbb{N} \), we need to define \( R(r, m) \) and so we need a value \( R^T(r, m) \in T \). Put

\[
R^T(r, m) = \rho_{n,t}(m)
\]

from the definition of \( \xi_\phi(f, v, <u_n>) \), for all \( r = (n, t) \in R_v \), and put

\[
R^T(c, m) = \rho^c(m)
\]

for all \( c \in C \subseteq R_v \). Then, Definition 3.9(i) is satisfied as, given \( n \in \mathbb{N} \) and \( t \in T \), taking \( r = (n, t) \in R_v \), we have, by \( p_{3.9(i)} \),

\[
R^T(r, n) = \rho_{n,t}(n) = t
\]

Definitions 3.9(iii) and (iv) hold for all \( r = (n, t) \in R_v \) as

\[
<T_m(R(r, m)) : m \in \mathbb{N} >= <T_m(R^T(r, m), u_{vm}) : m \in \mathbb{N}>
\]

31
\[ T_{0}(r,0) = f(0)L(\rho_{n,t}(0), u_{0}) \]

contains \( \phi \). Definitions 3.9(iii) and (iv) and Definition 3.10(ii) hold for \( r = c \in R_{\upsilon} \) similarly. Definition 3.9(ii) holds as, taking \( r_{c} = c \in R_{\upsilon} \), for all \( m \in \mathbb{N} \),

\[ T_{m}(R(r_{c}, m)) = T_{m}(R^{T}(c, m), u_{\upsilon m}) = f(m)L(\rho^{c}(m), u_{m}) \]

\[ = f(m)L(c, u_{m}) \text{ (by (\rho_{3.9(iii)}) = C_{m}(c, u_{\upsilon m})} \]