CSDS 600: Type System Written Exercise Answers

Question 1: In the following C code, whic variabes of a, b, c, d, e will a compiler consider to have compatible types under structural equivalcens? Under strict name equivalence? Under loos name equivalence?

```
typedef struct {
  int year;
  int average;
} S;
typedef S T;

S a;
S b;
T c;
struct {
  int year;
  double average;
} d, e;
```

Answer: In strict name equivalence, types must have the same name or be declared with the same anonymous type. So: a and b are compatible and d and e are compatible.

In loose name equivalence, types must have same name or aliases of the same name or be declared with the same anonymous type. So a, b and c are compatible with each other and d and e are compatible.

In structural equivalence, types are equivalent if they have the same structure. Since both the structures have the same components with the same name, they are considered the same structure. So, all a, b, c, d, e are compatible with each other.

Question 2:
Consider the following functions:

```
int g(String s) {
double f(int x, int y) {
```

(Although the functions are written in C-style, this is an arbitrary functional language.)

Let function `h` be defined as

```
h(s1, s2) = f(g(s1), g(s2))
```

Use only Curry-ing, "hypothetical syllogism" and "implication introduction" to prove that `h` has type String → String → double.

Answer:

First we Curry $f(x, y)$ to produce funtion $f_1(x)$ that takes an `int` as input and produces the function f_2 as output. f_2 takes an `int` as input and produces a `double`.

Thus, the type of f_1 is $\texttt{int} \to (\texttt{int} \to \texttt{double})$. The function g has type $\texttt{String} \to \texttt{int}$. The type of $f_1 \circ g$ is

$$
\begin{array}{ll}
\texttt{String} \to \texttt{int} & \text{(The type of } g\text{)} \\
\quad \texttt{int} \to (\texttt{int} \to \texttt{double}) & \text{(The type of } f_1\text{)} \\
\hline
\texttt{String} \to (\texttt{int} \to \texttt{double}) & \text{(Composition } f_1 \circ g\text{)}
\end{array}
$$

To get the type of h, we need the type of $f_2 \circ g$. To expose the type of f_2, we can use the the type of $f_1 \circ g$ and the "implication introduction" rule of logic where if you "assume p" and with the assumption prove q, then you have $p \to q$.

$$
\begin{array}{ll}
\texttt{String} \to (\texttt{int} \to \texttt{double}) & \text{(Type of } f_1 \circ g\text{)} \\
\text{Apply input } \texttt{String} \text{ to } f_1 \circ g & \text{(i.e. "assume" } \texttt{String}\text{)} \\
\qquad \texttt{int} \to \texttt{double} & \text{(The output is } f_2\text{)} \\
\qquad \texttt{String} \to \texttt{int} & \text{(The type of } g\text{)} \\
\cline{1-1}
\qquad \texttt{String} \to \texttt{double} & \text{(Function composition } f_2 \circ g\text{)} \\
\hline
\texttt{String} \to (\texttt{String} \to \texttt{double}) & \text{("implication introduction")}
\end{array}
$$