8

Communicating Finite-State Machines

In this chapter, we introduce and study *communicating machines* (CMs), a model of computation rather than a specification language, which is close to a real-life implementation of a communicating system where distributed components communicate via fifo channels (which might be reliable or faulty, bounded or unbounded).

8.1 Communicating (Finite-State) Machines

A CM is a collection of state machines that share one global initial state and several global final states. The machines are connected pairwise with (for the moment) unbounded reliable fifo buffers. The transitions of each component are labeled with send or receive actions. Hereby, a send action $i!j$ puts a message at the end of the channel from i to j. A receive action can be taken provided the requested message is found in the channel. To extend the expressive power, CMs can send certain *synchronization messages*. Let us be more precise:

Definition 8.1 (Communicating Machine). *A* communicating machine *(over Ag) is a structure*

$$\mathcal{A} = ((\mathcal{A}_i)_{i \in Ag}, \mathcal{D}, \overline{s}^{in}, F)$$

such that

- $\mathcal{D}$ *is a nonempty finite set of* synchronization messages *(or* data*),*
- *for each $i \in Ag$, $\mathcal{A}_i$ is a pair (S_i, Δ_i) where*
  - S_i *is a nonempty set of (i-)*local states *and*
  - $\Delta_i \subseteq S_i \times \Gamma_i(Ag) \times \mathcal{D} \times S_i$ *is the set of (i-)*local transitions*,*
- $\overline{s}^{in} \in \prod_{i \in Ag} S_i$ *is the* global initial state*, and*
- $F \subseteq \prod_{i \in Ag} S_i$ *is the finite set of* global final states.

A CM $\mathcal{A} = ((\mathcal{A}_i)_{i \in Ag}, \mathcal{D}, \overline{s}^{in}, F)$, $\mathcal{A}_i = (S_i, \Delta_i)$, is called

- an N-CM, $N \in \mathbb{N}_{\geq 1}$, if $|\mathcal{D}| = N$,
- a *communicating finite-state machine* (CFM) or *finite* if, for each $i \in Ag$, S_i is finite,
- *locally accepting* if, for any $i \in Ag$, there is a set $F_i \subseteq S_i$ such that $F = \prod_{i \in Ag} F_i$, and
- *deterministic* if, for any $i \in Ag$, Δ_i satisfies the following conditions:
  - If $(s, i!j, m_1, s_1) \in \Delta_i$ and $(s, i!j, m_2, s_2) \in \Delta_i$, then $m_1 = m_2$ and $s_1 = s_2$.
  - If $(s, i?j, m, s_1) \in \Delta_i$ and $(s, i?j, m, s_2) \in \Delta_i$, then $s_1 = s_2$.

By CM(Ag), we denote the class of CMs over Ag and by CFM(Ag) the class of CFMs.[1] However, as the underlying set of agents will be clear from the context, we henceforth omit any reference to Ag and just write CM and, respectively, CFM. For a set $\mathfrak{C}$ of CMs, we denote by N-$\mathfrak{C}$, $\mathfrak{C}_\ell$, and det-$\mathfrak{C}$ the classes of N-, locally accepting, and deterministic CMs $\mathcal{A}$ with $\mathcal{A} \in \mathfrak{C}$, respectively.

We now define the behavior of CMs and, in doing so, adhere to the style of [56]. In particular, an automaton will run on MSCs rather than linearizations of MSCs, allowing for its distributed behavior. Let $\mathcal{A} = ((\mathcal{A}_i)_{i \in Ag}, \mathcal{D}, \overline{s}^{in}, F)$, $\mathcal{A}_i = (S_i, \Delta_i)$, be a CM and moreover let $\mathcal{M} = (V, \lhd, \lambda) \in \mathbb{MSC}$ be an MSC. A *run* of $\mathcal{A}$ on $\mathcal{M}$ is a pair (ρ, μ) of mappings $\rho : V \to \bigcup_{i \in Ag} S_i$ with $\rho(u) \in S_{Ag(u)}$ for each $u \in V$ and $\mu : V \to \mathcal{D}$ such that

- for any $u, v \in V$ with $u \lhd_c v$, $\mu(u) = \mu(v)$, and
- for any $u \in V$, $(source^{\overline{s}^{in}}_{(\mathcal{M},\rho)}(u)[Ag(u)], \lambda(u), \mu(u), \rho(u)) \in \Delta_{Ag(u)}$.

We call (ρ, μ) *accepting* if $final^{\overline{s}^{in}}_{(\mathcal{M},\rho)} \in F$. By $L(\mathcal{A}) := \{\mathcal{M} \in \mathbb{MSC} \mid$ there is an accepting run of $\mathcal{A}$ on $\mathcal{M}\}$, let us denote the *language* of $\mathcal{A}$. In particular, $\mathbf{1}_{\mathbb{MSC}}$ is a member of $L(\mathcal{A})$ if (and only if) $\overline{s}^{in} \in F$. Given a class $\mathfrak{C}$ of CMs, we furthermore set $\mathcal{L}(\mathfrak{C})$ to be $\{L \subseteq \mathbb{MSC} \mid$ there is $\mathcal{A} \in \mathfrak{C}$ such that $L(\mathcal{A}) = L\}$, which is the class of languages of $\mathfrak{C}$. We consider $\mathcal{L}$(CFM) to be some kind of standard class, which is identified by $\mathcal{CFM} := \mathcal{L}$(CFM). We also say that the languages from $\mathcal{CFM}$ are the *implementable* ones. This nomenclature is arbitrary and rather geared to the literature, where the term *realizability* usually refers to locally accepting 1-CMs. The intuition behind local acceptance is that recognition by the whole system defers to acceptance by any single component. In other words, any component may decide independently of the others whether it accepts or not. In contrast, the general acceptance condition allows to take a global view of the system to decide upon acceptance.

Example 8.2. A (nondeterministic) locally accepting 2-CFM $\mathcal{A}$ over $\{1, 2\}$ with set of synchronization messages $\{\Diamond, \Box\}$ is illustrated in Fig. 8.1. As we

[1] Note that CM(Ag) does not impose any restriction on the state space, which can therefore produce non-recursive behavior.