

Model - Based Software Development

2024. 04. 15. Midterm – A

Name:

Neptun code:

Task	Points
1	/ 10p
2	/ 10p
3	/ 10p
4	/ 10p
5	/ 10p
6	/ 10p
7	/ 10p
Σ	/ 70p

- 1) What are the main differences between a compiler and an interpreter? (6-8 sentences) (10p)
- 2) What kinds of obfuscation techniques do you know? (1-1 sentence/technique) (10p)
- 3) What are the main differences between UML Profile and metamodeling? (6-8 sentences) (10p)
- 4) What does the following regular expression describe? Give a short 1-2 sentence explanation and an example matching text! Explain the meaning of each grouping (capturing group) separately! (10p)

 $\wedge(\text{class } [A-Z][a-zA-Z0-9]*) \left(\left([a-zA-Z]*=[a-zA-Z0-9]* \right)^+ \right) \}$$
- 5) Draw the syntax tree and compute the values of the attributes based on the following attribute grammar and program code! (10p)

Attribute grammar:

$S \rightarrow A$

$A.y = A.x$

$A \rightarrow pAq$

$A[1].x = A[2].x+2$

$A[2].y = A[1].y/2$

$A \rightarrow B$

$A.x = B.x$

$B.y = A.y$

$B \rightarrow bB$

$B[1].x = B[2].x+1$

$B[2].y = B[1].y*2$

$B \rightarrow w$

$B.x = 1$

Program code:

ppbbbwqq

Model - Based Software Development – Midterm – A

Name:

Neptun code:

- 6) Let's take the following C# code! Convert the body of the function `foo` to SSA form! Variables holding temporary values should be based on the name `t`! You can assume that 32-bit architecture is used. Evaluate the basic blocks in the following order: code before the loop, loop condition, code after the loop, body of the loop. Make sure to number the variables with consecutive numbers in the order of the evaluation! (10p)

```
int foo(int[] a, int c)
{
    int b = 1;
    for (int i = c-1; i >= 0; --i)
    {
        b += a[i];
    }
    return b;
}
```

- 7) Illustrate the optimization techniques using the code snippet below! The name of the technique is required and please provide the final (optimized) code snippet as well. You can suppose that only variables `a`, `e` and `f` are needed after the given code snippet. There is a penalty if wrong optimization is applied. (10p)

```
...
e = 3;
a = fun();
a = a*e;
b = c*e;
d = c+b;
f = 4*c;
print(f);
f = 5;
if (e > f) {
    a = e;
}
else {
    print(b+d);
}
```