

Lab2 说明

1.处理注释:

当碰见表示注释开始的“/*”时, 设置 `comment_level=1`, 并过渡到另一种相对独立的状态机的初始状态。

在 `<COMMENT>` 中, 忽略一切字符。当碰到 “/*” 时, 说明嵌套了一层注释, 将 `comment_level++`; 当碰到: “*/” 时, 将 `comment_level--`, 再进行判断: 若 `comment_level==0` 说明注释结束, 返回状态 `INITIAL`; 否则说明只是某层的嵌套注释结束了, 外面还有注释, 继续在当前状态执行。

2.处理字符串:

当碰见表示字符串开始的双引号时, 首先调用 `adjust()` 确定字符串 `token` 的开始位置, 然后清空 `string_buf_`, 过渡到另一个相对独立的状态机的初始状态。

在 `<STR>` 中:

对于 `\\, \n, \", \t`, 这种直接手动添加进 `string_buf_` 中;

对于 `\\ddd` 的这种情况, 首先使用 `matched().c_str()` 得到一个匹配到的串的指针, +1 略过 “\”, 然后用 `atoi` 得到对应的 ASCII 码, 最后(char)强制转换为该 ASCII 码对应的字符。

对于 `^c`, 查阅 ASCII 码表观察得知控制字符从 1-31, 且 1 对应的是 ^A 这样较为规整。故可以 `matched()[2] - 'A' + 1` 这样来得到实际所表示的控制字符。

对于 `\f__f`, 可以用 `\\[\n\t\f]+\\` 来省略掉其间的空格符, 换行符, 制表符和换页符。

对于所有其它的都可以直接加入到 `string_buf_` 中。

当碰到另一个双引号时, 将 `string_buf_` 设为 `match` 到的串, 返回 `STRING` 并回到 `INITIAL` 中。

需要注意的是, 除了第一个双引号外都应用 `adjustStr()`, 不然这个 `STRING` 型 `token` 出现的位置会出错。

3.处理错误:

对于没有规定过的规则以外的东西, 都当作错误来进行处理。

调用 `adjust()` 后, 通过 `errmsg->Error(errmsg->tok_pos_, "illegal token")` 来报告, 既能识别错误, 也不会碰见一个错误就停止识别。

4.文件结束处理

直接返回