Equivalencia PDA y CFG

Alan Reyes-Figueroa

Teoría de la Computación (Aula 19) 29.septiembre.2025

Equivalencia de PDAs y CFGs

Conversion of CFG to PDA

Conversion of PDA to CFG

Resumen

- Cuando hablamos de propiedades de cerradura de lenguajes regulares, fue muy útil saltar entre las representaciones *regexp* y los DFA.
- Similarlmente, para las CFGs y los PDAs, ambos son útiles para mostrar propiedades relacionadas con los lenguajes libres del contexto.

Overview

- Además, los PDAs, siendo "algorítmicos," son más fáciles de usar cuando argumentamos que un lenguajes debe ser CFL.

- Ejemplo: Es más fácil ver cómo un PDA puede reconocer paréntesis balanceados; y menos vía una CFG.

Convirtiendo una CFG a PDA

- Sea L = L(G).
- Construimos un autómata de pila P tal que N(P) = L.
- P tiene:
  - Un estado q.
  - Símbolos input = terminales de G.
  - Símbolos stack = todos los símbolos de G.
  - Símbolo stack inicial = símbolo inicial de G.

Intuición acerca de P

- Dado el input w, P pasará por una derivación más a la izquierda de w desde el símbolo inicial S.
- Dado que P no puede saber cuál es esta derivación, ni siquiera cuál es el final de w, utiliza el no determinismo para "adivinar" la producción a utilizar en cada paso.

Intuición acerca de P

- En cada paso, P representa una *forma sentencial a la izquierda* (*left-sentential form*) (paso de una derivación leftmost).
- Si el stack de P es α, y P ha consumido una parte x de su input, entonces P representa la forma left-sentential $x\alpha$.
- Cuando el stack sea vacío, el input consumido es una cadena en L(G).

Función de Transición de P

1. $\delta(q, a, a) = (q, \epsilon)$. (Reglas *Tipo 1*)
   - Este paso no cambia el LSF representado, sino que "mueve" la responsabilidad de *a* de la pila a la entrada consumida.
2. Si $A \rightarrow \alpha$ es una producción de G, luego $\delta(q, \epsilon, A) = (q, \alpha)$. (Reglas *Tipo 2*)
   - Adivinar una producción para A y representar el siguiente LSF en la derivación.

Prueba de que L(P) = L(G)

- Necesitamos demostrar que $(q, wx, S) \vdash^* (q, x, \alpha)$ para cualquier x si y solo si, $S =>^*_{lm} w\alpha$.
- Parte 1: “solo si” es una inducción sobre el número de pasos realizados por P.
- Base: 0 pasos.
  - Tenemos $\alpha = S$, $w = \epsilon$, y $S =>^*_{lm} S$ es verdadera.

Inducción para la Parte 1

- Considere n pasos de P:
  $(q, wx, S) \vdash^* (q, x, \alpha)$ y asuma que la hipótesis de inducción vale para n-1 pasos.

- Tenemos dos casos: dependiendo de si la última movida fue una regla Tipo 1 o una regla Tipo 2.

Uso de una regla Tipo 1

- La secuencia de movidas debe ser de la forma
  $(q, yax, S) \vdash^* (q, ax, a\alpha) \vdash (q, x, \alpha)$,
  con $ya = w$.
- Por la hipótesis de inducción aplicada a los primeros n-1 pasos, $S \Rightarrow^*_{lm} ya\alpha$.
- Pero, $ya = w$, así que $S \Rightarrow^*_{lm} w\alpha$.

Uso de una regla Tipo 2

- La secuencia de pasos debe ser de la forma
  $(q, wx, S) \vdash^* (q, x, A\beta) \vdash (q, x, \gamma\beta)$,
  con $A \rightarrow \gamma$ una producción, y $\alpha = \gamma\beta$.
- Por la hipótesis inductive aplicada a los primeros n-1 pasos, $S =>^*_{lm} wA\beta$.
- Luego, $S =>^*_{lm} w\gamma\beta = w\alpha$.

Prueba de la Parte 2 ("si")

- Debemos demostrar también que si $S =>*_{lm} w\alpha$, entonces $(q, wx, S) \vdash * (q, x, \alpha)$ para cualquier x.
- Una inducción sobre el número de pasos en la derivación-leftmost.
- Las ideas son similares al paso anterior; (leer en el texto).

Prueba – Final

- Ahora tenemos $(q, wx, S) \vdash^* (q, x, \alpha)$ para cualquier x, si y solo si, $S \Rightarrow^*_{lm} w\alpha$.
- En particular, cuando $x = \alpha = \epsilon$.
- Entonces $(q, w, S) \vdash^* (q, \epsilon, \epsilon)$ si y solo si, $S \Rightarrow^*_{lm} w$.
- Esto es, $w \in N(P)$, si y solo si, $w \in L(G)$.

Convertir de PDA a CFG

- Ahora, supongamos que L = N(P).
- Construirmos una CFG G tal que L = L(G).
- Intuición: G tendrá variables que generan exactamente las entradas que hacen que P tenga el efecto de quitar un símbolo de pila X mientras pasa del estado p al q.
  - P nunca cae por debajo de esta X mientras lo hace.

Variables of G

- Las variables de G son de la forma [pXq].
- Esta variable genera todas y sólo aquellas cadenas w, tales que

$$(p, w, X) \vdash^* (q, \epsilon, \epsilon).$$

- Además, añadimos un símbolo inicial S, del cual hablaremos luego.

Producciones de G

- Cada producción [pXq] viene de un paso de P en el estado p con símbolo pila X.
- Caso más simple: $\delta(p, a, X) = (q, \epsilon)$.
- Entonces la producción es $[pXq] \rightarrow a$.
  - Note que *a* puede ser un símbolo input ó ϵ.
- Aquí, [pXq] genera *a*, ya que al leer el input *a* entonces se hace pop X en la pila y se va del estado p al q.

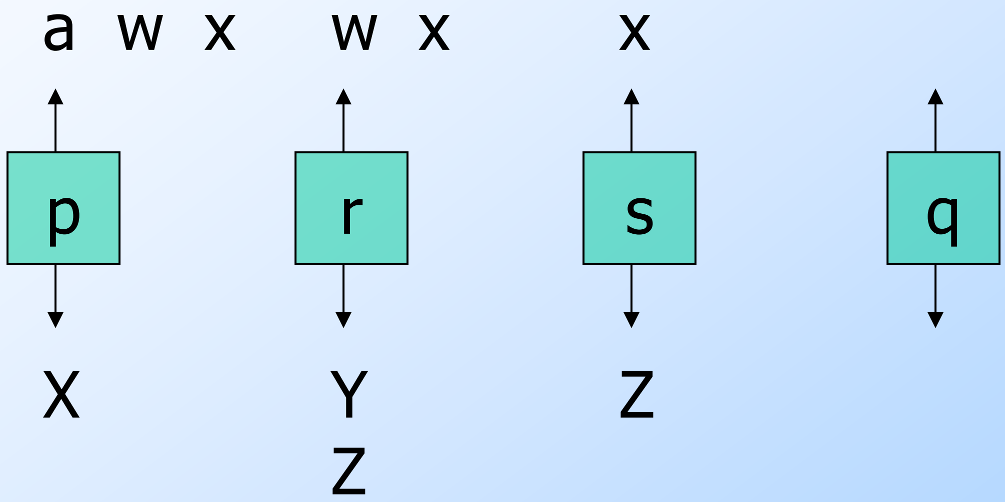
Producciones de G

- Siguiente caso: $\delta(p, a, X) = (r, Y)$ para algún estado r y símbolo pila Y.
- G tiene una producción $[pXq] \rightarrow a[rYq]$.
  - Borramos X de la pila, vamos del estado p al q leyendo el input *a* (entramos al estado r y reemplazamos la X por Y), y luego leyendo alguna cadena w que da P de r al q mientras borra la Y.
- Nota: $[pXq] \Rightarrow^* aw$, si $[rYq] \Rightarrow^* w$.

Producciones de G

- Tercer caso: $\delta(p, a, X) = (r, YZ)$ para algún estado r and símbolos Y, Z.
- Ahora, P reemplaza la X por YZ.
- Para entender el efecto completo de borrar la X, P debe borrar Y, e ir del estado r a algún estado s, y luego borrar la Z, pasando del estado s al q.

Picture of Action of P



Third-Simplest Case – Concluded

- Since we do not know state s, we must generate a family of productions:

  [pXq] -> a[rYs][sZq]

  for all states s.
- [pXq] =>* awx whenever [rYs] =>* w and [sZq] =>* x.

Productions of G: General Case

- Suppose $\delta(p, a, X)$ contains $(r, Y_1,\ldots Y_k)$ for some state r and $k \geq 3$.
- Generate family of productions

$$[pXq] \to a[rY_1s_1][s_1Y_2s_2]\ldots[s_{k-2}Y_{k-1}s_{k-1}][s_{k-1}Y_kq]$$

Completion of the Construction

- We can prove that $(q_0, w, Z_0) \vdash^* (p, \epsilon, \epsilon)$ if and only if $[q_0 Z_0 p] \Rightarrow^* w$.
  - Proof is in text; it is two easy inductions.
- But state p can be anything.
- Thus, add to G another variable S, the start symbol, and add productions $S \rightarrow [q_0 Z_0 p]$ for each state p.