

DEFINICION DE FORMAS NORMALES DE TERMINOS CCS

A. Soriano

Grupo de Computación Paralela

Escuela de Computación

U.C.V.

RESUMEN

Se presenta la descripción de procedimientos para la construcción de formas normales para la congruencia fuerte, la equivalencia y la congruencia observacional de términos regulares de CCS. Las formas normales definidas se caracterizan por corresponder a comportamientos compuestos por el mínimo número de estados y de transiciones necesarios para su expresión. Dichos procedimientos se basan en transformaciones realizadas sobre los sistemas de transiciones que representan los comportamientos descritos por los términos.

INTRODUCCION

Actualmente, con el desarrollo de sistemas distribuidos y la concepción de lenguajes de programación capaces de expresar la concurrencia, es posible construir sistemas cada vez mas poderosos, sin embargo dichos sistemas tienden a ser cada vez mas complejos, de alli que hoy por hoy en la fase de desarrollo de un sistema es indispensable considerar una etapa para la verificación. Dicha verificación puede realizarse via la construcción y comparación de formas normales. Estas formas normales resultan de suma utilidad en la etapa de depuración del sistema puesto que ayudan a su mejor comprensión al presentar un número minimal de estados y transiciones.

En este trabajo se describen procedimientos de construcción de formas normales de terminos regulares de CCS para la congruencia fuerte y la equivalencia y la congruencia observacional. Los procedimientos se basan en la aplicación de transformaciones apropiadas sobre relaciones de transición representando los comportamientos de los términos de CCS y en la noción de bisimulación.

El trabajo consiste de tres partes: en la primera se realiza una descripción del álgebra de CCS así como la definición y axiomatización de la equivalencia y la congruencia observacional, en la segunda se describen los procedimientos de construcción de las formas normales y por último se da una conclusión.

I. CCS: "CALCUL OF COMMUNICATING SYSTEMS"

CCS [Mi 80] es un cálculo para la descripción, la especificación y la verificación de sistemas comunicantes asincrónicos. Este cálculo está basado en dos ideas principales: la observación y la comunicación sincrónica.

Informalmente, en CCS, puede describirse el comportamiento observable de un sistema (noción de observación). Es a partir de esta noción de observación que se definen los criterios de comparación de sistemas (equivalencia y congruencia observacional).

La descripción de un sistema puede hacerse también por composición paralela (operación que se escribe como $\parallel$) de procesos que se comunican entre ellos. La comunicación de procesos se considera como una acción indivisible del sistema (idea de comunicación sincronizada), esta acción se denomina comunicación interna y se denota por τ .

En este trabajo, nos restringimos al álgebra de procesos que representa los comportamientos regulares y sin transmisión de valores. La presentación se divide en tres partes: en la primera se define el álgebra de CCS, en la segunda se da la semántica operacional y en la tercera parte se definen las tres principales relaciones de equivalencia de CCS, ésto es la congruencia fuerte, la equivalencia observacional y la congruencia observacional.

I.1.- Álgebra para CCS.

Sea:

- D , un conjunto de nombres de *acciones* o *puertas de comunicación* cuyos elementos se identifican por letras minúsculas,
- $\sim D$, un conjunto de nombres de puertos llamados *complementarios*, cuyos elementos son los elementos de D precedidos por el símbolo $\sim$,

- una biyección:

$$\alpha \longleftrightarrow \sim\alpha, \text{ donde } \alpha \in D,$$

$$\sim\alpha \in \sim D,$$

se dice que $\sim\alpha$ es el *puerto complementario* de α ,

ésto nos permite definir la interacción entre procesos,

- τ , ($\alpha \in D$ y $\tau \in \sim D$) una etiqueta que representa una *acción no observable* resultante de una comunicación interna entre α y $\sim\alpha$, $\alpha \in D$,

- P , el conjunto de etiquetas de puertos de comunicación,
 $P = D \cup \sim D$,
- P_τ $P_\tau = P \cup \{\tau\}$,
- F , el conjunto de funciones de redefinición f tales que:
 $f: L \subseteq P \rightarrow M \subseteq P$,
 f es una biyección,
 f respeta los complementos definidos por la biyección,
- V , el conjunto de variables de comportamiento,
- las firmas, $\Sigma_1 = (P_\tau, \text{nil}, +)$
 $\Sigma_2 = (P_\tau, \text{nil}, +, \parallel, [], \backslash)$

donde:

nil es una constante,
 $+$ y $\parallel$ son operadores binarios,
 $\forall \alpha \in P_\tau \ \alpha, \forall f \in F [f], \forall \alpha \in P \ \backslash \alpha$, son operadores unarios.

Los operadores $+$ y $\parallel$ son operadores infijos, α es un operador postfijo, $[f]$ y $\backslash \alpha$ son operadores postfijos. El símbolo "." se utiliza como separador después del operador α , por ejemplo se escribe $\alpha.\text{nil}$ y no αnil .

La prioridad es creciente para los operadores $+$ y $\parallel$, $[f]$ y $\backslash \alpha$. Los operadores $+$, $\parallel$, $[f]$ y $\backslash \alpha$ son asociativos a izquierda y el operador α es asociativo a derecha.

Representamos por:

T_f , el álgebra de firma Σ_2 . Los elementos de T_f se llaman *términos finitos*.

Se dice que un término finito está escrito en *forma de suma* si:

- sea $t = \text{nil}$,
- sea $t = \sum_{i \in I} \alpha_i.t_i$ y cada t_i es un término finito escritos en forma de suma.

T_s , es el subconjunto de términos finitos escritos en forma de suma. ($T_s \subseteq T_f$).

$\Pi[\Sigma_1, V]$, el álgebra de firma $\Sigma_1 \cup V$. ($T_s \subseteq \Pi[\Sigma_1, V]$).

$\Pi[\Sigma_2, V]$, el álgebra de firma $\Sigma_2 \cup V$. ($\Pi[\Sigma_1, V] \subseteq \Pi[\Sigma_2, V]$ y $T_f \subseteq \Pi[\Sigma_2, V]$).

Los elementos de $\Pi[\Sigma_1, V]$ y los de $\Pi[\Sigma_2, V]$ son expresiones a partir de las cuales se construyen los términos que definen comportamientos infinitos.

T_r el conjunto de pares de la forma: $u = (t, E)$
 donde $t \in T[\Sigma_2, V]$.

E es un conjunto de ecuaciones de la forma:

$$\{v_i = t_i\}_{i=1, \dots, n} \text{ donde } v_i \in V \text{ y } t_i \in T[\Sigma_1, V] \text{ y } \forall i, j \models j \Rightarrow v_i = v_j.$$

Se dice que t_i es la *definición* de v_i .

Los elementos de T_r definen los comportamientos infinitos. Los pares $(t, \emptyset) \in T_r$ (*términos finitos*) se escriben simplemente t . ($T_r \subseteq T_r$).

Para los términos de T_r se introducen las nociones de *variable útil*, *variable inútil*, *variable libre*, *término cerrado*, *variable no guardada*, *término guardado* y *término bien guardado*. Estas nociones se utilizan para definir los términos regulares.

Para $t \in T[\Sigma_2, V]$, sea $\text{var}(t)$ el conjunto de variables de t .

Para un término $u = (t, E)$, se dice que una variable v es *útil* si y solamente si $v \in \bigcup_{j=0}^{\text{card}(E)} \rho^j(\text{var}(t))$, donde $\rho(X)$ se define como: $\rho(X) = \{z \mid v_i = X \text{ y } v_i = t_i \text{ y } z = \text{var}(t_i)\}$,

ρ^0 es la identidad

$$\rho^{k+1}(X) = \rho(\rho^k(X)), \text{ para } k > 1.$$

Se dice que una variable v es *inútil* en un término $u = (t, E)$ si y solamente si $\exists t_i$ tal que $v = t_i$ pertenece a E y v no es una variable útil del término u .

Se dice que una variable v es *libre* en un término $(t, E) \in T_r$ si y solamente si no existe j tal que $v = t_j \in E$.

Se dice que un término es *cerrado* si y solamente si no contiene variables libres.

Se dice que una variable v *no es guardada* en un término $u = (t, E)$ si y solamente si v aparece en t o en la definición de una variable útil de u sin estar precedida por una acción.

Se dice que un término $u = (t, E)$ es *guardado* si y solamente si toda variable no guardada del término puede reemplazarse por nil o por una expresión de la forma $\sum \alpha_i t_i$, por substitución repetida de variables por sus definiciones.

Se dice que un término $u = (t, E)$ es *bien guardado* si y solamente si él es guardado y además, toda variable guardada por τ , puede ser reemplazada por nil o por una expresión de la forma $\sum_{i=1, \dots, n} \alpha_{i1} \dots \alpha_{ik} t_i \alpha_{k+1} \tau$.

Copyright © 1994 by John Wiley & Sons, Inc.

$$-T_r,$$

Los e

operacional d

relac

$$\alpha.t \xrightarrow{\alpha} t \quad (\text{ACCION})$$

(SELECCION NO DETERMINISTA)

(ID)

$$(1) \quad \frac{t_1 \xrightarrow{\alpha} t_1'}{t_1 \parallel t_2 \xrightarrow{\alpha} t_1' \parallel t_2}$$

$$(2) \quad \frac{t_2 \xrightarrow{\alpha} t_2'}{t_1 \parallel t_2 \xrightarrow{\alpha} t_1 \parallel t_2'} \quad (\text{COMPOSICION PARALELA})$$

$$(3) \quad \frac{t_1 \xrightarrow{\alpha} t_1', t_2 \xrightarrow{\alpha} t_2'}{t_1 \parallel t_2 \xrightarrow{\alpha} t_1' \parallel t_2'}$$

(V)

$$\frac{t \xrightarrow{\alpha} t', \alpha \neq \beta}{t \setminus \beta \xrightarrow{\alpha} t'} \quad (\text{OCULTADOR})$$

(D)

$$\frac{t \xrightarrow{\alpha} t'}{\eta \eta \xrightarrow{f(\alpha)} \eta \eta'} \quad (\text{REDEFINICION})$$

Notación:

- Sea $s = \alpha_1 \dots \alpha_n$, $n > 0$ y $\alpha_i \in P$
entonces $p \xrightarrow{(\tau)^*} \xrightarrow{\alpha_1} \xrightarrow{(\tau)^*} \dots \xrightarrow{\alpha_n} \xrightarrow{(\tau)^*} p'$
se escribe como $p \xrightarrow{s} p'$.
- $t \xrightarrow{(\tau)^*} t'$ se escribe como $t \Rightarrow t'$.
- $t \xrightarrow{(\tau)^*} t'$ se escribe como $t \rightleftharpoons t'$.

1.3.- Relaciones de equivalencia y de congruencia.

R. Milner define relaciones de equivalencia y de congruencia sobre T_P . Estas relaciones definen criterios de comparación entre términos. A continuación se presentan tres relaciones definidas para T_P . Estas relaciones son:

- la *congruencia fuerte*
- la *equivalencia observacional*
- la *congruencia observacional*

1.3.1. Congruencia Fuerte.

Sean: $p, q \in T_r$,

R una relación binaria sobre T_r y $F(R)$ la función:

- $(p, q) \in F(R)$ ssi (i) $p \xrightarrow{\alpha} p'$ implica $\exists q' (q \xrightarrow{\alpha} q' \text{ y } (p', q') \in R)$ y
(ii) $q \xrightarrow{\alpha} q'$ implica $\exists p' (p \xrightarrow{\alpha} p' \text{ y } (p', q') \in R)$,

Definición 1.1.3.

Se dice que R es una *bisimulación* si y solamente si $R \subseteq F(R)$.

Definición 2.1.3.

La *equivalencia fuerte* (escrita $\sim$) se define como la más grande relación de bisimulación, es decir:

$$\sim = \cup \{R \subseteq T_r \times T_r \mid R \subseteq F(R)\}.$$

La relación $\sim$ existe y es única porque F es monótona en el reticulado de las relaciones binarias con la inclusión de conjuntos ($\subseteq$).

Para los términos pertenecientes a T_r , la axiomatización de la *equivalencia fuerte* esta dada por: (A_1) - (A_{16}) de la Tabla 1. Esta relación de equivalencia es una relación de congruencia para los operadores α , $+$ et $\parallel$. En lo sucesivo se denominará *congruencia fuerte*.

1.3.2.- Equivalencia Observacional.

La equivalencia observacional se define exactamente como la congruencia fuerte, excepto que se utilizan las relaciones $\alpha \Rightarrow \gamma$ en lugar de $\xrightarrow{\alpha}$ y $\xrightarrow{\gamma}$. La equivalencia observacional se escribe como $\approx$.

Proposición 1.1.3.

Para dos términos $t_1, t_2 \in T_r$, $t_1 \approx t_2$ si y solamente si se verifican las condiciones siguientes:

- (i) $t_1 = \alpha \Rightarrow t'_1$ implica $(\alpha = \tau \text{ y } t'_1 \approx t_2)$
o $\exists t'_2 (t_2 = \alpha \Rightarrow t'_2 \text{ y } t'_1 \approx t'_2)$
(ii) $t_2 = \alpha \Rightarrow t'_2$ implica $(\alpha = \tau \text{ y } t'_2 \approx t_1)$
o $\exists t'_1 (t_1 = \alpha \Rightarrow t'_1 \text{ y } t'_1 \approx t'_2)$.

Proposición 2.1.3.

$p \sim q$ implica $p \approx q$.

La relación de equivalencia observacional no es una relación de congruencia. Por ejemplo: $a.nil \approx t.a.nil$ pero, $b.nil + a.nil$ *no es observacionalmente equivalente* a $b.nil + t.a.nil$.

Para términos pertenecientes a T_f , la equivalencia observacional está axiomatizada por: $(A_1)-(A_{20})$ de la Tabla 1.

1.3.3.- Congruencia Observacional.

La *congruencia observacional* (escrita $\approx^C$) se define como la mayor relación de congruencia para los operadores α , $+$ et $\parallel$ incluida en la equivalencia observacional.

Proposition 3.1.3.

Para dos términos $t_1, t_2 \in T_f$, $t_1 \approx^C t_2$ si y solamente si $\forall \alpha \in P_f$ se verifican las condiciones siguientes:

- (1) $t_1 \xrightarrow{\alpha} t'_1$ implica $\exists t'_2 (t_2 \xrightarrow{\alpha} t'_2 \text{ y } t'_1 \approx t'_2)$
- (2) $t_2 \xrightarrow{\alpha} t'_2$ implica $\exists t'_1 (t_1 \xrightarrow{\alpha} t'_1 \text{ y } t'_1 \approx t'_2)$

En la proposición siguiente, damos una nueva caracterización para la congruencia observacional. Esta caracterización se utilizará en la definición del procedimiento de decisión de la congruencia observacional.

Proposition 4.1.3.

Para dos términos $t_1, t_2 \in T_f$, $t_1 \approx^C t_2$ si y solamente si se verifican las condiciones siguientes:

- $t_1 \approx t_2$,
- $t_1 \xrightarrow{I} t'_1$ implica $\exists t'_2$ tal que $t_2 \xrightarrow{I} t'_2$ y $t'_1 \approx t'_2$,
- $t_2 \xrightarrow{I} t'_2$ implica $\exists t'_1$ tal que $t_1 \xrightarrow{I} t'_1$ y $t'_1 \approx t'_2$.

Para los términos pertenecientes a T_f , la axiomatización de la equivalencia observacional está dada por: $(A_1)-(A_{19})$ de la Tabla 1.

Milner a demostrado la completitud de una axiomatización para la congruencia observacional para los términos recursivos. Esta axiomatización está definida para términos recursivos escritos de

manera diferente a la nuestra, sin embargo, los axiomas y reglas de inferencia de nuestro formalismo pueden obtenerse de los suyos por transformaciones sintácticas. De esta manera, una axiomatización completa para la congruencia observacional de términos de T_r está dada por los axiomas (A_1) - (A_{19}) de la Tabla 1 y los axiomas y reglas de inferencia dados en la Tabla 2.

Cabe destacar que los axiomas (A_6) , (A_7) , (A_{12}) , (A_{14}) , (A_{16}) indicando las propiedades de asociatividad, conmutatividad y distributividad de los operadores $\parallel$, $\setminus$ y $[]$ no pueden deducirse de los otros axiomas; pero todas sus instancias si pueden deducirse.

TABLA 1:

(I):	(A ₁):	$t_1 + (t_2 + t_3) = (t_1 + t_2) + t_3$
	(A ₂):	$t_1 + t_2 = t_2 + t_1$
	(A ₃):	$t + t = t$
	(A ₄):	$t + \text{nil} = t$
	(A ₅):	Si t_1 y t_2 son de la forma: $t_1 = \sum_{i=1, \dots, n} \alpha_i \cdot t_i \quad \text{y} \quad t_2 = \sum_{j=1, \dots, n} \beta_j \cdot t_j$ entonces: $t_1 \parallel t_2 = \sum \alpha_i \cdot (t_i \parallel t_2) +$ $\sum \beta_j \cdot (t_1 \parallel t_j) +$ $\sum \tau(t_i \parallel t_j)$ $\alpha_i = -\beta_i \wedge -\alpha_i = \beta_i \wedge i=1, \dots, n \wedge i=1, \dots, n$
(II):	(A ₆):	$t_1 \parallel t_2 = t_2 \parallel t_1$
	(A ₇):	$t_1 \parallel (t_2 \parallel t_3) = (t_1 \parallel t_2) \parallel t_3$
	(A ₈):	$t \parallel \text{nil} = t$
	(A ₉):	$(\alpha.t) \parallel \eta = \eta(\alpha).(\eta \parallel \eta)$
(III):	(A ₁₀):	$(t_1 + t_2) \parallel \eta = t_1 \parallel \eta + t_2 \parallel \eta$
	(A ₁₁):	$\text{nil} \parallel \eta = \text{nil}$
	(A ₁₂):	$(t_1 \parallel t_2) \parallel \eta = t_1 \parallel \eta \parallel t_2 \parallel \eta$
(IV):		$(\text{nil} \text{ si } \alpha = \beta \circ \alpha = \sim \beta$
	(A ₁₃):	$(\alpha.t) \setminus \beta = \begin{cases} \text{ } & \text{si } \alpha.(t \setminus \beta) \end{cases}$
	(A ₁₄):	$(t_1 + t_2) \setminus \alpha = t_1 \setminus \alpha + t_2 \setminus \alpha$
	(A ₁₅):	$\text{nil} \setminus \alpha = \text{nil}$
	(A ₁₆):	$t \setminus \alpha_1 \setminus \alpha_2 = t \setminus \alpha_2 \setminus \alpha_1$
(V):	(A ₁₇):	$t + \tau.t = \tau.t$
	(A ₁₈):	$\alpha.(t_1 + \tau.t_2) = \alpha.(t_1 + \tau.t_2) + \alpha.t_2$
	(A ₁₉):	$\alpha.\tau.t = \alpha.t$
	(A ₂₀):	$\tau.t = t$

TABLA 2:

	$t_1 = t_2$
(1):	$(t, E \cup \{x_1=t_1\}) = (t, E \cup \{x_2=t_2\})$
(2):	$(t, E \cup \{x_1=t_1, x_2=t_2\}) = (t, E \cup \{x_1=t_1, x_2=t_2[t_1/x_1]\})$
(3):	$\frac{(z_i, E) = (t_i^2/x_j)_{j=1, \dots, n}, E)}{\text{var}(t_i) \{x_1, \dots, x_n\}}$ $(z_i, E) = (x_i, \{x_j=t_j\}_{j=1, \dots, n})$
(4):	$\frac{x \notin \text{util}(t, E)}{(t, E) = (t, E \cup \{x=t_1\})}$
(5):	$(t, E \cup \{x=\tau x+t_1\}) = (t, E \cup \{x=\tau t_1\})$
(6):	$(t, E \cup \{x=\tau.(x+t_1)+t_2\}) = (t, E \cup \{x=\tau x+t_1+t_2\})$

II. Formas Normales de Términos Regulares.

En este trabajo, se presenta la construcción de formas normales de términos regulares para la congruencia fuerte, la equivalencia y la congruencia observacional. Estas formas normales corresponden a términos cuyos sistemas de transiciones asociados poseen un número mínimo de estados y de transiciones.

Para la construcción de las formas normales, se utilizan las transformaciones *r-cadenas* y *r-circuitas* definidas en [So 87a] y la noción de *bisimulación*.

Para la construcción de cada una de las formas normales, se procede en tres etapas:

- 1.- Construcción del sistema de transiciones asociado al término.
- 2.- Aplicación de transformaciones adecuadas a dicho sistema de transiciones (reducción).
- 3.- Construcción del término de T_r asociado al sistema de

transiciones transformado.

La etapa 1 fue descrita en [5o 87a], a continuación se describen las etapas 2 y 3 de los algoritmos de construcción de formas normales.

II.1. Transformaciones de los sistemas de transiciones.

En esta parte se definen las transformaciones *NCF*, *NED* y *NCD* y se demuestra que estas transformaciones permiten construir las formas normales de los sistemas de transiciones para cada una de las equivalencias.

II.1.1.- Transformación *NCF*.

Definición 1.II.1.

Para todo sistema de transiciones $S=(Q, \cup_{\alpha \in P} P_{\alpha}, q_0)$, $NCF(S)$ es el sistema de transiciones cociente con respecto a la congruencia fuerte (las clases de equivalencia se obtienen por cálculo de la más grande bisimulación contenida en $Q \times Q$).

Proposición 1.II.1.

Para todo sistema de transiciones S , $NCF(S)$ es una forma normal para la congruencia fuerte.

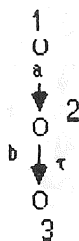
II.1.2.- Transformaciones *NED* y *NCD*

En el caso de la equivalencia y de la congruencia observacional, el sistema de transiciones cociente posee un número minimal de estados, pero no siempre un número minimal de transiciones. Por ejemplo, el sistema de transiciones reducido:



posee un número minimal de estados, pero no es minimal respecto al número de transiciones. En efecto, este sistema es

observacionalmente equivalente al sistema:



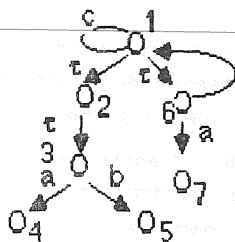
Este sistema es la forma normal (sistema bien guardado y compuesto por un mínimo número de estados y de transiciones).

Definición 1.11.2.

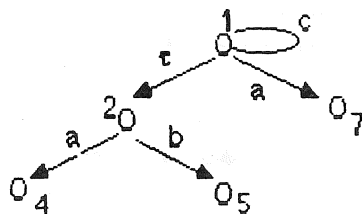
Para todo sistema de transiciones S , el sistema de transiciones $NE(S)$ se obtiene por aplicación de la siguiente secuencia de transformaciones:

- 1.- Transformaciones τ -cadenas y τ -circuitos.
- 2.- Reducción respecto a la equivalencia observacional.
- 3.- Eliminación de toda transición $p \xrightarrow{\alpha} q$ en el caso en que $\exists s \in \{\tau^* \alpha \tau^+, \tau^+ \alpha \tau^*\}$ tal que $p \xrightarrow{s} q$.

Por ejemplo, sea el siguiente sistema de transiciones:

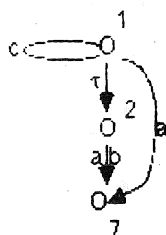


- Por aplicación de las transformaciones τ -cadenas y τ -circuitos, se obtiene el sistema de transiciones:

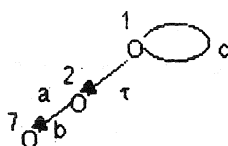


- El sistema de transiciones cociente respecto a la equivalencia

observacional es el siguiente:



- Por eliminación de transiciones superfluas (la transición $1 \xrightarrow{a} 3$ queda eliminada ya que $1 \xrightarrow{\tau a} 3$) se obtiene el sistema de transiciones $NED(S)$:



Proposición 1.11.2.

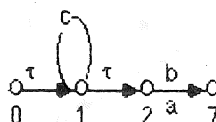
Para todo sistema de transiciones S , $NED(S)$ es una forma normal para la equivalencia observacional.

Definición 1.11.3.

Para todo sistema de transiciones S , $NED(S)$ es el sistema de transiciones obtenido de la manera siguiente:

- 1.- Construcción del sistema $S' = (Q, \cup_{\alpha \in P} \tau_{\alpha}, R_{\alpha}, q_0)$ por aplicación de la transformación NED sobre S .
- 2.- Si durante la construcción del sistema de transiciones $NED(S)$ eliminamos al menos una τ -transición partiendo del estado inicial, se agregará un nuevo estado inicial q'_0 , y una τ -transición $q \xrightarrow{\tau} q'_0$.

Por ejemplo, para el sistema de transiciones S del ejemplo anterior, el sistema de transiciones $NED(S)$ es el siguiente:



Proposición 1.11.3.

Para todo sistema de transiciones S , $NCO(S)$ es una forma normal para la congruencia observacional.

11.2.- Construcción del término asociado a un sistema de transiciones.

Definición 1.11.4.

El término regular asociado a un sistema de transiciones

$S=(Q, \cup_{\alpha \in P_{\tau}} R_{\alpha}, q_0)$ es $u_S=(C(q_0), E)$ donde:

$E=\{x_p=D(p) \mid p \in Q'\}$, con $Q' \subseteq Q$, $D: Q \rightarrow T(\Sigma_1, V)$, $C: Q \rightarrow T(\Sigma_1, V)$

definidos por,

$q_0 \in Q'$ ssi $\exists q \in Q, \exists \alpha \in P_{\tau}$ tales que: $q \xrightarrow{\alpha} q_0$,

$q \in Q'$, $q \neq q_0$ ssi $(\exists q_1, q_2, q_3 \in Q, \exists \alpha, \beta, \gamma \in P_{\tau})$ tales que: $q_1 \xrightarrow{\alpha} q$,

$q_2 \xrightarrow{\beta} q$ y $q \xrightarrow{\gamma} q_3$

$D(p)=\text{nil}$, si $\exists q \in Q, \exists \alpha \in P_{\tau}$ tales que $p \xrightarrow{\alpha} q$

$D(p)=\sum_{i \in I} \alpha_i.C(q_i)$,

ssi $p \xrightarrow{\alpha_i} q_i$ para $i \in I$

y para todo j tal que

$p \xrightarrow{\alpha_j} q_j$, $j \in I$,

$C(q)=x_q$,

si x_q es una variable de E .

$C(q)=D(q)$,

si no lo es.

Proposición 1.11.4.

Si S es un sistema de transiciones en forma normal para la congruencia fuerte, o para la equivalencia o la congruencia observacional, entonces el término $u=(t, E)$ asociado a S definido anteriormente esta también en forma normal para la equivalencia fuerte o la equivalencia o la congruencia observacional.

CONCLUSION.

El objetivo del trabajo fué el de definir procedimientos de construcción de formas normales.

Se realizó un estudio del cálculo CCS sin transmisión de valor. Para términos regulares, se definieron procedimientos de

construcción de formas normales minimales para la congruencia fuerte y la equivalencia y la congruencia observacional. Estos procedimientos se basan en transformaciones sobre los sistemas de transiciones que representan los comportamientos descritos por los términos.

REFERENCIAS.

- [Mi 80] Milner R.
A calculus of communicating systems.
Lecture Notes in Computer Science 92.
Springer-Verlag. 1980.
- [Mi 83] Milner R.
Calculi for synchrony and asynchrony.
T.C.S. 25; pp. 267-310. 1983.
- [Mi 85] Milner R.
A complete axiomatisation for observational congruence of finite state behaviors.
Manuscrito no publicado, diciembre 1985.
- [PI 81] Plotkin G.
A structural approach to operational semantics.
Aarhus University, Denmark.
DAIMI FN-19. Septembre 1981.
- [So 87] Soriano A.
VENUS: UN OUTIL D'AIDE A LA VERIFICATION DES SYSTEMES COMMUNICANTES.
Tesis de doctorado. INPG. Francia. 1987.
- [So 87a] Soriano, A.
Procedimientos de decisión para la equivalencia y la congruencia observacional de términos CCS
Anales de la XII Conferencia Latinoamericana de Informática.
1987. pp. 116-137.