Historia Mathematica Vol. 11, No. 2, 1998, 35-42

등식체계에서의 자동증명

아주대학교 정보 및 컴퓨터공학부 **위규범**

Abstract

It is an undecidable problem to determine whether a given equation logically follows from a given set of equations. However, it is possible to give the answer to many instances of the problem, even though impossible to answer all the instances, by using rewrite systems and completion procedures.

Rewrite systems and completion procedures can be implemented as computer programs. The new equations such a computer program generates are theorems that hold in the given equational theory. For example, a completion procedure applied on the group axioms generates simple theorems about groups.

Mathematics students' learning to know the existence and mechanisms of computer programs that prove simple theorems can be a significant help to promote the interests in abstract algebra and logic, and the motivation for studying.

0. 서론

현대 대수학을 처음 배우는 학생들은 반군(semigroup)이 좌항등원(left identity)을 가지고, 모든 원소가 좌역원(left inverse)을 가지면 군(group)이 된다는 연습문제를 풀게 된다. 이 문제를 기계적으로 증명하는 컴퓨터 프로그램에 대한 설명은 많은 학생들에게 흥미로운 주제일 것이다.

우리는 등식으로 표현된 체계를 많이 다룬다. 등식체계에 관한 가장 중요하고 근본적인 문제 중의 하나는 어떤 주어진 등식이 그 등식체계로부터 연역될 수 있는가를 결정하는 것이다. 이 문제를 word problem이라고 부른다. 위에서 말한 군에 관한 문제는 아래의 세 개의 등식으로 이루어진 체계로부터 등식 $*(X, e) = X$와 $*(X, i(X)) = e$가 논리적 결과로서 성립함을 증명하는 것이다.

(1) $*(e,X)=X$
(2) $*(i(X),X)=e$
(3) $*(*(X,Y),Z)=*(X,*(Y,Z))$

아래의 예는 적절한 등식을 선택하고 적절한 표현을 대입함으로서 $*(X,e)=X$가 성립함 보인다.

$$\begin{aligned}
*(X,e) &= *(*(e,X),e) && \text{등식(1)}\\
&= *(*(*(i(i(X)),i(X)),X),e) && \text{등식(2)}\\
&= *(*(i(i(X)),*(i(X),X)),e) && \text{등식(3)}\\
&= *(*(i(i(X)),e),e) && \text{등식(2)}\\
&= *(i(i(X)),*(e,e)) && \text{등식(3)}\\
&= *(i(i(X)),e) && \text{등식(1)}\\
&= *(i(i(X)),*(i(X),X)) && \text{등식(2)}\\
&= *(*(i(i(X)),i(X)),X) && \text{등식(3)}\\
&= *(e,X) && \text{등식(2)}\\
&= X && \text{등식(1)}
\end{aligned}$$

위의 예에서 보듯이, 원하는 결과를 얻기 위하여 적용해야하는 등식들의 순서와 변수들의 치환값을 알아내기는 쉬운 일이 아니다. Rewrite system이라는 도구를 이용하여 등식체계로부터 주어진 등식이 성립하는가를 판정하는 방법을 살펴본다.

1. Rewrite systems

Rewrite rule이란 두 개의 항(term)의 짝으로, $s\rightarrow t$로 표시한다. Rewrite system이란 rewrite rule들의 집합이다. 어떤 term M이 rewrite rule $s\rightarrow t$에 의하여 다른 term N으로 rewrite(reduce 또는 simplify)된다는 것은 s가 M의 어떤 부분항(subterm)과 match할 때(다시 말해서, 어떤 치환 σ에 대하여 $\sigma(s)$가 그 부분항과 일치할 때) 그 부분항을 $\sigma(t)$로 치환한 것이 N이다. 이때에 $M\Rightarrow N$으로 표기한다. 관계 $\Rightarrow$의 reflexive-transitive closure를 $\Rightarrow^*$로 표기한다. Rewrite system 안에 있는 어떤 rewrite rule도 term t를 reduce할 수 없을 때 t를 irreducible이라 한다. $t\Rightarrow^* u$이고 u가 irreducible이면, u를 t의 정규형(normal form)이라 한다.

예를 들면, rewrite system $R=\{h(a)\rightarrow b,\ f(b,X)\rightarrow g(X,X)\}$에 의해서 다음의 derivation이 가능하다: $k(f(h(a),f(a,Y)),f(X,b)) \Rightarrow k(f(b,f(a,Y)),f(X,b)) \Rightarrow k(g(f(a,Y),f(a,Y)),f(X,b))$.

Rewrite system이 종료(terminate)한다는 것은 infinite derivation이 없다는 뜻이다. Rewrite system이 합류적(confluent)이라는 것은 $M\Rightarrow^* N_1$이고 $M\Rightarrow^* N_2$일 때 어떤 P가

존재하여 $N_1 \Rightarrow^* P$와 $N_2 \Rightarrow^* P$를 만족함을 뜻한다. 합류성은 Church-Rosser 성질이라고도 말한다[1]. 종료는 정규형이 존재함을 의미하고 합류성은 정규형이 유일함을 의미한다. 종료하고 합류적인 rewrite system을 완전(complete)하다고 부른다.

주어진 등식체계의 각 등식에 방향을 부여하여 rewrite rule로 변환하면 rewrite system을 얻는다. 이 rewrite system이 만약 완전하다면, word problem의 decision procedure로 사용할 수 있다: 주어진 등식의 양변을 각각 rewrite system에 의한 정규형으로 reduce하여 두 개의 정규형이 일치하면 등식이 성립하는 것이고, 일치하지 않으면 성립하지 않는 것이다. 이 rewrite system이 완전하지 않은 경우에 완전하게 변환하는 절차를 완전화 절차(completion procedure)라고 부른다. 물론 완전화 절차는 항상 성공하지는 못한다. 왜냐하면, 만약 완전화 절차가 항상 성공한다면 word problem이 decidable problem이라는 결론에 이른다. 그러나 word problem은 undecidable problem이다.

대표적인 완전화 절차는 Knuth-Bendix completion procedure이다. 이 완전화 절차에 대한 설명을 위하여 먼저 종료와 합류성에 관한 좀더 자세한 논의가 필요하다.

2. 종료

임의의 rewrite system이 종료하는가 아닌가를 판정하는 것은 결정불가능(undecidable)한 문제이다[2]. 그러나 종료하는 rewrite system에 대하여 종료한다는 것을 많은 경우에 (모든 경우에 대해서는 불가능하지만) 확인할 수 있는 방법들이 있다.

항들의 집합 T에서 정의된 부분순서(partial ordering) $>$가 단조(monotone)하다는 것은 모든 항 t, u에 대해서 $t > u$이면 $f(\ldots t\ldots) > f(\ldots u\ldots)$가 성립함을 뜻한다. 모든 항 t, u와 치환 σ에 대해서 $t > u$이면 $\sigma(t) > \sigma(u)$가 성립하면 $>$가 안정적(stable)이라고 말한다. 단조하고 안정적인 well-founded partial ordering을 reduction ordering이라 한다. Rewrite system R의 모든 rule $\alpha \rightarrow \beta$에 대하여 $\alpha > \beta$이면 R은 종료한다. 많이 연구되어진 reduction ordering들은 polynomial ordering[3], recursive path ordering[4], paths of subterms ordering[5], recursive decomposition ordering[6], KNS ordering[7], Knuth-Bendix ordering[8] 등이다. 가장 널리 쓰이는 recursive path ordering을 아래에 소개한다.

$>$를 함수기호(function symbol)들의 집합 F에 정의된 부분순서라고 하자. 항들의 집합 T에 recursive path ordering $>_{RPO}$는 다음과 같이 정의된다:

$f > g$이고 모든 $i = 1, 2, \ldots, n$에 대하여 $s > t_i$

또는

$f = g$ 이고 $\{s_1, \ldots, s_m\} \gg_{RPO} \{t_1, \ldots, t_n\}$

또는

어떤 $i = 1, 2, \ldots, m$에 대하여 $s_i \geq_{RPO} t$이면,

$s = f(s_1, s_2, \ldots, s_m) >_{RPO} g(t_1, t_2, \ldots, t_n) = t$이다.

여기서 $\gg_{RPO}$는 $>_{RPO}$의 multiset extension이다.

예를 들면, $\neg > \wedge > \vee$ 일 때, 아래의 rewrite system은 recursive path ordering에 의하여 종료한다.

$$\neg\neg X \rightarrow X$$
$$\neg(X \vee Y) \rightarrow \neg X \wedge \neg Y$$
$$\neg(X \wedge Y) \rightarrow \neg X \vee \neg Y$$
$$X \wedge (Y \vee Z) \rightarrow (X \wedge Y) \vee (X \wedge Z)$$
$$(Y \vee Z) \wedge X \rightarrow (Y \wedge X) \vee (Z \wedge X)$$

3. 합류성

임의의 rewrite system에 대하여 합류적인지 아닌지 판정하는 것 역시 결정불가능한 문제이다. 그러나 종료하는 rewrite system들에 대하여는 합류성을 판정하는 절차가 있다.

모든 M, N, P에 대해서 $P \Rightarrow M$이고 $P \Rightarrow N$이면 어떤 Q가 존재하여 $M \Rightarrow^* Q$와 $N \Rightarrow^* Q$를 만족시킬 때 rewrite system이 국소적으로 합류적(locally confluent)이라고 말한다. Rewrite system이 종료하면 confluent와 locally confluent는 동치이다[9].

두 개의 항 s와 t에 대하여 $\sigma(s) = \sigma(t)$ 을 만족하는 치환 σ가 존재하면, s와 t가 unifiable이라 하고, σ를 unifier라 부른다. 항 s와 t의 임의의 unifier τ에 대하여 $\tau = \tau' \circ \sigma$ 을 만족하는 τ'이 존재하면 σ를 most general unifier(mgu)라고 부른다.

$\lambda \rightarrow \rho$와 $\lambda' \rightarrow \rho'$에서, λ의 어떤 위치 p가 존재하여, λ의 p에서의 부분항 $\lambda|_p$와 λ'이 unifiable하다고 가정하고, mgu를 σ라 하자. 다시 말해서, $\sigma(\lambda|_p) = \sigma(\lambda')$. 이때 $\sigma(\lambda)$는 두 개의 다른 term으로 reduce될 수 있다:

$\lambda \rightarrow \rho$에 의해서 $\sigma(\rho)$로 reduce되고, $\lambda' \rightarrow \rho'$에 의해서 $\sigma(\lambda[\rho']p)$로 reduce된다. $\lambda[\rho']p$는 λ의 위치 p에서의 부분항을 ρ'으로 대치한 것이다. 이때에 $\langle \sigma(\lambda[\rho']p), \sigma(\rho) \rangle$를 critical pair라 부르고, $\lambda \rightarrow \rho$와 $\lambda' \rightarrow \rho'$가 superpose되었다고 한다.

예를 들어서, $f(X, g(X, h(Y))) \rightarrow k(X, Y)$와 $g(a, Z) \rightarrow l(Z)$를 생각해 보자. 첫 번째 rule의 left hand side의 두 번째 부분항과 두 번째 rule의 left hand side가 mgu $\sigma = [X \rightarrow a,\ Z \rightarrow h(Y)]$로 unify된다. 이 때에 $f(a, g(a, h(Y)))$는 첫 번째 rule에 의해서 $k(a, Y)$로 reduce되고, 두 번째 rule에 의해서 $f(a, l(h(Y)))$로 reduce되므로, critical pair $\langle f(a, l(h(Y))), k(a, Y)\rangle$가 생긴다.

Rewrite system이 locally confluent함은 모든 critical pair에 대하여 어떤 항 t가 존재하여 양변이 t로 reduce됨과 동치이다[10]. 그러므로 rewrite system R이 종료할 때 R이 합류적임과 모든 critical pair의 양변의 정규형이 일치함이 동치이다[8]. 모든 critical pair의 양변의 정규형이 일치하는지 아닌지를 판정하는 것은 결정 가능한 문제이므로, 따라서 종료하는 rewrite system의 합류성을 판정하는 것도 결정 가능한 문제이다.

Knuth-Bendix completion procedure의 근본적인 역할은 양변의 정규형이 일치하지 않는 critical pair를 rewrite rule로 전환하여 합류적인 rewrite system을 만드는 것이다. 이때 물론 critical pair를 rewrite rule로 전환할 때 방향을 잘 설정하여서, 새로운 rewrite rule을 첨가한 rewrite system도 여전히 종료하도록 유지해야 한다.

4. Knuth-Bendix 완전화 절차

입력: - 등식체계 E
- reduction ordering $>$

출력: - complete irreducible rewrite system R로서 $s =_R t$와 $s =_E t$이 동치임.

$R := \emptyset$
while $E := \emptyset$ **do**
  E에서 등식을 하나 택해서 $s = t$라 하자.
  s와 t를 R에 의해서 각각 normal form으로 reduce하여 s'과 t'이라 하자.
  if $s' \equiv t'$ **then** { s' is identical to t' }
    $E := E - \{s = t\}$
  else
    if $s' > t'$ **then**
      $\alpha := s'$; $\beta := t'$
    else if $t' > s'$ **then**

$\alpha := t'$; $\beta := s'$

else

failure

fi;

$R := \{ \gamma \rightarrow \delta' \mid \gamma \rightarrow \delta \in R$이고 δ'은 $R \cup \{\alpha \rightarrow \beta\}$에 의한 δ의 정규형$\}$

R의 모든 rule과 $\alpha \rightarrow \beta$사이에 생기는 모든 critical pair들을 계산하여 이들의 집합을 CP라 하자.

$E := E - \{s = t\} \cup \{ \gamma \rightarrow \delta \mid \gamma \rightarrow \delta \in R$이고 γ가 $\alpha \rightarrow \beta$ 에 의하여 reduce된다$\} \cup CP$

$R := R \cup \{\alpha \rightarrow \beta\} - \{ \gamma \rightarrow \delta \mid \gamma \rightarrow \delta \in R$이고 γ가 $\alpha \rightarrow \beta$ 에 의하여 reduce된다$\}$

fi

od;

success

앞에서 언급했듯이 Knuth-Bendix completion procedure는 항상 성공하지는 못한다. 어떤 때에는 critical pair의 어느 한 쪽도 다른 쪽보다 reduction ordering에 의하여 크지 않아서 실패하고[8], 어떤 경우에는 무한히 많은 critical pair를 만들어 내며 정지하지 않는다[11].

Rewriting에 기초한 최초의 word problem의 decision procedure는 1951년 Evans에 의하여 제안되었다[12]. 일반적으로 완전화 절차가 항상 성공하지는 못하는 이유는 다양하다. 어떤 등식체계는 finitely based가 아니다[13]. 등식 체계가 finitely based라는 것은 그 등식체계에서 성립하는 모든 정리가 어떤 유한 개의 등식으로부터 도출될 수 있음을 말한다. 또한 모든 finitely based equational theory가 모두 결정 가능한 것도 아니다. 이에 대한 반례는 1947년 Markov와 Post에 의해서 최초로 보여졌다[14, 15]. 이러한 반례들 중에서 대표적인 것은 1958년 Curry가 제안한 Combinatory Logic이다[16]. 그러나 1954년 Ackermann은 모든 ground equational theory는 결정 가능함을 보였다[17]. Ground equation이란 variable들 없이 constant function symbol들로만 이루어진 equation을 말한다.

서론에서 본 군(abstract group)을 정의하는 등식체계에 Knuth-Bendix completion procedure를 시행하면 아래의 complete rewrite system을 얻는다.

$$
\begin{aligned}
&i(e) \rightarrow e \\
&*(e, X) \rightarrow X \\
&*(X, e) \rightarrow X \\
&i(i(X)) \rightarrow X \\
&*(i(X), X) \rightarrow e \\
&*(X, i(X)) \rightarrow e \\
&*(i(X), *(X, Y)) \rightarrow Y \\
&*(X, *(i(X), Y)) \rightarrow Y \\
&i(*(X, Y)) \rightarrow *(i(Y), i(X)) \\
&*(*(X, Y), Z) \rightarrow *(X, *(Y, Z))
\end{aligned}
$$

위의 rewrite system은 군에 관한 아래의 몇 가지 기초적인 정리가 성립함을 증명하고 있다:

(1) 항등원의 역원은 항등원이다.
(2) 좌항등원은 동시에 우항등원이다.
(3) 역원의 역원은 자신이다.
(4) 좌역원은 동시에 우역원이다.
(5) XY의 역원은 $Y^{-1}X^{-1}$이다.

5. 결론

등식들로 이루어진 공리 체계로부터 주어진 등식이 논리적 결과로서 성립하는지를 임의의 공리체계와 등식에 대하여 항상 판정하는 것은 불가능하지만 많은 경우에 대답할 수 있는 절차에 대하여 살펴보았다. 이러한 절차가 생성하는 등식들은 그 공리체계에서 성립하는 정리들이다.

현대 대수학에 처음 접하는 학생들이 군에 관한 기초적인 정리들이 컴퓨터 프로그램에 의해서 증명될 수 있다는 것을 알게 되는 것은 학생들의 대수학 또는 논리에 대한 흥미와 이해 및 학습 의욕을 진작시키는 데 적지 않은 도움이 되리라 믿는다.

참고 문헌

1. Church, A. and Rosser, J.B., "Some properties of conversion," *Trans. Amer. Math. Soc.* **39**(1936), 472-482.
2. Dershowitz, N., "Termination of rewriting," *J. of Symbolic Comput.* **3**(1987), 69-115.
3. Lankford, D.S., "On proving term rewriting systems are Noetherian," *Technical*

Report Memo MTP-3. Mathematics Department, Louisiana Tech. University, Ruston, LA, 1979.

4. Dershowitz, N., "Ordering for term rewriting systems," *Theoretical Computer Science* **17**(1982), 279-301.
5. Plaisted, D.A., "Recursively defined ordering for proving termination of term rewriting systems," *Technical Report R78-943*, Dept. of Computer Science, Univ. of Illinois, Urbana, IL, 1978.
6. Lescanne, P., "On the recursive decomposition ordering with lexicographic status and other related orderings," *Journal of Automated Reasoning* **6**(1990), 39-49.
7. Kapur, D., Narendren, P. and Sivakumar, G., "A path ordering for priving termination of term rewriting systems," *Proc. 10th Colloq. Trees in Algebra and Programming* (1985), 173-185.
8. Knuth, D.E. and Bendix, P.B., "Simple word problems in universal algebra," *Computational Problems in Abstract Algebra*, ed. J. Leech. Pergamon Press, Oxford, 1970, 263-297.
9. Newman, M.H.A., "On theories with a combinatorial definition of 'equivalence'," *Ann. Math.* **43**(1942), 223-243.
10. Huet, G., "Confluent reductions: Abstract properties and applications to term rewriting systems," *Journal of ACM* **27**(1980), 797-821.
11. Hermann M. and Privara, "On non-termination of Knuth-Bendix algorithm," *Proc. of the 13th Colloq. on Automata, Languages and Programming*(1986), 146-156.
12. Evans, T., "On multiplicative systems defined by generators and relations," *Proc. Cambridge Philos. Soc.* **47**(1951), 637-649.
13. Taylor, W., "Equational Logic," *Universal algebra*, ed. G. Gratzer. Springer, New York, 1979, 378-400.
14. Markov, A.A., "The impossibility of some algorithms in the theory of associative systems," *Dokl. Akad. Nauk SSSR* **55**(1947), 587-590.
15. Post, E.L., "Recursive unsolvability of a problem of Thue," *J. Symbolic Logic* **13**(1947), 1-11.
16. Curry, H.B. and Feys, R., *Combinatory Logic*, Amsterdam: North-Holland, 1958.
17. Ackermann, W., *Solvable Cases of the Decision Problem*, Amsterdam: North-Holland, 1954.