Combos de definiciones y convenciones notacionales

Combo 1

1. Defina cuando un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es llamado Σ-recursivo (no hace falta que defina "funcion Σ-recursiva")
2. Defina $\langle s_1, s_2, ...\rangle$
3. Defina "f es una funcion Σ-mixta"
4. Defina "familia Σ-indexada de funciones"
5. Defina $R(f, \mathcal{G})$

Combo 2

Defina:

1. $d \overset{n}{\vdash} d'$ (no hace falta que defina $\vdash$)
2. $L(M)$
3. $H(M)$
4. "f es una funcion de tipo (n, m, s)"
5. (x)
6. $(x)_i$

Combo 3

1. Defina cuando un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es llamado Σ-recursivamente enumerable (no hace falta que defina "funcion Σ-recursiva")
2. Defina $s^{\leq}$
3. Defina $*^{\leq}$
4. Defina $\#^{\leq}$

Combo 4

Defina cuando una funcion $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow \omega$ es llamada Σ-efectivamente computable y defina "el procedimiento $\mathbb{P}$ computa a la funcion f"

Combo 5

Defina cuando un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es llamado Σ-efectivamente computable y defina: "el procedimiento efectivo $\mathbb{P}$ decide la pertenencia a S"

Combo 6

Defina cuando un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es llamado Σ-efectivamente enumerable y defina: "el procedimiento efectivo $\mathbb{P}$ enumera a S"

Combo 7

Defina cuando una funcion $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow \omega$ es llamada Σ-Turing computable y defina "la maquina de Turing M computa a la funcion f"

Combo 8

Defina:

1. $M(P)$
2. Lt
3. Conjunto rectangular
4. "S es un conjunto de tipo (n, m)"

Combo 9

Defina:

1. "I es una instruccion de $\mathcal{S}^\Sigma$"
2. "$\mathcal{P}$ es un programa de $\mathcal{S}^\Sigma$"
3. $I_i^{\mathcal{P}}$
4. $n(\mathcal{P})$
5. Bas

Combo 10

Defina relativo al lenguaje $\mathcal{S}^\Sigma$:

1. "estado"
2. "descripcion instantanea"
3. $S_{\mathcal{P}}$
4. "estado obtenido luego de t pasos, partiendo del estado $(\vec{s}, \vec{\sigma})$"
5. "$\mathcal{P}$ se detiene (luego de t pasos), partiendo desde el estado $(\vec{s}, \vec{\sigma})$"

Combo 11

Defina:

1. $\Psi_{\mathcal{P}}^{n,m,\#}$
2. "f es Σ-computable"
3. "$\mathcal{P}$ computa a f"
4. $M^{\leq}(P)$

Combo 12

Defina cuando un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es llamado Σ-computable, cuando es llamado Σ-enumerable y defina "el programa $\mathcal{P}$ enumera a S"

Combo 13

Defina:

1. $i^{n,m}$
2. $E_{\#}^{n,m}$
3. $E_{*}^{n,m}$
4. $E_{\# j}^{n,m}$
5. $E_{* j}^{n,m}$
6. $Halt^{n,m}$
7. $T^{n,m}$
8. $AutoHalt^{\Sigma}$
9. Los conjuntos A y N

Combo 14

Explique en forma detallada la notacion lambda

Combo 15

Dada una funcion $f : D_f \subseteq \omega \times \Sigma^* \to \omega$, describa que tipo de objeto es y que propiedades debe tener el macro::

$$[\mathrm{V2} \leftarrow f(\mathrm{V1}, \mathrm{W1})]$$

Combo 16

Dado un predicado $P : D_f \subseteq \omega \times \Sigma^* \to \omega$, describa que tipo de objeto es y que propiedades debe tener el macro:

$$[\text{IF } P(\text{V1}, \text{W1}) \text{ GOTO A1}]$$

Combos de teoremas

La siguiente lista contiene 9 combos de resultados de la teoria los cuales seran utilizados para la parte teorica del examen. Algunas observaciones:

1. En algunos resultados se especifica que caso probar y tambien en un par de batallas se especifica que no se toma la prueba completa, es decir solo hay que dar una idea de como funciona la prueba y no es necesario seguir los detalles mas tecnicos de la demostracion

2. Cuando el alumno desarrolle una prueba de un resultado perteneciente a un combo, podra utilizar un resultado previo sin necesidad de demostrarlo, salvo que justo el combo exija la prueba de dicho resultado. Esto implica, por ejemplo, que se pueden usar en cualquier combo que la funciones suma, producto, etc son Σ-p.r.

3. Cuando el alumno aplique algun resultado que no figura en los resultados del combo que esta desarrollando, debera referirse a el en forma descriptivamente clara, preferentemente enunciandolo. Por ejemplo, no vale poner "por Lema 13 de la Guia 5, tenemos que".

Combo 1

Proposition 1 (caracterizacion de conjuntos p.r.) *Un conjunto S es Σ-p.r. sii S es el dominio de alguna funcion Σ-p.r.*

(En la induccion de la prueba hacer solo el caso de la composicion)

Theorem 2 (Neumann vence a Godel) *Si h es Σ-recursiva, entonces h es Σ-computable*

(En la induccion de la prueba hacer solo el caso $h = R(f, \mathcal{G})$)

Combo 2

Lemma 3 (lema de division por casos) *Supongamos* $f_i : D_{f_i} \subseteq \omega^n \times \Sigma^{*m} \to \Sigma^*$*,* $i = 1, ..., k$*, son funciones* Σ*-p.r. tales que* $D_{f_i} \cap D_{f_j} = \emptyset$ *para* $i \neq j$*. Entonces* $f_1 \cup ... \cup f_k$ *es* Σ*-p.r.*

Theorem 4 (Godel vence a Turing) *(solo idea de la prueba, ver abajo algunas observaciones) Supongamos* $f : S \subseteq \omega^n \times \Sigma^{*m} \to \Sigma^*$ *es* Σ*-Turing computable. Entonces* f *es* Σ*-recursiva. (Se puede ver la prueba en el apunte (descargarlo del mismo link de la guias))*

- No es necesario probar (ni dar la idea de la demostracion) que las funciones $\lambda n d d' \left[d \overset{n}{\underset{M}{\vdash}} d' \right]$ y $\lambda d d' \left[d \underset{M}{\vdash} d' \right]$ son $(\Gamma \cup Q)$-p.r.

- En la prueba del teorema "Godel vence a Turing" no es necesario probar que P es $(\Gamma \cup Q)$-p.r. aunque si hay que decir que lemas se aplicarian para probarlo y una idea de como se aplicarian estos lemas.

Combo 3

Theorem 5 (Godel vence a Neumann) *Si* $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \Sigma^*$ *es* Σ*-computable, entonces* f *es* Σ*-recursiva.*

Lemma 6 (sumatoria) *Sea* Σ *un alfabeto finito. Si* $f : \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m \to \omega$ *es* Σ*-p.r., con* $S_1, ..., S_n \subseteq \omega$ *y* $L_1, ..., L_m \subseteq \Sigma^*$ *no vacios, entonces la funcion* $\lambda xy\vec{x}\vec{\alpha} \left[\sum_{t=x}^{t=y} f(t, \vec{x}, \vec{\alpha}) \right]$ *es* Σ*-p.r.*

Combo 4

Theorem 7 (Turing vence a Neumann) *(solo idea de la prueba, ver abajo algunas observaciones) Si* $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \Sigma^*$ *es* Σ*-computable, entonces* f *es* Σ*-Turing computable. (Se puede ver la prueba en el apunte (descargarlo del mismo link de la guias junto con sus graficos) o tambien hay un video de esta batalla en granlogico.)*

- El lema de la maquina simuladora de un programa $\mathcal{P}$ relativo a un k hay que enunciarlo en forma precisa.

- En la descripcion de la maquina simuladora de un progarma $\mathcal{P}$ relativo a un k es importante explicar como se toma dicho k y la idea basica de como se representaran estados en la cinta.

- En la explicacion intuitiva de como se construye la maquina simuladora no es necesario dar explicitamente ninguna maquina de Turing. Las maquinas simuladoras de instrucciones (o sea $M^{+}_{j,k}$, $M^{\#,k}_{i \leftarrow j}$, etc) deberan ser descriptas simplemente dando las propiedades que deberan tener (tal como se hace en el apunte). O sea que las maquinas auxiliares tipo D_j, TD_j, etc pueden ser omitidas por completo en la exposicion. Es importante hacer el dibujo de como se unen las maquinas simuladoras de las instrucciones de $\mathcal{P}$ para formar la maquina simuladora de $\mathcal{P}$ relativo a k.

- No es necesario probar (solo mencionarlo) que si f es Σ-computable, entonces hay un programa $\mathcal{Q}$ el cual computa a f y el cual no tiene instrucciones de la forma GOTO L$\bar{\imath}$, etc etc

- En la prueba del teorema "Turing vence a Neumann", las maquinas auxiliares M_1 y M_2 deben ser solamente descriptas en funcion de sus propiedades. Luego explicar como la pegatina de las tres maquinas da una que computa a la funcion en cuestion.

Combo 5

Lemma 8 *Sean* $S_1, S_2 \subseteq \omega^n \times \Sigma^{*m}$ *conjuntos* Σ*-efectivamente enumerables. Entonces* $S_1 \cap S_2$ *es* Σ*-efectivamente enumerable. (Es ejercicio de la guia 3 y en el apunte esta probado.)*

Lemma 9 (cuantificacion acotada) *Sea* Σ *un alfabeto finito. Sea* $P : S \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m \rightarrow \omega$ *un predicado* Σ*-p.r., con* $S, S_1, ..., S_n \subseteq \omega$ *y* $L_1, ..., L_m \subseteq \Sigma^*$ *no vacios. Supongamos* $\bar{S} \subseteq S$ *es* Σ*-p.r.. Entonces* $\lambda x\vec{x}\vec{\alpha}\left[(\forall t \in \bar{S})_{t\leq x}\, P(t, \vec{x}, \vec{\alpha})\right]$

Combo 6

Lemma 10 *Si* $S \subseteq \omega^n \times \Sigma^{*m}$ *es* Σ*-efectivamente computable entonces* S *es* Σ*-efectivamente enumerable.*

Theorem 11 (caracterizacion de conjuntos r.e.) *(solo la prueba de (2)⇒(3), caso* $k = l = 1$ *y* $n = m = 2$*) Dado* $S \subseteq \omega^n \times \Sigma^{*m}$*, son equivalentes*

(1) S *es* Σ*-recursivamente enumerable*

(2) $S = I_F$*, para alguna* $F : D_F \subseteq \omega^k \times \Sigma^{*l} \rightarrow \omega^n \times \Sigma^{*m}$ *tal que cada* $F_{(i)}$ *es* Σ*-recursiva.*

(3) $S = D_f$*, para alguna funcion* Σ*-recursiva* f

Combo 7

Lemma 12 (lema de minimizacion) *Sean* $n, m \geq 0$*. Sea* $P : D_P \subseteq \omega \times \omega^n \times \Sigma^{*m} \rightarrow \omega$ *un predicado* Σ*-p.r.. Entonces*

(a) $M(P)$ *es* Σ*-recursiva.*

(b) Si hay una funcion Σ*-p.r.* $f : \omega^n \times \Sigma^{*m} \rightarrow \omega$ *tal que*

$$M(P)(\vec{x}, \vec{\alpha}) = \min_t P(t, \vec{x}, \vec{\alpha}) \leq f(\vec{x}, \vec{\alpha}), \text{ para cada } (\vec{x}, \vec{\alpha}) \in D_{M(P)},$$

entonces $M(P)$ *es* Σ*-p.r..*

Lemma 13 *Supongamos* $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow O$ *es* Σ*-recursiva y* $S \subseteq D_f$ *es* Σ*-r.e., entonces* $f|_S$ *es* Σ*-recursiva.*

Combo 8

Lemma 14 *Supongamos* $\Sigma \supseteq \Sigma_p$. *Entonces* $AutoHalt^{\Sigma}$ *no es* Σ*-recursivo.*

Theorem 15 *Supongamos* $\Sigma \supseteq \Sigma_p$. *Entonces* $AutoHalt^{\Sigma}$ *no es* Σ*-efectivamente computable. Es decir no hay ningun procedimiento efectivo que decida si un programa de* $\mathcal{S}^{\Sigma}$ *termina partiendo de si mismo.*

Lemma 16 *Supongamos que* $\Sigma \supseteq \Sigma_p$. *Entonces*

$$A = \left\{\mathcal{P} \in \mathrm{Pro}^{\Sigma} : AutoHalt^{\Sigma}(\mathcal{P}) = 1\right\}$$

es Σ*-r.e. y no es* Σ*-recursivo. Mas aun el conjunto*

$$N = \left\{\mathcal{P} \in \mathrm{Pro}^{\Sigma} : AutoHalt^{\Sigma}(\mathcal{P}) = 0\right\}$$

no es Σ*-r.e.*

Theorem 17 (Neumann vence a Godel) *Si* h *es* Σ*-recursiva, entonces* h *es* Σ*-computable*

(En la induccion de la prueba hacer solo el caso $h = M(P)$)

Combo 9

Lemma 18 *Supongamos* $f_i : D_{f_i} \subseteq \omega^n \times \Sigma^{*m} \to O$, $i = 1, ..., k$, *son funciones* Σ*-recursivas tales que* $D_{f_i} \cap D_{f_j} = \emptyset$ *para* $i \neq j$. *Entonces la funcion* $f_1 \cup ... \cup f_k$ *es* Σ*-recursiva.*

Lemma 19 $Halt^{n,m}$ *es* $(\Sigma \cup \Sigma_p)$*-p.r.*

Proposition 20 $T^{n,m}$ *es* $(\Sigma \cup \Sigma_p)$*-recursiva*