3

Graphs, Logics, and Graph Acceptors

Directed graphs are the most general structures we consider in this book. Many structures that will also be addressed, such as words and (graphs associated to) posets, can be embedded into graphs or at least have a corresponding one-to-one graph representation.

3.1 Graphs

In the following, let Σ and C be alphabets, which contain the elements the components of a graph are labeled with. Hereby, Σ is the supply of *actions* that a system may execute. The actions will label the nodes of a graph, which we will later refer to as *events*. The elements from C label (color) the edges of a graph and provide a kind of control-flow information.

Definition 3.1 (Graph). *A (directed)* graph *over* (Σ, C) *is a structure* $(V, \{\lhd_\ell\}_{\ell \in C}, \lambda)$ *where*

- V *is its finite set of* nodes*,*
- *the* $\lhd_\ell \subseteq V \times V$ *are disjoint binary relations on* V*, and*
- $\lambda : V \to \Sigma$ *is the* (node-)labeling function.

Thus, we consider a node $u \in V$ of a graph $\mathcal{G} = (V, \{\lhd_\ell\}_{\ell \in C}, \lambda)$ over (Σ, C) to be labeled with a letter $a \in \Sigma$ if $\lambda(u) = a$ and we consider a pair $(u, v) \in \bigcup_{\ell \in C} \lhd_\ell$ to be labeled with $\ell' \in C$ if $(u, v) \in \lhd_{\ell'}$. In the sequel, we call $\lhd := \bigcup_{\ell \in C} \lhd_\ell$ the *edge relation* or the set of *edges* of $\mathcal{G}$. Moreover, we sometimes write $\leq_\ell$ for $(\lhd_\ell)^*$, abbreviate $(\lhd_\ell)^+$ by $<_\ell$, set $\leq$ to be the relation $\lhd^*$, and abbreviate $\lhd^+$ by $<$. We call $\mathcal{G}$ *connected* if, for any $u, v \in V$, $(u, v) \in (\lhd \cup \lhd^{-1})^*$. The *cardinality* of $\mathcal{G}$, denoted by $|\mathcal{G}|$, is actually meant to be the cardinality $|V|$ of V. Moreover, for a subset Σ' of Σ, we set $|\mathcal{G}|_{\Sigma'}$ to be $|\lambda^{-1}(\Sigma')|$. Observe that $|\mathcal{G}|_\Sigma = |\mathcal{G}|$. Given $a \in \Sigma$, we moreover abbreviate $|\mathcal{G}|_{\{a\}}$ with $|\mathcal{G}|_a$.

The set of graphs over (Σ, C) is denoted by $\mathbb{DG}(\Sigma, C)$. Note that we silently assume a graph (actually, its set of nodes) to be nonempty if it seems more convenient, e.g., if we require a mapping on the set of nodes. However, it will always be clear how to extend the setting accordingly to deal with the empty graph.

Let $B \in \mathbb{N}$ be a natural number. For $\mathcal{G} = (V, \{\lhd_\ell\}_{\ell \in C}, \lambda) \in \mathbb{DG}(\Sigma, C)$, we say that the degree of $\mathcal{G}$ is *bounded* by B if, for any $u \in V$, $|\{v \in V \mid u \lhd v$ or $v \lhd u\}| \leq B$. Given a set $\mathcal{K}$ of graphs over (Σ, C), the *degree* of $\mathcal{K}$ is said to be *bounded* by B if, for any $\mathcal{G} \in \mathcal{K}$, the degree of $\mathcal{G}$ is bounded by B. We say that $\mathcal{K}$ has *bounded degree* if its degree is bounded by some B. By $\mathcal{K}[B]$, we denote the class of graphs $\mathcal{G} \in \mathcal{K}$ such that the degree of $\mathcal{G}$ is bounded by B.

It will be useful to define *extended* graphs, whose nodes are equipped with an additional labeling. Let Q be a nonempty and finite set. A (Q-)*extended graph* over (Σ, C) is a graph $(V, \{\lhd_\ell\}_{\ell \in C}, \lambda)$ over $(\Sigma \times Q, C)$, i.e, λ is a mapping $V \to \Sigma \times Q$. Note that λ can be seen as a pair (λ', ρ) of mappings $V \to \Sigma$ and $V \to Q$, respectively. Given a class $\mathcal{K}$ of graphs over (Σ, C) and a (possibly empty) finite set Q, we define $\langle \mathcal{K}, Q \rangle$ to be $\mathcal{K}$ if Q is empty and, otherwise, to be the set of Q-extended graphs $(V, \{\lhd_\ell\}_{\ell \in C}, (\lambda, \rho))$ over (Σ, C) such that $(V, \{\lhd_\ell\}_{\ell \in C}, \lambda) \in \mathcal{K}$. If Q is nonempty and we are given $\mathcal{G} = (V, \{\lhd_\ell\}_{\ell \in C}, \lambda) \in \mathbb{DG}(\Sigma, C)$ and a mapping $\rho : V \to Q$, then we write $(\mathcal{G}, \rho)$ to denote the extended graph $(V, \{\lhd_\ell\}_{\ell \in C}, (\lambda, \rho)) \in \langle \mathbb{DG}(\Sigma, C), Q \rangle$.

3.2 Monadic Second-Order Logic over Graphs

We recall the notion of *monadic second-order (MSO) logic* over graphs, i.e., in its most general case, which then carries over to the more specific cases of MSO logic over words, traces, and message sequence charts. Fragments of MSO logic will provide logical characterizations of respective automata models. For a comprehensive overview of MSO logics, see [39].

Throughout the book, we will use supplies $\mathrm{Var} = \{x, y, \ldots, x_1, x_2, \ldots\}$ of *individual variables* and $\mathrm{VAR} = \{X, Y, \ldots, X_1, X_2, \ldots\}$ of *set variables.*

Definition 3.2 (Monadic Second-Order Logic). *The set* $\mathrm{MSO}(\Sigma, C)$ *of* monadic second-order formulas *over* (Σ, C) *(or* $\mathrm{MSO}(\Sigma, C)$*-formulas) is built up from the atomic formulas*

- $\lambda(x) = a$ *(with* $x \in \mathrm{Var}$ *and* $a \in \Sigma$*),*
- $x \lhd_\ell y$ *(with* $x, y \in \mathrm{Var}$ *and* $\ell \in C$*),*
- $x \in X$ *(with* $x \in \mathrm{Var}$ *and* $X \in \mathrm{VAR}$*), and*
- $x = y$ *(with* $x, y \in \mathrm{Var}$*),*

and, furthermore, allow the boolean connectives $\neg$*,* $\vee$*,* $\wedge$*,* $\rightarrow$*, and* $\leftrightarrow$ *and the quantifiers* $\exists$ *and* $\forall$*, which can be applied to either kind of variable and are called individual (first-order) and set (second-order) quantifiers, respectively. More precisely, if* φ *and* ψ *are formulas from* $\mathrm{MSO}(\Sigma, C)$*, then so are* $\neg\varphi$*,*