

Compiler: exercises

Lecture 12

Formal Languages and Compilers 2011

Nataliia Bielova

Exercises!

- repeat-until without begin and end (test_repeat.cre, test_repeat2.cre)

```
repeat
  x := x + 1;
  sum := sum + x
until 10 < x
```

- Allow “;” at the end of the list ☺ (test_list.cre)

```
program
  var x : int;
  var sum : int;

begin
  x := x + 1;
  sum := sum + x;
end
```

Exercises!

- Allow if without else (test_it.cre)

```
if (true) then if (false) then write(0) else write(1);
```

- Shift/reduce conflict:

- If we reduce, THEN rule gets a precedence:

```
%nonassoc ELSE  
%nonassoc THEN
```

```
if (true) then  
    if (false) then write(0)  
else write(1); // no output
```

- If we decide to shift, ELSE rule should get a precedence:

```
%nonassoc THEN  
%nonassoc ELSE
```

```
if (true) then  
    if (false) then write(0)  
else write(1); // output 1
```

Exercises (2)

- Comments in one line: (test_comment.cre)

```
if z > 4 then    // No one can be greater
    z := 4      // than four!
else
    z := 0;     // Hey, it's cool to have comments!
```

- Comments in several lines: (test_comment2.cre)

```
fib := fib0 + fib1; /* calculate the sum
                    and write whatever you want
                    in many lines! */
fib0 := fib1;
```

- It is still possible to count the number of lines!

Exercises (3)

- Yet another type of cycle (test_do.cre)

```
do  
    cmd  
while bexp
```