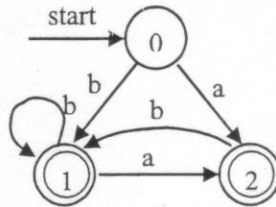


第 1 部分：编译原理 共 40 分

一、描述题（每小题 10 分，共 40 分）

- 1、正规式 $b^*a(bb^*a)^*b^*$ 体现的特点是，每个 a 的左边都有若干 b ，除非 a 是第一个字母。因此该正规式定义的语言是：不含子串 aa 的所有 a 和 b 的非空串的集合。最简 DFA 如下：



- 2、消除左递归后的文法如下：

$$B \rightarrow 1 B'$$

$$B' \rightarrow 0 B' \mid 1 B' \mid \epsilon$$

相应的翻译方案如下：

$$B \rightarrow 1 \{B'.i := 1\} B' \{B.val := B'.val\}$$

$$B' \rightarrow 0 \{B'.i := B'.i \times 2\} B'_1 \{B'.val := B'_1.val\}$$

$$| 1 \{B'.i := B'.i \times 2 + 1\} B'_1 \{B'.val := B'_1.val\}$$

$$| \epsilon \{B'.val := B'.i\}$$

3. 左边的编译器版本：一般只为局部变量分配空间。调用函数前，用若干次 `pushl` 指令将参数压栈，返回后用 `addl $n, %esp` 一次将所有参数退栈（常数 n 根据调用前做了多少次 `pushl` 来决定）。

右边的编译器版本：除了为局部变量分配空间外，同时还为本函数中出现的函数调用的参数分配空间，并且参数所用空间靠近栈顶。调用函数前，用 `movl` 指令将参数移入栈顶，调用结束后无需参数退栈指令。

优点是每次函数调用结束后不需要执行 `addl $n, %esp` 指令，另外增加优化的可能性。

4. 表达式 $\&i$ 的类型表达式是 `pointer(long)`，表达式 $\&i-\&j$ 的类型表达式是 `long`。按照 C 语言的规定，指向同一个类型的两个指针可以相加减，它们值的差是它们之间的元素个数。