Model Checking Gossip Modalities*

Michaela Huhn[1] and Peter Niebert[2] and Frank Wallner[3]

[1] Institut für Rechnerentwurf und Fehlertoleranz (Prof. D. Schmid), Univ. Karlsruhe, Postf. 6980, D-76128 Karlsruhe, huhn@ira.uka.de
[2] Institut für Informatik (Prof. U. Goltz), Univ. Hildesheim, Postf. 101363, D-31113 Hildesheim, niebert@informatik.uni-hildesheim.de
[3] Institut für Informatik, Technische Univ. München, D-80290 München, wallnerf@in.tum.de

Abstract. We present a model checking technique for $\mathcal{L}_{CSA}$, a temporal logic for *communicating sequential agents* (CSAs) introduced by Lodaya, Ramanujam, and Thiagarajan. $\mathcal{L}_{CSA}$ contains temporal modalities indexed with a local point of view of one agent and allows to refer to properties of other agents according to the *latest gossip* which is related to *local knowledge*.
The model checking procedure relies on a modularisation of $\mathcal{L}_{CSA}$ into temporal and gossip modalities. We introduce a hierarchy of formulae and a corresponding hierarchy of equivalences, which allows to compute for each formula and finite state distributed system a finite multi modal Kripke structure, on which the formula can be checked with standard techniques.

1 Introduction

A reasonable and lucid way of formally treating distributed systems is to consider them as a fixed collection of sequential components (agents) which can operate independently as well as cooperate by exchanging information. There is an increasing awareness, both in theory and practice, of the benefits of specifying the requirements of such systems by *localised*, component based formalisms that allow to refer to properties of the individual components.

The operational models for localised specification usually consist of *local* temporal orders (sequences in the linear time case, trees in branching time) together with an interrelation between these orders, descended from communication [LRT92,Ram95]. The most established models for the linear time case are partial orders, whereas in the branching time setting, *(prime) event structures* or closely related models like *occurrence nets* [NPW80,Win87] have been recognised to be a suitable formalism. In these models, partial orders are extended by an additional conflict relation, representing the moments of choice.

* This work was partially supported by the DFG within SFB 342, Teilprojekt A3, and within the priority program "*Design and design methodology of embedded systems*" and by the EEC program on *Training and Mobility in Research* (TMR).

Investigating partial order models has attained the interest of researchers for mainly two reasons: There is no distinction among computations that are equal up to possible total orderings of independent actions, which makes it a faithful and natural formalism for representing concurrency. Furthermore, restricting the attention to local states mitigates one of the most tackled difficulty of model checking, the so-called *state explosion problem*, which results from an explicit computation of the global state space of a distributed system.

For a component-oriented specification of behaviour, local linear time temporal logics have been investigated by Thiagarajan in [Thi94,Thi95] and Niebert [Nie98]. Local branching time logics were introduced in [LT87,LRT92,HNW98]. While for the linear time case there now exist sound model checking procedures based on automata [Thi94,Nie98], only recently the model checking problem for similar branching time logics has been inspected [Pen97,HNW98].

In this paper, we investigate model checking for a local branching time logic defined by Lodaya, Ramanujam and Thiagarajan in [LRT92], in the sequel called $\mathcal{L}_{CSA}$, which is intended to specify the behaviour of *communicating sequential agents* (CSAs). It allows a component i to refer to local properties of another component j according to the *latest gossip*, i.e., the most recent j-local state that causally precedes the current i-local state. This notion occurs in asynchronous network protocols, where several agents together perform a task without global synchronisation. In [LRT92], the authors instead describe this concept by referring to *local knowledge*.

Based on net unfoldings [Eng91], in particular McMillan's prefix construction [McM92], we solve the model checking problem for $\mathcal{L}_{CSA}$, which has remained open since [LRT92].

McMillan's prefix has successfully been applied to alleviate state explosion in many verification problems, for instance deadlock detection [McM92], and model checking S4 [Esp94], LTL [Wal98], and the distributed μ-calculus [HNW98]. All of the previous problems can principally be solved also with conventional state space exploration, but often with an exponentially higher effort than can be achieved using McMillan's prefix.

In contrast, the focus of this paper is to show decidability of model checking $\mathcal{L}_{CSA}$ generalising techniques developed in [HNW98]. We demonstrate that the unfolding approach yields a suitable data structure for solving the model checking problem for a wider class of local logics, for which previously the problem appeared to be too difficult. Moreover, we claim that the result shows the direction to solve the model checking for languages including knowledge operators [Pen98]. Having opened the general path to automatic verification for $\mathcal{L}_{CSA}$ and related logics, we leave the investigation of efficient techniques for future work.

Technically, we proceed as follows. We redefine the semantics of $\mathcal{L}_{CSA}$ on net unfoldings[1] and factorise the net unfolding with respect to an equivalence relation satisfying two key properties: It is a congruence for the $\mathcal{L}_{CSA}$-specification to be checked and it has finite index. Via the factorisation an $\mathcal{L}_{CSA}$ model checking problem can be transformed into a model checking problem for a multi modal

[1] The original definition [LRT92] is on an event structure model.

logic on a finite transition system computed from a modified McMillan prefix which uses the defined equivalence relation as cutoff condition. With an appropriate interpretation of the $\mathcal{L}_{CSA}$ modalities, on this transition system standard model checking algorithms can be applied, e.g. [CES86].

The approach follows the lines of [HNW98], but whereas the focus in [HNW98] was to derive an algorithm for the calculation of the transition system, the main difficulty here is to develop an appropriate equivalence relation.

A major similarity with the distributed μ-calculus of [HNW98] is that $\mathcal{L}_{CSA}$ looks at the state of a system from a local point of view. Technically, the smoothness of the algorithms and the almost immediate usability of McMillan's prefix in [HNW98] relies on the pure future character of the modalities of the distributed μ-calculus and similarly of the fragment of the logic DESL investigated in [Pen97]. As a consequence, the equivalence used in [HNW98] is close to McMillan's original cutoff condition and was fixed for arbitrary formulae of the logic.

In contrast, the gossip and the past modalities of $\mathcal{L}_{CSA}$ are not pure future modalities so that with increasing complexity formulae can refer to increasingly complex patterns in the past of a configuration. As a consequence, the coarsest equivalence preserving all $\mathcal{L}_{CSA}$ properties has non-finite index and it is not possible to construct a single transition system representing all $\mathcal{L}_{CSA}$ properties of a particular finite state distributed system. However, a single $\mathcal{L}_{CSA}$ formula has a limited power of looking into the past so that we can still construct a formula dependent equivalence. For this purpose, we introduce a hierarchy of properties and of corresponding equivalences. The construction of these equivalences and the proof of their soundness are both difficult, and the resulting model checking complexity of the construction given here is high.

For the technical presentation of the whole paper including the semantics of the logics we use notions from Petri net theory, in particular because of the prevalence of this formalism with respect to McMillan's unfoldings. Note however, that the entire method can easily be restated for other formalisms, like e.g. asynchronous automata, coupled finite state machines, and so forth.

The paper is structured as follows. In Section 2 we introduce basic definitions of our models, distributed net systems as Petri net representation of communicating sequential agents, and net unfoldings as semantic model of the branching behaviour of such systems. In Section 3 we introduce the logic $\mathcal{L}_{CSA}$ and our modularisation and embedding in the slightly more general logic $\mathcal{L}$. In Section 4 we introduce the McMillan prefix of net unfoldings in a form parametrised by an abstract equivalence relations which has to meet certain restrictions (in particular it must be of finite index and decidable). Then we give appropriate equivalences for $\mathcal{L}^+$, the fragment of $\mathcal{L}$ without past, and for $\mathcal{L}$ and show the preservation of properties under these equivalences. In Section 5 we use these equivalences to compute a finite state transition system, so that the original model checking problem for an $\mathcal{L}$ formula is reduced to a standard model checking problem for a straight forward interpretation of the formula over the computed system. Thus, we obtain a decision procedure. In Section 6, we discuss our results and indicate possible future work.

2 Distributed net systems and their unfoldings

Petri nets. Let P and T be disjoint, finite sets of *places* and *transitions*, generically called *nodes*. A *net* is a triple $N = (P, T, F)$ with a *flow relation* $F \subseteq (P \times T) \cup (T \times P)$. The *preset* of a node x is defined as ${}^{\bullet}x := \{y \in P \cup T \mid yFx\}$ and its *postset* as $x^{\bullet} := \{y \in P \cup T \mid xFy\}$. The preset (postset) of a set X of nodes is the union of the presets (postsets) of all nodes in X.

A *marking* of a net is a mapping $M : P \rightarrow \mathbb{N}_0$. If $M(p) = n$, we say that p contains n *tokens* at M. We call $\Sigma = (N, M_0)$ a *net system* with initial marking M_0 if N is a net and M_0 a marking of N. A marking M *enables* the transition t if every place in the preset of t contains at least one token. In this case the transition can *occur*. If t occurs, it removes one token from each place $p \in {}^{\bullet}t$ and adds one token to each place $p' \in t^{\bullet}$, yielding a new marking M'. We denote this occurrence by $M \xrightarrow{t} M'$. If there exists a chain $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} \ldots \xrightarrow{t_n} M_n$ for $n \geq 0$, then the sequence $t_1 t_2 \ldots t_n$ is called *occurrence sequence*, and the marking M_n is a *reachable marking*.

We will restrict our attention to *1-safe net systems*, in which every reachable marking M puts at most one token on each place, and thus can be identified by the subset of places that contain a token, i.e., $M \subseteq P$.

In the last years, 1-safe net systems have become a significant model [CEP95]. In [NRT90] it has been shown that an instance of 1-safe nets, called Elementary Net Systems, correspond to other models of concurrency, such as (Mazurkiewicz) traces and prime event structures. They can naturally be interpreted as a synchronised product of several finite automata, and thus are frequently used as a convenient formalism for modelling distributed systems. In the following we will exploit this compositional view by considering the notion of *locations*.

Distributed net systems. Let us introduce the formalism for describing distributed systems. Clearly, the *behaviour* of our models shall resemble the *Communicating Sequential Agents* of [LRT92]. This means, a system consists of several (spatially) distributed, autonomous agents, which mutually communicate. Each of the agents shall exhibit a strictly sequential, non-deterministic behaviour.

Let Σ be a 1-safe net system, and t, t' two transitions of Σ. A marking M *concurrently enables t and t'* if M enables t, and $(M \setminus {}^{\bullet}t)$ enables t'. We call Σ *sequential* if no reachable marking concurrently enables two transitions.

Let $\{\Sigma_i = (P_i, T_i, F_i, M_i^0) \mid i \in Loc\}$ be a family of 1-safe, sequential net systems with pairwise disjoint sets P_i of places, indexed by a finite set Loc of *locations*. The sets of transitions are not necessarily disjoint. In fact we interpret the execution of a transition that is common to several locations as a *synchronous communication* of these agents. A *distributed net system* $\Sigma_{Loc} = (N, M_0)$ is defined as the union of its components Σ_i:

$$P = \bigcup_{i \in Loc} P_i \ , \quad T = \bigcup_{i \in Loc} T_i \ , \quad F = \bigcup_{i \in Loc} F_i \ , \quad M_0 = \bigcup_{i \in Loc} M_i^0 \ .$$

Clearly, Σ_{Loc} is again 1-safe. The intention is to interpret such a system as a collection of sequential, non-deterministic agents with communication capabilities, namely the common execution of a joint transition. The *location* $loc(x)$ of a node x is defined by $loc(x) := \{i \in Loc \mid x \in P_i \cup T_i\}$. A simple distributed net system consisting of two components is depicted in Fig. 1.

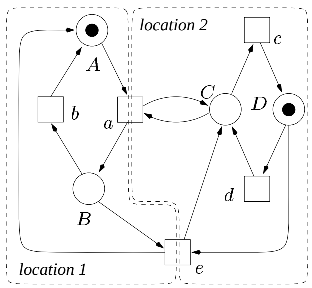

Fig. 1. Distributed net

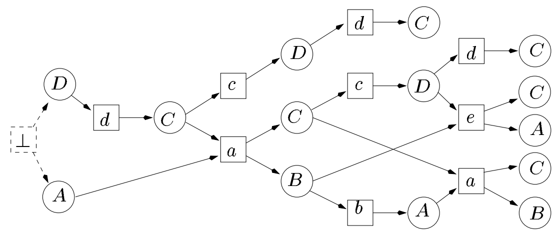

Fig. 2. Branching process

In [LRT92] also *asynchronous* communication (message passing) is considered. However, in general this leads to systems not only with an infinite behaviour, but also with infinitely many states, making an algorithmic, state space based approach to model checking impossible. To model asynchronous communication in the setting of distributed net systems we assume some (finite-state) communication-mechanisms like e.g. bounded channels or buffers. For instance, a buffer can be considered as an agent on its own, (synchronously) communication with both the agents that communicate asynchronously via this buffer.

Net unfoldings. As a partial order semantics of the behaviour of a distributed net system, we consider *net unfoldings*, also known as *branching processes*. They contain information about both concurrency and conflict.

Two nodes x, x' of a net (P, T, F) are *in conflict*, denoted $x \# x'$, if there exist two distinct transitions t, t' such that $^\bullet t \cap {}^\bullet t' \neq \emptyset$, and $(t, x), (t', x')$ belong to the reflexive, transitive closure of F. If $x \# x$, we say *x is in self-conflict*.

An *occurrence net* [NPW80] is a net $N' = (B, E, F)$ with the following properties: (1) for every $b \in B$, $|^\bullet b| \leq 1$, (2) the irreflexive transitive closure $<$ of F is well-founded and acyclic, i.e., for every node $x \in B \cup E$, the set $\{y \in B \cup E | y < x\}$ is finite and does not contain x, and (3) no element $e \in E$ is in self-conflict. The reflexive closure $\leq$ of $<$ is a partial order, called *causality relation*. In occurrence nets we speak of *conditions* and *events* instead of places and transitions, respectively. $Min(N')$ denotes the minimal elements of N' w.r.t. $\leq$.

Given two nets N_1, N_2, the mapping $h : P_1 \cup T_1 \to P_2 \cup T_2$ is called a *homomorphism* if $h(P_1) \subseteq P_2, h(T_1) \subseteq T_2$, and for every $t \in T_1$ the restriction of h to $^\bullet t$, denoted $h|_{^\bullet t}$, is a bijection between $^\bullet t$ and $^\bullet h(t)$, and similar for $h|_{t^\bullet}$.

A *branching process* [Eng91] of a net system $\Sigma = (N, M_0)$ is a pair $\beta = (N', \pi)$ where $N' = (B, E, F)$ is an occurrence net and $\pi : N' \to N$ is a homomorphism, such that the restriction of π to $Min(N')$ is a bijection between $Min(N')$ and M_0, and additionally for all $e_1, e_2 \in E$: if $\pi(e_1) = \pi(e_2)$ and ${}^\bullet e_1 = {}^\bullet e_2$ then $e_1 = e_2$. Loosely speaking, we unfold the net N to an occurrence net N', such that each node x of N' refers to node $\pi(x)$ of N. Two branching processes β_1, β_2 of Σ are *isomorphic* if there exists a bijective homomorphism $h : N_1 \to N_2$, such that the composition $\pi_2 \circ h$ equals π_1. In [Eng91] it is shown that each net system Σ has a unique maximal branching process up to isomorphism, which we call the *unfolding* of Σ, and denote by $Unf_\Sigma = (N', \pi)$.

Let $N_1 = (B'', E'', F'')$ be a subnet of N', such that $e \in E''$ implies $e' \in E''$ for every $e' < e$, and $B'' = Min(N') \cup E''^\bullet$, and let π'' be the restriction of π onto the nodes of N''. We call $\beta'' = (N'', \pi'')$ a *prefix* of Unf_Σ. Fig. 2 shows a prefix of the infinite unfolding of the net system drawn in Fig. 1.

In distributed net systems, the location $loc(x)$ of a node x of N' is given by $loc(x) = loc(\pi(x))$. By $E_i := \{e \in E \mid i \in loc(e)\}$, we denote the set of *$i$-events*.

Configurations and Cuts. For the remainder of the section, let us fix the unfolding $Unf_\Sigma = (N', \pi)$ of the distributed net system Σ with $N' = (B, E, F)$.

A *configuration* $C \subseteq E$ is a causally downward-closed, conflict-free set of events, i.e., $\forall\, e \in C$: if $e' \leq e$ then $e' \in C$, and $\forall\, e, e' \in C : \neg(e \# e')$. A finite configuration describes the initial part of a computation of the system. If we understand the *states* of the system as moments in time, then configurations represent the *past* (by exhibiting all the events that have occurred so far, and the causal structure among them), as well as the *present* and the *future*, as formalised in the following.

Two nodes of N' are *concurrent* if they are neither in conflict nor causally related. A set $B' \subseteq B$ of conditions of N' is called a *cut* if B' is a maximal set of pairwise concurrent conditions. Every finite configuration C determines a cut $Cut(C) := (Min(N') \cup C^\bullet) \setminus {}^\bullet C$. The corresponding set $\pi(Cut(C)) \subseteq P$ of places is a reachable marking of Σ, denoted by $\mathcal{M}(C)$ and called *the state of* C. Notice that for every reachable marking M of Σ, there exists a (not necessarily unique) finite configuration with state M. We will often identify configurations with their state. Given a configuration C and a disjoint set E' of events, we call $C \oplus E'$ an *extension* of C if $C \cup E'$ is a configuration.

Let $\uparrow C := \{x \in (B \cup E) \mid \exists b \in Cut(C).\ b \leq x \quad \text{and} \quad \forall y \in C.\ \neg(x \# y)\}$. The *(branching) future* of a configuration C is given by the branching process $\beta(C) := (N'_C, \pi_C)$, where N'_C is the unique subnet of N' whose set of nodes is $\uparrow C$, and π_C is the restriction of π onto the nodes of N'_C. Let us call two configurations *$\mathcal{M}$-equivalent*, denoted $C \equiv_\mathcal{M} C'$, if $\mathcal{M}(C) = \mathcal{M}(C')$. It is easy to show that if $C \equiv_\mathcal{M} C'$ then there exists an isomorphism $I_C^{C'}$ from $\beta(C)$ to $\beta(C')$. It induces a mapping from the extensions of C onto the extensions of C', mapping $C \oplus E'$ onto $C' \oplus I_C^{C'}(E')$, which are again $\mathcal{M}$-equivalent.

Local states and views. The notion of *local state* arises by considering configurations that are determined by single events. For an event e, we call the set $\downarrow e := \{e' \in E \mid e' \leq e\}$ the *local configuration of* e. It is indeed a configuration, because no event is in self-conflict. If $e \in E_i$ is an i-event, we consider $\downarrow e$ to be an i*-local state*. It determines the local past of component i, as well as the local past of every component that has communicated with i so far — directly, or indirectly via other components. In the sequel, we will often identify an event and its local configuration.

In distributed net systems, we define the i*-view* $\downarrow^i C$ of a configuration C as $\downarrow^i C := \{e \in C \mid \exists e_i \in (C \cap E_i). e \leq e_i\}$. Notice that the sequentiality of the subsystems implies that for each $i \in Loc$, the i-events form a tree in *Unf*, i.e., in each configuration the i-events are totally ordered. Thus, the i-view of C is the local configuration of the unique, causally maximal i-event in C. Intuitively, $\downarrow^i C$ can be understood as the most recent i-local configuration that the whole system is aware of in the (global) configuration C. The i-view of a local configuration $\downarrow e$ is written as $\downarrow^i e$. Note that $\downarrow^i e = \downarrow e$ iff $i \in loc(e)$. We will interpret the empty configuration as the local configuration of a virtual event $\bot$, which can be seen as *initial* event with empty preset and $Min(N')$ as postset. We assume the set of events of Unf_Σ to contain this virtual event, $\bot \in E$, and set $loc(\bot) := Loc$.

Let $\mathcal{C}_{loc}(Unf)$ denote the set of local configurations of *Unf* (abbreviated $\mathcal{C}_{loc}$ if *Unf* is clear), and by $\mathcal{C}^i_{loc} := \{\downarrow e \mid e \in E_i\}$ the set of i-local configurations.

Correspondence of CSAs and unfoldings. Since in [LRT92], the entire formalism relies on communicating sequential agents (CSAs), we will show that a *rooted* CSA is equivalent to the unfolding of a distributed net system.

A CSA is a structure $(E', \leq')$ such that

1. $\leq'$ is a partial order,
2. $E' := \bigcup_{i \in Loc} E'_i$ is the union of the sets[2] $\{E'_i\}_{i \in Loc}$,
3. for all $i \in Loc$ and all $e \in E'$ it holds that $\downarrow e \cap E'_i$ is totally ordered by $\leq'$, where $\downarrow e := \{e' \in E' \mid e' \leq' e\}$.

Moreover, $\leq' = (\bigcup_{i \in Loc} (\leq' |_{(E'_i \times E'_i)})^*$, i.e., $\leq'$ is generated by the suborders on the local sets of events E'_i.

The set $\downarrow e$ is the local state of e. Although the conflict relation $\# \subseteq E' \times E'$ is not represented explicitely in CSAs, it can be obtained as follows: if two events $e_1, e_2 \in E'_i$ are not ordered by $\leq'$, they are considered to be in conflict. Conflicts are inherited to causal successors, i.e., if $e_1 \# e_2$, and $e_2 \leq e_3$, then also $e_1 \# e_3$.

A CSA is called *finitary* if $\downarrow e$ is a finite set for all $e \in E'$. A CSA is called *rooted* if there is a least (w.r.t. $\leq'$) element $\bot \in E'$. It is easy to see that, given the unfolding (N', π) of a distributed net system, the structure $(E, \leq)$, where $\leq$ is the reflexive transitive closure of the flow relation of N', and E the set of events of N', is a rooted, finitary CSA.

[2] *Asynchronous CSAs (ACSAs)* require the sets $\{E_i\}_i$ to be pairwise disjoint. In the current setting, the difference is merely technical, and will not be considered further.

3 Temporal Logic for Communicating Sequential Agents

Lodaya, Ramanujam, and Thiagarajan defined and axiomatised the temporal logic $\mathcal{L}_{CSA}$ that allows to express properties referring to the *local knowledge* or, more precisely, the *latest gossip* of the agents in a distributed system. Let us give a brief idea of the logic, related to unfoldings of distributed net systems. For details, cf. [LRT92].

$\mathcal{L}_{CSA}$ is based on propositional logic. Additionally, it provides two temporal operators $\Diamond_i$ and $\ominus_i$ for each $i \in Loc$, referring to the *local future*, resp. *local past*, of agent i. All formulae are interpreted exclusively on the *local* configurations of a given unfolding.

Intuitively, $\ominus_i\varphi$ holds at $\downarrow e$ if some i-local configuration in the past of e satisfies φ. If e is a j-local event, this can be read as "agent j has at its current local state $\downarrow e$ sufficient gossip information to assert that φ was true in the past in agent i".

The local configuration $\downarrow e$ satisfies $\Diamond_i \varphi$ iff some i-local configuration in the i-local future of $\downarrow e$ satisfies φ, i.e., if there is some configuration $\downarrow e'$ with $e' \in E_i$ such that $\downarrow e' \supseteq \downarrow^i e$ and $\downarrow e'$ satisfies φ. For $e \in E_j$, this can be read as "at the j-local state where e has occurred, agent j has sufficient gossip information about agent i to assert that φ can hold eventually in i".

Typical specifications are properties like $\Diamond_i(x_i \rightarrow \bigwedge_{j\in Loc} \Diamond_j\, x_j)$: "whenever x_i holds in i, then agent i knows that x_j may hold eventually in all other agents j". For more examples see [LRT92]. The formal syntax and semantics of $\mathcal{L}_{CSA}$ is given in the appendix.

A generalised syntax – $\mathcal{L}$. We now introduce a slightly extended language in which the temporal modalities are separated from the gossip modalities. The separation yields a higher degree of modularity in the technical treatment and also saves redundant indices in nested formulae residing at a single location. The abstract syntax of $\mathcal{L}$ is

$$\varphi \;::=\; p \;\mid\; \neg\varphi \;\mid\; \varphi \vee \varphi \;\mid\; \Diamond\,\varphi \;\mid\; \ominus\,\varphi \;\mid\; @\,i\!:\varphi$$

where p ranges over AP and i over Loc. Additionally, we require that every occurrence of a temporal modality lies within the scope of a gossip modality. For technical simplicity, we set $AP := P$ the set of places of our systems[3]. The operators $\Diamond$ and $\ominus$ are now seen as temporal future and past modalities within a single location, which is determined by the next enclosing gossip modality $@\,i\!:$. The connection to $\mathcal{L}_{CSA}$ is established by $\Diamond_i\,\varphi \equiv @\,i\!: \Diamond\varphi$ and $\ominus_i\,\varphi \equiv @\,i\!: \ominus\,\varphi$.

$\mathcal{L}$-Formulae are interpreted at local configurations only. The models of $\mathcal{L}$ are pairs $M = (\mathit{Unf}, V)$, where Unf is the unfolding of a distributed net system, and $V : \mathcal{C}_{loc}(\mathit{Unf}) \longrightarrow \mathbf{2}^{AP}$ is a *valuation* mapping the local configurations of Unf onto subsets of AP, coinciding with the state function $\mathcal{M}(\downarrow e)$.

Formally, we define two satisfaction relations: a global relation $\models$, defined for the local configurations of *arbitrary* locations, and for each agent $i \in Loc$ a local

[3] Note that we do not loose expressive power by this convention.

relation $\models_i$, exclusively defined for the *i-local* configurations. These relations are inductively defined as follows:

$$
\begin{array}{llll}
{\downarrow}e \models p & \text{iff } p \in \mathcal{M}({\downarrow}e) & {\downarrow}e \models \varphi \vee \psi & \text{iff } {\downarrow}e \models \varphi \quad \text{or} \quad {\downarrow}e \models \psi \\
{\downarrow}e \models \neg\varphi & \text{iff } {\downarrow}e \not\models \varphi & {\downarrow}e \models @\, i : \varphi & \text{iff } {\downarrow}^i e \models_i \varphi
\end{array}
$$

$$
\begin{array}{llll}
{\downarrow}e \models_i p & \text{iff } p \in \mathcal{M}({\downarrow}e) & {\downarrow}e \models_i \varphi \vee \psi & \text{iff } {\downarrow}e \models_i \varphi \quad \text{or} \quad {\downarrow}e \models_i \psi \\
{\downarrow}e \models_i \neg\varphi & \text{iff } {\downarrow}e \not\models_i \varphi & {\downarrow}e \models_i \ominus\varphi & \text{iff } \exists e' \in E_i \,.\, e' \leq e \text{ and } {\downarrow}e' \models_i \varphi \\
{\downarrow}e \models_i @\, j : \varphi & \text{iff } {\downarrow}^j e \models_j \varphi & {\downarrow}e \models_i \Diamond\varphi & \text{iff } \exists e' \in E_i \,.\, e' \geq e \text{ and } {\downarrow}e' \models_i \varphi
\end{array}
$$

We say that the system Σ satisfies a formula φ if the empty configuration ${\downarrow}\bot$ of Unf_Σ satisfies φ, i.e., if ${\downarrow}\bot \models \varphi$.

The *future fragment* $\mathcal{L}^+$ of $\mathcal{L}$ consists of all formulae that do not contain the past-operator $\ominus$.

4 Factorisation of the Unfolding

In general, the unfolding of a net system is infinite, even if the net is finite-state. Therefore, most model checking algorithms cannot directly be applied on a modal logic defined over the unfolding. A way to overcome this problem is to look for a factorisation of the unfolding by a decidable equivalence relation $\equiv$ that is finer than the distinguishing power of the formula to be evaluated, i.e., $C \equiv C'$ shall imply $C \models \varphi \Leftrightarrow C' \models \varphi$. The second requirement on $\equiv$ is that a set of representatives of its finitely many equivalence classes and a representation of the (transition) relations between the classes can be computed effectively. Then we can decide $C \models \varphi$ on *Unf* by transferring the question to the model checking problem $(C/\equiv) \models \varphi$ on $(\mathit{Unf}/\equiv, \longrightarrow)$.

The finite prefix. The first construction of an appropriate finite factorisation was given by McMillan [McM92]. He showed how to construct a finite prefix of the unfolding of a finite-state net system in which every reachable marking is represented by some cut. In terms of temporal logic, his approach means to consider formulae of the type $\Diamond\,\psi$ where $\Diamond$ is "global reachability" and ψ is a boolean combination of atomic propositions P. The key to the construction is that if the prefix contains two events with $\mathcal{M}$-equivalent local configurations, then their futures are isomorphic, i.e., they cannot be distinguished by the logic. Consequently, only one of them needs to be explored further, while the other one becomes a *cutoff* event. The *finite prefix Fin* is that initial part of the unfolding, where the causal successors of each cutoff are discarded, i.e., an event e' belongs to *Fin* iff no event $e < e'$ is a cutoff.

In general, the formal definition of a cutoff requires two crucial relations on configurations: An instance of the equivalence relation $\equiv$ and a *partial order* $\prec$. On the one hand, an adequate partial order shall ensure that the expanded prefix contains a representative for each equivalence class. On the other hand, it shall guarantee that the prefix remains finite. The conditions for an *adequate* partial order $\prec$ in conjunction with $\mathcal{M}$-equivalence were examined very detailed in [ERV96]: Besides being well-founded, and respecting set inclusion ($C \subset C'$ implies $C \prec C'$), it must be preserved under *finite extensions*: if $C \prec C'$ and $C \equiv C'$ then $C \oplus E' \prec C' \oplus I_C^{C'}(E')$.

An adequate partial order given in [McM92] is the *size* of configurations, i.e., $C \prec C'$ iff $|C| < |C'|$. With this order, the prefix is often much smaller than the global state space of a given system. However, sometimes it is larger, namely if it is often the case that two equivalent local configurations $\downarrow e \equiv \downarrow e'$ are not ordered by $\prec$, and such neither e nor e' can be distinguished as a cutoff. In [ERV96], an elaborate *total* order for 1-safe nets was defined, such that the constructed prefix is *minimal*, i.e., never exceeds the global state space.

In [McM92,ERV96] just $\mathcal{M}$-equivalence is considered. In conjunction with an adequate order $\prec$ the definition of *Fin* guarantees that each reachable marking is represented by the state of a configuration contained *Fin*.

It was already observed in [HNW98] that refining $\mathcal{M}$-equivalence yields an *extended prefix*, which – although being possibly larger than the prefix of [McM92] and [ERV96] – allows to apply a standard μ-calculus model checker for a location based modal logic called the *distributed μ-calculus*. Following the idea from the beginning of the section, we defined an equivalence $\equiv_{\mathcal{M}-loc}$ by $\downarrow e \equiv_{\mathcal{M}-loc} \downarrow e'$ iff $\downarrow e \equiv_{\mathcal{M}} \downarrow e'$ and $loc(e) = loc(e')$ and proved that $\equiv_{\mathcal{M}-loc}$-equivalence equals the distinguishing power of the distributed μ-calculus.

Generalised cutoffs. Now we look for more general conditions on equivalence relations that ensure that all equivalence classes can be computed by a prefix construction. Let us call a decidable equivalence relation $\equiv$ on configurations of *Unf* to be *adequate* if it refines $\mathcal{M}$-equivalence and has finite index. I.e., $C \equiv C'$ implies $C \equiv_{\mathcal{M}} C'$ and $\equiv$ has only finitely many equivalence classes on *Unf*. We give a generalised definition of *cutoffs* by

$e \in E$ is called a *cutoff* iff $\exists e' \in E$, such that $\downarrow e' \equiv \downarrow e$ and $\downarrow e' \prec \downarrow e$

where $\equiv$ is an adequate equivalence relation and $\prec$ is an adequate partial order. The *finite prefix Fin* constructed for $\equiv$ is given by the condition: e' belongs to *Fin* iff no event $e < e'$ is a cutoff. It is obvious from the cutoff definition that *Fin* constructed for $\equiv$ contains a representative for each $\equiv$-class of *Unf*.

Proposition 1. *The finite prefix constructed for an adequate $\equiv$ is finite.*

The proof is not very difficult and can be found in the appendix.

An adequate equivalence finer than $\mathcal{L}$. In difference to S4 as used in [Esp94] and the distributed μ-calculus in [HNW98], an equivalence finer than the distinguishing power of $\mathcal{L}$ has infinite index. However, by each finite set of $\mathcal{L}$-formulae we can only discriminate finitely many classes of configurations. Thus we can hope for a model checking procedure following the outline from the beginning of the section, if we find an equivalence which is at least as discriminating as the Fisher-Ladner-closure of a $\mathcal{L}$-formula φ because this is the set of formulae relevant for model checking φ on *Unf.* First, we need some technical definitions.

Let us denote the *gossip-past-depth* of a given formula $\varphi \in \mathcal{L}$ by $gpd(\varphi)$. It shall count how often in the evaluation of φ we have to change the local view – with the gossip modality or by referring to a proposition, which also changes the view when the proposition belongs to another location. The inductive definition is as follows:

$$\begin{array}{ll} gpd(p) = 1 & gpd(\neg\varphi) = gpd(\varphi) \\ gpd(\varphi \vee \psi) = \max\{gpd(\varphi), gpd(\psi)\} & gpd(\Diamond\varphi) = gpd(\varphi) \\ gpd(@\, i\colon \varphi) = gpd(\varphi) + 1 & gpd(\diamondminus\, \varphi) = gpd(\varphi) + 1 \end{array}$$

Now we are ready to define the crucial equivalence relation $\equiv_i^n$, which is the basis for model checking $\mathcal{L}$. It is parameterised by a natural number n (which will be the gossip-past-depth of a given formula) and by a location i (at which the formula is interpreted). Formally, we define $\equiv_i^n \subseteq \mathcal{C}_{loc}^i \times \mathcal{C}_{loc}^i$ to be the coarsest equivalence relation satisfying:

$$\begin{array}{l} {\downarrow}e \equiv_i^0 {\downarrow}f \ \text{ implies } \ \forall p \in P_i \,.\ p \in \mathcal{M}({\downarrow}e) \Leftrightarrow p \in \mathcal{M}({\downarrow}f) \\ {\downarrow}e \equiv_i^1 {\downarrow}f \ \text{ implies } \ \forall j,k \in Loc \,.\, {\downarrow^j}e \subseteq {\downarrow^k}e \Leftrightarrow {\downarrow^j}f \subseteq {\downarrow^k}f \end{array}$$

and for all $n \geq 0$ moreover

$$\begin{array}{ll} {\downarrow}e \equiv_i^{n+1} {\downarrow}f \ \text{ implies } & \forall j \in Loc \,.\, {\downarrow^j}e \equiv_j^n {\downarrow^j}f \ \text{ for } n \geq 1 \\ \qquad\qquad (*) \textit{ and } & \forall e' \in ({\downarrow}e \cap E_i) \,.\, \exists f' \in ({\downarrow}f \cap E_i) \,.\, {\downarrow}e' \equiv_i^n {\downarrow}f' \quad \textit{and} \\ & \forall f' \in ({\downarrow}f \cap E_i) \,.\, \exists e' \in ({\downarrow}e \cap E_i) \,.\, {\downarrow}e' \equiv_i^n {\downarrow}f' \end{array}$$

The first condition is an i-localised version of $\mathcal{M}$-equivalence. The second one refers to the *latest information* concerning agents other than i, and the third condition inductively lifts the equivalence with respect to the levels of the gossip-past-depth. Let us briefly collect some important facts about the equivalence.

Observation 2. *The equivalence relation $\equiv_i^n$ is decidable and of finite index for every $n \geq 0$. Furtheron, $\equiv_i^{n+1}$ is refining $\equiv_i^n$, i.e., $\equiv_i^{n+1} \subseteq \equiv_i^n$ for all n. Finally, it respects $\mathcal{M}$-equivalence, i.e., ${\downarrow}e \equiv_i^n {\downarrow}f$ implies $\mathcal{M}({\downarrow}e) = \mathcal{M}({\downarrow}f)$ for all $n > 0$.*

The proof is given in the appendix.

Remark 3. Note that the last part of the third condition after (*) is only needed for the full logic $\mathcal{L}$ and can be omitted for $\mathcal{L}^+$ with considerable savings: With this condition, the number of equivalence classes of $\equiv_i^n$ may grow non-elementarily with n, forbidding any consideration of practicability, whereas without this condition the index grows exponentially with n.

The most important property of the equivalence is that it is preserved by local successors, as stated in the following Lemma.

Lemma 4. *Let $e \leq e'$, and $f \leq f'$ be i-events, such that ${\downarrow}e \equiv_i^n {\downarrow}f$, and let I be the isomorphism from $\beta({\downarrow}e)$ onto $\beta({\downarrow}f)$. If $f' = I(e')$ then also ${\downarrow}f' \equiv_i^n {\downarrow}e'$.*

Proof. This the most involved proof, and a main result of the paper. Let us define some notions and notations:

To handle the past modality we extend the i-view operator by a natural number which specifies the number of steps we intend to additionally go backward into the past of agent i. Let $i \in Loc$, $n \in \mathbb{N}$ and C a configuration. Then $\downarrow_n^i C$ is defined to be the least subconfiguration of C satisfying $|(\downarrow^i C \setminus \downarrow_n^i C) \cap E_i| = n$ if $|\downarrow^i C \cap E_i| \geq n$ and $\downarrow\perp$ otherwise. I.e., $\downarrow_n^i C$ removes the top n i-events from C and then takes the i-view. Note that $\downarrow_0^i C = \downarrow^i C$.

Since we will often talk about a number of view changes and past steps in sequence we introduce "paths" through the locations of the system: Let $\tau = (\sigma, i)$ be a pair where $\sigma = l_1 l_2 \ldots l_n$ is a sequence of locations and natural numbers, i.e., $l_k \in (Loc \uplus \mathbb{N})$ and $i \in Loc$. If σ is the empty sequence we denote this by ε. We call τ a *location path.* Given any configuration C, we define $\downarrow^{(\sigma,i)} C$ as follows:

$$\downarrow^{(\varepsilon,i)} C := \downarrow^i C \qquad \downarrow^{(\sigma n,i)} C := \downarrow^{(\sigma,i)} \downarrow_n^i C \qquad \downarrow^{(\sigma j,i)} C := \downarrow^{(\sigma,j)} \downarrow^i C$$

The length of a location path $\tau = (\sigma, i)$ is the length of σ. Note that a sequence σ may include repetitions of locations and subsequences of natural numbers, i.e., $l_i = l_j$ for $i \neq j$ and $l_n, \ldots, l_{n+m} \in \mathbb{N}$ are allowed.
Given an event g and some location path τ, we denote by g_τ the event that determines the τ-view of $\downarrow g$, i.e., $\downarrow^\tau g = \downarrow g_\tau$.
Now let $e \leq e'$ and $f \leq f'$ be events of E_i, and $n \geq 1$, as in the assumptions of the Lemma. First of all, we note that the required isomorphism I exists because $\equiv_i^n$-equivalence implies $\mathcal{M}$-equivalence.
We have to show $\downarrow f' \equiv_i^n \downarrow e'$.

The key observation is that for every location path τ if $e'_\tau \not\leq e$ then $I(e'_\tau) = f'_\tau \not\leq f$. This is the basis for the induction on $m \leq n$: for each sequence $\tau = (\sigma, i)$ of length $n - m$ with $e'_\tau \not\leq e$ (and also $f'_\tau \not\leq f$), it holds that $\downarrow e'_\tau \equiv_j^m \downarrow f'_\tau$, where j is either the first location occurring in the sequence σ, or $j := i$ (if $n = m$ and $\sigma = \varepsilon$ is the only sequence of length $n - m$). In the latter case, $\downarrow^i e' = \downarrow e'$ (because $e' \in E_i$), and $\downarrow^i f' = \downarrow f'$, we thus obtain $\downarrow e' \equiv_i^n \downarrow f'$ as required.
The induction relies on a case analysis according to the following cases:
$m = 0$, $n = m = 1$, $n = m > 1$, $n > m = 1$, and finally $n > m > 1$.

- For $m = 0$ we have to show that $\downarrow e'_\tau \equiv_j^0 \downarrow f'_\tau$. This is clear, because $I(e'_\tau) = f'_\tau \in E_j$ and thus the j-local part of the markings of $\downarrow e'_\tau$ and $\downarrow f'_\tau$ coincide, because $\pi(e'_\tau)^\bullet = \pi(f'_\tau)^\bullet$.
- For $m = n = 1$ we have to show that $\downarrow e \equiv_i^1 \downarrow f$ implies $\downarrow e' \equiv_i^1 \downarrow f'$, i.e.,
  1. $\downarrow^j e' \equiv_j^0 \downarrow^j f'$ for all $j \in Loc$,
  2. $\forall e'_p \in (\downarrow e' \cap E_i) \,.\, \exists f'_p \in (\downarrow f' \cap E_i) \,.\, \downarrow e'_p \equiv_i^1 \downarrow f'_p$ (and vice versa), and
  3. $e'_j \leq e'_k$ iff $f'_j \leq f'_k$ for all $j, k \in Loc$.

  If $e'_j \leq e$ then $\downarrow e' \setminus \downarrow e$ contains no j-event, which means that $e'_j = e_j$ and similarly $f'_j = f_j$, so (1) follows easily. If $e'_j \not\leq e$ then also $f'_j \not\leq f$, in which case $\downarrow^j e' \equiv_j^0 \downarrow^j f'$ follows by induction.
  We come to (2): Let $e'_p \in (\downarrow e \cap E_i)$. If $e'_p \leq e$ then by the assumtion we find an $f'_p \in (\downarrow f \cap E_i)$ with $\downarrow e'_p \equiv_i^0 \downarrow f'_p$ (remember that $e, e', f, f' \in E_i$). If $e'_p \not\leq e$

then $f'_p := I(e'_p) \equiv_i^0 e'_p$ and $f'_p \in (E_i \cap \downarrow f')$.
Now consider (3): Let $j, k \in Loc$. We show that $e'_j \leq e'_k$ iff $f'_j \leq f'_k$, using a similar case analysis. If $e'_j, e'_k \not\leq e$, then the isomorphism I preserves the order. If $e'_j, e'_k \leq e$, then $e'_j = e_j$ and $e'_k = e_k$, (and similarly $f'_j = f_j$, $f'_k = f_k$), and so the order is inherited from the corresponding local views of $\downarrow e$ and $\downarrow f$, which by assumption match. The third case is $e'_j \leq e$, but $e'_k \not\leq e$, and thus similarly $f'_j \leq f$, but $f'_k \not\leq f$. Since this is one of the sophisticated arguments and used also in the other cases the situation is illustrated in Figure 3. $e'_j \leq e$ implies $e'_j = e_j$.

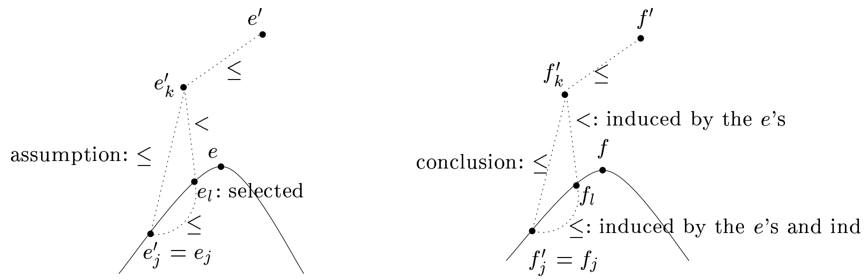

Fig. 3. Situation: $e'_k \not\leq e$ and $e'_j \leq e$

Now we choose an $l \in Loc$, such that $e_j \leq e_l \leq e'_k$, and moreover e_l is (causally) maximal with this respect. For at least one of the possible choices of l, there exists an event $e'' \in E_l$, such that $e'' \in (\downarrow^k e' \setminus \downarrow e)$. By the isomorphism, we have that $I(e'') = f'' \in (\downarrow^k f' \setminus \downarrow f)$. By assumption on the equivalence of e and f we can conclude $f'_j = f_j \leq f_l < f'' \leq f'_l \leq f'_k$, i.e., $\downarrow^j f' \subseteq \downarrow^k f'$ as desired.

- For $n = m > 1$ the reasoning is similar to the case $m = n = 1$, except that the argument for the gossip aspect (3) of the equivalence is not needed.
- For $n > m = 1$, we consider (1) a location path $\tau = (k\sigma, i)$ of length $n-1$ with $e'_\tau \not\leq e$ and $k \in Loc$. Again we have to show $\downarrow e'_\tau \equiv_k^1 \downarrow f'_\tau$.
  First consider $j \in Loc$. For the case of $e'_{(jk\sigma,i)} \not\leq e$ the $\equiv_j^0$-equivalence is a consequence of $I(e'_{(jk\sigma,i)}) = f'_{(jk\sigma,i)}$. For $e'_{(jk\sigma,i)} \leq e$ there exists again an $l \in Loc$ with $e'_{(jk\sigma,i)} \leq e_l \leq e'_{(k\sigma,i)}$, so that e_l is maximal in this respect, and as above we also obtain $f'_{(jk\sigma,i)} \leq f_l \leq f'_{(k\sigma,i)}$. Moreover, in this case it holds that $e_{(j,l)} = e'_{(jk\sigma,i)}$ and similarly $f_{(j,l)} = f'_{(jk\sigma,i)}$. By assumption, we have $\downarrow e_{(j,l)} \equiv_j^{n-2} \downarrow f_{(j,l)}$, and because of $n \geq 2$, in particular $\downarrow e_{(j,l)} \equiv_j^0 \downarrow f_{(j,l)}$, as desired.
  Additionally, we have to show that $\forall e'_p \in (\downarrow e'_\tau \cap E_k) \,.\, \exists f' \in (\downarrow f'_\tau \cap E_k) \,.\, \downarrow e'_p \equiv_k^0 \downarrow f'_p$ (and vice versa).
  If $e'_p \not\leq e$ then we select $f'_p := I(e'_p)$ and argue in the same lines as for $m = n = 1$. If $e'_p \leq e$ then we know that $e'_p \leq \downarrow^k e$. Since $n \geq 2$ we know that for e'_p there exists an $f'_p \leq \downarrow^k f \leq f$ such that $e'_p \equiv_k^{n-2} f'_p$. Together with the

second part of the obervation we conclude $e'_p \equiv^0_k f'_p$.
The argument concerning the relative orders of j-views and k-views of e_σ and e'_σ is the same as for the case of $m = n = 1$.
Now we consider a location path $\tau = (d\sigma, i)$ of length $n-1$ with $e'_\tau \not\leq e$ and $d \in \mathbb{N}$, i.e., the sequence starts with a natural number. Let k be the first location occurring in σ and $k := i$ if σ does not contain a location. We have to show that $\downarrow e'_\tau \equiv^1_k \downarrow f'_\tau$ and the proof is exactly as in the case where the initial element of the sequence is a location.

– For $1 < m < n$ let τ be of length $n - m$, such that $\tau = (k\sigma, i)$ has $k \in \mathit{Loc}$ as first element, and such that $e'_\tau \not\leq e$, and, similarly, $f'_\tau \not\leq f$. To show that $\downarrow e'_\tau \equiv^m_k \downarrow f'_\tau$ we prove (1) for each $j \in \mathit{Loc}$ it holds that $\downarrow^j e'_\tau \equiv^{m-1}_j \downarrow^j f'_\tau$. For $e'_{(jk\sigma,i)} \not\leq e$ and similarly $f'_{(jk\sigma,i)} \not\leq f$ this follows from the induction hypothesis. For $e'_{(jk\sigma,i)} \leq e$ there exists (again) a location l, such that $e'_{(jk\sigma,i)} \leq e_l \leq e'_\sigma$ and e_l is causally maximal in this respect. Then $\downarrow^j e'_\tau = \downarrow^j e_l \equiv^{n-2}_j \downarrow^j f_l = \downarrow^j f'_\tau$, where $n - 2 \geq m - 1$, so that the desired claim follows from the observation $\equiv^{\tilde{n}+\tilde{m}}_j \subseteq \equiv^{\tilde{n}}_j$. The second part is to prove (2) for each $e'_p \in (\downarrow e'_\tau \cap E_k)$ that there exists an $f'_p \in (\downarrow f'_\tau \cap E_k)$ such that $e'_p \equiv^{m-1}_k f'_p$. If $e'_p \leq e$ then an appropriate $f'_p \leq f \leq f'_\tau$ exists by assumption. Now let $e'_p \not\leq e$. Then there exists a $d \in \mathbb{N}$ such that $e'_p = \downarrow^k_d e'_\tau = \downarrow^{(dk\sigma,i)} e'$. Using the isomorphism I we get $f'_p := I(e'_p) = \downarrow^k_d f'_\tau = \downarrow^{(dk\sigma,i)} f'$. The length of $\tau' = (dk\sigma, i)$ is $n - (m-1)$. Thus by induction we know that $e'_p \equiv^{m-1}_k f'_p$. Now let $\tau = (d\sigma, i)$ where $d \in \mathbb{N}$ such that $e'_\tau \not\leq e$ and $f'_\tau \not\leq f$. Let k be the first location occurring in σ and if σ does not contain a location we set $k := i$. The other arguments correspond to the case above.

Using this Lemma, we can easily show our main result, namely that $\equiv^n_i$ is more discriminating than $\mathcal{L}$-formulae with a gossip depth smaller than n:

Theorem 5. *Let φ be an $\mathcal{L}$-formula of gossip-past-depth n, and let $e, f \in E_i$ with $\downarrow e \equiv^n_i \downarrow f$. Then $\downarrow e \models_i \varphi$ iff $\downarrow f \models_i \varphi$.*

Proof. By structural induction on φ: For atomic propositions, note that Observation 2 $\downarrow e \equiv^1_i \downarrow f$ implies $\downarrow e =_{\mathcal{M}} \downarrow f$ and hence $\downarrow e \models_i p$ iff $\downarrow f \models_i p$. The induction for boolean connectives is obvious. For $gpd(\Diamond\varphi) = gpd(\varphi) = n$ let $\downarrow e \models_i \Diamond\varphi$ and $\downarrow e \equiv^n_i \downarrow f$. We have to show that also $\downarrow f \models_i \Diamond\varphi$ (all other cases follow by symmetry).

By definition, there exists $e' \geq e$ with $e' \in E_i$ and $\downarrow e' \models_i \varphi$. By Lemma 4 the event $f' = I(e') \in E_i$ obtained from the isomorphism I due to the $\mathcal{M}$-equivalence of $\downarrow e$ and $\downarrow f$ satisfies $f \leq f'$ and $\downarrow e' \equiv^n_i \downarrow f'$. By induction, $\downarrow f' \models_i \varphi$ and finally $\downarrow f \models_i \Diamond\varphi$.

Now let $\varphi = @j : \psi$ with $gpd(\varphi) = gpd(\psi) + 1 = n$. $\downarrow e \models_i \varphi$ implies $\downarrow^j e \models_j \psi$ and by definition $\downarrow^j e \equiv^{n-1}_j \downarrow^j f$. Thus, by induction $\downarrow^j f \models_j \psi$ and finally $\downarrow f \models_i \varphi$.

The argument for formulae $\varphi = \diamondminus\psi$ is very similar to the case of $\varphi = @j : \psi$ and makes use of the last part of the third condition in the definition of $\equiv^n_i$. This is why this condition can be omitted for the subclass of $\mathcal{L}^+$ formulae.

Based on the local equivalences, we define an adequate equivalence relation for the construction of a finite prefix by $\downarrow e \equiv^n \downarrow f$ iff $loc(e) = loc(f)$ and $\downarrow e \equiv_i^n \downarrow f$ for all $i \in loc(e)$. The next and last step to transfer the $\mathcal{L}$ model checking problem from the unfolding to an equivalent model checking problem over a finite structure is the definition of the transitions between the $\equiv^n$-equivalence classes of *Unf*. This is done in the next section.

5 Model checking

In this section we propose a verification technique for $\mathcal{L}$. Following the lines of [HNW98], we will sketch a reduction of a given instance of the problem to a suitable input for well investigated model checkers like e.g. [CES86].

Let us consider a distributed net system Σ and an $\mathcal{L}$-formula φ of gossip-past-depth n. We have shown so far how to construct a finite prefix *Fin* of the unfolding Unf_Σ that contains representatives for all $\equiv_i^n$ equivalence classes. Now we want to compute a finite, multi-modal Kripke structure on the representatives that is equivalent to Unf_Σ with respect to the evaluation of φ. What is missing are the transitions between the representatives.

Computing a finite Kripke structure. Let $n \in \mathbb{N}$, and $Unf_\Sigma = (N', \pi)$ with $N' = (B, E, F)$ be fixed, and let $\equiv^n$ be the equivalence relation used for the construction of *Fin*. The state space $\mathcal{S}_n$ of the desired Kripke structure consists of one representative of each $\equiv^n$ equivalence class. Note that by using the adequate *total* partial order $\prec$ of [ERV96], these representatives are unique, and so the state space is given by $\mathcal{S}_n := \{\downarrow e \mid e \in Fin \text{ and } e \text{ is not a cutoff}\}$. If the used order $\prec$ is not total, we fix one non-cutoff (resp. its local configuration) of the prefix as the representative of each $\equiv^n$ equivalence class. For every local configuration $\downarrow e$ of Unf_Σ, let $rep(\downarrow e) \in \mathcal{S}_n$ denote the unique representative.

Now let us consider the transitions of the Kripke structure. We introduce a transition relation for each of the modalities of the logic. Let $\downarrow e, \downarrow f \in \mathcal{S}_n$:

$$\begin{aligned} \downarrow e \xrightarrow{\Diamond i}_n \downarrow f &\text{ iff } e, f \in E_i \text{ and } \exists f' \in E_i \,.\, f' \geq e \,\wedge\, rep(\downarrow f') = \downarrow f \\ \downarrow e \xrightarrow{@j} \downarrow f &\text{ iff } e \in E_i,\, f \in E_j \,\wedge\, \downarrow^j e = \downarrow f \\ \downarrow e \xrightarrow{\diamondminus i} \downarrow f &\text{ iff } e, f \in E_i \wedge f \leq e. \end{aligned}$$

Note that the definitions of $\xrightarrow{@j}$ and $\xrightarrow{\diamondminus i}$ rely on the fact that the set of configurations in *Fin* (and thus also in $\mathcal{S}_n$) is downward closed, i.e., the j-view of any element of $\mathcal{S}_n$ is again in $\mathcal{S}_n$ for every j, and of course past configurations as well. On the whole, we obtain the multi-modal Kripke structure $\mathcal{T}_n = (\mathcal{S}_n, \{\xrightarrow{\Diamond i}_n, \xrightarrow{@i}, \xrightarrow{\diamondminus i} \mid i \in Loc\}, \downarrow\perp)$ with root $\downarrow\perp$.

As a corollary to Theorem 5 we obtain the following characterisation of the semantics of $\mathcal{L}$ formulae over $\mathcal{T}_n$:

Corollary 6. *Let $\varphi \in \mathcal{L}$ be a formula of gossip-past-depth $m \leq n$, and let $\downarrow e \in \mathcal{S}_n$ be an i-local configuration, i.e., $e \in E_i$.*

1. *If* $\varphi = \Diamond\psi$ *then* $\downarrow e \models_i \varphi$ *iff* $\exists \downarrow f \in \mathcal{S}_n$ *with* $\downarrow e \xrightarrow{\Diamond i}_n \downarrow f$ *and* $\downarrow f \models_i \psi$.
2. *If* $\varphi = @j:\psi$ *then* $\downarrow e \models_i \varphi$ *iff* $\exists \downarrow f \in \mathcal{S}_n$ *with* $\downarrow e \xrightarrow{@j} \downarrow f$ *and* $\downarrow f \models_j \psi$.
3. *If* $\varphi = \Diamond\!\!\!\cdot\, \psi$ *then* $\downarrow e \models_i \varphi$ *iff* $\exists \downarrow f \in \mathcal{S}_n$ *with* $\downarrow e \xrightarrow{\Diamond\!\!\cdot\, i} \downarrow f$ *and* $\downarrow f \models_i \psi$.

Proof. (1) follows from the definition of the semantics of $\Diamond$ and the fact that by construction of $\mathcal{T}_n$ for any pair of states $\downarrow f'$ and $\downarrow f = rep(\downarrow f')$, we have that $\downarrow f \models_i \varphi$ iff $\downarrow f' \models_i \varphi$ for any formula φ of gossip-past-depth $m \leq n$. (2) and (3) are trivial.

Thus, if we are able to actually compute (the transitions of) $\mathcal{T}_n$ then we can immediately reduce the model checking problem of $\mathcal{L}^+$ to a standard model checking problem over finite transition systems, applying e.g. [CES86].

Computing the transitions $\downarrow e \xrightarrow{@j} \downarrow f$ in $\mathcal{T}_n$ is trivial: $\downarrow f = \downarrow^j e$. Similarly computing the $\xrightarrow{\Diamond\!\!\cdot\, i}$ *successors* of $\downarrow e$ is very easy. It is more difficult to compute the transitions $\downarrow e \xrightarrow{\Diamond i}_n \downarrow f$, if only *Fin* is given. To achieve this, we use a modified version of the algorithm proposed in [HNW98].

An algorithm to compute the $\xrightarrow{\Diamond i}_n$ transitions. We assume in the following, that the algorithm for constructing the prefix *Fin* uses a total, adequate order $\prec$. The construction of *Fin* provides some useful structural information: each cutoff e has a *corresponding* event e^0, such that $\downarrow e^0 \equiv^n \downarrow e$, and $\downarrow e^0 \prec \downarrow e$. Clearly, we choose $rep(\downarrow e) := \downarrow e^0$ for each cutoff e, and for non-cutoffs f, we set $rep(\downarrow f) := \downarrow f$. For technical reasons, we have to use an extended definition of $\xrightarrow{\Diamond i}_n$: we define $C \xrightarrow{\Diamond i}_n \downarrow e$ for any local *or* global configuration $C \subseteq \downarrow e'$, with $rep(\downarrow e') = \downarrow e$ and $e, e' \in E_i$. The construction of *Fin* also provides a function *shift**, which maps any configuration $C = C_1$ of Unf_Σ containing some cutoff, onto a configuration *shift*$^*(C) = C_m$ not containing a cutoff, hence being present in *Fin*. This function works by repeatedly applying $C_{k+1} := \downarrow e_k^0 \oplus I_{\downarrow e_k}^{\downarrow e_k^0}(C_k \setminus \downarrow e_k)$ with $e_k \in C_k$ being a cutoff of Fin, and e_k^0 being its corresponding, equivalent event. This repeating application terminates, because the sequence $C_1, C_2, ..$ decreases in the underlying (well-founded) order $\prec$. Obviously, this function implies the existence of an isomorphism I between $\beta(C)$ and $\beta(shift^*(C))$, which is the composition of the isomorphisms $I_{\downarrow e_i}^{\downarrow e_i^0}$ induced by the chosen cutoff events. Moreover, *shift*$^*(\downarrow e) \prec \downarrow e$ for any $e \in \beta(C)$, and hence for any e for which $C \xrightarrow{\Diamond i}_n \downarrow e$.

The most important part of the algorithm (cf. Fig. 4) is the recursive procedure *successors* which, when called *from the top level* with a pair $(\downarrow e, i)$, returns the $pfeilin$-successors of $\downarrow e$ in the finite structure. More generally, *successors* performs a depth first search through pairs (C, i), where C is an arbitrary, not necessarily local configuration not containing a cutoff and i is a location. It determines the subset of local configurations in $\mathcal{S}_n$ that represent the $\xrightarrow{\Diamond i}_n$ -successors of C. Formally, $\downarrow e \in successors(C, i)$ iff there exists $\downarrow e'$ in *Unf*, which is $\equiv^n$-equivalent to $\downarrow e$, and $C \xrightarrow{\Diamond i}_n \downarrow e'$.

Proposition 7. *Compute_Multi_Modal_Kripke_Structure computes the* $\xrightarrow{\Diamond i}_n$ *-,* $\xrightarrow{\Diamond\!\!\cdot\, i}$ *-, and* $\xrightarrow{@j}$ *-transitions.*

```
type Vertex = {C: Configuration; i: Location; pathmark: bool; (* for dfs *) }
prefix_successors(C, i) = {rep(↓e) | ↓e ∈ S_n ∧ C --◇i-->_n ↓e}
compatible_cutoffs(C) = {e | e is cut-off and ↓e ∪ C is a configuration in Fin}

proc successors(C, i): ConfigurationSet;
{
    var result: ConfigurationSet;          (* result accumulator for current vertex *)

    Vertex v := findvertex(C,i);           (* lookup in hash table, if not found then *)
                                           (* create new vertex with pathmark = false *)
    if  v.pathmark then return ∅; fi       (* we have closed a cycle *)
    result := prefix_successors(C, i);     (* directly accessible successors *)
    v.pathmark:=true;                      (* put vertex on path *)
    for e_c ∈ compatible_cutoffs(C) do     (* find successors outside Fin behind e_c *)
        result := result ∪ successors(shift*(C ∪ ↓e_c), i);
    od ;
    v.pathmark:=false;                     (* take vertex from path *)
    return result;
}

proc Compute_Multi_Modal_Kripke_Structure;
{
    InitializeTransitionSystem(T_n, Fin);  (* extract state space from Fin *)
    for ↓e ∈ S_n, i ∈ Loc do
        add transition ↓e --@i--> ↓^i e;
    for i ∈ Loc, ↓e, ↓f ∈ S_n ∩ C^i_loc, ↓f ⊆ ↓e do
        add transition ↓e --◇i--> ↓f;
        for ↓e' ∈ successors(↓e,i) do
            add transition ↓e --◇i-->_n ↓e';
        od
    od
}
```

Fig. 4. The conceptual algorithm to compute the transitions of $\mathcal{T}_n$.

The proof is exactly along the lines of a proof for a similar algorithm for the distributed μ-calculus given in [HNW98] and given in the appendix. Note that at top level, *successors* is always called with a *local* configuration $\downarrow e$ as parameter, but the extension of $\downarrow e$ with cutoffs requires that we can also handle global configurations. In this paper, we focus on decidability but not on efficiency. For comments on efficiency of related model checking procedures for the distributed μ-calculus we refer the reader to [HNW98].

6 Conclusion

We have shown the decidability of the model checking problem for $\mathcal{L}$, a location based branching-time temporal logic including temporal and gossip modalities.

The method is based on a translation of the modalities over net unfoldings (or prime event structures) into transitions of a sequential transition system, for which established model checkers for sequential logics can be applied.

While the method as presented is non elementary for the full logic $\mathcal{L}$, the restriction to the future fragment $\mathcal{L}^+$ still allows to express interesting properties and results in a more moderately growing complexity.

We also hope that the presented results can be used as a methodological approach to model checking temporal logics of *causal knowledge* [Pen98].

The main difficulty, the solution of which is also the major contribution of the paper, was to find an adequate equivalence relation on local states that allowed to construct a finite transition system containing a representative for each class of equivalent local states. If the method really is to be applied, then refinements of the equivalence bring it closer to the logical equivalence and thus leading to a smaller index will be crucial. We believe that the potential for such improvements is high at the price of much less understandable definitions.

For the treatment of past an alternative and potentially more efficient approach in the line of [LS95] – elimination of past modalities in CTL – might come to mind, but the techniques used there can at least not directly be transferred to $\mathcal{L}_{CSA}$ because of the intricate interaction between past and gossip modalities.

References

[CEP95] A. Cheng, J. Esparza, and J. Palsberg, *Complexity results for 1-safe nets*, Theoretical Computer Science (1995), no. 147, 117–136.

[CES86] E.M. Clarke, E.A. Emerson, and A.P. Sistla, *Automatic verification of finite-state concurrent systems using temporal logic specifications*, ACM Transactions on Programming Languages and Systems **8** (1986), no. 2, 244–263.

[Eng91] J. Engelfriet, *Branching processes of Petri nets*, Acta Informatica **28** (1991), 575–591.

[ERV96] J. Esparza, S. Römer, and W. Vogler, *An Improvement of McMillan's Unfolding Algorithm*, Tools and Algorithms for the Construction and Analysis of Systems TACAS '96 (Passau, Germany) (T. Margaria and B. Steffen, eds.), LNCS, vol. 1055, Springer, 1996, pp. 87–106.

[Esp94] J. Esparza, *Model checking using net unfoldings*, Science of Computer Programming **23** (1994), 151–195.

[HNW98] M. Huhn, P. Niebert, and F. Wallner, *Verification based on local states*, Proc. of TACAS '98 (B. Steffen, ed.), LNCS, vol. 1384, Springer, 1998, pp. 36–51.

[LRT92] K. Lodaya, R. Ramanujam, and P.S. Thiagarajan, *Temporal logics for communicating sequential agents: I*, Int. Journal of Foundations of Computer Science **3** (1992), no. 2, 117–159.

[LS95] F. Laroussinie and P. Schnoebelen, *A hierarchy of temporal logics with past*, Theoretical Computer Science **148** (1995), 303–324.

[LT87] K. Lodaya and P.S. Thiagarajan, *A modal logic for a subclass of event structures*, Automata, Languages and Programming (T. Ottmann, ed.), LNCS, vol. 267, Springer, 1987, pp. 290–303.

[McM92] K.L. McMillan, *Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits*, Proc. of the 4th Workshop on Computer Aided Verification (Montreal), 1992, pp. 164–174.

[Nie98] Peter Niebert, *A temporal logic for the specification and verification of distributed behaviour*, PhD thesis, Institut für Informatik, Universität Hildesheim, March 1998.

[NPW80] M. Nielsen, G. Plotkin, and G. Winskel, *Petri nets, event structures and domains*, Theoretical Computer Science **13** (1980), no. 1, 85–108.

[NRT90] M. Nielsen, G. Rozenberg, and P.S. Thiagarajan, *Behavioural notions for elementary net systems*, Distributed Computing **4** (1990), no. 1, 45–57.

[Pen97] W. Penczek, *Model-Checking for a Subclass of Event Structures*, Proc. of TACAS '97, LNCS, vol. 1217, 1997.

[Pen98] W. Penczek, *Temporal logic of causal knowledge*, Proc. of WoLLiC '98, 1998.

[Ram95] R. Ramanujam, *A Local Presentation of Synchronizing Systems*, Structures in Concurrency Theory (J. Desel, ed.), Workshops in Computing, 1995, pp. 264–279.

[Thi94] P.S. Thiagarajan, *A Trace Based Extension of PTL*, Proc. of the 9th IEEE Symposium on Logic in Computer Science, 1994.

[Thi95] P.S. Thiagarajan, *A Trace Consistent Subset of PTL*, Proc. of CONCUR '95 (Philadelphia, P.A., USA) (I. Lee and S.A. Smolka, eds.), LNCS, vol. 962, Springer, 1995, pp. 438–452.

[Wal98] F. Wallner, *Model checking LTL using net unfoldings*, Proc. 10th Conf. on Computer-Aided Verification CAV '98, LNCS, vol. 1427, Springer, 1998.

[Win87] Glynn Winskel, *Event structures*, Advances in Petri Nets, LNCS, vol. 255, 1987.

Appendix

A Formal Syntax and Semantics of $\mathcal{L}_{CSA}$

The abstract syntax of $\mathcal{L}_{CSA}$ is

$$\varphi \ ::= \ p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond_i \varphi \mid \ominus_i \varphi$$

where p ranges over a set AP of atomic propositions, and $i \in Loc$. The dual operators are defined by $\varphi \wedge \psi = \neg(\neg\varphi \vee \neg\psi)$, $\boxminus_i \varphi = \neg \ominus_i \neg\varphi$ and $\Box_i \varphi = \neg\Diamond_i\neg\varphi$.

The models of $\mathcal{L}_{CSA}$ are pairs $M = (Unf, V)$, where Unf is the unfolding of a distributed net system, and V is a valuation function from the local states of Unf onto subsets of AP. Formulae are interpreted on the local states of M, which is denoted by $M, \downarrow e \models \varphi$. An atomic proposition p is interpreted in accordance with V: $M, \downarrow e \models p$ iff $p \in V(\downarrow e)$. The operators of propositional logic are interpreted as usual, and for the temporal operators, we have

$$\begin{array}{l} M, \downarrow e \models \ominus_i \varphi \ \text{ iff } \ \exists e' \in E_i \,.\, e' \leq e \ \text{ and } \ M, \downarrow e' \models \varphi \\ M, \downarrow e \models \Diamond_i \varphi \ \text{ iff } \ \exists e' \in E_i \,.\, \downarrow^i e \subseteq \downarrow e' \ \text{ and } \ M, \downarrow e' \models \varphi \end{array}$$

B Proofs

B.1 Proof of Proposition 1

Let $Unf = (N', \pi)$, with $N' = (B, E, F)$. For a given event e let $\perp = e_1 < e_2 < \cdots < e_k = e$ be a longest causal chain of events. We define $d(e) := k$, the depth of e, and for each $k \geq 1$ a set $E^k \subseteq E$ by $E^k := \{e \mid d(e) \leq k\}$.

Let $e_1 < \cdots < e_{n+1}$ be a causal chain of events in Unf, where n is the index of $\equiv$. Clearly, there must exist two events e_k, e_l, such that $\downarrow e_k \equiv \downarrow e_l$. Without loss of generality, assume $e_k < e_l$, and thus $\downarrow e_k \subset \downarrow e_l$. Because the partial order $\prec$ respects set inclusion, we have also $\downarrow e_k \prec \downarrow e_l$, and so e_l is a cutoff. We conclude that all events of the prefix Fin belong to E^{n+1}.

Now we show by induction that for every $k \geq 1$, the set E^k contains only finitely many events. The only event in E^1 is $\perp$. Assume E^k is a finite set. Due to the conditions of π, every event in E^k has only finitely many causal successor events, thus also E^{k+1} is finite. So, Fin contains only finitely many events.

Since π is a bijection for all ${}^\bullet t$ and $t^\bullet$, and since our original nets are finite, Fin contains finitely many conditions.

B.2 Proof of Observation 2

$\equiv_i^{n+1}$ refines $\equiv_i^n$. Since $e, f \in E_i$, we have $\downarrow^i e = \downarrow e$ and similarly $\downarrow^i f = \downarrow f$, and thus $\downarrow e \equiv_i^{n+1} \downarrow f$ implies $\downarrow e \equiv_i^n \downarrow f$ by definition.

$\equiv_i^n$ implies $\mathcal{M}$-equivalence. Clearly, $\equiv_i^1$ implies $\mathcal{M}$-equivalence: consider $p \in P_j$ and $\downarrow e \equiv_i^1 \downarrow f$. Then $p \in \mathcal{M}(\downarrow e)$ iff $p \in \mathcal{M}(\downarrow^j e)$ iff $p \in \mathcal{M}(\downarrow^j f)$ iff $p \in \mathcal{M}(\downarrow f)$, where the intermediate equivalence follows from $\downarrow^j e \equiv_j^0 \downarrow^j f$. As seen above, $\equiv_i^n$ implies $\equiv_i^1$ for all $n > 0$.

$\equiv_i^n$ is decidable and of finite index. Since the system under consideration has only finitely many markings, the equivalence $\equiv_i^0$ is of finite index for every $i \in Loc$. Also $\equiv_i^1$ is evidently of finite index for each i. Since there only finitely many locations, by induction (and the definition of $\equiv_i^{n+1}$, relying on $\equiv_j^n$ for all locations j) and (in the case including past) the presence of representatives of $\equiv_i^n$ classes in the set of i-predecessor configurations, there are only finitely many equivalence classes w.r.t. $\equiv_i^{n+1}$. In the case including past this may result in an exponentially higher index, in the case without past polynomially bigger with exponent $|Loc|$.

To understand the decidability one has to think of configurations as data structures in an appropriate representation. Then, the definition of $\equiv_i^n$ can almost immediately be read and programmed as a primitive recursive function taking two configurations and the indices i and n as input.

B.3 Proof of Proposition 7

The procedure *successors* works as follows. Assume there exists at least one e' anywhere in *Unf* with $C \xrightarrow{\diamond i}_n \downarrow e'$; then there are two possibilities:

- One of these e' lies in the prefix. This is easy to determine. The corresponding state $rep(\downarrow e') \in \mathcal{S}_n$ is given back by *prefix_successors*(C, i).
- There exist such events e', but none of them lies in the prefix. The reason for $e' \notin Fin$ is the existence of a cutoff e_c, such that $e_c \leq e'$. So we can do a case analysis over the *compatible* cutoffs. A cutoff e_c is compatible with a configuration C if it is not in conflict with C, i.e., $\downarrow e_c \cup C$ is a configuration in *Fin*. If there is a compatible e_c, then for at least one of them, we have $(C \cup \downarrow e_c) \xrightarrow{\diamond i}_n \downarrow e'$. In this case we inherit the transition $C \xrightarrow{\diamond i}_n \downarrow e'$.

In the second case, we loop over all compatible cutoffs e_c looking at the configuration $C_c := C \cup \downarrow e_c$. If any $e' \in E_i$ and $C_c \xrightarrow{\diamond i}_n \downarrow e'$ exists, then there also exists an $\equiv^n$-equivalent $\downarrow e''$ for $C^* := shift^*(C_c)$ (by the isomorphism), where moreover $\downarrow e'' \prec \downarrow e'$. So *successors* is recursively called with (C^*, i). Note that C^* contains no cutoff.

Hence we apply depth first search with respect to pairs (C, i). Cycles may occur (if we hit a pair (C, i) with pathmark=true), at which we break off to ensure termination. Note that the search space is limited by the fact that C is represented in *Fin* and does not contain cutoffs.

It remains to show that the termination is correct: Assume an $e' \in E_i$ with $C \xrightarrow{\diamond i}_n \downarrow e'$ exists. Then we choose from all the suitable i-successors a $\prec$-minimal one, say $\downarrow e_{\min}$. Whenever a configuration $(C \cup \downarrow e_c)$ is shifted with $shift^*$ to obtain a configuration C' for the next call of *successors*, also $\downarrow e_{\min}$ is shifted to a stricly smaller $\downarrow e'_{\min}$, (i.e., $\downarrow e'_{\min} \prec \downarrow e_{\min}$). Thus in case we hit a configuration C twice, when searching for i-successors, $\downarrow e_{\min}$ is mapped by the various $shift^*$s to a strictly smaller state $\downarrow e^*_{\min}$ which contradicts the minimality of $\downarrow e_{\min}$. Thus whenever a configuration is investigated a second time for i-successors, we know that there cannot be one.

The main procedure *Compute_Multi_Modal_Kripke_Structure* thus only has to loop about all possible pairs $(\downarrow e, i)$ with $\downarrow e \in \mathcal{S}_n$ to check for transitions $\downarrow e \xrightarrow{\diamond i}_n \downarrow e'$ by calling *successors*.