Polynomial-time logic and liveness

(work in progress)

Peeter Laud

Cybernetica AS & Tartu University
(joint work with Dominique Unruh)

February 5th, 2011

Propositional Logic

- Let $\mathbf{Var}_{\mathrm{b}}$ be a set of propositional variables
- Let b range over $\mathbf{Var}_{\mathrm{b}}$
- Let $\mathbb{B} = \{\mathsf{true}, \mathsf{false}\}$

Propositional formulas F

$F ::= \mathsf{true} \mid b \mid \neg F \mid F_1 \vee F_2$

Aliases

- $\mathsf{false} \equiv \neg\mathsf{true}$
- $F_1 \wedge F_2 \equiv \neg(\neg F_1 \vee \neg F_2)$
- $F_1 \supset F_2 \equiv \neg F_1 \vee F_2$

Classical semantics of propositional formulas

Worlds

A world is a map $W : \mathbf{Var}_{\mathrm{b}} \to \mathbb{B}$.

A formula being true in a world

$$W \models \mathsf{true}$$
$$W \models b \text{ if } W(b) = \mathsf{true}$$
$$W \models \neg F \text{ if } W \not\models F$$
$$W \models F_1 \vee F_2 \text{ if } W \models F_1 \text{ or } W \models F_2$$

A formula is valid if it is true in all worlds

First-order logic

- Let **F** and **P** be finite sets of functional and predicate symbols.
  - With fixed arities
  - Ranged over by f and P, respectively
- Let $\mathbf{Var}_i$ be a set of first-order variables.
  - Ranged over by x

Terms t

$$t ::= x \mid f(t_1, \ldots, t_n)$$

First-order formulas F

$$F ::= \ldots \mid P(t_1, \ldots, t_n) \mid \exists x.F$$

Aliases

- $\forall x.F \equiv \neg\exists x.\neg F$

Classical semantics of first-order formulas

Frames

A frame is $\mathcal{F} = (\mathbf{S}, \iota_F, \iota_P)$, where

- $\mathbf{S} \subseteq \{0,1\}^*$ is a set
- $\iota_F(f)$ is a mapping $\mathbf{S}^{\text{arity}(f)} \to \mathbf{S}$
- $\iota_P(P)$ is a mapping $\mathbf{S}^{\text{arity}(P)} \to \mathbb{B}$.

Worlds

A world in frame $\mathcal{F}$ is a pair (W_i, W_b), where

- $W_i : \mathbf{Var}_{\mathrm{i}} \to \mathbf{S}$
- $W_b : \mathbf{Var}_{\mathrm{b}} \to \mathbb{B}$

Classical semantics of first-order formulas

Let $\mathcal{F} = (\mathbf{S}, \iota_F, \iota_P)$ be fixed.

Evaluating terms

- Let $W_i : \mathbf{Var}_i \to \mathbf{S}$
- Define $W_i(f(t_1, \ldots, t_n)) = \iota_F(f)(W_i(t_1), \ldots, W_i(t_n))$

A formula being true in a world

$W \models \text{true}$

$W \models b$ if $W_b(b) = \text{true}$

$W \models \neg F$ if $W \not\models F$

$W \models F_1 \vee F_2$ if $W \models F_1$ or $W \models F_2$

$W \models P(t_1, \ldots, t_n)$ if $\iota_P(P)(W_i(t_1), \ldots, W_i(t_n)) = \text{true}$

$W \models \exists x.F$ if exists $v \in \mathbf{S}$, such that$(W_i[x \mapsto v], W_b) \models F$

First-order linear temporal logic

FO LT formulas F

$$F ::= \ldots \mid \bigcirc F \mid F_1 \,\mathcal{U}\, F_2$$

Aliases

- $\Diamond F \equiv \mathsf{true}\,\mathcal{U}\, F$
- $\Box F \equiv \neg\Diamond\neg F$
- $F_1 \,\mathcal{W}\, F_2 \equiv (F_1 \,\mathcal{U}\, F_2) \vee \Box F_1$

Used to argue about traces produced by systems

Classical semantics of FO LT formulas

Traces

A trace $\mathcal{T}$ is a sequence $\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \ldots$ of worlds

A formula being true in point i of trace $\mathcal{T}$

$(\mathcal{T}, i) \vDash \exists x.F$ if exists $v \in \mathbf{S}$, such that$(\mathcal{T}[x \mapsto v], i) \vDash F$

$(\mathcal{T}, i) \vDash \bigcirc F$ if $(\mathcal{T}, i+1) \vDash F$

$(\mathcal{T}, i) \vDash F_1 \,\mathcal{U}\, F_2$ if exists $j \geq i$, such that

$(\mathcal{T}, j) \vDash F_2$

$(\mathcal{T}, k) \vDash F_1$ for all $k \in \{i, i+1, \ldots, j-1\}$

where $\mathcal{T}[x \mapsto v]$ is $(\mathcal{T}_{0,i}[x \mapsto v], \mathcal{T}_{0,b}), (\mathcal{T}_{1,i}[x \mapsto v], \mathcal{T}_{1,b}), \ldots$

Two views of cryptography

Formal ("Dolev-Yao") view

- Messages — elements of a term algebra.
- Possible operations on messages are enumerated.
- Choices in semantics — non-deterministic.
  - Protocol and the adversary are easily represented in some process calculus.
- Simpler to analyse.

Computational view

- Messages — bit strings.
- Possible operations on messages — everything in PPT.
- Choices in semantics — probabilistic.
  - Protocol and adversary — a set of probabilistic interactive Turing machines.
- Closer to the real world.

LT logic and protocol properties

- A run of a protocol produces a trace.
- A trace may or may not satisfy a LT formula.
- Many important protocol properties can be stated as trace properties:
  - In formal model: all traces satisfy the formula.
  - In computational model: the probability that a trace does not satisfy the formula is negligible in the security parameter.
- Safety properties state that bad things never occur
  - Extensively studied
- Liveness properties state that something good eventually occurs
  - Much less studied
  - Especially in the computational model

Games

- Two players, $\mathcal{E}$ (proponent) and $\mathcal{A}$ (opponent).
- A game is determined by its tree.
  - Each node labeled by $\mathcal{E}$ or $\mathcal{A}$.
  - Each edge labeled by the name of the move.
  - The label of a non-leaf node denotes the party on the move.
  - The label of a leaf node denotes the winner.
- A game tree may be infinite both in width and in depth.
  - Not really in this talk.

Game-based semantics for propositional formulas

Let $W : \mathbf{Var}_{\mathrm{b}} \to \mathbb{B}$.

Game trees for formulas

- The game tree $[\![\mathrm{true}]\!]$ of true has a single node labeled $\mathcal{E}$.
- The game tree $[\![b]\!]$ of $b \in \mathbf{Var}_{\mathrm{b}}$ has a single node labeled with $\mathcal{E}$ is $W(b) = \mathrm{true}$.
- The game tree $[\![\neg F]\!]$ is the same as $[\![F]\!]$, with node labels swapped.
- The game tree $[\![F_0 \vee F_1]\!]$ consists of
  - the root node v, labeled with $\mathcal{E}$;
  - the trees $[\![F_0]\!]$ and $[\![F_1]\!]$;
  - Edges labeled $z \in \{0, 1\}$ from the node v to the root nodes of $[\![F_z]\!]$.

Semantics

$W \models F$ if $\mathcal{E}$ has winning strategy in $[\![F]\!]$.

Example

$A \wedge (B \vee C) \supset (A \wedge B) \vee (A \wedge C)$

- If A and (B or C) then pick $(A \wedge B) \vee (A \wedge C)$
  - If B then pick $A \wedge B$
  - Otherwise (if C then) pick $A \wedge C$

  otherwise pick $A \wedge (B \vee C)$ to be attacked
  - If A then pick $B \vee C$
  - Otherwise pick A

Semantics is not good enough

$F \vee \neg F$

- Give a winning strategy for $\mathcal{E}$.
  - Without deeply examining F.
- Do you pick F or $\neg F$???

Semantics is not good enough

$F \vee \neg F$

- Give a winning strategy for $\mathcal{E}$.
  - Without deeply examining F.
- Do you pick F or $\neg F$???

We'd like to make $\mathcal{A}$ play against itself...

- Let $\mathcal{A}$ attack both F and $\neg F$.
- $\mathcal{E}$ causes $\mathcal{A}$ to play against itself, winning one game and losing one.
- $\mathcal{E}$ wins because it wins when $\mathcal{A}$ loses.

Need to change semantics of disjunction

Game-based semantics of disjunction

Some notation

- For game tree T and available move m, let T^m be the subtree after the move m.
- For game trees T_0, T_1,... T_n, let $T_0|T_1|\cdots|T_n$ denote the following tree:
  - Root node v is labeled with $\mathcal{E}$, has children w_0, w_1, ..., w_n.
    - Edge from v to w_z is labeled with z.
  - The node w_z has the same label and possible moves as the root node of T_z.
  - The tree following the move m from w_i is $T_0|\cdots|T_{i-1}|T_i^m|T_{i+1}|\cdots|T_n$.

Game corresponding to $F_0 \vee F_1$

$[\![F_0 \vee F_1]\!] = [\![F_0]\!]|[\![F_1]\!]$.

Game-based semantics for FO and LT formulas

Let the frame $\mathcal{F}$ and the world W / trace $\mathcal{T}$ be fixed.

Game tree for $\exists$

- $[\![\exists x.F]\!] = \left|_{v \in \mathbf{S}} [\![F[v/x]]\!]\right.$

Game-based semantics for FO and LT formulas

Let the frame $\mathcal{F}$ and the world W / trace $\mathcal{T}$ be fixed.

Game tree for $\exists$

- $[\![\exists x.F]\!] = \Big|_{v \in \mathbf{S}} [\![F[v/x]]\!]$

Another formula constructor

$@^t F$ is a formula for a FO LT formula F and a term t.

$$(\mathcal{T}, i) \models @^t F \text{ if } (\mathcal{T}, i + |t|) \models F$$

In game-based setting, shift the trace.

Introduce the predicate $|\cdot| < |\cdot|$ with obvious interpretation.

Syntactic sugar

- $\bigcirc F \equiv @^1 F$
- $F_1 \,\mathcal{U}\, F_2 \equiv \exists x.(@^x F_2 \wedge \forall y.(|y| < |x| \supset @^y F_1))$

Example

$\forall^1 x.(P^2(x) \supset^3 Q^4(x)) \supset^5 (\forall^6 x.P^7(x) \supset^8 \forall^9 x.Q^{10}(x))$

1. Pick $\forall^6 x.P^7(x) \supset^8 \forall^9 x.Q^{10}(x)$, then pick $\forall^9 x.Q^{10}(x)$.
2. Get t^9 from $\mathcal{A}$.
3. If $Q(t)$, then we're done.
4. Otherwise check $P(t)$.
   - If $P(t)$, then back up, select $\forall^1 x.(P^2(x) \supset^3 Q^4(x))$ to attack, put t as x^1.
   - If $\neg P(t)$, then back up, select $\forall^6 x.P^7(x)$ to attack, put t as x^6.
5. If $\mathcal{A}$ backs up and chooses a new t, then repeat from 3.

Example

$$\forall^1 x.(P^2(x) \supset^3 Q^4(x)) \supset^5 (\forall^6 x.P^7(x) \supset^8 \forall^9 x.Q^{10}(x))$$

1. Pick $\forall^6 x.P^7(x) \supset^8 \forall^9 x.Q^{10}(x)$, then pick $\forall^9 x.Q^{10}(x)$.
2. Get t^9 from $\mathcal{A}$.
3. If $Q(t)$, then we're done.
4. Otherwise check $P(t)$.
   - If $P(t)$, then back up, select $\forall^1 x.(P^2(x) \supset^3 Q^4(x))$ to attack, put t as x^1.
   - If $\neg P(t)$, then back up, select $\forall^6 x.P^7(x)$ to attack, put t as x^6.
5. If $\mathcal{A}$ backs up and chooses a new t, then repeat from 3.

Stopping?

- The game may be infinite.
- We require that when first considering an $\exists$-node in the AST of the formula, the player must state how many descendants of the tree it will consider.
  - $\mathcal{A}$ must state it in step 2.
  - $\mathcal{E}$ states "1" in both branches of step 4.

What if $\mathcal{E}$ and $\mathcal{A}$ were PPT?

What it means to be "polynomial-time"

- Each execution step in time polynomial in the security parameter.
- Care necessary with one player "outspending" the other one.
- The number of attempts on quantifiers must be bounded, too.
  - The stated number may not be more that the previous largest number plus $p(\eta)$.

What does it give us?

- Liveness properties in the computational setting.
  - Precise meanings for $\Diamond F$ and $\Box\Diamond F$.

Deductive power

All axioms and inference rules of FO and LT logic are still valid.

Contract-signing protocols

- Two parties A and B, and a message M (the contract).
  - Signature verification keys pk_A and pk_B known to all.
- As a result of the protocol
  - A obtains a "signature" of B on M and B obtains a "signature" of A on M;
  - or, neither A nor B obtain each others signature.
- Requires a trusted third party.
- A contract signing protocol is optimistic if the TTP has to be contacted only if one of the parties misbehaves.

Asokan-Shoup-Waidner fair contract signing protocol

Let $[\{M\}]_A$ denote A's signature on M.

Main protocol

$A \to B : [\{M, A, B, T\}]_A$.
$B \to A : [\{M, A, B, T\}]_B$. If time-out, A invokes Abort protocol
$A \to B : [\{M, A, B\}]_A$. If time-out, B invokes Resolve protocol
$B \to A : [\{M, A, B\}]_B$. If time-out, A invokes Resolve protocol

Abort protocol

$A \to T : [\{M, A, B, abort\}]_A$
$T \to A :$
$\begin{cases} [\{[\{M, A, B, T\}]_A, [\{M, A, B, T\}]_B\}]_T \\ [\{[\{M, A, B, abort\}]_A\}]_T \end{cases}$

Resolve protocol

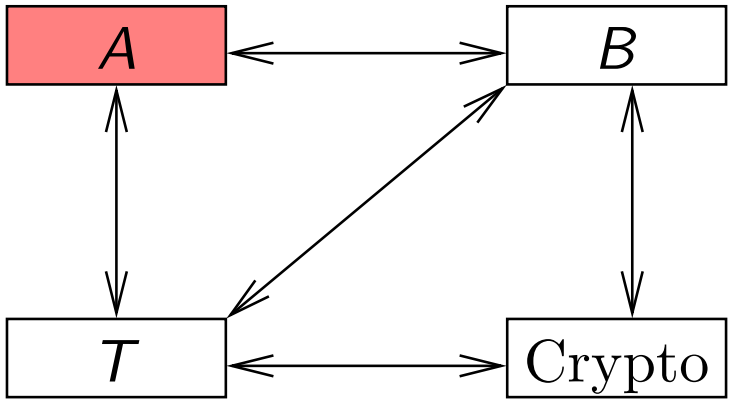
$A \to T :$
$[\{M, A, B, T\}]_A, [\{M, A, B, T\}]_B$
$T \to A :$
$[\{[\{M, A, B, T\}]_A, [\{M, A, B, T\}]_B\}]_T$

A signed contract is either $([\{M, A, B\}]_A, [\{M, A, B\}]_B)$ or $[\{[\{M, A, B, T\}]_A, [\{M, A, B, T\}]_B\}]_T$.

Modeling the ASW protocol

The property we consider

If A obtains a signed contract for message M then B eventually also has a signed contract on M.



Some variables

pk_A, pk_B, pk_T — public keys out_A, out_B — output of A and B
$msg_{\overrightarrow{XY}}$ — outgoing messages from X to Y
$msg_{\overleftarrow{XY}}$ — messages incoming to X from Y
ts_B, ts_T — messages B and T want to get signed
fs_B, fs_T — signatures to messages from previous step
L_B, L_T — all messages ever signed by B / T

Modeling the ASW protocol

Behaviour of the network

- $\Box(\forall x.(x \in msg_{\overrightarrow{BT}} \supset \Diamond x \in msg_{\overleftarrow{TB}}))$
  - Same for communication $T \to B$.
- $\Box(\forall x.(x \in msg_{\overrightarrow{AT}} \supset \bigcirc x \in msg_{\overleftarrow{TA}}))$
  - Same for other communication with A.
- $\forall x.(x \notin msg_{\overleftarrow{YX}} \mathcal{W}\, x \in msg_{\overrightarrow{XY}})$
  - for all $X, Y \in \{A, B, T\}$

Modeling the ASW protocol

Behaviour of the network

- $\Box(\forall x.(x \in msg_{\overrightarrow{BT}} \supset \Diamond x \in msg_{\overleftarrow{TB}}))$
  - Same for communication $T \to B$.
- $\Box(\forall x.(x \in msg_{\overrightarrow{AT}} \supset \bigcirc x \in msg_{\overleftarrow{TA}}))$
  - Same for other communication with A.
- $\forall x.(x \notin msg_{\overleftarrow{YX}} \mathcal{W} x \in msg_{\overrightarrow{XY}})$
  - for all $X, Y \in \{A, B, T\}$

The signing machine

- $\Box(next(L_X) = L_X \cup ts_X) \qquad L_X = \emptyset$
  - Actually $\Box(\exists x.(x = L_X \cup ts_X \wedge \bigcirc(x = L_x)))$
- $\Box(\texttt{all}\ (\backslash(m, s)\texttt{->}Vfy(pk_X, m, s))\ (\texttt{zip}\ ts_X\ next(fs_X)))$

Modeling the ASW protocol

State of B and T

- Formula relating S_X, $next(S_X)$, $msg_{\overleftarrow{XY}}$, $next(msg_{\overrightarrow{XY}})$, fs_X, $next(ts_X)$.
  - S_X — the internal state of X
- Timeouts will happen: $\Box\Diamond\neg wait\_state(S_B)$

Modeling the ASW protocol

State of B and T

- Formula relating S_X, $next(S_X)$, $msg_{\overleftarrow{XY}}$, $next(msg_{\overrightarrow{XY}})$, fs_X, $next(ts_X)$.
  - S_X — the internal state of X
- Timeouts will happen: $\Box\Diamond\neg wait\_state(S_B)$

Security of signatures

$\Box(\forall(m,s).(Vfy(pk_X,m,s) \supset \Diamond m \in L_X))$

- Otherwise $A, B, T, \mathcal{A}, \mathcal{E}$ together have broken the EF-CMA security of signatures.

Modeling the ASW protocol

State of B and T

- Formula relating S_X, $next(S_X)$, $msg_{\overleftarrow{XY}}$, $next(msg_{\overrightarrow{XY}})$, fs_X, $next(ts_X)$.
  - S_X — the internal state of X
- Timeouts will happen: $\Box\Diamond\neg wait\_state(S_B)$

Security of signatures

$\Box(\forall(m,s).(Vfy(pk_X,m,s) \supset \Diamond m \in L_X))$

- Otherwise $A, B, T, \mathcal{A}, \mathcal{E}$ together have broken the EF-CMA security of signatures.

The property we consider

$\Box(\forall x.(contract(x,A,B) \in out_A \supset \Diamond contract(x,A,B) \in out_B))$

Conclusions

- Semantics for first-order linear temporal logic, giving meaning to polynomial-time FO and LT effects.
- Analysis of the liveness properties of the ASW protocol in the computational model.

Open question

- How to model that the adversary does not know a certain value?
  - In computational model, this is not a trace property.