Equational Reasoning with Limited Knowledge

P. Purdon, University of Tasmania

Abstract

Rational automata [2] are autonomous communicating equational reasoning agents, that operate within an algebraic model of a world of which they only possess partial information. They seek to expand their knowledge of the world.

To do this they reason about what they know, and converse with other rational automata within the world. Part of such conversations is the asking of questions. Questions are useful to a rational automaton as they allow the automaton to seek information than can be used to expand its knowledge of the environment, and hence improve its responses to that environment.

There are two types of questions used by an automaton - requests for information, and requests for confirmation. We have found methods to generate such questions, and have implemented these methods in Mathematica. We give examples of the questions generated by these methods when they are applied to a (small) world.

Gröbner Bases [3], Knuth-Bendix [1], and similiar algorithms are usually used in algebras where the equations that define the algebra are viewed as expressing all things that are true in the world that the equations model. If we define the meaning of a term in the algebra as being the normal form that these algorithms reduce that term to, then we can use the algorithm to perform equational reasoning. This reasoning process is thus aimed at simplifying the set of equations that define the algebra, to make questions about equality of terms easier to answer. Terms that cannot be proved to be equal using these equations and the rules of inference of the reasoner; are assumed to be unequal. This is commonly known as the *closed world assumption*, meaning that the world being modelled by the algebra is closed and that no new information about the world can be found. Hence if something is not provably true now, it never will be. Thus the world being modelled is static and unchanging.

In principle, the closed world assumption is equivalent to explicitly listing all the pairs of terms that are not provably equal in our model of the world. Such pairs we call inequations, and are just equations over which equality does not hold in the model. We denote them by $a \neq b$.

Removing the assumption in favour of a set of inequations can only be done if the word problem for the algebra is solvable, and even then the set of inequations is (potentially) infinite. So inequations look to be of limited use.

But consider what happens if we allow the world being modelled to change with time. Our model of the world must also reflect these changes. We could still keep the closed world assumption, just applying it at each discrete step of the evolution of the model. As such, it would still correspond to a (potentially) infinite set of inequations. But what if we only have limited information about the world to construct our model from, say only those things we initially knew and those things we have learned since then.

Then there are advantages to easing the restriction that the closed world assumption makes on what is known about the world. We do keep the assumption, but we restrict it to operating only on grounded functions; as such it becomes the *unique names assumption* instead. We still use inequations to store those terms that we cannot prove to be equal, but now the interpretation of what that means changes. Before it meant that the terms could never be proven to be equal, now it has become that we cannot as yet prove them to be equal or unequal; the truth value of the sentence is unknown at this time. Thus inequations are used to store sentences whose truth value in the model is currently unknown.

The algebra we are using to model the world consists of a set of non-deterministic n-ary functions with composition. The nullary functions are called the *ground* elements (or *constants*) of the algebra. A term of the algebra is said to be *grounded*, or called a *ground term*, if it ends in a ground element. We extend the definition of equality to be

$$f = g \Leftrightarrow \forall x \{a : f[x] = a\} = \{b : g[x] = b\}$$

As normal, inequality holds when equality does not. For brevity, we denote $f[a]$ by fa from now on.

1 Rational Automata

Rational automata [2] are autonomous communicating equational reasoning agents, that operate over a limited type of algebraic structure, namely algebras of deterministic unary partial functions. They are designed with working over non-deterministic partial functions in mind, but currently the implementation does not support this. They are an attempt to investigate what is actually required for efficient knowledge acquisition, and have so far managed to make reasonable human-like conversation about their limited worlds. When several automata are conversing we say they are particpating in a *dialogue*. The knowledge of any automaton is stored as equations and inequations of the term algebra. Perhaps their most important feature is that they are designed to function with limited and partial information about their world, and the functions in it. They are also limited in contemplation time, with a requirement for real-time interaction also imposed on them. They use a restriction of the Knuth-Bendix algorithm to generate new knowledge, and to simplify what they know.

The question arises as to why rational automata would find inequations useful. There are several reasons :

1) because they allow an automaton to express negative knowledge,

2) because with them an automaton can derive contradictions earlier,

3) and because they allow an automaton to form more questions.

The first reason means that automata can disagree with each other. That is, if one says that $a = b$, and another happens to know the inequation $a \neq b$, the second automata can say that it thinks that the first one is wrong.

We can still generate contradictions without inequations if we can simplify an equation down to $a = b$, where a and b are distinct grounded functions of the algebra. Then the unique names assumption comes into consideration, allowing us to find the contradiction still.

However, having inequations allows contradictions to be found that are not yet reducible to such grounded equalities, and indeed may never be.

In this paper we are primarily interested in the third reason, the ability to form questions. Questions allow the automaton to seek information about the world it is in, and hence to refine its model of that world. In evolutionary terms, such an improved model equates to improved fitness, and hence ability to survive. Inequations allow the formation of many more questions than just equations, as can be seen in later in this paper.

Of course, to be useful, the automaton needs someone to answer its questions. To give our automata someone to talk to, we run several together in a dialogue.

2 Active systems

Computer algebra systems and theorem provers are usually *passive*, that is they do not seek additional information about the world they exist in. This is due to the closed world assumption being applied at all levels of the algebra, not just the ground level as we do with rational automata. With the closed world assumption present this behaviour is justified, as no additional information will alter the answer the system gives to a question, as it already knows all there is to know.

Consider what would happen in any system to a question if the closed world assumption is present at all levels. The system could ask itself the question it has generated, and it would either reduce the question down to True, meaning that the question is a consequence of the equations it already knows, or reduce it down to an irreducible equation. The assumption would then take that irreducible equation and say that it must be False (as it cannot be proved True). Thus there is no need to ask anything else the question, as the system can answer the question itself. With the closed world assumption any system can *always* answer its own questions (and any other question as well). Such a system is an oracle for the world modelled by the algebra.

Clearly for any kind of system that is going to truly interact with another system (human or otherwise) this is not the behaviour that is wanted. So we have chosen to create an *active* system, one that asks questions about its world, without being prompted. Also clearly, such an active system cannot have the closed world assumption acting as just described.

Wang [4] has created in his NARS what we call a *weakly active* system, that is a system that will ask questions, but only to help it answer a question it was asked. Thus the question it asks is a derivative of what it was asked and what it knows. It does not spontaneously ask a question that may be unrelated to the dialogue so far.

Wang has his own working definition of intelligence, which is *"Intelligence is the capacity of an information-processing system to adapt to its environment while operating with insuffícent knowledge and resources"* [5].

However, his NARS system does not seek to adapt to its environment except to re-prioritize what to think about next. The possesion of curiousity seems to be implicit in his definition, as before one can decide how to adapt to an environment, one needs to acquire knowledge of that environment; and to do so it helps to ask questions.

Independently, we have made automata with the goal of investigating methods of acquiring and sharing knowledge efficiently. A property of such a system would seem to be the ability to find gaps in their own knowledge of the world, and seeking to fill these gaps in. Hence our automata exhibit a desire to know about their environment, so they can improve their model of the world in which they lives. Thus our automata seem to fit Wang's definition also.

We are trying to make a system that strives within the bounds of limited resources to model its world. The system must find out about its world, and an effective method of doing so is to ask questions of any other entities within that world. Of course, to ask questions one must first generate them, which is the main topic here.

As stated before, there are two types of questions. Requests for confirmation are just sentences that the automaton suspects may hold in the world, and are based on an inductive style of generation from its knowledge. So we may have the sentence *father Bart = father Lisa*, which is interpreted in english to be *"Is the father of Bart the same person as the father of Lisa?"*.

As we see later, one advantage of this style of generation is that it will produce questions that make sense. This is particularly important for the automatons, as they have no information about the form of their world embedded within them.

Requests for information are terms within the algebra, but they are terms that are drawn from the equations and inequations the automaton knows already. They can be drawn from any inequation or equation which does not have just a ground element on any side. So from the above example of *father Bart = father Lisa*, we can get two requests for information, which are *father Bart* and *father Lisa*. These are interpreted to be *"Who is the father of Bart?"* and *"Who is the father of Lisa?"*.

Any attempt to generate questions is obviously going to need more than just a declarative view of the world, as questions are implicitly not declarative in nature. Thus we need to provide the automata with a means of going beyond the declarative nature of their knowledge, into the realm of speculation. As we want them to maintain their knowledge as consistent with the world as they can, we do not want untested speculations as part of that knowledge. We also do not want wild speculation about circumstances that cannot exist in their world. So we need a method of generating questions about the world they know, that are a reasonable attempt to extend what they know, and that they cannot themselves answer quickly. We have found methods of generating such questions, that make use of the fact that in an open world we are unlikely to have total knowledge about any of the functions in that world.

In order to ask a question the automaton needs to be able to form questions from the information it knows, without prompting from any other participant in the dialogue. Indeed, it should be capable of generating questions when there are no other participants, and keeping those questions until it has someone to talk to.

3 Generating Questions

3.1 The Scattershot Approach

Perhaps the most obvious way of generating questions is the *scattershot approach*. This is to simply choose two terms from the algebra, and use the current state of knowledge to reduce them to their normal forms. If they are equal (or provably unequal), then choose another two terms and start again. If not, then these two normal forms make a potential question, that will expand our knowledge of the structure of the algebra.

This approach is best kept in storage, and only used if the other methods of generating questions do not generate any questions. This is particularly due to the sheer amount of computational resources that this method may consume to find possible questions.

In practice, to use this approach efficiently we would need heuristics for guiding the selection of the terms to consider, but doing this without involving outside semantic knowledge in the heuristic seems improbable. Without these heuristics, the process would consider many terms that are syntactically correct, but are semantically meaningless (that is, they correspond to nothing in the world being modelled). This means that questions generated by this scattershot style approach are likely to be irrelevant to any conversation.

Indeed, for this approach this problem is likely as we have not placed any form of typing restrictions on our system. Thus the syntax will allow us to generate (semantically) nonsensical terms, wherein the domains and codomains of the operators in the terms do not even match.

3.2 Requests for Information

The scattershot approach can obviously be trivially altered to produce questions that ask for information, rather than just confirmation as described. This is as simple as only asking about one of the two terms being considered. However, the problem mentioned at the end is still present.

We do have a source of terms that are reasonable questions though, the automatons own knowledge. If we process the knowledge of an automaton, keeping in a set all those terms we find that are not just ground elements, then we have accumulated a set of questions that will expand the automatons knowledge of its world.

Using this method, we can quickly produce list of reasonable questions, which as a bonus will have the advantage of simplifying the automatons knowledge (with no loss of information content).

3.3 Requests for Confirmation

We have found ways of combining equations and inequations to form questions that are both well-formed terms in the model and also correspond to things in the world, using only the assumption that all knowledge the automaton has accurately corresponds to the world also (that is, the knowledge is correct semantically). This will of course be met in pratice, as long as any human input is also semantically correct. This is due to the Knuth-Bendix algorithm [1] always preserving the syntax and semantics of the model, and noticing that all other automata in the dialogue operate in this way too. The Knuth-Bendix algorithm performs in this fashion as it equates to replacing equals with equals, which of course preserves the correctness of the syntax and semantics of the model.

The generation of questions is split into sections, each based on how the questions are being formed. These sets of questions are then merged to remove duplicates, and a preliminary reduction using the automatons knowledge is done. The question is only left in the set of questions to ask if the automaton that generated it cannot reduce it to either True or False. The automaton can only do so if it is a consequence of the automatons knowledge. Thus if the question survives reduction by the automaton that generates it, the question is a reasonable conjecture by that automaton at that time. This allows us to meet the above criteria that the questions be such that the automaton itself cannot quickly answer them.

Here we define these methods in terms of non-deterministic partial unary functions, for relative simplicity. They can be restricted to grounded equations, and also extended to n-ary functions, easily. The restriction to grounded equations renders some rules no longer applicable, and when extending to n-ary functions we can use the properties of the function to generate additional questions. That is, from $a(b,c) = d$ and $a(e,f) = d$ we can get the questions $\{\{b,e\},\{c,f\}\}$. If we know that a is also commutative, then the additional questions $\{\{b,f\},\{c,e\}\}$ should also be considered. Note that in this case, as well as some described below, the set of questions is not independent. This is actually useful, as it provides the automaton with several alternate routes to finding an answer.

We also do not give the converses of these rules, which nearly all of them have (the exception is noted). By converse, we merely mean the variant of the rule with the order of composition reversed. That is, $ba = c$ instead of $ab = c$ etcetera.

From two equations

If we take two equations like $ab = d$ and $ac = d$, what can we conclude about the functions b and c? We can conclude that for all elements within the domain of a, they must agree (or else we could generate a counter-example to one of the equations by using that element). What happens though if b and c have domains that include elements not in the domain of a? What can we say then? Obviously we cannot conclude that b and c are equal, without knowing the domains of all the functions. We also cannot conclude they are not equal (as we do not have the closed world assumption unless b and c are ground elements). Also, we do not know if they may become equal in the future, due to additions and/or revisions to our knowledge. Therefore the only reasonable recourse is to put the matter of 'does $b = c$?' into the category of *ask someone else*. Thus it becomes a reasonable question, one based on the available knowledge. It is also a reasonable extension of our knowledge to seek, being based on what we know to hold.

$$\textbf{TE} : \{ab = d, ac = d\} \text{ generates } \{\{b,c\}\}$$

From two inequations

Somewhat similiar reasoning can be done with two inequations, $ab \neq c$ and $b \neq d$. Here we have more uncertainty about possible equalities. Consider that although $b \neq d$, that inequality of b and d may occur outside the domain of a. Thus, within a's domain, we may have $b = d$. That would make $ad = c$ an equation that holds as well. Thus as we do not know where b is unequal to d, we are justified in adding 'does $ad = c$?' to the set of questions. We can also raise the question of whether $ab = ad$ as well, for the same reason.

$$\textbf{TI1} : \{ab \neq c, b \neq d\} \text{ generates } \{\{ad,c\},\{ab,ad\}\}$$

There are several variants on this, based on which of the sentences above are considered to be inequations or questions.

$$\textbf{TI2}: \{ab \neq c, ad \neq c\} \text{ generates } \{\{b, d\}, \{ab, ad\}\}$$

$$\textbf{TI3}: \{ab \neq c, ab \neq ad\} \text{ generates } \{\{ad, c\}\}$$

A separate path comes from $ab \neq d$ and $bc \neq e$. It is based on replacing the use of equations in the Knuth-Bendix inference rule *Deduce* with inequations, and seeking to find whether any of the conclusions from this deduction hold, that is whether $abc = dc$, $abc = ae$, or $dc = ae$. The first two are also the case of considering whether we have made a sufficient restriction to the domain of the previous inequation that is has become an equation. That is, that by pre (or post) composing the inequation with another function we have restricted the domain of the new inequation to such an extent that equality now holds. We are assured that the new terms will still make sense as long as the input inequations do.

This gives us the following (which is its own converse),

$$\textbf{TI4}: \{ab \neq d, bc \neq e\} \text{ generates } \{\{abc, dc\}, \{abc, ae\}, \{dc, ae\}\}$$

From one equation

This is a variant of the above cases, where we consider stripping the first/last function off existing equations. That is, if we remove the restriction that a places on $ab = ad$, does $b = d$?

$$\textbf{OE}: \{ab = ad\} \text{ generates } \{\{b, d\}\}$$

From one each

Another rearrangment of equalities from the FromTwoIneq reasoning leads us to the cases of

$$\textbf{OIOE1}: \{ab = c\}, \{b \neq d\} \text{ generates } \{\{ad, c\}\}$$

$$\textbf{OIOE2}: \{ab = e\}, \{bc \neq d\} \text{ generates } \{\{ad, ec\}\}$$

From one inequation

We could consider the possibility of pre/post composing an inequation $b \neq d$ by some a, and it would generate something that we could ask. Having the a before/after the inequation restricts the set of cases to consider, possibly removing from consideration those cases that make the inequality. However, such a method takes no accound at all of the semantics of the resulting question 'is $ab = ad$?'. It is quite likely that we have semantic nonsense again, as we have no guide as to whether the composition makes sense. In the previous cases, such semantic issues where handled by the involvement of the second equation/inequation, or by stripping the first/last function off an equation.

4 Examples

We deal here with a toy world, based on the normal cartesian plane. It has four ground functions representing points, A, B, C, D. It also has two unary functions r and h, which take points to points, with r performing a 90 degree counter-clockwise rotation, and h a reflection about the x-axis.

Such a simple world is used so as to not obscure the workings of the question generation with unnecessary detail in the model.

If we allow $\{rA = B, hA = B\}$, then from **TE** we get the question does $r = h$?. This is reasonable, as if for all points we know about we have their reflection and their rotation being the same point, it is sensible to ask whether this always is the case.

For **TI1**, having $\{rB \neq C, B \neq D\}$ means generate the questions $\{\{rB, rD\}, \{rD, C\}\}$. These are both reasonable, as if $rB \neq C$ and $B \neq D$ then it is possible that $rD = C$ and $rB = rD$. Remember that we have no other information to work with, in particular, we know nothing about the point D, other than it not being the same point as B. Note that this particular instance would not occur in practice, as $B \neq D$ would never be explicitly in the knowledge of an automaton (it is a consequence of the unique names assumption). We have used it here for clarity.

Similiarly for **TI2**, and **TI3**; if we have $\{rB \neq C, B \neq D, hB \neq hD, rA \neq C\}$ then we get the set of questions $\{\{B, A\}, \{rB, rA\}, \{rB, rD\}, \{rD, C\}\}$. Note that $\{B, A\}$ would be removed from the list of questions, as it is just an example of the unique names assumption again.

Starting with $\{rh \neq h, hB \neq C\}$, **TI4** gives $\{\{hB, rC\}, \{rhB, hB\}, \{rhB, rC\}\}$. These are all the result of considering the term rhB, and what we could simplify it to. We know that $rh \neq h$, but the possibility exists that even so rhB could equal hB. Similiarly for the other two questions.

For **OE** we have left the results of the converse in, to further illustrate what we mean by converse. So from $\{rhA = rA\}$, it gives us $\{\{hA, A\}, \{rh, r\}\}$. As you can see, it has matched on both $ab = ad$ and $ba = da$, to give us two questions.

Finally, **OIOE1** generates $\{\{rC, B\}\}$ from $\{rA = B, A \neq C\}$, and **OIOE2** generate $\{\{hrrrB, rC\}\}$ from $\{rh = hrrr, hB \neq C\}$.

Examples for the other methods given previously are not given, as they are straightforward to generate.

5 Conclusion

We have described the basics of an automaton that uses equational reasoning to manipulate its algebraic model of the world. As the automaton is only given partial knowledge of that world initially, it seeks out additional information via the process of generating and asking questions. The methods given here produce questions that seek to extend the automatons knowledge, or that will verify the inductive conjectures of the automaton. So our automatons are active systems, that initiate the process of knowledge acquisition, and by doing so they share knowledge among themselves in an efficient manner.

References

[1] Dershowitz, N., Jouannaud, J.P., Rewrite Systems. In Leeuwen, J. van, (ed.): Handbook of Theoretical Computer Science Vol. B, 243-320 (1990).

[2] Fearnley-Sander, D., Mathematical Structures for Rational Discourse. In Fitz-Gerald, G., Wang, D. and Yang, W-C. (eds.): Proceedings of the 4th Asian Technology Conference in Mathematics, 311-320 (1999).

[3] Cox, D., Little, J. O'Shea, D., Ideals, Varieties, and Algorithms, chapter 2, 48-112 (1992).

[4] Wang, P., Non-Axiomatic Reasoning System (Version 4.1), Proceedings of the Seventeenth National Conference on Artificial Intelligence, 1135–1136, (2000).

[5] Wang, P., On the Working Definition of Intelligence, Technical Report 96, Center for Research on Concepts and Cognition, Indiana University (1994).