

Trabajo Práctico - Validación de archivos PGN

Resumen

Los objetivos del presente trabajo fueron verificar la sintaxis de un archivo de tipo PGN (*portable game notation*), analizar si su formato es válido y determinar el máximo nivel de comentarios anidados en que no haya capturas, considerando todas las partidas.

Metodología

Para resolver el problema planteado en este TP se implementó código en lenguaje `Python 3.9` y se utilizó la biblioteca `PLY` para construir el *lexer* y el *parser* necesarios. Las gramáticas intermedias y finales fueron verificadas mediante [Grammophone](#).

Proceso de resolución

En una primera etapa, se plantearon los *tokens* a utilizar, realizando pruebas de ejecución con los ejemplos propuestos en el enunciado. Luego, se procedió a armar la gramática para el *parser*. En este punto, se hizo evidente que los *tokens* elegidos eran insuficientes para poder plantear una gramática no ambigua por lo que se procedió a realizar una segunda iteración.

En la segunda iteración, se implementaron los *tokens* finales y se procedió a realizar una primera aproximación mediante una gramática extendida. La gramática derivada obtenida no resultó ser siquiera **LR(1)**, por lo que se realizaron múltiples revisiones sobre la gramática extendida hasta poder obtener una gramática derivada **SLR(1)**. Finalmente, como la gramática resultaba muy compleja, se realizó un proceso de simplificación que derivó finalmente en la gramática expuesta a continuación.

En último lugar, fue necesario agregar reglas semánticas que permitieran verificar las siguientes condiciones.

- La numeración consecutiva de cada jugada
- La jugada interrumpida presenta la misma numeración para blancas y negras
- El máximo nivel de anidamiento en comentarios sin jugadas de captura

Para esto se diseñaron dos estructuras `Comentario` y `PGN`. `Comentario` permite almacenar el valor del `nesting` o anidamiento acumulado y determinar si dentro de su texto poseía una captura (`capture`). `PGN` almacena el máximo valor de anidamiento en comentarios en `nesting` y la validación de la numeración en el nodo `number`.

Tanto para el *lexer* como para el *parser*, en caso de haber errores se imprime un error y se detiene la ejecución.

Conjunto de tokens

El conjunto de *tokens* utilizado es el siguiente.

```
tokens = [
    'LPAREN', 'RPAREN', 'LLLAVE', 'RLLAVE',
    'DOT', 'DOTS', 'NUM', 'STRING', 'JUGADA',
    'DESCRIPTOR', 'RESULTADO', 'SPACE'
]
```

A continuación se detallan y explican las expresiones regulares que representan los *tokens* listados.

```
t_LPAREN    = r'\('
t_RPAREN    = r'\)'
t_LLLAVE    = r'\{'
t_RLLAVE    = r'\}'
t_DOT       = r'\.'
t_DOTS      = r'\.\.\.'
t_SPACE     = r'\ '
t_NUM       = r'[1-9][0-9]*'
t_STRING    = r'[a-zA-ZñÑáéíóúÁÉÍÓÚ\~! , ' ~]+'
t_JUGADA    = r'([PNBRQK]?[a-h]?[1-8]?x?[a-h][1-8]\+?\+?|O-O(-O)?(!|\?)?)'
t_DESCRIPTOR = r'\[[a-zA-Z0-9ñÑáéíóúÁÉÍÓÚ~]+\ "[a-zA-Z0-9ñÑáéíóúÁÉÍÓÚ\.\? \ -~]+"\]'
t_RESULTADO = r'0\~1|1\~0|1\~/2\~1\~/2'
```

El *token* `NUM` representa valores numéricos cuyo primer dígito no puede ser cero.

El de `JUGADA`, permite comenzar con la letra mayúscula que indica la pieza (P, N, B, R, Q, K) o no tener ninguno, en cuyo caso se interpreta como un peón. Luego aparecen las coordenadas opcionales de partida y el símbolo `x` que indica si la jugada representa una captura, a continuación las coordenadas de llegada que son obligatorias y un símbolo `+` o `++` que indica un jaque o jaque mate respectivamente. Otra opción es que la jugada constituya un enroque que puede representarse como `o-o-o` o `o-o`. Finalmente, cada jugada puede ser marcada como mala `?` o buena `!`.

El *token* `DESCRIPTOR` representado por dos cadenas de caracteres separadas por un espacio, una simple y otra encerrada por comillas, encerradas entre corchetes.

Finalmente, el `RESULTADO` consiste en una de las siguientes tres combinaciones `0~1`, `1~0` o `1/2~1/2`.

Gramática final

Primero se expone la gramática extendida.

```

S  -> (DESCRIPTOR+ P)+
P  -> J+
J  -> B (Ns | C SPACE RESULTADO | RESULTADO)
Ns -> N (Cs | RESULTADO | λ)
Cs -> C SPACE RESULTADO?
N  -> (C SPACE NUM DOTS SPACE)? JUGADA SPACE
B  -> NUM DOT SPACE JUGADA SPACE
C  -> { Cu } | ( Cu )
Cu -> T? (C T?)? | C T?
T  -> (STRING | NUM | DOT | JUGADA | SPACE | DOTS)+

```

Luego, a partir de la gramática extendida se obtuvo la gramática derivada que corresponde a una gramática SLR(1).

```

S  -> D P S | D P                                // Start
D  -> DESCRIPTOR D | DESCRIPTOR                  // Descriptores
P  -> J P | J                                      // Partida
J  -> B Ns | B C SPACE RESULTADO | B RESULTADO    // Jugada
B  -> NUM DOT SPACE JUGADA SPACE                  // Jugada de Blanca
Ns -> N Cs | N RESULTADO | N                      // Jugada de Negras
Cs -> C SPACE RESULTADO | C SPACE                  // Fin de jugada negra
comenzando con comentario
N  -> C SPACE NUM DOTS SPACE JUGADA SPACE | JUGADA SPACE // Jugada de Negras
C  -> { Cu } | ( Cu )                             // Comentario
Cu -> CT | CC                                      // Cuerpo del comentario
CT -> T CC | T                                     // Cuerpo que comienza con
texto
CC -> C T | C                                     // Cuerpo que comienza con
otro comentario
T  -> T1 T | T1                                    // Texto
T1 -> STRING | NUM | DOT | JUGADA | SPACE | DOTS . // Subpartes del texto

```

Ejecución

Para descargar las dependencias necesarias se debe ejecutar el comando.

```
pip install -r requirements.txt
```

Para ejecutar el programa, se debe tener en un archivo de texto el valor de entrada.

```
$ python3 pgn.py entrada.txt
```

La salida arrojará un error o el valor de máximo anidamiento registrado entre todas las partidas de la entrada.

Conclusiones

La principal dificultad del trabajo fue lograr entender cómo funcionaba la biblioteca `ply`. Creemos que hubiera sido más simple nuestro trabajo si hubiéramos contado con una clase introductoria para este tipo de herramientas. Para poder entender un poco más el uso de esta herramienta, primero realizamos una implementación de una gramática simple. Esta genera todas las posibles cadenas con parentesis anidados y balanceados (esto puede observarse en el archivo `parentesis-anidados.py`).

El siguiente conflicto encontrado, como mencionamos anteriormente, es el de encontrar los `tokens` adecuados para utilizar dentro del `lexer` y para definirlo realizamos un proceso iterativo. Basado en esto diseñamos una gramática *left-to-right rightmost derivation*(LR), validándola con la herramienta ya mencionada [Grammophone](#).

Finalmente el último desafío a afrontar fue la generación de las reglas semánticas que conforman la gramática de atributos. Para esto, a nivel implementativo, trabajamos con dos estructuras `Comentario` y `PGN` que nos permitieron tener más de un valor almacenado. Además definimos la función `condition`, la cual presenta un comportamiento análogo a la utilizada en la práctica levantando una excepción en el caso de no satisfacer la condición evaluada.

Algo a resaltar sobre los atributos utilizados en las reglas semánticas es que todos son de tipo *sintetizado*, esto indica que la gramática presentada es S-atribuida. Gracias a esto podemos asegurar que el grafo de dependencias no va a presentar ciclos para ninguna cadena del lenguaje, implicando que todos los atributos van a poder ser calculados.

Cómo trabajo a futuro podría proponerse un cambio dentro de la gramática que permita garantizar la alternancia entre comentarios que utilicen llaves y parentesis. Esto implicaría el cambio en las producciones utilizadas, agregando así mayor claridad sobre los comentarios utilizados dentro de una jugada. También sería posible obtener una mayor sobre la partida agregando o modificando las reglas semánticas existentes, por ejemplo: "máxima cantidad de jaques por archivo".

Anexo

Ejemplo de corrida 1

Entrada

```
[prueba "loca"]
[Nzscf5qWgtg~NVX "56B~n~nQIeAhy"]
[gvk7dXkliRpR "2LakQJGhz81"]
[~NFS51BHW4Mm~NmJsP "e4ZhVulzl"]
[yZ1PSI4r78KP "XwWzscEtUqkAu~nNt7Hq5"]
[GArzOdNa~nITcsbFO9ES "WUodxeqxi"]
```

1. 1d7 d7 2. Rf8 gg7 1-0

```
[2ujZrN6LTmOBGss5jzqw "pnAGk"]
[kRM "lihgftp5kLaiAvuNSub2"]
[vUPhgAiKx "a9C7isOkq0vyep7svNE"]
[Lbfh "j6htrDlwKNFryNGHdUcG"]
```

1. Pxe7 Pad3 2. Be6 b6 3. Bb2 h3 4. Qc1 Kh3 5. Pb1 Pb2 6. Ra1 a7! 7. Nce4 Qxa3 8. Kd4 Qh1 9. Kd8 fh6 10. Nc1 Pg1 11. Nh2 Bf4 12. ca6 Kd8 13. Nb4 d5 14. e1 Ra5 15. f8 K5xd3 16. b6 (Ka4 h5a2) 16... g7 17. e5 Bg2 18. f3 Re2 19. Qc6 Kb4 20. Rg4 Rg3 21. g3 N2c7 22. Pb7 Rh7 23. Qd3 {bf8 B b3 K Nxf5 wuFzW c1 Pbd2 T c8 ka Pg3 K Ke6+ n Kg5 Pb7 f8 e2 O 1c5 Rxe4 Pxc5} 23... e6 24. d5 Nh7 25. Qxc6 Bxc3 26. Bf5 Qfa3 27. f4 Ka7 28. Nxb6 d6 29. Bg1 Qf2 30. Pb3 d6 31. Ba3 e7 32. Bab4 Kda1 33. Ph1 e2 34. cf4 Nxd7 35. Nc7 (gr Pfg2 iONp utl Pcd2 t (Pd3 Rc2 Bxd6 Kxg8 Kxh2 h6 Bxc8 zAx g2 qX Nh2 N c3 ~n Bxb5 D Nd1 h8 a1 c7 PVAO Rg7 g5 Kb6 oGq) rt Pf8 Qp) 35... e2 36. Ra6 Bhb8 37. Q5a4 c5 38. Qg7 g1 39. g6 Pg7 40. Kd6 d1 41. h7+ Kcc5 42. Qd8 cel 43. Qxh3 Pa6 44. Bxe2 e4 45. Bh7 d2+ 1/2-1/2

[BYsR4sFynE2R~n2 "lNyKA4WPhvh2kKPTjfaD"]
[loCuhQ "WDOTwZIISKjDik~n6"]

1. Pb7 ec6 2. f4 Nxf7 3. Kxe5 d7 4. Kc7 Rd6 5. e7 Qc7 6. h1 {s Rxb7 iI d5 Nh7} 6... e4+ 7. 7d6 Nf7a3 8. Pa7 e2 9. h8 h4 10. g8 Kxf6 11. 2b3 Bxg3 12. Q5h1 Pb6 13. fd4 e5 14. 5c6 f3 15. cb6 {1 Pb6} 15... Pb1 16. Nb3 h7 17. b2 Nb7 18. Ra1 N8h6 19. f2 c8 20. P5g1 e1 21. Pg3 c3 22. f5 Kh1 23. Pd7 Nf3 24. Nd1 Rd1 25. Kd5 Ng8 26. e3 Nxg6 27. Rh8 fc3 28. Kxf1 Ncc4 29. c1+ e3 30. cc6 (a Kdxa3 e7 JQ Qe2 VLzeFq c7 FU f4 N Qxb7 Nd8++ Tf Nd2 yN ba2 me 1b6 hu Kd7 R Pxe6 Nxg7 DaR Ba8 Kxd5 Nd7 ttxI fe7 GK a2 m f6 Skp) 30... h8 31. Qh1 a1 32. fb7 b7 0-1

[4e "ws20"]

1. Bxd1 Pa6 0-1

[a "b"]

1. e4 d5 {defensa escandinava (es com'un 2. exd5 Da5 {no es com'un 2... c6})} 1/2-1/2

[Event "Mannheim"]
[Site "Mannheim GER"]
[Date "1914.08.01"]
[EventDate "1914.07.20"]
[Round "11"]
[Result "1-0"]
[White "Alexander Alekhine"]
[Black "Hans Fahrni"]
[ECO "C13"]
[WhiteElo "?"]
[BlackElo "?"]
[PlyCount "45"]

```
1. e4 {Notes by Richard Reti} 1... e6 2. d4 d5 3. Nc3 Nf6 4. Bg5 Be7 5. e5 Nfd7 6. h4
{This ingenious method of play which has subsequently been adopted by all modern
masters is characteristic of Alekhine's style.} 6... Bxg5 7. hxg5 Qxg5 8. Nh3 {! The
short-stepping knight is always brought as near as possible to the actual battle field.
Therefore White does not make the plausible move 8 Nf3 but 8 Nh3 so as to get the
knight to f4.} 8... Qe7 9. Nf4 Nf8 10. Qg4 f5 {The only move. Not only was 11 Qxg7
threatened but also Nxd5.} 11. exf6 gxf6 12. O-O-O {He again threatens Nxd5.} 12... c6
13. Re1 Kd8 14. Rh6 e5 15. Qh4 Nbd7 16. Bd3 e4 17. Qg3 Qf7 {Forced - the sacrifice of
the knight at d5 was threatened and after 17...Qd6 18 Bxe4 dxe4 19 Rxe4 and 20 Qg7
wins.} 18. Bxe4 dxe4 19. Nxe4 Rg8 20. Qa3 {Here, as so often happens, a surprising move
and one difficult to have foreseen, forms the kernel of an apparently simple Alekhine
combination.} 20... Qg7 {After 20.Qe7 21.Qa5+ b6 22.Qc3 would follow.} 21. Nd6 Nb6 22.
Ne8 Qf7 {White mates in three moves.} 23. Qd6+ 1-0
```

Salida

```
Máximo nivel de anidamiento: 3
```

Ejemplo de corrida 2

Entrada

```
1. 1d7 d7 2. Rf8 gg7 1-0
```

Salida

```
Exception: Parser: Syntax error at '1'
```

Ejemplo de corrida 3

Entrada

```
[elias "capo"]
```

```
1. 1d7 d7 3. Rf8 gg7 1-0
```

Salida

```
Exception: numeración no consecutiva
```

Ejemplo de corrida 4

Entrada

```
[elias "capo"]
```

```
2. 1d7 d7 3. Rf8 gg7 1-0
```

Salida

Exception: la numeración no comienza en 1

Ejemplo de corrida 5

Entrada

```
[elias "capo"]
```

```
1. 1d7 {esto es un comentario.} 2... d7 1-0
```

Salida

Exception: no mantiene numeración entre jugada de negras y blancas

Ejemplo de corrida 6

Entrada

```
[elias "capo"]
```

```
1. 1d7 {esto es un comentario (con un comentario adentro) que luego continúa (e intenta tener otro comentario).} 1... d7 1-0
```

Salida

Exception: Parser: Syntax error at '('

Aclaración: la gramática no permite tener un comentario que contenga dos comentarios al mismo nivel.

Ejemplo de corrida 7

Entrada

```
[elias "capo"]
```

```
1. 1d7 {esto es un comentario (con un comentario adentro xd7) que luego no continúa.}  
1... d7 1-0
```

Salida

Máximo nivel de anidamiento: 1