

According to the definitions, the requirement for a language to be computably enumerable is clearly a relaxation of the requirement for a language to be computable; the stipulation that the DTM be *halting* is lifted. We record this fact, to be used tacitly in the sequel, as follows.

Proposition 2.1.21. *Each language that is computable is also computably enumerable.*

To recapitulate, a language B is computable if there exists a DTM that accepts each string in B , and rejects each string outside B ; a language B is computably enumerable if there exists a DTM that accepts each string in B , and does not accept any string outside B —so, the DTM either rejects or loops on each string outside B . One might succinctly say that the definition of *computable* is based on the distinction between acceptance and rejection, whereas the definition of *computably enumerable* is based on the distinction between acceptance and nonacceptance.

Remark 2.1.22 (On the presentation of DTMs). When we claim the existence of a DTM with a particular behavior, in the sequel we typically do *not* present a DTM formally by giving all parts of the 7-tuple in the definition of DTM, as we did in Examples 2.1.3, 2.1.6, 2.1.12, and 2.1.13. Rather, we give a high-level description of what the DTM should do; we appeal to the reader’s sense, intuition, and judgment that the description could be implemented by a DTM (if arduously). Recall that the original impetus behind introducing the DTM was, in any case, to formalize the *intuitive* notion of algorithm. \diamond

2.1.4 Summary of models and language classes

The following table shows the computational models that have been studied so far, along with the language classes that they define:

Computational model	Defined class of languages	Justification
DFA	regular languages	Definition 1.1.1
NFA	regular languages	Theorem 1.3.24
ϵ -NFA	regular languages	Theorem 1.3.24
halting DTM	computable languages	Definition 2.1.15
DTM	CE languages	Definition 2.1.20

As discussed in Section 2.1.3, each language that is regular is also computable, but not vice versa; and each language that is computable is also CE. The relationship between the computable languages and the CE languages will be clarified later in this chapter (specifically, in Section 2.5.1).