



Faculty of Information Technology  
and Electrical Engineering

Department of Computer Science

Midterm examination paper for **TD4165 Programming Languages**

Academic contact during examination: *Øystein Nytrø*

Phone: +47 91897606, Email: *nytroe@ntnu.no*

Examination date: **October 9, 2019**

Examination time (from-to): **17.00-18.00**

Permitted examination support material: **Code E: None**

This course has only an english version of the exam.

This examination has 16 tasks. All tasks have the same weight (1/16).

Wrong answers are not scored negatively.

There is an ungraded text-entry at the end of the exam that you can use for comment.

Students will find the examination results in Studentweb after grading has been completed. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer.

1

Select the correct completion of the sentence: 'For a programming language, syntactic sugar ...'

**Select one alternative:**

is an addition to the language that extends the expressiveness.

is a shortcut notation that makes a program more compact or readable.

is a shorthand for defining the syntax of the language using context-free grammars.

is a popular term for operators and built-in functions for processing text and string data.

is what you remove when you transform a parse tree for an expression to the corresponding syntax tree.

Maximum marks: 1

2

Select the correct completion of the sentence: 'A lexical analyzer ...'

Select one alternative:

- will read program code as strings of characters.
- will traverse the syntax tree and transform lexemes to tokens.
- will produce a syntax tree.
- will read a parse tree.
- will produce a stream of tokens.

Maximum marks: 1

3

Consider the following grammar for statements in a language similar to Oz.

```
<s> ::= skip
      | <s> <s>
      | local <x> in <s>
      | <x> = <x>
      | <x> = <v>
      | if <x> then <s> else <s>
```

<s> is a sentence, and is also the start symbol.  
<x> is an identifier, as in Oz.  
<v> is a value expression, as in Oz.

Which alternative is not a valid sentence in the grammar?

Select one alternative:

- local A in if A then A=A else A=A
- local A in if B then B=A else A=B
- local A in local B in A=A B=B if A then A=B else B=A
- local A in local B in A=B B=A if A then B else A
- All alternatives are syntactically valid.

Maximum marks: 1

4

Consider the following grammar for statements in a language similar to Oz.

```
<s> ::= skip
      | <s> <s>
      | local <x> in <s>
      | <x> = <x>
      | <x> = <v>
      | if <x> then <s> else <s>
```

<s> is a sentence, and is also the start symbol.  
<x> is an identifier, as in Oz.  
<v> is a value expression, as in Oz.

The grammar is ...  
**Select one alternative:**

- neither ambiguous nor context free.
- context free but not ambiguous.
- ambiguous but not context free.
- both ambiguous and context free.
- None of the other alternatives.

Maximum marks: 1

5

Consider the following state in the execution of a program in the declarative kernel language on the abstract machine (variable names are given as v1, v2, ...

```
( [ ( {A A}, {A->v1} ) ], {v1->(proc{$ A} {A A} end, {})} )
```

What will be the next state in the execution?  
**Select one alternative:**

- There is no next state, because of infinite recursion.
- (nil, {v1->(proc{\$ A} {A A} end, {})} )
- ([{A A},{A->v1}],{v1->(proc{\$A}{A A}end,{})})
- ([{A A},{A->v1}]{A A},{A->v1}],{v1->(proc{\$A}{A A}end,{})})
- None of the state alternatives

Maximum marks: 1

6 If you consult (feed, run) the following code in Mozart:

```
local Y T Z in
try
local X=bar(baz) Y=boom Z T in
  try
    raise X end
    Z = 1
    catch bar(Z) then {Browse T#Z} end
end
catch bar(Y) then {Browse T#Z} end end
```

What would the browser show?

Select one alternative:

- baz#baz
- None of the other alternatives.
- baz#1, \_#1
- \_#1
- \_#baz

Maximum marks: 1

7 Oz has

Select one alternative:

- Strong scoping
- Dynamic typing
- Weak typing
- Weak scoping
- Typical scoping

Maximum marks: 1

8

Complete with the correct alternative: 'Dataflow computation ...'

**Select one alternative:**

is the same as lazy evaluation.

implies lazy evaluation.

requires exceptions.

may delay unification.

cannot delay procedure invocation.

Maximum marks: 1

9

Complete with the correct alternative 'A closure in Oz is...'

**Select one alternative:**

the final explicit catch sentence in the program. (The system will always catch uncaught exceptions, but is not considered to be the closure.)

the result of binding the unbound part of a tuple-representation of a difference-list with 'nil', thus making an ordinary list.

the complete environment for a procedure or function definition.

a representation of the environmental mapping of free variables and a procedure body.

not a term related to Oz semantics.

Maximum marks: 1

10

Given the Oz values

1. [1 2 3]
2. 1|2|3|nil
3. 'l'(1 'l'(2 'l'(3 nil)))

Which represent the same data structure?

**Select one alternative:**

None.

1 and 2.

2 and 3.

1 and 3.

All.

Maximum marks: 1

11

Consider the following two definitions:

1. fun {Reverse List}  
 fun {Reverse List Accumulator}  
 Head|Tail = List in  
 {Reverse Tail Head|Accumulator}  
 end in {Reverse List nil}  
end
2. fun {Reverse List}  
 case List of Head|Tail  
 then {Reverse Tail}|Head  
 else nil end

Which (one) of the following alternatives are true (Here ‘constant’ and ‘linear’ is relative to the length of the input list.)

Select one alternative:

- 1 defines constant-stack computation, but 2 defines linear-stack computation.
- Both 1 and 2 define linear-stack computation.
- Both 1 and 2 define constant-stack computation.
- 2 defines constant-stack computation, but 1 defines linear-stack computation.
- The answer is compiler-dependent.

Maximum marks: 1

12

Consider the following execution state of the abstract machine in the computational model of the declarative kernel language:

( [ ( {X Y R}, {X->v1, Y->v2, Z->v3, R->v4} ) ], {v1->( proc {\$ Y R} R=Y+Z end, {Z->v5} )  
v2->5, v3->7, v4, v5->3} )

What will be the next execution state?

Select one alternative:

- There is no next execution state (computation halts)
- ( [ ], {v1->( proc {\$ Y R} R=Y+Z end, {Z->v5} ),v2->5, v3->7, v4->8, v5->3} )
- ( [ ], {v1->( proc {\$ Y R} R=Y+Z end, {Z->v5} ) v2->5, v3->7, v4->10, v5->3})
- [ ( R=Y+Z, {Y->v2, Z->v5, R->v4} ) ], {v1->( proc {\$ Y R} R=Y+Z end, {Z->v5} ) v2->5, v3->7, v4, v5->3} )
- None of the other state alternatives

Maximum marks: 1

13

```
In case you do not remember the definitions:
fun {FoldR X F S}
  case X of E|Xr then {F E {FoldR Xr F S}} else S
end
end
fun {FoldL X F Ac}
  case X of E|Xr then {FoldL Xr F {F Ac E}} else Ac
end
end
```

```
Given two functions
declare fun {G1 L R} L|R end
declare fun {G2 L R} R|L end
```

Which of the following calls will give the result [3 2 1]?

Select one alternative:

- ☐ {FoldL [1 2 3] G1 nil}
- ☐ {FoldL [1 2 3] G2 nil}
- ☐ {FoldR [1 2 3] G1 nil}
- ☐ {FoldR [1 2 3] G2 nil}
- ☐ None of the alternatives.

Maximum marks: 1

14

What is the result of running

{Browse {Bar [f o o | X]#X Y#Y}}

given the following definition of Bar:

```
fun {Bar X Y}
  (A#B)#(C#D) = X#Y
in
  B=C
  A#D
end
```

Select one alternative:

- [f o o]
- [f o o | \_]#\_
- No reaction (it will suspend)
- Unification error during runtime.
- Error from compiler or interpreter.

Maximum marks: 1

15

Which function defines an append between lists (plain or difference-lists)?

Select one alternative:

- fun {Append L1 L2} case L1 of nil then L2 [] H|T then H|{Append T L1} end end
- fun {Append L1 L2} case L2 of nil then L1 [] H|T then {Append T L1}|H end end
- fun {Append L1 L2} case L1 of nil then L2 [] H|T then H|{Append L1 L2} else raise notAListException end end end
- fun {Append L1 L2} case L1#L2 of nil#L then L [] L#nil then L [] (H|T)#L then H|{Append T L} end end
- None of the definitions.

Maximum marks: 1



16

If you feed

```
declare X Y = en(X Y) {Browse Y}
```

to Mozart, what will happen?

**Select one alternative:**

None of the other alternatives.

It will complain that Y is not introduced.

It will show 'en(X Y)'

It will show 'en(\_\_)'

It will show something like 'en(\_ en(\_ en(\_ en( , , , )))')

Maximum marks: 1

17

### New Question

This section is not marked or graded, leave comments or clarifications if needed here.

Maximum marks: 0