Gramáticas LR

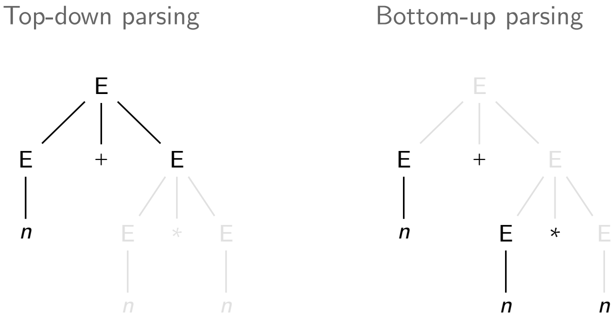
Clase 28

IIC 2223

Prof. Cristian Riveros

Bottom-up parsing (recordatorio)

$$E \quad \rightarrow \quad E + E \mid E * E \mid n$$



Prefijos viables, reducibles y handles (recordatorio)

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una gramática y $\mathcal{G}'$ su gramática aumentada.

Definiciones

- $\alpha \in (V \cup \Sigma)^*$ es un **prefijo viable** de $\mathcal{G}$ ssi
  existe una derivación $S' \underset{\text{rm}}{\overset{\star}{\Rightarrow}} \alpha\beta w$ tal que $\beta \in (V \cup \Sigma)^*$ y $w \in \Sigma^*$.
- $\alpha \cdot \beta \in (V \cup \Sigma)^*$ es **reducible** a $\alpha \cdot X$ ssi
  existe una derivación $S' \underset{\text{rm}}{\overset{\star}{\Rightarrow}} \alpha X w \underset{\text{rm}}{\Rightarrow} \alpha\beta w$ con $w \in \Sigma^*$.

  En cuyo caso, decimos que $X \to \beta$ es un **handle** de $\alpha\beta$.
- $\alpha \cdot \beta \in (V \cup \Sigma)^*$ es un **prefijo reducible** ssi
  $\alpha \cdot \beta$ es un prefijo viable y existe X tal que $\alpha \cdot \beta$ es reducible a $\alpha \cdot X$.

Prefijos viables y prefijos reducibles
representan **configuraciones válidas** del apilador bottom-up.

Autómata característico (recordatorio)

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una gramática libre de contexto cualquiera.

Definición

El **autómata característico** de $\mathcal{G}$ es un ϵ-NFA:

$$\text{char}[\mathcal{G}] \;=\; (\underbrace{\text{Items}_{\mathcal{G}} \cup \left\{[S' \to .S], [S' \to S.]\right\}}_{Q_0}, V \cup \Sigma, \Delta_0, \underbrace{[S' \to .S]}_{I_0}, \underbrace{\{[X \to \alpha.]\}}_{F_0})$$

Dos tipos de transiciones en Δ_0:

Bajar: $[X \to \alpha.Y\beta] \xrightarrow{\epsilon} [Y \to .\gamma]$ para cada $X \to \alpha Y\beta \in Q_0$ y $Y \to \gamma \in P$

Avanzar: $[X \to \alpha.Y\beta] \xrightarrow{Y} [X \to \alpha Y.\beta]$ $Y \in V \cup \Sigma$

char$[\mathcal{G}]$ **navega** por un árbol de derivación.

Autómata característico y prefijos reducibles (recordatorio)

Teorema

1. $[S' \to .S] \xrightarrow{\gamma} [X \to \alpha.\beta]$ si, y solo si, γ es un **prefijo viable** de $\mathcal{G}$.
2. $\mathcal{L}(\text{char}[\mathcal{G}]) = \{\gamma \mid \gamma \text{ es un prefijo reducible de } \mathcal{G}\}$

char$[\mathcal{G}]$ sirve para **identificar** las configuraciones viables del bottom-up PDA

LR parser (recordatorio)

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una gramática libre de contexto cualquiera.

Definición

El **LR-autómata apilador** de $\mathcal{G}$ (LR-PDA) es un PDA alternativo:

$$\mathrm{LR}[\mathcal{G}] \;=\; (\underbrace{Q_0^{\mathrm{det}} \cup \{f\}}_{Q_1}, \Sigma, \Delta_1, \underbrace{I_0^{\mathrm{det}}}_{I_1}, \underbrace{\{f\}}_{F_1})$$

Tres tipos de transiciones en $\Delta_1 \subseteq Q_1^+ \times \Sigma \times Q_1^*$ para $C, C_1, \ldots, C_n \in Q_0^{\mathrm{det}}$:

Shift: $C \xrightarrow{a} C\, \Delta_0^{\mathrm{det}}(C, a)$ si $\Delta_0^{\mathrm{det}}(C, a) \neq \varnothing$

Reduce: $C\, C_1 \ldots C_n \xrightarrow{\epsilon} C\, \Delta_0^{\mathrm{det}}(C, X)$ si $[X \to \alpha.] \in C_n$ y $n = |\alpha|$

Término: $I_0^{\mathrm{det}} C \xrightarrow{\epsilon} f$ si $[S' \to S.] \in q$

¿qué representa el stack del LR-PDA?

Definición LR-PDA

$$\mathrm{LR}[\mathcal{G}] \;=\; (\underbrace{Q_0^{\mathrm{det}} \cup \{f\}}_{Q_1}, \Sigma, \Delta_1, \underbrace{I_0^{\mathrm{det}}}_{I_1}, \underbrace{\{\,f\,\}}_{F_1})$$

Proposición

Si $(I_0^{\mathrm{det}}, uv) \vdash^*_{\mathrm{LR}[\mathcal{G}]} (I_0^{\mathrm{det}} C_1 \ldots C_n, v)$, entonces existe $\gamma_1 \ldots \gamma_n \in (V \cup \Sigma)^*$:

$$I_0^{\mathrm{det}} \xrightarrow{\gamma_1} C_1 \xrightarrow{\gamma_2} \ldots \xrightarrow{\gamma_n} C_n$$

es una **ejecución** de $\mathrm{char}[\mathcal{G}]^{\mathrm{det}}$ sobre $\gamma_1 \ldots \gamma_n$.

El stack $I_0^{\mathrm{det}} C_1 \ldots C_n$ de una configuración cualquiera corresponde a una **ejecución** de la determinización $\mathrm{char}[\mathcal{G}]$.

Relación entre LR-PDA y bottom-up PDA

Sea $\mathcal{G}$ una gramática libre de contexto.

Teorema

Sea $\gamma = \gamma_1 \ldots \gamma_n \in (V \cup \Sigma)^*$ y
$I_0^{\text{det}} \xrightarrow{\gamma_1} C_1 \xrightarrow{\gamma_2} \ldots \xrightarrow{\gamma_n} C_n$ la ejecución de $\text{det}[\mathcal{G}]$ sobre γ. Para todo $u, v \in \Sigma^*$:

- $(\$, uv) \vdash^*_{\mathcal{P}_\uparrow} (\$\gamma, v) \vdash^*_{\mathcal{P}_\uparrow} (\$S', \epsilon)$ si, y solo si,
- $C_n \neq \varnothing$ y $(I_0^{\text{det}}, uv) \vdash^*_{\text{LR}[\mathcal{G}]} (I_0^{\text{det}} C_1 \ldots C_n, v) \vdash^*_{\text{LR}[\mathcal{G}]} (f, \epsilon)$.

En particular, $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\text{LR}[\mathcal{G}])$.

En otras palabras, $\text{LR}[\mathcal{G}]$ simula a $\mathcal{P}_\uparrow$ llevando solo **configuraciones viables**.

(NO haremos esta demostración en clases)

Outline

¿cómo usar LR-Parser?

Gramáticas LR

Algunas conclusiones sobre LR

Outline

¿cómo usar LR-Parser?

Gramáticas LR

Algunas conclusiones sobre LR

LR parser y su uso

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una gramática libre de contexto cualquiera.

Definición

El **LR-autómata apilador** de $\mathcal{G}$ (LR-PDA) es un PDA alternativo:

$$\mathrm{LR}[\mathcal{G}] \;=\; (\underbrace{Q_0^{\mathrm{det}} \cup \{f\}}_{Q_1}, \Sigma, \Delta_1, \underbrace{I_0^{\mathrm{det}}}_{I_1}, \underbrace{\{f\}}_{F_1})$$

Tres tipos de transiciones en $\Delta_1 \subseteq Q_1^+ \times \Sigma \times Q_1^*$ para $C, C_1, \ldots, C_n \in Q_0^{\mathrm{det}}$:

Shift:	$C \xrightarrow{a} C \cdot \Delta_0^{\mathrm{det}}(C, a)$	si $\Delta_0^{\mathrm{det}}(C, a) \neq \varnothing$
Reduce:	$C\, C_1 \ldots C_n \xrightarrow{\epsilon} C \cdot \Delta_0^{\mathrm{det}}(C, X)$	si $[X \to \alpha.] \in C_n$ y $n = \lvert\alpha\rvert$
Término:	$I_0^{\mathrm{det}} C \xrightarrow{\epsilon} f$	si $[S' \to S.] \in q$

¿dónde esta el **no-determinismo** en el LR-PDA?

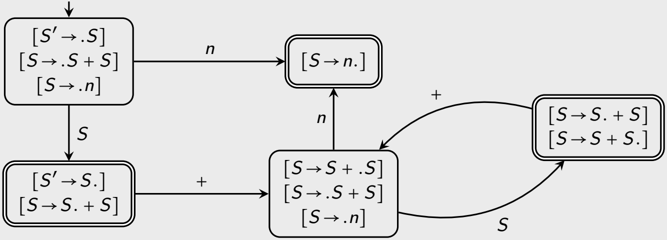
Determinación de conflictos en LR-PDA

Definición

Para un estado $C \in Q_0^{\text{det}}$ decimos que:

1. C tiene un conflicto **Shift-Reduce** si existen $[X \to \alpha.a\beta]$ y $[Y \to \gamma.]$ en C, simultáneamente.
2. C tiene un conflicto **Reduce-Reduce** si existen $[Y_1 \to \gamma_1.]$ y $[Y_2 \to \gamma_2.]$ distintos en C.

¿qué conflicto tiene nuestra determinización de char$[\mathcal{G}]$?



Determinación de conflictos en LR-PDA

Definición

Para un estado $C \in Q_0^{\text{det}}$ decimos que:

1. C tiene un conflicto **Shift-Reduce** si existen $[X \to \alpha.a\beta]$ y $[Y \to \gamma.]$ en C, simultáneamente.
2. C tiene un conflicto **Reduce-Reduce** si existen $[Y_1 \to \gamma_1.]$ y $[Y_2 \to \gamma_2.]$ distintos en C.

Decimos que C es **libre de conflictos** si
NO tiene un conflicto Shift-Reduce o Reduce-Reduce.

Notar que...

Si C es libre de conflictos para todo $C \in Q_0^{\text{det}}$,
entonces LR$[\mathcal{G}]$ es un autómata apilador **determinista**.

Determinación de conflictos en LR-PDA

Notar que...

Si C es libre de conflictos para todo $C \in Q_0^{\text{det}}$,
entonces $\text{LR}[\mathcal{G}]$ es un autómata apilador **determinista**.

Shift:	$C \xrightarrow{a} C \cdot \Delta_0^{\text{det}}(C, a)$	si $\Delta_0^{\text{det}}(C, a) \neq \varnothing$
Reduce:	$C\, C_1 \dots C_n \xrightarrow{\epsilon} C \cdot \Delta_0^{\text{det}}(C, X)$	si $[X \to \alpha.] \in C_n$ y $n = \lvert\alpha\rvert$
Término:	$I_0^{\text{det}}\, C \xrightarrow{\epsilon} f$	si $[S' \to S.] \in q$

Por lo tanto, podemos usar $\text{LR}[\mathcal{G}]$ para hacer **parsing** y
encontrar una **derivación por la derecha (invertida)** de $\mathcal{G}$ para cada input.

Outline

Definición gramáticas LR(k)

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una gramática libre de contexto y $k \geq 0$.

Intuición

Intuitivamente, $\mathcal{G}$ es una gramática LR(k) si para toda derivación:

$$v \underset{\text{rm}}{\Leftarrow} \alpha_n \underset{\text{rm}}{\Leftarrow} \alpha_{n-1} \underset{\text{rm}}{\Leftarrow} \cdots \underset{\text{rm}}{\Leftarrow} \alpha_1 \underset{\text{rm}}{\Leftarrow} S'$$

podemos para cada α_i:

1. localizar el **handle** y
2. determinar la regla en P a ocupar

mirando a lo más k símbolos "**a la derecha del handle**".

Para cada $\alpha_i = \gamma\beta w$, la regla $X \to \beta$ tal que $\alpha_{i-1} = \gamma X w$ esta **únicamente determinado** por $\gamma\beta$ y $w|_k$.

Definición gramáticas LR(k)

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una gramática libre de contexto y $k \geq 0$.

Definición

Decimos que $\mathcal{G}$ es una gramática LR(k) si para todas derivaciones:

- $\alpha\beta v_1 \underset{\text{rm}}{\Leftarrow} \alpha X v_1 \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S'$
- $\alpha\beta v_2 \underset{\text{rm}}{\Leftarrow} \alpha' Y v_2' \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S'$ y
- $v_1|_k = v_2|_k$

entonces se cumple que $\alpha = \alpha'$, $X = Y$ y $v_2 = v_2'$. $\qquad (\alpha X v_2 = \alpha' Y v_2')$

Notar que la elección de $X \to \beta$ depende de $\alpha \cdot \beta \cdot v_1|_k$.

Ejemplos y contra-ejemplos de gramáticas LR(k)

Definición LR(k)

- $\alpha\beta v_1 \underset{\text{rm}}{\Leftarrow} \alpha X v_1 \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S'$
- $\alpha\beta v_2 \underset{\text{rm}}{\Leftarrow} \alpha' Y v_2' \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S'$
- $v_1|_k = v_2|_k$

$\Bigg\}$ entonces $\alpha X v_2 = \alpha' Y v_2'$.

Ejemplo 1

$$\mathcal{G}_1 : \quad S \to aXc \qquad X \to Xbb \mid b \qquad \mathcal{L}(\mathcal{G}_1) = \{ab^{2n+1}c \mid n \geq 0\}$$

Todas las **posibles** derivaciones por la derecha de $\mathcal{G}_1$ son:

$$S \qquad aXc \qquad aXb^{2n}c \qquad abb^{2n}c$$

- $\underbrace{a}_{\alpha}\underbrace{Xbb}_{\beta}\underbrace{b^{2n}c}_{v_1} \underset{\text{rm}}{\Leftarrow} \underbrace{a}_{\alpha} X \underbrace{b^{2n}c}_{v_1} \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S'$
- $\underbrace{a}_{\alpha}\underbrace{Xbb}_{\beta}\underbrace{b^{2m}c}_{v_2} \underset{\text{rm}}{\Leftarrow} \underbrace{a}_{\alpha'} X \underbrace{b^{2m}c}_{v_2'} \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S'$

$\Bigg\}$ entonces $\alpha X v_2 = \alpha' X v_2'$.

Ejemplos y contra-ejemplos de gramáticas LR(k)

Definición LR(k)

$$\left.\begin{array}{l} \blacksquare\ \alpha\beta v_1 \underset{\text{rm}}{\Leftarrow} \alpha X v_1 \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S' \\ \blacksquare\ \alpha\beta v_2 \underset{\text{rm}}{\Leftarrow} \alpha' Y v_2' \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S' \\ \blacksquare\ v_1|_k = v_2|_k \end{array}\right\} \quad \text{entonces} \quad \alpha X v_2 = \alpha' Y v_2'.$$

Ejemplo 1

$$\mathcal{G}_1: \quad S \to aXc \qquad X \to Xbb \mid b \qquad \mathcal{L}(\mathcal{G}_1) = \{ab^{2n+1}c \mid n \geq 0\}$$

Todas las **posibles** derivaciones por la derecha de $\mathcal{G}_1$ son: ...

$$\left.\begin{array}{l} \blacksquare\ \underbrace{a}_{\alpha}\underbrace{Xbb}_{\beta}\underbrace{b^{2n}c}_{v_1} \underset{\text{rm}}{\Leftarrow} \underbrace{a}_{\alpha} X \underbrace{b^{2n}c}_{v_1} \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S' \\ \blacksquare\ \underbrace{a}_{\alpha}\underbrace{Xbb}_{\beta}\underbrace{b^{2m}c}_{v_2} \underset{\text{rm}}{\Leftarrow} \underbrace{a}_{\alpha'} X \underbrace{b^{2m}c}_{v_2'} \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S' \end{array}\right\} \quad \text{entonces} \quad \alpha X v_2 = \alpha' X v_2'.$$

Como la elección de β no depende v_1 o v_2, entonces $\mathcal{G}_1$ es LR(0).

Ejemplos y contra-ejemplos de gramáticas LR(k)

Definición LR(k)

- $\alpha\beta v_1 \underset{\text{rm}}{\Leftarrow} \alpha X v_1 \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S'$
- $\alpha\beta v_2 \underset{\text{rm}}{\Leftarrow} \alpha' Y v_2' \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S'$
- $v_1|_k = v_2|_k$

entonces $\alpha X v_2 = \alpha' Y v_2'$.

Ejemplo 2

$$\mathcal{G}_2 : \quad S \to aXc \qquad X \to bbX \mid b \qquad \mathcal{L}(\mathcal{G}_2) = \{ab^{2n+1}c \mid n \geq 0\}$$

Todas las **posibles** derivaciones por la derecha de $\mathcal{G}_2$ son:

$$S \qquad aXc \qquad ab^{2n}Xc \qquad ab^{2n}bc$$

- $\underbrace{ab^{2n}}_{\alpha}\underbrace{b}_{\beta} v_1 \underset{\text{rm}}{\Leftarrow} \underbrace{ab^{2n}}_{\alpha} X v_1 \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S'$
- $\underbrace{ab^{2n}}_{\alpha}\underbrace{b}_{\beta} v_2 \underset{\text{rm}}{\Leftarrow} \underbrace{ab^{2m}}_{\alpha'} X v_2' \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S'$

$$\underbrace{ab^{2n}}_{\alpha} X \underbrace{c}_{v_2} = \underbrace{ab^{2m}}_{\alpha'} X \underbrace{c}_{v_2'}$$

Ejemplos y contra-ejemplos de gramáticas LR(k)

Definición LR(k)

- $\alpha\beta v_1 \underset{rm}{\Leftarrow} \alpha X v_1 \overset{\star}{\underset{rm}{\Leftarrow}} S'$
- $\alpha\beta v_2 \underset{rm}{\Leftarrow} \alpha' Y v_2' \overset{\star}{\underset{rm}{\Leftarrow}} S'$
- $v_1|_k = v_2|_k$

$\Bigg\}$ entonces $\alpha X v_2 = \alpha' Y v_2'$.

Ejemplo 2

$$\mathcal{G}_2: \quad S \to aXc \qquad X \to bbX \mid b \qquad \mathcal{L}(\mathcal{G}_2) = \{ab^{2n+1}c \mid n \geq 0\}$$

Todas las **posibles** derivaciones por la derecha de $\mathcal{G}_2$ son: ...

- $\underbrace{ab^{2n}}_{\alpha}\underbrace{b}_{\beta}v_1 \underset{rm}{\Leftarrow} \underbrace{ab^{2n}}_{\alpha}Xv_1 \overset{\star}{\underset{rm}{\Leftarrow}} S'$
- $\underbrace{ab^{2n}}_{\alpha}\underbrace{b}_{\beta}v_2 \underset{rm}{\Leftarrow} \underbrace{ab^{2m}}_{\alpha'}Xv_2' \overset{\star}{\underset{rm}{\Leftarrow}} S'$

$\Bigg\}$ $\underbrace{ab^{2n}}_{\alpha}X\underbrace{c}_{v_2} = \underbrace{ab^{2m}}_{\alpha'}X\underbrace{c}_{v_2'}$

Como la elección de β depende si $v_1|_1 = v_2|_1 = c$, entonces $\mathcal{G}_2$ es LR(1).

Ejemplos y contra-ejemplos de gramáticas LR(k)

Definición LR(k)

$$\left.\begin{array}{l} \blacksquare\ \alpha\beta v_1 \underset{\text{rm}}{\Leftarrow} \alpha X v_1 \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S' \\ \blacksquare\ \alpha\beta v_2 \underset{\text{rm}}{\Leftarrow} \alpha' Y v_2' \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S' \\ \blacksquare\ v_1|_k = v_2|_k \end{array}\right\} \text{ entonces } \quad \alpha X v_2 = \alpha' Y v_2'.$$

Ejemplo 3

$\mathcal{G}_3: \quad S \to aXc \qquad X \to bXb \mid b \qquad \mathcal{L}(\mathcal{G}_3) = \{ab^{2n+1}c \mid n \geq 0\}$

Para cualquier $k \geq 0$ y $n > k$ tenemos:

$$\left.\begin{array}{l} \blacksquare\ \underbrace{ab^n}_{\alpha}\,\underbrace{b}_{\beta}\,\underbrace{b^n c}_{v_1} \underset{\text{rm}}{\Leftarrow} \underbrace{ab^n}_{\alpha} X \underbrace{b^n c}_{v_1} \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S' \\ \blacksquare\ \underbrace{ab^n}_{\alpha}\,\underbrace{b}_{\beta}\,\underbrace{b^{n+2} c}_{v_2} \underset{\text{rm}}{\Leftarrow} \underbrace{ab^{n+1}}_{\alpha'} X \underbrace{b^{n+1} c}_{v_2'} \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S' \\ \blacksquare\ v_1|_k = v_2|_k \end{array}\right\} \text{ pero } \quad \alpha X v_2 \neq \alpha' X v_2'$$

Como esto se cumple para todo k, entonces $\mathcal{G}_3$ **NO** es LR(k) para todo k.

Ejemplos y contra-ejemplos de gramáticas LR(k)

Definición LR(k)

$$\left.\begin{array}{l}\blacksquare\ \alpha\beta v_1 \underset{\text{rm}}{\Leftarrow} \alpha X v_1 \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S' \\ \blacksquare\ \alpha\beta v_2 \underset{\text{rm}}{\Leftarrow} \alpha' Y v_2' \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S' \\ \blacksquare\ v_1|_k = v_2|_k \end{array}\right\} \quad \text{entonces} \quad \alpha X v_2 = \alpha' Y v_2'.$$

Ejemplo 4

$$\mathcal{G}_4: \quad \begin{array}{lcl} S & \to & X \mid Y \\ X & \to & aXb \mid 0 \\ Y & \to & aYbb \mid 1 \end{array} \qquad \mathcal{L}(\mathcal{G}_4) = \{a^n 0 b^n \mid n \geq 0\} \cup \{a^n 1 b^{2n} \mid n \geq 0\}$$

Todas las **posibles** derivaciones por la derecha de $\mathcal{G}_4$ son:

$$S \quad X \quad a^n aXbb^n \quad a^n a0bb^n \quad Y \quad a^n aYbbb^{2n} \quad a^n a1bbb^{2n}$$

Ejemplos y contra-ejemplos de gramáticas LR(k)

Definición LR(k)

$$\left.\begin{array}{l} \blacksquare\ \alpha\beta v_1 \underset{\text{rm}}{\Leftarrow} \alpha X v_1 \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S' \\ \blacksquare\ \alpha\beta v_2 \underset{\text{rm}}{\Leftarrow} \alpha' Y v_2' \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S' \\ \blacksquare\ v_1|_k = v_2|_k \end{array}\right\} \text{ entonces } \alpha X v_2 = \alpha' Y v_2'.$$

Ejemplo 4

$$\mathcal{G}_4: \begin{array}{lcl} S & \to & X \mid Y \\ X & \to & aXb \mid 0 \\ Y & \to & aYbb \mid 1 \end{array} \qquad \mathcal{L}(\mathcal{G}_4) = \{a^n 0 b^n \mid n \geq 0\} \cup \{a^n 1 b^{2n} \mid n \geq 0\}$$

Todas las **posibles** derivaciones por la derecha de $\mathcal{G}_4$ son:

$$S \quad \underbrace{X}_{\beta} \quad a^n\underbrace{aXb}_{\beta}b^n \quad a^n\underbrace{a0b}_{\beta}b^n \quad \underbrace{Y}_{\beta} \quad a^n\underbrace{aYbb}_{\beta}b^{2n} \quad a^n\underbrace{a1bb}_{\beta}b^{2n}$$

Como la elección de β no depende del sufijo, entonces $\mathcal{G}_4$ es LR(0).

Ejemplos y contra-ejemplos de gramáticas LR(k)

Definición LR(k)

- $\alpha\beta v_1 \underset{\text{rm}}{\Leftarrow} \alpha X v_1 \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S'$
- $\alpha\beta v_2 \underset{\text{rm}}{\Leftarrow} \alpha' Y v_2' \overset{\star}{\underset{\text{rm}}{\Leftarrow}} S'$
- $v_1|_k = v_2|_k$

$\Bigg\}$ entonces $\alpha X v_2 = \alpha' Y v_2'$.

Ejemplo 4

$$\begin{array}{rlcl} \mathcal{G}_4: & S & \to & X \mid Y \\ & X & \to & aXb \mid 0 \\ & Y & \to & aYbb \mid 1 \end{array} \qquad \mathcal{L}(\mathcal{G}_4) = \{a^n 0 b^n \mid n \geq 0\} \cup \{a^n 1 b^{2n} \mid n \geq 0\}$$

Todas las **posibles** derivaciones por la derecha de $\mathcal{G}_4$ son:

$$S \quad X \quad a^n aXbb^n \quad a^n a0bb^n \quad Y \quad a^n aYbbb^{2n} \quad a^n a1bbb^{2n}$$

Como la elección de β no depende del sufijo, entonces $\mathcal{G}_4$ es LR(0).

¿es $\mathcal{G}_4$ una gramática LL(k)?

Outline

Sobre gramáticas LR(k)

Es posible demostrar que ...

- Toda gramática LL(k) es una gramática LR(k).
- Existe un lenguaje libre de contexto L, tal que para toda gramática $\mathcal{G}$ con $\mathcal{L}(\mathcal{G}) = L$, se cumple que $\mathcal{G}$ **NO** es LR(k) para todo $k \geq 0$.

¿a qué corresponden las gramáticas LR(0)?

Sobre gramáticas LR(0) y estados libres de conflicto

Teorema

$\mathcal{G}$ es una gramática LR(0) si, y solo si, la determinización del autómata característico de $\mathcal{G}$ tiene **solo estados libres de conflicto**.

Demostración (sketch)

($\Rightarrow$) Por **contrapositivo**: suponemos que $\mathcal{G}$ tiene un estado con conflicto y llegamos a que $\mathcal{G}$ no puede ser LR(0).

($\Leftarrow$) **Directo**: suponemos que $\mathcal{G}$ tiene solo estados libres de conflicto y demostramos que $\mathcal{G}$ es LR(0).

Sobre gramáticas LR(0) y estados libres de conflicto

Teorema

$\mathcal{G}$ es una gramática LR(0) si, y solo si, la determinización del autómata característico de $\mathcal{G}$ tiene **solo estados libres de conflicto**.

Para toda gramática LR(0) existe
un autómata apilador determinista que define el mismo lenguaje.

Es posible demostrar que ...

Para todo autómata apilador determinista existe una gramática LR(0) que define el mismo lenguaje y, por lo tanto, son equivalentes.

FIN . . .   :)