Revista de Matemática: Teoría y Aplicaciones 2006 **13**(2) : 111–116
cimpa – ucr – ccss issn: 1409-2433

CONDICIONES GENERALES PARA LA CONSTRUCCIÓN DE MODELOS DE $\hat{\Sigma}_2^b - L_mIND$

Eugenio Chinchilla*

Recibido/Received: 19/05/06 — Aceptado/Accepted: 22/09/06

Resumen

Se dan condiciones generales que resultan suficientes para la existencia de modelos de $\hat{\Sigma}_2^b - L_mIND$ contenidos en cierto "conjunto reserva".

Palabras clave: aritmética débil, modelo no estándar, clases de complejidad.

Abstract

We give general conditions that are sufficient to prove existence of models of $\hat{\Sigma}_2^b - L_mIND$ inside a special set.

Keywords: weak arithmetic, non standard model, complexity classes.

Mathematics Subject Classification: 03H15

1 Introducción

Las teorías aritméticas débiles y los problemas de conservación entre ellas, están íntimamente ligados a clases de complejidad de algoritmos y problemas de inclusión estricta entre ellas, tales como P vs. NP. Este trabajo está motivado por el problema de Σ_1^b-conservación entre las teorías $LIND$-Σ_1^b y $LLIND$-Σ_2^b. Se desconoce si existe esta relación entre las teorías, pero se sabe que esto llevaría a la igualdad entre las clases de tiempo paralelo NC^1 y NC (ver [1]). Es conocido que para obtener tal resultado de conservación, es suficiente construir un modelo de $LLIND$-Σ_2^b dentro de una "reserva polinomial no estándar", el conjunto de imágenes de un punto bajo funciones calculables en tiempo polinomial, de índice acotado por un elemento no estándar arbitrario.

*Escuela de Matemática, Universidad de Costa Rica. E-Mail: echinch@emate.ucr.ac.cr

El autor ha intentado sin éxito llevar a cabo una tal construcción. Incluso se ha intentado con teorías más débiles, cambiando $||x||$ por $|||x|||$, ..etc. En todos los casos la reserva polinomial parece demasiado pequeña. En este artículo se construyen modelos para estas teorías $\hat{\Sigma}_2^b - L_mIND$ (el entero m indica el número de iteraciones del logaritmo $|x|$). Esto se hace dentro de reservas cada vez más pequeñas, las cuales sin embargo siguen siendo superpolinomiales.

La técnica para construir estos modelos ya ha sido utilizada por el autor para probar un teorema de testimonio para funciones calculables en tiempo subexponencial [3]. Por lo natural y económico de esta técnica, los resultados aquí presentados sugieren que hay una separación entre estas teorías $\hat{\Sigma}_2^b - L_mIND$ (que parecen corresponder a tiempos de cálculo superpolinomial) y $\Sigma_1^b - LIND$ (que corresponde a tiempo polinomial). Para demostrarlo bastaría probar que la existencia de tales modelos dentro de la reserva polinomial, es condición necesaria para los resultados de conservación. Esta parece ser una vía interesante para investigación.

En la próxima sección, se presentan rápidamente los conceptos básicos. En la siguiente se presenta y demuestra el resultado principal. La prueba depende de algunos lemas que son demostrados en la última sección.

2 Conceptos básicos

Para las definiciones detalladas de los siguientes conceptos, se refiere al lector a [2] o [4]. El lenguaje es $L_2 = \{0, 1, +, \cdot, <, [x/2], |x|, \#\}$, donde $[x/2]$ es la parte entera de la mitad de x, $|x|$ es el largo de la expansión binaria de x y $x\#y = 2^{|x|\cdot|y|}$. Las cuantificaciones de la forma $Qx \leq t$, donde t es un término, son llamadas cuantificaciones *acotadas*. Aquellas de la forma $Qx \leq |t|$ se llaman *fuertemente acotadas*. Las fórmulas cuyos cuantificadores son todos fuertemente acotados, forman la clase Δ_0^b y definen predicados pertenecientes a la clase de complejidad P (calculables en tiempo polinomial). Se dice que una fórmula es $\hat{\Sigma}_1^b$ si es de la forma $\exists y \leq t[\Delta_0^b]$. En general, una fórmula es $\hat{\Sigma}_i^b$ si es de la forma $\exists y \leq t[\hat{\Pi}_{i-1}^b]$. La clase $\hat{\Pi}_i^b$ se define análogamente. Las fórmulas $\hat{\Sigma}_i^b$ definen exactamente los predicados del i-ésimo nivel en la jerarquía polinomial PH definida por [5]. Se denota por $\alpha(x)$-IND *hasta* l la fórmula

$$[\alpha(0) \wedge \forall x < l(\alpha(x) \rightarrow \alpha(x+1))] \;\rightarrow\; \alpha(l)$$

y si Γ es una clase de fórmulas y $m \in \mathbb{N}$, $\Gamma - L_mIND$ es el conjunto de fórmulas $\alpha(x)$-IND *hasta* $|l|_m$ para $\alpha \in \Gamma$ y donde $|l|_m = |(|l|_{m-1})|$ y $|l|_0 = l$. Varias funciones básicas pueden definirse en estas teorías. En lo que sigue $\langle a, b, c\rangle$ denota una codificación de tripletes usando la función de Cantor.

Suponemos una codificación natural de las máquinas de Turing. La función calculada por la máquina de Turing de código e, será denotada por $\{e\}$. FP denota la clase de funciones calculables en tiempo polinomial. Es conocido que existe una fórmula $C(e, T, x, y)$ que expresa "y se calcula a partir de x en tiempo T por la máquina de Turing de código e" (ver [4]).

3 Construcción de los modelos

A continuación probaremos que el permitir cálculos en tiempo $|a|^{|a|_m^r}$ para máquinas de índice hasta r, es suficiente para contener un modelo de $\hat{\Sigma}_2^b - L_m IND$.

Teorema 1 *Sea M elementalmente equivalente a $\mathbb{N}$, no estándar, numerable, y sean a, r elementos no estándar de M. Sea $m \in \mathbb{N}$, $m \geq 2$ y sea $R = \{y \in M \ : \ \exists e \leq r \ \exists T \leq 2^{||a||.|a|_m^r} \ C(e, T, a, y)\}$. Existe una L_2-subestructura $K^* \subseteq M$ tal que*

1. $a \in K^*$
2. K^* *es* FP*-cerrada*
3. $K^* \subseteq R$
4. $K^* \models \ \hat{\Sigma}_2^b - L_m IND.$

Prueba

Suponga enumerados los axiomas $\hat{\Sigma}_2^b - L_m IND$ con parámetros en M, donde θ varía en el conjunto de fórmulas $\hat{\Sigma}_2^b$. Se construye K^* como unión de una cadena creciente de estructuras $(K_n)_{n<\omega}$. Sea $K_0 = FP(a) = \{f(a) : f \in FP\}$ y sea θ_1-IND *hasta* l_1 el primer axioma en la enumeración cuyos parámetros están en K_0. Se desea construir $K_1 \supset_{L_2} K_0$, cerrado bajo funciones en FP y que satisfaga

$$\neg\theta_1(0) \vee \exists j < l_1[\theta_1(j) \wedge \neg\theta_1(j+1)] \vee \theta_1(l_1)$$

donde $\theta_1(j) \equiv \exists y \leq t \forall z \leq s \, \psi(j, y, z)$. Cambiando eventualmente r por otro elemento no estándar menor, podemos suponer sin pérdida de generalidad que r es una potencia de 2 estrictamente menor que $||a||$. Sea $T_j^1 = 2^{||a||.|a|_m^r - (j+1)||a||.|a|_n^{r/2}}$. Para $j = 0, \ldots, l_1 + 2$ sea $R_j^1(x) = \{ \ y : \exists e \leq r \ C(e, T_j^1, x, y) \ \}$. K_1 será generado por un elemento a_1, el cual se obtiene a partir del siguiente programa Π_1 con entrada a:

1: Calcule $r = 2^{|r|-1}$.
2: Calcule los parámetros de θ_1-IND *hasta* l_1, así como T_0^1, a partir de la entrada a.
3: $j := 0$, $y_{-1} := 0$.
4: Calcule T_{j+1}^1.
5: Busque $y_j \in R_j^1(\langle j, a, y_{j-1}\rangle)$, $y_j \leq t$, tal que para todo $z \in R_{j+1}^1(\langle j+1, a, y_j\rangle)$ tal que $z \leq s$, se tenga $M \models \psi(j, y_j, z)$.
6: Si no hay un tal y_j, el programa se detiene con salida $a_1 = \langle j, a, y_{j-1}\rangle$.
7: Si se encuentra y_j y $j < l_1$, entonces ponga $j := j+1$ y vaya a 4.
8: Si se encuentra y_{l_1} , el programa se detiene con salida $a_1 = \langle l_1 + 1, a, y_{l_1}\rangle$.

Sea $a_1 = \langle J_1 + 1, a, y_{J_1}\rangle$ y suponga por ejemplo que $0 \leq J_1 < l_1$. Entonces:

- Para todo $z \in R_{J_1+1}^1(a_1)$ tal que $z \leq s$, $M \models \psi(J_1, y_{J_1}, z)$.
- Para todo $y \in R_{J_1+1}^1(a_1)$ tal que $y \leq t$, existe $z \in R_{J_1+2}^1(\langle J_1 + 2, a, y\rangle)$ tal que $z \leq s$ y $M \models \neg\psi(J_1 + 1, y, z)$.

Entonces para que se cumpla $K_1 \models \theta_1(J_1) \wedge \neg\theta_1(J_1+1)$, se define K_1 de modo que esté contenido en $R^1_{J_1+1}(a_1)$ y permita realizar cálculos en tiempo $T^1_{J_1+2}$:

$$K_1 = \{\ y < 2^{2^{||a||.|a|^\omega_m}} : \exists e < |r|^\omega\ \exists T < \omega.r^2.T^1_{J_1+2}\ C(e,T,a_1,y)\}\ .$$

Claramente $K_0 \subset_{L_2} K_1$ y K_1 es cerrado bajo funciones en FP (lema 1). Para probar que $K_1 \subset R$ (lema 4) usamos el hecho de que Π_1 puede codificarse mediante algún $p_1 < |r|^\omega$ (lema 2), y calcular a_1 en menos de $r^2.T^1_0$ etapas (lema 3).

Ahora considere θ_2-IND *hasta* l_2, el siguiente axioma en la enumeración cuyos parámetros estén en K_1. Se quiere construir $K_2 \supset_{L_2} K_1$ de modo que satisfaga ese axioma y preserve $\theta_1(J_1)\wedge\neg\theta_1(J_1+1)$. El nuevo axioma será satisfecho al construir K_2 de la misma manera que K_1, reemplazando a, θ_1, l_1 por a_1, θ_2, l_2 y la sucesión T^1_i por otra sucesión T^2_i. Como se explicó arriba, $\theta_1(J_1) \wedge \neg\theta_1(J_1+1)$ será preservado si $K_2 \subset R_{J_1+1}(a_1)$ y K_2 permite cálculos en tiempo $T^1_{J_1+2}$. Para esto se escoge la nueva sucesión de tiempos de cálculo T^2_i entre $T^1_{J_1+1}$ y $T^1_{J_1+2}$: $T^2_j = T^1_{J_1+1}/2^{(j+1)||a||.|a|^{r/4}_m}$.

Sea Π_2 un programa similar a Π_1, con entrada a_1, y con θ_2-IND *hasta* l_2 y T^2_i en lugar de θ_1-IND *hasta* l_1 y T^1_i. Sea $a_2 = \langle J_2+1, a_1, y_{J_2}\rangle$ la salida del programa y sea $K_2 = \{\ y < 2^{2^{||a||.|a|^\omega_m}} : \exists e < |r|^\omega\ \exists T < \omega.r^2.T^2_{J_2+2}\ C(e,T,a_2,y)\}$. Se prueba como antes que $K_1 \subset_{L_2} K_2$, K_2 es cerrado bajo funciones en FP, $K_2 \subset R$ y $K_2 \models \theta_1$-IND *hasta* l_1 $\wedge$ θ_2-IND *hasta* l_2.

Siguiendo con este procedimiento se obtienen $K_3, K_4, \ldots$. La construcción se puede iterar ω veces pues siempre existe una sucesión adecuada de tiempos T^{n+1}_i, $i = 0,\ldots,l_{n+1}+2$, comprendida entre $T^n_{J_n+1}$ y $T^n_{J_n+2}$. Basta definir

$$T^{n+1}_j = T^n_{J_n+1}/2^{(j+1)||a||.|a|^{r/2^n}_m}\ .$$

Finalmente $K^* = \bigcup_{n<\omega} K_n$ es el modelo buscado . □

4 Pruebas de los lemas

Lema 1 *Sea M un modelo de Th($\mathbb{N}$) no estándar, numerable, y sean a, r elementos no estándar tales que $r < ||a||$. Sea $T_0 > 2^{||a||.|a|^\omega_m}$. Sea $K = \{\ y < 2^{2^{||a||.|a|^\omega_m}} : \exists e < |r|^\omega\ \exists T < \omega.r^2.T_0\ C(e,T,a,y)\}$. Entonces K es FP-cerrado.*

Prueba Sea $y \in K$ y sea $f \in FP$. Sea $k, l \in \mathbb{N}$ y $e < |r|^k$, $T < l.r^2.T_0$ tal que $C(e,T,a,y)$. Sea $z = \{f\}(y)$. Como $y < 2^{2^{||a||.|a|^\omega_m}}$ y $f \in FP$, entonces $z < 2^{2^{||a||.|a|^\omega_m}}$. Tenemos que $z = \{f\}(\{e\}(a))$ y la composición de estas dos funciones posee un código menor que $|r|^\omega$, pues $e < |r|^k$ y f es estándar. Por otro lado el tiempo de cálculo al componer es inferior a $l.r^2.T_0 + 2^{||a||.|a|^\omega_m} < (l.r^2+1).T_0 < \omega.r^2.T_0$. Por lo tanto $z \in K$. □

Lema 2 Π_n *se puede codificar por un $p_n < |r|^\omega$.*

Prueba Como $r = 2^{|r|-1}$, la primera línea se puede ejecutar mediante un programa de código inferior a $|r|^\omega$. Igualmente para los parámetros de la fórmula θ_n-IND *hasta* l_n, pues estos pertenecen a K_{n-1}. Los tiempos T_j^n se calculan a partir de a y r mediante un programa estándar. Teniendo r y los demás parámetros de las reservas $R_j^n(\langle j, a, y_{j-1}\rangle)$, se puede generar sus elementos mediante programas estándar. Igualmente, con programas estándar se evalúa la validez de las fórmulas Δ_0^b. Como el código de la composición de programas está esencialmente acotado por el producto de los códigos, podemos concluir. □

Lema 3 *El tiempo de cálculo de a_n por Π_n es menor o igual a $r^2 \cdot T_0^n$.*

Prueba El cálculo de r toma tiempo $|r|$. Los parámetros de θ_n-IND *hasta* l_n, se calculan en tiempo estrictamente menor que $\omega.r^2.T_{J_n+2}^n$ pues pertenecen a K_{n-1}. Los tiempos T_j^n se calculan en tiempo $2^{||a||.|a|_m^\omega}$ pues están acotados por $2^{2^{||a||.|a|_m^\omega}}$ y son calculados por programas estándar. Los elementos de las reservas $R_j^n(\langle j, a, y_{j-1}\rangle)$ se calculan en tiempo estrictamente inferior a T_j^n. La validez de las fórmulas Δ_0^b se realiza en tiempo inferior a $2^{||a||.|a|_m^\omega}$, pues se hace en tiempo polinomial y los parámetros en los cuales se evalúan, o bien pertenecen a K_{n-1}, o bien están acotados por términos del leguaje aplicados a elementos de K_{n-1} (en el caso de los elementos de las reservas buscados por el programa). Tenemos entonces que el tiempo de cálculo está acotado superiormente por

$$\omega.r^2.T_{J_n+2}^n + 2^{||a||.|a|_m^\omega} + \sum_{j=0}^{l_n} r.(T_j^n + 2^{||a||.|a|_m^\omega} + r.(T_{j+1}^n + 2^{||a||.|a|_m^\omega}))$$

Como se tiene $T_j^n > 2^{||a||.|a|_m^\omega}$ para todo j, n, el tiempo es inferior a

$$\omega.r^2.T_{J_n+2}^n + \sum_{j=0}^{l_n} r.(2.T_j^n + r.(2.T_{j+1}^n))$$

Pero $2r.T_{j+1}^n < T_j^n$ y $\omega.r^2.T_{J_n+2}^n < T_0^n$ pues $r < ||a||$. Luego, tenemos que el tiempo no excede

$$T_0^n + 3r.\sum_{j=0}^{l_n} T_j^n < (3r+1).T_0^n + 3r.l_n.T_1^n$$

pues la sucesión T_j^n es decreciente. Ahora, como $l_n < 2^{||a||.|a|_m}$ y $r < ||a||$, se tiene que $3r.l_n.T_1^n < T_0$. Entonces el tiempo es inferior a

$$(3r+2).T_0^n < r^2.T_0^n.$$

□

Lema 4 *Para todo $n \geq 1$, $K_n \subset R$.*

Prueba Observe que a_n se calcula a partir de a al componer los programas $\Pi_1, \ldots, \Pi_n$. Por los lemas anteriores esta composición equivale a una función de código $< |r|^\omega$ y el tiempo de cálculo es a lo sumo $\sum_{k=1}^{n} r^2 \cdot T_0^k < r^3 \cdot T_0^1$. Sea $y \in K_n = \{\, y < 2^{2^{||a||.|a|_m^\omega}} : \exists e < |r|^\omega \; \exists T < \omega.r^2.T^n_{J_n+2} \; C(e,T,a_n,y)\}$. Entonces y se calcula a partir de a por un programa de código inferior a $|r|^\omega < r$ en tiempo menor a $r^3 \cdot T_0^1 + \omega.r^2.T^n_{J_n+2} < 2r^3 \cdot T_0^1 < 2^{||a||.|a|_m^r}$. Por lo tanto $y \in R$. $\square$

Referencias

[1] Bloch, S. (1997) "On parallel hierarchies and R_k^i", *Ann. Pure Appl. Logic* **89**(2–3): 231–273.

[2] Buss, S. (1986) *Bounded Arithmetic*. Bibliopolis, Naples.

[3] Chinchilla, E. (1998) "A model theoretic proof of a subexponential time witnessing theorem", *Comptes Rendus de l'Académie de Sciences de Paris Sér. I Math.* **326** (8): 913–917.

[4] Hájek, P.; Pudlák, P. (1993) *Metamathematics Of First-Order Arithmetic.* Springer-Verlag, Berlin.

[5] Stockmeyer, L. (1976) "The polynomial-time hierarchy", *Theoretical Computer Science* **3** (1): 1–22.