Modal Logic and Distributed Message Passing Automata

Antti Kuusisto

Institute of Computer Science
University of Wrocław
antti.j.kuusisto@gmail.com

Abstract

In a recent article, Lauri Hella and co-authors identify a canonical connection between modal logic and deterministic distributed constant-time algorithms. The paper reports a variety of highly natural logical characterizations of classes of distributed message passing automata that run in constant time. The article leaves open the question of identifying related logical characterizations when the constant running time limitation is lifted. We obtain such a characterization for a class of finite message passing automata in terms of a recursive bisimulation invariant logic which we call *modal substitution calculus* (MSC). We also give a logical characterization of the related class $\mathcal{A}$ of infinite message passing automata by showing that classes of labelled directed graphs recognizable by automata in $\mathcal{A}$ are exactly the classes co-definable by a modal theory. A class $\mathcal{C}$ is co-definable by a modal theory if the complement of $\mathcal{C}$ is definable by a possibly infinite set of modal formulae. We also briefly discuss expressivity and decidability issues concerning MSC. We establish that MSC contains the Σ_1^{μ} fragment of the modal μ-calculus in the finite. We also observe that the single variable fragment MSC^1 of MSC is not contained in MSO, and that the SAT and FINSAT problems of MSC^1 are complete for PSPACE.

1998 ACM Subject Classification F.1.1 Models of Computation, F.4.1 Mathematical Logic, C.2.4 Distributed Systems

Keywords and phrases Modal logic, message passing automata, descriptive characterizations, distributed computing

Digital Object Identifier 10.4230/LIPIcs.CSL.2013.452

1 Introduction

Distributed computing concerns itself with the investigation of computation processes carried out by computer networks. In addition to performing local computation tasks, computers or processors in the network communicate with each other by sending messages back and forth. A distributed system can be modelled by a graph, where the nodes correspond to individual computers and the edges are communication channels through which messages can be sent, see [10]. For example, a distributed system can easily determine the sets of nodes that are directly linked to another node that has a local property P: each node with the property P simply sends a message "I have property P" to each of of its neighbours. Much of the theory of distributed computing abstracts away details related to local computation, concentrating on investigations concerning the network topology.

In the recent article [7], Hella and co-authors identify a highly natural connection between modal logic [2] and local distributed algorithms. While modal logic has been successfully applied in the distributed computing context before, the perspective in [7] is a radical departure from most of the traditional approaches, where the domain elements of a Kripke model correspond to possible states of a distributed computation process. In the framework

Computer Science Logic 2013 (CSL'13).
Editor: Simona Ronchi Della Rocca; pp. 452–468
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of [7], a distributed system ***is*** a Kripke model, where the domain elements are individual computers and the arrows of the accessibility relation are communication channels. While such an interpretation is of course always possible, it turns out to be particularly helpful in the study of *weak models of distributed computing* (see [7, 11]). The article [7] identifies *descriptive characterizations* for a comprehensive collection of *complexity classes of distributed computing* in terms of modal logics. For example, it is shown that the class SB(1) is *captured*— in the sense of descriptive complexity theory [6, 4, 8]— by ordinary modal logic ML. A graph property is in SB(1) iff it can be defined by a formula of ML. Various other characterizations are also obtained. For example the class MB(1) is captured by *graded modal logic*, i.e., a modal logic which can count the number of accessible nodes. Furthermore, the logical characterizations enable the use of logical tools in the investigation of distributed complexity classes. The article [7] provides a *complete classification* of the investigated complexity classes with respect to their computational capacities. The proofs behind the related separation results make significant use of logical methods. In particular, the notion of bisimulation turns out to be very useful in this context.

While there are various characterization results in classical descriptive complexity theory, separation results are rare, and related questions have proved very difficult. Therefore the separation results in [7] are rather *delightful*, since they nicely demonstrate the potential of the *descriptive complexity approach in the framework of non-classical computing.*

A *local algorithm* [11] is a distributed constant-time algorithm that distributed systems carry out by executing a fixed finite number of synchronized *communication rounds*. Our example above concerning the property P is an example of a trivial local algorithm. The characterizations in [7] concern local algorithms carried out by *message passing automata* that run in constant time. The article leaves open the question of identifying related logical characterizations when the constant running time limitation is lifted. We obtain such a characterization for a class of *finite message passing automata* in terms of a recursive bisimulation invariant logic which we call *modal substitution calculus* (MSC). The characterization extends directly to multimodal contexts and to systems with graded modalities, and thereby provides a nice characterization of cellular automata. We also give a logical characterization of the related class $\mathcal{A}$ of general (possibly infinite) message passing automata by showing that classes of labelled directed graphs recognizable by automata in $\mathcal{A}$ are exactly the classes co-definable by a modal theory. A class $\mathcal{C}$ is is co-definable by a modal theory if the complement of $\mathcal{C}$ is definable by a possibly infinite set of modal formulae. In distributed computing attention is often directed towards understanding issues concerning network topologies of distributed systems, and therefore it is often convenient to study infinite message passing automata with even non-recursive local computation capacities. See [5] for further elaborations on related matters.

In addition to logical characterizations, we briefly discuss expressivity and decidability issues concerning MSC. We establish that MSC contains the Σ_1^μ fragment of the modal μ-calculus in the finite. We also observe that the single-variable fragment MSC^1 of MSC is not contained in MSO, and that the SAT and FINSAT problems of MSC^1 are complete for PSPACE.

The aim of this article is two-fold. On one hand, we wish to investigate further the intimate link between distributed computing and modal logic identified in [7]. Advancing the understanding of this link can ideally be beneficial to both research on distributed computing and research on (modal) logic. Bringing together these two seemingly unrelated research fields could turn out to be a fruitful and refreshing research programme. For example, it seems that the *local model* [10, 9] of distributed computing is intimately related to *hybrid* *logic* [1]. On the other hand, we aim to promote the potential of the descriptive complexity approach in the framework on non-classical computing.

2 Preliminaries

Let S be an arbitrary set. We let $\bigcup S$ denote the set of elements x such that $x \in L$ for some $L \in S$. We let $Pow(S)$ denote the power set of S.

Let Π be an arbitrary set of *proposition symbols* $p \in \Pi$. The language $\mathrm{ML}(\Pi)$ of ordinary modal logic is generated by the grammar

$$\varphi \ ::= \ \top \mid p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \Diamond\varphi,$$

where $p \in \Pi$ and $\top$ is a logical constant symbol. Formulae in $\mathrm{ML}(\Pi)$ are called Π-*formulae*. We define the abbreviations $\bot = \neg\top$ and $\Box = \neg\Diamond\neg$. We also use the symbols $\vee$, $\rightarrow$ and $\leftrightarrow$ in the usual way.

A *Kripke model of the vocabulary* Π (Π-model) is a structure $M = (W, R, V)$, where W is a nonempty set, called the *domain* of the model, $R \subseteq W \times W$ is a binary relation, and $V : \Pi \rightarrow Pow(W)$ is a *valuation function*. The semantics of $\mathrm{ML}(\Pi)$ is defined with respect to *pointed* Π-*models* (M, w), where $M = (W, R, V)$ is a *Kripke model* of the vocabulary Π and $w \in W$ a *point* or a *node* in the domain W of the Kripke model. For $p \in \Pi$, we define $(M, w) \models p$ iff $w \in V(p)$. We also define $(M, w) \models \top$. For the connectives, we define

$$\begin{array}{lcl} (M,w) \models \neg\varphi & \Leftrightarrow & (M,w) \not\models \varphi, \\ (M,w) \models (\varphi \wedge \psi) & \Leftrightarrow & \big((M,w) \models \varphi \text{ and } (M,w) \models \psi\big), \\ (M,w) \models \Diamond\varphi & \Leftrightarrow & \exists v \in W\big(wRv \text{ and } (M,v) \models \varphi\big). \end{array}$$

The set of *subformulae* of a formula φ is defined in the standard way and denoted by $\mathrm{SUBF}(\varphi)$. The *modal depth* $md(\varphi)$ of a formula is defined recursively such that $md(\top) = md(p) = 0$, $md(\neg\psi) = md(\psi)$, $md(\psi \wedge \chi) = max\{md(\psi), md(\chi)\}$, and $md(\Diamond\psi) = md(\psi) + 1$. In a Kripke model $\big((W, R, V), w\big)$, the set $succ(w)$ of *successors* of w is the set $\{\, u \in W \mid wRu \,\}$. The set $\{\, u \in W \mid uRw \,\}$ is the set of *predecessors* of w. If φ is a modal formula and M a Kripke model, we let $||\varphi||^M$ to the the set of points w such that $(M, w) \models \varphi$.

Let Π be a set of proposition symbols. Define the set $\mathcal{S} := \{\, X_i \mid i \in \mathbb{N} \,\}$ of *schema variable symbols*. Let $\mathcal{K} \subseteq \mathcal{S}$. The set of $(\Pi, \mathcal{K})$-*schemata* of *modal substitution calculus* (MSC) is the set generated by the grammar

$$\varphi \ ::= \ \top \mid p \mid X_i \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \Diamond\varphi,$$

where $p \in \Pi$, $X_i \in \mathcal{K}$, and $\top$ is a logical constant symbol. A *terminal clause* of the vocabulary Π of modal substitution calculus is a string of the form $X_i(0) \ :- \ \varphi$, where $X_i \in \mathcal{S}$ is a schema variable and φ a formula of $\mathrm{ML}(\Pi)$. An *iteration clause* of the vocabulary Π of modal substitution calculus is a string of the form $X_i \ :- \ \psi$, where $X_i \in \mathcal{S}$ is a schema variable and ψ is a $(\Pi, \mathcal{K})$-schema for some set $\mathcal{K} \subseteq \mathcal{S}$ of schema variable symbols. The symbol X_i of a terminal clause $X_i(0) \ :- \ \varphi$ or an iteration clause $X_i \ :- \ \psi$ is called the *head predicate* of the clause, and the formulae φ and ψ are the *bodies* of the clauses. Let $\mathcal{K} = \{\, Y_1, ..., Y_n \,\} \subseteq \mathcal{S}$ be a finite nonempty set of n distinct schema variable symbols. A $(\Pi, \mathcal{K})$-*program* Λ of MSC consists of a pair

$$
\begin{array}{lllllll}
Y_1(0) & :- & \varphi_1 & \qquad & Y_1 & :- & \psi_1 \\
. & & & & . & & \\
. & & & & . & & \\
. & & & & . & & \\
Y_n(0) & :- & \varphi_n & & Y_n & :- & \psi_n
\end{array}
$$

of lists of clauses, where the first list contains n terminal clauses $Y_i(0) \;:-\; \varphi_i$ of the vocabulary Π, and the other list contains n iteration clauses $Y_i \;:-\; \psi_i$ such that each ψ_i is a $(\Pi,\mathcal{K})$-schema. Furthermore, the $(\Pi,\mathcal{K})$-program Λ specifies a set $\mathcal{A} \subseteq \mathcal{K}$ of *appointed predicates*, so formally Λ is a triple $(\mathcal{G},\mathcal{I},\mathcal{A})$, where $\mathcal{G}$ and $\mathcal{I}$ are the lists of terminal clauses and iteration clauses, respectively, and $\mathcal{A}$ is an arbitrary subset of $\mathcal{K}$ specifying the appointed predicates of Λ. A program Λ is a Π-*program* if Λ is a $(\Pi,\mathcal{K})$-program for some $\mathcal{K} \subseteq \mathcal{S}$.

We let ATOM(Λ) be the set of symbols $s \in \Pi \cup \{\top\}$ that occur in the clauses of Λ. The set HEAD(Λ) is the set of schema variable symbols that occur in the clauses of Λ. The set SUBS(φ) of *subschemata* of a schema φ is defined in the obvious way. The set SUBS(Λ) of subschemata of Λ is defined to be the smallest set such that HEAD(Λ) $\subseteq$ SUBS(Λ) and SUBS(φ) $\subseteq$ SUBS(Λ) for each φ that occurs as a body of any clause (terminal or iteration) of Λ. We define SUBF(Λ) to be the set of all schemata $\varphi \in$ SUBS(Λ) that do not contain any schema variable symbols, i.e., SUBF(Λ) is the set of sub*formulae* of Λ.

For each variable $Y_i \in$ HEAD(Λ) of Λ, we let Y_i^0 denote the right hand side of the terminal clause $Y_i(0) :- \;\varphi_i$. Recursively, assume we have defined an ML(Π)-formula Y_i^n for each $Y_i \in$ HEAD(Λ). Let $Y_j :- \;\varphi_j$ be the iteration clause corresponding to the variable Y_j. We define Y_j^{n+1} to be the ML(Π)-formula obtained by simultaneously replacing each variable Y_i of the schema φ_j by the formula Y_i^n. Let φ be an arbitrary schema in SUBS(Λ). We let φ^n denote the ML(Π)-formula obtained from φ by simultaneously replacing each variable $Y_i \in$ HEAD(Λ) in φ by the formula Y_i^n.

Let (M,w) be a pointed Π-model and Λ a Π-program of MSC. We define that $(M,w) \models \Lambda$ if there is an appointed variable Y of Λ such that for some $n \in \mathbb{N}$, we have $(M,w) \models Y^n$. We say that Λ is true in (M,w), or that (M,w) satisfies Λ.

Let Π be a finite set of proposition symbols. A *message passing automaton* A of the vocabulary Π is a tuple (Q,M,π,δ,μ,F). The object Q is a nonempty set of *states*. The set Q can be finite or countably infinite. The object M is a nonempty set of *messages*. The set M can be finite or countably infinite. The object $\pi : Pow(\Pi) \longrightarrow Q$ is an *initial transition function* that determines the beginning state of A. The object $\delta : Pow(M) \times Q \longrightarrow Q$ is a *transition function* that constructs a new state in Q based on a set $N \in Pow(M)$ of messages received and a previous state in Q. The object $\mu : Q \longrightarrow M$ is a *message construction function* that constructs a message for the automaton to send forward based on the state of the automaton. The object $F \subseteq Q$ is a set of *accepting states* of the automaton. A message passing automaton such that the sets Q and M are finite, is a *finite message passing automaton* FMPA. (MPA stands for a message passing automaton.)

A message passing automaton A of a vocabulary Π is *run* on a Kripke model (W,R,V) of the vocabulary Π, considered to be a distributed system. Intuitively, we put a copy (A,w) of the automaton to each node $w \in W$. Then, each automaton (A,w) first scans the propositional information of the node w, and then makes a transition to a beginning state based on this. Then, the automata (A,u), where $u \in W$, begin running in *synchronized steps*. During each step, each automaton first broadcasts a message to each of its neighbours with respect to R, and then updates its state based on the set of messages it receives from its neighbours. More formally, the automaton A and Kripke model (W,R,V) define a *synchronized distributed* *system* which executes an omega-sequence of *communication rounds* defined as follows. Each round $n \in \mathbb{N}$ defines a *global configuration* $f_n : W \longrightarrow Q$. The configuration of the zeroth round is the function f_0 such that $f_0(w) = \pi(\{\ p \in \Pi \mid w \in V(p)\ \})$. Recursively, assume that we have defined f_n, and call $N\ = \big\{\ m \in M \mid m = \mu(f_n(v)),\ v \in succ(w)\ \big\}$. Then $f_{n+1}(w)\ =\ \delta\big(N,\ f_n(w)\big)$.

Notice that the automaton A at node w receives messages from its *successors*, so messages flow in the direction opposite to the arrows (or pairs) of the relation R. This may seem strange at first, and indeed a more natural definition would stipulate that messages flow in the direction of the arrows. The reason behind the choice here is mainly technical, and related to the technical relationship between message passing automata and modal logic. An alternative approach would be to consider modal logics with only backwards looking diamonds, or to define a Kripke structure M corresponding to a distributed system S such that M would be obtained from S by reversing the arrows of S.

When we talk about *the state of an automaton A at the node w in round n*, we mean the state $f_n(w)$. We define that an automaton A *accepts* a pointed model (M, w) if there exists some $n \in \mathbb{N}$ such that $f_n(w) \in F$, in other words, if the automaton A at w visits an accepting state during the execution of the distributed system. Note that the automaton A at w does not stop passing messages even if it has visited an accepting state. Therefore this model of computing can be regarded as a kind of a *semidecision framework for distributed computation*: an accepting node will eventually know it has accepted, but a nonaccepting node can keep running forever without knowledge of acceptance. These kinds of asymmetric acceptance conditions are common in distributed computing (see for example [7]). It would be natural to consider even more complex acceptance conditions, for example we could define a subset $G \subseteq Q$ of *rejecting* states. For the sake of space limitations, we shall not consider such possibilities here. However, the considerations below can easily be adapted to deal with various more complex acceptance scenarios.

3 MSC captures FMPA-recognizability

3.1 Specifying FMPAs in MSC

Let $(M, w) = ((W, R, V), w)$ be a pointed model and A an automaton of the same vocabulary as M. Let Q be the set of states of A. For each $u \in W$, let $A\big((M, u), n\big)$ denote the state of A at u in round n. The set $\{\ q \in Q \mid q = A\big((M, u), n\big)$ for some $u \in succ(w)\ \}$ is called the *set of states defined by the successors of w in round n.*

▶ **Theorem 1.** *Let Π be a finite set of proposition symbols. Let A be a finite message passing automaton of the vocabulary Π. There exists a Π-program Λ_A of* MSC *such that for all pointed Π-models (M, w), the automaton A accepts (M, w) iff $(M, w) \models \Lambda_A$.*

Proof. Let $A = (Q, M, \pi, \delta, \mu, F)$. Define a formula variable X_q for each state $q \in Q$. For each $q \in Q$, define the terminal clause

$$X_q(0)\ :-\ \bigvee_{P \subseteq \Pi,\ \pi(P) = q} \Big(\bigwedge_{p \in P} p\ \wedge \bigwedge_{p \in \Pi \setminus P} \neg p \Big). \tag{1}$$

(Note that $\bigvee \emptyset = \bot$ and $\bigwedge \emptyset = \top$.) Let $S \subseteq Q$ be a set of states. Define the schema

$$\varphi_S\ :=\ \bigwedge_{q \in S} \Diamond X_q\ \wedge \bigwedge_{q \notin S} \neg \Diamond X_q.$$

If $S \subseteq Q$ is a set of states, we denote the set $\{\ \mu(q) \mid q \in S\ \}$ by $\mu(S)$. We define $\mathcal{M}(q,q')$ to be the set of exactly all sets $S \subseteq Q$ such that $\delta(\mu(S),q) = q'$. For each state $q' \in Q$, define the iteration clause

$$X_{q'} \ :- \ \bigwedge_{q \in Q} \big(\ X_q \ \rightarrow \bigvee_{S \in \mathcal{M}(q,q')} \varphi_S\ \big). \tag{2}$$

The program Λ_A is the Π-program defined by the terminal clauses given by Equation 1 above and the iteration clauses given by Equation 2. The set of appointed predicates is the set of symbols X_q such that $q \in F$.

Let $M = (W, R)$ be a Kripke model of the vocabulary Π. We will show that for each node $v \in W$, each state $q \in Q$, and each round $n \in \mathbb{N}$, the state of the automaton A at node v in round n is q if and only if $(M,v) \models X_q^n$. This is shown by an induction on n. The case for $n = 0$ follows immediately by the definition of the initial transition function π and the definition of $X_q(0)$.

Assume that $(M,w) \models X_{q'}^{n+1}$. Thus

$$(M,w) \models \bigwedge_{q \in Q} \big(\ X_q^n \ \rightarrow \bigvee_{S \in \mathcal{M}(q,q')} \varphi_S^n\ \big).$$

Let $r \in Q$ be the state of A at w in round n. By the induction hypothesis, we have $(M,w) \models X_r^n$, and therefore

$$(M,w) \models \bigvee_{S \in \mathcal{M}(r,q')} \varphi_S^n.$$

Thus $(M,w) \models \varphi_S^n$ for some $S \in \mathcal{M}(r,q')$. By the definition of schema φ_S, each formula X_q^n such that $q \in S$ is satisfied by some successor of w, and there exists no successor of w that satisfies a formula X_q^n such that $q \notin S$. Therefore, by the induction hypothesis, the set of states defined by $succ(w)$ in round n is S. Since $S \in \mathcal{M}(r,q')$, we conclude that the state of the automaton at w in round $n+1$ is q'.

For the converse, assume that the state of A at w in round $n+1$ is q'. Let r be the state of A at w in round n. Let S be the set of states defined by $succ(w)$ in round n. Hence, by the induction hypothesis, we have $(M,w) \models \varphi_S^n$. We also have $S \in \mathcal{M}(r,q')$ by the definition of r, q' and S. Therefore

$$(M,w) \models \bigvee_{S \in \mathcal{M}(r,q')} \varphi_S^n.$$

We also know, by the induction hypothesis, that for all $q \in Q$, $(M,w) \models X_q^n$ iff $q = r$. Therefore

$$(M,w) \models \bigwedge_{q \in Q} \big(\ X_q^n \ \rightarrow \bigvee_{S \in \mathcal{M}(q,q')} \varphi_S\ \big),$$

and thus $(M,w) \models X_{q'}^{n+1}$, as desired. ◀

3.2 Simulating MSC programs by FMPAs

Let Λ be a program of MSC, and let $\mathrm{HEAD}(\Lambda) = \{\, Y_1, ..., Y_m \,\}$. For each $n \in \mathbb{N}$, we define $md(\Lambda, n) = max\{\, md(Y_1^n), ..., md(Y_m^n) \,\}$. We let $mdt(\Lambda)$ denote the maximum modal depth of the body formulae in the terminal clauses of Λ. Similarly, we let $mdi(\Lambda)$ denote the maximum modal depth of the body schemata of the iteration clauses of Λ.

Define $scope(\Lambda, 0) = md(\Lambda, 0)$ and $scope(\Lambda, n+1) = scope(\Lambda, n) + max\{1, mdi(\Lambda)\}$. If $(M,w) \models \Lambda$, then the *scope of* Λ *at* w is the number $scope(\Lambda, n)$, where n is the smallest number $k \in \mathbb{N}$ such that we have $(M, w) \models Y_i^k$ for some appointed predicate Y_i. If $(M, w) \not\models \Lambda$, the scope of Λ at w is ω. Scope is a relatively natural spatio-temporal complexity measure for the execution of an MSC program, when the execution is done by first evaluating each formula Y_i^0, then each formula Y_i^1, and so on. Notice that even if $mdi(\Lambda) = 0$, scope is increased after each iteration step. It is of course possible to define other natural complexity measures for MSC programs.

Let A be an automaton and (M, w) a pointed model. If A accepts (M, w), then the *decision time of A at w* is the smallest number k such that the state of A at w is an accepting state in round k. If A does not accept (M, w), the decision time of A at w is ω.

Next we show how to define, when given a program Λ of MSC, a corresponding automaton A_Λ that accepts exactly the pointed models (M, w) such that $(M, w) \models \Lambda$. Furthermore, the decision time of A_Λ at each node w will be equal to the scope of Λ at w. Roughly, the states of A_Λ will encode finite sets of formulae satisfied by nodes of the underlying model. For more of the intuition behind the definition of A_Λ, see the proof of Theorem 2.

Let Π be a finite set of proposition symbols and fix a Π-program Λ of MSC. We assume that $mdi(\Lambda) \geq 1$. The pathological case where $mdi(\Lambda) = 0$ is discussed separately.

The set Q_Λ of states of A_Λ contains all pairs (S, m), where $m \leq mdi(\Lambda) - 1$ is a nonnegative integer and $S \subseteq \mathrm{SUBS}(\Lambda)$ a set of schemata φ such that $md(\varphi) \leq m$. The set Q_Λ also contains all triples (S, m, f), where $m \leq mdt(\Lambda) - 1$ is a nonnegative integer, $S \subseteq \mathrm{SUBF}(\Lambda)$ is a set of *formulae* φ such that $md(\varphi) \leq m$, and f is simply a symbol indicating that this state encodes sets of *formulae* in $\mathrm{SUBF}(\Lambda)$. There are no other states in Q_Λ. The set of messages M_Λ is $Pow\big(\mathrm{SUBS}(\Lambda)\big)$. (Some states and some messages may turn out to be irrelevant for the computation of A_Λ.)

We then define the transition function π of A_Λ. Assume first that $mdt(\Lambda) \geq 1$. Let $P \subseteq \Pi$ be a set of proposition symbols. Define a set $U \subseteq \mathrm{SUBF}(\Lambda)$ to be the smallest set such that the following conditions hold.

1. $\big(P \cap \mathrm{SUBF}(\Lambda)\big) \cup \big(\{\top\} \cap \mathrm{SUBF}(\Lambda)\big) \subseteq U$.
2. For each $\neg\varphi \in \mathrm{SUBF}(\Lambda)$ of the modal depth 0, $\neg\varphi \in U$ iff $\varphi \notin U$.
3. For each $(\varphi \wedge \psi) \in \mathrm{SUBF}(\Lambda)$ of the modal depth 0, $(\varphi \wedge \psi) \in U$ iff both $\varphi \in U$ and $\psi \in U$.

We define $\pi(P) = (U, 0, f)$. If $mdt(\Lambda) = 0$, we define $\pi(P)$ for the set $P \subseteq \Pi$ of proposition symbols differently. First define a set $T \subseteq \mathrm{SUBF}(\Lambda)$ to be the smallest set such that the following conditions hold.

1. $\big(P \cap \mathrm{SUBF}(\Lambda)\big) \cup \big(\{\top\} \cap \mathrm{SUBF}(\Lambda)\big) \subseteq T$.
2. For each formula $\neg\varphi \in \mathrm{SUBF}(\Lambda)$ of the modal depth 0, we have $\neg\varphi \in T$ iff $\varphi \notin T$.
3. For each formula $(\varphi \wedge \psi) \in \mathrm{SUBF}(\Lambda)$ of the modal depth 0, we have $(\varphi \wedge \psi) \in T$ iff both $\varphi \in T$ and $\psi \in T$.

Now let T' be the set of symbols in $\mathrm{ATOM}(\Lambda) \cup \mathrm{HEAD}(\Lambda)$ of the modal depth 0 such that the following conditions hold.

1. For each $X \in \mathrm{HEAD}(\Lambda)$, we have $X \in T'$ iff $X^0 \in T$.
2. For each $\varphi \in \mathrm{ATOM}(\Lambda)$, we have $\varphi \in T'$ iff $\varphi \in T$.

Define U to be the set of schemata in $\mathrm{SUBS}(\Lambda)$ of the modal depth 0 such that the following conditions hold.

1. For each $\varphi \in \mathrm{ATOM}(\Lambda) \cup \mathrm{HEAD}(\Lambda)$, $\varphi \in U$ iff $\varphi \in T'$.
2. For each schema $\neg\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0, $\neg\varphi \in U$ iff $\varphi \notin U$.
3. For each schema $(\varphi \wedge \psi) \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0, $(\varphi \wedge \psi) \in U$ iff both $\varphi \in U$ and $\psi \in U$.

We define $\pi(P) = (U, 0)$.

We then define the transition function δ of A_Λ. Let (S, m) be a state of A_Λ. Let $N \subseteq M_\Lambda$ be a set of messages. Assume that $m < mdi(\Lambda) - 1$. Assume there exists a smallest set U such that the following conditions hold.

1. For each schema $\varphi \in \mathrm{SUBS}(\Lambda)$ such that $md(\varphi) < m + 1$, we have $\varphi \in U$ iff $\varphi \in S$.
2. For each schema $\Diamond\varphi \in \mathrm{SUBS}(\Lambda)$ such that $md(\Diamond\varphi) \leq m+1$, we have $\Diamond\varphi \in U$ iff $\varphi \in \bigcup N$.
3. For each schema $(\varphi \wedge \psi) \in \mathrm{SUBS}(\Lambda)$ such that $md(\varphi \wedge \psi) \leq m+1$, we have $(\varphi \wedge \psi) \in U$ iff both $\varphi \in U$ and $\psi \in U$.
4. For each schema $\neg\varphi \in \mathrm{SUBS}(\Lambda)$ such that $md(\neg\varphi) \leq m + 1$, we have $\neg\varphi \in U$ iff $\varphi \notin U$.

We then define $\delta\big(N, (S, m)\big)$ to be the state $(U, m + 1)$. If no set U satifying the above conditions exists, we define $\delta\big(N, (S, m)\big)$ arbitrarily.

If $m = mdi(\Lambda) - 1$, we define $\delta\big(N, (S, m)\big)$ differently. Assume there exists a smallest set $T \subseteq \mathrm{SUBS}(\Lambda)$ such that the following conditions hold. (If no such set T exists, $\delta((S, m), N)$ is defined arbitrarily.)

1. For each schema $\varphi \in \mathrm{SUBS}(\Lambda)$ such that $md(\varphi) < m + 1$, we have $\varphi \in T$ iff $\varphi \in S$.
2. For each schema $\Diamond\varphi \in \mathrm{SUBS}(\Lambda)$ such that $md(\Diamond\varphi) \leq m+1$, we have $\Diamond\varphi \in T$ iff $\varphi \in \bigcup N$.
3. For each schema $(\varphi \wedge \psi) \in \mathrm{SUBS}(\Lambda)$ such that $md(\varphi \wedge \psi) \leq m+1$, we have $(\varphi \wedge \psi) \in T$ iff both $\varphi \in T$ and $\psi \in T$.
4. For each schema $\neg\varphi \in \mathrm{SUBS}(\Lambda)$ such that $md(\neg\varphi) \leq m + 1$, we have $\neg\varphi \in T$ iff $\varphi \notin T$.

Now define a set $T' \subseteq \mathrm{HEAD}(\Lambda) \cup \mathrm{ATOM}(\Lambda)$ such that the following conditions hold.

1. For each $X \in \mathrm{HEAD}(\Lambda)$, we have $X \in T'$ iff $\varphi \in T$, where φ is the body of the iteration clause for X.
2. For each $\varphi \in \mathrm{ATOM}(\Lambda)$, we have $\varphi \in T'$ iff $\varphi \in T$.

Define U to be the set of schemata of the modal depth 0 in $\mathrm{SUBS}(\Lambda)$ such that the following conditions hold.

1. For each $\varphi \in \mathrm{ATOM}(\Lambda) \cup \mathrm{HEAD}(\Lambda)$, $\varphi \in U$ iff $\varphi \in T'$.
2. For each $\neg\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0, $\neg\varphi \in U$ iff $\varphi \notin U$.
3. For $(\varphi \wedge \psi) \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0, $(\varphi \wedge \psi) \in U$ iff both $\varphi \in U$ and $\psi \in U$.

Then $\delta(N, (S, m))$ is defined to be the state $(U, 0)$.

Let (S, m, f) be state of A_Λ. Let $N \subseteq M_\Lambda$ be a set of messages. Assume that $m < mdt(\Lambda) - 1$. Assume there exists a smallest set U such that the following conditions hold.

1. For each formula $\varphi \in \mathrm{SUBF}(\Lambda)$ such that $md(\varphi) < m + 1$, we have $\varphi \in U$ iff $\varphi \in S$.
2. For each formula $\Diamond\varphi \in \mathrm{SUBF}(\Lambda)$ such that $md(\Diamond\varphi) \leq m + 1$, we have $\Diamond\varphi \in U$ iff $\varphi \in \bigcup N$.
3. For each formula $(\varphi \wedge \psi) \in \mathrm{SUBF}(\Lambda)$ such that $md(\varphi \wedge \psi) \leq m+1$, we have $(\varphi \wedge \psi) \in U$ iff both $\varphi \in U$ and $\psi \in U$.
4. For each formula $\neg\varphi \in \mathrm{SUBF}(\Lambda)$ such that $md(\neg\varphi) \leq m + 1$, we have $\neg\varphi \in U$ iff $\varphi \notin U$.

We then define $\delta\big(N, (S, m, f)\big)$ to be the state $(U, m + 1, f)$. If no set U satifying the above conditions exists, we define $\delta\big(N, (S, m, f)\big)$ arbitrarily.

If $m = mdt(\Lambda) - 1$, we define $\delta\big(N, (S, m, f)\big)$ differently. Assume there exists a smallest set $T \subseteq \mathrm{SUBS}(\Lambda)$ such that the following conditions hold. (If no such set T exists, $\delta(N, (S, m, f))$ is defined arbitrarily.)

1. For each formula $\varphi \in \mathrm{SUBF}(\Lambda)$ such that $md(\varphi) < m + 1$, we have $\varphi \in T$ iff $\varphi \in S$.
2. For each formula $\Diamond\varphi \in \mathrm{SUBF}(\Lambda)$ such that $md(\Diamond\varphi) \leq m+1$, we have $\Diamond\varphi \in T$ iff $\varphi \in \bigcup N$.
3. For each formula $(\varphi \wedge \psi) \in \mathrm{SUBF}(\Lambda)$ such that $md(\varphi \wedge \psi) \leq m+1$, we have $(\varphi \wedge \psi) \in T$ iff both $\varphi \in T$ and $\psi \in T$.
4. For each formula $\neg\varphi \in \mathrm{SUBF}(\Lambda)$ such that $md(\neg\varphi) \leq m + 1$, we have $\neg\varphi \in T$ iff $\varphi \notin T$.

Now define a set $T' \subseteq \mathrm{HEAD}(\Lambda) \cup \mathrm{ATOM}(\Lambda)$ such that the following conditions hold.

1. For each $X \in \text{HEAD}(\Lambda)$, we have $X \in T'$ iff $X^0 \in T$.
2. For each $\varphi \in \text{ATOM}(\Lambda)$, we have $\varphi \in T'$ iff $\varphi \in T$.

Define U to be the set of schemata in $\text{SUBS}(\Lambda)$ of the modal depth 0 such that the following conditions hold.

1. For each $\varphi \in \text{ATOM}(\Lambda) \cup \text{HEAD}(\Lambda)$, $\varphi \in U$ iff $\varphi \in T'$.
2. For each $\neg\varphi \in \text{SUBS}(\Lambda)$ of the modal depth 0, $\neg\varphi \in U$ iff $\varphi \notin U$.
3. For all $(\varphi \wedge \psi) \in \text{SUBS}(\Lambda)$ of the modal depth 0, $(\varphi \wedge \psi) \in U$ iff both $\varphi \in U$ and $\psi \in U$.

Then $\delta(N, (S, m, f))$ is defined to be the state $(U, 0)$.

The message construction function μ of A_Λ is defined such that $\mu\big((S, m)\big) = S$ and $\mu\big((S, m, f)\big) = S$. The set F of accepting states of A_Λ is the set of states $(S, 0)$ such that we have $Y \in S$ for some appointed head predicate of Λ.

We have now defined the automaton A_Λ, assuming that $mdi(\Lambda) \neq 0$. The definition of A_Λ in the pathological case where $mdi(\Lambda) = 0$ is discussed in the proof of Theorem 2.

▶ **Theorem 2.** *Let Π be a finite set of proposition symbols. Let Λ be a Π-program of* MSC. *Let (M, w) be a pointed Π-model. We have $(M, w) \models \Lambda$ if and only if A_Λ accepts (M, w). Furthermore, the scope of Λ at w equals the decision time of A_Λ at w.*

Proof. We begin by describing the idea of the proof. Let W be the domain of M. The automata A_Λ at the nodes $u \in W$ first compute the extensions $||X^0||^M$ of formulae X^0 for each $X \in \text{HEAD}(\Lambda)$. The automata then operate in *cycles* of communication rounds. During a cycle, the automata compute the extensions of formulae X^{n+1} based on the extensions of formulae X^n computed during the previous cycle. The communication steps during the cycle contribute to the information about extensions of formulae of greater and greater modal depths. The proof will proceed by induction on the iteration number n, and each step of the induction will be a subinduction on modal depth of schemata. We assume that $mdi(\Lambda) \neq 0$. The case where $mdi(\Lambda) = 0$ will be briefly discussed at the end of the proof.

Define a set C_0 such that $C_0 = \{-1\} \times \{0, ..., mdt(\Lambda) - 1\}$ if $mdt(\Lambda) \neq 0$, and $C_0 = \emptyset$ if $mdt(\Lambda) = 0$. Define also $C_1 = \mathbb{N} \times \{0, ..., mdi(\Lambda) - 1\}$. Let $C = C_0 \cup C_1$. Order the pairs in C lexicographically, i.e., $(i, j) < (i', j') \Leftrightarrow (i < i' \vee (i = i' \wedge j < j'))$. Let $<_C$ denote this order. Let g be the isomorphism from $(C, <_C)$ to $(\mathbb{N}, <)$. We let $Q_v(i, j)$ denote the set of schemata φ occurring in the state (S, m) or (S, m, f) of the automaton A_Λ at node v in the round $g((i, j))$. Observe that $Q_v(i, j)$ contains schemata of the modal depth up to j.

We will show by induction on n that the equivalence

$$(M, v) \models \varphi^n \ \text{ iff } \ \varphi \in Q_v\big(n, 0\big)$$

holds for all $v \in W$, all $n \in \mathbb{N}$, and all schemata $\varphi \in \text{SUBS}(\Lambda)$ of the modal depth 0. Each step of the induction is a subinduction on the modal depth of schemata.

Let $n = 0$. Some of the details of the case $n = 0$ are straightforward or rather similar to corresponding details of the case $n > 0$, and therefore omitted here. (See the appendix for the omitted cases.) The case $n > 0$ is discussed in detail, and the omitted details for the case $n = 0$ can be easily constructed from the corresponding arguments for the case $n > 0$.

Assume that $mdt(\Lambda) \neq 0$. For the case $mdt(\Lambda) = 0$, see the appendix. Call $\Phi = \text{SUBF}(\Lambda) \cap \Pi$. By the definition of the transition function π, we have $(M, v) \models p \Leftrightarrow p \in \pi\big(\{\, p \in \Pi \mid v \in V(p)\,\}\big)$ for each $p \in \Phi$. Therefore, for each atomic formula $\varphi \in \text{ATOM}(\Lambda)$, we have $(M, v) \models \varphi \Leftrightarrow \varphi \in Q_v(-1, 0)$. Hence, since every formula $\varphi \in \text{SUBF}(\Lambda)$ of the modal depth 0 is a Boolean combination of formulae in $\varphi \in \text{ATOM}(\Lambda)$, we have $(M, v) \models \varphi \Leftrightarrow \varphi \in Q_v(-1, 0)$ for all $\varphi \in \text{SUBF}(\Lambda)$ of the modal depth 0.

We then need to establish that the equivalence $(M, v) \models \psi \;\Leftrightarrow\; \psi \in Q_v\big(-1, mdt(\Lambda) - 1\big)$ holds for each $v \in W$ and each $\psi \in \mathrm{SUBF}(\Lambda)$ such that $md(\psi) \leq mdt(\Lambda) - 1$. If $mdt(\Lambda) = 1$, we are done. If not, the equivalence can be proved by induction on the modal depth of formulae. We shall omit the details here (see the appendix). Once we have established that $(M, v) \models \psi \;\Leftrightarrow\; \psi \in Q_v(-1, mdt(\Lambda) - 1)$, we can show that therefore $(M, v) \models \varphi^0 \;\Leftrightarrow\; \varphi \in Q_v\big(0, 0\big)$ for all *schemata* $\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0 and all $v \in W$, thereby concluding the argument for the case $n = 0$. We omit the details here (see the appendix).

Now assume the claim of the main induction holds for $n \in \mathbb{N}$, and consider the case for $n+1$. By the induction hypothesis, we have $(M, v) \models \varphi^n \;\Leftrightarrow\; \varphi \in Q_v(n, 0)$ for all $v \in W$ and all $\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0. We need to prove that

$$(M, v) \models \varphi^{n+1} \;\Leftrightarrow\; \varphi \in Q_v(n+1, 0)$$

for all $v \in W$ and all schemata in $\mathrm{SUBS}(\Lambda)$ of the modal depth 0. In order to show this, we shall first establish that $(M, v) \models \varphi^n \;\Leftrightarrow\; \varphi \in Q_v(n, k)$ for all $v \in W$ and all $\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth k such that $0 \leq k \leq mdi(\Lambda) - 1$. This is proved by by induction on the modal depth k of schemata.

Since we already know that $(M, v) \models \varphi^n \;\Leftrightarrow\; \varphi \in Q_v(n, 0)$ for all $v \in W$ and all $\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0, the basis of the subinduction on modal depth is clear. In the case $mdi(\Lambda) = 1$, this suffices, and no subinduction is actually needed. Therefore assume that $mdi(\Lambda) > 1$ and let $k \in \{\, 0, \ldots, mdi(\Lambda) - 2 \,\}$. Assume that $(M, v) \models \varphi^n \;\Leftrightarrow\; \varphi \in Q_v(n, k)$ for all schemata in $\mathrm{SUBF}(\Lambda)$ of the modal depth up to k and all $v \in W$. Let $\varphi \in \mathrm{SUBS}(\Lambda)$ be a schema of the modal depth $k+1$. The schema φ is a Boolean combination of schemata $\Diamond\psi$, where $md(\psi) \leq k$. It therefore suffices to show that for each such schema $\Diamond\psi$, we have $(M, v) \models \Diamond\psi^n \;\Leftrightarrow\; \Diamond\psi \in Q_v(n, k+1)$.

Assume first that $(M, v) \models \Diamond\psi^n$. Therefore some successor u of v satisfies $(M, u) \models \psi^n$. By the induction hypothesis, $\psi \in Q_u(n, k)$. Therefore the automaton A_Λ at u sends a message L such that $\psi \in L$ to its predecessors in round $g((n, k+1))$. Thus $\Diamond\psi \in Q_v(n, k+1)$.

Conversely, assume that $\Diamond\psi \in Q_v(n, k+1)$. Therefore v receives a message L such that $\psi \in L$ from some successor u in round $g((n, k+1)) = g((n, k)) + 1$. Hence $\psi \in Q_u(n, k)$. By the induction hypothesis, $(M, u) \models \psi^n$. Therefore $(M, v) \models \Diamond\psi^n$.

We have now established that $(M, v) \models \varphi^n \;\Leftrightarrow\; \varphi \in Q_v(n, k)$ for all $v \in W$ and all $\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth k such that $0 \leq k \leq mdi(\Lambda) - 1$. We shall next show that therefore $(M, v) \models \varphi^{n+1} \;\Leftrightarrow\; \varphi \in Q_v(n+1, 0)$ for all $v \in W$ and all $\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0.

Recall the definition of the sets T, T' and U in the definition of δ on input states (S, m) in the case where $m = mdi(\Lambda) - 1$. We shall first show that $(M, w) \models \varphi^n \;\Leftrightarrow\; \varphi \in T$ holds for each $\varphi \in \mathrm{SUBS}(\Lambda)$ such that $md(\varphi) \leq mdi(\Lambda)$.

Let $\varphi \in \mathrm{SUBS}(\Lambda)$ be a schema such that $md(\varphi) \leq mdi(\Lambda)$. The schema φ is a Boolean combination of schemata $\Diamond\psi$, where $md(\psi) < mdi(\Lambda)$. It therefore suffices to show that for each such schema $\Diamond\psi$, we have $(M, v) \models \Diamond\psi^n \;\Leftrightarrow\; \Diamond\psi \in T$. This is shown by an argument analogous to the corresponding argument discussed above. (See the appendix.)

We can now conclude that $(M, v) \models X^{n+1} \;\Leftrightarrow\; X \in T'$ for all head symbols $X \in \mathrm{HEAD}(\Lambda)$, and also $(M, v) \models \varphi \;\Leftrightarrow\; \varphi \in T'$ for all atomic formulae $\varphi \in \mathrm{ATOM}(\Lambda)$. Therefore, since every schema $\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0 is a Boolean combination of formulae in $\mathrm{ATOM}(\Lambda)$ and head symbols in $\mathrm{HEAD}(\Lambda)$, we have $(M, v) \models \varphi^{n+1} \;\Leftrightarrow\; \varphi \in Q_v(n+1, 0)$ for all $v \in W$ and all $\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0, as required.

Finally, if $mdi(\Lambda) = 0$, it is easy to define an automaton Λ that satisfies the requirements of the Theorem. The number $|\mathrm{HEAD}(\Lambda)|$ of different head predicates is finite, so there are finitely many different truth distributions that the set of head predicates can obtain. Therefore, once we have computed the extensions of the formulae X^0, we can directly check at each node, without further communication with neighbouring automata, whether *any* iteration X^n of any appointed head predicate X of Λ is true. This holds because the Boolean combinations obtained by the head predicate set must begin repeating periodically after sufficiently many iterations. ◀

4 Modal theories capture complements of MPA-recognizable classes

Let Π be a finite set of proposition symbols. Let $\mathcal{C}$ be the class of pointed Π-models. A class $\mathcal{K} \subseteq \mathcal{C}$ of pointed Π-models is said to be *definable by a modal theory* if there exists a set Φ of modal Π-formulae (Π-theory) such that for all $(M, w) \in \mathcal{C}$, we have $(M, w) \models \Phi$ iff $(M, w) \in \mathcal{K}$. By $(M, w) \models \Phi$ we mean that $(M, w) \models \varphi$ for all $\varphi \in \Phi$. A class $\mathcal{K}' \subseteq \mathcal{C}$ is said to be *co-definable by a modal theory* Φ if $\mathcal{C} \setminus \mathcal{K}'$ is definable by the modal theory Φ.

Let Π be a finite set of proposition symbols. The set T_0 of Π-*types* of the modal depth 0 is defined to be the set containing a conjunction

$$\bigwedge_{p \in S} p \;\wedge\; \bigwedge_{p \notin S} \neg p$$

for each set $S \subseteq \Pi$, and no other formulae. We assume some canonical bracketing and ordering of conjuncts, so that there is exactly one conjunction for each set S in T_0. Note also that $\bigwedge \emptyset \;=\; \top$. The type $\tau_{(M,w),0}$ of a pointed Π-model (M, w) is the unique formula φ in T_0 such that $(M, w) \models \varphi$.

Assume then that we have defined the set T_n of Π-types of the modal depth n. Assume that T_n is finite, and assume also that each pointed Π-model (M, w) satisfies exactly one type $\tau_{(M,w),n}$ of the modal depth n. Define

$$\tau_{(M,w),n+1} \;:=\; \tau_{(M,w),n} \;\wedge\; \bigwedge\{\, \Diamond\tau \mid \tau \in T_n,\; (M, w) \models \Diamond\tau \,\} \;\wedge\; \bigwedge\{\, \neg\Diamond\tau \mid \tau \in T_n,\; (M, w) \not\models \Diamond\tau \,\}.$$

The formula $\tau_{(M,w),n+1}$ is the Π-type of the modal depth $n + 1$ of (M, w). We assume some standard ordering of conjuncts and bracketing, so that if two types $\tau_{(M,w),n+1}$ and $\tau_{(N,v),n+1}$ are equivalent, they are actually the same formula. We define T_{n+1} to be the set $\{\, \tau_{(M,w),n+1} \mid (M, w) \text{ is a pointed } \Pi\text{-model} \,\}$. We observe that the set T_{n+1} is finite, and that for each pointed Π-model (M, w), there is exactly one type $\tau \in T_{n+1}$ such that $(M, w) \models \tau$.

It is easy to show that each Π-formula φ of modal logic is equivalent to the disjunction of exactly all Π-types τ of the modal depth $md(\varphi)$ such that $\tau \models \varphi$. By $\tau \models \varphi$ we mean that for all pointed Π-models (M, w), we have $(M, w) \models \tau \;\Rightarrow\; (M, w) \models \varphi$. (Note that $\bigvee \emptyset = \bot$).

Define a *type automaton* A for Π to be message passing automaton whose set of states is exactly the set $\mathcal{T}$ of all Π-types. The set of messages is also the set $\mathcal{T}$. Furthermore, the initial transition function π is defined such that the state of A at (M, w) in round $n = 0$ is the type $\tau_{(M,w),0}$. Let N be a set of types. If all types in N are types of the same modal depth n, and if τ is a type of the modal depth n, we define $\delta(N, \tau)$ to be the type

$$\tau_{n+1} \;=\; \tau \wedge \bigwedge_{\sigma \in N} \Diamond\,\sigma \;\wedge \bigwedge_{\sigma \in T_n \setminus N} \neg\Diamond\,\sigma.$$

On other inputs, δ is defined arbitrarily. The message construction function μ is defined such that $\mu(\tau) = \tau$. The set of accepting states can be defined differently for different type automata A of the vocabulary Π. It is easy to see that the state of any type automaton A at (M, v) in round n is τ iff the type of the modal depth n of (M, v) is τ.

▶ **Theorem 3.** *Let Π be a finite set of proposition symbols. Each class of pointed Π-models co-definable by a modal Π-theory is recognizable by a message passing automaton.*

Proof. Let $\mathcal{K}$ be a class of Π-models co-definable by a modal Π-theory Φ. Let φ be an arbitrary formula in Φ. The formula $\neg\varphi$ is equivalent to the disjunction of Π-types τ of the modal depth $md(\varphi)$ such that $\tau \models \neg\varphi$. Let $D(\neg\varphi)$ denote the disjunction. We write $\tau \in D(\neg\varphi)$ in order to indicate that τ is one of the disjuncts of $D(\neg\varphi)$.

Let $\mathcal{T}$ denote the set of exactly all Π-types. Define a Π-type automaton A such that the set of accepting states is the set $\{\, \tau \in \mathcal{T} \mid \tau \in D(\neg\varphi) \text{ for some } \varphi \in \Phi \,\}$. It is straightforward to show that the automaton accepts exactly the class $\mathcal{K}$ of pointed Π-models. ◀

▶ **Theorem 4.** *Let Π be a finite set of proposition symbols. Each class of pointed Π-models recognizable by a message passing automaton is co-definable by a modal theory.*

Proof. Let (M, w) be a pointed Π-model. Let A be a message passing automaton whose set of proposition symbols is Π. Let $n \in \mathbb{N}$. We let $A\big((M, w), n\big)$ denote the state of the automaton A at the node w in round n. We shall begin the proof by showing that the following statements are equivalent for all pointed Π-models (M, w) and (N, v) and all $n \in \mathbb{N}$.

1. The models (M, w) and (N, v) satisfy exactly the same Π-type of the depth n.
2. $A\big((M, w), k\big) = A\big((N, v), k\big)$ for each $k \leq n$ and each message passing automaton A whose set of proposition symbols is Π.

We prove the claim by induction on n. For $n = 0$, the claim holds trivially by definition of the transition function π.

Let (M, w) and (N, v) be pointed Π-models that satisfy exactly the same Π-types of the modal depth $n + 1$. Let A be an automaton and δ the transition function of A. Call $q_n = A\big((M, w), n\big)$ and $q_{n+1} = A\big((M, w), n + 1\big)$. Let $\sigma_1, ..., \sigma_k$ enumerate the Π-types of the modal depth n and assume that

$$\tau_{(M,w),n+1} \;=\; \tau_{(M,w),n} \;\wedge \bigwedge_{i \,\in\, \{1,...,m\}} \Diamond\, \sigma_i \;\wedge \bigwedge_{i \,\in\, \{m+1,...,k\}} \neg\Diamond\, \sigma_i$$

Since (M, w) and (N, v) satisfy the same Π-type $\tau_{(M,w),n+1}$ of the depth $n + 1$, they also satisfy the same Π-type $\tau_{(M,w),n}$ of the depth n. By the induction hypothesis, we therefore conclude that $A\big((N, v), n\big) = q_n$. Also, since (M, w) and (N, v) satisfy the same type of the depth $n + 1$, the set of types of the depth n satisfied by the successors of w is the same as the set satisfied by the successors of v. That set is $\{\, \sigma_1, ..., \sigma_m \,\}$ in both cases. Therefore, by the induction hypothesis, the set of states defined by $succ(w)$ in round n is exactly the same as the set of states defined by $succ(v)$ in round n. Therefore the set of messages received by w in round $n + 1$ is exactly the same as the set of messages received by v in round $n + 1$. Therefore, since $A\big((N, v), n\big) = q_n$, we conclude that $A\big((N, v), n + 1\big) = q_{n+1}$, as required.

Let (M, w) and (N, v) be pointed Π-models and assume that $A\big((M, w), k\big) = A\big((N, v), k\big)$ for each $k \leq n + 1$ and each message passing automaton A whose set of proposition symbols is Π. Since this is true for an arbitrary automaton A of the vocabulary Π, this holds for any *type automaton* of the vocabulary Π. Hence (M, w) and (N, v) satisfy exactly the same Π-types of the depth $n + 1$.

We have now established equivalence of the conditions 1 and 2 above. We are ready to show that each class of pointed Π-models recognizable by an automaton can also be co-defined by a modal theory.

Let A be an arbitrary Π-automaton. Let $\mathcal{C}$ be the class of exactly all pointed Π-models accepted by A. Define $\mathcal{T}$ to be the collection of exactly all Π-types. Let Φ denote the set of exactly all Π-types $\tau \in \mathcal{T}$ such that for some n, the type τ is the Π-type of the depth n of some pointed Π-model (M, w), and furthermore, the automaton A accepts (M, w) in round n. Define the infinite disjunction $\bigvee \Phi$. We shall establish that for all pointed Π-models (M, w), we have $(M, w) \models \bigvee \Phi$ iff A accepts (M, w).

Assume that $(M, w) \models \bigvee \Phi$. Thus $(M, w) \models \tau_n$ for some type τ_n of the depth n of some pointed model (M', w') accepted by A in round n. Now (M, w) and (M', w') satisfy the same type τ_n, so by the equivalence of the conditions 1 and 2 above, (M, w) and (M', w') must both be accepted by A in round n.

Assume that (M, w) is accepted by the automaton A. The pointed model (M, w) is accepted in some round n, and thus the type of the depth n of (M, w) is one of the disjuncts of Φ. Therefore $(M, w) \models \bigvee \Phi$. The modal theory $\{\ \neg\tau \mid \tau \in \Phi\ \}$ co-defines the class $\mathcal{C}$ of pointed Π-models accepted by A. ◀

5 Expressivity and Decidability

In this section we very briefly investigate expressivity and decidability issues concerning MSC. We first investigate the *single variable fragment* MSC^1 of MSC. This fragment contains the programs Λ such that $|\mathrm{HEAD}(\Lambda)| = 1$. In the finite, the single variable fragment MSC^1 can simulate formulae of the μ-calculus of the type $\mu X.\varphi$, where φ is free of fixed point operators (see the proof of Proposition 7). Also, MSC^1 is not contained in MSO (proof of Proposition 6). It turns out that decidability and PSPACE-completeness of the satisfiability and finite satisfiability problems of MSC^1 follow rather trivially by the following delightful argument.

▶ **Proposition 5.** *The* SAT *and* FINSAT *problems for* MSC^1 *are* PSPACE*-complete.*

Proof. Let Λ be a program of MSC^1. Let X be the appointed head predicate symbol of Λ. (If Λ has no appointed symbol, Λ is not satisfiable.) We first check whether the formula X^0 is satisfiable by using a decision algorithm for ordinary modal logic. If not, we check whether the formula X^1 is satisfiable, again using a decision algorithm for ordinary modal logic. If not, we know that Λ is not satisfiable, for the following simple reason.

Let $(M, w) = ((W, R, V), w)$ be an arbitrary model of the same vocabulary as Λ. Let φ be the schema such that the iteration clause of Λ for X is $X :- \varphi$. Define the function $F : Pow(W) \rightarrow Pow(W)$ such that $F(U) = \{\ u \in W \mid \big((W, R, V[X \mapsto U]), u\big) \models \varphi\ \}$. Since $||X^0||^M = ||X^1||^M = \emptyset$, we observe that $F(\emptyset) = \emptyset$. Since $||X^{n+1}||^M = F(||X^n||^M)$ for all $n \in \mathbb{N}$, we conclude that no formula X^k is satisfied by any node of M.

The claim of the current proposition now follows from the PSPACE-completeness of ordinary modal logic. ◀

We leave the question of decidability of MSC open at this stage, and sketch some proofs concerning expressivity instead. The μ-calculus (μML) is a bisimulation invariant logic that expands modal logic with a recursion mechanism based on least and greatest fixed point operators μX and νX. For the semantics and basic properties of μML, see [3].

▶ **Proposition 6.** $\mathrm{MSC}^1 \not\leq \mu\mathrm{ML}$. *This holds already in the finite.*

Proof Sketch. (Note that we only sketch a proof of this proposition.) Define a program Λ of MSC^1 which is true in (M, w) iff the following conditions hold.

1. There exists some $n \in \mathbb{N}$ such that there is a directed path of the length n from w to a point v without successors. We call v a dead-end.

2. There are no directed paths shorter than n from w to a dead-end, and each directed path of the length n originating from w ends in a dead-end.

If a pointed model (M, w) satisfies the above property, with n being the unique distance to a dead-end, we say that (M, w) has the n-path property.

Define $X^0 :- \ \Box\bot$ and $X :- \ \Diamond X \wedge \Box X$. It is easy to show by induction on n that for all pointed models (M, w), the model (M, w) satisfies the n-path property iff $(M, w) \models X^n$.

If a pointed model has the n-path property for some $n \in \mathbb{N}$, we say that (M, w) has the centre point property. The class of pointed models with the centre point property is not definable by any formula of μML. This is shown by establishing that there exists no formula $\varphi(x)$ of MSO such that $M \models \varphi(w)$ iff (M, w) has the centre point property. The claim that $\mathrm{MSC}^1 \not\leq \mu\mathrm{ML}$ in the finite then follows, as it is well known that $\mu\mathrm{ML} < \mathrm{MSO}$.

Assume, for the sake of contradiction, that there exists a formula $\varphi(x)$ of MSO that defines the centre point property. Therefore MSO can define the corresponding property in restriction to the class of rooted finite ranked trees with two successor relations. By the pumping lemma for tree languages it is then trivial to establish that this is a contradiction. ◀

Alternation of μ and ν-operators is a tricky issue in μ-calculus, and alternation hierarchies have been defined in various ways. We define Σ_1^μ to be the fragment of μ-calculus without ν-operators and with negations on the atomic level, i.e., the language built from literals with $\wedge, \vee, \Diamond$ and $\Box$, and μX when X occurs only positively in the scope of μX. We define Π_1^μ analogously.

▶ **Proposition 7.** $\Sigma_1^\mu < \mathrm{MSC}$ *in the finite.*

Proof Sketch. (Note that we only sketch the proof of this proposition.) It is folklore that μ-calculus can be defined with or without the capacity of using simultaneous fixed points, without change in expressive power. There are translations both ways, from standard μ-calculus into one with simultaneous fixed points and back. It is also folklore that μ-calculus can be defined in terms of modal equation systems (see [3]). For instance, a formula $\mu X.\psi\big(X,\ \mu Y.\varphi(X,Y)\big)$ translates to the *equation block*

$$\begin{aligned} X &:- \ \psi(X,Y) \\ Y &:- \ \varphi(X,Y), \end{aligned}$$

where $\psi(X,Y)$ is the formula obtained from the formula $\psi\big(X,\ \mu Y.\varphi(X,Y)\big)$ by replacing the subformula $\mu Y.\varphi(X,Y)$ by the variable Y. For a more concrete example, the formula $\mu X.\big(\Box X \vee \mu Y.(p \vee \Diamond(Y \vee X))\big)$ translates to the block

$$\begin{aligned} X &:- \ \Box X \vee Y \\ Y &:- \ p \vee \Diamond(Y \vee X). \end{aligned}$$

If $M = (W, R)$ is a model, the block

$$\begin{aligned} X &:- \ \psi(X,Y) \\ Y &:- \ \varphi(X,Y) \end{aligned}$$

defines a monotone function $F : (Pow(W))^2 \longrightarrow (Pow(W))^2$ such that

$$F(U,V) = \big(||\psi(U,V)||^M, ||\varphi(U,V)||^M\big).$$

The least fixed point $F^\infty(\emptyset,\emptyset)$ of this monotone operator is a pair (X^∞, Y^∞) such that $X^\infty = ||\mu X.\,\psi\big(X,\ \mu Y.\varphi(X,Y)\big)||^M$.

An arbitrary formula φ of Σ_1^μ translates into an equation block with a finite number of equations. We may assume that φ is of the form $\mu X.\psi$. (If not, we may use a dummy variable X.) The set X^∞ is then exactly the set $||\varphi||^M$. The very same equation block also defines a program Λ_φ of MSC, with the terminal clause corresponding to each variable Z being $Z(0) :- \bot$, and the set of appointed variables being $\{X\}$. Now, Λ_φ is true in a finite model M exactly in the nodes belonging to X^∞. This follows immediately, since there is a finite number $n \in \mathbb{N}$ such that $F_M^n(\emptyset, ..., \emptyset) = F_M^{n+1}(\emptyset, ..., \emptyset)$, i.e., the *closure ordinal* of F is finite. Therefore, for all $w \in W$, we have $w \in X^\infty$ iff there is some $n \in \mathbb{N}$ such that we have $(M, w) \models X^n$ for the appointed variable X of Λ.

The strictness of the inclusion $\Sigma_1^\mu <$ MSC in the finite follows by Proposition 6. ◀

▶ **Proposition 8.** $\Pi_1^\mu \not\leq$ MSC. *This holds already in the finite.*

Proof. MSC cannot define non-reachability: there exists no program of MSC true in exactly those pointed models (M, w) where there is no directed path to, say, a point v without successors, i.e., a dead-end. Assume that such a program exists. Run it in a *directed successor ring*, i.e, a connected finite model (W, R), where R is a binary relation and where both the out-degree and in-degree of each node is one. Let w be a node of the ring. If non-reachability is definable, there is an automaton A such that if we run it on the ring, it accepts w in some finite number n of rounds. However, let (N, S) be a finite model, where S is a successor ordering of N and $|N| \geq n + 10$. Let u be the least element of N with respect to S. It is straightforward to show that in the n-th round of running A, the state of A at w is exactly the same as at u. Therefore A accepts $((N, S), u)$, which is a contradiction.

The formula $\nu X.(\Diamond\top \wedge \Box X)$ states that a deadend cannot be reached from the point of observation. ◀

Finally, it is worth noting that the model checking problem for MSC is clearly decidable, as the sequence of global configurations defined by an FMPA and a finite model must eventually loop. With an MPA it is possible to recognize, with respect to the class of finite pointed Kripke models, even undecidable classes of models.

References

1 C. Areces and B. ten Cate. Hybrid Logics. In P. Blackburn, F. Wolter and J. van Benthem (eds.), *Handbook of Modal Logic*, Elsevier, 2006.

2 P. Blackburn, M. de Rijke and Y. Venema. *Modal Logic*. Cambridge University Press, 2011.

3 J. Bradfield and C. Stirling. Modal mu-calculi. In P. Blackburn, J. van Benthem and F. Wolter (eds.), *The Handbook of Modal Logic*, Elsevier, 2006.

4 H. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer, 2nd edition, 2005.

5 P. Fraigniaud, M. Göös, A. Korman and Jukka Suomela. What can be decided locally without identifiers? arXiv:1302.2570, 2013.

6 N. Immerman. *Descriptive Complexity*. Springer, 1999.

7 L. Hella, M. Järvisalo, A. Kuusisto, J. Laurinharju, T. Lempiäinen, K. Luosto, J. Suomela and J. Virtema. Weak models of distributed computing, with connections to modal logic. In *Proc. 31st ACM Symposium on Principles of Distributed Computing* (PODC), 2012.

8 L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.

9 N. Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193-201, 1992.

10 D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

11 J. Suomela. Survey of local algorithms. *ACM Computing Surveys* 45, 2013.

A Appendix

A.1 Addenda to the proof of Theorem 2

Argument for the special case where $n = 0$ and $mdt(\Lambda) = 0$

Recall the definition of the sets T, T' and U in the definition of π on initial inputs $P \subseteq \Pi$ for the case where $mdt(\Lambda) = 0$. Let V be the valuation of M. Call $\Phi = \mathrm{SUBF}(\Lambda) \cap \Pi$. By the definition π, we have $(M, v) \models p$ iff $p \in \pi\big(\{\, p \in \Pi \mid v \in V(p) \,\}\big)$ for each $p \in \Phi$, and therefore, the equivalence $(M, v) \models \varphi$ iff $\varphi \in T$ holds for each *atomic formula* $\varphi \in \mathrm{ATOM}(\Lambda)$. Hence, since every formula $\varphi \in \mathrm{SUBF}(\Lambda)$ of the modal depth 0 is a Boolean combination of formulae in $\mathrm{ATOM}(\Lambda)$, we have $(M, v) \models \varphi$ iff $\varphi \in T$ for all *formulae* $\varphi \in \mathrm{SUBF}(\Lambda)$ of the modal depth 0. Hence we have $(M, v) \models X^0$ iff $X \in T'$ for all schema variable symbols $X \in \mathrm{HEAD}(\Lambda)$, and also $(M, v) \models \varphi$ iff $\varphi \in T'$ for all atomic formulae $\varphi \in \mathrm{ATOM}(\Lambda)$. Therefore, since every *schema* $\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0 is a Boolean combination of formulae in $\mathrm{ATOM}(\Lambda)$ and head predicate symbols in $\mathrm{HEAD}(\Lambda)$, the equivalence $(M, v) \models \varphi^0$ iff $\varphi \in Q_v(0, 0)$ holds for all schemata $\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0, as required.

$(M, v) \models \psi \;\Leftrightarrow\; \psi \in Q_v(-1, mdt(\Lambda) - 1)$

We have $(M, v) \models \varphi$ iff $\varphi \in Q_v(-1, 0)$ for all $v \in W$ and all $\varphi \in \mathrm{SUBF}(\Lambda)$ of the modal depth 0, so the base step of the induction is clear. Let $k \in \mathbb{N}$ such that $k \leq mdt(\Lambda) - 2$, and assume that the equivalence $(M, v) \models \psi$ iff $\psi \in Q_v(-1, k)$ holds for each $v \in W$ and each $\psi \in \mathrm{SUBF}(\Lambda)$ such that $md(\psi) \leq k$. Let $v \in W$ and let $\varphi \in \mathrm{SUBF}(\Lambda)$ be a formula of the modal depth $k + 1$. We must show that $(M, v) \models \varphi$ iff $\varphi \in Q_v(-1, k + 1)$. The formula φ is a Boolean combination of formulae $\Diamond\psi$, where $md(\psi) \leq k$. It therefore suffices to show that for each such formula $\Diamond\psi$, we have $(M, v) \models \Diamond\psi$ iff $\Diamond\psi \in Q_v(-1, k + 1)$.

Assume first that $(M, v) \models \Diamond\psi$. Therefore some successor u of v satisfies $(M, u) \models \psi$. By the induction hypothesis, $\psi \in Q_u(-1, k)$. Hence, in round $k + 1$, the automaton A at u sends a message L such that $\psi \in L$ to the predecessors of u. Thus $\Diamond\psi \in Q_v(-1, k + 1)$.

Assume then that $\Diamond\psi \in Q_v(-1, k + 1)$. Therefore the automaton A_Λ at node v receives a message L such that $\psi \in L$ from some successor u in round $k + 1$. Therefore $\psi \in Q_u(-1, k)$. By the induction hypothesis, $(M, u) \models \psi$. Therefore $(M, v) \models \Diamond\psi$.

$(M, v) \models \varphi^0 \;\Leftrightarrow\; \varphi \in Q_v\big(0, 0\big)$

Recall the definition of the sets T, T' and U in the definition of δ on input states (S, m, f) in the case where $m = mdt(\Lambda) - 1$. We shall first show that $(M, v) \models \varphi$ iff $\varphi \in T$ holds for each $v \in W$ and each formula $\varphi \in \mathrm{SUBF}(\Lambda)$ such that $md(\varphi) \leq mdt(\Lambda)$.

Let $v \in W$. Let $\varphi \in \mathrm{SUBF}(\Lambda)$ be a formula such that $md(\varphi) \leq mdt(\Lambda)$. The formula φ is a Boolean combination of formulae $\Diamond\psi$, where $md(\psi) < mdt(\Lambda)$. By the definition of T, it suffices to show that for each such formula $\Diamond\psi$, we have $(M, v) \models \Diamond\psi$ iff $\Diamond\psi \in T$.

Assume first that $(M, v) \models \Diamond\psi$. Therefore some successor u of v satisfies $(M, u) \models \psi$. Therefore, since $md(\psi) < mdt(\Lambda)$, we know that $\psi \in Q_u\big(-1, mdt(\Lambda) - 1\big)$. Thus the automaton A_Λ at u sends a message L such that $\psi \in L$ to its predecessors in round $mdt(\Lambda)$. Thus $\Diamond\psi \in T$.

Conversely, assume that $\Diamond\psi \in T$. Therefore v receives a message L such that $\psi \in L$ from some successor u in round $mdt(\Lambda)$. Hence we have $\psi \in Q_u\big(-1, mdi(\Lambda) - 1\big)$. Therefore we know that $(M, u) \models \psi$. Thus $(M, v) \models \Diamond\psi$.

We have now established that $(M, v) \models \varphi \;\Leftrightarrow\; \varphi \in T$ for all $v \in W$ and all formulae $\varphi \in \mathrm{SUBF}(\Lambda)$ of the modal depth up to $mdt(\Lambda)$. Thus $(M, v) \models X^0 \;\Leftrightarrow\; X \in T'$ for all head predicate symbols $X \in \mathrm{HEAD}(\Lambda)$, and also $(M, v) \models \varphi$ iff $\varphi \in T'$ for all atomic formulae $\varphi \in \mathrm{ATOM}(\Lambda)$. Therefore, since every *schema* $\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0 is a Boolean combination of formulae in $\varphi \in \mathrm{ATOM}(\Lambda)$ and head predicate symbols in $\mathrm{HEAD}(\Lambda)$, we have $(M, v) \models \varphi^0$ iff $\varphi \in Q_v\big(0, 0\big)$ for all $v \in W$ and all schemata in $\varphi \in \mathrm{SUBS}(\Lambda)$ of the modal depth 0, as required. This concludes the base case of our argument by induction on n.

$(M, v) \models \Diamond\psi^n \;\Leftrightarrow\; \Diamond\psi \in T$

Assume first that $(M, v) \models \Diamond\psi^n$. Therefore some successor u of v satisfies $(M, u) \models \psi^n$. Hence, since $md(\psi) < mdi(\Lambda)$, we know that $\psi \in Q_u\big(n, mdi(\Lambda) - 1\big)$. Therefore the automaton A_Λ at u sends a message L such that $\psi \in L$ to its predecessors in round $g(n + 1, 0)$. Thus $\Diamond\psi \in T$.

Conversely, assume that $\Diamond\psi \in T$. Therefore v receives a message L such that $\psi \in L$ from some successor u in round $g(n + 1, 0)$. Hence, $\psi \in Q_u\big(n, mdi(\Lambda) - 1\big)$. Therefore we know that $(M, u) \models \psi^n$. Therefore $(M, v) \models \Diamond\psi^n$.