Temporal reasoning about messages

R. Ramanujam

The Institute of Mathematical Sciences, Chennai, India

jam@imsc.res.in

http://www.imsc.res.in/~jam

You have got new mail

Every time
Every time the spray of the drizzle
chases me underneath an awning
and drenches my face
Every time I pull off my gajra
and a few flowers obstinately
stick in my hair
Every such time
I receive a message from you

A Tamil poem circa 9th cent AD.

Protocols

- Sequences of idealized communications.
- Usually to achieve a specific goal.
- A finite set of message types.
- Rules specify:
  - When a sender may send a message of a particular type,
  - What a receiver should do on receipt of a message.
- Communications travel on public channels.

Semantic issues

Assigning meaning to messages can get tricky, since the meaning of a message may well vary:

- depending on who sends it, who receives it;
- depending on the state at which it was sent / received;
- by use of cryptographic primitives;

and so on.
In synchronized systems, absence of messages may reveal information as well.
Crucially, the protocol determines the meaning(s) of a message.

Composing messages

- Protocol design: typically, constraints on the communication medium are specified, a set of desired goal states is given, and a sequence of communications is to be provided.
- If we had Hoare-style rules, $\{P\}m\{Q\}$, we could compose them.
- Typically the predicates P and Q above constrain not only what is true of the system, but also the *knowledge* of agents in the system.
- Security protocols specify negated knowledge (ignorance) assertions as well.
- It would be nice to have such clean logical constructions.

Messages as strategies

- It is natural to conceive of protocols as non-zero sum games of partial information between agents.
- Agents exercise a strategic choice when they decide on what message to send when.
- These are *large* games, in the sense of the game arena having nontrivial structure.
- Standard solution concepts of game theory, developed for small (normal form) games are not directly applicable.
- A natural framework for modelling interactions on the Web.
- Many interesting questions, and a few answers.

Security theory

Security considerations complicate analysis of protocols considerably.

- We have only probabilistic assertions.
- The presence of a hypothetical adversary with unbounded computational power typically implies undecidability of analysis.
- Combining protocols developed to meet different security objectives can easily lead to inconsistency.
- Information flow analysis is usually quite nontrivial.

Much room for logic: designing one at the "right" level of abstraction, without missing the tricky details but yet allowing decision procedures, is challenging.

Reasoning about messages

- In theory of distributed systems, messages are typically identified with their content.
- Determining the content depends on intended applications.
- In algorithms for coordination, message gets equated with the sender's local state when the message is sent (or a transition in the automaton).
- In security theory, patterns in messages are important, often more than the actual content.
- In distributed games, messages constitute (partial) information about players' strategies.

What kind of logic ?

- Logic can play several rôles in the contexts we have been discussing, but we focus on two:
  - Requirement specification.
  - Verification.
- Then a natural question is whether messages need to figure as a syntactic category, at all.
- The second criterion above dictates decidability of the satisfiability problem, and efficient truth checking.

Temporal Logic

- Linear time temporal logic plays a similar rôle in the context of *reactive systems*.

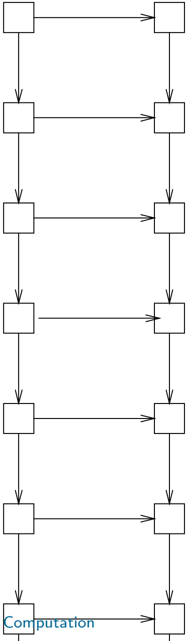
$$\Phi ::= p \in P \mid \neg\, \alpha \mid \alpha_1 \vee \alpha_2 \mid \bigcirc\, \alpha \mid \alpha_1\ \mathbf{U}\ \alpha_2$$

- Note that model elements are abstracted away, and relative ordering of events in computation is important.
- We are guided by such an approach in what follows.
- The logics we discuss are extensions of linear time temporal logics and message behaviour is abstracted in terms of how they affect causal ordering.

Lamport Diagrams

- Partial orders depicting computations of systems of communicating automata.
- The model is that of a system of n agents that communicate by asynchronous message passing.
- The computation of each agent is given as a linear order of event occurrences.
- Causal ordering between i-events and j-events, for distinct agents i and j, is introduced by communication.
- Note that there are implicit unbounded buffers, leading to nonregular behaviour (in terms of sequentializations).

An example



The definition

- Let [n] denote the set $\{1, \ldots, n\}$, the set of n agents.
- $D = (E_1, \ldots, E_n, \leq)$, where:
  E_i is the set of event occurrences of agent i,
  $\leq$ is a partial order on $E \stackrel{\text{def}}{=} \bigcup_i E_i$ called the causality relation such that:
  1. for all $i \in [n]$, E_i is totally ordered by $\leq$,
  2. for all $e \in E$, $\downarrow e \stackrel{\text{def}}{=} \{e' \mid e' \leq e\}$ is finite.
- Let $e \in E_i$. $\downarrow e$ is the local state of agent i when the event e has just occurred.
- $LC_i \stackrel{\text{def}}{=} \{\epsilon_i\} \cup \{\downarrow e \mid e \in E_i\}$.
  $LC \stackrel{\text{def}}{=} \bigcup_i LC_i$.

Remarks

- $e_1 \leq e_2$ denotes that in any computation when e_2 has occurred, the event e_1 has occurred earlier.
- Let $\lessdot$ denote the covering relation of the partial order.
- When $e_1 \in E_i$, $e_2 \in E_j$, $i \neq j$ and $e_1 \lessdot e_2$, we can read e_1 as the sending of a message from i to j, and e_2 as its receipt; we denote this by $e_1 <_c e_2$.
- Note that under this reading, there is an implicit *FIFO assumption*: messages are delivered in the same order as they were sent.
- More general notions of information transfer are consistent.

The temporal logic m-LTL

- **Syntax**: $(P_1, P_2, \ldots, P_n)$; $P \stackrel{\text{def}}{=} \bigcup_i P_i$.
- i-local formulas:

$$\Phi_i ::= p \in P_i \mid \neg\, \alpha \mid \alpha_1 \vee \alpha_2 \mid \bigcirc\, \alpha \mid \alpha_1 \,\mathbf{U}\, \alpha_2$$

$$\mid \oslash_j\, \alpha,\ j \neq i,\ \alpha \in \Phi_j$$

- Global formulas:

$$\Psi ::= \alpha @ i,\ \alpha \in \Phi_i \mid \neg\, \psi \mid \psi_1 \vee \psi_2$$

- $\oslash_j \alpha$, asserted by i, says that α held in the last j-local state visible to i.

Semantics: local formulas

$M = (D, V)$, where $D = (E_1, \ldots, E_n, \leq)$ is a Lamport diagram such that $E = \bigcup_i E_i$ is a countably infinite set and

$V : LC \rightarrow 2^P$ is the valuation map such that for $d \in LC_i$, $V(d) \subseteq P_i$.

- $M, d \models_i \bigcirc\alpha$ iff there exists $d' \in LC_i$ such that $d \lessdot d'$ and $M, d' \models_i \alpha$.
- $M, d \models_i \alpha \mathbf{U} \beta$ iff $\exists d' \in LC_i$: $d \subseteq d', M, d' \models_i \beta$ and $\forall d'' \in LC_i : d \subseteq d'' \subset d' : M, d'' \models_i \alpha$.
- $M, d \models_i \oslash_j \alpha$ iff there exists $d' \in LC_j$ such that $d' \lessdot d$ and $M, d' \models_j \alpha$.

Note that the future modalities are local.

Semantics: global formulas

The global formulas are simply boolean combinations of local formulas.

- $M \models \alpha @ i$ iff $M, \epsilon_i \models_i \alpha$.
- $M \models \neg\psi$ iff $M \not\models \psi$.
- $M \models \psi_1 \vee \psi_2$ iff $M \models \psi_1$ or $M \models \psi_2$.

There are no global modalities in the logic.

Examples

- $([\,](p \wedge \oslash_2 \neg\ \text{`}OK\text{'} \implies \bigcirc(q \implies \oslash_2\ \text{`}OK\text{'})))@1$.
  agent 1 can make a transition from a state satisfying p into a state in which q holds only after hearing an '*OK*' from agent 2, and must block otherwise.
- $(p@3 \wedge ([\,]\oslash_3 p)@1) \implies ([\,]\oslash_3 p)@2$.
  Any information about agent 3 that agent 2 gets (regarding p) is communicated through agent 1.

Decidability

Let ψ be a m-LTL formula of length m.

- **Theorem**: Satisfiability of ψ over n-agent Lamport diagrams can be decided in time $2^{O(mn)}$.
- Proof is by construction of a system of n communicating automata (n-SCA), which runs on n-agent Lamport diagrams.
- Given an n-SCA S, we can define the verification problem: $S \models \psi$ iff $\mathcal{L}(S) \subseteq \mathcal{L}(\psi)$.
- **Theorem**: The verification question $S \models \psi$ can be answered in time $k \cdot 2^{O(mn)}$, where $k = |S|$.

Strengthening

- m-LTL is quite a weak logic, as temporal logics go.
- We can strengthen the past: alongwith $\oslash_j \alpha$, we can add $\alpha \mathbf{S}_j \beta$, without disturbing the results.
- We can consider global future: $\bigcirc_j$ and $\alpha \mathbf{U}_j \beta$ modalities, as well as modalities in global formulas.
- These lead to undecidability.

First order logic

m-LTL is a fragment of a natural first order logic F on n-agent Lamport diagrams.

$$\Gamma ::= P_i(x) \mid P_a(x) \mid S(x,y) \mid x < y \mid \neg\, \phi \mid \phi_1 \ \vee\ \phi_2 \mid \exists x.\phi,$$

where $i \in [n]$, $a \in \Sigma$.

- The satisfiability problem for F is undecidable.
- More interestingly, even the *two-variable fragment* is undecidable.
- The $FO(S)$ sub-logic is also undeciable.

The problem

- The main culprit is the existence of *unbounded buffers*.
- Since a sender can proceed asynchronously and pile up messages which a receiver may look at arbitrarily later, we have excessive computational power.
- One natural solution is to place bounds on channel capacity.
- Over such systems, we can not only get decidability, but can also do interesting automata theory.

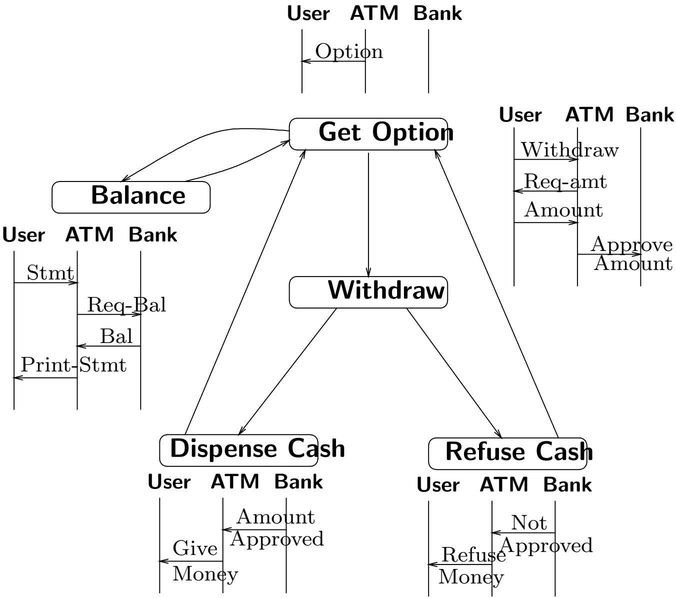
Generating infinite LDs

- Typically, we wish to reason not about individual messages but about message patterns.
- Message sequence charts: a standard grphical notation for describing system requirements in the design of communication protocols.
- MSCs are very similar to LDs: a graph rather than the Hasse diagram of a partial order.
- Infinite behaviours are obtained by concatenating a fixed number of MSCs in some periodic manner.

An ATM example

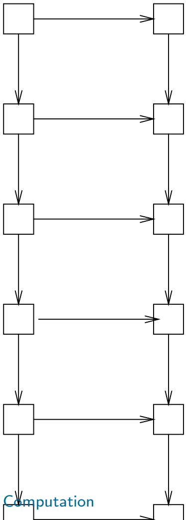
- The communication scenario of an automatic teller machine (ATM):
- There are three agents—User, Bank and the ATM.
- The ATM provides options for the user to check the balance in his/her account and to withdraw cash after validating the balance.

The ATM example

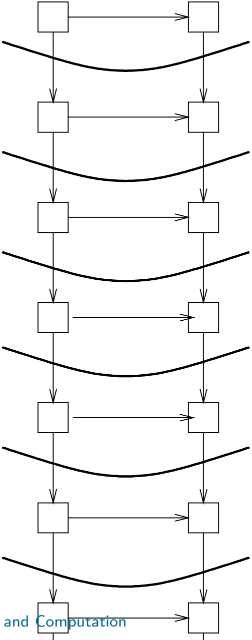


Layered Diagrams

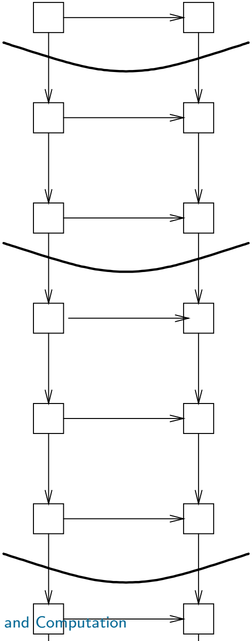
Following this intuition, we define here a class of Layered Lamport Diagrams (LLDs), in such a way that every LD can be thought of as a concatenation of finite layers.



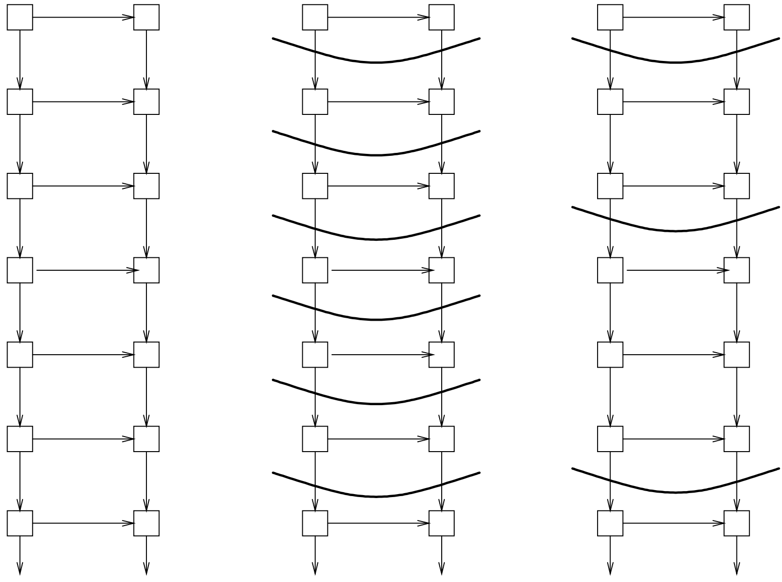
Layering



Unbounded layering



Layered diagrams



Definition

A **layered Lamport diagram** is a tuple $D = (E, \leq, \phi, \lambda)$ where $(E, \leq, \phi)$ is a Lamport diagram and $\lambda : E \rightarrow \mathbf{N}$ is a layering function which satisfies the following conditions:

- for all $e \in E$, if $\lambda(e) = k$ then, for all $i \in [n]$, there exists $e' \in E_i$ such that $\lambda(e') = k$.
- for $e, e' \in E$, $e \leq e'$ implies $\lambda(e) \leq_{\mathbf{N}} \lambda(e')$.
- for every k, $\lambda^{-1}(k)$ is finite.

Thus a layer is a finite set of events that includes at least one event of each agent, and the layering respects the causality relation.

The layers

- Given a countable layered Lamport diagram $D = (E, \leq, \phi, \lambda)$, $\lambda(E)$ is an infinite set and can be denoted by an increasing (infinite) sequence of natural numbers.
- More precisely, let ν_D denote the sequence of natural numbers $k_1, k_2, \ldots$ such that $\lambda(E) = \{k_1, k_2, \ldots\}$ and $k_1 <_{\mathbf{N}} k_2 <_{\mathbf{N}} k_3 \ldots$. For $i, j \in \lambda(E)$, we say that j is a *successor of* i iff there exists l such that $k_l = i$ and $k_{l+1} = j$ in ν_D.
- For $k \in \lambda(E)$, $\lambda^{-1}(k)$ induces a (finite) Lamport diagram which we call a layer of D and denote by D_k. Note that the sequence of layers $D_{k_1}, D_{k_2}, \ldots$, where $\nu_D = k_1, k_2 \ldots$, completely specifies the diagram D.

Natural conditions on layering

Consider a layered Lamport diagram $D = (E, \leq, \phi, \lambda)$. Let $delay_D = \{\lambda(e') - \lambda(e) \mid e <_c e'\}$ denote the set of *communication delays* associated with D.

1. D is said to be **communication-closed** if for every $e, e' \in E$ such that $e <_c e'$, $\lambda(e) = \lambda(e')$.
2. Let $b > 0$. D is said to be b-**bounded**, if for all $k \in \lambda(E)$, $|\lambda^{-1}(k)| \leq b$. D is said to be **bounded** if there exists $b \in \mathbf{N}$ such that D is b-bounded.
3. Let $b > 0$. D is said to be **strongly** b-**bounded**, if D is b-bounded and for all $k \in delay_D$, $k \leq b$.

The temporal logic λ-**LTL**

Syntax: $(P_1, P_2, \ldots, P_n)$; $P \stackrel{\mathrm{def}}{=} \bigcup_i P_i$. Γ, a finite set of *layer names*.

Layer formulas:

$$\Phi_1 ::= p \in P_i \mid \tau_i \mid \neg\, \alpha \mid \alpha_1 \vee \alpha_2$$
$$X\, \alpha \mid Y\, \alpha \mid F\, \alpha \mid P\, \alpha$$

Temporal formulas:

$$\Psi ::= \alpha @ i,\ \alpha \in \Phi_1,\ i \in [n] \mid a,\ a \in \Gamma$$
$$\neg\, \phi \mid \phi_1 \vee \phi_2 \mid \bigcirc\, \phi \mid \phi_1\, \mathbf{U}\, \phi_2$$

This is the same as the standard propositional temporal logic of linear time, but built up from layer formulas and layer propositions.

Semantics: layer formulas

Models are $M = (D, V_E, V_\lambda)$, where $D = (E, \leq, \phi, \lambda)$ is a layered Lamport diagram, $V_E : E \to 2^P$ and $V_\lambda : \lambda(E) \to \Gamma$.

Let $\alpha \in \Phi_l$ and $e \in E$. The notion that α holds at e in M is denoted $M, e \models_l \alpha$ and is defined inductively as follows:

- $M, e \models_l p$ iff $p \in V_E(e)$.
- $M, e \models_l \tau_i$ iff $\tau_i \in V_E(e)$ and $\phi(e) = i$.
- $M, e \models_l X\alpha$ iff there exists $e' \in E$ such that $e \lessdot e'$ and $M, e' \models_l \alpha$.
- $M, e \models_l F\alpha$ iff there exists $e' \in E$ such that $e \leq e'$, $\lambda(e) = \lambda(e')$ and $M, e' \models_l \alpha$.
- $M, e \models_l Y\alpha$ iff there exists $e' \in E$ such that $e' \lessdot e$ and $M, e' \models_l \alpha$.
- $M, e \models_l P\alpha$ iff there exists $e' \in E$ such that $e' \leq e$, $\lambda(e) = \lambda(e')$ and $M, e' \models_l \alpha$.

Semantics: temporal formulas

Temporal formulas are interpreted at layers of a layered Lamport diagram.
Given a model $M = (D, V_E, V_\lambda)$ and $\phi \in \Psi$, the notion that ϕ holds in the layer k of D is denoted $M, k \models \phi$ and is defined inductively as follows:

- $M, k \models \alpha@i$ iff $M, e \models_I \alpha$ where e is the i-minimum event of D_k.
- $M, k \models a,\ a \in \Gamma$ iff $V_\lambda(k) = a$.
- $M, k \models \bigcirc\phi$ iff $M, k' \models \phi$ where k' is the successor of k in ν_D.
- $M, k \models \phi \mathbf{U} \psi$ iff there exists $k' \in I(E)$: $k \leq_{\mathbf{N}} k', M, k' \models \psi$ and for all $k'' \in I(E) : k \leq_{\mathbf{N}} k'' <_{\mathbf{N}} k' : M, k'' \models \phi$.

Satisfiability

Note that several notions of satisfiability are relevant now.

- C-satisfiability (over models based on communication closed layers),
- B-satisfiability (over models based on bounded LLDs),
- S_b-satisfiability (over strongly b-bounded LLDs),
- C_b-satisfiability.

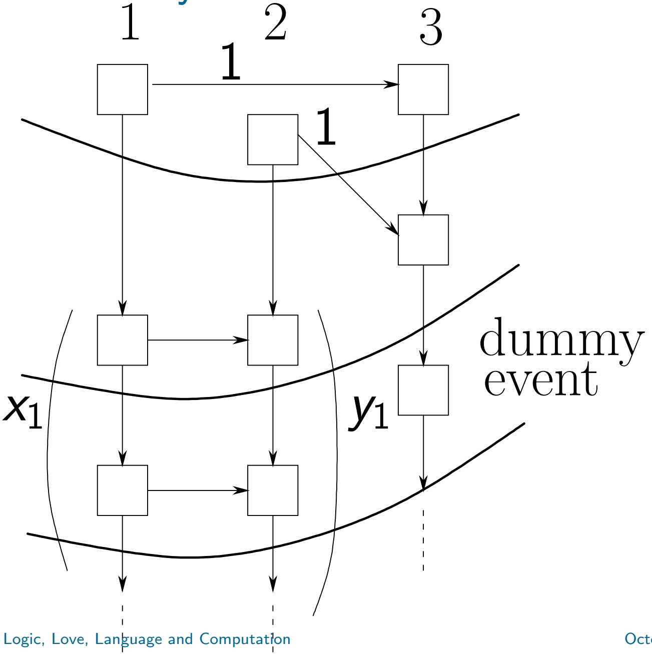
Results

Theorem

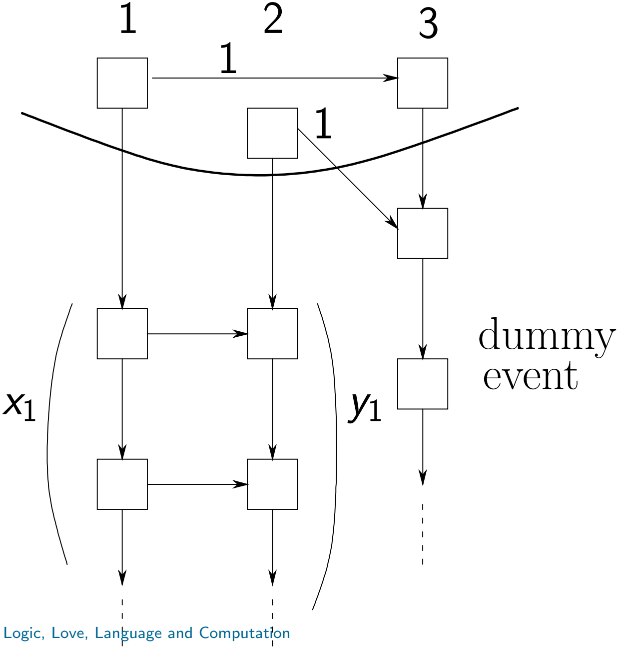
B-satisfiability and C-satisfiability are undecidable, whereas, for $b > 0$, C_b-satisfiability and S_b-satisfiability are decidable in Exptime.

- The negative results here mainly stem from the fact that instances of the Post Correspondence Problem (PCP) can be described easily using Lamport Diagrams.
- The fact that B-satisfiability is undecidable is a little surprising, considering that the layers are uniformly bounded.
- The decidability of S_b-satisfiability, is technically nontrivial.

PCP: B-satisfiability



PCP: C-satisfiability

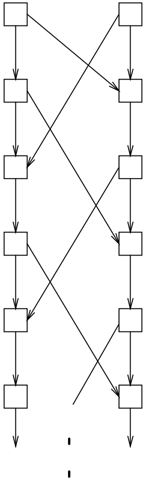


The decidable cases

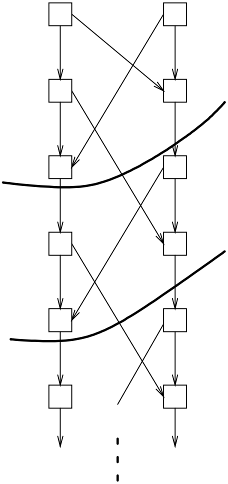
- Observe that satisfiability of layer formulas within bounded communication closed layers is decidable.
- Now, given a temporal formula ϕ_0 we can construct the formula automaton $\mathcal{A}_0$ in the standard manner to decide C_b-satisfiability.
- When it comes to S_b-satisfiability, the automaton construction gets considerably more complicated.
- We can no longer check for satisfiability of layer formulas within b-bounded layers; to satisfy a formula of the form $X\alpha$, we may need to consider subsequent layers, and similarly previous layers need to be remembered for formulas of the form $Y\alpha$.

MSGs are not enough

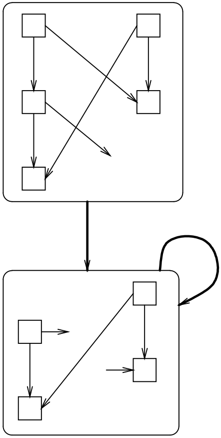
The Alternating Bit protocol.



Lamport diagram



Bounded layering



CMSG representation

Fragments of LDs

- MSG where nodes are not labelled with LDs but with fragments, having "sticky ends".
- Under FIFO assumption, there is a well-defined notion of concatenation of fragments.
- Infinite diagrams are generated by tiling such fragments.
- Protocols like Alternating Bit can be modelled thus.

Bounded channel capacity

- When we consider only channel bounded diagrams, good automata theory is possible.
- The monadic second order theory, and corresponding message passing automata are well studied.
- Interesting results exist on *realizability* of MSG specifications by systems of communicating automata.
- *Blaise Genest*, 2004, has a nice survey.

Message overtaking

- The undecidability proofs crucially use FIFO assumption on channels.
- When the medium can re-order messages, the logics are less expressive.
- $FO(S, <)$ remains undecidable, but the two-variable fragment is now decidable.
- The proof is by reduction to emptiness of multi-counter automata, for which the complexity is very high (not known to be elementary).

Unboundedly many processes

- In "real world" MSCs, the number of agents is not fixed.
- In many applications, we have a dynamic network of processes, which processes join and leave at will.
- *Web services* are an especially interesting instance.
- There are many applications where temporal properties depend on the *number* of live processes.
- Natural extensions to such systems are typically undecidable.

Unbounded clients

- We consider a model of n communicating servers, handling an unbounded number of clients.
- Clients are very simple: a client sends a request to a server, and waits for an answer. On receiving the answer, the client exits.
- We extend m-LTL with formulas of the form $\exists x.\phi(x)$, where $\phi(x)$ is a boolean formula over monadic predicates of the form $p(x)$.
- The language includes $x = y$ as well; thus we can speak of the number of live clients.
- The satisfiability and verification problem are shown to be decidable.

Security

- In security theory, we have an *unbounded message alphabet*, generated by an algebraic structure.
- The set of all message terms is given by the structure:

$$t \in T := t_0 \in T_0 \mid (t_1, t_2) \mid \{t\}_k$$

- Message *generation rules* govern when a message can be sent by a principal, as well as what a receiver learns from the message.
- An all-powerful intruder is assumed, who can monitor the network and eavesdrop on messages.
- The model is easier: every send / receive is an instantaneous communication with the intruder.
- Even simple reachability questions are undecidable, and we obtain decidability results for the interesting subclass of tagged protocols.

Relation to knowledge theory

- Clearly, every message m can be seen as a view transformer: it can be described as a set of pairs of the form $(K_i\alpha, K_i\beta)$, where if the receiver i knows A before receiving the message, then i knows B before receiving it.
- There is a natural notion of knowledge here: given a Lamport diagram D, and configurations c, c' (down-closed subsets of E), $c \sim_i c'$ iff $c \cap E_i = c' \cap E_i$.
- Thus we can give the semantics of messages as sets of pairs as above, where the formulas come from an epistemic temporal logic.
- Ths gives a specific kind of *update semantics*.
- Relating such semantics to automata theoretic techniques seems difficult.

In conclusion

- Formal reasoning about messages is logically interesting, has a wide range of applications in computer science but computationally difficult.
- Many foundational questions remain: what is the algebra of LDs ? What are star-free collections ? Where does m-LTL fit in ?
- We have abstracted away all data in messages; this is not reasonable.
- MSCs typically involve time-outs; bringing in clocks and clock values again complicates matters.
- At present, we are very far from a theory that is general, nice as well as tractable.

Joint work

- on Lamport diagrams: with *B. Meenakshi*.
- on unbounded processes: with *S. Sheerazuddin*.
- on games: with *Sunil Simon*.
- on security: with *S. P. Suresh*.
- on semantics of messages: with *Rohit Parikh*.

Many people have worked on the theory of message sequence charts: many ideas presented here intersect with the work of: *Rajeev Alur, Benedikt Böllig, Paul Gastin, Blaise Genest, Narayan Kumar, P. Madhusudan, Madhavan Mukund, Anca Muscholl, Doron Peled*.

Welcome to India!

- We have an Association for Logic in India, a forum for interaction between researchers in logic, mathematics, philosophy and computer science. (www.cmi.ac.in/∼ali)
- *Odd years*: Conference on logic and its applications.
- *Even years*: School on logic. Next one – January 2008, IIT, Kanpur.
- We also have an annual conference in theoretical computer science in December. Next one – December 2007, IIT, Delhi. (www.fsttcs.org, www.iarcs.org.in).

You are also welcome at the *Institute of Mathematical Sciences, Chennai* (www.imsc.res.in).