labelled digraph

Canonical name	LabelledDigraph
Date of creation	2013-03-22 15:15:07
Last modified on	2013-03-22 15:15:07
Owner	GrafZahl (9234)
Last modified by	GrafZahl (9234)
Numerical id	4
Author	GrafZahl (9234)
Entry type	Definition
Classification	msc 05C20
Classification	msc 05C12
Classification	msc 05C78
Classification	msc 05C90
Defines	label
Defines	weighted digraph
Defines	weight
Defines	vertex-weighted digraph
Defines	edge-weighted digraph

A triple (V, E, w) is called a *labelled digraph*, if (V, E) is a digraph and w is an association of elements from some set L, the *labels*, to some of the edges and vertices of the digraph. In other words, w is a mapping from a subset $A \subseteq V \cup E$ to L. Most often, L is a subset of the real numbers, in which case (V, E, w) is called a *weighted* digraph and its labels are called *weights*. Typically, either $A = V$ or $A = E$, in which case (V, E, w) is called either a *vertex-weighted digraph* or an *edge-weighted digraph*, respectively.

Application examples

We give two typical "real life" examples. The first features an edge-weighted digraph, while the second requires the implementation of a vertex-weighted digraph.

Railway network

A railway network consists of railway stations connected by rails. A train needs a certain time (measured in minutes) to fare from one station to another. In a formalisation, V is the set of train stations, E the set of direct connexions between them and $w\colon E \to \mathbb{N}$ a weighting corresponding to the journey times, so (V, E, w) is an edge-weighted digraph. Although typically (V, E) is a `http://planetmath.org/node/1702`symmetric digraph, w does not need to be symmetric: for example, the journey from a to b might take longer than the return journey because b is located on a mountain.

An important optimisation problem is the efficient determination of the fastest way from one station to another. An even harder problem is to find the fastest round trip (usually called a *tour*) via a given number of stations. This is the *travelling salesman problem*.

Dependency graph

A software bundle consists of a number of packages each of which is either installed or not. An installed package occupies a certain amount of bytes on a storage medium. Packages may depend on other packages, that is installation of a package may require other packages to be installed first, which in turn may require still other packages and so forth. One is interested in the complete storage requirement incurred by the installation of one package and all its dependencies.

In a formalisation, the packages are vertices of a digraph (V, E), and an edge $(a, b) \in E$ means "a depends on b". Such a digraph is typically not symmetric. The weighting $w\colon V \to \mathbb{N}$ associates sizes to packages. A subset W of V is *dependency-closed*, if for any $w \in W$, all dependencies of w are in W. Given a to-be-installed package v, the storage requirement incurred by the installation of v and all its dependencies is the sum of the vertex weights of the smallest dependency-closed subset of V containing v.