Alcino Cunha

SPECIFICATION AND MODELING

FIRST-ORDER LINEAR TEMPORAL LOGIC

Universidade do Minho & INESC TEC

2019/20

DISTRIBUTED ATOMIC TRANSACTION PROTOCOL

DISTRIBUTED ATOMIC TRANSACTION PROTOCOL TRASH

```
sig Worker {}
var sig Prepared in Worker {}
var sig Committed in Prepared {}
var sig Aborted in Worker {}

...

fact transitions {
  always (
    nop or some w : Worker | finish[w] or commit[w] or abort[w]
  )
}
```

SOME DESIRED PROTOCOL ASSERTIONS

```
assert Consistency {
    -- It is never the case that we have both committed
    -- and aborted workers
    always (no Committed or no Aborted)
}

assert Stability {
    -- Once committed a worker stays committed (same for aborted)
    all w : Worker {
        always (w in Committed implies always w in Committed)
        always (w in Aborted implies always w in Aborted)
    }
}
```

OTHER (HOPEFULLY) TRUE PROTOCOL ASSERTIONS

- Once some worker is committed all will eventually be committed
- A worker can only be committed after finishing its task
- All workers will eventually be forever committed or aborted
- After finishing its task a worker is prepared

SOME FALSE PROTOCOL ASSERTIONS

- The aborted workers never finished their task
- After a worker finishes its task it will commit and remain prepared until then
- Workers will repeatedly finish their tasks

FIRST-ORDER LINEAR TEMPORAL LOGIC

FIRST-ORDER LINEAR TEMPORAL LOGIC

- Electrum includes temporal connectives from *Linear Temporal Logic* (LTL)
  - Both future and past operators
- But also the prime operator, that evaluates an expression in the next state
  - Introduced by Lamport in the *Temporal Logic of Actions* (TLA)
  - Since Electrum also has first-order quantifiers it does not increase expressivity, but considerably simplifies specifications
- A linear temporal formula is interpreted in a state of a *trace* (an infinite sequence of states)
  - A formula is satisfied in a trace iff it is satisfied in its initial state
  - A formula is satisfied in a system iff it is satisfied in all possible traces

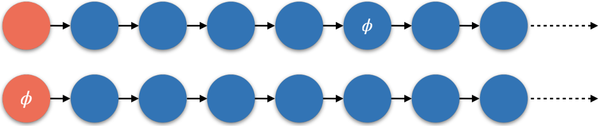
FUTURE OPERATORS

Electrum	Math	Meaning
always ϕ	$\mathsf{G}\,\phi$ $\quad \Box\,\phi$	ϕ is always true from now on
eventually ϕ	$\mathsf{F}\,\phi$ $\quad \Diamond\,\phi$	ϕ will eventually be true
after ϕ	$\mathsf{X}\,\phi$ $\quad \bigcirc\,\phi$	ϕ will be true in the next state
ϕ **until** ψ	$\phi\ \mathsf{U}\ \psi$	ψ will eventually be true and ϕ is true until then
ϕ **releases** ψ	$\phi\ \mathsf{R}\ \psi$	ψ can only be false after ϕ is true

SEMANTICS BY EXAMPLE

always ϕ

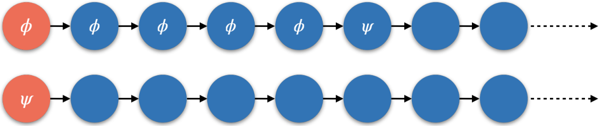
eventually ϕ



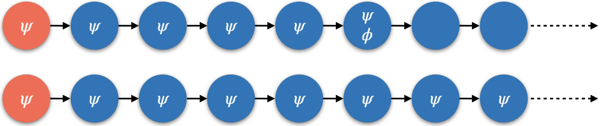
after ϕ

SEMANTICS BY EXAMPLE

ϕ **until** ψ



ϕ **releases** ψ



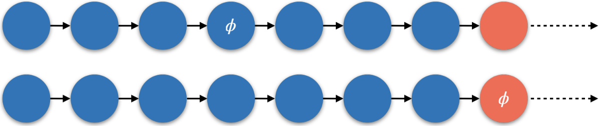
PAST OPERATORS

Electrum	Math	Meaning
historically ϕ	$\mathsf{H}\,\phi$	ϕ was always true
once ϕ	$\mathsf{O}\,\phi$	ϕ was once true
before ϕ	$\mathsf{Y}\,\phi$	ϕ was true in the previous state
ϕ **since** ψ	$\phi\ \mathsf{S}\ \psi$	ψ was once true and ϕ has been true afterwards
ϕ **triggered** ψ	$\phi\ \mathsf{T}\ \psi$	ψ can only be false before ϕ was true

SEMANTICS BY EXAMPLE

historically ϕ

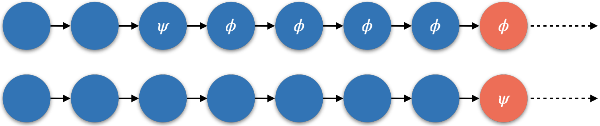
once ϕ



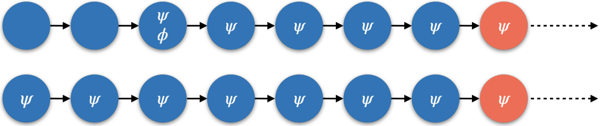
before ϕ

SEMANTICS BY EXAMPLE

ϕ **since** ψ



ϕ **triggered** ψ



DISTRIBUTED ATOMIC TRANSACTION PROTOCOL

OTHER (HOPEFULLY) TRUE PROTOCOL ASSERTIONS

```
assert Prop {
    -- Once some worker is committed all will eventually be committed
    always (some Committed implies eventually Worker in Committed)
}
```

OTHER (HOPEFULLY) TRUE PROTOCOL ASSERTIONS

```
assert Prop {
    -- A worker can only be committed after finishing its task
    all w : Worker | finish[w] releases w not in Committed
}
```

```
assert Prop {
    -- A worker can only be committed after finishing its task
    all w : Worker | always (w in Committed implies once finish[w])
}
```

OTHER (HOPEFULLY) TRUE PROTOCOL ASSERTIONS

```
assert Prop {
    -- All workers will eventually be forever committed or aborted
    all w : Worker | eventually always w in Committed or
                     eventually always w in Aborted
}
```

OTHER (HOPEFULLY) TRUE PROTOCOL ASSERTIONS

```
assert Prop {
    -- After finishing its task a worker is prepared
    all w : Worker | always (finish[w] implies after w in Prepared)
}
```

SOME FALSE PROTOCOL ASSERTIONS

```
assert Prop {
    -- The aborted workers never finished their task
    all w : Worker | always (w in Aborted implies
                                historically not finish[w])
}
```

SOME FALSE PROTOCOL ASSERTIONS

```
assert Prop {
    -- After a worker finishes its task it will eventually commit
    -- and remain prepared until then
    all w : Worker | always (finish[w] implies
                             after (w in Prepared until commit[w]))}
```

SOME FALSE PROTOCOL ASSERTIONS

```
assert Prop {
    -- Workers will repeatedly finish their tasks
    all w : Worker | always eventually finish[w]
}
```

FIRST-ORDER LINEAR TEMPORAL LOGIC

SYNTAX

$$
\begin{array}{rcl}
\phi & ::= & \mathsf{G}\,\phi \\
& | & \mathsf{F}\,\phi \\
& | & \mathsf{X}\,\phi \\
& | & \phi\ \mathsf{U}\ \psi \\
& | & \phi\ \mathsf{R}\ \psi \\
& | & \mathsf{H}\,\phi \\
& | & \mathsf{O}\,\phi \\
& | & \mathsf{Y}\,\phi \\
& | & \phi\ \mathsf{S}\ \psi \\
& | & \phi\ \mathsf{T}\ \psi \\
& | & \ldots
\end{array}
$$

$$
\begin{array}{rcl}
\Phi & ::= & \Phi' \\
& | & \ldots
\end{array}
$$

FIRST-ORDER TEMPORAL STRUCTURES

- The semantics of a first-order *temporal* formula is defined over a first-order *temporal* structure $\mathcal{U}, \mathcal{M}$ where
  - $\mathcal{U}$ is a non-empty domain (or *universe*) of interpretation with equality
  - $\mathcal{M}$ is an infinite sequence of possible interpretations for constants, predicates, and variables
  - For all $i \in \mathbb{N}$, we have
    - $\mathcal{M}(i)(c) \in \mathcal{U}$, with $\mathcal{M}(i)(c) = \mathcal{M}(i')(c)$ for all $i' \in \mathbb{N}$
    - $\mathcal{M}(i)(x) \in \mathcal{U}$
    - $\mathcal{M}(i)(P) \subseteq \mathcal{U}^{\mathrm{ar}(P)}$
- The fact that a formula ϕ is true in the i-th state of a model $\mathcal{M}$ with universe $\mathcal{U}$ is denoted by $\mathcal{U}, \mathcal{M}, i \models \phi$
- When $\mathcal{U}$ is implicit or clear from context we write just $\mathcal{M}, i \models \phi$
- A formula ϕ is true in a model $\mathcal{M}$, denoted by $\mathcal{M} \models \phi$, iff $\mathcal{M}, 0 \models \phi$

SEMANTICS

$$
\begin{aligned}
\mathcal{M}, i \models \mathsf{G}\,\phi &\quad \text{iff} \quad \forall j \geq i \,.\, \mathcal{M}, j \models \phi \\
\mathcal{M}, i \models \mathsf{F}\,\phi &\quad \text{iff} \quad \exists j \geq i \,.\, \mathcal{M}, j \models \phi \\
\mathcal{M}, i \models \mathsf{X}\,\phi &\quad \text{iff} \quad \mathcal{M}, i+1 \models \phi \\
\mathcal{M}, i \models \phi \;\mathsf{U}\; \psi &\quad \text{iff} \quad \exists j \geq i \,.\, (\mathcal{M}, j \models \psi \;\wedge\; \forall i \leq k < j \,.\, \mathcal{M}, k \models \phi) \\
\mathcal{M}, i \models \phi \;\mathsf{R}\; \psi &\quad \text{iff} \quad \forall j \geq i \,.\, (\mathcal{M}, j \models \psi \;\vee\; \exists i \leq k < j \,.\, \mathcal{M}, k \models \phi)
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{M}, i \models \mathsf{H}\,\phi &\quad \text{iff} \quad \forall 0 \leq j \leq i \,.\, \mathcal{M}, j \models \phi \\
\mathcal{M}, i \models \mathsf{O}\,\phi &\quad \text{iff} \quad \exists 0 \leq j \leq i \,.\, \mathcal{M}, j \models \phi \\
\mathcal{M}, i \models \mathsf{Y}\,\phi &\quad \text{iff} \quad i > 0 \wedge \mathcal{M}, i-1 \models \phi \\
\mathcal{M}, i \models \phi \;\mathsf{S}\; \psi &\quad \text{iff} \quad \exists 0 \leq j \leq i \,.\, (\mathcal{M}, j \models \psi \;\wedge\; \forall j < k \leq i \,.\, \mathcal{M}, k \models \phi) \\
\mathcal{M}, i \models \phi \;\mathsf{T}\; \psi &\quad \text{iff} \quad \forall 0 \leq j \leq i \,.\, (\mathcal{M}, j \models \psi \;\vee\; \exists j < k \leq i \,.\, \mathcal{M}, k \models \phi)
\end{aligned}
$$

SEMANTICS

$$
\begin{aligned}
\mathcal{M}, i \models \Phi \subseteq \Psi \quad & \text{iff} \quad \mathcal{M}, i \models \forall \vec{x} \cdot \Phi(\vec{x}) \rightarrow \Psi(\vec{x}) \\
\mathcal{M}, i \models P(\vec{t}) \quad & \text{iff} \quad (\mathcal{M}(i)(t_1), \ldots, \mathcal{M}(i)(t_n)) \in \mathcal{M}(i)(P) \\
\mathcal{M}, i \models \Phi'(\vec{t}) \quad & \text{iff} \quad \mathcal{M}, i+1 \models \Phi(\vec{t}) \\
\mathcal{M}, i \models \forall x \cdot \phi \quad & \text{iff} \quad \mathcal{M}[\mathbb{N} \mapsto x \mapsto a], i \models \phi \text{ for all } a \in \mathcal{U} \\
& \quad \ldots
\end{aligned}
$$