CISC/CMPE 422/835

Practice Midterm

(to provide a sense of the *kinds* of questions the real midterm is likely to contain)

Fall 2022

Instructor: Juergen Dingel

Student id (very legibly, please!): ____________________

Name (optional): ____________________

Shred permission (optional): ____________________

To allow the instructor to shred this test if you do not pick it up by the end of this term, please put your signature on the line above.

Instructions:

- The exam begins on page 2. Do not turn over this page until told to do so.
- You have 50 minutes to complete the exam.
- You may use one 8.5x11" data sheet. No other notes or books or calculators are allowed.
- Answer each question in the space provided. If you answer a question on a different page, you must indicate where your answer is.
- Use a pen or fine marker with blue or black ink. If you use erasable ink or pencil, or write in red, pink, or purple, your test will not be considered for re-marking.
- If a question is unclear, ask for clarification. If it is still unclear, make whatever assumptions you think are necessary and write them down. Not all assumptions are equally valid.

Copyright note:

For marking use only:

Q1	/7
Q2	/16
Q3	/16
Q4	/12
Total	/51

Id: ______________________

Question 1 (7 points)

Consider the following partial Alloy specification `Blocks` of blocks and variables as typically found in block-structured programming languages:

```
module Blocks

sig Var {}
```

```
sig Block {
   contains : set Block,
   declares : some Var,
   uses : set Var }
```

a) (3 points) In the space below, draw the graphical representation (i.e., meta model) of `Blocks` including all multiplicity constraints that `Blocks` contains (and not more).

b) (4 points) Consider the predicate

```
pred P[] {
   #contains = 2
}
```

Draw an instance of `Blocks` that satisfies all the constraints in `Blocks` and `P`, that is, draw an instance that could be created by Alloy in response to executing the command '`run P for 3`' on `Blocks`.

Id: ____________________

Question 2 (16 points)

For your convenience, the Alloy specification from the previous page is repeated here.

```
module Blocks

sig Var {}
```

```
sig Block {
   contains : set Block,
   declares : some Var,
   uses : set Var }
```

Using the Alloy specification above, express each of the following properties formally in Alloy. We say that a block `b1` is an *ancestor* of another block `b2` iff there is a non-empty sequence of `contains` edges from `b1` to `b2`.

a) (4 points) *"All blocks use all the variables they declare"*

b) (4 points) *"If a block* `b` *uses a variable, that variable is declared either in* `b` *or one of* `b`*'s ancestors"*

c) (4 points) *"All variables are declared in at most one block"*

d) (4 points) *"There is exactly one block that is not contained in any other block"*

Id: ______________________

Question 3 (16 points)

For your convenience, the Alloy specification from the previous page is repeated here.

```
module Blocks

sig Var {}
```

```
sig Block {
   contains : set Block,
   declares : some Var,
   uses : set Var }
```

For each question below, complete the function definition such that an invocation of the function would return the set of values described. Note that some functions take parameters. As before, we say that a block `b1` is an *ancestor* of another block `b2` iff there is a non-empty sequence of `contains` edges from `b1` to `b2`.

a) (4 points) *"The set of all blocks that contain at least one block that uses no variable"*

```
fun fa[] : set Block {

}
```

b) (4 points) *"The set of all variables that are declared in block* `b` *or one of its ancestors"*

```
fun fb[b:Block] : set Var {

}
```

c) (4 points) *"The set of all blocks such that all blocks that they contain all use the same variable and only that variable"*

```
fun fc[] : set Block {

}
```

d) (4 points) *"The set of all blocks that contain exactly two blocks that declare no variables"*

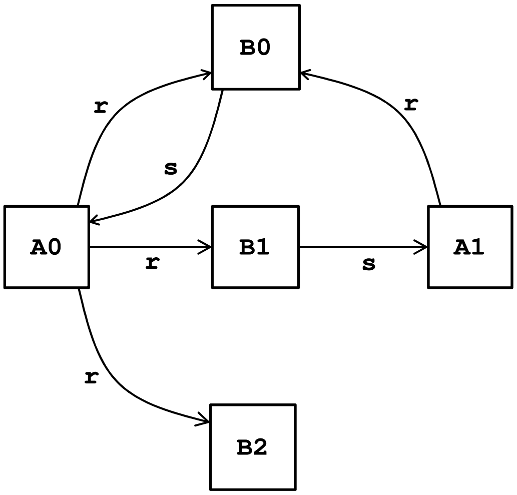
```
fun fd[] : set Block {
```

Id: ______________________

Question 4 (12 points)

Consider the Alloy specification `Test` on the left and the instance satisfying all constraints in `Test` produced by the Alloy analyzer on the right.

```
module Test

sig A {
  r : set B }

sig B {
  s : set A }




}

run {some A && some B} for 3
```



For each of the following Alloy expressions and formulas, determine which value the expression or formula evaluates to in the instance on the right and write down that value.

a) `r.(A.r)` evaluates to:

b) `some b:B | all a:b.s | a.r=b` evaluates to:

c) `B1->B1 in ^(s.r)` evaluates to:

d) `A->A - r.s` evaluates to:

e) `all b:B | some a:A | a.r=b` evaluates to:

f) `{a:A | a.r.s in A0}` evaluates to:

Id: ______________________

Scratch sheet: