

Un método para eliminar ambigüedad en conjuntos de patrones en sistemas de reescritura

Jorge F. Salas O.

Universidad Central de Venezuela

Escuela de Computación

Apartado 47002, Caracas 1041, Venezuela

E-mail: jsalas@isys.ciens.ucv.ve

Resumen

Un patrón es un molde que describe un conjunto de términos con los cuales el patrón se aparea. Un conjunto de patrones es ambigüo si existen al menos dos patrones en el conjunto que se aparean con un mismo término. En la regla de prioridad textual, los patrones que aparecen textualmente primero tienen preferencia para el apareamiento con los términos. En este trabajo presentamos un método para eliminar la ambigüedad en un conjunto de patrones preservando el particionamiento inducido por el conjunto ambigüo cuando el apareamiento se realiza con prioridad textual. Adicionalmente, nuestro método permite decidir la incompletitud del conjunto inicial de patrones, la existencia de patrones inútiles y encontrar conjuntos de cobertura que tienen aplicación en pruebas de teoremas por inducción por reescritura.

Palabras claves: Patrones, patrones ambigüos, regla de prioridad textual.

1 Introducción

En los lenguajes funcionales que admiten paso de parámetros por apareamiento de patrones como, por ejemplo, Caml [Ler97], Sml [Pau96] y Haskell [Hud90], se permite que el programador utilice conjuntos de patrones ambigüos e incompletos en las definiciones funcionales. Esto facilita la programación sobre todo cuando la firma del tipo de los parámetros es extensa. Es tarea de los procesadores de esos lenguajes resolver la ambigüedad en los conjuntos de patrones, generar estrategias de apareamiento y emitir advertencias en el caso de incompletitud [PS93a, PS93b]. Lo primero, normalmente se logra a través de una pragmática en el uso del lenguaje, por ejemplo, la regla de prioridad textual [Lav91], donde el programador escribe primero los casos más definidos, luego los más generales y la manera de resolver la ambigüedad es tratar de aparear con el argumento prioritariamente los patrones que aparecen primero textualmente.

Tales convenciones pragmáticas en el uso de lenguajes de programación no pueden trasladarse directamente a formalismos más abstractos como los sistemas de reescritura [Klo92, CJ97, BN98],

donde esencialmente las reglas son aplicadas no determinísticamente a los términos que se reescriben. Por lo tanto, si queremos modelar las computaciones de un programa funcional (esencialmente determinístico) por medio de un sistema de reescritura [Sal00], previamente debemos eliminar cualquier ambigüedad presente en los conjuntos de patrones de las definiciones funcionales antes de tratar de formular el correspondiente sistema de reescritura modelo.

Por ejemplo, considere el siguiente programa Caml:

```
type tree = N of tree * tree | H;;
let rec flat = function N(N(x,y),z) → flat (N(x,N(y,z)))
                | N(x,y) → N(x, flat y)
                | x → x;;
```

El conjunto de patrones $\{N(N(x,y),z), N(x,y), x\}$ es ambigüo ya que, por ejemplo, el término $N(N(H,H),H)$ se puede aparear (ignorando la convención de prioridad textual) con cualquiera de los patrones. De igual forma, el término $N(H,H)$ se puede aparear con el segundo o con el tercer patrón. Más aún, cualquier término que se pueda aparear con alguno de los dos primeros patrones, también se podrá aparear con el tercero. El correspondiente sistema de reescritura asociado después de eliminar la ambigüedad es:

$$\begin{aligned} flat(N(N(x,y),z)) &\rightarrow flat(N(x,N(y,z))) \\ flat(N(H,y)) &\rightarrow N(H, flat(y)) \\ flat(H) &\rightarrow H \end{aligned}$$

Note como el segundo patrón ha cambiado de $N(x,y)$ a $N(H,y)$ por medio de la sustitución $\{x \mapsto H\}$ y el tercer patrón han cambiado de x a H por medio de $\{x \mapsto H\}$, obteniéndose un sistema equivalente pero no ambigüo. Además, en ambos casos (programa funcional y sistema de reescritura) el conjunto de patrones es completo, es decir, para cualquier término cerrado de tipo *tree*, existe al menos un patrón que se aparear con él.

Podemos expresar el problema de eliminar ambigüedad en un conjunto de patrones así: dado un conjunto (posiblemente ambigüo) de patrones $\Pi = \{p_1, \dots, p_n\}$, se desea hallar un conjunto (posiblemente vacío) de sustituciones $\sigma = \{\sigma_1^j, \dots, \sigma_{i_j}^j / i_j \geq 0, j = 1, \dots, n\}$ tales que el conjunto de patrones $\Pi' = \{p_1\sigma_1^1, p_1\sigma_2^1, \dots, p_1\sigma_{i_1}^1, p_2\sigma_1^2, \dots, p_n\sigma_{i_n}^n\}$ obtenido al aplicar las sustituciones σ no sea ambigüo. Note que a cada patrón p_i del conjunto inicial Π corresponde un subconjunto no ambigüo de patrones $\{p_1\sigma_1^1, \dots, p_1\sigma_{i_1}^1\} \subseteq \Pi'$ que se deben aparear con los mismos términos cerrados que p_i para conservar el particionamiento inducido por Π usando la regla de prioridad textual.

2 Definiciones básicas y notación

Definición 1 (Términos, términos cerrados) Sean $\mathcal{V}$ un conjunto enumerable de elementos denominados variables y $\mathcal{F}$ un conjunto enumerable de elementos denominados símbolos de función, con $\mathcal{F} \cap \mathcal{V} = \emptyset$, dotado de una función de aridad $a : \mathcal{F} \rightarrow \mathbb{N}$. El conjunto $\mathcal{T}$ de términos está definido como la $\mathcal{F}$ -álgebra libre generada por $\mathcal{V}$. Es decir, un término es una variable o es de la forma $F(t_1, \dots, t_n)$ para algunos $F \in \mathcal{F}_n = \{F \in \mathcal{F} / a(F) = n\}$ y $t_1, \dots, t_n \in \mathcal{T}$. Si un término no contiene variable diremos que es cerrado.

Definición 2 (Orden prefijo) Sea $\mathbb{N}_+^*$ el conjunto de secuencias de enteros positivos, Λ la secuencia vacía en $\mathbb{N}_+^*$ y $\cdot$ la operación de concatenación de secuencias. $\mathbb{N}_+^*$ está dotado de la relación de orden $\sqsubseteq_{pre}$ definida por: $\forall p, q \in \mathbb{N}_+^*, p \sqsubseteq_{pre} q$ ssi $\exists r \in \mathbb{N}_+^* [p \cdot r = q]$. Diremos que p es un prefijo de q cuando $p \sqsubseteq_{pre} q$.

Definición 3 (Conjunto de posiciones, subtérminos, profundidad) Para cualquier término $t \in \mathcal{T}$, definimos su conjunto de posiciones $Pos(t) \subseteq \mathbb{N}_+^*$ y el subtérmino de t en o , $t|_o \in \mathcal{T}$, para cualquiera $o \in Pos(t)$, de la manera siguiente: i) si $t = x \in \mathcal{V}$ entonces $Pos(t) = \{\Lambda\}$ y $t|_\Lambda = t$; ii) si $t = f(t_1, \dots, t_n)$, entonces $Pos(t) = \{\Lambda\} \cup \{i \cdot o \mid 1 \leq i \leq n, o \in Pos(t_i)\}$, $t|_\Lambda = t$ y $t|_{i \cdot o} = t_i|_o$. Diremos que el subtérmino $t|_o$ tiene una ocurrencia en t en la posición o . Si t es un término, su profundidad es la longitud de su más larga posición: $Prof(t) = \max\{|o| \mid o \in Pos(t)\}$.

Definición 4 (Constructores, tipos) Un constructor es un símbolo funcional definido en una declaración de tipo: $T = A_1(t_1^1, \dots, t_{a_1}^1) \mid \dots \mid A_n(t_1^n, \dots, t_{a_n}^n) \mid B_1 \mid \dots \mid B_m$, donde T es el identificador del tipo declarado, A_i son constructores de aridad a_i ($i = 1, \dots, n$), B_j constructores de aridad 0 ($j = 1, \dots, m$) y los t_k^i son identificadores de tipos ($i = 1, \dots, n; k = 1, \dots, a_i$). En este trabajo suponemos que todo término t es de algún tipo T y denotamos $t : T$ esta relación. Al conjunto de términos cerrados de tipo T lo denotaremos T_Σ .

Definición 5 (Sustitución) Una sustitución σ es un endomorfismo en el conjunto de términos, es decir, una función de $\mathcal{T}$ en $\mathcal{T}$ tal que $\sigma(F(t_1, \dots, t_n)) = F(\sigma(t_1), \dots, \sigma(t_n))$. Por lo tanto, una sustitución está totalmente definida por sus valores sobre las variables. La aplicación de una sustitución σ a un término t la denotaremos $t\sigma$.

Definición 6 (Instanciación válida) Sea t un término. Diremos que la sustitución σ es válida para t si t y $t\sigma$ son del mismo tipo. Si σ es una sustitución válida para t diremos que $t\sigma$ es una instancia válida de t . La obtención de instancias válidas la llamaremos instanciación válida.

Definición 7 (Igualdad hasta renombramiento de variables) Sean t_1 y t_2 dos términos del mismo tipo. Diremos que t_1 es igual a t_2 hasta renombramiento de variables, $t_1 \approx t_2$, si hay una sustitución σ tal que $t_1\sigma = t_2$ y σ consiste sólo de sustituciones de variables por variables.

Definición 8 (La relación de apareamiento $\preceq$) El casiordenamiento¹ $\preceq$ sobre términos está definido por $t \preceq t'$ si existe una sustitución válida σ tal que $t\sigma = t'$. Si $t \preceq t'$ diremos que t se aparece con t' , o que t es más general que t' , o que t' está más definido que t .

Definición 9 (Patrones) Un patrón es un término lineal, es decir, un término en el cual ninguna variable aparece más de una vez. Un patrón puede ser visto como representante de un conjunto de términos que comparten un prefijo común, los subtérminos localizados bajo este prefijo común están ligados a las variables del patrón.

Definición 10 (Patrones ambiguos) Un conjunto de patrones es ambiguo si existen al menos dos patrones distintos en el conjunto que se aparecen con un mismo término.

¹Un casiordenamiento es una relación reflexiva y transitiva.

Definición 11 (Conjunto cubierto) Sea Π un conjunto de patrones de tipo T . El conjunto cubierto por Π , denotado $\llbracket \Pi \rrbracket$, es el conjunto de términos cerrados que pueden ser obtenidos a partir de algún patrón en Π vía instanciación válida: $\llbracket \Pi \rrbracket = \{t \in T_\Sigma \mid \exists p \in \Pi [p \preceq t]\}$.

Definición 12 (Conjunto completo de patrones) Un conjunto Π de patrones de tipo T es completo si cubre al conjunto T_Σ de términos cerrados de tipo T , es decir, si $\llbracket \Pi \rrbracket = T_\Sigma$.

Definición 13 (Regla de prioridad textual) Sean $\Pi = \{p_1; \dots; p_n\}$ una secuencia de patrones de tipo T y Q un conjunto de términos de tipo T . Definimos $\mathcal{C}_Q(p_i)$ como el conjunto formado por los términos en Q con los cuales el patrón $p_i \in \Pi$ se aparea bajo la regla de prioridad textual: $\mathcal{C}_Q(p_i) = \{t \in Q \mid p_i \preceq t \wedge \forall j < i [p_j \not\preceq t]\}$ ($i = 1, \dots, n$).

Definición 14 (Secuencia completa de patrones) Sea $\Pi = \{p_1; \dots; p_n\}$ una secuencia de patrones de tipo T . Diremos que Π es completa si cubre al conjunto T_Σ de términos cerrados de tipo T bajo la regla de prioridad textual: $\forall t \in T_\Sigma \exists p_i \in \Pi [t \in \mathcal{C}_{T_\Sigma}(p_i)]$.

Proposición 1 (Particionamiento inducido) Si $Q \subseteq T_\Sigma$ y $\Pi = \{p_1; \dots; p_n\}$ es una secuencia completa de patrones de tipo T entonces $\{\mathcal{C}_Q(p_1), \dots, \mathcal{C}_Q(p_n)\}$ es una partición de Q .

Demostración. Primero mostraremos que $\bigcup \mathcal{C}_Q(p_i) = Q$. Como $\forall p_i \in \Pi [\mathcal{C}_Q(p_i) \subseteq Q]$ tenemos que $\bigcup \mathcal{C}_Q(p_i) \subseteq Q$. Por otra parte, como Π es una secuencia completa de patrones de tipo T y $Q \subseteq T_\Sigma$: $\forall t \in Q \exists p_i \in \Pi [t \in \mathcal{C}_Q(p_i)]$. Luego, $Q \subseteq \bigcup \mathcal{C}_Q(p_i)$. Ahora mostraremos que $\forall i, j \in \{1, \dots, n\} [i \neq j \Rightarrow \mathcal{C}_Q(p_i) \cap \mathcal{C}_Q(p_j) = \emptyset]$. Suponga que $i \neq j$ y $\mathcal{C}_Q(p_i) \cap \mathcal{C}_Q(p_j) \neq \emptyset$. Es decir, $\exists t \in Q [t \in \mathcal{C}_Q(p_i) \wedge t \in \mathcal{C}_Q(p_j) \wedge i \neq j]$. Notando que $t \in \mathcal{C}_Q(p_i) \Rightarrow \forall k < i [t \notin \mathcal{C}_Q(p_k)]$ y $t \in \mathcal{C}_Q(p_j) \Rightarrow \forall k < j [t \notin \mathcal{C}_Q(p_k)]$. Como $i \neq j \Rightarrow i < j \vee j < i$, tenemos claramente una contradicción. ■

Definición 15 (Patrones inútiles, secuencia equitativa) Sea $\Pi = \{p_1; \dots; p_n\}$ una secuencia de patrones de tipo T , diremos que $p_i \in \Pi$ es inútil si $\mathcal{C}_{T_\Sigma}(p_i) = \emptyset$. Diremos que Π es equitativa si es una secuencia completa y no contiene patrones inútiles.

Por ejemplo, si $T = N(T, T) \mid H$ y $\Pi = \{N(x, y); N(H, H); H\}$, entonces $N(H, H)$ es un patrón inútil.

3 Generación de formas

Sea el tipo $T = A_1(t_1^1, \dots, t_{a_1}^1) \mid \dots \mid A_n(t_1^n, \dots, t_{a_n}^n) \mid B_1 \mid \dots \mid B_m$. Definimos los conjuntos f_i^T de formas del tipo T a nivel i de la siguiente manera:

$$\begin{aligned} f_0^T &= \{A_1(x_1, \dots, x_{a_1}), \dots, A_n(y_1, \dots, y_{a_n})\} \cup \mathbf{B} \\ f_{i+1}^T &= \{A(\tilde{\alpha}_1, \dots, \tilde{\alpha}_k) \mid A(x_1, \dots, x_k) \in f_0^T \wedge \alpha_j \in f_i^{\tau(x_j)} (j = 1, \dots, k)\} \cup \mathbf{B} \\ &= \bigcup_{k=1}^n \{A_k(\tilde{\alpha}_1, \dots, \tilde{\alpha}_{a_k}) \mid \alpha_j \in f_i^{t_j^k} (j = 1, \dots, a_k)\} \cup \mathbf{B} \quad (i = 0, 1, \dots) \end{aligned}$$

donde $x_1, \dots, x_{a_1}, \dots, y_1, \dots, y_{a_n}$ son variables nuevas y distintas, $\mathbf{B} = \{B_1, \dots, B_m\}$, $\tilde{\alpha}$ es α con sus variables renombradas y $\tau(x)$ el tipo de x .

Por ejemplo, si $T = N(T, T) \mid H$, entonces

$$\begin{aligned} f_0^T &= \{N(x, y), H\} \\ f_1^T &= \{N(N(x, y), N(z, w)), N(N(x, y), H), N(H, N(x, y)), N(H, H), H\} \end{aligned}$$

Por su construcción, las formas son conjuntos de patrones que presentan propiedades importantes:

Lema 1 *Para cualesquiera nivel i y tipo T , el conjunto de formas f_i^T no es ambigüo:*

$$\forall i \geq 0 \ \forall T \ \forall t : T \ \forall p_1, p_2 \in f_i^T \ [p_1 \preceq t \wedge p_1 \neq p_2 \Rightarrow p_2 \not\preceq t]$$

Demostración. Por inducción natural en el nivel i del conjunto de formas.

Caso $i = 0$. Probaremos que si un patrón $p_1 \in f_0^T$ se aparea con un término $t : T$, entonces no puede haber otro patrón $p_2 \in f_0^T$ que se aparea con t :

$$\forall T \ \forall t : T \ \forall p_1, p_2 \in f_0^T \ [p_1 \preceq t \wedge p_1 \neq p_2 \Rightarrow p_2 \not\preceq t]$$

De acuerdo a la forma de los elementos de f_0^T , habrán dos posibilidades:

Primera, $p_1 = A_l(z_1, \dots, z_{a_l})$ con $z_k : t_k^l$ ($k = 1, \dots, a_l$). En este caso, t debe ser de la forma $A_l(\alpha_1, \dots, \alpha_{a_l})$. Notando que el único patrón en f_0^T con símbolo raíz A_l es p_1 ; y que un patrón sólo se puede aparear con t si su símbolo raíz es A_l , concluimos que $\forall p_2 \ [p_1 \neq p_2 \Rightarrow p_2 \not\preceq t]$.

La otra posibilidad es $p_1 = B_l$. En este caso necesariamente $t = B_l$; y trivialmente se cumple que $\forall p_2 \ [p_2 \neq B_l \Rightarrow p_2 \not\preceq t]$. Por lo tanto, el lema se cumple para $i = 0$.

Hipótesis inductiva: para $i = k$ suponemos que para cualquier tipo T , f_k^T no es ambigüo.

Caso $i = k + 1$. Demostraremos este caso por contradicción. Supongamos que existe un tipo T tal que f_{k+1}^T es ambigüo. Entonces, existen un término $t : T$ y dos patrones $p_1, p_2 \in f_{k+1}^T$ con $p_1 \neq p_2$ tales que $p_1 \preceq t$ y $p_2 \preceq t$. La única posibilidad para que ésto suceda es cuando $t = A_l(\alpha_1, \dots, \alpha_{a_l})$, $p_1 = A_l(\beta_1, \dots, \beta_{a_l})$, y $p_2 = A_l(\gamma_1, \dots, \gamma_{a_l})$. Entonces, para $j = 1, \dots, a_l$:

$$p_1 \preceq t \Rightarrow \exists \sigma_1 \ [p_1 \sigma_1 = t] \Rightarrow \exists \sigma_1 \forall j \ [\beta_j \sigma_1 = \alpha_j] \quad p_2 \preceq t \Rightarrow \exists \sigma_2 \ [p_2 \sigma_2 = t] \Rightarrow \exists \sigma_2 \forall j \ [\gamma_j \sigma_2 = \alpha_j]$$

Por lo tanto, para $j = 1, \dots, a_l$: $\beta_j \sigma_1 = \gamma_j \sigma_2$. Como $p_1 \neq p_2$, existe al menos un j' tal que $\beta_{j'} \neq \gamma_{j'}$. Además, por construcción de f_{k+1}^T , tenemos que $\beta_{j'} \in f_k^{t'}$ y $\gamma_{j'} \in f_k^{t'}$, donde $t' = t_{j'}^l$. Finalmente, como $\beta_{j'} \sigma_1 = \gamma_{j'} \sigma_2 = \alpha_{j'}$ y $\beta_{j'} \neq \gamma_{j'}$, tenemos que existe un tipo t' tal que $f_k^{t'}$ es ambigüo. Con lo cual se contradice la hipótesis inductiva y el lema queda demostrado. ■

Lema 2 *Para cualesquiera nivel i y tipo T , el conjunto de formas f_i^T es completo:*

$$\forall i \geq 0 \ \forall T \ \forall t \in T_\Sigma \ \exists q \in f_i^T \ [q \preceq t]$$

Demostración. Por inducción natural en el nivel i del conjunto de formas.

Note que si $t \in T_\Sigma$, entonces $t = B_l$; o $t = A_l(\alpha_1, \dots, \alpha_{a_l})$ con $\alpha_j \in (t_j^l)_\Sigma$ ($j = 1, \dots, a_l$).

Caso $i = 0$. Si $t = B_l$ entonces $q = B_l \in f_0^T$ y $q \preceq t$. Si $t = A_l(\alpha_1, \dots, \alpha_{a_l})$ entonces $q = A_l(x_1, \dots, x_{a_l}) \in f_0^T$ y $q \preceq t$. En ámbos casos, $q \in f_0^T$ y $q \preceq t$.

Hipótesis inductiva: para $i = k$ suponemos que $\forall T \forall t \in T_\Sigma \exists q \in f_k^T [q \preceq t]$.

Caso $i = k + 1$. Si $t = B_l$ entonces $q = B_l \in f_{k+1}^T$ y $q \preceq t$. Sea ahora $t = A_l(\alpha_1, \dots, \alpha_{a_l})$. Considere el subconjunto $\bar{A}_l$ de las formas de T a nivel $k + 1$ con A_l como constructor principal: $\bar{A}_l = \{A_l(\tilde{\beta}_1, \dots, \tilde{\beta}_{a_l}) / \beta_j \in f_k^{t_j^l} (j = 1, \dots, a_l)\} \subseteq f_{k+1}^T$. Por hipótesis inductiva, sabemos que $\forall \alpha_j \in (t_j^l)_\Sigma \exists \beta'_j \in f_k^{t_j^l} [\beta'_j \preceq \alpha_j]$ ($j = 1, \dots, a_l$). Por lo tanto, $A_l(\beta'_1, \dots, \beta'_{a_l}) \preceq t$. En consecuencia, $A_l(\tilde{\beta}'_1, \dots, \tilde{\beta}'_{a_l}) \preceq t$. Además, $A_l(\tilde{\beta}'_1, \dots, \tilde{\beta}'_{a_l}) \in \bar{A}_l$. Haciendo $q = A_l(\tilde{\beta}'_1, \dots, \tilde{\beta}'_{a_l})$, concluimos que $q \in f_{k+1}^T$ y $q \preceq t$. ■

Proposición 2 Para cualesquiera tipo T , nivel i y patrón $q \in f_i^T$, la profundidad de q es menor o igual a $i + 1$.

Demostración. Por inducción natural en el nivel i .

Caso $i = 0$: $f_0^T = \{A_1(x_1, \dots, x_{a_1}), \dots, A_n(y_1, \dots, y_{a_n}), B_1, \dots, B_m\}$. Claramente, $\forall q \in f_0^T [Prof(q) \leq 1 = i + 1]$.

Caso $i = k$: Suponemos que $\forall T \forall q \in f_k^T [Prof(q) \leq k + 1]$.

Caso $i = k + 1$: $q \in f_{k+1}^T \Rightarrow q = B_j \in \mathbf{B} \vee q = A_l(\tilde{\alpha}_1, \dots, \tilde{\alpha}_{a_l})$ con $\alpha_j \in f_k^{t_j^l}$ ($j = 1, \dots, a_l$). Evidentemente, si $q = B_l$ entonces $Prof(q) = 0 \leq k + 1$. Si $q = A_l(\tilde{\alpha}_1, \dots, \tilde{\alpha}_{a_l})$ como (por hipótesis inductiva) $\forall j = 1, \dots, a_l : Prof(\alpha_j) \leq k + 1$ tenemos que $Prof(q) \leq k + 2 = i + 1$ ya que el renombramiento de variables no altera la profundidad de un término. ■

Proposición 3 Para cualesquiera tipo T , nivel i y patrón $q \in f_i^T$, si o es la posición de una variable $x \in Vars(q)$ entonces $|o| = i + 1$.

Demostración. Por inducción natural en el nivel i .

Caso $i = 0$: los únicos términos en f_0^T con variables son $\{A_1(x_1, \dots, x_{a_1}), \dots, A_n(y_1, \dots, y_{a_n})\}$. Para cada uno de estos términos $A_j(z_1, \dots, z_{a_j})$ ($j = 1, \dots, n$) las posiciones de las variables son $\{1, \dots, a_j\}$ y cada posición tiene longitud $1 = i + 1$.

Caso $i = k$: Suponemos que $\forall T \forall q \in f_k^T \forall o \in Pos(q) [q(o) \in Vars(q) \Rightarrow |o| = k + 1]$.

Caso $i = k + 1$: Si $q \in f_{k+1}^T$ tiene variables, q debe ser de la forma $A_l(\tilde{\alpha}_1, \dots, \tilde{\alpha}_{a_l})$ para algunos $\alpha_j \in f_k^{t_j^l}$, ($j = 1, \dots, a_l$) donde al menos uno de los $\tilde{\alpha}_j$ contiene variables. Para cualquier $\alpha_j \in f_k^{t_j^l}$ que contenga variables, por hipótesis inductiva, las posiciones de éstas tendrán longitud $k + 1$. Por lo tanto, las variables en $q = A_l(\tilde{\alpha}_1, \dots, \tilde{\alpha}_{a_l})$ tendrán posiciones de longitud $k + 2 = i + 1$, porque el renombramiento de variables no altera las longitudes de las posiciones de las variables. ■

Definición 16 (Profundidad máxima de un constructor) La profundidad máxima de un constructor en un término no variable p es la longitud máxima de las posiciones de los subtérminos de p cuya raíz no es un símbolo variable, o 0 si p es una variable:

$$PMC(p) = \begin{cases} 0 & \text{si } p \in \mathcal{V} \\ \text{máx } \{|u| / u \in Pos(p) \wedge p|_u \notin \mathcal{V}\} & \text{si } p \notin \mathcal{V} \end{cases}$$

Proposición 4 Para cualesquiera tipo T , nivel i y patrón $q \in f_i^T$, la profundidad máxima de un constructor en q es menor o igual a i .

Demostración. Por inducción natural en el nivel i .

Caso $i = 0$: q es de la forma B_l o $A_l(x_1, \dots, x_{a_l})$. En ámbos casos $PMC(q) = 0 = i$.

Caso $i = k$: Suponemos que $\forall T \forall q \in f_k^T [PMC(q) \leq k]$.

Caso $i = k + 1$: $q \in f_{k+1}^T \Rightarrow q = B_l \in \mathbf{B} \vee q = A_l(\tilde{\alpha}_1, \dots, \tilde{\alpha}_{a_l})$ con $\alpha_j \in f_k^{t_j^l}$ ($j = 1, \dots, a_l$). Si $q = B_l$ entonces $PMC(q) = 0 \leq k + 1$. Si $q = A_l(\tilde{\alpha}_1, \dots, \tilde{\alpha}_{a_l})$ como (por hipótesis inductiva) $\forall j = 1, \dots, a_l : PMC(\alpha_j) \leq k$ tenemos que $PMC(q) \leq k + 1 = i$ ya que el renombramiento de variables no altera la profundidad máxima de un constructor en un término. ■

Note que como consecuencia de las proposiciones 3 y 4, en un patrón $q \in f_i^T$ sólo las variables pueden estar a profundidad $i + 1$.

Para eliminar la ambigüedad en un conjunto de patrones Π tendremos que conseguir un conjunto de formas cuyos elementos sean instancias de c/u de los patrones en Π . Esta idea está implícita en la siguiente definición.

Definición 17 (Nivel de una secuencia de patrones) Sea $\Pi = \{p_1; \dots; p_n\}$ una secuencia completa de patrones de tipo T . El nivel de Π es el menor entero ω tal que $\forall p_i \in \Pi [\mathcal{C}_{f_\omega^T}(p_i) \neq \emptyset]$. Al conjunto de formas de tipo T al nivel de Π lo abreviaremos f_Π^T y lo llamaremos las formas para Π .

Por ejemplo, para el tipo $T = N(T, T) \mid H$, el nivel de $\Pi = \{N(N(x, y), z); N(x, y); x\}$ es 1 y $f_\Pi^T = \{N(N(x, y), N(z, w)), N(N(x, y), H), N(H, N(x, y)), N(H, H), H\}$.

Proposición 5 Sea Π una secuencia de patrones de tipo T . Si ω es el nivel de Π entonces

$$\forall p_i \in \Pi [PMC(p_i) \leq \omega]$$

Demostración. Por contradicción. Supongamos $\exists p_i \in \Pi [PMC(p_i) > \omega]$. Sea u la posición de un constructor en p_i con $|u| = k > \omega$. Es decir, $p_i|_u = c \in \mathcal{F}$. Como ω es la profundidad de Π , tenemos que $\mathcal{C}_{f_\omega^T}(p_i) \neq \emptyset$. Por lo tanto, $\exists q \in f_\omega^T [p_i \leq q \wedge \forall j < i [p_j \not\leq q]]$. Sea σ tal que $p_i\sigma = q$. Si $q \in f_\omega^T$ tenemos por la proposición 2, que $Prof(q) \leq \omega + 1$. Es decir, $\forall v \in Pos(q) : |v| \leq \omega + 1$. Como $u \in Pos(p_i) \Rightarrow u \in Pos(p_i\sigma)$, tenemos que la única posibilidad para $k > \omega$ es $k = \omega + 1$. Además, consecuencia de las proposiciones 3 y 4, si $u \in Pos(q)$, $|u| = \omega + 1$ y $q \in f_\omega^T$ entonces $q|_u = x \in Vars(q)$. Luego, $p_i|_u = c \in \mathcal{F}$ y $q|_u = x \in Vars(q)$. Claramente, no existe sustitución σ tal que $(p_i|_u)\sigma = q|_u$. En consecuencia, $\mathcal{C}_{f_\omega^T}(p_i) = \emptyset$ y ω no es el nivel de Π . Lo cual contradice la hipótesis de la proposición. ■

Definición 18 (Formas para un patrón) Sea $\Pi = \{p_1; \dots; p_n\}$ una secuencia de patrones de tipo T . Definimos las formas asociadas al patrón $p_i \in \Pi$ como el subconjunto de las formas para Π con cuyos elementos se aparea p_i bajo la regla de prioridad textual, es decir, $\mathcal{C}_{f_\Pi^T}(p_i)$.

Por ejemplo, para los T y Π del ejemplo anterior, $\mathcal{C}_{f_\Pi^T}(x) = \{H\}$.

Lema 3 (Corrección del método de formas) Sea $\Pi = \{p_1; \dots; p_n\}$ una secuencia equitativa de patrones de tipo T . El conjunto de términos cerrados de tipo T con los cuales $p_i \in \Pi$ se aparea bajo la regla de prioridad textual es igual al conjunto cubierto por las formas asociadas a p_i :

$$\forall p_i \in \Pi [\mathcal{C}_{T_\Sigma}(p_i) = \llbracket \mathcal{C}_{f_\Pi^T}(p_i) \rrbracket]$$

Demostración. a) $\forall p_i \in \Pi [\mathcal{C}_{T_\Sigma}(p_i) \subseteq \llbracket \mathcal{C}_{f_\Pi^T}(p_i) \rrbracket]$. Por contradicción. Suponga $\exists p_i \in \Pi \exists t \in T_\Sigma$ tales que $t \in \mathcal{C}_{T_\Sigma}(p_i) \wedge t \notin \llbracket \mathcal{C}_{f_\Pi^T}(p_i) \rrbracket$. Es decir, $p_i \preceq t \wedge \forall j < i [p_j \not\preceq t] \wedge \forall q \in \mathcal{C}_{f_\Pi^T}(p_i) [q \not\preceq t]$. Como f_Π^T es completo y $\{\mathcal{C}_{f_\Pi^T}(p_1), \dots, \mathcal{C}_{f_\Pi^T}(p_n)\}$ es una partición de f_Π^T tenemos: $\exists j \neq i [t \in \llbracket \mathcal{C}_{f_\Pi^T}(p_j) \rrbracket]$. Note que $q \in \mathcal{C}_{f_\Pi^T}(p_j) \wedge q \preceq t \Rightarrow p_j \preceq t$. Como $t \in \mathcal{C}_{f_\Pi^T}(p_i)$, tenemos que $j < i$ no es posible. Por lo tanto, tenemos que $\forall q \in \mathcal{C}_{f_\Pi^T}(p_i) [q \not\preceq t] \wedge \exists j > i \exists q' \in \mathcal{C}_{f_\Pi^T}(p_j) [q' \preceq t]$. Observe que ni q ni p_i pueden ser términos cerrados: si q es un término cerrado, entonces $q = t$ y $p_i \preceq q$; si p_i es un término cerrado, $p_i = t$ y $\mathcal{C}_{f_\Pi^T}(p_i) = \{t\}$ (porque no hay patrones inútiles), lo cual contradice la suposición que $\forall q \in \mathcal{C}_{f_\Pi^T}(p_i) [q \not\preceq t]$. Además, p_i tampoco puede ser una variable. Sean $Vars(p_i) = \{x_1, \dots, x_r\}$ y $Vars(q) = \{y_1, \dots, y_s\}$ las variables de p_i y q , con ocurrencias en las posiciones $O_{p_i} = \{o_{x_1}, \dots, o_{x_r}\}$ y $O_q = \{o_{y_1}, \dots, o_{y_s}\}$, respectivamente. Observe que no es posible $O_q \subseteq O_{p_i}$ porque si en las únicas posiciones donde hay variables en q hay variables en p_i , entonces $p_i \preceq q$ y $q \in \mathcal{C}_{f_\Pi^T}(p_i)$. Por lo tanto, $O_q - O_{p_i} \neq \emptyset$. Es decir, debe existir una posición o en q tal que $q|_o$ es una variable y $o \notin Pos(p_i)$ ó $p_i|_o$ no es una variable. Si ω es el nivel de Π , por la proposición 3, $|o| = \omega + 1$. Como $\forall p_i \in \Pi [PMC(p_i) \leq \omega]$ (proposición 4), tenemos que si $o \in Pos(p_i)$ entonces $p_i|_o$ es una variable. Luego, la única posibilidad es $o \notin Pos(p_i)$. Note que si $o \in Pos(q)$ y $q \preceq t$ entonces $o \in Pos(t)$. Como $p_i \preceq t$ y $o \in Pos(t)$ y $o \notin Pos(p_i)$ entonces $\exists o' \in Pos(p_i)$ tal que $o' \sqsubseteq_{pre} o$ y $p_i|_{o'} \in Vars(p_i)$. Por lo tanto, $p_i|_{o'} \preceq q|_{o'}$. Como ésto ocurre para toda $o \in O_q - O_{p_i}$, concluimos que $p_i \preceq q$ y $q \in \mathcal{C}_{f_\Pi^T}(p_i)$. Con lo cual hemos arribado a una contradicción.

b) $\forall p_i \in \Pi [\llbracket \mathcal{C}_{f_\Pi^T}(p_i) \rrbracket \subseteq \mathcal{C}_{T_\Sigma}(p_i)]$. Por contradicción. Suponga $\exists p_i \in \Pi \exists t \in T_\Sigma$ tal que $t \in \llbracket \mathcal{C}_{f_\Pi^T}(p_i) \rrbracket \wedge t \notin \mathcal{C}_{T_\Sigma}(p_i)$. Note que, $t \in \llbracket \mathcal{C}_{f_\Pi^T}(p_i) \rrbracket \Rightarrow p_i \preceq t$ y $t \notin \mathcal{C}_{T_\Sigma}(p_i) \Rightarrow p_i \not\preceq t \vee \exists j < i [p_j \preceq t]$. Por lo tanto, la única posibilidad es $\exists j < i [p_j \preceq t]$. Tomemos el menor de esos j 's, si tales j 's existen, es decir, $t \in \mathcal{C}_{T_\Sigma}(p_j)$. Por la parte (a): $t \in \mathcal{C}_{T_\Sigma}(p_j) \Rightarrow \exists q' \in \mathcal{C}_{f_\Pi^T}(p_j) [q' \preceq t] \Rightarrow \exists q' \in f_\Pi^T [q' \preceq t]$. Además, $j < i \Rightarrow p_i \neq p_j \Rightarrow \mathcal{C}_{f_\Pi^T}(p_i) \cap \mathcal{C}_{f_\Pi^T}(p_j) = \emptyset \Rightarrow q \neq q'$. Luego, $\exists q, q' \in f_\Pi^T [q \preceq t \wedge q' \preceq t \wedge q \neq q']$. Lo cual es una contradicción porque el conjunto de formas f_Π^T no es ambigüo. ■

Los lemas anteriores aseguran que si Π es una secuencia completa de patrones ambigüos de tipo T , las formas para Π son una secuencia equivalente pero no ambigüa. A menudo es posible reducir f_Π^T de manera tal que se conserve su poder de expresividad y su característica de no ambigüedad.

4 Optimización de patrones

Sea p un patrón de tipo T . Sea $Vars(p) = \{x_1, \dots, x_m\}$, las variables de p . Consideremos el conjunto de sustituciones $\{\sigma_1, \dots, \sigma_n\}$ sobre p . Suponga que en todas las sustituciones $\sigma_i = \{x_1 \mapsto \alpha_1^i, \dots, x_m \mapsto \alpha_m^i\}$ ($i = 1, \dots, n$), los términos asociados a las variables $x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_m$ son iguales, para un cierto k fijo ($1 \leq k \leq m$): $\forall j \neq k [\alpha_j^1 = \alpha_j^2 = \dots = \alpha_j^n]$ ($j = 1, \dots, m$). Es decir, si llamamos σ'_i a σ_i con la sustitución para x_k eliminada, tenemos que $\sigma'_1 = \sigma'_2 = \dots = \sigma'_n =$

σ' . Note que σ' y σ_i ($i = 1, \dots, n$) afectan a las variables $Vars(p) - \{x_k\}$ de forma idéntica. En estas condiciones, tenemos el siguiente

Lema 4 (Corrección de la optimización de sustituciones) *Si las sustituciones para x_k son las formas a nivel 0 para $\tau(x_k)$ hasta renombramiento de variables, $\{\alpha_k^1, \dots, \alpha_k^n\} \approx f_0^{\tau(x_k)}$, entonces los conjuntos $\Pi = \{p\sigma_1, \dots, p\sigma_n\}$ y $\Pi' = \{p\sigma'\}$ cubren los mismos términos cerrados.*

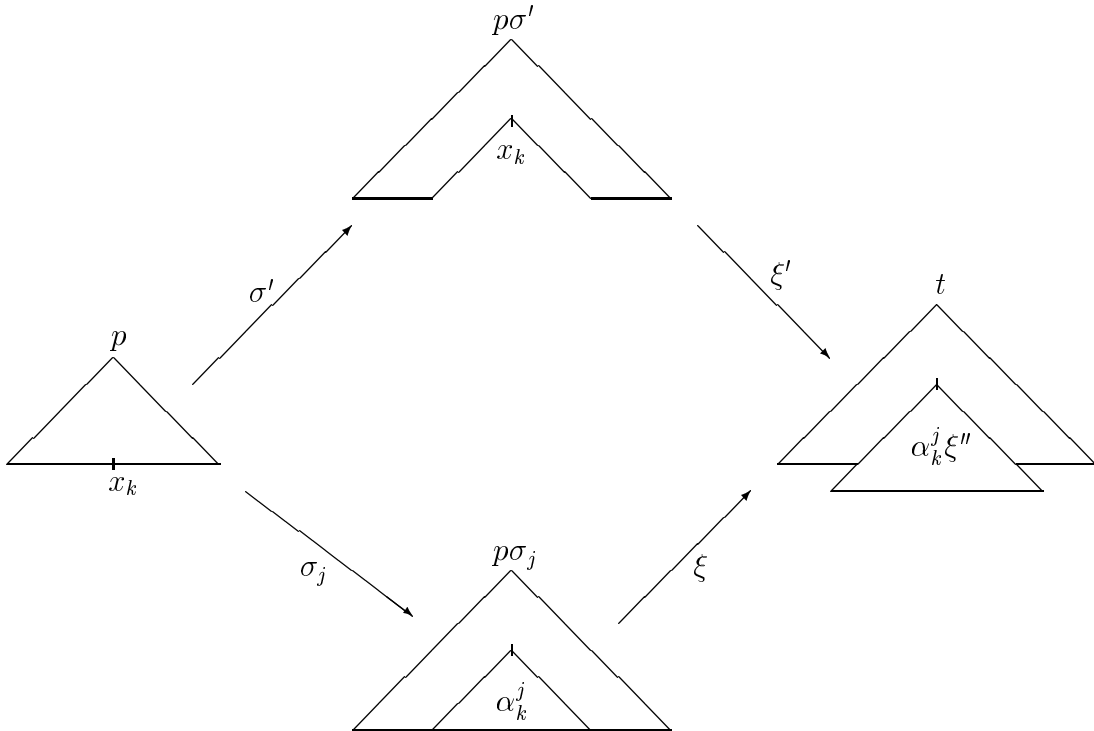
Demostración. La prueba que $\llbracket \Pi \rrbracket = \llbracket \Pi' \rrbracket$ se hará en dos partes.

a) $\llbracket \Pi \rrbracket \subseteq \llbracket \Pi' \rrbracket$. En este caso hay que demostrar $\forall t \in T_\Sigma[\exists j \exists \xi[p\sigma_j \xi = t] \Rightarrow \exists \xi'[p\sigma' \xi' = t]]$. Para un $t \in T_\Sigma$ cualquiera, suponga que existen j y ξ tales que $p\sigma_j \xi = t$. Procedamos ahora a construir la sustitución ξ' . Como σ_j y σ' afectan a todas las variables excepto a x_k de forma idéntica, haremos que ξ' coincida con ξ para las variables $Vars(p\sigma') - \{x_k\}$. Para x_k , haremos $\xi'(x_k) = x_k \sigma_j \xi = \alpha_k^j \xi$. De esta forma hemos definido completamente una sustitución ξ' :

$$x^{\xi'} = \begin{cases} x\xi & \text{si } x \in Vars(p\sigma') - \{x_k\} \\ \alpha_k^j \xi & \text{si } x = x_k \end{cases}$$

tal que $p\sigma' \xi' = t$. Por lo tanto, $\llbracket \Pi \rrbracket \subseteq \llbracket \Pi' \rrbracket$.

b) $\llbracket \Pi' \rrbracket \subseteq \llbracket \Pi \rrbracket$. Hay que demostrar $\forall t \in T_\Sigma[\exists \xi'[p\sigma' \xi' = t] \Rightarrow \exists j \exists \xi[p\sigma_j \xi = t]]$. Para un $t \in T_\Sigma$ cualquiera suponga que existe una sustitución ξ' tal que $p\sigma' \xi' = t$. Procedamos ahora a determinar el índice j y la sustitución ξ . Note que en σ' no hay sustitución para x_k y que en ξ' sí debe haber (pues t es un término cerrado). Luego, $x_k \xi' \in \tau(x_k)_\Sigma$. Como por hipótesis, $\{\alpha_k^1, \dots, \alpha_k^n\} \approx f_0^{\tau(x_k)}$, se puede aplicar el lema 2 y obtener que $\exists j[\alpha_k^j \preceq x_k \xi']$ ($1 \leq j \leq n$). Es decir, $\exists j \exists \xi''[\alpha_k^j \xi'' = x_k \xi']$ ($1 \leq j \leq n$). Por lo tanto, haremos que ξ coincida con ξ'' para las variables en $Vars(\alpha_k^j)$. Además, como σ' y σ_j coinciden en todo, excepto en la sustitución para x_k , haremos que ξ coincida con ξ' para las variables en $Vars(p\sigma') - \{x_k\}$.



Como $Vars(p\sigma_j) = (Vars(p\sigma') - \{x_k\}) \oplus Vars(\alpha_k^j)$, hemos hallado un j ($1 \leq j \leq n$) y se ha definido completamente una sustitución ξ sobre $p\sigma_j$:

$$x\xi = \begin{cases} x\xi' & \text{si } x \in Vars(p\sigma') - \{x_k\} \\ x\xi'' & \text{si } x \in Vars(\alpha_k^j) \end{cases}$$

tales que $p\sigma_j\xi = t$. Por lo tanto, $[[\Pi']] \subseteq [[\Pi]]$. ■

5 Descripción del método

Sean $\Pi = \{p_1; \dots; p_n\}$ la secuencia inicial de patrones (posiblemente ambiguos) de tipo T y Π' el conjunto de patrones finales. Π' se obtiene a partir de Π a través de los siguientes pasos:

1. Determinar f_Π^T las formas para Π .
2. Calcular $\mathcal{C}_{f_\Pi^T}(p_i) = \{q_1^i, \dots, q_{r_i}^i\}$, el subconjunto de formas para Π con las cuales se aparea p_i bajo la regla de prioridad textual ($i = 1, \dots, n$).
3. Obtener $\sigma_i = \{\sigma_j^i \mid p_i\sigma_j^i = q_j^i \ (j = 1, \dots, r_i)\}$, las sustituciones para p_i ($i = 1, \dots, n$).
4. Hallar $\bar{\sigma}_i = \{\bar{\sigma}_1^i, \dots, \bar{\sigma}_{\bar{r}_i}^i\}$, las sustituciones optimizadas para el patrón p_i ($i = 1, \dots, n$).
5. Finalmente, $\Pi' = \{p_1\bar{\sigma}_1^1; \dots; p_1\bar{\sigma}_{\bar{r}_1}^1; \dots; p_n\bar{\sigma}_1^n; \dots; p_n\bar{\sigma}_{\bar{r}_n}^n\}$.

Para el ejemplo planteado en la introducción: $\Pi = \{N(N(x, y), z), N(x, y), x\}$, el conjunto de formas es $f_\Pi^T = \{N(N(x, y), N(z, w)), N(N(x, y), H), N(H, N(x, y)), N(H, H), H\}$ y el conjunto final es $\Pi' = \{N(N(x, y), z), N(H, y), H\}$. Note que la aplicación del lema 4 reduce significativamente el tamaño del conjunto no ambiguo.

6 Conclusiones y extensiones

Hemos presentado un método que permite asociar a un conjunto ambiguo de patrones un conjunto no ambiguo conservando la misma expresividad del conjunto original. La corrección del método ha sido probada bajo los supuestos que el conjunto inicial es completo y no posee patrones inútiles. El método permite determinar la incompletitud del conjunto inicial ($\exists q \in f_\Pi^T \ \forall p_i \in \Pi \ [p_i \not\leq q]$) y la existencia de patrones inútiles ($\exists p_i \in \Pi \ [\mathcal{C}_{f_\Pi^T}(p_i) = \emptyset]$). Además, en [Sal00] se demuestra que f_Π^T es un conjunto de cobertura para la aplicación de inducción por reescritura [Red90] en la prueba semiautomática de teoremas. El método presentado puede mejorarse tomando en cuenta la forma de los patrones para influir en la composición del conjunto de formas, tal como en [PS93b, Sal00]. De esta manera, el método podría tener campo de aplicación en la compilación de patrones de los lenguajes funcionales. Cabe destacar que el método fue diseñado para su utilización en el contexto de transformar programas funcionales en sistemas de reescritura para la demostración de teoremas por inducción en la teoría ecuacional definida por el sistema de reescritura.

Agradecimientos

Quiero agradecer al Dr. Ascánder Suárez, mi tutor en la Universidad Simón Bolívar, sus valiosos comentarios y sugerencias que en mucho permitieron mejorar la parte formal de la presente investigación. En especial, lo relacionado a la optimización del método al utilizar posiciones junto con constructores para precisar el camino a expandir en los conjuntos de formas.

Referencias

- [BN98] Frank Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
- [CJ97] Hubert Comon and Jean-Pierre Jouannaud. *Les termes en logique et en programmation, Version préliminaire*. Laboratoire de Recherche en Informatique, CNRS et Université de Paris Sud, Orsay, Dec. 1997. <http://www.lri.fr/~jouannau/biblio.html>.
- [Hud90] Paul Hudak. Report on the programming language Haskell. Technical Report YALEU DCS/RR-777, Yale University, New Heaven, 1990. <http://haskell.org/hugs>.
- [Klo92] Jan Willem Klop. Term rewriting systems. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 2–116. Oxford University Press, 1992.
- [Lav91] Alain Laville. Comparison of priority rules in pattern matching and term rewriting. *Journal of Symbolic Computation*, 11:321–347, 1991.
- [Ler97] Xavier Leroy. *The Caml Light system, Documentation and user's manual*. INRIA, Paris, 1997. <http://www.inria.fr/caml>.
- [Pau96] Lawrence Paulson. *ML for the Working Programmer*. Cambridge University Press, 2nd edition, 1996. <ftp://ftp.research.bell-labs.com/dist/smlnj/release/110/>.
- [PS93a] Laurence Puel and Ascánder Suárez. Compiling pattern matching by term decomposition. *Journal of Symbolic Logic*, 15(1):1–26, 1993.
- [PS93b] Laurence Puel and Ascánder Suárez. Optimal solutions to pattern matching problems. In M.-C. Gaudel and J.-P. Jouannaud, editors, *Proc. of the 4th Int. Joint Conference CAAP/FASE*, volume 668 of *Lecture Notes in Computer Science*, pages 501–518, Orsay, France, April 1993. Springer-Verlag.
- [Red90] Uday S. Reddy. Term rewriting induction. In M. Stickel, editor, *Proc. of the 10th Int. Conference on Automated Deduction*, volume 449 of *Lecture Notes in Computer Science*, pages 162–177, Kaiserslautern, Germany, July 1990. Springer-Verlag.
- [Sal00] Jorge F. Salas O. *Prueba de propiedades de programas funcionales vía sistemas de re-escritura*. Tesis Doctoral, Universidad Simón Bolívar, 2000.