

Diagram illustrating the LR(0) item sets and transitions for the grammar:

- States (Items):**
 - S** (Start state)
 - line comm**
 - IND**
 - EOL**
 - EOF**
 - tab space**
- Transitions:**
 - S** → **tab space** → **S**
 - S** → **#** → **line comm**
 - S** → **EOL** → **EOL**
 - S** → **EOF** → **EOF**
 - line comm** → **O_A** → **line comm**
 - line comm** → **EOL** → **EOL**
 - line comm** → **EOF** → **EOF**
 - line comm** → **IND** (labeled **is_indent_enabled??**)
 - IND** → **#** → **line comm**
 - IND** → **EOL** → **IND**
 - IND** → **EOF** → **EOF**
 - IND** → **tab space** → **IND**
 - IND** → **EOL** → **tab space**
 - tab space** → **tab space** (self-loop)
 - tab space** → **EOF** → **EOF**
 - EOL** → **EOL** (self-loop)
 - EOL** → **EOF** → **EOF**
 - EOF** → **EOF** (self-loop)

The diagram illustrates the states and transitions of a lexical analyzer. States are represented by circles, with double circles indicating final states. Transitions are labeled with tokens or character classes in boxes. The states and transitions are as follows:

- Start State:** `S` (double circle) with an incoming arrow.
- Initial States:** `int`, `dot`, `0`, `bin pref`, `hex pref`, `okt pref`.
- Final States:** `bin`, `bin_u_line`, `okt`, `okt_u_line`, `hex`, `hex_u_line`.
- Transitions:**
 - `S` to `int` (labeled `1-9`), `dot` (labeled `dot`), `0` (labeled `0`), `bin pref` (labeled `b, B`), `hex pref` (labeled `x, X`), `okt pref` (labeled `o, O`).
 - `int` to `int_u_line` (labeled `u_line`), `dot` (labeled `dot`), `e, E` (labeled `e, E`), `0-9` (labeled `0-9`).
 - `dot` to `dbl dot` (labeled `0-9`), `e, E` (labeled `e, E`).
 - `0` to `bin` (labeled `0-1`), `bin_u_line` (labeled `u_line`), `hex pref` (labeled `0-9, a-f, A-F`), `okt pref` (labeled `0-7`).
 - `bin` to `bin_u_line` (labeled `u_line`), `0-1` (labeled `0-1`).
 - `bin_u_line` to `bin` (labeled `0-1`).
 - `okt pref` to `okt` (labeled `u_line`), `bin_u_line` (labeled `u_line`).
 - `okt` to `bin_u_line` (labeled `u_line`), `0-7` (labeled `0-7`).
 - `hex pref` to `hex` (labeled `0-9, a-f, A-F`), `hex_u_line` (labeled `u_line`).
 - `hex` to `hex_u_line` (labeled `u_line`), `0-9, a-f, A-F` (labeled `0-9, a-f, A-F`).
 - `hex_u_line` to `hex` (labeled `u_line`).
 - `dbl dot` to `dbl` (labeled `0-9`), `e, E` (labeled `e, E`).
 - `dbl` to `dbl_u_line` (labeled `u_line`), `e, E` (labeled `e, E`), `e dbl` (labeled `e, E`).
 - `dbl_u_line` to `dbl` (labeled `u_line`).
 - `e dbl` to `e dbl sig` (labeled `+, -`), `e dbl F` (labeled `0-9`).
 - `e dbl sig` to `e dbl` (labeled `0-9`).
 - `e dbl F` to `e dbl` (labeled `0-9`), `0-9` (labeled `0-9`).
 - `e dbl` to `int` (labeled `0-9`).

under line	u_line
prefix	pref

ASCII>31	A31
other ASCII>31	O_A31
apostrophe	apos
string = =string	

