Teoría de la Computación

Clase 16: Lenguajes independientes del contexto y autómatas de pila

Mauro Artigiani

17 septiembre 2021

Universidad del Rosario, Bogotá

Equivalencia entre CFL y PDA

La equivalencia

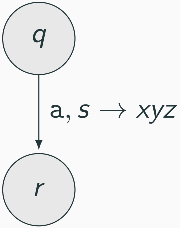
En esta y en las próxima clase demostraremos que `PDAs` y `CFLs` son equivalentes.

Teorema

Un lenguaje es independiente del contexto si y solo si es reconocido por un autómata de pila.

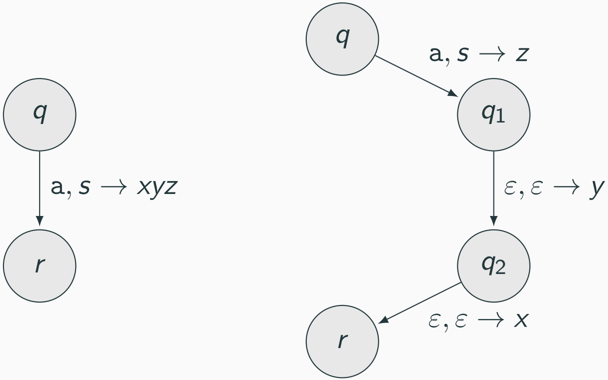
Una construcción intermedia

Antes de ver la demostración de la implicación directa, vamos a introducir una manera para escribir más de un símbolo en la pila al mismo tiempo.



Una construcción intermedia

Antes de ver la demostración de la implicación directa, vamos a introducir una manera para escribir más de un símbolo en la pila al mismo tiempo.



La implicación directa

Si L es un `CFL`, existe, por definición, una `CFG` G que lo describe. Queremos construir un `PDA` P que reconozca L a partir de las reglas de G. La idea es la siguiente.

La implicación directa

Si L es un CFL, existe, por definición, una CFG G que lo describe. Queremos construir un PDA P que reconozca L a partir de las reglas de G. La idea es la siguiente.

Un palabra w pertenece a L si existe una secuencia de derivaciones que empieza con la variable inicial S de L y al final produce w. Podemos entonces construir un autómata que empiece con la variable S en la pila y haga varias substituciones hasta llegar a tener solo terminales.

La implicación directa

Si L es un CFL, existe, por definición, una CFG G que lo describe. Queremos construir un PDA P que reconozca L a partir de las reglas de G. La idea es la siguiente.

Un palabra w pertenece a L si existe una secuencia de derivaciones que empieza con la variable inicial S de L y al final produce w. Podemos entonces construir un autómata que empiece con la variable S en la pila y haga varias substituciones hasta llegar a tener solo terminales. En este momento controlamos si la palabra que hemos obtenido es w o no y aceptamos o rechazamos la palabra.

La implicación directa

Hay dos problemas con nuestra idea.

1. En cada etapa hay varias reglas posibles para substituir una variable dada. En este momento nos ayuda el no determinismo: el autómata se dividirá en varios procesos, cada uno de ellos siguiendo una posibilidad.

La implicación directa

Hay dos problemas con nuestra idea.

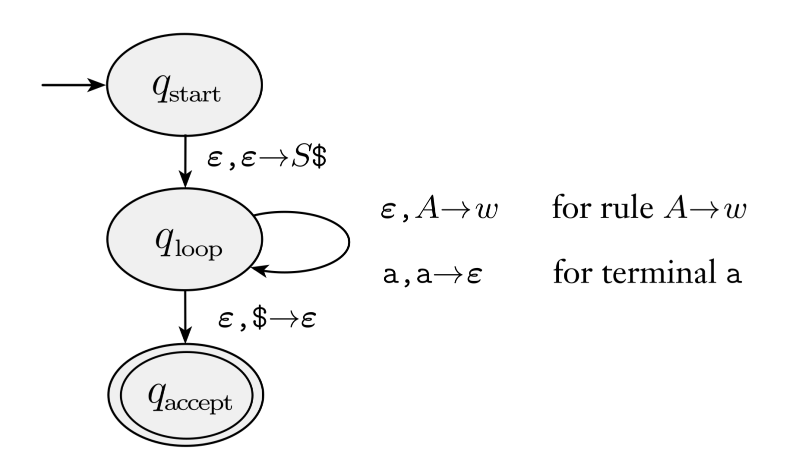
1. En cada etapa hay varias reglas posibles para substituir una variable dada. En este momento nos ayuda el no determinismo: el autómata se dividirá en varios procesos, cada uno de ellos siguiendo una posibilidad.
2. Un problema más grande es qué hacer con las *cadenas intermedias* que obtenemos paso a paso que derivamos. Se podría escribirlas en la pila, pero en este caso, dado que el autómata solo tiene acceso al tope de la pila, tendríamos un problema si en el tope estuviera un terminal.

La implicación directa

Hay dos problemas con nuestra idea.

1. En cada etapa hay varias reglas posibles para substituir una variable dada. En este momento nos ayuda el no determinismo: el autómata se dividirá en varios procesos, cada uno de ellos siguiendo una posibilidad.
2. Un problema más grande es qué hacer con las *cadenas intermedias* que obtenemos paso a paso que derivamos. Se podría escribirlas en la pila, pero en este caso, dado que el autómata solo tiene acceso al tope de la pila, tendríamos un problema si en el tope estuviera un terminal. Para solucionar este problema, vamos a comparar cada terminal con la palabra de input w paso a paso que lo generamos, y escribimos en la pila solo un pedazo de una cadena intermedia, de manera que en el tope siempre esté una variable.

La implicación directa: construcción del autómata



Un ejemplo

Por ejemplo, supongamos de tener la gramática:

$$S \to \mathtt{a}T\mathtt{b} \mid \mathtt{b}$$

$$T \to T\mathtt{a} \mid \varepsilon$$

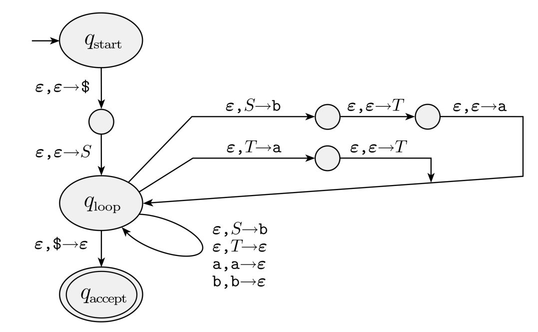
Un ejemplo

Por ejemplo, supongamos de tener la gramática:

$$S \to \mathtt{a}T\mathtt{b} \,|\, \mathtt{b}$$

$$T \to T\mathtt{a} \,|\, \varepsilon$$

El procedimiento descrito antes produce el `PDA`:



Ideas para la implicación inversa

Hacia la implicación inversa

Tenemos un `PDA` P y queremos diseñar una gramática independiente de contexto G que genere todas (y solamente) las palabras aceptadas por P.

Hacia la implicación inversa

Tenemos un `PDA` P y queremos diseñar una gramática independiente de contexto G que genere todas (y solamente) las palabras aceptadas por P.

Para simplificar, hacemos unas modificaciones a P:

Lema 1

Sea P un `PDA`. Luego existe un `PDA` P' *equivalente* a P tales que

1. P' solo tiene un estado de aceptación q_{accept};

Hacia la implicación inversa

Tenemos un PDA P y queremos diseñar una gramática independiente de contexto G que genere todas (y solamente) las palabras aceptadas por P.

Para simplificar, hacemos unas modificaciones a P:

Lema 1

Sea P un PDA. Luego existe un PDA P' *equivalente* a P tales que

1. P' solo tiene un estado de aceptación q_{accept};
2. P' vacía la pila *antes* de aceptar;

Hacia la implicación inversa

Tenemos un `PDA` P y queremos diseñar una gramática independiente de contexto G que genere todas (y solamente) las palabras aceptadas por P.

Para simplificar, hacemos unas modificaciones a P:

Lema 1

Sea P un `PDA`. Luego existe un `PDA` P' *equivalente* a P tales que

1. P' solo tiene un estado de aceptación q_{accept};
2. P' vacía la pila *antes* de aceptar;
3. Cada transición añada un símbolo en la pila (paso de empuje) o borra un símbolo de la pila (paso de extracción), pero no hace los dos al mismo tiempo.

El Lema clave

Lema 2

Dado un `PDA` que satisfaga el Lema 1, se puede construir una `CFG` G que satisfaga lo siguiente. Si p y q son dos estados de P, G contiene una variable A_{pq} y

$$(p, w, \varepsilon) \overset{*}{\vdash} (q, \varepsilon, \varepsilon) \quad \text{sii} \quad A_{pq} \overset{*}{\Rightarrow} w$$

El Lema clave

Lema 2

Dado un PDA que satisfaga el Lema 1, se puede construir una CFG G que satisfaga lo siguiente. Si p y q son dos estados de P, G contiene una variable A_{pq} y

$$(p, w, \varepsilon) \overset{*}{\vdash} (q, \varepsilon, \varepsilon) \quad \text{sii} \quad A_{pq} \overset{*}{\Rightarrow} w$$

Es decir: procesando w el autómata P se mueve desde p, con la pila vacía, hasta q, con la pila vacía, sii G puede derivar w a partir de A_{pq}.

El Lema clave

Lema 2

Dado un PDA que satisfaga el Lema 1, se puede construir una CFG G que satisfaga lo siguiente. Si p y q son dos estados de P, G contiene una variable A_{pq} y

$$(p, w, \varepsilon) \overset{*}{\vdash} (q, \varepsilon, \varepsilon) \quad \text{sii} \quad A_{pq} \overset{*}{\Rightarrow} w$$

Es decir: procesando w el autómata P se mueve desde p, con la pila vacía, hasta q, con la pila vacía, sii G puede derivar w a partir de A_{pq}.

Nótese que las mismas palabras permiten ir desde p a q teniendo la misma pila al comienzo y al final.

Hacia la implicación inversa

¿Cómo se mueve P de un estado p a un estado q, empezando y terminando con la pila vacía?

Hacia la implicación inversa

¿Cómo se mueve P de un estado p a un estado q, empezando y terminando con la pila vacía?

Dado que al comienzo la pila es vacía, y que cada paso o de empuje o de extracción, necesariamente el primer paso tiene que se de empuje. De manera similar, el último tiene que ser de extracción.

Hacia la implicación inversa

¿Cómo se mueve P de un estado p a un estado q, empezando y terminando con la pila vacía?

Dado que al comienzo la pila es vacía, y que cada paso o de empuje o de extracción, necesariamente el primer paso tiene que se de empuje. De manera similar, el último tiene que ser de extracción.

Tenemos dos casos: la pila queda vacía en algún paso computacional o no. Para ambos casos tenemos un tipo de reglas de G.

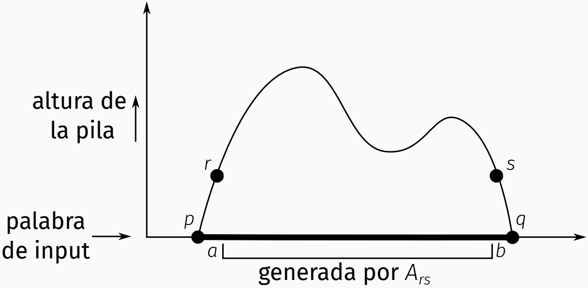
Hacia la implicación inversa

Si la pila queda vacía solo al final:

Hacia la implicación inversa

Si la pila queda vacía solo al final:

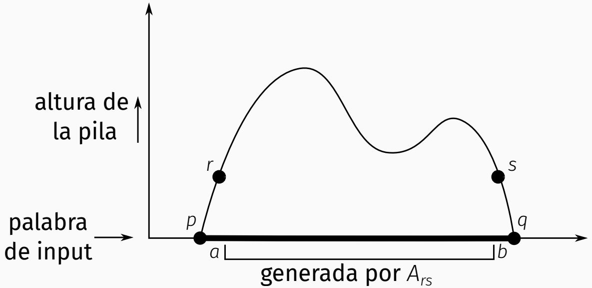
Computación de P correspondiente a la regla $A_{pq} \longrightarrow \mathrm{a} A_{rs} \mathrm{b}$



Hacia la implicación inversa

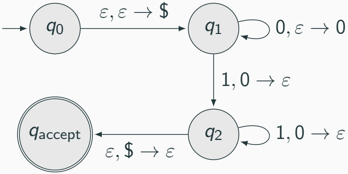
Si la pila queda vacía solo al final:

Computación de P correspondiente a la regla $A_{pq} \longrightarrow \mathrm{a}A_{rs}\mathrm{b}$



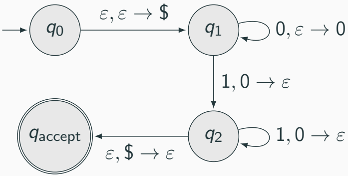
En este caso, definimos la regla $A_{pq} \to \mathtt{a}A_{rs}\mathtt{b}$, donde $\mathtt{a}$ es el primer símbolo leído, $\mathtt{b}$ el último; r es el estado que sigue p y s el estado que precede q.

Ejemplo



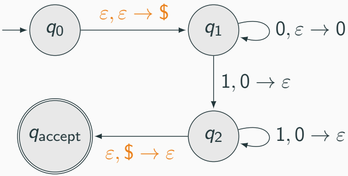
La pila queda vacía solo al final.

Ejemplo



La pila queda vacía solo al final. Como no sabemos esto de antemano, incluimos todas las reglas basadas en las transiciones que incluyen un símbolo en la pila y en las que lo retiran:

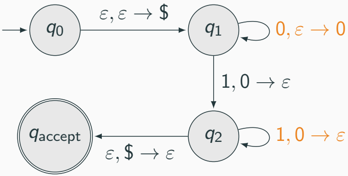
Ejemplo



La pila queda vacía solo al final. Como no sabemos esto de antemano, incluimos todas las reglas basadas en las transiciones que incluyen un símbolo en la pila y en las que lo retiran:

$A_{q_0 q_{\text{accept}}} \to \varepsilon A_{q_1 q_2} \varepsilon$

Ejemplo

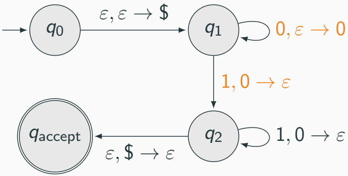


La pila queda vacía solo al final. Como no sabemos esto de antemano, incluimos todas las reglas basadas en las transiciones que incluyen un símbolo en la pila y en las que lo retiran:

$A_{q_0 q_{accept}} \to \varepsilon A_{q_1 q_2} \varepsilon$

$A_{q_1 q_2} \to 0 A_{q_1 q_2} 1$

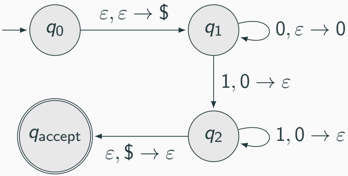
Ejemplo



La pila queda vacía solo al final. Como no sabemos esto de antemano, incluimos todas las reglas basadas en las transiciones que incluyen un símbolo en la pila y en las que lo retiran:

$A_{q_0 q_{\text{accept}}} \to \varepsilon A_{q_1 q_2} \varepsilon$
$A_{q_1 q_2} \to 0 A_{q_1 q_2} 1$
$A_{q_1 q_2} \to 0 A_{q_1 q_1} 1$

Ejemplo

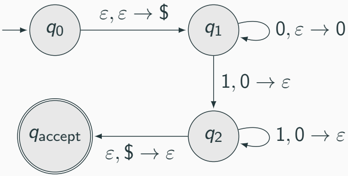


$$A_{q_0 q_{accept}} \to \varepsilon A_{q_1 q_2} \varepsilon$$
$$A_{q_1 q_2} \to 0 A_{q_1 q_2} 1$$
$$A_{q_1 q_2} \to 0 A_{q_1 q_1} 1$$
$$A_{q_1 q_1} \to \varepsilon$$

Simulación de la aceptación de 0011:

$$A_{q_0 q_{accept}} \Rightarrow \varepsilon A_{q_1 q_2} \varepsilon$$

Ejemplo

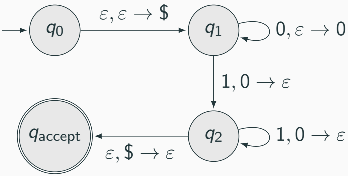


$$A_{q_0 q_{\text{accept}}} \to \varepsilon A_{q_1 q_2} \varepsilon$$
$$A_{q_1 q_2} \to 0 A_{q_1 q_2} 1$$
$$A_{q_1 q_2} \to 0 A_{q_1 q_1} 1$$
$$A_{q_1 q_1} \to \varepsilon$$

Simulación de la aceptación de 0011:

$A_{q_0 q_{\text{accept}}} \Rightarrow \varepsilon A_{q_1 q_2} \varepsilon \Rightarrow \varepsilon 0 A_{q_1 q_2} 1 \varepsilon$

Ejemplo

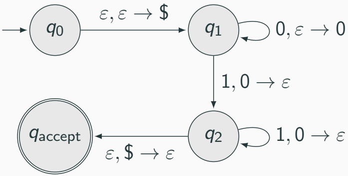


$$A_{q_0 q_{accept}} \to \varepsilon A_{q_1 q_2} \varepsilon$$
$$A_{q_1 q_2} \to 0 A_{q_1 q_2} 1$$
$$A_{q_1 q_2} \to 0 A_{q_1 q_1} 1$$
$$A_{q_1 q_1} \to \varepsilon$$

Simulación de la aceptación de 0011:

$$A_{q_0 q_{accept}} \Rightarrow \varepsilon A_{q_1 q_2} \varepsilon \Rightarrow \varepsilon 0 A_{q_1 q_2} 1 \varepsilon \Rightarrow \varepsilon 00 A_{q_1 q_1} 11 \varepsilon$$

Ejemplo



$A_{q_0 q_{accept}} \to \varepsilon A_{q_1 q_2} \varepsilon$

$A_{q_1 q_2} \to 0 A_{q_1 q_2} 1$

$A_{q_1 q_2} \to 0 A_{q_1 q_1} 1$

$A_{q_1 q_1} \to \varepsilon$

Simulación de la aceptación de 0011:

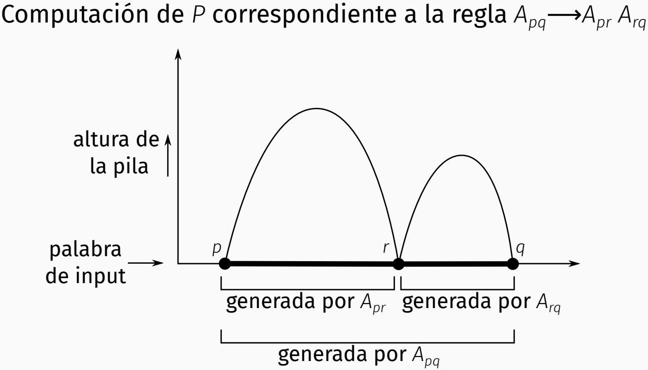
$A_{q_0 q_{accept}} \Rightarrow \varepsilon A_{q_1 q_2} \varepsilon \Rightarrow \varepsilon 0 A_{q_1 q_2} 1 \varepsilon \Rightarrow \varepsilon 00 A_{q_1 q_1} 11 \varepsilon \Rightarrow \varepsilon 00 \varepsilon 11 \varepsilon$

Hacia la implicación inversa

Si la pila queda vacía en algún paso intermedio:

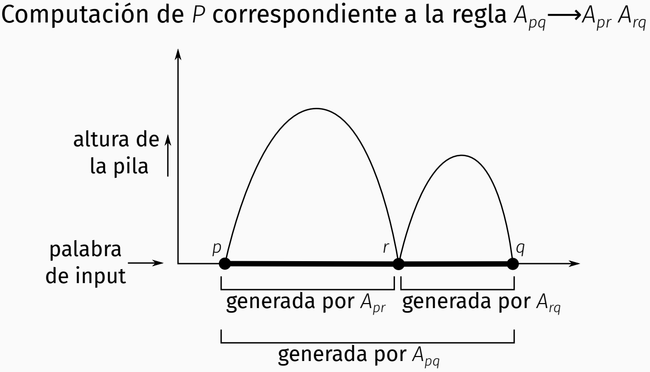
Hacia la implicación inversa

Si la pila queda vacía en algún paso intermedio:



Hacia la implicación inversa

Si la pila queda vacía en algún paso intermedio:



En este caso, si la pila se vacía en el estado r, definimos la regla $A_{pq} \to A_{pr} A_{rq}$.

La implicación inversa: la gramática G

Definiciones

Estamos listos para dar la definición formal de la gramática G.

Definiciones

Estamos listos para dar la definición formal de la gramática G. Supongamos que $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{\text{accept}}\})$.

Definiciones

Estamos listos para dar la definición formal de la gramática G. Supongamos que $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{\text{accept}}\})$.

Las variables de la gramática G son $\{A_{pq}, p, q \in Q\}$. La variable inicial es $A_{q_0 q_{\text{accept}}}$.

Definiciones

Estamos listos para dar la definición formal de la gramática G. Supongamos que $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{\text{accept}}\})$.

Las variables de la gramática G son $\{A_{pq}, p, q \in Q\}$. La variable inicial es $A_{q_0 q_{\text{accept}}}$.

Las reglas de G son las siguientes:

1. Para cada $p, q, r, s \in Q$, $u \in \Gamma$, $\mathtt{a}, \mathtt{b} \in \Sigma_\varepsilon$, si $(r, u) \in \delta(p, \mathtt{a}, \varepsilon)$ y $(q, \varepsilon) \in \delta(s, \mathtt{b}, u)$, definimos la regla $A_{pq} \to \mathtt{a}A_{rs}\mathtt{b}$.

Definiciones

Estamos listos para dar la definición formal de la gramática G. Supongamos que $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{\text{accept}}\})$.

Las variables de la gramática G son $\{A_{pq}, p, q \in Q\}$. La variable inicial es $A_{q_0 q_{\text{accept}}}$.

Las reglas de G son las siguientes:

1. Para cada $p, q, r, s \in Q$, $u \in \Gamma$, $\mathtt{a}, \mathtt{b} \in \Sigma_\varepsilon$, si $(r, u) \in \delta(p, \mathtt{a}, \varepsilon)$ y $(q, \varepsilon) \in \delta(s, \mathtt{b}, u)$, definimos la regla $A_{pq} \to \mathtt{a}A_{rs}\mathtt{b}$.
2. Para cada $p, q, r \in Q$ ponemos la regla $A_{pq} \to A_{pr}A_{rq}$.

Definiciones

Estamos listos para dar la definición formal de la gramática G. Supongamos que $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{\text{accept}}\})$.

Las variables de la gramática G son $\{A_{pq}, p, q \in Q\}$. La variable inicial es $A_{q_0 q_{\text{accept}}}$.

Las reglas de G son las siguientes:

1. Para cada $p, q, r, s \in Q$, $u \in \Gamma$, $\mathtt{a}, \mathtt{b} \in \Sigma_\varepsilon$, si $(r, u) \in \delta(p, \mathtt{a}, \varepsilon)$ y $(q, \varepsilon) \in \delta(s, \mathtt{b}, u)$, definimos la regla $A_{pq} \to \mathtt{a} A_{rs} \mathtt{b}$.
2. Para cada $p, q, r \in Q$ ponemos la regla $A_{pq} \to A_{pr} A_{rq}$.
3. Finalmente, para cada $p \in Q$ ponemos $A_{pp} \to \varepsilon$.

Definiciones

Estamos listos para dar la definición formal de la gramática G. Supongamos que $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{\text{accept}}\})$.

Las variables de la gramática G son $\{A_{pq}, p, q \in Q\}$. La variable inicial es $A_{q_0 q_{\text{accept}}}$.

Las reglas de G son las siguientes:

1. Para cada $p, q, r, s \in Q$, $u \in \Gamma$, $\mathtt{a}, \mathtt{b} \in \Sigma_\varepsilon$, si $(r, u) \in \delta(p, \mathtt{a}, \varepsilon)$ y $(q, \varepsilon) \in \delta(s, \mathtt{b}, u)$, definimos la regla $A_{pq} \to \mathtt{a} A_{rs} \mathtt{b}$.
2. Para cada $p, q, r \in Q$ ponemos la regla $A_{pq} \to A_{pr} A_{rq}$.
3. Finalmente, para cada $p \in Q$ ponemos $A_{pp} \to \varepsilon$.

En la próxima clase, mostraremos que G es de hecho equivalente a P.

Resumen

Resumen

Hoy aprendimos:

- Cómo escribir una cadena en la pila de un `PDA`;
- Cómo construir un `PDA` que reconozca un `CFL`;
- Cómo construir un `CFG` a partir de un `PDA`.