

移进 - 归约 (shift-reduce) 分析器的设计

移进 - 归约

| stack | action | rest of scan |
|----------------------------|---------|--------------|
| \$ exp PLUS term | shift | TIMES ID \$ |
| \$ exp PLUS term TIMES | shift | ID \$ |
| \$ exp PLUS term TIMES ID | reduce | \$ |
| \$ exp PLUS term TIMES fac | reduce | \$ |
| \$ exp PLUS term | reduce | \$ |
| \$ exp | success | \$ |

例子

移进 - 归约 (shift-reduce) 分析器的设计

移进 - 归约

stack

\$ exp PLUS term

action

shift

rest of scan

TIMES ID \$

\$ exp PLUS term TIMES

shift

ID \$

\$ exp PLUS term TIMES ID

reduce

\$

\$ exp PLUS term TIMES fac

reduce

\$

\$ exp PLUS term

reduce

\$

\$ exp

success

\$

例子



移进 - 归约 (shift-reduce) 分析器的设计

移进 - 归约

stack

\$ exp PLUS term

\$ exp PLUS term TIMES

\$ exp PLUS term TIMES ID

\$ exp PLUS term TIMES fac

\$ exp PLUS term

\$ exp

action

shift

shift

reduce

reduce

reduce

success

rest of scan

TIMES ID \$

ID \$

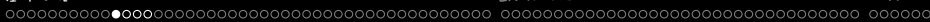
\$

\$

\$

\$

例子



移进 - 归约 (shift-reduce) 分析器的设计

移进 - 归约

stack

\$ exp PLUS term

\$ exp PLUS term TIMES

\$ exp PLUS term TIMES ID

\$ exp PLUS term TIMES fac

\$ exp PLUS term

\$ exp

action

shift

shift

reduce

reduce

reduce

success

rest of scan

TIMES ID \$

ID \$

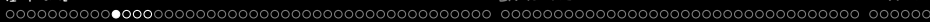
\$

\$

\$

\$

例子

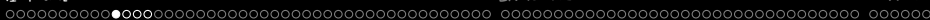


移进 - 归约 (shift-reduce) 分析器的设计

移进 - 归约

| stack | action | rest of scan |
|----------------------------|---------|--------------|
| \$ exp PLUS term | shift | TIMES ID \$ |
| \$ exp PLUS term TIMES | shift | ID \$ |
| \$ exp PLUS term TIMES ID | reduce | \$ |
| \$ exp PLUS term TIMES fac | reduce | \$ |
| \$ exp PLUS term | reduce | \$ |
| \$ exp | success | \$ |

例子

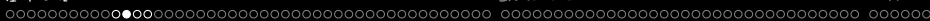


移进 - 归约 (shift-reduce) 分析器的设计

移进 - 归约

| stack | action | rest of scan |
|--------------------------------------|---------|--------------|
| \$ exp PLUS term | shift | TIMES ID \$ |
| \$ exp PLUS term TIMES | shift | ID \$ |
| \$ exp PLUS term TIMES ID | reduce | \$ |
| \$ exp PLUS term TIMES fac | reduce | \$ |
| \$ exp PLUS term | reduce | \$ |
| \$ exp | success | \$ |

例子



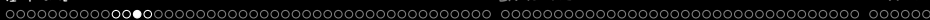
移进 - 归约 (shift-reduce) 分析器的设计.

分析栈

- 保留已经归约的句型.
- 自左向右扫描.
- "stack" + "rest of scan" = 最右句型.
- 句柄总是在栈顶形成.
- 无回溯.

问题

- 在归约无误的前提下, 句柄是否能保证总是在栈顶形成.
- 如何不回看栈中元素, 仅根据栈顶的状态和当前的输入就能够正确地做出移进或归约的操作.

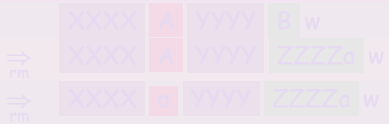


句柄一定在栈顶 (1/2)

句柄一定在栈顶

| stack | action | rest of scan |
|----------------------------|--------|--------------|
| \$ XXXX a YYYY ZZZZ | shift | a b \$ |
| \$ XXXX a YYYY ZZZZ a | reduce | b \$ |
| \$ XXXX a YYYY B | reduce | b \$ |
| \$ XXXX A YYYY B | error | b \$ |

这与最右推导矛盾：

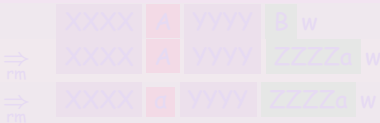


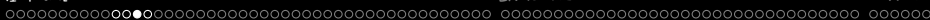
句柄一定在栈顶 (1/2)

句柄一定在栈顶

| stack | action | rest of scan |
|-----------------------|--------|--------------|
| \$ XXXX a YYYY ZZZZ | shift | a b \$ |
| \$ XXXX a YYYY ZZZZ a | reduce | b \$ |
| \$ XXXX a YYYY B | reduce | b \$ |
| \$ XXXX A YYYY B | error | b \$ |

这与最右推导矛盾:



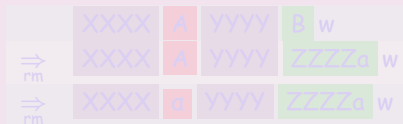


句柄一定在栈顶 (1/2)

句柄一定在栈顶

| stack | action | rest of scan |
|-------------------------------------|--------|--------------|
| \$ XXXX a YYYY ZZZZ | shift | a b \$ |
| \$ XXXX a YYYY ZZZZ a | reduce | b \$ |
| \$ XXXX a YYYY B | reduce | b \$ |
| \$ XXXX A YYYY B | error | b \$ |

这与最右推导矛盾:



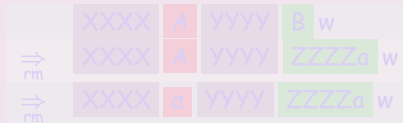


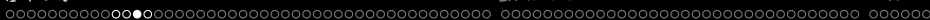
句柄一定在栈顶 (1/2)

句柄一定在栈顶

| stack | action | rest of scan |
|-----------------------|--------|--------------|
| \$ XXXX a YYYY ZZZZ | shift | a b \$ |
| \$ XXXX a YYYY ZZZZ a | reduce | b \$ |
| \$ XXXX a YYYY B | reduce | b \$ |
| \$ XXXX A YYYY B | error | b \$ |

这与最右推导矛盾:





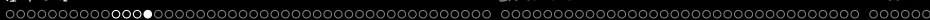
句柄一定在栈顶 (1/2)

句柄一定在栈顶

| stack | action | rest of scan |
|----------------------|--------|--------------|
| \$ XXXX a YYY Y ZZZZ | shift | a b \$ |
| \$ XXXX a YYY ZZZZ a | reduce | b \$ |
| \$ XXXX a YYY B | reduce | b \$ |
| \$ XXXX A YYY B | error | b \$ |

这与最右推导矛盾:

| | | | | |
|---------------|------|---|-----|--------|
| | XXXX | A | YYY | B w |
| \Rightarrow | XXXX | A | YYY | ZZZa w |
| \Rightarrow | XXXX | a | YYY | ZZZa w |

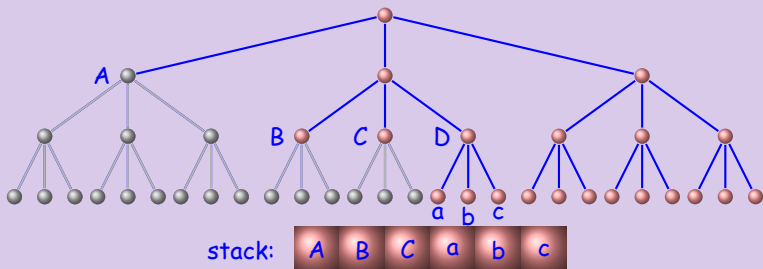


句柄一定在栈顶 (2/2)

原因

- 栈内的元素一定是已经归约到不能再归约的对象，否则，由于是句柄，在新的移进前还要归约。
- 句柄的最后一个文法符号一定是栈顶元素。
- 最右归约过程是自底向上从左到右的**截枝**过程。

截枝





如何判断句柄已经在栈顶形成。

活前缀 (viable prefix)

- 最右句型中句柄之前 (含句柄) 的文法符号串。
- 最右句型的一个前缀, 该前缀不含有句柄以后的文法符号。
- 移进 - 归约分析栈中的符号串。

Example

Sentential form : exp PLUS term TIMES ID w

Viable prefix {

| |
|-------------------------------|
| exp |
| exp PLUS |
| exp PLUS term |
| exp PLUS term TIMES |
| exp PLUS term TIMES <u>ID</u> |