Assignment 4

**Part 1**:

Question 1:

1. {f : [T1 → T2], g : [T1 → T2], a : T1} ` (f (g a)) : T2

False.   
 receives but return .

1. {x : T1, y : T2, f : [T2 → T1]} ` (f y) : T1

True.

is of type and indeed receives . Also, returns which is exactly as needed.

1. {f : [T1 → T2]} ` (lambda (x) (f x)) : [T1 → T2]

True.

Given , the will “give” it to which will operate on it and return .

(The is somewhat redundant).

1. {f : [T1 ∗ T2 → T3]} ` (lambda (x) (f x 100)) : [T1 → T3]

False.

receives two arguments which are of types and but doesn’t have to be a number.

**2.a.**

|  |  |
| --- | --- |
| **Expression** | **Variable** |
|  |  |
|  |  |
|  |  |
| + |  |
|  |  |
| 1 |  |
| 4 |  |

|  |  |
| --- | --- |
| **Expression** | **Equation** |
|  |  |
|  |  |
|  |  |
| + |  |
| 1 |  |
| 4 |  |

|  |  |
| --- | --- |
| Equation | Substitution |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Equation | Substitution |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Equation | Substitution |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Equation | Substitution |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Equation | Substitution |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Equation | Substitution |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Equation | Substitution |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Equation | Substitution |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Equation | Substitution |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Equation | Substitution |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Equation | Substitution |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Equation | Substitution |
|  |  |

We can conclude .

**2.b**

|  |  |
| --- | --- |
| **Expression** | **Variable** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Expression** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Equation** | **Substitution** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Equation** | **Substitution** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Equation** | **Substitution** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Equation** | **Substitution** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Equation** | **Substitution** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Equation** | **Substitution** |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Equation** | **Substitution** |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Equation** | **Substitution** |
|  |  |
|  |  |
|  |  |

**Part 2**

**2.2.b**

The wrapped function returns a Promise since there are calculations that needs to be made: check if the element exists, if not calculate and insert it, and after all, return it.   
Therefore, we have several possible states, including rejection, and we must have control flow and error handling.

**Part 3**

**3.1**

1. Typing rule define:
2. For every: type environment \_Tenv,
3. variable \_x1
4. expressions \_e1 and
5. type expressions \_S1, \_U1:
6. If \_Tenv o {\_x1: \_S1} |- \_e1: \_U1,
7. Then \_Tenv |- (define \_x1 \_e1) : \_U1
8. Typing rule set!:
9. For every: type environment \_Tenv,
10. variable \_x1
11. expressions \_e1 and
12. type expressions \_S1, \_U1:
13. If \_Tenv o {\_x1: \_S1, \_U1: \_S1} |- \_e1: \_S1,
14. Then \_Tenv |- (set! \_x1 \_e1) : \_S1