ELiC11: Transductores

Clase 4: Pesos (2nda Parte)
y Arboles

Carlos Areces
carlos.areces@gmail.com

La clase pasada

- Operaciones sobre Transductores
  - Determinización
  - Eliminación de épsilon
  - Composición
  - Ejemplos
- Pesos
  - Autómatas

La clase de hoy

- Repaso
- Transductores con Pesos
  - Ejemplo
- Arboles
  - Autómatas
    - Top-down
    - Bottom-up
    - Propiedades
  - Transductores

Repaso

Composición

- Dados dos autómatas sin transiciones ϵ

  T1 = $<Q,\Sigma,\Sigma',\Delta,q_0>$

  T2 = $<Q',\Sigma',\Sigma'',\Delta',q'_0>$

  la composición de T1 y T2 (T1oT2) se define como

  T1oT2 = $<Q \times Q',\Sigma,\Sigma'',\Delta'',(q_0,q'_0)>$

  donde

  $\Delta'' = \{(q_s,q_s')a \vdash b(q_d,q_d') \mid$

  $q_s a \vdash c q_d,\ q'_s c \vdash b q'_d$ para algún $c \in \Sigma'\}$

Ultimo

- Cada vez que enviamos un SMS estamos usando un transductor:
  - {a,b,c} => 2 {d,e,f} => 3 {g,h,i} => 4
  - {j,k,l} => 5 {m,n,o} => 6 {p,q,r,s} => 7
  - {t,u,v} => 8 {w,x,y,z} => 9 {▫} => #
- Es fácil construir este transductor T_{cel} y tenemos, por ejemplo:
  - En T_{cel}: $q_0 casa\# \vdash^* 2272\#$
- Como hacemos al reves? (de números a palabras).
  - En $inv(T_{cel})$: $q_0 2272\# \vdash^* casa\#$

    $q_0 2272\# \vdash^* bbqb\#$
- Componer con un diccionario: $inv(T_{cel})\ o\ T_{dic}$

Autómatas con pesos

- Resumiendo entonces:
  - Un autómata con pesos no es otra cosa que un transductor generando sobre un semi-anillo particular.
  - Todas las propiedades de transductores se aplican a los autómatas con pesos.

Transductores con pesos

- Vimos entonces que los autómatas con pesos son transductores
- Qué seran entonces los transductores con pesos?
- Ayudita:
  - Siempre podemos ver una tripla como un par donde una componente es un par:

    (a,b,c) ~ (a, (b,c))

Fin del repaso

Ejemplo de transductores con pesos

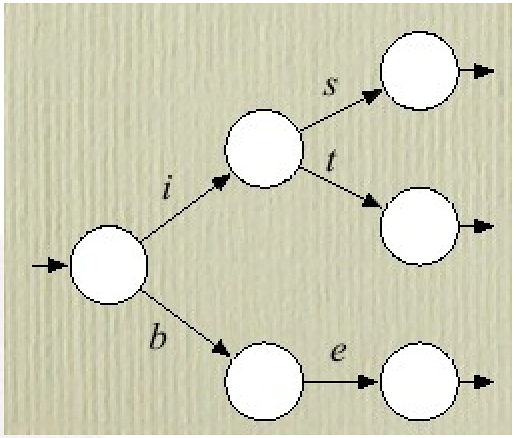
- Asumamos (y no estamos muy lejos de la realidad, visto los errores de ayer) que soy un poco disléxico:
  - Muchas veces (digamos un 30%), cuando quiero escribir s, escribo t.
  - Cómo sería mi transductor?

Ejemplo de transductores con pesos

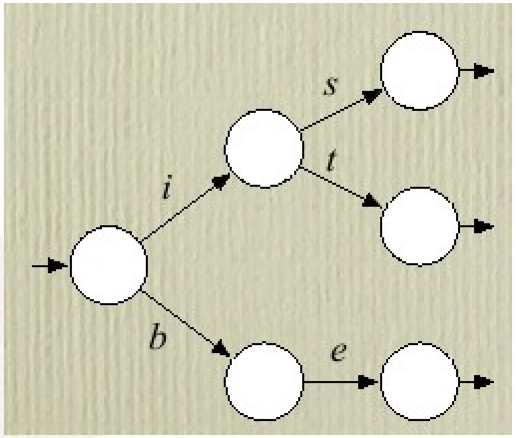
- Asumamos (y no estamos muy lejos de la realidad, visto los errores de ayer) que soy un poco disléxico:
  - Muchas veces (digamos un 30%), cuando quiero escribir s, escribo t.
  - Cómo sería mi transductor?
  - Concentrémonos en las palabras is y it
  - Si una persona 'normal' las escribe más o menos la misma cantidad de veces. Cómo es la distribución en mi caso?

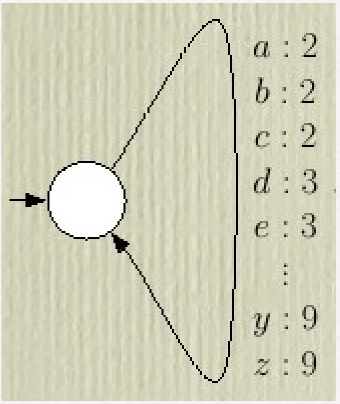
Un ejemplo más interesante

- Volvamos al ejemplo del celular
  - Diccionario: D =



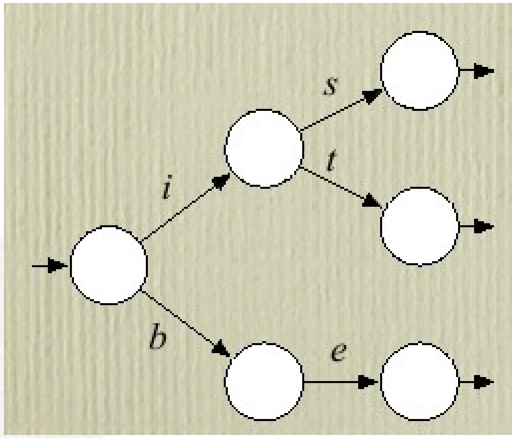
Un ejemplo más interesante

- Volvamos al ejemplo del celular
  - Diccionario: D = 

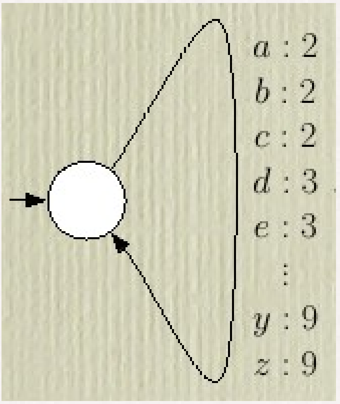
  - Keypad: K = 

Un ejemplo más interesante

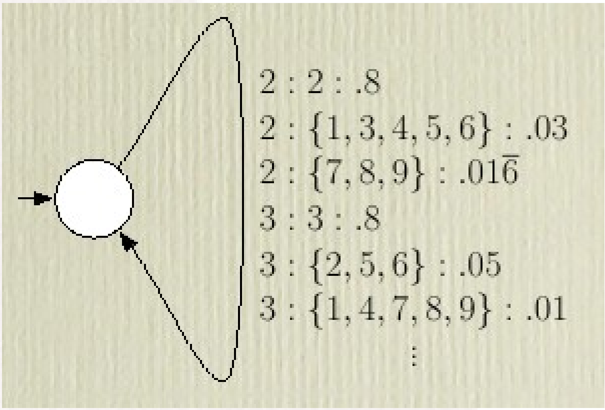
- Volvamos al ejemplo del celular
  - Diccionario: D =



  - Keypad: K =



  - Errores de tipeo: E =



Un final feliz

- E es un **trasductor con pesos**: produce en el semi-anillo (string, prob.).
- K es un **transductor** (a secas): produce en el semi-anillo strings, pero podemos 'subirlo' a (string, prob.) asignando probabilidades equitativamente (K^{prob}).
- D es un **autómata**: no produce nada (el pobre), pero podeos 'subirlo' a (string, prob.) haciendole producir la identidad y asignando probabilidades equitativamente ($D^{id,prob}$).
- Ahora todos pueden vivir juntos y en armonía en la composición: inv($D^{id,prob}$ o K^{prob} o E)

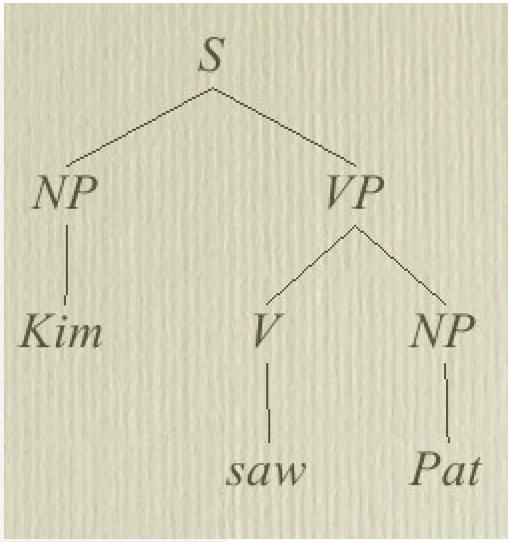
Arboles

Arboles o cadenas?

- “Cuando uno tiene un martillo todo se parece a un clavo”

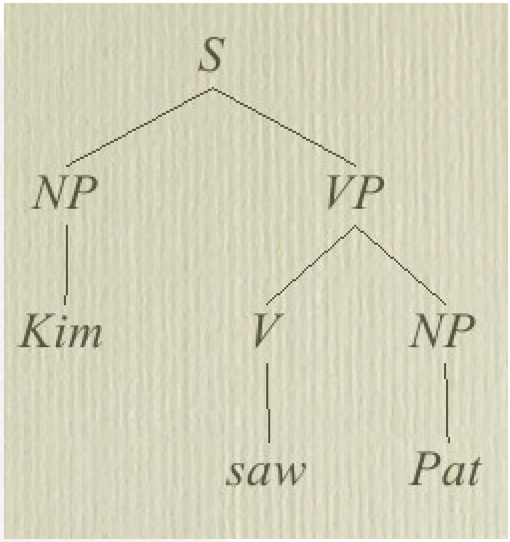
Arboles o cadenas?

- “Cuando uno tiene un martillo todo se parece a un clavo”
- Miremos con fuerza este árbol. Cómo podemos verlo como una cadena?

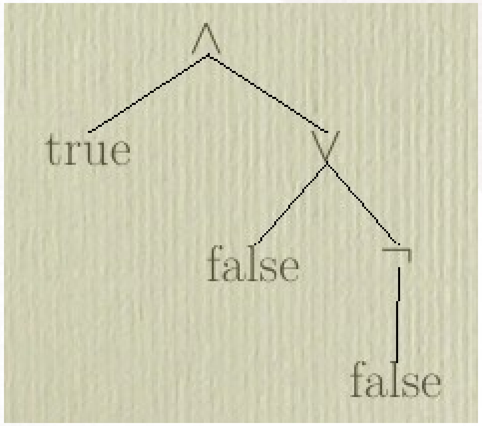


Arboles o cadenas?

- “Cuando uno tiene un martillo todo se parece a un clavo”
- Miremos con fuerza este árbol. Cómo podemos verlo como una cadena?

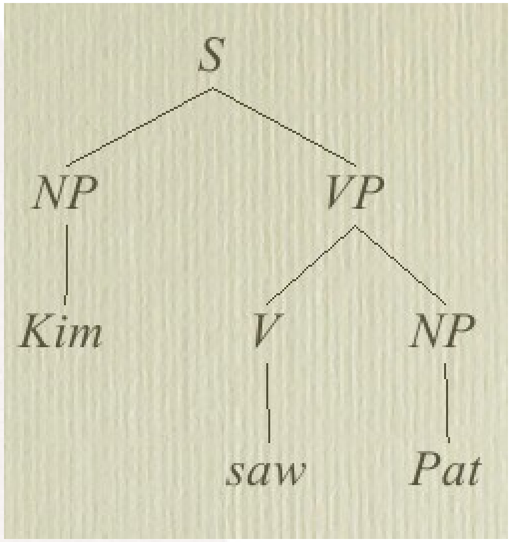


- Uno un poco más fácil:

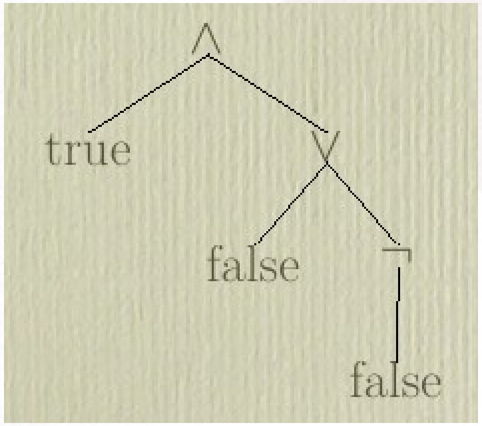


Arboles o cadenas?

- “Cuando uno tiene un martillo todo se parece a un clavo”
- Miremos con fuerza este árbol. Cómo podemos verlo como una cadena?



- Uno un poco más fácil:



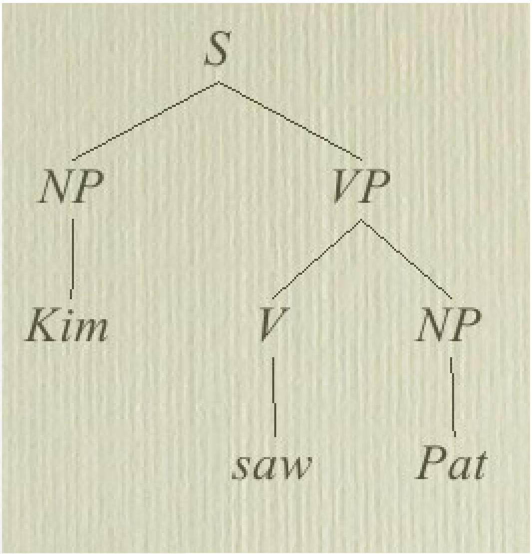
true ^ (false v - false)

Arboles o cadenas

- Todo árbol puede verse como un término sobre el alfabeto adecuado.

  S(NP(Kim),VP(V(saw),NP(Pat)))

  $F=\{S^2, NP^1, Kim^0, VP^2, V^1, saw^0, NP^1, Pat^0\}$}



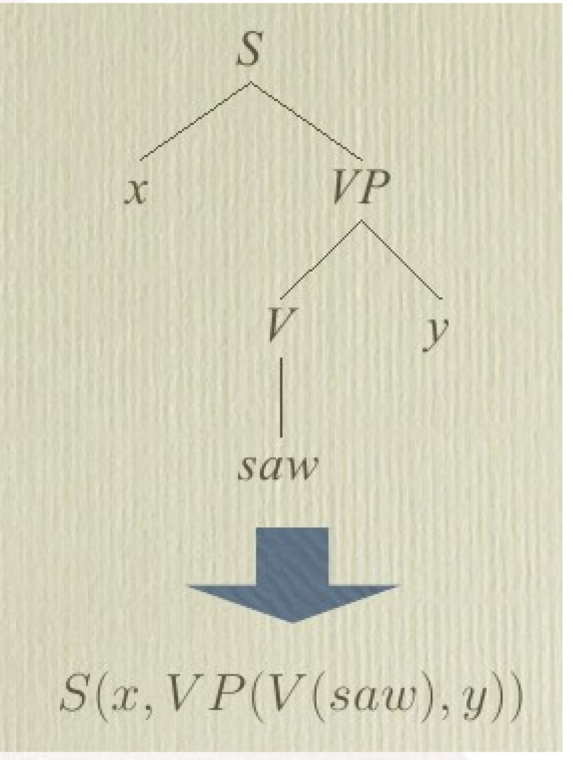
Arboles o cadenas

- Todo árbol puede verse como un término sobre el alfabeto adecuado.

  S(NP(Kim),VP(V(saw),NP(Pat)))

  $F=\{S^2, NP^1, Kim^0, VP^2, V^1, saw^0, NP^1, Pat^0\}$

- Vamos a usar variables para representar árboles incompletos



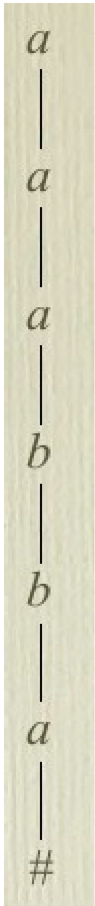
Definiendo árboles

- El conjunto de los árboles sobre un conjunto de funciones *F* y un conjunto de variables *X* (not. $\mathfrak{T}(F,X)$) es el menor conjunto tq.
  - $f \in \mathfrak{T}(F,X)$, para aridad(f) = 0.
  - $x \in \mathfrak{T}(F,X)$, para $x \in X$.
  - $f(t_1,\ldots,t_n) \in \mathfrak{T}(F,X)$ si aridad(f) = n y $t_1,\ldots,t_n \in \mathfrak{T}(F,X)$

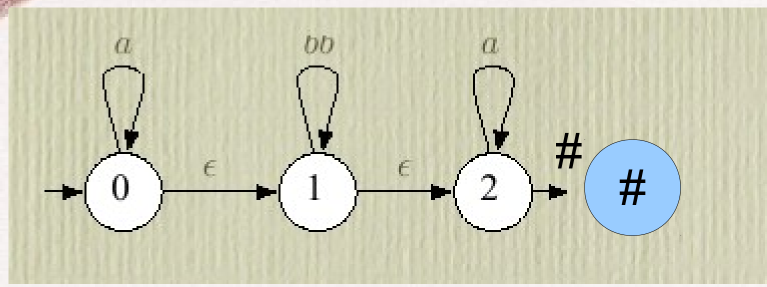
Cadenas como árboles

- Sea w es una cadena sobre Σ.
- La podemos ver como un árbol lineal:
  - Los símbolos de Σ son funciones unarias.
  - El símbolo # es una función 0-aria.

aaabbaa => a(a(a(b(b(a(a(#)))))))



Autómatas de strings



Reglas de reescritura

$q_0\, a \vdash q_0$	$q_0 \vdash q_1$
$q_1\, bb \vdash q_1$	$q_1 \vdash q_2$
$q_2\, a \vdash q_2$	$q_2 \sharp \vdash \sharp$

Una derivación:

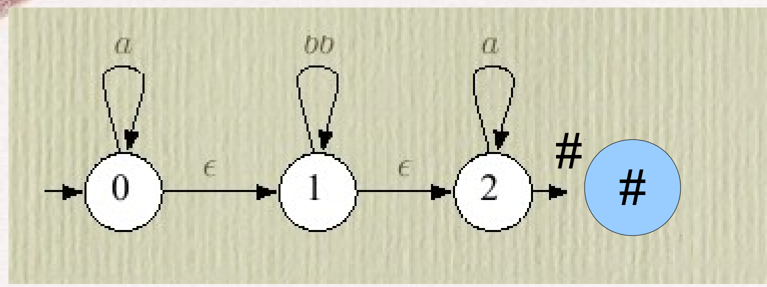
$q_0 aaabba\# \vdash q_0 aabba\# \vdash q_0 abba\# \vdash$

$\vdash q_0 abba\# \vdash q_1 bba\# \vdash q_1 a\#$

$\vdash q_2 a\# \vdash q_2\# \vdash \#$

Def. de aceptación: aceptar w sii $q_0 w\# \vdash^* \#$

Autómatas de strings



Reglas de reescritura

$$
\begin{array}{ll}
q_0\, a \vdash q_0 & q_0 \vdash q_1 \\
q_1\, bb \vdash q_1 & q_1 \vdash q_2 \\
q_2\, a \vdash q_2 & q_2 \sharp \vdash \sharp
\end{array}
$$

Una derivación:

$q_0 aaabba\# \vdash q_0 aabba\# \vdash q_0 abba\# \vdash$

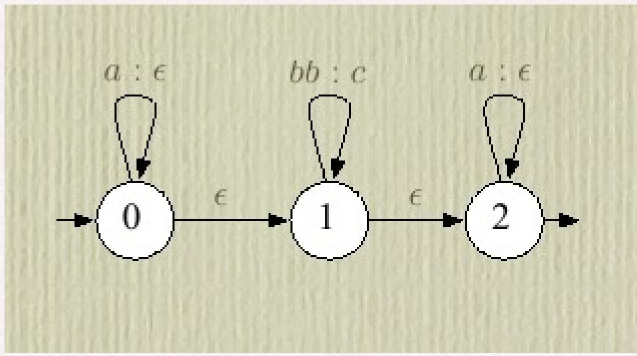
$\vdash q_0 abba\# \vdash q_1 bba\# \vdash q_1 a\#$

$\vdash q_2 a\# \vdash q_2\# \vdash \#$

Def. de aceptación: aceptar w sii $q_0 w\# \vdash^* \#$

AGREGAR PARENTESIS DONDE CORRESPONDA

Transductores



Función de transición

$$
\begin{array}{ll}
q_0\, a \vdash q_0 & q_0 \vdash q_1 \\
q_1\, bb \vdash c\, q_1 & q_1 \vdash q_2 \\
q_2\, a \vdash q_2 & q_2 \sharp \vdash \sharp
\end{array}
$$

Una derivación:

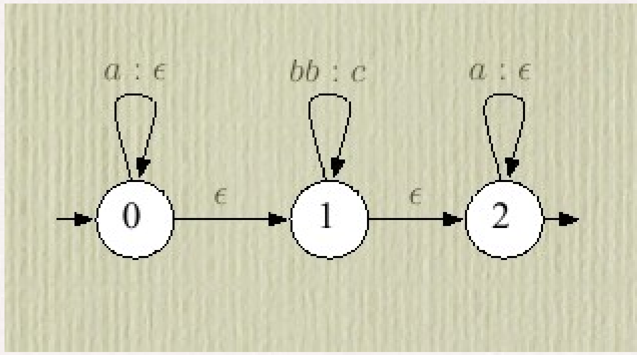
$q_0 aaabba\# \vdash q_0 aabba\# \vdash q_0 abba\# \vdash$

$\vdash q_0 abba\# \vdash q_1 bba\# \vdash c q_1 a\#$

$\vdash c q_2 a\# \vdash c q_2 \# \vdash c\#$

Def. de aceptación: aceptar s:t sii $q_0 s\# \vdash^* t\#$

Transductores



Función de transición

$q_0\, a \vdash q_0$	$q_0 \vdash q_1$
$q_1\, bb \vdash c\, q_1$	$q_1 \vdash q_2$
$q_2\, a \vdash q_2$	$q_2 \sharp \vdash \sharp$

Una derivación:

$$\mathrm{q}_0 aaabba\# \vdash \mathrm{q}_0 aabba\# \vdash \mathrm{q}_0 abba\# \vdash$$

$$\vdash \mathrm{q}_0 abba\# \vdash \mathrm{q}_1 bba\# \vdash c\mathrm{q}_1 a\#$$

$$\vdash c\mathrm{q}_2 a\# \vdash c\mathrm{q}_2 \# \vdash c\#$$

Def. de aceptación: aceptar s:t sii $\mathrm{q}_0 s\# \vdash^* t\#$

AGREGAR PARENTESIS DONDE CORRESPONDA

Upper y Lower trees

- El conjunto de los **upper trees** sobre *F*, Q, X (not. $\mathfrak{T}^{Q}$*(F,X)*), es el conjunto de árboles $q(t) \in \mathfrak{T}$*(F* $\cup$ *Q,X)*

  donde $q \in$ Q, y $t \in$ *T(F,X)*.

  [En 'humano': empiezan con q y después no aparece nunca más un estado.]

Upper y Lower trees

- El conjunto de los **upper trees** sobre *F*, Q, X (not. $\mathfrak{T}^Q$*(F,X)*), es el conjunto de árboles $q(t) \in \mathfrak{T}$*(F* $\cup$ *Q,X)*

  donde $q \in$ Q, y $t \in \mathfrak{T}$*(F,X)*.

  [En 'humano': empiezan con q y después no aparece nunca más un estado.]

- EL conjunto de los lower trees sobre F, Q, X (not. $\mathfrak{T}_Q$*(F,X)*), es el conjunto T $\subseteq \mathfrak{T}$*(F* $\cup$ *Q,X)* tq.:

  - $f \in$ *T* si f 0-aria.
  - $q(x) \in$ *T* para $x \in$ X, $q \in$ Q.
  - $f(t_1,\ldots,t_n) \in$ *T* si $t_1,\ldots,t_n \in$ *T*.

  [En 'humano': los estados sólo aparecen sobre variables.]

Ejemplos

$F = \{f^2, g^2, a^0, b^0, c^0\}$
$Q = \{q_0, q_1\}$
$X = \{x_1, x_2, x_3\}$

$\mathfrak{T}^Q(F,X) \supset \{$ q0(x3),

q1(f(g(x1,x2),x3)),
q0(a)}

$\mathfrak{T}_Q(F,X) \supset \{$ q0(x3),

f(g(q0(x1),q0(x2)),q1(x3),
a}

Linearidad y altura

- Un árbol t es **lineal** si no tiene dos ocurrencias de la misma variable.
- La **altura de un árbol** (not. h(t)) se define como
  - h(x) = 0
  - h(a^0) = 1
  - h(f(t_1,...,t_n)) = 1 + max{h(t_1),...,h(t_n)}
  - h(q(t)) = h(t)

Subclases de árboles

- Los "**consecutively numbered linear upper trees of height at most 1**", T^Q, son de la forma

$$q(f(x_1,\ldots,x_n)).$$

- Los "**consecutively numbered linear lower trees of height at most 1**", T_Q, son de la forma

$$f(q(x_1),\ldots,q(x_n))$$

Autómata de árboles

Un **autómata de árboles no determinístico top-down (↓TA)** sobre F es una upla <Q,F,Δ,q_0> donde:

- Q es un conjunto finito de estados.
- F es un conjunto de símbolos de funcion.
- $\Delta \subseteq T^Q(F,X) \times T_Q(F,X)$ es un conjunto de transiciones tal que si Δ(t,t'), prim(t) = prim(t')
- q_0 es un estado de Q.

Un t de $\mathfrak{T}(F)$ es aceptado por un ↓TA si

$$q_0(\mathrm{t}) \vdash^* \mathrm{t}$$

Ejemplo

- Consideremos el siguiente $\downarrow$TA

$$q_0(f(x_1,x_2)) \vdash f(q_a(x_1),q_b(x_2))$$

$$q_a(a) \vdash a$$

$$q_a(f(x_1,x_2)) \vdash f(q_a(x_1),q_b(x_2))$$

$$q_b(b) \vdash b$$

$$q_b(f(x_1,x_2)) \vdash f(q_a(x_1),q_b(x_2))$$

Autómata de árboles

Un **autómata de árboles no determinístico bottom-up (↑TA)** sobre F es una upla $<Q,F,\Delta,Q_f>$

donde:

- Q es un conjunto finito de estados.
- F es un conjunto de símbolos de funcion.
- $\Delta \subseteq T_Q(F,X) \times T^Q(F,X)$ es un conjunto de transiciones tal que si Δ(t,t'), prim(t) = prim(t')
- Q_f es un subconjunto de Q.

Un t de $\mathfrak{T}(F)$ es aceptado por un ↑TA si

$$t \vdash^* q(t), \text{ para algún } q \text{ en } Q_f$$

Propiedades

- Los autómatas de árboles están relacionados con los lenguajes libres de contexto.
- Tienen varias de las propiedades de los autómatas sobre cadenas. Están cerrados por
  - (Left-right reversal)
  - Top-down bottom-up reversal
  - Unión
  - Granularidad
  - Épsilon removal $q(x) \vdash q'(x)$
  - Determinización (bottom-up solamente)

Transductores de árboles

Un **autómata de árboles no determinístico top-down (↓TA)** sobre *F* es una upla <Q,F,Δ,q_0> donde:

- *Q* es un conjunto finito de estados.
- *F* es un conjunto de símbolos de funcion.
- $\Delta \subseteq T^Q(F,X) \times T_Q(F,X)$ es un conjunto de transiciones tal que si Δ(t,t'), prim(t) = prim(t')
- q_0 es un estado de Q.

Un t de $\mathfrak{T}(F)$ es aceptado por un ↓TA si

$$q_0(t) \vdash^* t$$

Transductores de árboles

Un **transductor de árboles no determinístico top-down (↓TT)** sobre *F, F'* es una upla <Q,F,F',Δ,q_0>

donde:

- Q es un conjunto finito de estados.
- F es un conjunto de símbolos de funcion.
- $\Delta \subseteq T^Q(F,X) \times \mathfrak{T}_Q(F',X)$ es un conjunto de transiciones
- q_0 es un estado de Q.

Un par t:s de $\mathfrak{T}(F)$ es aceptado por un ↓TA si

$$q_0(t) \vdash^* s$$

Ejemplo

- Consideremos el siguiente $\downarrow$TT

$$q_0(\mathrm{f}(\mathrm{x}) \vdash \mathrm{g}(\mathrm{f}q_c(\mathrm{x}),\mathrm{f}q_c(\mathrm{x}))$$

$$q_c(\mathrm{f}(\mathrm{x})) \vdash \mathrm{f}(q_c(\mathrm{x})$$

$$q_c(\mathrm{a}) \vdash \mathrm{a}$$

Resumen del Curso

- Autómatas sobre cadenas
  - Muy buen comportamiento
- Transductores sobre cadenas
  - Comportamiento bastante bueno, sobre todo por estar cerrados bajo composición e inversión.
- Autómatas con pesos
  - Un tipo de transductor
- Transductores con pesos
  - Un tipo de transductor
- Autómatas sobre árboles
  - La cosa se complica. Manejables sólo porque imponemos restricciones en las transiciones posibles.
- Transductores sobre árboles
  - Extremandamente poderosos (en muchos aspectos).
  - No suficientes en otros.