

***Curso de Doctorado:
Programación Internet con Lenguajes
Declarativos Multiparadigma.***

PARTE I: Fundamentos

Pascual Julián Iranzo

Pascual.Julian@uclm.es

Universidad de Castilla – La Mancha. Departamento de Informática.

Lenguajes Integrados

Multiparadigma: Fundamentos

Indice

- 1.- Introducción.
- 2.- Sistemas ecuacionales.
 - ⇒ Sistemas de reescritura de términos.
- 4.- Narrowing, estrategias de narrowing y residuación.

Sistemas de reescritura de términos

Motivación y Objetivos

- Hemos estudiado la lógica ecuacional como fundamentación de la programación funcional (aproximación algebraica).
- Un conjunto de ecuaciones podía asimilarse a un **programa** y el problema de la validez a un proceso de **reducción** (sintáctico):
 - La ecuación $s \approx t$ es **valida** si el s puede transformarse en t (o ambos “**convergen**” a un mismo término).

Sistemas de reescritura de términos

Motivación y Objetivos

- **Ejemplo (Axiomas de Grupo):** Dado un conjunto de ecuaciones, \mathcal{E} ,

$$X + 0 = X \quad (e_1) \qquad -X + X = 0 \qquad (e_3)$$

$$0 + X = X \quad (e_2) \quad X + (Y + Z) = (X + Y) + Z \quad (e_4)$$

comprobamos que $\mathcal{E} \vdash -(-X) \approx X$, reduciendo $-(-X)$ a X

$$\begin{aligned} -(-X) &\approx -(-X) + 0 \approx -(-X) + (-X + X) \\ &\approx (-(-X) + -X) + X \approx 0 + X \approx X \end{aligned}$$

Sistemas de reescritura de términos

Motivación y Objetivos

- Para efectuar una reducción, en cada paso deben tomarse ciertas decisiones y precauciones:
 - Elegir el axioma ecuacional apropiado y usarlo en el sentido adecuado.
 - Encontrar la sustitución justa.
 - Elegir el camino adecuado para evitar reducciones infinitas.

Sistemas de reescritura de términos

Motivación y Objetivos

- Respecto a evitar reducciones infinitas, que las ecuaciones puedan emplearse en ambos sentidos es una desventaja:
 - En la anterior reducción se podría haber procedido así:

$$\begin{aligned} -(-X) &\approx -(-X) + 0 \approx -(-X) \\ &\approx -(-X) + 0 \approx -(-X) \dots \end{aligned}$$

Sistemas de reescritura de términos

Motivación y Objetivos

- Una forma de resolver el problema es utilizar las ecuaciones como **ecuaciones orientadas** (e.g., de izquierda a derecha).
- Esto nos lleva al concepto de sistema de reescritura de términos.

$$\begin{array}{ll} X + 0 \rightarrow X & (R_1) \qquad \qquad \qquad -X + X \rightarrow 0 \qquad \qquad \qquad (R_3) \\ 0 + X \rightarrow X & (R_2) \quad X + (Y + Z) \rightarrow (X + Y) + Z \quad (R_4) \end{array}$$

Sistemas de reescritura de términos

Motivación y Objetivos

- Sin embargo, para el problema que nos ocupa, esta manipulación no es suficiente:
 - El término λX no puede reducirse a X (o viceversa)
 - Tampoco pueden reducirse a un mismo término.
- Los términos λX y X son formas irreducibles (**formas normales**) distintas.

Sistemas de reescritura de términos

Motivación y Objetivos

- Habíamos probado que $--X \approx X$.
 - Pero hemos sido incapaces de probar la identidad de $--X$ y X usando reglas de reescritura.
- **Problema:** El sistema de reglas generado no es capaz de reproducir las propiedades del sistema de ecuaciones original.

Sistemas de reescritura de términos

Motivación y Objetivos

- Para que la reescritura sea una herramienta útil debemos construir sistemas reglas (a partir de un sistema de ecuaciones \mathcal{E}) con las siguientes propiedades:
 - Los términos tengan una única forma normal.

Sistemas de reescritura de términos

Motivación y Objetivos

- Para que la reescritura sea una herramienta útil debemos construir sistemas reglas (a partir de un sistema de ecuaciones \mathcal{E}) con las siguientes propiedades:
 - Los términos tengan una única forma normal.
 - Las cadenas de reducción no sean infinitas.

Sistemas de reescritura de términos

Motivación y Objetivos

- Para que la reescritura sea una herramienta útil debemos construir sistemas reglas (a partir de un sistema de ecuaciones \mathcal{E}) con las siguientes propiedades:
 - Los términos tengan una única forma normal.
 - Las cadenas de reducción no sean infinitas.
 - El sistema sea correcto y completo para \mathcal{E} .

Sistemas de reescritura de términos

Motivación y Objetivos

- **Observación:** Es posible **completar** el sistema de reglas de nuestro ejemplo para que cumpla estas propiedades.
- **Conclusión:** Los sistemas de reescritura de términos pueden ser útiles para la mecanización del razonamiento ecuacional.

Sistemas de reescritura de términos

Motivación y Objetivos

- **Objetivo:** Estudiar los sistemas de reescritura de términos y algunas de sus propiedades más importantes (que aseguren la corrección y completitud con respecto al cálculo original).
- Estudiaremos principalmente:
 - Confluencia.
 - Terminación.
 - Estrategias de reescritura.

Sistemas de reescritura de términos

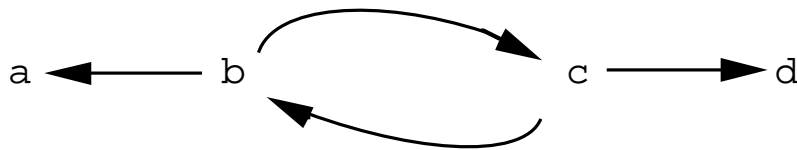
Sistemas de Reducción Abstractos

- Ciertas propiedades básicas de los sistemas de reescritura de términos pueden estudiarse de forma abstracta, independientemente de la estructura de las expresiones involucradas.
- Un **sistema de reducción abstracto** es un par $\langle A, \rightarrow \rangle$ donde:
 1. A es un conjunto y
 2. \rightarrow una relación binaria sobre A (**relación de reducción**).

Sistemas de reescritura de términos

Sistemas de Reducción Abstractos

- **Ejemplo:** Sea el conjunto $A = \{a, b, c, d\}$. Podemos definir la relación $\rightarrow = \{\langle b, a \rangle, \langle b, c \rangle, \langle c, b \rangle, \langle c, d \rangle\}$



- **Notación:** escribimos $a \rightarrow b$ en lugar de $\langle a, b \rangle \in \rightarrow$.

Sistemas de reescritura de términos

Sistemas de Reducción Abstractos

- Una **secuencia de reducción** finita (de longitud n) es una secuencia a_0, a_1, \dots, a_n de $n + 1$ elementos de A tal que, para todo $1 \leq i \leq n$, $a_{i-1} \rightarrow a_i$.
- **Notación:** $a_0 \rightarrow a_1 \rightarrow \dots \rightarrow a_n$.
- Se dice que a_n es un **reducto** de a_0 .
- $a_{i-1} \rightarrow a_i$ es un **paso de reducción**.
- **Ejemplo:** $b \rightarrow c \rightarrow b \rightarrow a$.

Sistemas de reescritura de términos

Sistemas de Reducción Abstractos

- **Definiciones y Notaciones:**

- (Identidad) $\rightarrow^0 = \{ \langle x, x \rangle \mid x \in A \} .$

- (potencia $i + 1$) $\rightarrow^{i+1} = \rightarrow \circ \rightarrow^i .$

- (cierre reflexivo) $\rightarrow^= = \rightarrow \cup \rightarrow^0 .$

- (cierre transitivo) $\rightarrow^+ = \bigcup_{i>0} \rightarrow^i .$

- (cierre reflexivo y transitivo) $\rightarrow^* = \rightarrow^+ \cup \rightarrow^0 .$

Sistemas de reescritura de términos

Sistemas de Reducción Abstractos

- **Definiciones y Notaciones:**

- **(Inversa)** $\leftarrow \equiv \rightarrow^{-1} = \{\langle y, x \rangle \mid x \rightarrow y\}.$

- **(cierre simétrico)** $\leftrightarrow = \rightarrow \cup \leftarrow.$

- **(cierre simétrico transitivo)** $\overset{+}{\leftrightarrow} = (\leftrightarrow)^+.$

- **(cierre simétrico reflexivo y transitivo)** $\overset{*}{\leftrightarrow} = (\leftrightarrow)^*.$

Sistemas de reescritura de términos

Sistemas de Reducción Abstractos

- **Nomenclatura:** \leftrightarrow^* se denomina **relación de convertibilidad**.
- **Ejercicio:** Hallar \rightarrow^0 , \rightarrow^3 , $\rightarrow^=$, \rightarrow^+ y \rightarrow^* para el SRA de la página 14.
- **Ejercicio:** Mostrar que \leftrightarrow^* es la menor relación de equivalencia que contiene a \rightarrow .

Sistemas de reescritura de términos

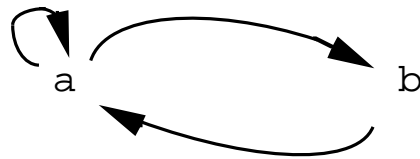
Sistemas de Reducción Abstractos

- **Definiciones y Notaciones:**
 - a es **reducible** sii existe un b tal que $a \rightarrow b$.
 - c está **forma normal** (es **irreducible**) sii no es reducible.
 - c es **una forma normal de** a sii $a \rightarrow^* c$ y c es irreducible.
 - a y b son **joinables** (denotado $a \downarrow b$) sii existe un c tal que $a \rightarrow^* c \leftarrow^* b$.

Sistemas de reescritura de términos

Sistemas de Reducción Abstractos

- **Ejemplo:** Dado el SRA de la página 14, a y d son formas normales. Ambos elementos son reductos de b o c .
- **Ejemplo:** Sea el conjunto $A = \{a, b, \}$. Podemos definir la relación $\rightarrow = \{\langle a, a \rangle, \langle a, b \rangle, \langle b, a \rangle\}$



Este SRA no contiene formas normales, pero a y b son *joinables*.

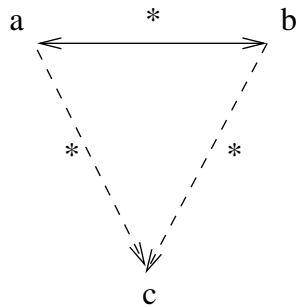
Sistemas de reescritura de términos

Sistemas de Reducción Abstractos

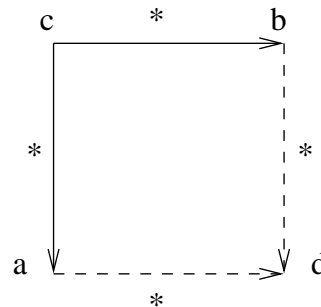
- **Definiciones y Notaciones:** Una relación de reducción, \rightarrow , se dice que
 - es **Church–Rosser** sii $(\forall a, b)(a \xleftrightarrow{*} b \Rightarrow a \downarrow b)$.
 - es **confluente** sii $(\forall a, b, c)(a \xleftarrow{*} c \rightarrow^{*} b \Rightarrow a \downarrow b)$.
 - es **semiconfluente** sii $(\forall a, b, c)(a \leftarrow c \rightarrow^{*} b \Rightarrow a \downarrow b)$.
- Un SRA $\langle A, \rightarrow \rangle$ es **confluente** si \rightarrow es confluente (Idem. resto de propiedades).

Sistemas de reescritura de términos

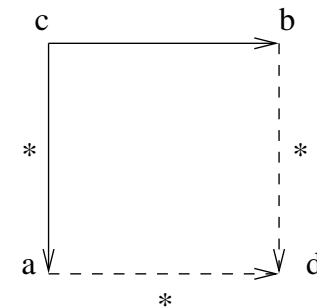
Sistemas de Reducción Abstractos



Church–Rosser



Confluente



Semiconfluente

- Las líneas continuas expresan cuantificación universal; las discontinuas existencial.
- **Ejercicio:** Comprobar que el SRA de la página 14 no es confluente.

Sistemas de reescritura de términos

Sistemas de Reducción Abstractos

- **Definiciones y Notaciones:** Una relación de reducción, \rightarrow , se dice que
 - es **terminante** (o **noetheriana**) sii no existen secuencias de reducción infinitas.
 - es **normalizante** sii todo elemento tiene una forma normal.
 - es **canónico** (o **convergente**) sii es confluente y terminante.

Sistemas de reescritura de términos

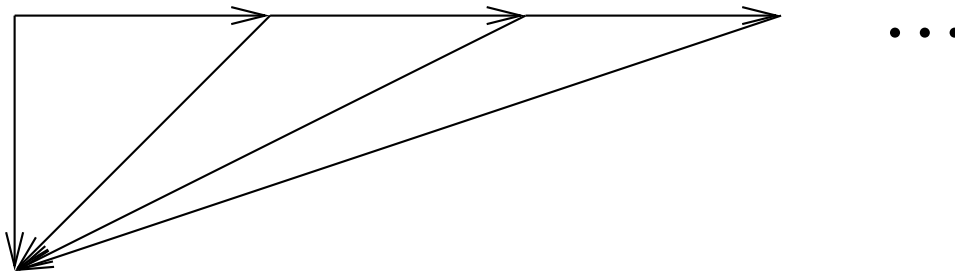
Sistemas de Reducción Abstractos

- **Ejemplo:** ninguno de los SRA analizados hasta ahora es terminante.
- **Ejemplo:** El conjunto (estríctamente) ordenado $\langle \mathbb{N}, < \rangle$, visto como SRA, es terminante.
- **Observación:** Relación terminante equivale a **relación bien fundada**. Por consiguiente, sobre un SRA terminante puede aplicarse el **principio de inducción completo** (o **principio de inducción noetheriano**).

Sistemas de reescritura de términos

Sistemas de Reducción Abstractos

- Toda relación terminante es normalizante (¿por qué?). Lo contrario no es cierto.
- **Ejemplo:** La siguiente relación es normalizante pero no es terminante.



Sistemas de reescritura de términos

Sistemas de Reducción Abstractos: Resultados sobre confluencia

- **Proposición:** Sea una relación $\rightarrow \subseteq A \times A$ confluente. Si $a \in A$ tiene una forma normal, ésta es **única**.
- **Ejercicio:** demostrar por reducción al absurdo.
- **Observación:** La confluencia permite la **implementación determinista de los cálculos**, ya que, cualquiera que sea la secuencia de reducción que sigamos, computaremos la misma (única) forma normal.

Sistemas de reescritura de términos

Sistemas de Reducción Abstractos: Resultados sobre confluencia

- **Teorema:** Si \rightarrow es una relación de reducción, las siguientes proposiciones son equivalentes:
 1. \rightarrow tiene la propiedad de Church–Rosser.
 2. \rightarrow es confluente.
 3. \rightarrow es semiconfluente.

Sistemas de reescritura de términos

Sistemas de Reducción Abstractos: Resultados sobre confluencia

- **Observación:** Para comprobar que un SRA es confluente hay que estudiar puntos “divergentes” de longitud arbitraria (i.e., $a \xrightarrow{*} c \rightarrow^* b$). Este es un problema difícil.
- **Solución:** localizar la confluencia estudiar puntos “divergentes” de un paso de longitud (i.e., $a \leftarrow c \rightarrow b$).

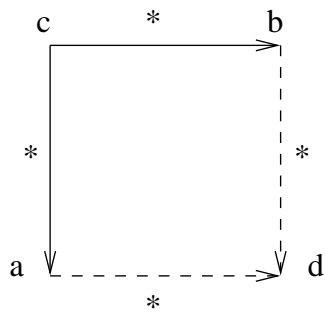
Sistemas de reescritura de términos

Sistemas de Reducción Abstractos: Resultados sobre confluencia

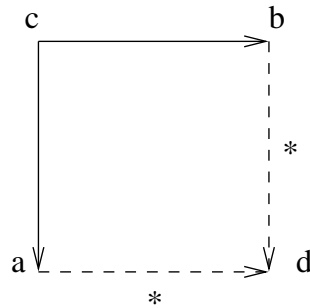
- **Definiciones y Notaciones:** Una relación de reducción, \rightarrow , se dice que
 - es **localmente confluyente** sii
 $(\forall a, b, c)(a \leftarrow c \rightarrow b \Rightarrow a \downarrow b)$.
 - es **fuertemente confluyente** sii
 $(\forall a, b, c)(a \leftarrow c \rightarrow b \Rightarrow (\exists d) a \rightarrow^* d \leftarrow b)$.

Sistemas de reescritura de términos

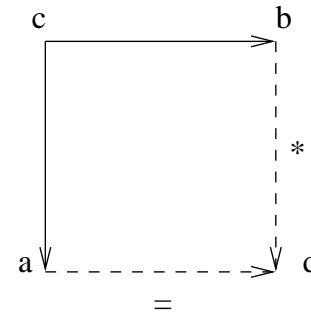
Sistemas de Reducción Abstractos: Resultados sobre confluencia



Confluente



Localmente confluente



Fuertemente confluente

- **Proposición:** Toda relación fuertemente confluente es confluente.
- **Proposición [Lema de Newman]:** Una relación terminante y localmente confluente es confluente.

Sistemas de reescritura de términos

Sistemas de Reducción Abstractos: Convergencia

- **Proposición:** Si \rightarrow es convergente, $a \xleftrightarrow{*} b$ sii $a \downarrow = b \downarrow$.
- **Observación:** Este resultado proporciona un **test para la equivalencia** de dos elementos a y b :
 1. Reducir a y b hasta alcanzar su forma normal $a \downarrow$ y $b \downarrow$.
 2. comprobar que $a \downarrow$ y $b \downarrow$ son iguales.
- **Observación:** Si $\xleftrightarrow{*}$ coincide con \approx_{ε} tendremos un mecanismo de decisión para el problema de la validez.

Sistemas de reescritura de términos

Sistemas de reescritura de términos y relación de reescritura

- Una **regla de reescritura** es un par ordenado $\langle l, r \rangle$, escrito $l \rightarrow r$, donde
 1. $l, r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$,
 2. $l \notin \mathcal{X}$ y
 3. $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$.
- l se denomina la **parte izquierda (lhs)** y r la **parte derecha (rhs)**.

Sistemas de reescritura de términos

Sistemas de reescritura de términos y relación de reescritura

- Ejemplo: reglas de reescritura válidas:

$$if(True, x, y) \rightarrow x$$

$$from(x) \rightarrow x : from(S(x))$$

$$first(S(x), y : z) \rightarrow y : first(x, z)$$

Sistemas de reescritura de términos

Sistemas de reescritura de términos y relación de reescritura

- **Ejemplo:** expresiones que no son reglas de reescritura:

$$g(x, y) \rightarrow f(g, b)$$

La *rhs* no es un término

$$x \rightarrow f(x)$$

La parte izquierda es una variable

$$f(x) \rightarrow y$$

No se cumple la condición 3

Sistemas de reescritura de términos

Sistemas de reescritura de términos y relación de reescritura

- Si no se cumple la condición 3, decimos que $l \rightarrow r$ es una regla de reescritura con **variables extra**.
- Un **Sistema de Reescritura de Términos** (TRS) es un par $\mathcal{R} = \langle \mathcal{F}, R \rangle$ donde R es un conjunto finito de reglas de reescritura.
- La noción estándar de TRS prohíbe la presencia de reglas de reescritura con **variables extra**.

Sistemas de reescritura de términos

Sistemas de reescritura de términos y relación de reescritura

- Ejemplo de TRS:

$$R_1 : if(True, x, y) \rightarrow x \qquad R_5 : and(True, x) \rightarrow x$$

$$R_2 : if(False, x, y) \rightarrow y \qquad R_6 : and(False, y) \rightarrow False$$

$$R_3 : Cero + x \rightarrow x$$

$$R_4 : S(x) + y \rightarrow S(x + y)$$

- Muchas veces identificaremos el TRS \mathcal{R} con el conjunto de reglas R .

Sistemas de reescritura de términos

Sistemas de reescritura de términos y relación de reescritura

- **Convenciones y notaciones:**
 - $L(\mathcal{R})$: El conjunto de las partes izquierdas de un TRS.
 - Toda instancia $\sigma(l)$ de una lhs $l \in L(\mathcal{R})$ de una regla es un **redex**.
 - $\mathcal{P}os_{\mathcal{R}}(t)$: El conjunto de posiciones de redexes en un término t .

Sistemas de reescritura de términos

Sistemas de reescritura de términos y relación de reescritura

- Las reglas de reescritura de un TRS inducen una relación de reducción \rightarrow entre los términos.
- [Relación de Reescritura]
Un término t se **reescrive** al término s (en la posición p), escrito $t \rightarrow_{\mathcal{R}, \alpha}^p s$, si
 1. existe una posición $p \in \mathcal{Pos}(t)$,
 2. una regla de reescritura $\alpha : l \rightarrow r$ y
 3. una sustitución σ con $t|_p = \sigma(l)$ y $s = t[\sigma(r)]_p$.

Sistemas de reescritura de términos

Sistemas de reescritura de términos y relación de reescritura

- Decimos que $t|_p$ es el **redex** de t que se ha **contraído** (o **reducido**) y que $\sigma(r)$ es su **contracto**.
- **Observación:** Escribiremos $t \rightarrow_{p,\alpha} s$, $t \rightarrow_p s$ o $t \rightarrow s$ cuando no sea necesario especificar el TRS \mathcal{R} , la regla y/o la posición.
- El par $\langle \mathcal{T}(\mathcal{F}, \mathcal{X}), \rightarrow_{\mathcal{R}} \rangle$ es el sistema de reducción abstracto asociado al TRS \mathcal{R} .

Sistemas de reescritura de términos

Sistemas de reescritura de términos y relación de reescritura

- **Ejercicio:** Para el TRS de la página 36, calcular los distintos pasos de reescritura que pueden darse con el término

$$t = if(and(True, False), Cero + S(Cero), Cero).$$

- **Ejercicio:** Dados los términos $t \equiv f(h(h(a)), h(x))$ y $s \equiv f(h(h(a)), h(a))$ y el TRS

$$R_1 : h(h(x)) \rightarrow h(x) \quad R_2 : h(a) \rightarrow a \quad R_3 : f(x, x) \rightarrow x$$

Obtener todas las secuencias de reescritura a partir de t y s .

Sistemas de reescritura de términos

Relación de reescritura y problema de la validez

- **Proposición:** Sea el TRS $\mathcal{R} = \langle \mathcal{F}, R \rangle$, la relación $\rightarrow_{\mathcal{R}}$ es la mínima relación en $\mathcal{T}(\mathcal{F}, \mathcal{X})$ que:
 1. $R \subset \rightarrow_{\mathcal{R}}$
[$\rightarrow_{\mathcal{R}}$ contiene a R , i.e., $(l \rightarrow r) \in R \Rightarrow l \rightarrow_{\mathcal{R}} r$];
 2. $(\forall s, t \in \mathcal{T})(\forall f \in \mathcal{F})(s \rightarrow_{\mathcal{R}} t \Rightarrow f(., s, .) \rightarrow_{\mathcal{R}} f(., t, .))$
[Compatible bajo las operaciones de \mathcal{F}];
 3. $(\forall s, t \in \mathcal{T})(\forall \sigma)(s \rightarrow_{\mathcal{R}} t \Rightarrow \sigma(s) \rightarrow_{\mathcal{R}} \sigma(t))$
[Estable bajo sustituciones];

Sistemas de reescritura de términos

Relación de reescritura y problema de la validez

- Una relación binaria \rightarrow sobre $\mathcal{T}(\mathcal{F}, \mathcal{X})$ es **estable bajo las operaciones de \mathcal{F}** sii

$$(\forall f \in \mathcal{F})(\forall s_1, \dots, s_n, t_1, \dots, t_n \in \mathcal{T}) \\ [s_1 \rightarrow t_1 \wedge \dots \wedge s_n \rightarrow t_n \Rightarrow f(\overline{s_n}) \rightarrow f(\overline{t_n})].$$

- **Lema:** Sea una relación binaria \rightarrow sobre $\mathcal{T}(\mathcal{F}, \mathcal{X})$, reflexiva y transitiva. Entonces, \rightarrow es estable bajo las operaciones de \mathcal{F} sii es compatible bajo las operaciones de \mathcal{F} .

Sistemas de reescritura de términos

Relación de reescritura y problema de la validez

- **Teorema:** Sea \mathcal{R} un TRS. $\overset{*}{\leftrightarrow}_{\mathcal{R}}$ es la mínima congruencia sobre $\mathcal{T}(\mathcal{F}, \mathcal{X})$ que contiene a \mathcal{R} y es estable bajo sustituciones.
- **Corolario:** Sea \mathcal{E} un conjunto de ecuaciones y \mathcal{R} el TRS que resulta de orientar (convenientemente) las ecuaciones de \mathcal{E} .

$$(\forall s, t \in \mathcal{T})(s \overset{*}{\leftrightarrow}_{\mathcal{R}} t \Leftrightarrow \mathcal{E} \vdash s \approx t)$$

- Por tanto, se cumple que $\overset{*}{\leftrightarrow}_{\mathcal{R}}$ es equivalente a $\approx_{\mathcal{E}}$.

Sistemas de reescritura de términos

Relación de reescritura y problema de la validez

- El teorema anterior junto a las observaciones de la página 31 proporcionan un **procedimiento de decisión para el problema de la validez** cuando trabajamos con TRS convergentes.
- Este resultado justifica la importancia de las propiedades de terminación y confluencia y la necesidad de una mejor caracterización en el ámbito de los TRSs.

Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- Dadas dos reglas $\alpha_1 : l_1 \rightarrow r_1$ y $\alpha_2 : l_2 \rightarrow r_2$ para las que existe una posición $p \in \mathcal{FP}_{os}(l_1)$ tal que $l_1|_p$ y l_2 unifican con *mgu* σ , entonces el par de reductos $\langle \sigma(r_1), \sigma(l_1)[\sigma(r_2)]_p \rangle$ es un **par crítico**.
- α_1 y α_2 no comparten variables, para lo que pueden ser renombradas si es preciso.
- Excluimos el caso trivial $\alpha_1 = \alpha_2$ y cuando $p = \Lambda$.

Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- Un par crítico $\langle t, s \rangle$, con $t = \sigma(r_1)$ y $s = \sigma(l_1)[\sigma(r_2)]_p$
 - es un **cubrimiento** (*overlay*) si $p = \Lambda$;
 - es **trivial** si $t \equiv s$;
 - es **convergente** si $t \downarrow s$.
- **Ejemplo:** El TRS $\{g(B, a) \rightarrow B, a \rightarrow B\}$ tiene un par crítico $\langle g(B, a)[B]_2, B \rangle = \langle g(B, B), B \rangle$ que no es convergente.

Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- Ejemplo: El TRS

$$\{g(B, a) \rightarrow B, a \rightarrow B, g(B, B) \rightarrow B\}$$

tiene un único par crítico $\langle g(B, a)[B]_2, B \rangle$
 $= \langle g(B, B), B \rangle$ que es convergente.

- Un TRS es **lineal por la izquierda** si para toda $l \in L(\mathcal{R})$, l es un término lineal (i.e., sin variables repetidas).

Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- Un TRS \mathcal{R} se dice **ortogonal** si es
 1. Lineal por la izquierda.
 2. \mathcal{R} no contiene pares críticos [(**fuertemente**) **no ambiguo**]
- También se dice que las lhs de las reglas de \mathcal{R} no **solapan**.

Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- Un TRS \mathcal{R} se dice **ortogonal** si es
 1. Lineal por la izquierda.
 2. \mathcal{R} no contiene pares críticos [(**fuertemente**) **no ambiguo**]
- La ausencia de pares críticos en un sistema ortogonal tiene un significado muy intuitivo en términos del concepto de **patrón de una regla**.

Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- Un TRS \mathcal{R} se dice **ortogonal** si es
 1. Lineal por la izquierda.
 2. \mathcal{R} no contiene pares críticos [(**fuertemente**) **no ambiguo**]
- Un **patrón de una regla** es una lhs en la que las variables se han sustituido por “huecos” (\square).
- **Ejemplo:** El patrón de la regla $and(True, x) \rightarrow x$ es:
 $and(True, \square)$

Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- **Ejemplo:** El siguiente TRS es ortogonal:

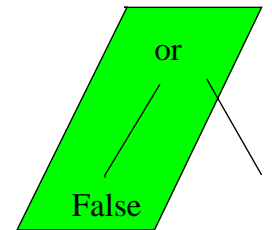
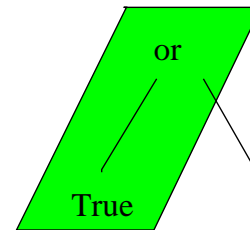
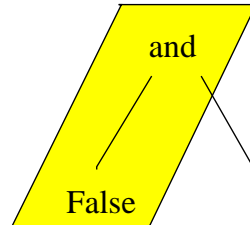
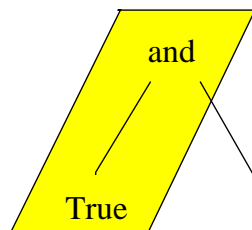
$$R_1 : \text{and}(\text{True}, x) \rightarrow x;$$

$$R_3 : \text{or}(\text{True}, x) \rightarrow \text{true};$$

$$R_2 : \text{and}(\text{False}, x) \rightarrow \text{False};$$

$$R_4 : \text{or}(\text{False}, x) \rightarrow x$$

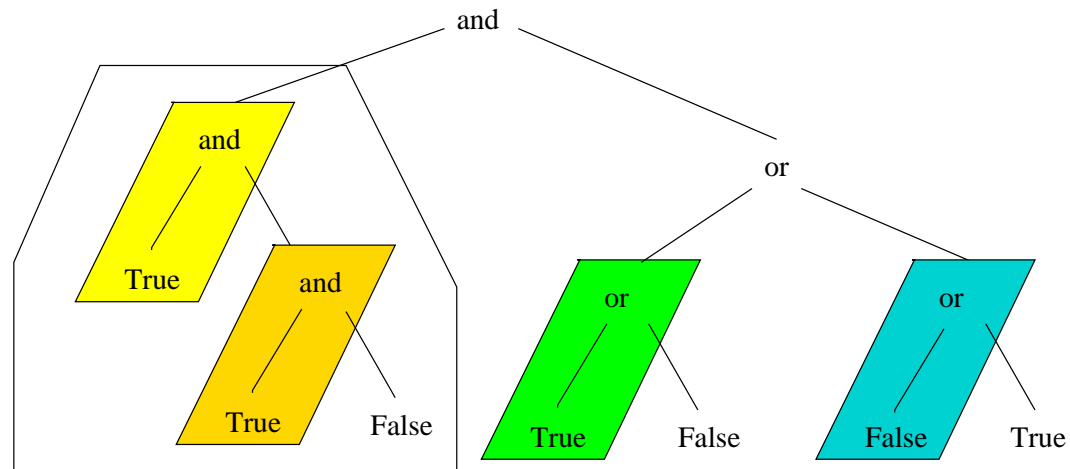
- Es lineal por la izquierda y los patrones de las reglas no solapan:



Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- En un TRS ortogonal, los patrones de las reglas que aparecen en un término no solapan (a pesar de que algunos redexes puedan solaparse).
- **Ejemplo:**



Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- Ejemplo: El siguiente TRS no es ortogonal.

$$R_1 : or(True, x) \rightarrow True; \quad R_2 : or(x, True) \rightarrow True;$$
$$R_3 : or(False, False) \rightarrow False$$

- Los patrones de las reglas R_1 y R_2 solapan:



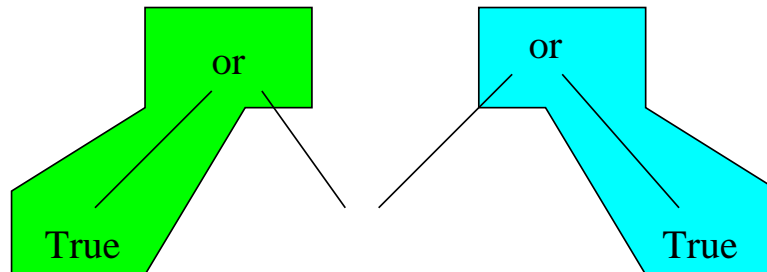
Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- **Ejemplo:** El siguiente TRS no es ortogonal.

$$R_1 : or(True, x) \rightarrow True; \quad R_2 : or(x, True) \rightarrow True;$$
$$R_3 : or(False, False) \rightarrow False$$

- Los patrones de las reglas R_1 y R_2 solapan:



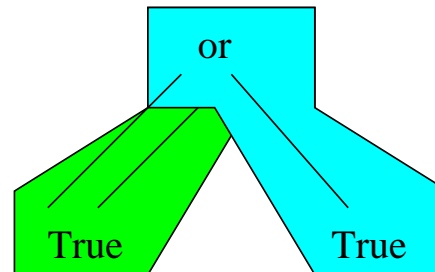
Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- **Ejemplo:** El siguiente TRS no es ortogonal.

$$R_1 : or(True, x) \rightarrow True; \quad R_2 : or(x, True) \rightarrow True;$$
$$R_3 : or(False, False) \rightarrow False$$

- Los patrones de las reglas R_1 y R_2 solapan:



Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- En un TRS ortogonal tiene sentido en concepto de **descendiente** (o **residuo**) de un redex.
- El conjunto $v \setminus A$ de **descendientes** del redex $t|_v$ a través de $A : t \rightarrow_{[u, \alpha]} s$ es un subconjunto de $\mathcal{P}os(s)$ definido como sigue:

$$v \setminus A = \begin{cases} \emptyset & \text{si } v = u \\ \{v\} & \text{si } (v \parallel u \vee v < u) \\ \{u.w_1.v_1 \mid r|_{w_1} = x\} & \text{si } (v = u.w.v_1 \wedge l|_w = x \in \mathcal{X}) \end{cases}$$

Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- Intuitivamente, el concepto de **descendiente** puede entenderse como sigue:
 1. marcamos la raíz del redex $t|_v$;
 2. damos el paso $t \rightarrow_{[u,\alpha]} s$ para obtener s (manteniendo las marcas introducidas);
 3. Las posiciones p (respec., los términos $t|_p$) que aparecen marcadas (marcados en su raíz) en el término s son las (los) descendientes de v ($t|_v$).

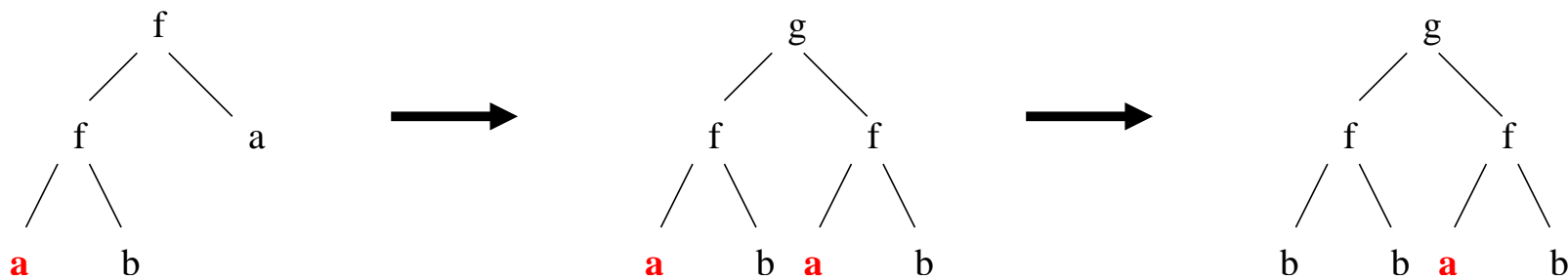
Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- Ejemplo: dado el TRS

$$\{R_1 : f(x, y) \rightarrow g(x, x); \quad R_2 : a \rightarrow b\},$$

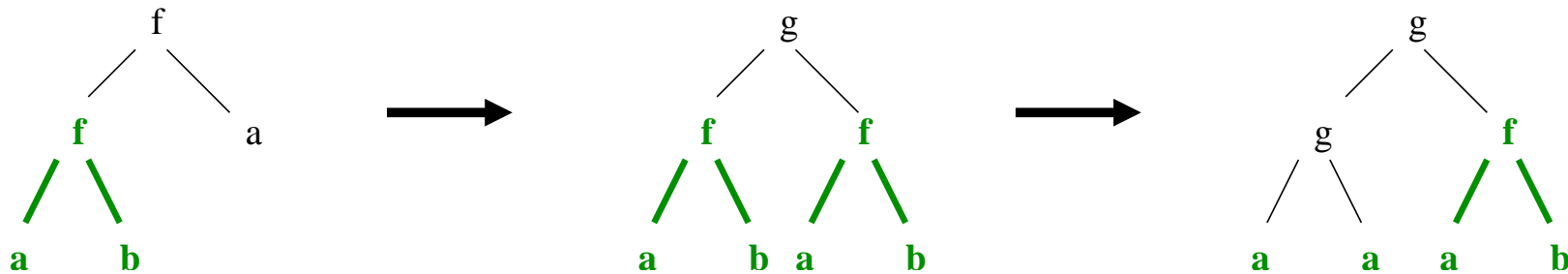
la figura muestra los descendientes del redex a en la posición 1.1 para la derivación.



Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

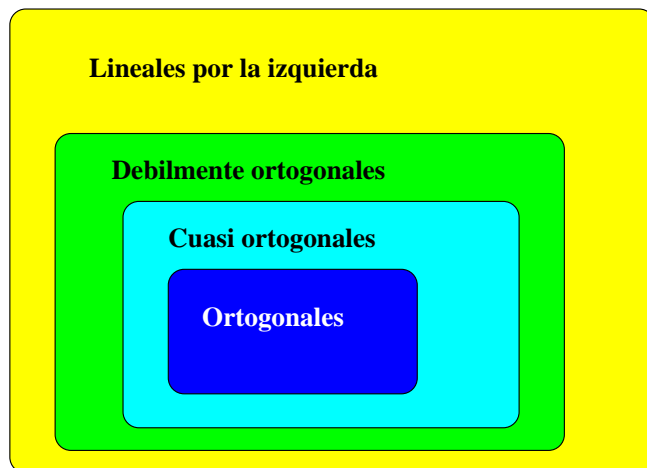
- Los TRSs ortogonales tienen la propiedad de que los descendientes de un redex continúan siendo redexes.



Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- La condición de no ambigüedad (fuerte) puede relajarse para dar lugar a clases, sucesivamente más amplias, de TRSs.

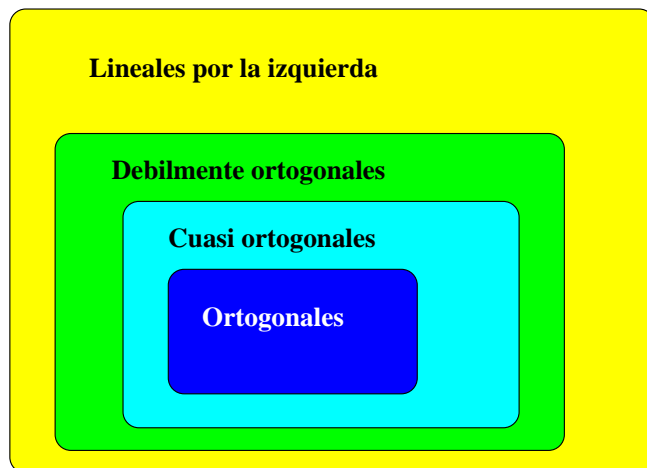


- Un SRT lineal por la izquierda cuyos pares críticos son cubrimientos triviales, se denomina **cuasi ortogonal** (*almost orthogonal*).

Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- La condición de no ambigüedad (fuerte) puede relajarse para dar lugar a clases, sucesivamente más amplias, de TRSs.



- Si un TRS sólo tiene pares críticos triviales se denomina **débilmente ortogonal** (*weakly orthogonal*).

Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- **Ejercicio:** Comprobar que el siguiente TRS es cuasiortogonal.

$$R_1 : or(True, x) \rightarrow True; \quad R_2 : or(x, True) \rightarrow True;$$
$$R_3 : or(False, False) \rightarrow False$$

- **Ejercicio:** Comprobar que el siguiente TRS es débilmente ortogonal.

$$R_1 : pred(suc(x)) \rightarrow x; \quad R_2 : suc(pred(x)) \rightarrow x;$$

Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- Dado un TRS $\mathcal{R} = \langle \mathcal{F}, R \rangle$, siempre podemos establecer una partición $\mathcal{F} = \mathcal{C} \uplus \mathcal{D}$ de la signatura \mathcal{F} donde:
 - $\mathcal{D} = \{f \in \mathcal{F} \mid (\exists l \in L(\mathcal{R})). \text{Root}(l) = f\}$
[símbolos de función **definidos** (u **operaciones**)]
 - $\mathcal{C} = \mathcal{F} \setminus \mathcal{D}$
[símbolos **constructores**]
- $\mathcal{T}(\mathcal{C}, \mathcal{X})$: conjunto de los **términos constructores**.

Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- Un **patrón** es un término de la forma $f(\overline{d_n})$ donde $f \in \mathcal{D}$ y los argumentos d_1, \dots, d_n son términos constructores o variables.
- Una regla de reescritura $l \rightarrow r$ cumple la **disciplina de constructores** si l es un patrón.
- Si todas las reglas de un TRS cumplen la disciplina de constructores, se dice que es un TRS **Basado en Constructores** (CB).

Sistemas de reescritura de términos

Clases de Sistemas de Reescritura.

- **Ejemplo:** El siguiente TRS es basado en constructores.

$$R_1 : f(a, x) \rightarrow c(x); \quad R_3 : g(c(x), y) \rightarrow g(y, a).$$

$$R_2 : f(x, b) \rightarrow b;$$

- En un TRS basado en constructores los pares críticos son siempre *overlays*.
- **Ejercicio:** Estudiar los TRSs de los ejemplos anteriores y determinar a qué clase pertenecen.

Sistemas de reescritura de términos

Confluencia

- La confluencia de un TRS es indecidible en el caso general.
- Es decidable para SRTs:
 - terminantes;
 - también para, SRTs (finitos) lineales por la izquierda y básicos por la derecha.

Sistemas de reescritura de términos

Confluencia: caracterizaciones sintácticas

- A pesar de las dificultades para establecer esta propiedad, existen caracterizaciones sintácticas de la confluencia.
- Una **caracterización sintáctica** es la que puede establecerse mediante inspección del TRS, sin necesidad de acudir a propiedades auxiliares

Sistemas de reescritura de términos

Confluencia: caracterizaciones sintácticas

- Dos métodos para demostrar la confluencia son:
 1. Análizar la **convergencia de los pares críticos** del TRS.
 2. Imponer restricciones sintácticas al TRS.

Sistemas de reescritura de términos

Confluencia: caracterizaciones sintácticas

- **Teorema [Huet 1980]:** Un TRS es localmente confluente si y sólo si todo par crítico es convergente.
- **Ejercicio:** Comprobar que el TRS

$$\begin{aligned} f(x) &\rightarrow A; & g(f(x)) &\rightarrow f(H(x)); & g(A) &\rightarrow B \\ f(x) &\rightarrow g(f(x)); & g(f(x)) &\rightarrow B; \end{aligned}$$

es localmente confluente (estudiando la convergencia de sus pares críticos).

Sistemas de reescritura de términos

Confluencia: caracterizaciones sintácticas

- **Teorema** [Confluencia de los TRSs terminantes – Huet 1980]: Un SRT terminante \mathcal{R} es confluente si y sólo si todos sus pares críticos son convergentes.
- La confluencia local, por sí misma, no garantiza la confluencia.
- **Ejercicio:** Comprobar que el TRS de la página anterior (que no es terminante) no es confluente, aunque vimos que era localmente confluente.

Sistemas de reescritura de términos

Confluencia: caracterizaciones sintácticas

- Los TRS ortogonales cumplen el llamado *parallel moves lemma*, un corolario de la propiedad de los TRSs ortogonales de que los descendientes de un redex continúan siendo redexes.
- La aplicación reiterada del *parallel moves lemma* está en la base de las pruebas de muchas propiedades de los sistemas ortogonales.

Sistemas de reescritura de términos

Confluencia: caracterizaciones sintácticas

- Lema [*parallel moves lemma*]: Dado un TRS ortogonal, sean $t \rightarrow^* t_2$ y sea $t \rightarrow t_1$ un paso de reducción dado al contraer un redex s de t . Entonces, existe un reducto t_3 de t_1 y t_2 , que puede encontrarse por contracción en t_2 de todos los descendientes del redex s que son mutuamente disjuntos.

Sistemas de reescritura de términos

Confluencia: caracterizaciones sintácticas

- Todo TRS (**débilmente**) ortogonal es confluente.
- La sólo condición de no ambigüedad no garantiza la confluencia.
- **Ejemplo:** El siguiente TRS es no ambiguo (**¿por qué?**), pero no es confluente.

$$\begin{array}{l} f(x, x) \rightarrow a \qquad g(x) \rightarrow c(g(x)) \\ f(y, c(y)) \rightarrow b \end{array}$$

(**Ayuda:** el término $f(g(a), g(a))$ tiene dos formas normales.)

Sistemas de reescritura de términos

Terminación

- Aunque la terminación es una de las propiedades más importantes de los TRSs, simplemente realizaremos unos comentarios ésta (ver Baader & Nipkow, 1998, para un tratamiento más completo).
- La propiedad de saber si un SRT es terminante es **indecidable** en el caso general
- Es habitual utilizar **ordenes bien fundados** (i.e., antirreflexivo, antisimétrico, transitivo, noetheriano) en las pruebas de terminación.

Sistemas de reescritura de términos

Terminación

- Aunque la terminación es una de las propiedades más importantes de los TRSs, simplemente realizaremos unos comentarios ésta (ver Baader & Nipkow, 1998, para un tratamiento más completo).
- La propiedad de saber si un SRT es terminante es **indecidible** en el caso general.
- Es habitual utilizar **ordenes bien fundados** (i.e., antirreflexivo, antisimétrico, transitivo, noetheriano) en las pruebas de terminación.

Sistemas de reescritura de términos

Terminación

- Un orden estricto $>$ sobre $\mathcal{T}(\mathcal{F}, \mathcal{X})$ es un **orden de reescritura** si:
 1. $\forall t, s, s' \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \forall p \in \mathcal{Pos}(t),$
 $s > s' \Rightarrow t[s]_p > t[s']_p, \text{ y}$
 2. (cerrado bajo sustituciones) $\forall \sigma \in \text{Subst}(\mathcal{F}, \mathcal{X})$
 $\forall t, s \in \mathcal{T}(\mathcal{F}, \mathcal{X}), t > s \Rightarrow \sigma(t) > \sigma(s).$
- Un **orden de reducción** es un orden de reescritura bien fundado.

Sistemas de reescritura de términos

Terminación

- **Ejemplo:** El orden $>$ sobre $\mathcal{T}(\mathcal{F}, \mathcal{X})$ definido como sigue (donde $|t|$ es el número de símbolos del término t y $|t|_x$ el número de ocurrencias de x en t):
 $t > s$ si y sólo si $|t| > |s|$ y, para todo $x \in \mathcal{X}$, $|t|_x \geq |s|_x$
es un orden de reducción.
- **Teorema:** Un TRS es terminante si y sólo si existe un orden de reducción que satisface $l > r$ para toda regla de reescritura $l \rightarrow r \in R$.

Sistemas de reescritura de términos

Terminación

- **Ejemplo:** Empleando el orden de reducción del Ejemplo anterior y el último Teorema, puede demostrarse la terminación del TRS:

$$\text{xor}(\text{True}, \text{False}) \rightarrow \text{True}$$

$$\text{xor}(\text{False}, \text{True}) \rightarrow \text{True}$$

$$\text{xor}(x, x) \rightarrow \text{False}$$

- ***Sistemas de reescritura de términos***

Estrategias de Reescritura

- Aunque hemos introducido la reescritura como una mecanización del razonamiento ecuacional, la reescritura es poderoso mecanismo de cómputo de expresiones.
- Computar consiste en una secuencia de reducciones hasta alcanzar una forma normal.
- Si la forma normal es única se considera el **valor** de la expresión.

Sistemas de reescritura de términos

Estrategias de Reescritura

- Dado un término, en general, existen diferentes secuencias de reducción que computan su forma normal.
- Para los TRSs canónicos cada término tiene una forma normal única que puede computarse mediante cualquier secuencia de reducción.

Sistemas de reescritura de términos

Estrategias de Reescritura

- Cuando el TRS no es terminante y queremos asegurarnos de que se computará la forma normal de una expresión (si existe), estamos obligados a:
 - construir todas las posibles secuencias de reducción (mediante un mecanismo de búsqueda en amplitud).
 - o bien, escoger cuidadosamente los pasos de reducción realizados (evitando las reducciones infinitas).

Sistemas de reescritura de términos

Estrategias de Reescritura

- Una **estrategia de reducción** es una función con la que encontrar los redexes que deben ser contraídos en cada paso.
- Dos tipos de estrategias (deterministas):
 1. las **secuenciales** (o **monopaso**) (seleccionan un único redex para ser reducido en un paso);
 2. las **paralelas** (o **multipaso**) (seleccionan un conjunto de redexes disjuntos para ser contraídos simultáneamente).

Sistemas de reescritura de términos

Estrategias de Reescritura: Ejemplos

- i.- La estrategia (monopaso) *leftmost-innermost* reduce, de entre los redexes más internos, el del extremo izquierdo.
- ii.- La estrategia (multipaso) *parallel-innermost* reduce todos los redexes más internos de forma simultánea.
- iii.- La estrategia (monopaso) *leftmost-outermost* reduce, de entre los redexes externos, el del extremo izquierdo.

Sistemas de reescritura de términos

Estrategias de Reescritura: Ejemplos

- iv.- La estrategia (multipaso) *parallel-outermost* reduce todos los redexes externos de forma simultánea.
- v.- La regla de *sustitución completa* (*full substitution rule*, una estrategia multipaso) reduce simultáneamente todos los redexes.
- **Observación:** Las estrategias (i) a (iv) están bien definidas para cualquier TRS (en (ii) y (iv), **los redexes seleccionados son disjuntos**); la (v) sólo para TRSs ortogonales.

Sistemas de reescritura de términos

Estrategias de Reescritura

- **Nomenclatura:** un redex $s = t|_p$, en un término t ,
 - es **interno** (*innermost*) sii $p \in \text{maximal}_{\leq}(\text{Pos}_{\mathcal{R}}(t))$ (i.e., si no contiene ningún redex);
 - es **externo** (*outermost*) sii $s = t|_p$ y $p \in \text{minimal}_{\leq}(\text{Pos}_{\mathcal{R}}(t))$ (i.e., si no está contenido en ningún redex).
- **Observación:** \leq es el orden de prefijos entre posiciones (ver Tema 2, página 10).

Sistemas de reescritura de términos

Estrategias de Reescritura

- **Nomenclatura:** un redex $s = t|_p$, en un término t , es un **extremo izquierdo** (*leftmost*) sii $p = \min_{\leq_L} (\mathcal{Pos}_{\mathcal{R}}(t))$.
- **Observación:** \leq_L es el orden **lexicográfico** entre posiciones:
 - $\Lambda \leq_L p$ para toda posición p ;
 - si $p = i.p'$ y $q = j.q'$ entonces $p \leq_L q$ si y sólo si $i < j \vee (i = j \wedge p' \leq_L q')$.

Sistemas de reescritura de términos

Estrategias de Reescritura

- **Ejercicio:** Dado el TRSs de la página 49 y el término

$$\text{and}(\text{and}(\text{True}, \text{and}(\text{True}, \text{False})), \\ \text{or}(\text{and}(\text{True}, \text{False}), \text{or}(\text{False}, \text{True})))$$

1. Señalar los redexes más internos (externos); el redex en el extremo izquierdo y el redex en el extremo izquierdo más interno (externo);
2. Hallar su forma normal empleando las diversas estrategias de reescritura.

Sistemas de reescritura de términos

Estrategias de Reescritura

- Una **estrategia de reescritura** es una función

$$\mathcal{S}_{\mathcal{R}} : \mathcal{T}(\mathcal{F}, \mathcal{X}) \rightarrow \wp(\text{Occ})$$

tal que si \mathcal{R} es un TRS y t un término, $\mathcal{S}_{\mathcal{R}}(t)$ es el conjunto de redexes a contraer.

- Si el TRS \mathcal{R} es (débilmente) ortogonal a cada posición $p \in \mathcal{S}_{\mathcal{R}}(t)$ le corresponde una única regla a emplear en la contracción.

Sistemas de reescritura de términos

Estrategias de Reescritura

- **Ejercicio:** Definir formalmente la función $\mathcal{S}_{\mathcal{R}}$ que corresponde a las estrategias de reescritura (i) a (v). [**Ayuda:** utilizar las funciones $maximal_{\leq}$, $minimal_{\leq}$ y $min_{\leq L}$.]
- **Notación:** $t \rightarrow_{\mathcal{S}_{\mathcal{R}}} s$ denota un monopaso (multipaso) dado al reducir las posiciones de redexes seleccionadas por \mathcal{S} . Cuando el TRS \mathcal{R} esté claro por el contexto no se hará referencia a él.

Sistemas de reescritura de términos

Estrategias de Reescritura

- Una secuencia de reducción $t \rightarrow_{\mathcal{S}}^* s$ decimos que es una secuencia que **respet**a la estrategia \mathcal{S} .
- Una estrategia \mathcal{S} es **normalizante** si para todo término t (que posee forma normal $t\downarrow$) existe una secuencia de reducción (que **respet**a la estrategia \mathcal{S}) tal que $t \rightarrow_{\mathcal{S}}^* t\downarrow$.
- **Observación:** Para TRSs ortogonales, las estrategias (iv) y (v) son estrategias normalizantes.

Sistemas de reescritura de términos

Estrategias de Reescritura: reducciones necesarias

- Si bien la estrategia *parallel-outermost* es normalizante, no siempre conduce a secuencias de reducción óptimas.
- Ejemplo: Sea el TRS

$$\begin{aligned}if(true, x, y) &\rightarrow x; & a &\rightarrow b. \\if(false, x, y) &\rightarrow y;\end{aligned}$$

y el término $if(if(true, false, true), a, a)$.

Sistemas de reescritura de términos

Estrategias de Reescritura: reducciones necesarias

- **Ejemplo:** (continuación)

Entonces es posible la siguiente secuencia de reducción:

$$\underline{if(if(true, false, true), \underline{a}, \underline{a})} \rightarrow_{S_{po}} \underline{if(false, b, b)} \rightarrow_{S_{po}} b$$

que respeta la estrategia *parallel-outermost* y contrae 4 redexes.

Sistemas de reescritura de términos

Estrategias de Reescritura: reducciones necesarias

- **Ejemplo:** (continuación)

Sin embargo puede encontrarse la forma normal b reduciendo sólo 3 redexes:

$$\underline{if(if(true, false, true), a, a)} \rightarrow \underline{if(false, a, a)} \rightarrow \underline{a} \rightarrow b$$

Observación: no ha sido **necesario** reducir el redex a en la posición 2 del término inicial.

Sistemas de reescritura de términos

Estrategias de Reescritura: reducciones necesarias

- Un redex en un término t es **necesario** (para la normalización) si es contraído, él mismo o alguno de sus descendientes, en cualquier secuencia de normalización de t .
- Una **posición necesaria** (*needed*) es aquella que señala la aparición de un redex necesario.
- Las secuencias de reducción que sólo contraen redexes necesarios se denominan secuencias de **reducción necesaria**.

Sistemas de reescritura de términos

Estrategias de Reescritura: reducciones necesarias

- Teorema [Huet y Levi 1979]: Sea \mathcal{R} un TRS ortogonal.
 1. Todo término que no es una forma normal contiene un redex necesario.
 2. Sea t un término normalizable. No existen secuencias de reducción para t que reduzcan un número infinito de redexes necesarios.
 3. Toda secuencia de reducción necesaria para t es normalizante.

Sistemas de reescritura de términos

Estrategias de Reescritura: reducciones necesarias

- Las secuencias de reducción necesarias tienen, por tanto, un interés doble:
 1. son óptimas, en el sentido de que no realizan pasos de reducción inútiles;
 2. y además son normalizantes (la contracción repetida de los redexes necesarios conducirá a la forma normal).

Sistemas de reescritura de términos

Estrategias de Reescritura: reducciones necesarias

- Es indecidible saber, en general, si un redex es necesario o no.
- Sin embargo, existen subclases de los TRSs ortogonales en los que esta propiedad es decidible.
 1. TRS **fuertemente secuencial**;
 2. TRS **inductivamente secuencial** (= TRS basado en constructores y fuertemente secuencial).

Sistemas de reescritura de términos

Bibliografía

- Klop J., 1992. *Term Rewriting Systems*. En S. Abramsky, D. Gabbay y T. Maibaum, eds., *Handbook of Logic in Computer Science*, volumen I, págs. 1–112. Oxford University Press.
- N. Dershowitz y J.-P. Jouannaud, 1990. *Rewrite Systems*, En J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, págs. 243–320 volumen B: *Formal Models and Semantics*, Elsevier, Amsterdam.

Sistemas de reescritura de términos

Bibliografía

- Baader F. y Nipkow T., 1998. *Term Rewriting and All That*. Cambridge University Press.
- D.A. Plaisted, 1996. *Equational Reasoning and Term Rewriting Systems*, En D.M. Gabbay, C.J. Hogger y J.A. Robinson, eds., *Handbook of Logic in Artificial Intelligence and Logic Programming*, volumen I, *Logical Foundations*, págs. 274–364. Oxford University Press.