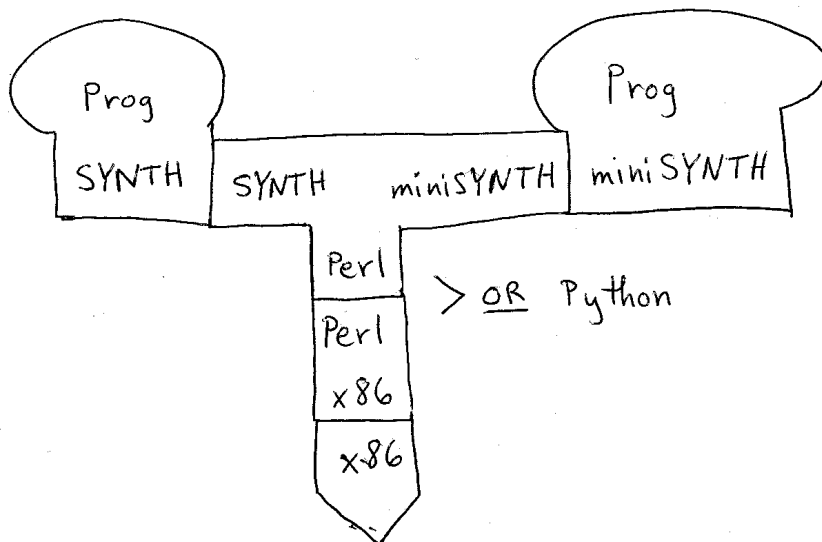


Exercise 1 Solutions

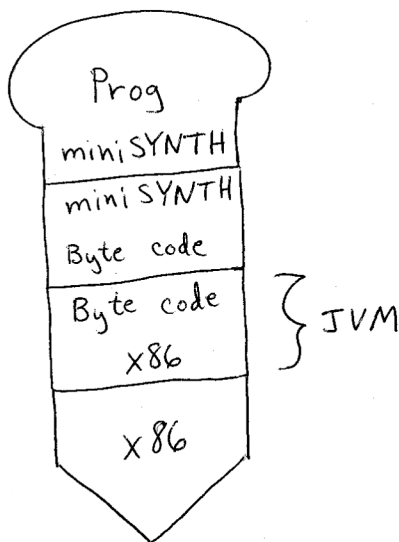
Due in class (on paper) at 3:30 PM, January 29, 2009.

1. (20 pts) Construct tombstone diagrams to illustrate the use of the tools developed in Programming Assignment 1:

a. compiling a SYNTH program



b. running a miniSYNTH program



2. (40 pts) For each of the following Java program fragments, identify the type of error: Lexical (detected by scanner), Syntactic (detected by parser), Semantic static (detected by semantic analysis at compile time), Semantic dynamic (detected at run time)

```
a.    int num = 2.3;

// semantic static (type mismatch)

b.    int;

// syntactic (invalid statement structure)

c.    int@

// lexical (no token matches '@')

d.    int num1, num2;
      num1 = 12;
      num2 = num1 / 0;

// semantic dynamic (divide by zero exception at run time -
// could be semantic static in this case if the Java
// compiler were "smarter"...but it's not)

e.    int num;
      String str = "dook";
      num = str;

// semantic static (type mismatch)

f.    int int;

// syntactic (invalid statement structure)

g.    int num;
      if ("unc") { num = 1; }

// semantic static (type mismatch)

h.    class A {
      private int num;
      }
      class B {
      A a;
      int num;
      void m() { a = new A();
                num = a.num;
      }
```

```

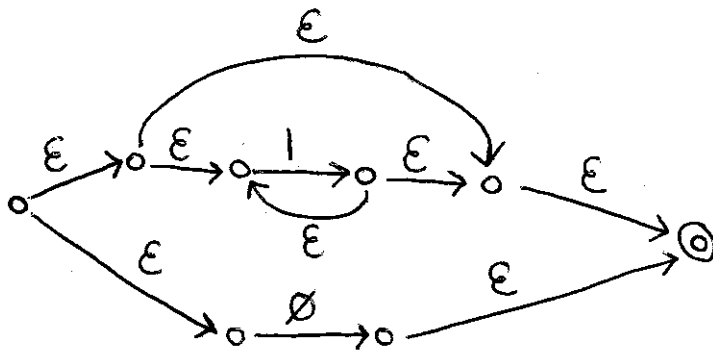
    }
}

```

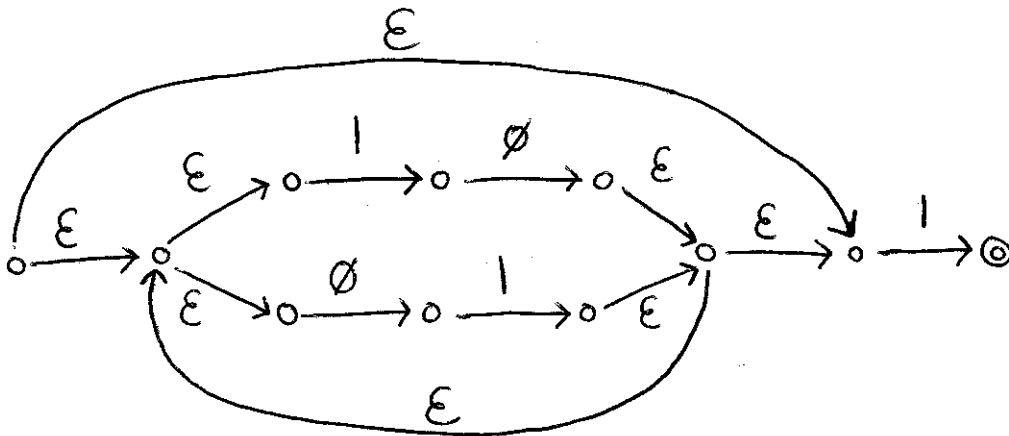
// semantic static (field not visible)

3. (20 pts) Construct an NFA for the following REs

a. $1^*|0$



b. $(10|01)^*1$



4. (10 pts) Ex. 2.1 part (e), pages 96-97 in PLP. Ignore the requirement to group digits with commas.

```
nonzerodigit -> 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

```
digit -> 0 | nonzerodigit
```

without commas:

```
quantity -> $ (*)* (0 | (nonzerodigit digit*)) (. digit  
digit | ε)
```

with commas:

```
quantity -> $ (*)* (0 | (nonzerodigit (digit digit | digit  
| ε) (, digit digit digit)*)) (. digit digit | ε)
```

5. (10 pts) Consider the following “hello world” program in C:

```
#include <stdio.h>

int main(int argv, char *argc[])
{
    printf ("Hello world!\n");
    return 0;
}
```

Log in to the machine classroom.cs.unc.edu and create the file hello.c containing the code above.

a. Compile the code with the verbose option turned on, as shown below:

```
cc -v -o hello hello.c
```

The compiler gives some output about what it is doing. Attach that output to your homework submission.

```
classroom(6)% cc -v -o hello hello.c
Using built-in specs.
Target: i386-redhat-linux
Configured with: ../configure --prefix=/usr --
mandir=/usr/share/man --infodir=/usr/share/info --enable-
shared --enable-threads=posix --enable-checking=release --
with-system-zlib --enable-__cxa_atexit --disable-libunwind-
```

```

exceptions --enable-libgcj-multifile --enable-
languages=c,c++,objc,obj-c++,java,fortran,ada --enable-
java-awt=gtk --disable-dssi --enable-plugin --with-java-
home=/usr/lib/jvm/java-1.4.2-gcj-1.4.2.0/jre --with-
cpu=generic --host=i386-redhat-linux
Thread model: posix
gcc version 4.1.2 20080704 (Red Hat 4.1.2-44)
/usr/libexec/gcc/i386-redhat-linux/4.1.2/cc1 -quiet -v
hello.c -quiet -dumpbase hello.c -mtune=generic -auxbase
hello -version -o /tmp/ccKsa3Rc.s
ignoring nonexistent directory "/usr/lib/gcc/i386-redhat-
linux/4.1.2/../../../../i386-redhat-linux/include"
#include "... " search starts here:
#include <...> search starts here:
/usr/local/include
/usr/lib/gcc/i386-redhat-linux/4.1.2/include
/usr/include
End of search list.
GNU C version 4.1.2 20080704 (Red Hat 4.1.2-44) (i386-
redhat-linux)
    compiled by GNU C version 4.1.2 20080704 (Red Hat
4.1.2-44).
GGC heuristics: --param ggc-min-expand=99 --param ggc-min-
heapsize=129364
Compiler executable checksum:
ca3063d3242816cbb24ab2a824493154
as -V -Qy -o /tmp/ccWE514E.o /tmp/ccKsa3Rc.s
GNU assembler version 2.17.50.0.6-6.el5 (i386-redhat-linux)
using BFD version 2.17.50.0.6-6.el5 20061020
/usr/libexec/gcc/i386-redhat-linux/4.1.2/collect2 --eh-
frame-hdr -m elf_i386 --hash-style=gnu -dynamic-linker
/lib/ld-linux.so.2 -o hello /usr/lib/gcc/i386-redhat-
linux/4.1.2/../../../../crt1.o /usr/lib/gcc/i386-redhat-
linux/4.1.2/../../../../crti.o /usr/lib/gcc/i386-redhat-
linux/4.1.2/crtbegin.o -L/usr/lib/gcc/i386-redhat-
linux/4.1.2 -L/usr/lib/gcc/i386-redhat-linux/4.1.2 -
L/usr/lib/gcc/i386-redhat-linux/4.1.2/../../../../
/tmp/ccWE514E.o -lgcc --as-needed -lgcc_s --no-as-needed -
lc -lgcc --as-needed -lgcc_s --no-as-needed
/usr/lib/gcc/i386-redhat-linux/4.1.2/crtend.o
/usr/lib/gcc/i386-redhat-linux/4.1.2/../../../../crtn.o

```

b. Compile the code to assembly, as shown below:

```
cc -S hello.c
```

The file hello.s will be created containing the assembly code. Attach its contents to your homework submission.

```
classroom(7)% cc -S hello.c
classroom(8)% cat hello.s
    .file      "hello.c"
    .section   .rodata
.LC0:
    .string   "Hello world!"
    .text
.globl main
    .type     main, @function
main:
    leal 4(%esp), %ecx
    andl $-16, %esp
    pushl    -4(%ecx)
    pushl    %ebp
    movl %esp, %ebp
    pushl    %ecx
    subl $4, %esp
    movl $.LC0, (%esp)
    call puts
    movl $0, %eax
    addl $4, %esp
    popl %ecx
    popl %ebp
    leal -4(%ecx), %esp
    ret
    .size     main, .-main
    .ident    "GCC: (GNU) 4.1.2 20080704 (Red Hat 4.1.2-
44)"
    .section  .note.GNU-stack,"",@progbits
```