On the Undecidability of $UN^{=}$

Luis de Moraes ltdemoraes@uh.edu

UNIVERSITY of HOUSTON

Motivation

Term rewriting systems are a Turing complete model of computation.

Previous work has shown that shallow term rewriting systems have the property that uniqueness of normal forms is decidable for the underlying equivalence relation.

We show that relaxing the shallow restriction leads to a class of term rewriting systems for which this same property is undecidable.

Thus, the boundary for the decidability of this property is revealed.

Term Rewriting Systems

Term rewriting system can be thought of as rule-based programming.

First we define the signature of the term rewriting system, i.e. what symbols we have at our disposal to generate terms.

The signature defines all function symbols and their arities (number of arguments).

An example would be:

$$\{f{:}2,\ \ g{:}1,\ \ a{:}0,\ \ b{:}0\}$$

Terms are then generated by composing these function symbols: f(g(a),b); g(b); a; f(a,a); f(b,g(g(b))); etc...

A final remark on notation: terms of arity zero are called constants.

A term rewriting system consists not only of terms but also of rules.

Rules are defined in the form $l \to r$

Terms used in rules generally include variables. For instance:

$$f(x,x) \to g(x)$$
$$g(y) \to f(g(y),b)$$

The first rule can take the term f(a,a) to g(a) where x=a while the second rule can take the term g(g(a)) to f(g(g(a)),b) where y=g(a). If no rule applies to a term, then it is called a *normal form*.

The first rule is *shallow*, i.e. variables only appear in depth 0 or 1. The second rule is *linear*, i.e. variables only appear once on each side.

Another common constraint is flatness, where rules can only use terms of height 1 or 0 (constants have height equal to zero).

The underlying equational relation of a term rewriting system is the relation whether one term can reach another if we ignore the orientation of the rules, i.e. their direction.

Post Correspondence Problem

In order to prove undecidability for a property, we must reduce the problem to one we already know is undecidable.

The Post correspondence problem is known to be undecidable.

Definition 1 *Given a finite set of tiles* $\{\langle u_i, v_i\rangle \mid 1 \le i \le n\}$ *where* u_i, v_i *are words under some finite alphabet* Γ*, we must decide whether a sequence of indices* $i_1 \cdots i_k$ *exists such that* $u_{i_1} \cdots u_{i_k} = v_{i_1} \cdots v_{i_k}$*.*

An example would be the following tile set:

T1= $\begin{matrix}a\\baa\end{matrix}$ **T2=** $\begin{matrix}ab\\aa\end{matrix}$ **T3=** $\begin{matrix}bba\\bb\end{matrix}$ **With a solution...** $\begin{matrix}bba&ab&bba&a\\bb&aa&bb&baa\end{matrix}$

Now we must emulate this problem with a term rewriting system!!

The goal is for our term rewriting system to NOT have the property of uniqueness of normal forms under the equivalence relation $(UN^{=})$ **when the PCP instance in question has a solution.**

On the other hand, if the PCP instance has NO solution then our term rewriting system must have the property $(UN^{=})$

Rules

We group the rules according to their function in the process of verifying a solution.

The normal form 0 can "try" different tile sequences. Once one is chosen it is transformed into two different words. These words must match otherwise we won't be able to proceed.

Proceeding in this case means reaching the other normal form, 1.

$$\mathcal{R}_f := \left\{ \begin{matrix} 0 & \leftarrow f(T_i(x),\emptyset,\emptyset,\emptyset,\emptyset,\emptyset,\emptyset) \\ f(\emptyset,\emptyset,\emptyset,\emptyset,\emptyset,\emptyset,\gamma(x)) & \to 1 \end{matrix} \;\middle|\; \begin{matrix} 1 \le i \le |P| \\ \gamma \in \Gamma \end{matrix} \right\}$$

$$\mathcal{R}_T := \{ f(T_i(x),\emptyset,\emptyset,y,z,\emptyset,\emptyset) \to f(x,U_{i1},V_{i1},y,z,\emptyset,\emptyset) \mid 1 \le i \le |P| \}$$

$$\mathcal{R}_U := \left\{ \begin{matrix} f(x,U_{ij},w,y,z,\emptyset,\emptyset) & \leftarrow f(x,U_{i(j+1)},w,\gamma(y),z,\emptyset,\emptyset) \\ f(x,U_{i|u_i|},w,y,z,\emptyset,\emptyset) & \leftarrow f(x,\emptyset,w,\gamma(y),z,\emptyset,\emptyset) \end{matrix} \;\middle|\; \begin{matrix} 1 \le i \le |P| \\ 1 \le j < |u_i| \\ \gamma \in \Gamma \end{matrix} \right\}$$

$$\mathcal{R}_V := \left\{ \begin{matrix} f(x,\emptyset,V_{ij},y,z,\emptyset,\emptyset) & \leftarrow f(x,\emptyset,V_{i(j+1)},y,\gamma(z),\emptyset,\emptyset)\} \\ f(x,\emptyset,V_{i|v_i|},y,z,\emptyset,\emptyset) & \leftarrow f(x,\emptyset,\emptyset,y,\gamma(z),\emptyset,\emptyset) \end{matrix} \;\middle|\; \begin{matrix} 1 \le i \le |P| \\ 1 \le j < |v_i| \\ \gamma \in \Gamma \end{matrix} \right\}$$

$$\mathcal{R}_S := \left\{ \begin{matrix} f(\emptyset,\emptyset,\emptyset,\gamma(y),\gamma(z),\emptyset,x) & \to f(\emptyset,\emptyset,\emptyset,y,z,\gamma^c,x) \\ f(\emptyset,\emptyset,\emptyset,y,z,\gamma^c,x) & \leftarrow f(\emptyset,\emptyset,\emptyset,y,z,\emptyset,\gamma(x)) \end{matrix} \;\middle|\; \gamma,\gamma^c \in \Gamma \right\}$$

$$\mathcal{R}_n := \left\{ \begin{matrix} f(x_1,x_2,x_3,x_4,x_5,x_6,x_7) \to f(x_1,x_2,x_3,x_4,x_5,x_6,x_7) \\ T_i(x) \to T_i(x) \quad U_{ij} \to U_{ij} \quad V_{ij} \to V_{ij} \\ \gamma(x) \to \gamma(x) \quad \gamma^c \to \gamma^c \quad \emptyset \to \emptyset \end{matrix} \right\}$$

Proof Sketch

We prove by induction on the length of words that R_U **correctly translates the term** U_{11} **in position 2 to the corresponding word (albeit reversed) in position 4.**

Again, by induction, we prove a similar claim but with respect to R_V **. We use the previous proof as a step, since before we can apply any rules from** R_V **we must first obtain a** $\emptyset$ **in position 2.**

Finally, using both proofs we show the tile sequences are converted correctly. It is much easier to see how the two words must match to reach the normal form 1.

An informal method of demonstrating the only-if part is by counting the rules that apply at any given time. Except for R_f **there should be only two rules applicable to any proper term (one with subterms without** $f{:}7$ **itself). This greatly facilitates dealing with the underlying equivalence relation, since one of the common problems in this case is accidentally creating more paths than intended.**

Since there are only 2 rule applications this means there are only two ways to arrive at a term. In particular, once we arrive at a term we are left with a single choice in order to proceed otherwise we end up backtracking.

Conclusion

We have shown that relaxing the shallowness restriction leads to term rewriting systems with $UN^{=}$ **undecidable.**

Our construction also provides a bit of insight into how we can manipulate a term rewriting system. The key to avoiding the creation of accidental paths is to make sure rules have a precondition as well as a postcondition. Then it is just a matter of making sure preconditions and postconditions only overlap when needed.

Future work involves checking the decidability of different classes of term rewriting systems in addition to looking at a few less common restrictions.