Solutions to Exercises 6.2, 6.3, 6.4, and 6.5

Exercise 6.2. Define a relation R by setting

$$R(a_1, \ldots, a_n, i) \Leftrightarrow \forall b < i\,(g(a_1, \ldots, a_n, b) \neq 0).$$

By Lemma 6.5, R is primitive recursive. (The notes omit the $\forall$ part of the proof of Lemma 6.5, but it is similar to the $\exists$ part, with generalized product in the place of generalized sum.) Observe that

$$h(a_1, \ldots, a_n) = \sum_{i < f(a_1, \ldots, a_n)} K_R(a_1, \ldots, a_n, i).$$

Exercise 6.3. The definition of $(a)_b$ needs to be corrected to make $(0)_b = 0$. (The problem is that every number divides 0.) Let $a > 0$. For all $i < (a)_b$, ${p_b}^{i+2}$ divides a. Hence either $(a)_b = 0$ or ${p_b}^{(a)_b+1}$ divides a, and so $(a)_b \leq {p_b}^{(a)_b+1} \leq a$.

The prime factors of a include p_i for each $i < \mathrm{lh}(a)$. Hence $a = \mathrm{lh}(a) = 0$ or $a \geq \prod_{i<\mathrm{lh}(a)} p_i \geq \mathrm{lh}(a)$.

Either $a = 0$, and so $a\lceil b = 0$, or else $a > 0$ and $a\lceil b = \prod_{i<b} {p_i}^{a_i}$, where a_i is the exponent of p_i in the prime factorization of a.

The proof of Lemma 5.19 directly shows that the graph of $a \mapsto p_a$ is representable in Q, and its argument can be used to show that the graph of that function is primitive recursive. To get the representability of the function itself, we tacitly appealed to Corollary 5.10, which states that a function is representable in Q if its graph is representable. Corollary 5.10 was proved using the fact that the functions representable in Q are closed under the μ-operator. The primitive recursive functions are not closed under the μ-operator, and the graph of a function can be primitive recursive without the function's being primitive recursive.

Exercise 6.4. (1) Define $\mathrm{Sb}(a, b, c) = d$ to hold just in case one of the following conditions is satisfied.

(i) a is the # of a variable and $a = b$ and $d = c$.

(ii) $\exists i < a^{a \cdot \mathrm{lh}(a)} \exists k < a[i \in \mathrm{Seq}$ and, for all $j < \mathrm{lh}(i)$, $(((i)_j$ is the # of a term and k is the # number of a $\mathrm{lh}(i)$ place relation or function symbol and $a = k * \ast_{j<\mathrm{lh}(i)} (i)_j$ and $d = k * \ast_{j<\mathrm{lh}(i)} \mathrm{Sb}((i)_j, b, c)))]$.

(iii) $\exists i < a\, \exists j < a$ [i and j are numbers of terms and $a = i * \#(=) * j$ and $d = \text{Sub}(i,b,c) * \#(=) * \text{Sb}(j,b,c)$].

(iv) $\exists i < a$[i is the # of a formula and $a = \#(\neg) * i$ and $d = \#(\neg) * \text{Sb}(i,b,c)$].

(v) $\exists i < a\, \exists j < a$[$i$ and j are #s of formulas and $a = \#(() * i * \#(\rightarrow) * j * \#())$ and $d = \#(() * \text{Sb}(i,b,c) * \#(\rightarrow) * \text{Sb}(j,b,c) * \#())$]

(vi) $\exists i < a\, \exists j < a$[$i$ is the # of a variable and $i \neq b$ and j is the # of a formula and $a = \#(\forall) * i * j$ and $d = \#(\forall) * i * \text{Sb}(j,b,c)$].

(vii) None of the conditions on a and b are met (ignoring the clauses beginning "$d = \ldots$") and $d = a$.

Let $\text{Sb}'(b,c,a) = \text{Sb}(a,b,c)$. Since the definition of $\text{Sb}'(b,c,a)$ uses values $\text{Sb}'(b,c,i)$ only for $i < a$, we can define $\overline{\text{Sb}'}(b,c,a+1)$ as a primitive recursive function of b, c, a, and $\overline{\text{Sb}'}(b,c,a)$. (See the proof of part (2) of Lemma 6.10.)

(2) For $\mathcal{L}^{\text{A}}$, and other languages containing $\mathbf{0}$, define $\text{Fr}(a,b) \Leftrightarrow \text{Sb}(a,b,\#\mathbf{0}) \neq a$. For languages without $\mathbf{0}$ (or any other constants), define $\text{Fr}(a,b) \Leftrightarrow \text{Sb}(a,b,\#v_1) \neq \text{Sb}(a,b,\#v_2)$.

(3) a is the # of a sentence if and only if a is the the # of a formula and, for all $b < a$ that is the # of a variable, $\neg\text{Fr}(a,b)$.

(4) Define $\text{Sbl}(a,b,c)$ to hold just in case a is the # of a formula, b is the number of a variable, c is the number of a term, and one of the following conditions is satisfied.

(i) $\neg\text{Fr}(a,b)$.

(ii) a is the # of an atomic formula.

(iii) $\exists i < a(a = \#(\neg) * i$ and $\text{Sbl}(i,b,c))$.

(iv) $\exists i < a\, \exists j < a(a = \#(() * i * \#(\rightarrow) * j * \#())$ and $\text{Sbl}(i,b,c)$ and $\text{Sbl}(j,b,c))$.

(v) $\exists i < a\, \exists j < a$[$a = \#(\forall) * i * j$ and i is the # of a variable and $\forall n < \text{lh}(c)$ (if $(c)_n$ is the symbol number of a variable then $(\text{Sbl}(j,b,2^{(c)_n+1})$ and $i \neq 2^{(c)_n+1})))$].

Now let $\text{Sbl}'(b,c,a) \Leftrightarrow \text{Sb}(a,b,c)$, and proceed as with part (2) of Lemma 6.10 and part (1) of Lemma 6.12.

Exercise 6.5. The set asked about is the set of all (a,b,c) such that a, b, and c are #s of formulas and either $b = \#(() * a * \#(\rightarrow) * c * \#())$ or $a = \#(() * b * \#(\rightarrow) * c * \#())$. The reason for the either/or is that the problem did not specify an order for the premises of the rule.