**Grammer:**

G -> < N > G < \ N >

G -> ''

N -> TABLE

TABLE -> table

TABLE -> TableTags

TableTags -> TD

TableTags -> TH

TableTags -> TR

TableTags -> tr

TableTags -> th

TableTags -> td

|  |  |  |
| --- | --- | --- |
| Non-Terminals | FOLLOW | FIRST |
| G | {$,<} | {<,''} |
| N | {>} | {table,TD,TH,TR,tr,th,td,''} |
| TABLE | {>} | {table,TD,TH,TR,tr,th,td} |
| TableTags | {>} | {TD,TH,TR,tr,th,td} |

Parsing with LL(1) parser:

Table:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Non-Terminal | table | TD | TH | TR | tr | th | td | < | > | \ | $ |
| G |  |  |  |  |  |  |  | G ->  <N>G<\N>  G -> '' |  |  | G -> '' |
| N | N -> TABLE | N -> TABLE | N -> TABLE | N -> TABLE | N -> TABLE | N -> TABLE | N->TABLE |  |  |  |  |
| TABLE | TABLE -> table | TABLE -> TableTags | TABLE -> TableTags | TABLE -> TableTags | TABLE -> TableTags | TABLE -> TableTags | TABLE -> TableTags |  |  |  |  |
| TableTags |  | TableTags -> TD | TableTags -> TH | TableTags -> TR | TableTags -> tr | TableTags -> th | TableTags->td |  |  |  |  |

There are double entries in the table thus the grammar can’t be parsed by the LL(1) Parser.

Parsing with SLR Parser:

|  |  |  |  |
| --- | --- | --- | --- |
| SLR CLOSURE TABLE | | |  |
| GOTO | Kernel | State | Closure |
|  | {G -> .< N > G < \ N >} | 0 | {G -> .< N > G < \ N >} |
| goto(0, <) | {G -> <.N > G < \ N >} | 1 | {G -> <.N > G < \ N >;  N -> .TABLE;  TABLE -> .table; TABLE -> .TableTags; TableTags -> .TD; TableTags -> .TH; TableTags -> .TR; TableTags -> .tr; TableTags -> .th; TableTags -> .td} |
| goto(1, N) | {G -> < N.> G < \ N >} | 2 | G -> < N.> G < \ N >} |
| goto(1, TABLE) | {N -> TABLE.} | 3 | {N -> TABLE.} |
| goto(1, table) | {TABLE -> table.} | 4 | {TABLE -> table.} |
| goto(1, TableTags) | {TABLE -> TableTags.} | 5 | {TABLE -> TableTags.} |
| goto(1, TD) | {TableTags -> TD.} | 6 | {TableTags -> TD.} |
| goto(1, TH) | {TableTags -> TH.} | 7 | {TableTags -> Th.} |
| goto(1, TR) | {TableTags -> TR.} | 8 | {TableTags -> TR.} |
| goto(1, tr) | {TableTags -> tr.} | 9 | {TableTags -> tr.} |
| goto(1, th) | {TableTags -> th.} | 10 | {TableTags -> th.} |
| goto(1, td) | {TableTags -> td.} | 11 | {TableTags -> td.} |
| goto(2, >) | {G -> < N >.G < \ N >} | 12 | {G -> < N >.G < \ N >; G -> .< N > G < \ N >; G -> .} |
| goto(12, G) | {G -> < N > G.< \ N >} | 13 | {G -> < N > G.< \ N >} |
| goto(12, <) | {G -> <.N > G < \ N >} | 1 |  |
| goto(13, <) | {G -> < N > G <.\ N >} | 14 | {G -> < N > G <.\ N >} |
| goto(14, \) | {G -> < N > G < \.N >} | 15 | {G -> < N > G < \.N >;  N -> .TABLE;  TABLE -> .table; TABLE -> .TableTags; TableTags -> .TD; TableTags -> .TH; TableTags -> .TR; TableTags -> .tr; TableTags -> .th; TableTags -> .td} |
| goto(15, N) | {G -> < N > G < \ N.>} | 16 | {G -> < N > G < \ N.>} |
| goto(15, TABLE) | {N -> TABLE.} | 3 |  |
| goto(15, table) | {TABLE -> table.} | 4 |  |
| goto(15, TableTags) | {TABLE -> TableTags.} | 5 |  |
| goto(15, TD) | {TableTags -> TD.} | 6 |  |
| goto(15, TH) | {TableTags -> TH.} | 7 |  |
| goto(15, TR) | {TableTags -> TR.} | 8 |  |
| goto(15, tr) | {TableTags -> tr.} | 9 |  |
| goto(15, th) | {TableTags -> th.} | 10 |  |
| goto(15, td) | {TableTags -> td.} | 11 |  |
| goto(16, >) | {G -> < N > G < \ N >.} | 17 | {G -> < N > G < \ N >.} |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LR Table | | | | | | | | | | | | | | | |
| State | ACTION | | | | | | | | | | | GOTO | | | |
| < | > | \ | table | TD | TH | TR | tr | th | td | $ | G | N | TABLE | TableTags |
| 0 | s1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 |  |  |  | s4 | s6 | s7 | s8 | s9 | s10 | s11 |  |  | 2 | 3 | 5 |
| 2 |  | S12 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  | r2 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  | r3 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  | r4 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  | r5 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  | r6 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  | r7 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 |  | r8 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  | r9 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  | r10 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 | s1/r1 |  |  |  |  |  |  |  |  |  | r1 | 13 |  |  |  |
| 13 | S14 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 |  |  | s15 |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 |  |  |  | s4 | s6 | s7 | s8 | s9 | s10 | s11 |  |  | 16 | 3 | 5 |
| 16 |  | s17 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 17 | acc |  |  |  |  |  |  |  |  |  | acc |  |  |  |  |

As there are conflicting entries in the table thus the grammar is not suitable for SLR Parser.

LL(0) parser:

Grammer:

G -> < N > G < \ N >

G -> ''

N -> LIST

LIST -> list

LIST -> ListTags

ListTags -> ul

ListTags -> li

ListTags -> ol

ListTags -> dl

ListTags -> dt

ListTags -> dd

ListTags -> UL

ListTags -> LI

ListTags -> OL

ListTags -> DL

ListTags -> DT

ListTags -> DD

ListTags -> ''

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Non-Terminal | list | ul | li | ol | dl | dt | dd | UL | LI | OL | DL | DT | DD | < | > | \ | br | $ |
| LIST | LIST -> list | LIST -> ListTags | LIST -> ListTags | LIST -> ListTags | LIST -> ListTags | LIST -> ListTags | LIST -> ListTags | LIST -> ListTags | LIST -> ListTags | LIST -> ListTags | LIST -> ListTags | LIST -> ListTags | LIST -> ListTags |  | LIST -> ListTags |  |  | LIST -> ListTags |
| ListTags |  | ListTags -> ul | ListTags -> li | ListTags -> ol | ListTags -> dl | ListTags -> dt | ListTags -> dd | ListTags -> UL | ListTags -> LI | ListTags -> OL | ListTags -> DL | ListTags -> DT | ListTags -> DD |  | ListTags -> '' |  |  | ListTags -> '' |
| G |  |  |  |  |  |  |  |  |  |  |  |  |  | G ->  <N>G<\N> G -> '' |  |  |  |  |
| N | N -> LIST N' | N -> LIST N' | N -> LIST N' | N -> LIST N' | N -> LIST N' | N -> LIST N' | N -> LIST N' | N -> LIST N' | N -> LIST N' | N -> LIST N' | N -> LIST N' | N -> LIST N' | N -> LIST N' |  | N ->LIST N’  N -> N’ |  |  |  |
| N’ |  |  |  |  |  |  |  |  |  |  |  |  |  |  | N'->N' N'->'' |  |  |  |

|  |  |  |
| --- | --- | --- |
| Non Terminals | FOLLOW | FIRST |
| LIST | {$,>} | {list,ul,li,ol,dl,dt,dd,UL,LI,OL,DL,DT,DD,''} |
| ListTags | {$,>} | {ul,li,ol,dl,dt,dd,UL,LI,OL,DL,DT,DD,''} |
| G | {<} | {<,''} |
| N | {>} | {list,ul,li,ol,dl,dt,dd,UL,LI,OL,DL,DT,DD,''} |
| N’ | { > } | {''} |