*6.16 Prove that there exist two languages A and B that are Turing-incomparable—that is, where $A \not\leq_{\mathrm{T}} B$ and $B \not\leq_{\mathrm{T}} A$.

*6.17 Let A and B be two disjoint languages. Say that language C *separates* A and B if $A \subseteq C$ and $B \subseteq \overline{C}$. Describe two disjoint Turing-recognizable languages that aren't separable by any decidable language.

6.18 In Corollary 4.18 we showed that the set of all languages is uncountable. Use this result to prove that languages exist that are not recognizable by an oracle Turing machine with oracle for A_{TM}.

6.19 Recall the Post correspondence problem that we defined in Section 5.2 and its associated language PCP. Show that PCP is decidable relative to A_{TM}.

6.20 Show how to compute the descriptive complexity of strings $\mathrm{K}(x)$ with an oracle for A_{TM}.

6.21 Use the result of Problem 6.20 to give a function f that is computable with an oracle for A_{TM}, where for each n, $f(n)$ is an incompressible string of length n.

6.22 Show that the function $\mathrm{K}(x)$ is not a computable function.

6.23 Show that the set of incompressible strings is undecidable.

6.24 Show that the set of incompressible strings contains no infinite subset that is Turing-recognizable.

*6.25 Show that for any c, some strings x and y exist, where $\mathrm{K}(xy) > \mathrm{K}(x) + \mathrm{K}(y) + c$.

SELECTED SOLUTIONS

6.3 Say that M_1^B decides A and M_2^C decides B. Use an oracle TM M_3, where M_3^C decides A. Machine M_3 simulates M_1. Every time M_1 queries its oracle about some string x, machine M_3 tests whether $x \in B$ and provides the answer to M_1. Because machine M_3 doesn't have an oracle for B and cannot perform that test directly, it simulates M_2 on input x to obtain that information. Machine M_3 can obtain the answer to M_2's queries directly because these two machines use the same oracle, C.

6.5 The statement $\exists x\,\forall y\,\big[\,x{+}y{=}y\,\big]$ is a member of $\mathrm{Th}(\mathcal{N},+)$ because that statement is true for the standard interpretation of $+$ over the universe $\mathcal{N}$. Recall that we use $\mathcal{N} = \{0,1,2,\ldots\}$ in this chapter and so we may use $x = 0$. The statement $\exists x\,\forall y\,\big[\,x{+}y{=}x\,\big]$ is not a member of $\mathrm{Th}(\mathcal{N},+)$ because that statement isn't true in this model. For any value of x, setting $y = 1$ causes $x{+}y{=}x$ to fail.

6.9 Assume for the sake of contradiction that some TM X decides a property P, and P satisfies the conditions of Rice's theorem. One of these conditions says that TMs A and B exist where $\langle A\rangle \in P$ and $\langle B\rangle \notin P$. Use A and B to construct TM R:

```
R = "On input w:
    1. Obtain own description ⟨R⟩ using the recursion theorem.
    2. Run X on ⟨R⟩.
    3. If X accepts ⟨R⟩, simulate B on w.
       If X rejects ⟨R⟩, simulate A on w."
```

If $\langle R\rangle \in P$, then X accepts $\langle R\rangle$ and $L(R) = L(B)$. But $\langle B\rangle \notin P$, contradicting $\langle R\rangle \in P$, because P agrees on TMs that have the same language. We arrive at a similar contradiction if $\langle R\rangle \notin P$. Therefore our original assumption is false. Every property satisfying the conditions of Rice's theorem is undecidable.

6.10 The statement ϕ_{eq} gives the three conditions of an equivalence relation. A model (A, R_1), where A is any universe and R_1 is any equivalence relation over A, is a model of ϕ_{eq}. For example, let A be the integers $\mathcal{Z}$ and let $R_1 = \{(i,i) | \; i \in \mathcal{Z}\}$.

6.12 Reduce $\mathrm{Th}(\mathcal{N}, <)$ to $\mathrm{Th}(\mathcal{N}, +)$, which we've already shown to be decidable. To do so, show how to convert a sentence ϕ_1 over the language of $\mathrm{Th}(\mathcal{N}, <)$, to a sentence ϕ_2 over the language of $\mathrm{Th}(\mathcal{N}, +)$ while preserving truth or falsity in the respective models. Replace every occurrence of $i < j$ in ϕ_1 by the formula $\exists k \big[(i{+}k{=}j) \land (k{+}k{\neq}k) \big]$ in ϕ_2, where k is a different new variable each time.

Sentence ϕ_2 is equivalent to ϕ_1 because "i is less than j" means that we can add a nonzero value to i and obtain j. Putting ϕ_2 into prenex-normal form, as required by the algorithm for deciding $\mathrm{Th}(\mathcal{N}, +)$, requires a bit of additional work. The new existential quantifiers are brought to the front of the sentence. To do so, these quantifiers must pass through Boolean operations that appear in the sentence. Quantifiers can be brought through the operations of $\land$ and $\lor$ without change. Passing through $\neg$ changes $\exists$ to $\forall$ and vice-versa. Thus $\neg\exists k\, \psi$ becomes the equivalent expression $\forall k\, \neg\psi$, and $\neg\forall k\, \psi$ becomes $\exists k\, \neg\psi$.