P1

a) Si tenemos A conjunto finito sabemos que $\exists\, f: A \to \{1, \ldots n\}$ biyectiva, esto ya que al ser finito podemos contar sus elementos, con esto, a cada elemento en a podemos escribirlo así:

$$a_i \text{ con } i = f(a)$$

Ahora podemos ver que si definimos el sgte orden:

$$a_i \preccurlyeq a_j \iff i \leq j, \text{ es decir } f(a_i) \leq f(a_j)$$

Esta relación hereda las propiedades fundamentales del orden de los naturales.

b) Recordemos que un número cualquiera se puede escribir en binario usando solo 0s y 1s, a cada uno de estos digitos lo llamamos un bit.

Construyamos nuestra función $\Psi: P(A) \to M_n$ con M_n los números representables en binario.

$\forall\, B \subseteq A \quad \Psi(B) = s \;,\; s \in M_n$ donde s cumple lo siguiente:

el i-ésimo bit de s (s_i) es un $1 \iff a_i \in B$

el i-ésimo bit de s es un $0 \iff a_i \notin B$

Queremos ver que Ψ es biyectiva:

Inyectiva

Sean B, C tq $\Psi(B) = \Psi(C)$

$\forall i$ se cumple $\Psi(B)_i = 1 \iff a_i \in B$

$\Rightarrow (\forall i)\ a_i \in B \Rightarrow \Psi(B)_i = \Psi(C)_i = 1 \Rightarrow a_i \in C$

Por lo tanto si $a_i \in B$, también está en C

Además se tiene lo mismo para C, por lo que

$\Psi(B) = \Psi(C) \Rightarrow B = C$

Sobreyectiva

Tomemos $s \in M_n$. Definamos el conjunto D de la siguiente forma:

$D = \{a_i \mid s_i = 1\}$ es decir que tiene todos los elementos a tales que $s_{f(a)} = 1$

Ahora, queremos ver que $D \subseteq A$ y $\Psi(D) = s$

$D \subseteq A$ $\forall i$ $a_i \in A$ y D sólo contiene algunos a_i. Es decir $a_i \in D \Rightarrow a_i \in A$

$\Psi(D) = s$ Por la forma en que definimos D, sabemos que:

$\forall i$ $a_i \in D \iff s_i = 1$ y esta es la definición de $\Psi(D)$, por lo tanto $\Psi(D) = s$.

$\therefore$ Ψ es biyectiva

Ejemplo de Ψ:

$A = \{*, \triangle, \square, \$\}$

$f(*) = 1$, $f(\triangle) = 2$, $f(\square) = 3$, $f(\$) = 4$

Tomemos $\underbrace{\{*, \square\}}_{O}$, $\underbrace{\{\square, \$\}}_{P}$, $\underbrace{\{*, \triangle, \square\}}_{Q}$

$\Psi(O) = 1010$

$\Psi(P) = 0011$

$\Psi(Q) = 1110$

c) Encontramos una biyección entre $P(A)$ y M_n

$\Rightarrow |P(A)| = |M_n|$ ahora: ¿Cuantos elementos tiene M_n?

M_n contiene todos los elementos representables con $|A| = n$ bits $\Rightarrow |M_n| = 2^n$

$\therefore |P(A)| = |M_n| = 2^n$

Ahora, nuestro algoritmo es el siguiente:

Tenemos A ordenado según f.

comenzamos con $k = 0$.

mientras $k \leq 2^n$:

Para cada a_i si el bit i-ésimo de k está prendido, imprimo a_i.

le sumo 1 a k

facilmente se ve que este algoritmo en las 2^n iteraciones pasa por todos los subconjuntos de K.

d) Finalmente, analizemos por qué esto algoritmo falla en el caso infinito. Debe fallar, ya que si no, $P(\mathbb{N})$ sería enumerable y no lo es.
Vamos a ver que si $n = \infty$ entonces hay elementos en M_∞ a los que el algoritmo nunca va a llegar.

Sea $\overline{B} \in P(A)$ tal que:

$a_i \in \overline{B}$ si i es par

$a_i \notin \overline{B}$ si i es impar

Ahora, asumamos, por contradicción, que existe C tal que luego de C iteraciones nuestro algoritmo imprime $\overline{B}$.

Pero, con C iteraciones solamente podemos haber alcanzado los primeros $\log_2 C$ bits de nuestro número K, lo que quiere decir que no es posible que el bit $2C$-ésimo esté prendido.

Pero $2C$ es par $\Rightarrow a_{2C} \in \overline{B}$

No podemos haber impreso a $\overline{B}$ si no imprimimos a a_{2C}

P2

a) Demostraremos por contradicción:

Asumamos que $\mathcal{F}$, ~~el~~ conjunto de las funciones computables es enumerable.

Como $\mathcal{F}$ es enumerable, $\exists\, \Psi: \mathcal{F} \to \mathbb{N}$ bixectiva.

Llamaremos f_i a $f \in \mathcal{F}$ tq $\Psi(f) = i$.

Ahora, definamos $g(n)$ de la siguiente forma:

$$g(i) = f_i(\cancel{x}i) + 1$$

g no puede estar en $\mathcal{F}$, ya que $\forall i$ $g \neq f_i$ en al menos una posición, ya que $g(i) = f_i(i)+1 \Rightarrow g(i) \neq f_i(i)$

$\Rightarrow g(i) \neq \mathcal{F}$, pero g es computable, porque está compuesto de otras funciones computables

$\Rightarrow \forall i$ $g(i)$ se puede computar.

$\rightarrow\leftarrow$ $\mathcal{F}$ contiene a todas las funciones computables pero $g \notin \mathcal{F}$.

$\therefore$ $\mathcal{F}$ no es enumerable, $\mathcal{F}$ tampoco es no vacío, así que debe ser infinito no enumerable.

P2

b) P.d.Q. $\Sigma \times \mathbb{N}$ es numerable

Σ es el conjunto de las palabras con un alfabeto dado, llamémoslo Φ.
Φ es finito, ya que los lenguajes de programación se escriben con una cantidad finita de caracteres.
Observemos que si $|\Phi| = m$ es la cantidad de caracteres en Φ, entonces podemos analizar lo siguiente:

Si usamos $|\Phi| = 2$, podemos expresar números con 2 caracteres, 0 y 1
si $|\Phi| = 10$, podemos expresar números con 10 caracteres, en base 10.
si $|\Phi| = 16$, podemos extender la base 10 e incorporar A, B, C, D, F y obtenemos la base hexadecimal.

Esto se cumple $\forall m = |\Phi| \rightarrow$ podemos expresar cualquier número con m+1 caracteres.

Con esto, podemos ver que $|\Sigma| = |\mathbb{N}|$ con la siguiente función φ (recuerde $m = |\Phi|$).
Como Φ es finito, $\exists\, h: \Phi \rightarrow \{1, \dots, m\}$

$$\forall s \in \Sigma \quad \varphi(s) \equiv \sum_{i=0}^{m} m^{i} \cdot h(s_i)$$

donde s_i es el i-ésimo caracter de s.

P2

Si escribimos $\varphi(s)$ en base m podemos facilmente que φ es biyectiva, ya que en base $m+1$

$$\varphi(s) = h(s_1)\,h(s_2)\ldots h(s_k) \quad \text{con } k = \text{largo de } s$$

Donde, en cada posición de $\varphi(s)$ hay un número que se corresponde con un caracter en Φ. Por lo que φ es biyectiva.

$\Rightarrow \Sigma$ es enumerable

Sabemos que el producto cruz de 2 enumerables es, a su vez enumerable $\Rightarrow \Sigma \times \mathbb{N}$ es enumerable

Ejemplo:

$$\Phi = \{a, b, c, d, e, f\} \qquad |\Phi| = 6 \Rightarrow \text{usamos base } 7$$

$$h(a) = 1,\ h(b) = 2 \ldots h(f) = 6$$

$$\varphi(abcab) = \underbrace{12312}_{\text{base } 7} = \underbrace{3243}_{\text{base } 10}$$

La gracia de escribirlo en base 7 es que se cumple:

$$12312 = h(a)\,h(b)\,h(c)\,h(a)\,h(b)$$

P2 |

c) Queremos concluir que H no puede producir todas las funciones en $\mathcal{F}$.

Veamos que: $\forall i$

$(\forall f \in \mathcal{F}$ H produce a $f) \iff (\exists e \in \Sigma)$ $H(e,i) = f(i)$ $\forall i$ $\forall f$ $(*)$

definamos la función

$$\chi : \mathcal{F} \to \Sigma \quad \text{de la siguiente forma}$$

$$\chi(f) = e \iff H(e,i) = f(i) \ \forall i$$

Veamos que χ es inyectiva

sean $\bar{f}, \hat{f}$ $\quad \chi(\bar{f}) = \chi(\hat{f})$

$\Rightarrow \exists e = \chi(\bar{f}), (\forall i)\ H(e,i) = \bar{f}(i) = \hat{f}(i)$

$\Rightarrow \forall i \ \bar{f}(i) = \hat{f}(i)$

$\therefore \chi$ es inyectiva

como existe una inyección de $\mathcal{F}$ en Σ

$|\mathcal{F}| \leq |\Sigma|$, pero por a) $|\mathcal{F}| > |\mathbb{N}|$ y

$|\Sigma| = |\mathbb{N}|$

$\Rightarrow |\mathcal{F}| \leq |\Sigma|$ y $|\mathcal{F}| > |\Sigma|$ No puede cumplirse $(*)$

$\Rightarrow H$ no puede producir todas las funciones computables

P3

a) Recordar que no podemos usar constantes. Para solventar esto notamos:

X tal que $(\nexists Y \ \ Y<X) \Leftrightarrow X$ es el menor elemento

Con esto escribimos la proposición:

$$(\forall x) \ \ [(\nexists Y \ \ Y<X) \Rightarrow A(X)] \wedge [(\exists z \ \ z>X) \Rightarrow B(X)]$$

b) Esto se puede escribir de muchas formas:

1) $(\forall x) (A(x)) \ \ \vee \ \ (\forall x)(B(x))$

2) $(\forall x,y) [(A(x) \wedge A(y)) \vee (B(x) \wedge B(y))]$

3) $(\forall x,y) [\ A(x) \Rightarrow \neg B(y)\ ]$

Se deja como ejercicio convencerse de su validez.

c) Para esto definiremos el sucesor de x con la siguiente función:

$$succ(x) = Y \Leftrightarrow (\nexists z)(z>x \wedge Y>z)$$

Pero si x es el último elemento, $succ(x)$ no existe, por lo que debemos agregar ese caso

$$succ(x) = Y \Leftrightarrow (\nexists z)(z>x \wedge Y>z) \wedge (\exists w)(w>x)$$

$\Rightarrow$ si x es el último, no existe susesor de x.

P3|

Con Succ definido, escrivimos

$(\forall x)(B(x) \Rightarrow (\exists y)((y = succ(x) \wedge A(y)))$

d) Primero veamos que las a y b están intercaladas:

(*) $(\forall x,y)(y = succ(x) \Rightarrow [A(x) \wedge B(y)] \vee [B(x) \wedge A(x)]$

Para ver si es par, notamos que, al estar intercaladas las letras, la última letra y la primera deben ser diferentes:

(**) $(\forall x,y)(\nexists z\ z<x \wedge \nexists w\ w>y \Rightarrow [A(x) \wedge B(y)] \vee [B(x) \wedge A(y)]$

ambas tienen que cumplirse, por lo que

(*) $\wedge$ (**) representa lo pedido.

P4

Definamos:

a esta en la guild A $\Leftrightarrow$ $a \in A$ (minuscula $\Rightarrow$ jugador)

A es una guild $\Leftrightarrow$ $A \in G$ (Mayúscula $\Rightarrow$ guild)

a) $(\forall p)(\forall A \in G, B \in G)\ p \in A \wedge p \in B \Rightarrow A = B$

Ojo, aquí usamos la igualdad, sin embargo esto sólo se puede porque nuestra estructura lo permite, podría exigirse que no usen la igualdad.

b) p y j son aliados $\Leftrightarrow$ $a(p,j)$, a es una relación

p es lider de su guild $\Leftrightarrow$ $L(p)$

$(\forall p)[(\forall A \in G)\ p \in A \wedge (\forall j \in A\ a(p,j) \Rightarrow L(p)]$

c) p y j jugadores son enemigos $\Leftrightarrow$ $e(p,j)$

A y B guilds son enemigos $\Leftrightarrow$ $E(A,B)$

Ahora, primero consideramos el caso de que los líderes sean enemigos:

$(\forall A, B \in G)[E(A,B) \Leftrightarrow \exists p \wedge \exists j\ \ p \in A \wedge j \in B \wedge L(p) \wedge L(j) \wedge e(p,j)]$

Ahora el otro caso:

$(\forall A, B \in G)[E(A,B) \Leftrightarrow [(\exists p, j)\ p \in A \wedge j \in B \wedge e(p,j)] \wedge$
$[(\nexists \bar{p}, \bar{j})\ \bar{p} \in A \wedge \bar{j} \in B \wedge a(\bar{p}, \bar{j})]]$

[4] d)

$$(\forall A)(\forall j)[(\forall p \in A)\ e(p,j)) \Rightarrow (\forall p \in A)(\forall w)\ a(j,w) \Rightarrow e(p,w)]$$

Hay que tener ojo con los paréntesis, por ejemplo:

$$(\forall A)(\forall j)(\forall p \in A)[e(p,j) \Rightarrow (\forall p \in A)(\forall w)\ a(j,w) \Rightarrow e(p,w)]$$

Esta forma tiene el problema de que elimina el requisito de que toda la guild ~~sea~~ enemiga de j.