CSE216
Foundations of Computer Science

Instructor: Zhoulai Fu

State University of New York, Korea

Part 1. Regular Expressions (20)

1. Give a regular expression to match strings over {a, b} that has aabb as a substring.

2. Write a regular expression to match valid hexadecimal numbers, which start with 0x followed by one or more hexadecimal digits. Hexadecimal digits can be any combination of the numbers 0-9 and the letters A-F (both uppercase and lowercase).

```
Example valid inputs: 0x3A, 0xA2F, 0xFF, 0xaF

Example invalid inputs: 3A, 0xG5, 0x
```

   Your task is to construct the regular expression pattern to accurately match valid hexadecimal numbers in the given format.

3. Give a regular expression for positive integers, not including ones with leading 0s such as 07 or 007. Examples: 12345, 7070.

4. Give a regular expression for the language over {0,1} of strings that contains at least two occurrences of 001. Examples: 001001, 001100100.

Part 2. Context-free Grammar (20)

Consider the following grammar where {a, b} are terminals, {S, A, B} are non-terminals, and S is the starting symbol. Find the languages generated by each grammar. Example: If you are asked to find the language for grammar "S → AB, A → a | aa, B → b", your answer will be {ab,aab}.

1. S → AB, A → ab, B → bb

2. S → AB, A → a | aa, B → b | bb

Make a context-free grammar generating

3. {$a^n b^{3n}$: n ≥ 0}. Examples: abbb, aabbbbbb, empty string.

4. Which of the following context-free grammars generates the language over {a,b} consisting of the strings that read the same forward and backward, e.g. aba, baab, abaaba, b?

A. $S \rightarrow aSa|bSb|a|b|\epsilon$

B. $S \rightarrow aS|bS|a|b$

C. $S \rightarrow aAa|bBb|\epsilon$, $A \rightarrow a|b$, $B \rightarrow a|b$

D. $S \rightarrow Aa|Bb$, $A \rightarrow aSa|bB|\epsilon$, $B \rightarrow bSb|aA|\epsilon$

Part 3. Lambda Calculus (40)

Apply β-reduction to the following λ-terms as much as possible

1. (λz.z) (λy.y)
2. (λz.z) (λy.y y)
3. (λz.z) (λy.y y) (λx.x a)
4. (λz.z) (λz.z z) (λz.z y)
5. (λx.λy.x y y) (λa.a) b
6. (λx.λy.x y y) (λy.y) y
7. (λx.x x) (λy.y x) z
8. (λx. (λy. x y) y) z
9. ((λx.x x) λy.y) (λy.y)
10. ((λx. λy.x y)λy.y) w

Part 4. Ocaml (20)

Determine the types of the following expressions:

1. 2.0 +. 3.0

2. let x = 3 in x + 5

3. let b = false in if b then "hello" else "hi"

4. print_endline “hello”

5. print_endline

Determine the types of the following expressions. If an error will arise, reply "error".

6. let f x = x * x in f 2

7. let average x y = (x +. y) /. 2.0 in average 4

Determine the types of the functions defined below:

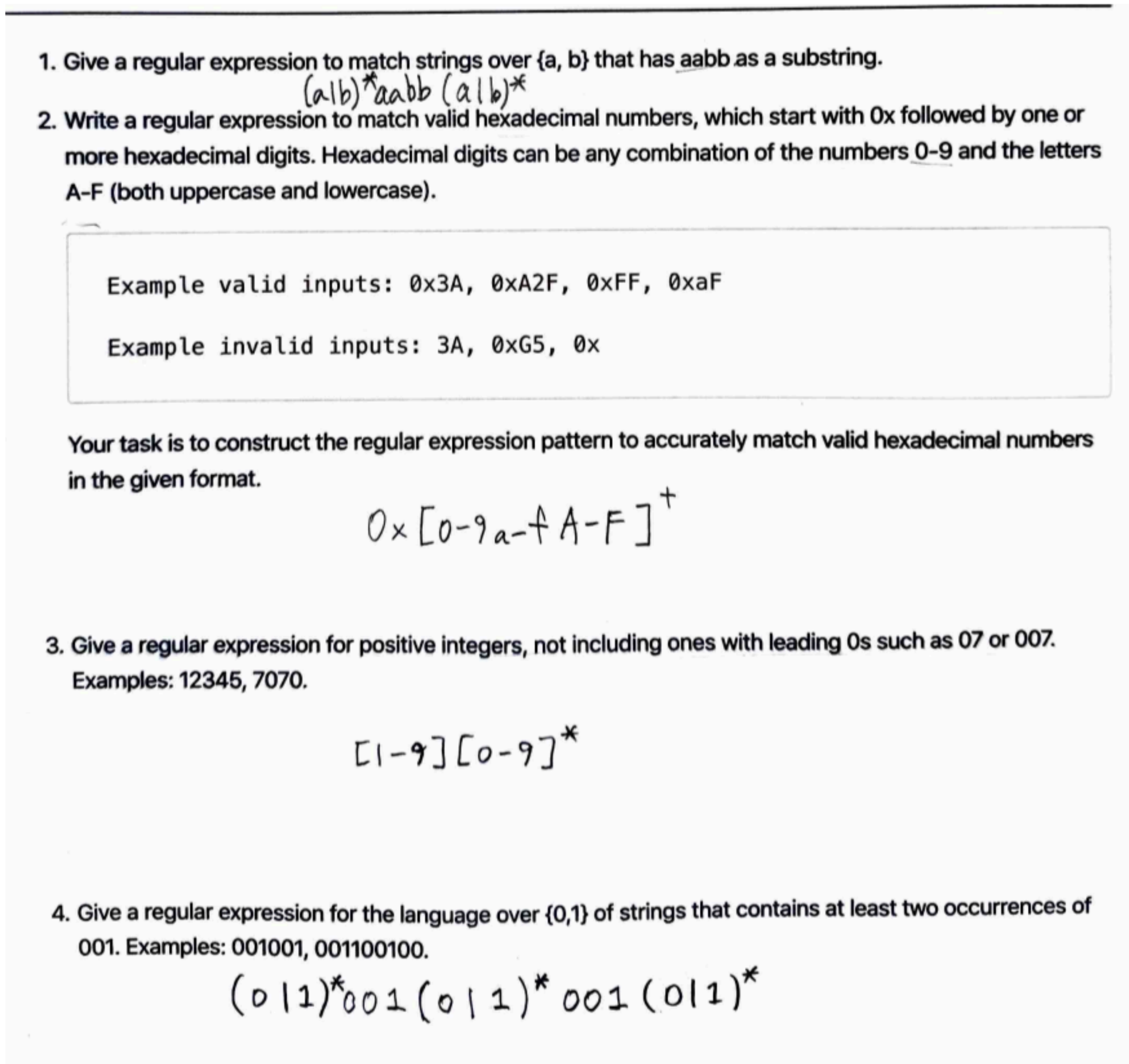
8. let f x = x * x

9. let g x y = if x then y else 2 * y

Evaluate the expression below. What is the value of this expression?

10. let x= 3 in (let y= x+2 in y) - (let z = x * x * x in z)

Solution



1. Give a regular expression to match strings over {a, b} that has aabb as a substring.

$(a|b)^*aabb\,(a|b)^*$

2. Write a regular expression to match valid hexadecimal numbers, which start with 0x followed by one or more hexadecimal digits. Hexadecimal digits can be any combination of the numbers 0-9 and the letters A-F (both uppercase and lowercase).

```
Example valid inputs: 0x3A, 0xA2F, 0xFF, 0xaF

Example invalid inputs: 3A, 0xG5, 0x
```

Your task is to construct the regular expression pattern to accurately match valid hexadecimal numbers in the given format.

$0x[0-9a-fA-F]^+$

3. Give a regular expression for positive integers, not including ones with leading 0s such as 07 or 007. Examples: 12345, 7070.

$[1-9][0-9]^*$

4. Give a regular expression for the language over {0,1} of strings that contains at least two occurrences of 001. Examples: 001001, 001100100.

$(0|1)^*001\,(0|1)^*\,001\,(0|1)^*$

Part 2. Context-free Grammar (20)

Consider the following grammar where {a, b} are terminals, {S, A, B} are non-terminals, and S is the starting symbol. Find the languages generated by each grammar. Example: If you are asked to find the language for grammar "S → AB, A → a | aa, B → b", your answer will be {ab,aab}.

1. S → AB, A → ab, B → bb

{abbb}

2. S → AB, A → a | aa, B → b | bb

{ab , abb, aab , aabb}

Make a context-free grammar generating

3. {$a^n b^{3n}$: n ≥ 0}. Examples: abbb, aabbbbbb, empty string.

$S \rightarrow aSbbb \mid \varepsilon$

4. Which of the following context-free grammars generates the language over {a,b} consisting of the strings that read the same forward and backward, e.g. aba, baab, abaaba, b?

[✓ A.] $S \rightarrow aSa|bSb|a|b|\epsilon$

B. $S \rightarrow aS|bS|a|b$

C. $S \rightarrow aAa|bBb|\epsilon,\ A \rightarrow a|b,\ B \rightarrow a|b$

D. $S \rightarrow Aa|Bb,\ A \rightarrow aSa|bB|\epsilon,\ B \rightarrow bSb|aA|\epsilon$

A

Part 3. Lambda Calculus (40)

Apply β-reduction to the following λ-terms as much as possible

1. (λz.z) (λy.y)

   λy.y

2. (λz.z) (λy.y y)

   λy.yy

3. (λz.z) (λy.y y) (λx.x a)

   aa

4. (λz.z) (λz.z z) (λz.z y)

   yy

5. (λx.λy.x y y) (λa.a) b

   bb

6. (λx.λy.x y y) (λy.y) y

   yy

7. (λx.x x) (λy.y x) z

   xxz

8. (λx. (λy. x y) y) z

   zy

9. ((λx.x x) λy.y) (λy.y)

   λy.y

10. ((λx. λy.x y)λy.y) w

    w

Part 4. Ocaml (20)

Determine the types of the following expressions:

1. 2.0 +. 3.0

float

2. let x = 3 in x + 5

int

3. let b = false in if b then "hello" else "hi"

string

4. print_endline “hello”

unit

5. print_endline

string → unit

Determine the types of the following expressions. If an error will arise, reply "error".

6. let f x = x * x in f 2

int

7. let average x y = (x +. y) /. 2.0 in average 4

error

Determine the types of the functions defined below:

8. let f x = x * x

int → int

9. let g x y = if x then y else 2 * y

bool → int → int

Evaluate the expression below. What is the value of this expression?

10. let x= 3 in (let y= x+2 in y) - (let z = x * x * x in z)

-22

Homework week 07

Exercise 1. Toplevel (2)

If you have finished Exercise 0, you can continue using Toplevel from your machine in this exercise.

If you have not finished Exercise 0, you can use an online Ocaml IDE from the site:

https://try.ocamlpro.com/

Find the "#" on your Toplevel. That is a prompt, which is not part of the Ocaml language. Try to type `42;;` in the Toplevel following "#". Your Toplevel may or may not generate double-semicolon `;;` for you.

Search on the Internet. Write in English about your understanding on the role of the double-semicolon `;;` in Ocaml. This is an open question.

Exercise 2. Expressions (54)

Build-in-types

Consider the following expressions:

1. 42
2. 42.1
3. "hello"
4. true
5. ()
6. 'a'

Type each expression at Toplevel followed by `;;` if it is not automatically generated. Check the evaluation result of the expression and the type, and fill in the table below.

Expression	type	evaluation

Functions

Consider the following expressions:

7. print_endline
8. fun x -> x + 1
9. fun x y -> x+y
10. (+)
11. (+.)
12. (^)

Type each expression at Toplevel followed by `;;` if it is not automatically generated. Check the evaluation result of the expression and the type, and fill in the table below.

Expression	type	evaluation

Function Evaluations

Consider the following expressions:

13. (fun x -> x + 1) 3
14. (fun x y -> x+y) 2 3
15. (fun x y -> x+y) 2
16. let x = 3 in x + 1
17. let y = 3 in let x = 2 in x + y
18. print_endline "hi"

Type each expression at Toplevel followed by `;;` if it is not automatically generated. Check the evaluation result of the expression and the type, and fill in the table below.

Expression	type	evaluation

Exercise 3. Definitions (24)

Consider the following definitions.

1. let x = 3
2. let s = "hello"
3. let a = ()
4. let b = true
5. let f = fun x -> x + 1
6. let f x = x + 1
7. let f = fun x y -> x + y
8. let f x y = x + y

Type each definition at Toplevel, followed by `;;` if it is not automatically generated. Check which variable is defined, of what type, and evaluation result. Fill in the table below.

Definition	defined variables	type	evaluation

Exercise 4. Put it all together (20)

1. Define a function avg in Ocaml that computes the average of two float.

2. Complete the gcd function of type int -> int -> int which calculates the gcd using the Euclidean method, that is:

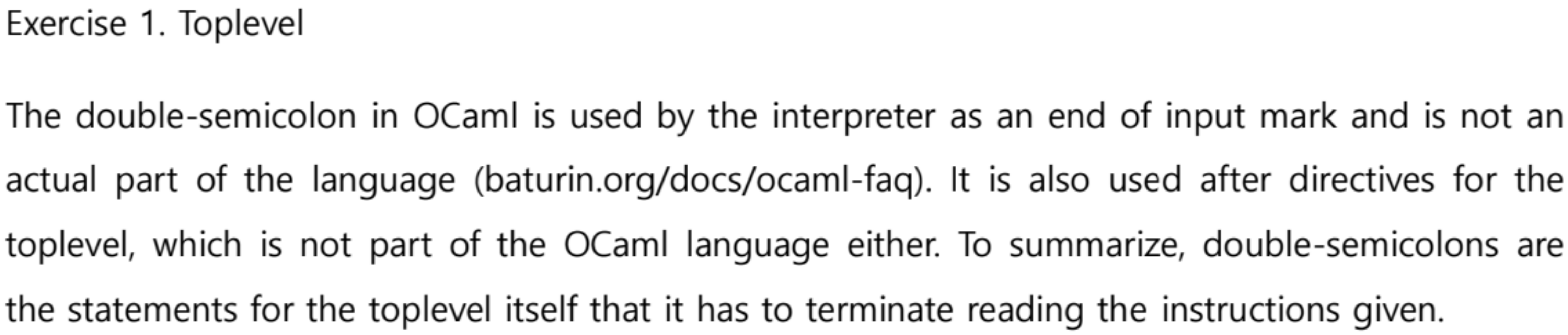
- gcd(u, 0) = u
- gcd(u, v) = gcd(v, u mod v) otherwise"

```
let rec gcd u v =
  0 (* To complete *)






(* some tests *)
let () = Printf.printf "%d\n" (gcd 8 12)
let () = Printf.printf "%d\n" (gcd 48 18)
```

Once you finish, copy the code to Toplevel and check out the result.

Feel free to search online. You may not understand everything in the code. That is fine; we will learn things incrementally.

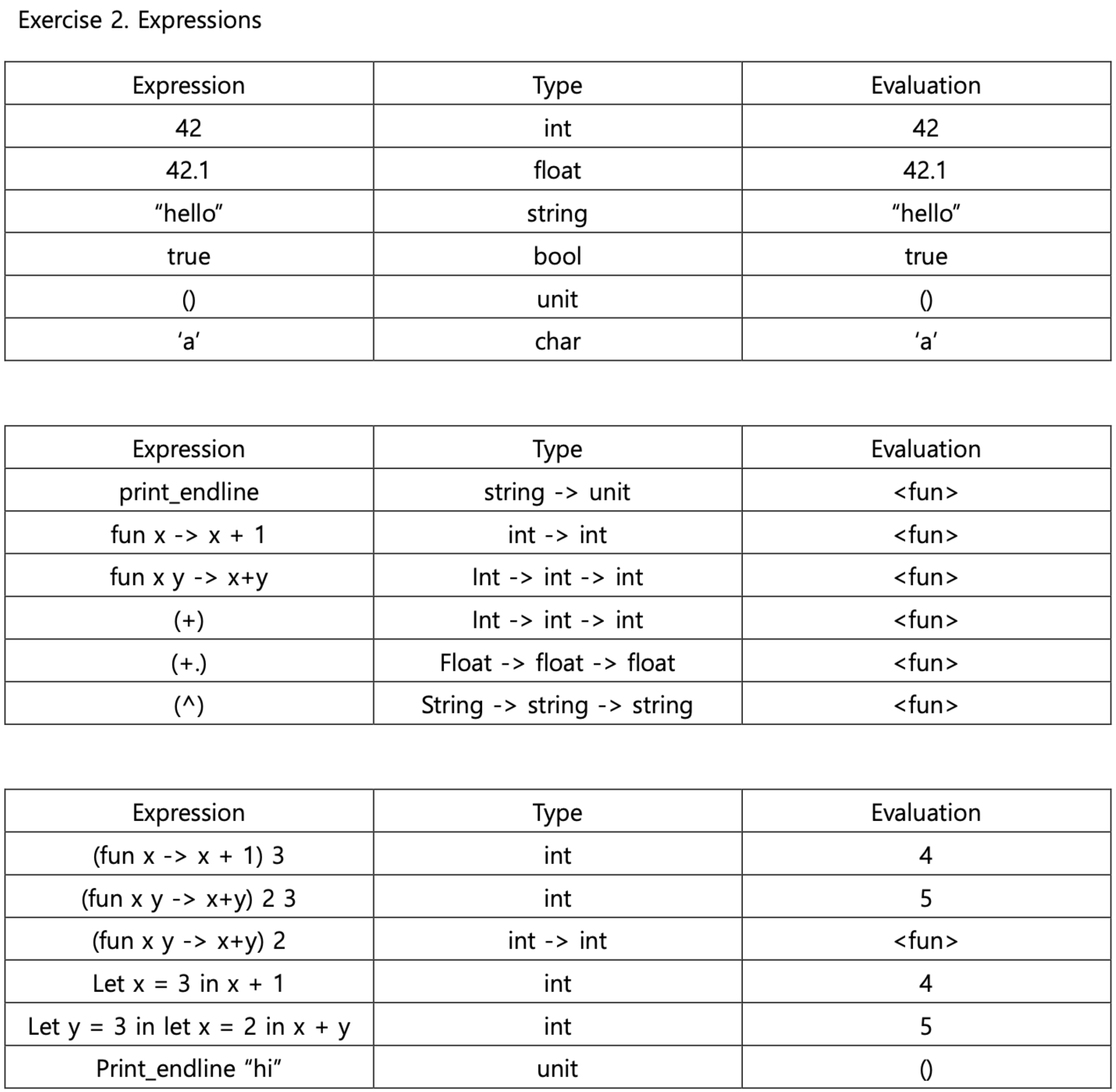
Solution



Exercise 1. Toplevel

The double-semicolon in OCaml is used by the interpreter as an end of input mark and is not an actual part of the language (baturin.org/docs/ocaml-faq). It is also used after directives for the toplevel, which is not part of the OCaml language either. To summarize, double-semicolons are the statements for the toplevel itself that it has to terminate reading the instructions given.

Solution



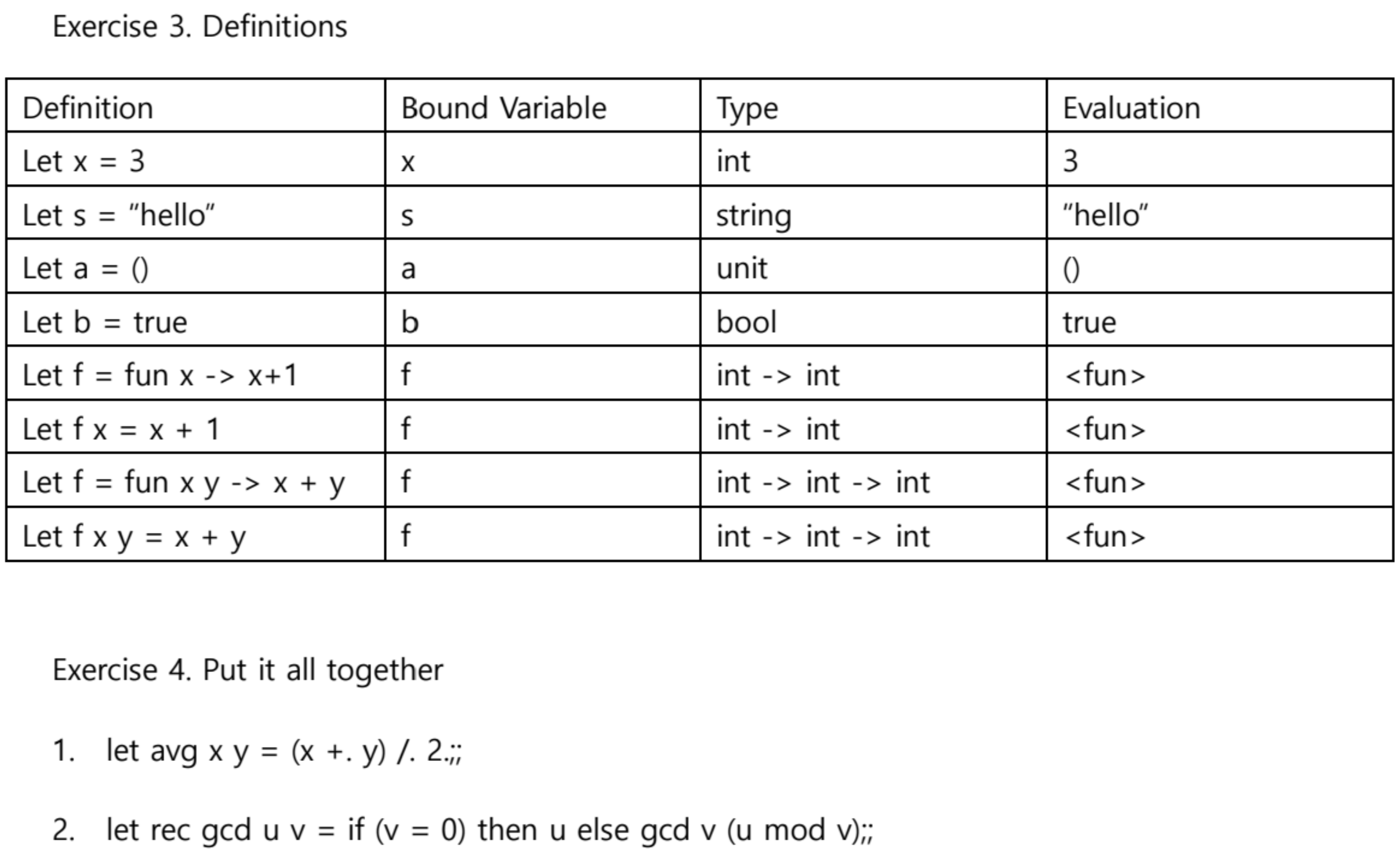
Exercise 2. Expressions

Expression	Type	Evaluation
42	int	42
42.1	float	42.1
“hello”	string	“hello”
true	bool	true
()	unit	()
‘a’	char	‘a’

Expression	Type	Evaluation
print_endline	string -> unit	<fun>
fun x -> x + 1	int -> int	<fun>
fun x y -> x+y	Int -> int -> int	<fun>
(+)	Int -> int -> int	<fun>
(+.)	Float -> float -> float	<fun>
(^)	String -> string -> string	<fun>

Expression	Type	Evaluation
(fun x -> x + 1) 3	int	4
(fun x y -> x+y) 2 3	int	5
(fun x y -> x+y) 2	int -> int	<fun>
Let x = 3 in x + 1	int	4
Let y = 3 in let x = 2 in x + y	int	5
Print_endline “hi”	unit	()

Solution



Exercise 3. Definitions

Definition	Bound Variable	Type	Evaluation
Let x = 3	x	int	3
Let s = “hello”	s	string	“hello”
Let a = ()	a	unit	()
Let b = true	b	bool	true
Let f = fun x -> x+1	f	int -> int	<fun>
Let f x = x + 1	f	int -> int	<fun>
Let f = fun x y -> x + y	f	int -> int -> int	<fun>
Let f x y = x + y	f	int -> int -> int	<fun>

Exercise 4. Put it all together

1. let avg x y = (x +. y) /. 2.;;

2. let rec gcd u v = if (v = 0) then u else gcd v (u mod v);;