Normalization By Evaluation

1 An Starter Example

Normalization by evaluation is a technique for studying the normal forms of rewrite systems. The idea that terms often contain too much data. Consider the "free monoid on $\{x, y, z\}$". That is, $\mathcal{M} = \{x, y, z, \epsilon\}*$ Now, we can define some rewrite rules so

1. $(e_1 \cdot e_2) \cdot e_3 \to e_1 \cdot (e_2 \cdot e_3)$
2. $e \cdot \epsilon \to e$
3. $\epsilon \cdot e \to e$

Then a term with no empties, and fully right-associated is in normal form. So insight is that all that really matters is the order of non-empty terms. All the parenthesis and ϵs are extra data. So we magic-up a semantics which preserves exactly the needed info. Turns out lists suffice! To build our semantic domain, we first need a way of translating the variables. So let's define a type $T = X \mid Y \mid Z$. So our semantic domain is $\mathcal{L} = [T]$. Our goal is to turn $\mathcal{M}$ terms into $\mathcal{L}$ terms, do the compute there, and then return to $\mathcal{M}$.

The process of turning syntactic things into semantic things is called reflection.

1. $f\,\varepsilon = []$
2. $f(e_1 \cdot e_2) = f\,e_1 \;{+}{+}\; f\,e_2$
3. $f\,x = X \quad f\,y = Y \quad f\,z = Z$

And the reverse if reification.

1. $g\,[] = \varepsilon$
2. $g(X : tl) = x \cdot g(tl)$
3. $g(Y : tl) = y \cdot g(tl)$
4. $g(Z : tl) = z \cdot g(tl)$

The intention is to reflect then reify to evaluate things.

$$g(f((x \cdot x) \cdot (\varepsilon \cdot y \cdot z))) = g([X, X, Y, Z]) = x \cdot x \cdot y \cdot z \cdot \varepsilon$$

Now we get a trailing ε due to a quirk of how we defined g. But that doesn't change the good news:

1. Completeness: if e_1 and e_2 are equal in the rewrite system sense (E.G. in the same component of the rewrite graph), $g(f(e_1)) = g(f(e_2))$
2. Soundness: forall e, e and $g(f(e))$ are equal in the rewrite system.

Also critically, f and g terminate. So everything has a normal form, computable in this way (with a minor patch to g to fix that trailing ε). So the system is strongly normalizing! The proofs are, unsurprisingly, by induction. Observe that f respects the rewrite relations

$$f(e \cdot \varepsilon) = f(e) + +[] = f(e) \qquad f(\varepsilon \cdot e) = [] + +f(e) = f(e)$$

and

$$f((e_1 \cdot e_2) \cdot e_3) = (f(e_1) {+}{+} f(e_2)) {+}{+} f(e_3) = f(e_1) {+}{+} (f(e_2) {+}{+} f(e_3)) = f(e_1 \cdot (e_2 \cdot e_3))$$

So completness follows by induction on the length of the path from e_1 to e_2. and soundness follows by induction on the structure of e.

2 NBE for System F

The crux of the correctness of this approach is that the rewrite rules turn into actual equalities in the semantics. That is,

$$(x \cdot y) \cdot z \to x \cdot (y \cdot z) \quad \text{but} \quad (x +\!\!+ y) +\!\!+ z = x +\!\!+ (y +\!\!+ z)$$

So of course all of the terms related by a rewrite turn into literally equal things. Can we replicate that in System F? Short answer is no. We will need to some some much more clever things. Let's rule out some ideas that definitely don't work.

The obvious induction fails somewhat catastrophically. The following is quite false

$$e_1 \,\text{and}\, e_2 \,\text{strongly normalizing} \Rightarrow e_1\, e_2 \,\text{strongly normalizing}$$

In fact, consider $e_1 = e_2 = \lambda x.x\, x$. This term is strongly normalizing, but $e_1\, e_2$ is not. So this approach is a bit hopeless. Instead we will interpret things into relations built up by the types of terms. So for each type A, we will build a $[\![A]\!]$ that captures something about strongly normalizing terms of type A. Then we want to interpret each $\Gamma \vdash t : A \to [\![A]\!]$.

Turns out even this isn't enough. One of the key problems is building up that relation inductively doesn't make sense for $[\![\forall X.\, A]\!]$. Since $(\forall X.\, A)[X := (\forall X.\, A)]$ is substituting a type back in itself. This is called impredicativity, and it makes things hard. We will need a final trick to resolve this, called realizability. Intuitively, we think of A realizing a relation $\mathcal{A}$ when it witnesses the normalization of the terms in the relation. We'll formalize this shortly. But it will be enough to strength our induction hypothesis to finish the proof. From here we will follow https://www.cse.chalmers.se/ abela/lpar08.pdf who has a nice, detailed presentation of the construction for System F's semantics.