

CMSC435 Software Engineering

Quiz 11

Name:

Program generators are popular examples of “software reuse”. When writing a program generator requires considerable intellectual effort, it is pleasant to amortize that effort by using the generator to build more than one application. A few program generators are intended for reuse. Examples include the lexer generator Lex and the parser generator Yacc.

Discuss the type of reuse that program generators offer. Give a detailed example of how a “compiler generator” promotes reuse. You may use code examples, high-level specifications, diagrams or a combination to explain your answer. Points will be awarded for providing a self-contained example.

Solution:

As discussed in <http://www.cs.umd.edu/~atif/Teaching/Spring2011/Slides/10.pdf>, program generators offer “**Generator-based reuse**” (Slide 8).

A compiler generator is a tool that creates a parser, interpreter, or compiler from some form of formal description of a language and machine. End users can vary this description and quickly generate/regenerate a new compiler. Code that is common to most compilers (e.g., error handling) is reused and hence does not have to be rewritten by the end user.

Example: Let’s say we have a thermostat that we want to control using a simple language. A session with the thermostat may look like this:

```
heat on
```

```
Heater on!
```

```
heat off
```

```
Heater off!
```

```
target temperature 22
```

```
New temperature set!
```

The tokens we need to recognize are: heat, on/off (STATE), target, temperature, and NUMBER.

As our language is very basic, so is the grammar:

```
commands: /* empty */
| commands command
;
command:
heat_switch
| target_set
;
heat_switch:
TOKHEAT STATE
{
printf("\tHeat turned on or off\n");
};
target_set:
TOKTARGET TOKTEMPERATURE NUMBER
{
printf("\tTemperature set\n");
};
```

More complete example at:

http://publib.boulder.ibm.com/infocenter/aix/v6r1/index.jsp?topic=/com.ibm.aix.genprogc/doc/genprogc/ie_prog_4lex_yacc.htm