Assignment #5

Name: _____ ID: _____

This assignment has **3** questions, for a total of **25** marks.

Question 1: ..... 5 marks

Prove that for any closed term f of type $\forall\alpha.\forall\beta.\alpha \to (\alpha \uplus \beta)$ and for any closed types τ_1, τ_2 value $v : \tau_1$, we have $f\ \tau_1\ \tau_2\ v \rightsquigarrow^* inl\ v$.

Question 2: **Z combinator typing** ..... 5 marks

This is the Z combinator in ULC:

$$\lambda f.\,(\lambda x.\,f(\lambda y.\,((x\ x)\ y)))(\lambda x.\,f(\lambda y.\,((x\ x)\ y)))$$

Add type annotations as well as fold/unfolds and prove it can be typed in System F + isorecursive types. Its type is $((\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)) \to (\tau_1 \to \tau_2)$ for arbitrary τ_1 and τ_2.

Question 3: **Encoding ULC into System F using recursive types** ..... 15 marks

Try to define type τ_u, which is the type that any ULC term can be given in F+isorecursive types. If you can define τ_u, define a function inductively on ULC terms so that it maps any ULC term to a term of F+isorecursive types whose type is τ_u and that has the same behaviour as the original ULC term (i.e., if you map an ULC application, you get something that *eventually* behaves like an application). If you cannot define τ_u, argue why it cannot exist.

In this case, consider ULC terms to be: $t ::= \lambda x.\, t \mid t\ t \mid \langle t, t\rangle \mid t.1 \mid t.2 \mid inl\ t \mid inr\ t \mid case\ t\ of\ inl\ x_1 \mapsto t \mid inr\ x_2 \mapsto t$. Encoding these terms into lambdas is not an option.