

APPENDIX

A Java source for the running example

This appendix contains a simplified Java source code for the running example.

```
public class Sale {
    ...
    protected Sale(...) { ... }
    public int getNumberOfItems() { ... }
    public PaymentMode getPaymentMode() { ... }
    public long getTimeofSale() { ... }
}

public class CoordinatorEventHandler {
    private List<Sale> sales;
    ...

    public void onEvent(SaleRegisteredEvent saleRegisteredEvent) {
        updateStatistics(...);
        if (isExpressModeNeeded()) {
            coordinatorPublisher.publish(expressModeEnabledEvent);
        }
    }

    private void updateStatistics(...) {
        sales.add(new Sale(...));
    }

    private boolean isExpressModeNeeded() {
        ...
        for ( Iterator<Sale> i = sales.iterator(); i.hasNext(); ) {
            Sale s = i.next();
            if (s.getTimeofSale() + 3600000 < now) {
                i.remove();
            } else {
                if (s.getPaymentMode().equals(CASH) && s.getNumberOfItems() <= 8) {
                    ...
                }
            }
        }
        return ...;
    }
}
```

B CI-LTL

Syntax. For a given set of labels, formulas of CI-LTL are defined as

1. $\mathcal{P}(l)$ and $\mathcal{E}(l)$ are formulas, where l is a label.
2. If Φ and Ψ are formulas, then also $\Phi \wedge \Psi$, $\neg \Phi$, $\mathcal{X} \Phi$ and $\Phi \mathcal{U} \Psi$ are formulas.
3. Every formula can be obtained by a finite number of applications of steps 1. and 2.

Semantics. We define a *run* of a CI automaton $\mathcal{C} = (Q, Act, \delta, I, H)$ as an infinite sequence $\sigma = q_0, l_0, q_1, l_1, q_2, \dots$ where $q_i \in Q$, and $\forall i. (q_i, l_i, q_{i+1}) \in \delta$. We further define:

- $\sigma(i) = q_i$ (i -th state of σ)
- $\sigma^i = q_i, l_i, q_{i+1}, l_{i+1}, q_{i+2}, \dots$ (i -th sub-run of σ)
- $\mathcal{L}(\sigma, i) = l_i$ (i -th label of σ)

CI formulas are interpreted over runs where the satisfaction relation \models is defined inductively as:

$$\begin{aligned}
\sigma \models \mathcal{E}(l) & \iff \exists q. \sigma(0) \xrightarrow{l} q \\
\sigma \models \mathcal{P}(l) & \iff \mathcal{L}(\sigma, 0) = l \\
\sigma \models \Phi \wedge \Psi & \iff \sigma \models \Phi \text{ and } \sigma \models \Psi \\
\sigma \models \neg \Phi & \iff \sigma \not\models \Phi \\
\sigma \models \mathcal{X} \Phi & \iff \sigma^1 \models \Phi \\
\sigma \models \Phi \mathcal{U} \Psi & \iff \exists j \in \mathbb{N}_0. \sigma^j \models \Psi \text{ and } \forall k \in \mathbb{N}_0, k < j. \sigma^k \models \Phi
\end{aligned}$$

C The main theorem

This appendix rephrases the main theorem of [1] and its proof, even if only informally. For formal transcription please see the original paper.

Theorem 1. *Let \mathcal{D} be a dynamic system model, $\{\varphi_i\}_{i \in \mathbb{N}} \in \text{Property}(\mathcal{D}, m)$, X a set of observable labels containing all labels necessary for verification of $\{\varphi_i\}_{i \in \mathbb{N}}$, denote $n = |\mathcal{D}|_X$, and suppose a few minor restrictions on $\{\mathcal{D}_i\}_{i \geq n}$ and \mathcal{F} (see [1]). Then for every $k \in \mathbb{N}$: $\mathcal{D}_{m+n} \models \varphi_{m+n} \Rightarrow \mathcal{D}_{m+n+k} \models \varphi_{m+n+k}$.*

Proof. The idea of the proof is the following. If the property is independent on the clients $\{\varphi_i\}_{i \in \mathbb{N}} \in \text{Property}(\mathcal{D}, 0)$, and $|\mathcal{D}|_X = n$, then from the definition of $|\mathcal{D}|_X$ any run in \mathcal{D}_{n+k} violating the property has its equivalent in \mathcal{D}_n and hence the violation can be detected already while verifying $\mathcal{D}_0, \mathcal{D}_1, \dots, \mathcal{D}_n$. If the property involves some clients $\{\varphi_i\}_{i \in \mathbb{N}} \in \text{Property}(\mathcal{D}, m)$ then we can place these m clients inside the provider (denote the result $\overline{\mathcal{D}}$), interpret $\{\varphi_i\}_{i \in \mathbb{N}}$ as a property involving only these internal clients $\{\overline{\varphi}_i\}_{i \in \mathbb{N}}$, and hence get $\{\overline{\varphi}_i\}_{i \in \mathbb{N}} \in \text{Property}(\overline{\mathcal{D}}, 0)$. Now it suffices to do the following. We verify $\{\overline{\varphi}_i\}_{i \in \mathbb{N}}$ on $\overline{\mathcal{D}}_0, \overline{\mathcal{D}}_1, \dots, \overline{\mathcal{D}}_n$, which is equivalent to checking $\{\varphi_i\}_{i \in \mathbb{N}}$ on $\mathcal{D}_m, \mathcal{D}_{m+1}, \dots, \mathcal{D}_{m+n}$. If false, the counterexample has its equivalent on the original model. If true, it implies the validity of $\{\varphi_i\}_{i \in \mathbb{N}}$ on $\mathcal{D}_m, \mathcal{D}_{m+1}, \dots, \mathcal{D}_{m+n}, \dots$ (application of the previous case). To complete the process, we only check $\{\varphi_i\}_{i \in \mathbb{N}}$ on the remaining $\mathcal{D}_0, \mathcal{D}_1, \dots, \mathcal{D}_{m-1}$.