13068. Teoria d'Autòmats i Llenguatges Formals

4. Gramàtiques incontextuals i autòmats amb pila

Francesc J. Ferri
Dept. d'Informàtica. Universitat de València

15 de gener de 2003

Incontextuals?

Gramàtiques independents del context o també de context lliure (tipus 2 en la jerarquia de Chomsky). Les anomenarem *incontextuals* per fer-ho curt.

> Atenció: en el context de les gramàtiques contextuals i incontextuals contextual no s'oposa a incontextual. Per tant, contextual, no contextual, incontextual i no incontextual seran coses diferents.

> Què lio, no?

Definició: Gramàtiques amb produccions de la forma:

Símbol→qualsevol cosa

Però, per què?

Perque hi ha coses més enllà del mon regular:

- cadenes amb algun tipus de parèntesis.
- cadenes amb algun factor repetit.
- cadenes que requeresquen algun tipus de comptatge.
- cadenes amb algun tipus d'estructura.

La majoria d'aquestes coses es poden modelitzar mijantçant gramàtiques incontextuals

Alguns exemples

Un vell conegut: $\{a^n b^n | n \geq 0\}$ $S \rightarrow aSb|\varepsilon$

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow \cdots \Rightarrow a^k S b^k \Rightarrow a^k b^k$$

Aquesta gramàtica és de les incontextuals més senzilles. S'anomenen lineals (un únic no terminal a la dreta).

Un exemple més interessant: $S \rightarrow S + A|a \quad A \rightarrow S + A|a$

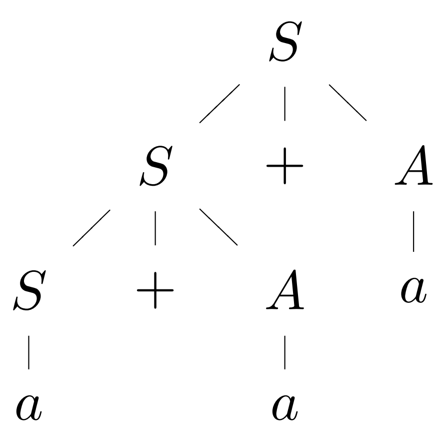
$$S \Rightarrow S + \underline{A} \Rightarrow \underline{S} + S + A \Rightarrow \cdots S + A + \cdots + A \overset{\star}{\Rightarrow} a + a + \cdots + a$$

Expressions aritmètiques amb un únic operador.

Cadenes "arborescents"?

En permetre més d'un símbol no terminal en les formes sentencials estem donant una **estructura** a les cadenes generades.

$$S \Rightarrow S + A \Rightarrow S + A + A \overset{\star}{\Rightarrow} a + a + a$$



Aquesta estructura implica una certa **interpretació** de la cadena.

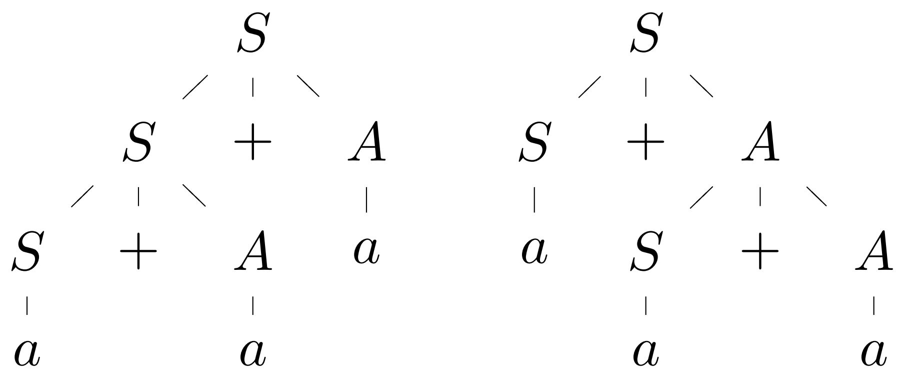
Arbres de derivació

- A tota derivació li correspon un únic arbre de derivació (no al revés)

- Una manera de representar un arbre de derivació (i totes les derivacions que li corresponen) és mijantçant la **derivació més a l'esquerra (o dreta)** o derivació on les produccions s'apliquen primer als no terminals més a l'esquerra (o dreta).

- Un arbre de derivació representa una posible **interpretació** d'una cadena

- Es possible associar a una mateixa cadena més d'una interpretació


S
S + A
S + A a
a a
S
S + A
a S + A
a a

Ambigüitat

Gramàtica ambigua: si pot generar alguna cadena amb més d'una interpretació (la de l'exemple anterior és ambigua.)

- **Vol dir això que no podem interpretar unívocament expressions aritmètiques?**

No, perque pot haver una gramàtica equivalent que no siga ambigua.

$$S \rightarrow a + S | a$$

Gramàtiques inherentment ambigües

G. inherentment ambigües gramàtiques per a les quals no existeixen gramàtiques equivalents no ambigües.

Els corresponentes llenguatges s'anomenen **llenguatges ambigus**.

Per cert, el llenguatge anterior no només no era ambigu sinó que tampoc no era no regular!

O siga, que era regular: $a(+a)^\star$

Moral: La estructura (interpretació) la imposa la gramàtica, no el llenguatge.

Si però... algun llenguatge ambigu?

Bé, doncs, jo ...

La veritat és que són un poc rarets.

En fi, un relativament senzill és:

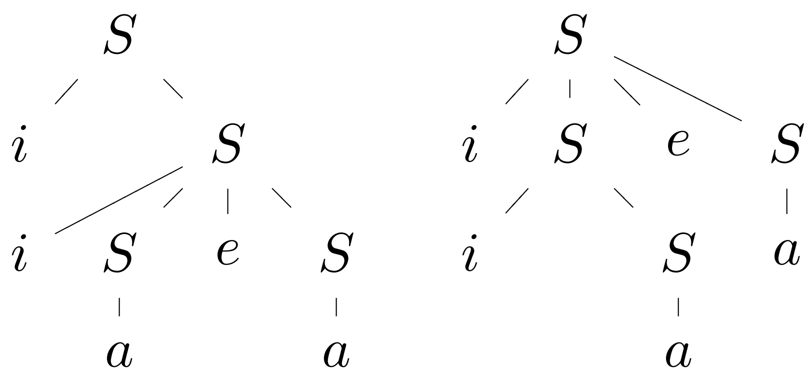
$$\{a^i b^j c^k | i = j \vee j = k\}$$

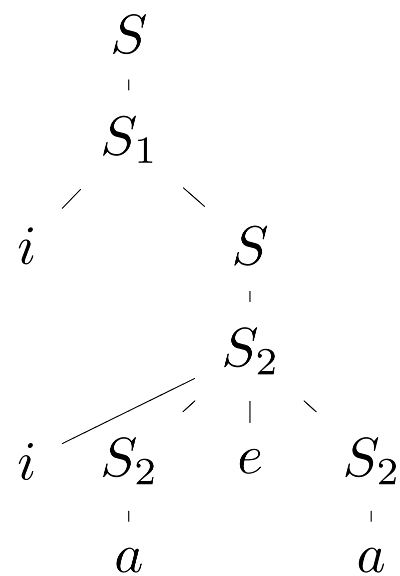
Exercici: trobar gramàtiques equivalents i comprovar que totes són ambigües

(Les demostracions d'ambigüitat de llenguatges són molt complexes.)

Un exemple interessant i pràctic

$S \longrightarrow iS|iSeS|a$





La gramàtica no ambigua equivalent és:

$$
\begin{aligned}
S &\longrightarrow S_1|S_2 \\
S_1 &\longrightarrow iS|iS_2eS_1 \\
S_2 &\longrightarrow iS_2eS_2|a
\end{aligned}
$$

Manipulació de gramàtiques

• Les produccions de les gramàtiques es poden manipular sense que canvie el llenguatge generat.

Les modificacions bàsiques més importants són:

substitució d'un no terminal per les corresponents parts dretes.

factorització: substitució d'un factor comú en diverses parts dretes per un nou no terminal.

inversió de bucles: bucles d'esquerres a dretes i de dretes a esquerres.

Aplicacions: eliminació de regles nul·les, eliminació de regles de redenominació, eliminació de recursivitat a esquerres.

Gramàtiques ben formades

Símbol productiu: si a partir d'ell es pot arribar a una cadena de terminals

Símbol abastable: si a partir de l'axioma es pot arribar a una forma sentencial que el continga

Si un símbol no es productiu o no és abastable es pot eliminar.

Gramàtica ben formada: Només amb símbols productius i abastables, sense regles nul·les ni de redenominació.

Símbols productius

A **és productiu si** $\exists w \in {V_T}^\star$ tal que $A \overset{\star}{\Rightarrow} w$

- si $A \rightarrow w \in P$ i $w \in {V_T}^\star \implies A$ és productiu (trivial).
- Siga V_P un conjunt de símbols productius:

  si $A \rightarrow x \in P$ i $x \in (V_T \cup V_P)^\star \implies A$ és productiu.

Es poden calcular els símbols productius comprovant les parts dretes de les produccions de la gramàtica.

Símbols abastables

X **és abastable si** $\exists \alpha, \beta \in V^{\star}$ tal que $s \stackrel{\star}{\Rightarrow} \alpha X \beta$

- si $S \rightarrow \alpha X \beta \in P \implies X$ és abastable (trivial).
- Siga $A \in V_N$ un símbol abastable:

  si $A \rightarrow \alpha X \beta \in P \implies X$ és abastable.

Es poden calcular els símbols abastables a partir de les parts dretes de les produccions de la gramàtica.

Eliminació de bucles a esquerres

Una regla recursiva a esquerres és en realitat un conjunt de produccions de la forma:

$$A \rightarrow A\alpha_1 | \cdots | A\alpha_n | \beta_1 | \cdots | \beta_m$$

O, en forma compacta:

$$A \rightarrow AB | C$$

on $B \rightarrow \alpha_1 | \cdots | \alpha_n$ i $C \rightarrow \beta_1 | \cdots | \beta_m$.

[Eliminació de bucles a esquerres]

La regla $A \rightarrow AB|C$, dóna lloc a derivacions de la forma:

$$A \quad \stackrel{\star}{\Rightarrow} \quad CB^k, k \geq 0$$

i és equivalent a les produccions:

$A \rightarrow C|CD$
$D \rightarrow BD|\varepsilon$

Forma Normal de Chomsky (FNC)

Una gramàtica està en Forma Normal de Chomsky (FNC) si totes les seues produccions són de la forma

$$A \rightarrow BC \text{ o } A \rightarrow a$$

Conseqüència: Tots els arbres de derivació són binaris

Teorema: Tota gramàtica incontextual es pot posar en FNC (N'existeix una equivalent en FNC)

Obtenció d'una gramàtica en FNC

Sempre es poden eliminar els símbols terminals intercalats en les parts dretes de les produccions introduint nous no terminals A_a per cada símbol terminal a.

Una vegada G no té terminals intercalats, les produccions de la forma $A \to B_1 B_2 \dots B_k$ es poden descomposar en:

$$\boxed{\begin{array}{l} A \to B_1 C_1 \\ C_1 \to B_2 C_2 \\ \vdots \\ C_{k-2} \to B_{k-1} B_k \end{array}}$$

amb la qual cosa s'obté una gramàtica en FNC.

Forma Normal de Greibach (FNG)

Una gramàtica està en FNG si totes les seues produccions són de la forma

$$A \rightarrow a\gamma, \text{ per a qualsevol } \gamma \in {V_N}^\star.$$

Conseqüència Tota producció introdueix un nou terminal en la forma sentencial corresponent. Les derivacions més a l'esquerra són sempre de la forma:

$$S \stackrel{\star}{\Rightarrow} a_1 \cdots a_n A_1 \cdots A_m$$

Teorema: Tota gramàtica incontextual es pot posar en FNG.

Obtenció d'una gramàtica en FNG

Suposem G ben formada sense terminals intercalats i sense recursivitats a esquerres.

pas 1: Obtenció d'una gramàtica ordenada.

Siga $A_1, \ldots, A_k$ una ordenació qualsevol de $V_N = \{A_i\}_{i=1}^k$.

G està ordenada si totes les produccions són de la forma $A_i \rightarrow A_j\gamma$ amb $i < j$ (o $A_i \rightarrow a\gamma$) per a qualsevol γ.

Raonament inductiu: Si les produccions associades a $A_1, \ldots, A_i$ estan ordenades, podem ordenar les produccions associades a A_{i+1} substituint el primer no terminal de la part dreta en el cas que siga menor. Si fóra igual, eliminaríem la recursivitat a esquerres.

Cas base: Les produccions associades a A_1 sempre estan ordenades.

[Obtenció d'una gramàtica en FNG]

pas 2: Obtenció d'una gramàtica en FNG a partir d'una ordenada.

Raonament inductiu: Si les produccions associades a $A_i, \ldots, A_k$ estan en FNG, podem convertir les de A_{i-1} a FNG substituint el primer no terminal de la part dreta.

Cas base: les produccions associades a A_k sempre estan en FNG.

Les produccions associades a no terminals resultat d'eliminar recursivitats a esquerres es poden convertir a FNG de la mateixa manera.

Exemple (FNC)

$S \to S + S|(S)|a$

Introduïm els nous símbols M, O i T i obtenim

$S \to SMS|OST|a$
$M \to +$, $O \to ($, $T \to)$

Si descomposem les dues primeres produccions:

$$
\begin{array}{lll}
S \to SA|OB|a & & \\
A \to MS & B \to ST & \\
M \to + & O \to (& T \to)
\end{array}
$$

que està en FNC.

Exemple (FNG, pas 1)

Numerem ara els no terminals de la gramàtica anterior:

$$
\begin{array}{lll}
S_1 \rightarrow S_1A_2|O_5B_3|a & & \\
A_2 \rightarrow M_4S_1 & B_3 \rightarrow S_1T_6 & \\
M_4 \rightarrow + & O_5 \rightarrow (& T_6 \rightarrow)
\end{array}
$$

i, sense recursivitat a esquerres:

$$
\begin{array}{lll}
S_1 \rightarrow O_5B_3P|aP|O_5B_3|a & & \\
P \rightarrow A_2P|A_2 & & \\
A_2 \rightarrow M_4S_1 & B_3 \rightarrow S_1T_6 & \\
M_4 \rightarrow + & O_5 \rightarrow (& T_6 \rightarrow)
\end{array}
$$

[Exemple (FNG, pas 1)]

L'única producció no ordenada és $B_3 \rightarrow S_1T_6$.

que és equivalent a $B_3 \rightarrow O_5B_3PT_6|aPT_6|O_5B_3T_6|aT_6$

En conjunt:

$$
\begin{array}{lllll}
1) & S_1 \rightarrow O_5B_3P|O_5B_3 & & & \\
 & A_2 \rightarrow M_4S_1 & B_3 \rightarrow O_5B_3PT_6|O_5B_3T_6 & & \\
 & & & & \\
2) & P \rightarrow A_2P|A_2 & & & \\
 & & & & \\
3) & S_1 \rightarrow aP|a & B_3 \rightarrow aPT_6|aT_6 & & \\
 & M_4 \rightarrow + & O_5 \rightarrow (& & T_6 \rightarrow)
\end{array}
$$

Exemple (FNG, pas 2)

Considerant les produccions del primer grup:

$$\begin{array}{l} S_1 \to (B_3P|(B_3 \\ A_2 \to +S_1 \\ B_3 \to (B_3PT_6|(B_3T_6 \end{array}$$

i les del segon grup: $P \to +S_1P| + S_1$.

El resultat global és:

$$\boxed{\begin{array}{l} S_1 \to (B_3P|(B_3 \\ A_2 \to +S_1 \\ B_3 \to (B_3PT_6|(B_3T_6 \\ P \to +S_1P| + S_1 \\ S_1 \to aP|a \\ B_3 \to aPT_6|aT_6 \\ M_4 \to + \qquad O_5 \to (\qquad T_6 \to) \end{array}}$$

que està en FNG.

Autòmats amb pila

- Anem a estendre el concepte d'autòmat finit per tal d'acceptar llenguatges més generals.
- Afegim una pila de símbols especials a l'autòmat finit.
- Una pila permet emmagatzemar una quantitat **infinita** de símbols.
- Entre d'altres coses amb una pila es pot comptar i emmagatzemar parts de la cadena d'entrada.

Una pila?

- Una pila es pot veure com un dispositiu abstracte que permet emmagatzemar símbols i extraure'n de manera que el símbol que s'extrau és l'últim que s'ha afegit.

- Com a tipus abstracte de dades, la pila ve donada per un conjunt d'operacions que han de complir una sèrie d'equacions.

- Com a estructura, la pila es pot implementar de moltes maneres per tal que complesca les especificacions corresponents.

- Per a nosaltres, una pila és una cadena de símbols sobre la qual es fa una única operació.

Pila de símbols

Donada la pila $\boxed{ABCDE}$ i la cadena XYZ, l'única operació que considerem consisteix a substituir el "cap" o "cim" de la pila (A en aquest cas) per la cadena XYZ. El resultat és $\boxed{\text{XYZBCDE}}$.

Important: l'operació no està definida si la pila està buida!

Casos particulars:

Apilar un símbol X: cadena XA

Desapilar un símbol: cadena ε

Autòmat amb pila informalment

L'única cosa que cal afegir a un autòmat finit és un alfabet de pila i modificar la funció de transició per tal que incloga la operació sobre la pila.

Inicialment, se suposarà que la pila conté un símbol de pila especial que anomenarem *inicial*.

Ara, el comportament de l'autòmat dependrà també del símbol al cim de la pila (a banda de l'estat actual i del símbol a l'entrada). Per tant,

$$\delta : Q \times \Sigma \times \Gamma \longrightarrow Q \times \Gamma^{\star}$$

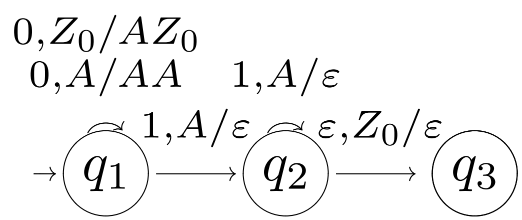
Autòmat amb pila: definició

Un **autòmat amb pila**, $A = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$

- Q és un conjunt finit d'estats.
- Σ és l'alfabet d'entrada.
- Γ és l'alfabet de pila ($\Sigma \cap \Gamma = \emptyset$).
- $q_0 \in Q$ és l'estat inicial.
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \longrightarrow 2^{Q \times \Gamma^\star}$ és la **funció de transició**.
- $Z_0 \in \Gamma$ és el símbol de pila inicial.
- $F \subseteq Q$ és el (sub)conjunt d'estats finals o d'acceptació.

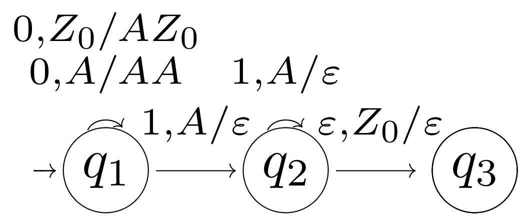
Exemple

Les transicions són ara de la forma $\delta(q, a, X) = (p, \alpha)$.



	$(0, Z_0)$	$(0, A)$	$(1, Z_0)$	$(1, A)$	(ε, Z_0)
$\rightarrow q_1$	(q_1, AZ_0)	(q_1, AA)		(q_2, ε)	
q_2				(q_2, ε)	(q_3, ε)
q_3					

Com funciona?



Inicialment, la pila conté Z_0, i per cada símbol 0 s'apila un símbol A.

A continuació, per cada símbol 1, es desapila una A.

L'autòmat només passa a l'estat q_3 si Z_0 torna a estar al cim de la pila.

Descripció instantània

La configuració o descripció instantània d'un autòmat amb pila ve donada per l'estat actual, la cadena pendent de lectura i el contingut de la pila: (q, x, γ).

Els canvis de configuració seran:

$$(q, ax, Z\gamma) \vdash (p, x, \alpha\gamma)$$

si i només si $(p, \alpha) \in \delta(q, a, Z)$

L'exemple anterior:
$(0011, q_1, Z_0) \vdash (011, q_1, AZ_0) \vdash (11, q_1, AAZ_0) \vdash (1, q_2, AZ_0) \vdash (\varepsilon, q_2, Z_0) \vdash (\varepsilon, q_3, \varepsilon)$

Autòmats amb pila deterministes

Els autòmats amb pila es defineixen de la forma més general (indeterministes i amb transicions buides).

Com són els autòmats amb pila deterministes? És determinista el de l'exemple anterior?

definició intuïtiva: un autòmat és determinista si en tot moment la configuració següent es pot deduir de forma unívoca a partir de la configuració actual i la seua descripció.

Conclusió: les transicions buides no sempre impliquen indeterminisme!

Autòmats amb pila deterministes. Definició

Direm que un autòmat amb pila és **determinista** si per a tot estat q i símbol de pila A es compleix

1. $|\delta(q, x, A)| \leq 1$ per a tot $x \in \Sigma \cup \{\varepsilon\}$.

2. $\delta(q, x, A) = \emptyset \;\; \forall x \in \Sigma$, si $\delta(q, \varepsilon, A) \neq \emptyset$.

L'autòmat de l'exemple anterior és determinista.

Llenguatge(s) acceptat(s)

L'addició de la pila obre la possibilitat a fer la servir per decidir l'acceptació o no de cadenes (a banda dels estats finals).

Criteri de pila nul·la: Una cadena és acceptada si en llegir tots els seus símbols l'autòmat buida la pila.

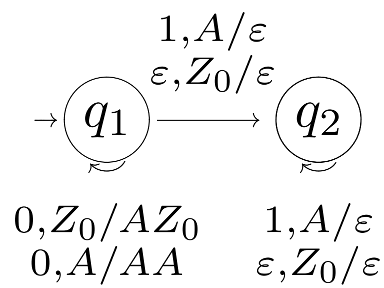
$$L_N(A) = \{x \in \Sigma^\star | (q_0, x, Z_0) \vdash (p, \varepsilon, \varepsilon) p \in Q\}$$

Criteri d'estat final: Una cadena és acceptada si en llegir tots els seus símbols l'autòmat arriba a un estat final.

$$L_F(A) = \{x \in \Sigma^\star | (q_0, x, Z_0) \vdash (p, \varepsilon, \gamma) \wedge p \in F,\ \gamma \in \Gamma\}$$

En l'exemple anterior $L_N(A) = L_F(A) = \{0^n 1^n | n \geq 1\}$

Un altre exemple



$$L_N(A) = \{0^n 1^n | n \geq 0\}$$

$$L_F(A) = \{0^n 1^m | n \geq 0,\ n \geq m\}$$

De L_F a L_N

Donat $A = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle$, es pot trobar A' tal que $L_F(A) = L_N(A')$.

$$A' = \langle Q \cup \{p_0, r\}, \Sigma, \Gamma \cup \{Y_0\}, \delta', p_0, Y_0, \emptyset\rangle$$

1. $\delta'(p_0, \varepsilon, Y_0) = \{(q_0, Z_0Y_0)\}$.

2. $\delta'(q, a, X) = \delta(q, a, X)$, $\forall q \in Q, a \in \Sigma \cup \{\varepsilon\}, X \in \Gamma$.

3. $\delta'(q, \varepsilon, X) = \delta(q, \varepsilon, X) \cup \{(r, \varepsilon)\}$, $\forall q \in F, X \in \Gamma$.

4. $\delta'(r, \varepsilon, X) = \{(r, \varepsilon)\}$, $\forall X \in \Gamma \cup \{Y_0\}$.

De L_N a L_F

Donat $A = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$, es pot trobar A' tal que $L_N(A) = L_F(A')$.

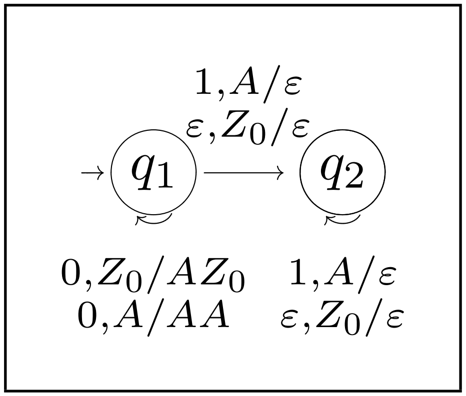
$$A' = \langle Q \cup \{p_0, r\}, \Sigma, \Gamma \cup \{Y_0\}, \delta', p_0, Y_0, \{r\} \rangle$$

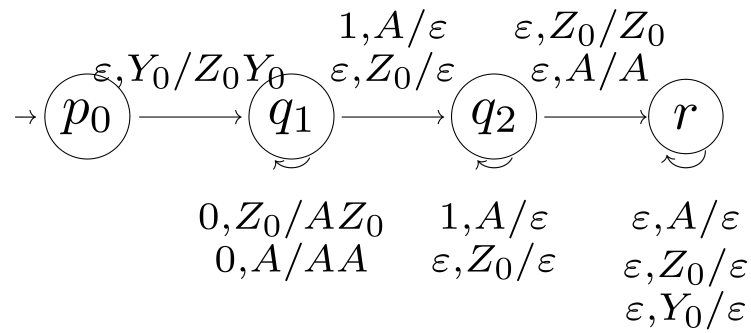
1. $\delta'(p_0, \varepsilon, Y_0) = \{(q_0, Z_0 Y_0)\}$.

2. $\delta'(q, a, X) = \delta(q, a, X)$, $\forall q \in Q, a \in \Sigma \cup \{\varepsilon\}, X \in \Gamma$.

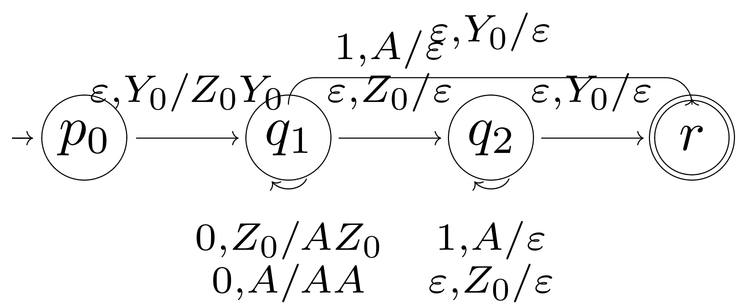
3. $\delta'(q, \varepsilon, Y_0) = \{(r, \varepsilon)\}$, $\forall q \in Q$.

Exemple





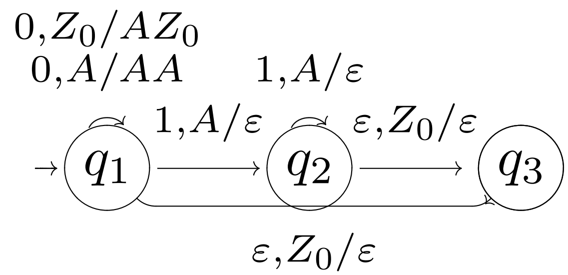
$$L_F(A) = L_N(A') = \{0^n 1^m | n \geq 0,\ n \geq m\}$$



$$L_N(A) = L_F(A') = \{0^n 1^n | n \geq 0\}$$

Autòmats deterministes

Altre autòmat que accepta $\{0^n 1^n | n \geq 0\}$ buidant la pila:



Es possible trobar-ne un d'equivalent que siga determinista?

No, perque si l'autòmat ha de buidar la pila amb ε com a entrada i és determinista, sempre buidarà la pila sense llegir símbols i no podrà acceptar res més.

Llenguatges acceptats per AP deterministes

Més general, un autòmat amb pila determinista no pot acceptar (mijantçant el criteri de pila buida) una cadena i un prefix d'aquesta simultàniament.

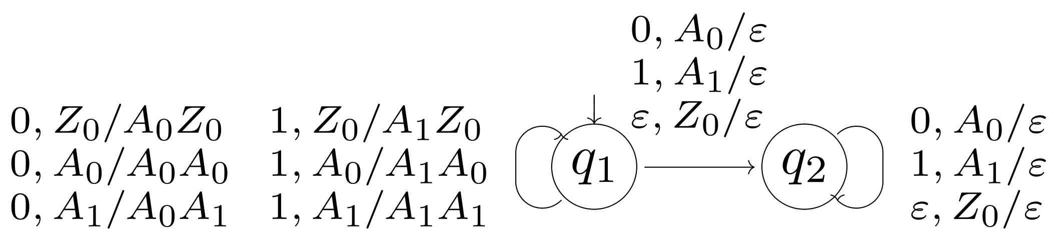
(si buida la pila per al prefix, no podrà llegir cap símbol més enllà)

Els autòmats amb pila que buiden la pila no poden acceptar llenguatges que complesquen la propietat de prefix. (Llenguatges que contenen algun parell de cadenes i una és prefix de l'altra).

Aquesta limitació desapareix si acceptem per estat final.

Llenguatges acceptats per AP deterministes

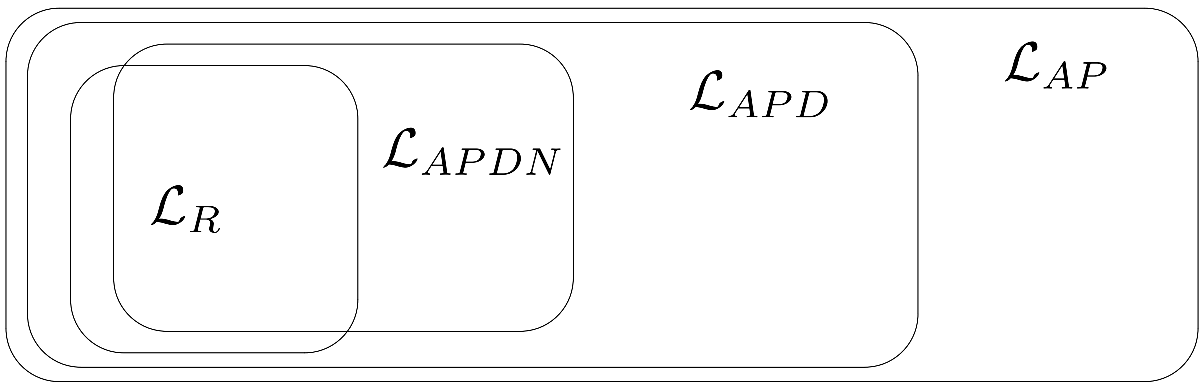
Existeixen llenguatges incontextuals que no són acceptats per autòmats amb pila deterministes.



$$L = \{ww^{-1} | w \in \{0,1\}^\star\}$$

Llenguatges acceptats per AP

Podem classificar els diferents tipus de llenguatges que accepten els autòmats amb pila.



Autòmat equivalent a una gramàtica

Donada una gramàtica en FNG, es pot construir un autòmat amb pila equivalent.

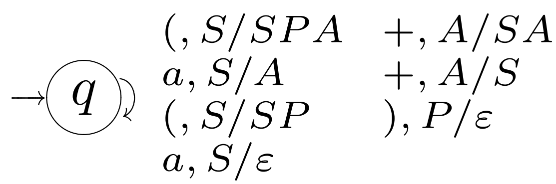
Siga $G = \langle V_T, V_N, S, P \rangle$, amb produccions de la forma $A \rightarrow aB_1 \cdots B_m$.

L'autòmat equivalent conté transicions de la forma $(q, B_1 \cdots B_m) \in \delta(q, a, A)$.

Cada derivació de la forma:
$x_1 \cdots x_k A X_1 \cdots X_\ell \quad \Rightarrow \quad x_1 \cdots x_k a B_1 \cdots B_m \;\cdot\; X_1 \cdots A_\ell$
es correspon amb el canvi de configuracions:
$(q, a\alpha, AX_1 \cdots X_\ell) \longmapsto (q, \alpha, B_1 \cdots B_m \cdot X_1 \cdots A_\ell)$

Exemple

$$S \to S + S | (S) | a$$



Gramàtica equivalent a un autòmat

Donat un autòmat amb pila $A = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$ definim $[pXq]$ com el conjunt de cadenes que fa passar A de p a q i consumeix X del cim de la pila.

$$[pXq] = \{w \in \Sigma^\star | (p, w, X\gamma) \underset{A}{\overset{\star}{\vdash}} (q, \gamma),\ \forall \gamma \in \Gamma^\star\}$$

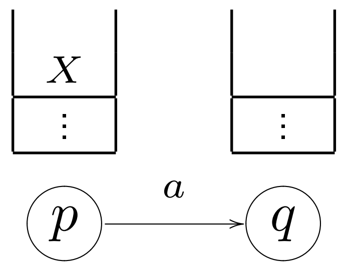
Es compleix que

$$L_N(A) = \bigcup_{p \in Q} [q_0 Z_0 p]$$

[Gramàtica equivalent a un autòmat]

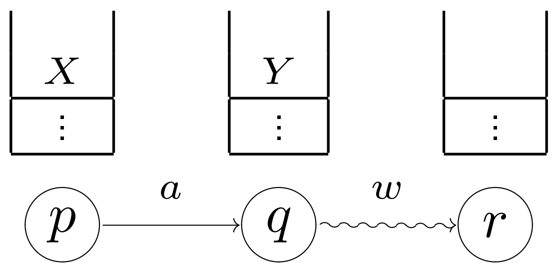
Els conjunts $[pXq]$ es poden definir inductivament a partir de les transicions de A.

$$\boxed{\text{si } (q, \varepsilon) \in \delta(p, a, X) \text{ aleshores } a \in [pXq]}$$



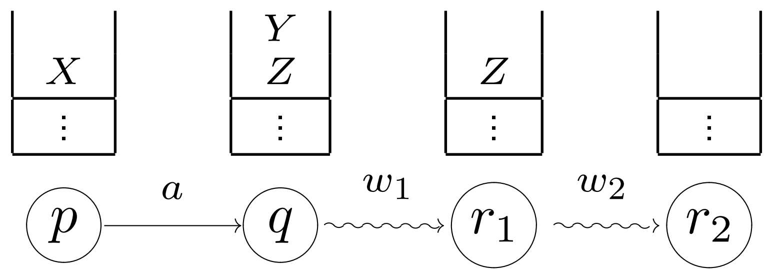
[Gramàtica equivalent a un autòmat]

$$\boxed{\text{si } (q, Y) \in \delta(p, a, X) \text{ aleshores } \{a\} \cdot [qYr] \subseteq [pXr]}$$



[Gramàtica equivalent a un autòmat]

si $(q, YZ) \in \delta(p, a, X)$ aleshores $\{a\} \cdot [qYr_1] \cdot [r_1Zr_2] \subseteq [pXr_2]$



Regla general:

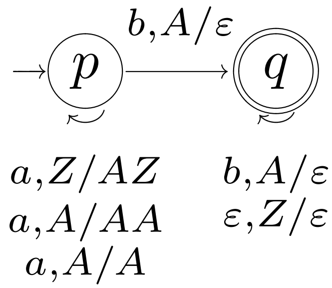
$$\boxed{\text{si } (q, Y_1 \cdots Y_k) \in \delta(p, a, X) \text{ aleshores } \{a\} \cdot [qY_1r_1] \cdot [r_1Y_2r_2] \cdots [r_{k-1}Y_kr_k] \subseteq [pXr_k]}$$

$$\boxed{L_N(A) = \bigcup_{p \in Q} [q_0Z_0p]}$$

Aquestes dues regles es poden convertir directament en produccions:

$$S \longrightarrow [q_0Z_0q_0] | [q_0Z_0q_1] | \ldots | [q_0Z_0q_n]$$

$$[pXr_k] \longrightarrow a[qY_1r_1][r_1Y_2r_2] \cdots [r_{k-1}Y_kr_k] \quad \forall r_1, \ldots r_k \in Q$$


$b,A/\varepsilon$
p
q
$a,Z/AZ$
$a,A/AA$
$a,A/A$
$b,A/\varepsilon$
$\varepsilon,Z/\varepsilon$

Propietats dels incontextuals

- Condició necessària per a la incontextualitat.
- Propietats de clausura.
- Propietats de decisió (anàlisi).

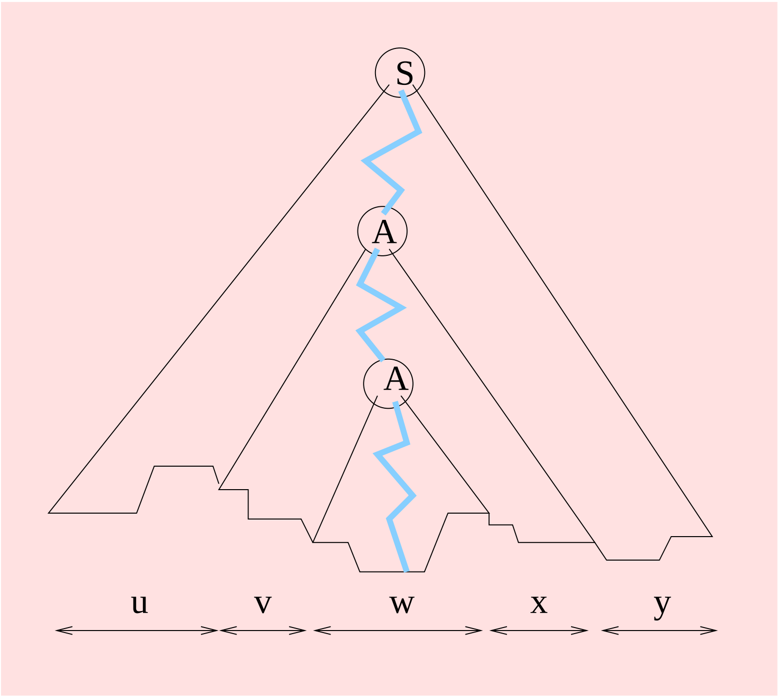
Lema del bombament per a incontextuals

Si un llenguatge és incontextual, cadenes suficientment llargues han de contindre certes subcadenes (amb alguna característica) que es puguen "bombar" d'alguna manera.

??

[Lema del bombament per a incontextuals]

Imaginem una gramàtica en FNC i una cadena suficientment llarga del llenguatge que genera.



Lema del bombament. Enunciat

Si $L \in \mathcal{L}_I$ **aleshores** $\exists n \in \mathbb{N}$ de manera que $\forall z \in L$, $|z| \geq n$ es compleix que

- $\exists z = uvwxy$ (factorització de z) : $|vwx| \leq n \wedge |vx| \geq 1$
- $uv^iwx^iy \in L$, $\forall i \geq 0$

Propietats de clausura

- Unió
- Concatenació
- Clausura de Kleen
- Substitucions (i homomorfismes)
- Homomorfismes inversos
- Intersecció **amb regulars**
- Quocient **amb regulars**

Propietats de decisió

Es poden comprovar metòdicament les següents propietats d'una gramàtica incontextual:

- $L(G) = \emptyset$?
- $|L(G)| < \infty$?
- $x \in L(G)$? (eficientment?)

El problema de l'anàlisi (parsing)

Donada una cadena x i una gramàtica G cal:

- comprovar si $x \in L(G)$
- construir l'arbre de derivació de x en G

Existeix un algorisme de programació dinàmica que resol el problema en temps $O(n^3)$ i espai $O(n^2)$.

Anàlisi eficient

Hi ha dues maneres de construir directament l'arbre:

- Anàlisi descendent
- Anàlisi ascendent

El problema és l'indeterminisme.

que es pot paliar si es restringeixen el tipus de gramàtiques (que és el que es fa en la pràctica).