CS 321: Programming Languages — Homework 2

1 Binding Identifiers

Implement the function `binding-ids`, which takes a `WAE` (see lecture code or lecture slides) and produces a set of symbols. The set should contain the symbol corresponding to every binding identifier in the given `WAE`.

To represent these sets of symbols, use lists of symbols where each symbol appears at most once, and where symbols are ordered according to the `symbol<?` comparison procedure.

For example:

```
(test (binding-ids (with 'x (num 2) (add (id 'x) (num 1))))
      '(x))
```

Hint: First, write a function that generates an unordered list of symbols with duplicates, and then write separate functions to re-order the list and then remove duplicates.

Hint: The PLAI language includes several helpful list-processing functions:

- `filter` - takes a predicate and a list, and returns a list that contains only values for which the predicate returned true. This is useful for removing a value from a list.
- `sort` - takes a list and a less-than comparison function (i.e., a function of two arguments where the result is true when the first argument is less than the second) and produces a sorted list.
- `member` - takes a value and a list, and returns `false` if the value is not in the list, and otherwise returns the portion of the list after (inclusively) the value of interest, which counts as a true value as far as conditionals are concerned.
- `remove-duplicates` - takes a list and returns a new list that has the same elements, but without duplicates.

Of course, you're also welcome to use other PLAI functions as you see fit.

As a reminder, you can find the PLAI documentation here.

Hint: You are welcome to reuse the functions you write for one part in subsequent parts.

2 Bound Identifiers

Implement the function `bound-ids`, which is like `binding-ids`, but the result list contains a symbol for each bound identifier in the given `WAE`. The result list of symbols must be sorted and have no duplicates.

For example:

```
(test (bound-ids (with 'x (num 2) (add (id 'x) (num 1))))
      '(x))
```

3 Free Identifiers

Implement the function free-ids, which is like binding-ids, but the result list contains a symbols for each free identifier in the given WAE. The result list of symbols must be sorted and have no duplicates.

For example:

```
(test (free-ids (with 'x (num 2) (add (id 'y) (num 1))))
      '(y))
```

4 Shadowed Identifiers

Implement the function shadowed-ids, which accepts a WAE and returns a list of the binding identifiers that are shadowed by another binding in the given WAE. The result list of symbols must be sorted and have no duplicates.

For example:

```
(test (shadowed-ids (with 'x (num 2) (with 'x (num 3) (add (id 'x) (num 1)))))
      '(x))
```

5 Handin Instructions

Provide definitions for binding-ids, bound-ids, free-ids, and shadowed-ids, as above.

Use the same definition for WAE we used in class.

Have the 8 rules from the Provost's website (see homework 1 for more details).

Submit your code via Canvas.

Your submission must include your test cases; submissions without test cases will get a grade of 0.