CS113 Lab 7 - SML

SML

```
PolyML.print_depth 100;
infix --;
fun (i -- n) = if (i > n)
    then [ ]
    else i::(i + 1 -- n);

fun elementOf(a, [ ]) = false |
    elementOf(a, x::xs) = if (a = x)
    then true
    else elementOf(a, xs);

fun insert(x, [ ]) = [x] |
    insert(x, ys) = x::ys;
```

SML

```
fun remove(x, [ ]) = [ ] |
    remove(x, y::ys) = if (x = y)
    then remove(x, ys)
    else y::remove(x, ys);

fun subset([ ], _) = true |
    subset(x::xs, ys) = if not (elementOf(x, ys))
    then false
    else subset(xs, ys);

fun subsetEqual(A, B) =
    subset(A, B) andalso subset(B, A);

fun properSubset(A, B) =
    subset(A, B) andalso not(subset(B, A));
```

SML

```
fun relativeComplement(xs, [ ]) = [ ] |
    relativeComplement([ ], ys) = ys |
    relativeComplement(x::xs, ys) = if (elementOf(x, ys))
    then relativeComplement(xs, remove(x, ys))
    else relativeComplement(xs, ys);

fun union(xs, [ ]) = xs |
    union([ ], ys) = ys |
    union(x::xs, ys) = if not (elementOf(x, ys))
    then x::union(xs, ys)
    else union(xs, ys);
```

SML

```
fun intersection([ ], _) = [ ] |
    intersection(_, [ ]) = [ ] |
    intersection(x::xs, ys) = if (elementOf(x, ys))
    then x::intersection(xs, ys)
    else intersection(xs, ys);

fun disjointSets(A, B) = if (intersection(A, B) = [ ])
    then true
    else false;

fun symmetricDifference(xs, [ ]) = xs |
    symmetricDifference([ ], ys) = ys |
    symmetricDifference(x::xs, ys) = if (elementOf(x, ys))
    then symmetricDifference(xs, remove(x, ys))
    else x::symmetricDifference(xs, ys);
```

SML

```
fun cartesianProduct([ ], _) = [ ] |
    cartesianProduct(x::xs, ys) =
    (map (fn v => (x, v)) ys) @ cartesianProduct(xs, ys);

fun powerSet([ ]) = [ [ ] ] |
    powerSet(x::xs) =
    powerSet(xs) @ (map (fn S => x::S) (powerSet(xs)));

fun domain([ ]) = [ ] |
    domain((a, b)::ps) = if elementOf(a, domain(ps))
                         then domain(ps)
                         else a::domain(ps);
fun range([ ]) = [ ] |
    range((a, b)::ps) = if elementOf(b, range(ps))
                        then range(ps)
                        else b::range(ps);
```

SML

```
fun divides(a, b) = b mod a = 0;

fun noDuplicates([ ]) = true |
    noDuplicates(x::xs) = (not(elementOf(x, xs)))
                          andalso noDuplicates(xs);

fun isFunction(R, A) =
    let
        fun f([ ]) = [ ] |
            f((a, b)::ps) = a::f(ps);
    in
        subsetEqual(f(R), A) andalso noDuplicates(f(R)) end;

fun inverse([ ]) = [ ] |
    inverse((a, b)::ps) = (b, a)::inverse(ps);
```

SML

```
fun composition([ ], ss) = [ ] |
    composition(r::rs, ss) =
    let
        fun f((a, b), [ ]) = [ ] |
            f((a, b), (c, d)::ss) = if (b = c)
            then (a, d)::f((a, b), ss)
            else f((a, b), ss);
    in
        f(r, ss)@composition(rs, ss) end;

fun lessEqu(a, b) = a <= b;
```

CS113 Lab 7 - Example 15.5

```
SML
val A = (2 -- 4);
val B = (3 -- 7);

"Example 15.5(a)";
val r = List.filter divides(cartesianProduct(A, B));
domain(r);
range(r);

val A = (1 -- 4);

"Example 15.5(b)";
val r = List.filter lessEqu(cartesianProduct(A, A));
domain(r);
range(r);
```

(a) Let A = {2,3,4} and B = {3,4,5,6,7}.
Define the relation R by aRb if and only if $adividesb$.
Find, R, Dom(R), Range(R).

- R = {(2, 4),(2, 6),(3, 3),(3, 6),(4, 4)}
- Dom(R) = {2, 3, 4}
- Range(R) = {3, 4, 6}

(b) Let A = {1,2,3,4}
Define the relation R by aRb if and donly if $a \leq b$.
Find, R, Dom(R), Range(R).

- R = {(1, 1),(1, 2),(1, 3),(1, 4),(2, 2), (2, 3),(2, 4),(3, 3),(3, 4),(4, 4)}
- Dom(R) = {2, 3, 4}
- Range(R) = {3, 4, 6}

```
it = "Example 15.5(a)": string
r = [(2, 4), (2, 6), (3, 3), (3, 6), (4, 4)]:
it = [2, 3, 4]: int list
it = [3, 6, 4]: int list
```

```
val it = "Example 15.5(b)": string
val r =
 [(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4),
  (4, 4)]: (int * int) list
val it = [1, 2, 3, 4]: int list
val it = [1, 2, 3, 4]: int list
```

CS113 Lab 7 - Example 15.6

```
SML
val A = [1,2,3];
val B = ["a", "b", "c"];

"Example 15.6(a)";
val R = [(1, "a"), (2, "b"), (3, "a")];
isFunction(R, A);
range(R);

"Example 15.6(b)";
val R = [(1, "a"), (2, "b"), (3, "c"), (1, "b")];
isFunction(R, A);
```

(a) Show that the relation $f = (1, a), (2, b), (3, a)$ defines a function from $A = \{1, 2, 3\}$ to $B = \{a, b, c\}$.

- Note that each element of A has exactly one image. Hence, f is a function with domain A and range Range$(f) = \{a, b\}$
- Dom(R) = {2, 3, 4}
- Range(R) = {3, 4, 6}

(b) Show that the relation $f = \{(1, a), (2, b), (3, c), (1, b)\}$ does not define a function from A to B.

- The relation f does not define a function since the element 1 has two images, namely a and b
- Dom(R) = {2, 3, 4}
- Range(R) = {3, 4, 6}

```
it = "15.6a": string
R = [(1, "a"), (2, "b"), (3, "a")]:
it = true: bool
it = [1, 2, 3]: int list
it = ["b", "a"]: string list
```

```
it = "15.6b": string
R = [(1, "a"), (2, "b"), (3, "c"), (1, "b")]:
it = false: bool
```

CS113 Lab 7 - Example 15.8

```
SML
val R = [(1, "y"), (1, "z"), (3, "y")];
val A = [1, 2, 3];
val B = ["x", "y", "z"];

"Example 15.8(a)";
inverse(R);

"Example 15.8(b)";
subsetEqual(R, inverse(inverse(R)));
```

(a) Find R^{-1}.

- R^{-1} = {(y, 1),(z, 1),(y, 3)}

(b) Compare $(R^{-1})^{-1} and R$.

- $(R^{-1})^{-1} = R$

```
it = "Example 15.8(a)": string
it = [("y", 1), ("z", 1), ("y", 3)]:
```

```
it = "Example 15.8(b)":
it = true: bool
```

CS113 Lab 7 - Example 15.9

```
SML
val A = [1, 2, 4];
val B = [2, 6, 8, 10];
fun f(a, b) = b - 4 = a;

"Example 15.9";
val R = List.filter divides(cartesianProduct(A, B));
val S = List.filter f(cartesianProduct(A, B));
union(R, S);
intersection(R, S);
```

- Given the following two relations from A = {1, 2, 4} to B = {2, 6, 8, 10} :
  aRb if and only if $a|b$.
  aSb if and only if $b - 4 = a$.
  List the elements of $R, S, R \cup S$, and $R \cap S$.

$$R = \{(1,2),(1,6),(1,8),(1,10),(2,2),(2,6),(2,8),(2,10),(4,8)\}$$
$$S = \{(2,6),(4,8)\}$$
$$R \cup S = R$$
$$R \cap S = S$$

```
val it = "Example 15.9": string
val R =
 [(1, 2), (1, 6), (1, 8), (1, 10), (2, 2), (2, 6), (2, 8), (2, 10), (4, 8)]:
 (int * int) list
val S = [(2, 6), (4, 8)]: (int * int) list
val it =
 [(1, 2), (1, 6), (1, 8), (1, 10), (2, 2), (2, 8), (2, 10), (2, 6), (4, 8)]:
 (int * int) list
val it = [(2, 6), (4, 8)]: (int * int) list
```

SML

```
val R = [(1, 2), (1, 6), (2, 4), (3, 4), (3, 6), (3, 8)];
val S = [(2, "u"), (4, "s"), (4, "t"), (6, "t"), (8, "u")];

"Example 15.10";
composition(R, S);
```

- Find $S \circ R$.
  - $S \circ R = \{(1, u), (1, t), (2, s), (2, t), (3, s), (3, t), (3, u)\}$

```
val it = "Example 15.10": string
composition(R, S);
val it =
  [(1, "u"), (1, "t"), (2, "s"), (2, "t"), (3, "s"), (3, "t"), (3, "t"),
    (3, "u")]: (int * string) list
```

CS113 Lab 7 - Problem 15.5

```
SML
val A = (1 -- 3); val B = ["a", "b", "c", "d"];

"Problem 15.5(a)";
val R = [(1, "d"), (2, "d"), (3, "a")];
isFunction(R, A);
range(R);
```

```
SML
"Problem 15.5(b)";
val R = [(1, "a"), (2, "b"), (2, "c"), (3, "d")];
isFunction(R, A);
```

(a) Is the relation $f = \{(1, d), (2, d), (3, a)\}$ a function from A to B?
If so, find its range.

- f is a function from A to B since every element in A maps to exactly one element in B. The range of f is the set $\{a, d\}$.

(b) Is the relation $f = \{(1, a), (2, b), (2, c), (3, d)\}$ a function from A to B?
If so, find its range.

- f is not a function since the element 2 is related to two members of B.

```
it = "Problem 15.5(a)": string
R = [(1, "d"), (2, "d"), (3, "a")]:
it = true: bool
it = ["d", "a"]: string list
```

```
it = "Problem 15.5(b)": string
R = [(1, "a"), (2, "b"), (2, "c"), (3, "d")]:
it = false: bool
```

CS113 Lab 7 - Problem 15.7

SML

```
val R = [("a", 1), ("b", 5), ("c", 2), ("d", 1)];

"Problem 15.7";
val I = inverse(R);
```

- Find the inverse relation of $R = \{(a, 1), (b, 5), (c, 2), (d, 1)\}$.
  Is the inverse relation a function?
  - $R^{-1} = \{(1, a), (5, b), (2, c), (1, d)\}$.
    R^{-1} is not a function since 1 is related to a and d.

```
it = "Problem 15.7": string
I = [(1, "a"), (5, "b"), (2, "c"), (1, "d")]:
```

CS113 Lab 7 - Problem 15.9

SML

```
val A = ["a", "b", "c"];
val B = [1, 2];
val C = ["a", "b", "g"];
val R = [("a", 1), ("a", 2), ("b", 2), ("c", 1)];
val S = [(1, "a"), (2, "b"), (2, "g")];

"Problem 15.9";
composition(R, S);
```

```
l it =
 [("a", "a"), ("a", "b"), ("a", "g"), ("b", "b"), ("b", "g"), ("c", "a")]:
 (string * string) list
```

- Find $S \circ R$.
  - $S \circ R = \{(a,a),(a,b),(a,g),(b,b),(b,g),(c,a)\}$