Available online at www.sciencedirect.com

ScienceDirect

Mathematical and Computer Modelling 44 (2006) 1169–1177

MATHEMATICAL AND COMPUTER MODELLING

www.elsevier.com/locate/mcm

Automated and readable simplification of trigonometric expressions

Hongguang Fu[a,1], Xiuqin Zhong[a,*], Zhenbing Zeng[b]

[a] *Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu 610041, China*
[b] *Software Engineering Institute, East China Normal University, Shanghai 200062, China*

Received 19 August 2005; received in revised form 31 March 2006; accepted 6 April 2006

Abstract

Automated simplification of trigonometric expressions is an important problem that hasn't been completely solved by current computer algebra systems. This paper presents a number of unique prescriptions for the ordering of some trigonometric transformation rules, which have been derived by observing how human experts follow their intuitive rules. We have implemented the procedure in Lisp because of its suitability for formula manipulations and rule-based reasoning systems. Consequently, it can simplify many trigonometric expressions which are even difficult to do by hand, and it achieves much better results for many hard problems than any of Maple, Mathematica, and Maxima do.

Keywords: Simplification of trigonometric expressions; Combination rules; Rule lists; Pattern matching; Automated reasoning; Readable proof

1. Introduction

It is well-known that the simplification of mathematical expressions by computer is one of the most fundamental functions of computer algebra systems. Many experts have devoted themselves to research in the simplification of rational polynomials, factorization of polynomials and integrals of various mathematical expressions. General algorithms for these problems have been implemented in most commonly used software like Maple, Mathematica and Maxima. However, there is a lot of difficulty in the simplification of trigonometric expressions, for which there has not yet been discovered a universal method. Unlike simplification of polynomials such as collection and factorization, simplification of trigonometric expressions is hard to define clearly; nevertheless standards for simplification are accepted through common usage.

Therefore, artificial intelligence methods such as expert systems could be useful for this problem. In this paper, we follow rule-based automated reasoning and present a number of new unique prescriptions for the ordering of some trigonometric transformation rules, which have been derived by observing how human experts follow their intuitive rules. Though this method is not a complete method, it is a very practical approach to simplify trigonometric expressions.

* Corresponding author.
E-mail addresses: fu_hongguang@hotmail.com (H. Fu), zhongxiuqin@hotmail.com (X. Zhong).

[1] Hongguang Fu is with the Automated Reasoning Lab., Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu 610041, China. Tel.: +86 028 85224289; fax: +86 028 85229357.

0895-7177/$ - see front matter
doi:10.1016/j.mcm.2006.04.002

Table 1
Induced formula

	sin	cos	tan	cot
$-\alpha$	$-\sin\alpha$	$\cos\alpha$	$-\tan\alpha$	$-\cot\alpha$
$\pi-\alpha$	$\sin\alpha$	$-\cos\alpha$	$-\tan\alpha$	$-\cot\alpha$
$\pi+\alpha$	$-\sin\alpha$	$-\cos\alpha$	$\tan\alpha$	$\cot\alpha$
$2\pi-\alpha$	$-\sin\alpha$	$\cos\alpha$	$-\tan\alpha$	$-\cot\alpha$
$2k\pi+\alpha$	$\sin\alpha$	$\cos\alpha$	$\tan\alpha$	$\cot\alpha$

Table 2
Values of special angles

α	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$
$\sin\alpha$	0	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$	1
$\cos\alpha$	1	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	0
$\tan\alpha$	0	$\frac{\sqrt{3}}{3}$	1	$\sqrt{3}$	

Firstly, we give the definition for trigonometric expressions, and list some trigonometric transformation rules which will be used in the following parts.

Definition 1 (*Trigonometric Expression*). F is called a trigonometric expression, if F is a real coefficient polynomial of trigonometric functions such as *sin*, *cos*, *tan*, *cot*, *sec* or *csc*.

The basic trigonometric rules cited here are the trigonometric transformations commonly stated in normal textbooks, and expressed as **TR1**, **TR2**, ..., **TR13**:

TR1: Converting *sec*, *csc* to *sin*, *cos*:

$$\sec\alpha = \frac{1}{\cos\alpha}, \quad \csc\alpha = \frac{1}{\sin\alpha}.$$

TR2: Converting *tan*, *cot* to *sin*, *cos*:

$$\tan\alpha = \frac{\sin\alpha}{\cos\alpha}, \quad \cot\alpha = \frac{\cos\alpha}{\sin\alpha}.$$

TR3: Induced formula, refer to Table 1.

TR4: Values of special angles, refer to Table 2.

TR5: Substitution of *sin* square:

$$\sin^2\alpha = 1 - \cos^2\alpha.$$

TR6: Substitution of *cos* square:

$$\cos^2\alpha = 1 - \sin^2\alpha.$$

TR7: Lowering the degree of *cos* square:

$$\cos^2\alpha = \frac{1+\cos(2\alpha)}{2}.$$

TR8: Converting product to sum or difference:

$$\sin\alpha\cdot\cos\beta=\frac{1}{2}[\sin(\alpha+\beta)+\sin(\alpha-\beta)],$$
$$\cos\alpha\cdot\sin\beta=\frac{1}{2}[\sin(\alpha+\beta)-\sin(\alpha-\beta)],$$
$$\cos\alpha\cdot\cos\beta=\frac{1}{2}[\cos(\alpha+\beta)+\cos(\alpha-\beta)],$$
$$\sin\alpha\cdot\sin\beta=-\frac{1}{2}[\cos(\alpha+\beta)-\cos(\alpha-\beta)].$$

TR9: Converting sum or difference to product:

$$\sin\alpha+\sin\beta=2\sin\frac{\alpha+\beta}{2}\cdot\cos\frac{\alpha-\beta}{2},$$
$$\sin\alpha-\sin\beta=2\cos\frac{\alpha+\beta}{2}\cdot\sin\frac{\alpha-\beta}{2},$$
$$\cos\alpha+\cos\beta=2\cos\frac{\alpha+\beta}{2}\cdot\cos\frac{\alpha-\beta}{2},$$
$$\cos\alpha-\cos\beta=-2\sin\frac{\alpha+\beta}{2}\cdot\sin\frac{\alpha-\beta}{2}.$$

TR10: Sum or difference of angles:

$$\sin(\alpha+\beta)=\sin\alpha\cos\beta+\cos\alpha\sin\beta,$$
$$\sin(\alpha-\beta)=\sin\alpha\cos\beta-\cos\alpha\sin\beta,$$
$$\cos(\alpha+\beta)=\cos\alpha\cos\beta-\sin\alpha\sin\beta,$$
$$\cos(\alpha-\beta)=\cos\alpha\cos\beta+\sin\alpha\sin\beta.$$

TR11: Double angle formulas:

$$\sin 2\alpha=2\sin\alpha\cos\alpha,$$
$$\cos 2\alpha=\cos^2\alpha-\sin^2\alpha=2\cos^2\alpha-1=1-2\sin^2\alpha.$$

TR12: Sum or difference of *tan*:

$$\tan(\alpha+\beta)=\frac{\tan\alpha+\tan\beta}{1-\tan\alpha\tan\beta},$$
$$\tan(\alpha-\beta)=\frac{\tan\alpha-\tan\beta}{1+\tan\alpha\tan\beta}.$$

TR13: Product of *tan* or *cot*:

$$\tan\alpha\cdot\tan\beta=1-(\tan\alpha+\tan\beta)\cdot\cot(\alpha+\beta),$$
$$\cot\alpha\cdot\cot\beta=1+(\cot\alpha+\cot\beta)\cdot\cot(\alpha+\beta).$$

In addition, the simplification of rational polynomials is named as **TR0**, and the reverse rules of the above rules are named as $\mathbf{TRi}^{-1}$ ($i=1,\ldots,13$).

2. The measurement of simplification and combination rules

The widely accepted goal of trigonometric simplification is converting a trigonometric expression into a single trigonometric function or the product of several such functions. To depict the standard more clearly, we introduce a definition for the length of a trigonometric expression. (For a formal definition of the concept of simplification, and the minimal description length of general expressions, refer to [1].)

Definition 2 (*The Length of a Trigonometric Expression*). If F is a trigonometric expression, L(F) means the number of monomials contained in F which is regarded as a polynomial of *sin*, *cos*, *tan*, *cot*, *sec* or *csc*.

TRi(F) stands for the resulting expression of applying **TRi** to F. By comparing L(**TRi**(F)) with L(F), taking or rejecting **TRi**(F) can be decided according to necessities.

Just like auxiliary lines and auxiliary points in geometric theorem proofs, many difficult problems can be solved with these special technologies. Therefore, in order to improve the capability of automated simplifications, we bring forward many techniques for some particular patterns in trigonometric simplifications which are realized by several trigonometric rules simultaneously. In order to make it clear, we need to introduce the concept of combination rules.

Definition 3 (*Combination Rules*). An ordered combination of a group of basic rules which can accomplish a specific technique of simplification with conditions, is called a combinatorial transformation rule, **CTR** = {**STR**|**C**}, in which **STR** is a subset of {**TR1**, **TR2**, ..., **TR13**}, and **C** is the condition of taking or rejecting **STR**.

Some commonly used techniques for the simplification of trigonometric expressions are noted by observing how human experts follow their intuitive rules, and they are expressed as four combination rules **CTR1**, **CTR2**, **CTR3**, **CTR4** respectively. These combination rules will be used in our following automated reasoning.

CTR1: Combination rule 1.

Intuitive rules = {**TR5**(Substitution of *sin* square), **TR6**(Substitution of *cos* square), **TR0**(Simplification of rational polynomials)}

$$F \xrightarrow{\{\mathbf{TR5},\mathbf{TR0}\}} F1, \quad F \xrightarrow{\{\mathbf{TR6},\mathbf{TR0}\}} F2.$$

If $L(F1) < L(F)$ and $L(F1) \le L(F2)$, then $\mathbf{TR}(F) = F1$.
If $L(F2) < L(F)$ and $L(F2) \le L(F1)$, then $\mathbf{TR}(F) = F2$.
Or else $\mathbf{TR}(F) = F$.
Example:
$\sin^4(\alpha) - \cos^2(\beta) + \sin^2(\beta) + 2\cos^2(\alpha)$
$= 2 - 2\cos^2(\beta) + \cos^4(\alpha)$ (Substitution of *sin* square, expanding and combination.)

CTR2: Combination rule 2.

Intuitive rules = {**TR5**(substitution of *sin* square), **TR6**(substitution of *cos* square), **TR11**(double angle formulas)}

$$F \xrightarrow{\{\mathbf{TR11},\mathbf{TR5}\}} F1, \quad F \xrightarrow{\{\mathbf{TR11},\mathbf{TR6}\}} F2, \quad F \xrightarrow{\{\mathbf{TR11}\}} F3.$$

If $L(F1) < L(F3)$ and $L(F1) \le L(F2)$, then $\mathbf{TR}(F) = F1$.
If $L(F2) < L(F3)$ and $L(F2) \le L(F1)$, then $\mathbf{TR}(F) = F2$.
Or else $\mathbf{TR}(F) = F3$.
Example:
$\frac{1}{2} - \frac{1}{2}\cos(2\beta)$
$= \sin^2(\beta)$ (double angle of *cos*, substitution of *cos* square)

CTR3: Combination rule 3.

Intuitive rules = {**TR8**(converting product to sum or difference), $\mathbf{TR10}^{-1}$(reverse of sum or difference of angles)}

$$F \xrightarrow{\{\mathbf{TR8}\}} F1, \quad F \xrightarrow{\{\mathbf{TR8},\mathbf{TR10}^{-1}\}} F2.$$

If $L(F2) < L(F)$, then $\mathbf{TR}(F) = F2$.
If $L(F1) < L(F)$, then $\mathbf{TR}(F) = F1$.
Or else $\mathbf{TR}(F) = F$.
Example:
$\sin(\alpha)[\cos(\beta) - \sin(\beta)] + \cos(\alpha)[\sin(\beta) + \cos(\beta)]$
$= [\sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta)] + [\cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)]$
$= \sqrt{2}\sin(\alpha + \beta + \frac{\pi}{4})$.
(Converting product to sum or difference, reverse of sum or difference of angles)

CTR4: Combination rule 4.
Intuitive rules = {**TR4**(values of special angles), $\mathbf{TR10}^{-1}$(reverse of sum or difference of angles)}

$$F \xrightarrow{\{\mathbf{TR4}, \mathbf{TR10}^{-1}\}} F2.$$

If $L(F1) < L(F)$, then $\mathbf{TR}(F) = F1$.
Or else $\mathbf{TR}(F) = F$.
Example:
$\frac{\sqrt{3}}{2}\cos\alpha + \frac{1}{2}\sin\alpha$
$= \cos(\alpha - \frac{\pi}{6})$ (values of special angles, reverse of sum or difference of angles)

The introduction of combination rules provides the basis for machine learning. Obviously one can introduce more combination rules, corresponding to some new techniques, to extend the capability of these automated simplifications.

3. Rule-based automated reasoning

During the simplification of trigonometric expressions, human reasoning processes are very complicated and many special skills are applied. It is almost impossible for modern computers to completely simulate human reasoning capabilities, because the process is still an unsolved puzzle and the calculation speed of a computer is extremely limited. However, human reasoning abilities are not innate. They come from postnatal learning and training [2]. In addition to the application of human ingenuity, we can also train a computer by certain approaches and make it partly simulate some fixed human reasoning abilities. These capabilities can be expressed as rule lists by a number of unique prescriptions for ordering, and the computer will utilize these rule lists in automated reasoning [3] to obtain new conclusions.

Unlike typical algebraic simplification methods, the rule-based automated reasoning method is not a complete algorithm. But it can generate a readable process [4], and at the same time it can append the name of the corresponding rule being applied at that step. This helps people verify the whole process step by step, which is most important in mathematics education. In this paper, we adopt the complete pattern matching approach for the rules in the simplification of trigonometric functions [5]; the computer will apply a rule automatically to match variables in the expression only if it satisfies the pattern of the rule's condition. For example, if we apply the formula of converting sum or difference to product to simplify $sin(x + y) + sin(x - y)$, the computer will apply **TR9** to $sin(x + y)$ and $sin(x - y)$ respectively, and then add them together to obtain the result $2sin(x)cos(y)$.

This simplification process tries to follow common experiences and skills used by experts. From observing how they follow their intuitive rules [6], the following three stages can be derived.

The first stage is converting the trigonometric expression to the one with less trigonometric variables (such as *sin*, *cos*, ...) and expanding the expression, and this mainly consists of the following steps:

1. Transform a trigonometric expression into the one with *sin*, *cos* or *tan*.
2. If *tan* is contained, apply the trigonometric transformations about simplifying *tan* to an expression in order to eliminate *tan*, then if *tan* still exists, convert *tan* to the quotient of *sin* and *cos*.
3. Convert sum angles or double angles to single angles where possible.
4. Convert different trigonometric functions to the same trigonometric function.

The rules **TR** called in the first stage generally expand the original expression appropriately in order to call more rules later.

The second stage is lowering the degree and combining trigonometric expressions, and this mainly consists of the following steps:

1. Convert even powers of *sin* to *cos*;
2. Lower degree of *cos*;
3. Combine similar terms.

The rules **TR** called in the second stage need to satisfy $|L(\mathbf{TR}(F)) - L(F)| \leq c$, here we set $c = 3$ according to necessities. If it expands excessively, then stop calling the rule.

The third stage is the convergence and cluster of trigonometric expressions, and this mainly consists of the following steps:

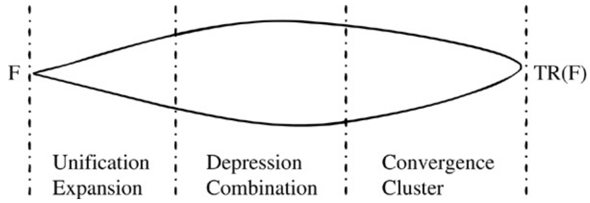

Fig. 1. The simplification process.

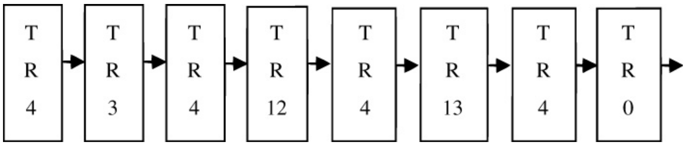

Fig. 2. Rule List 1.

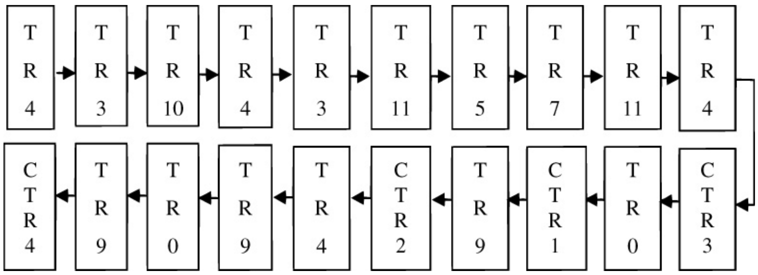

Fig. 3. Rule List 2.

1. Apply converting sum or difference to product;
2. Apply inverse of sum or difference of angles;
3. Apply inverse of double angles;
4. Apply combination rules.

The rule **TR** called in the third stage satisfies $L(\mathbf{TR}(F)) \leq L(F)$, in order to make the length of the simplified expression as small as possible.

In fact, we can illustrate the three stages above as Fig. 1. The above simplification process is standardized, so we can realize it by rule-based automated reasoning, that is, a simplified expression $\mathbf{TR}(F)$ is obtained by calling a series of rules, i.e. $F \xrightarrow{\{\mathbf{RL}\}} \mathbf{TR}(F)$ here **RL** is a rule list including several ordered rules which are selected from $\{\mathbf{TR1}, \mathbf{TR2}, \ldots, \mathbf{TR13}, \mathbf{CTR1}, \ldots, \mathbf{CTR4}\}$ according to the simplification strategy.

It's important to construct a rule list so as to make it approach experts' intuitions for the experiences and skills as much as possible. According to the above three stages and techniques of combination rules, we artificially try a large number of sample training repeatedly, then construct two rule lists **RL1** and **RL2** based on $\{\mathbf{TR1}, \mathbf{TR2}, \cdots, \mathbf{TR13}, \mathbf{CTR1}, \ldots, \mathbf{CTR4}\}$. It seems that the two rule lists are enough to simplify the general trigonometric expressions.

RL1: Rule List 1 simplifies *tan* and *cot*, as Fig. 2.

RL2: Rule List 2 simplifies *sin* and *cos*, as Fig. 3.

In the above rule lists, $\mathbf{RL1} \bigcup \{\mathbf{TR4}, \mathbf{TR3}, \mathbf{TR10}, \mathbf{TR4}, \mathbf{TR3}, \mathbf{TR11}\}$ belongs to the first stage, $\{\mathbf{TR5}, \mathbf{TR7}, \mathbf{TR11}, \mathbf{TR4}\}$ belongs to the second stage, and $\{\mathbf{CTR3}, \mathbf{TR0}, \mathbf{CTR1}, \mathbf{TR9}, \mathbf{CTR2}, \mathbf{TR4}, \mathbf{TR9}, \mathbf{TR0}, \mathbf{TR9}, \mathbf{CTR4}\}$ belongs to the third stage. Every stage will carry out its corresponding simplifications independently. Therefore, it is easy to insert a new rule, or change the order in the rule lists, in order to improve the capability of simplifying trigonometric expressions.

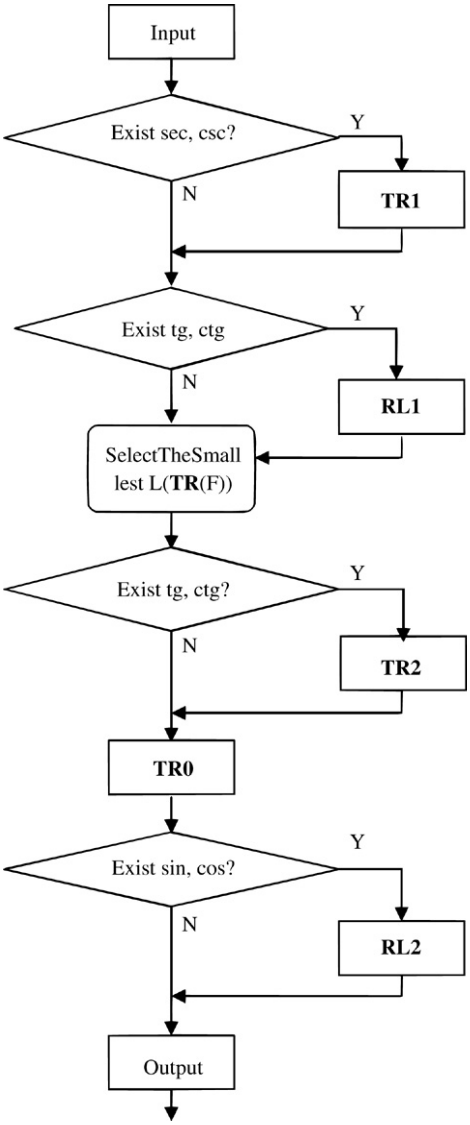

Fig. 4. Flowchart of the simplification.

Based on rule lists **RL1** and **RL2**, we have designed a procedure to simplify trigonometric expressions, and implemented it by Common Lisp. The flowchart of simplification is shown as Fig. 4.

4. Comparison with Maple, Mathematica and Maxima

Maple, Mathematica and Maxima are three excellent computer algebra systems available today, and they each possess very strong powers in the simplification of rational polynomials and trigonometric expressions. Therefore, we select a lot of examples for comparison with them, and find that our method produces results that are close to those performed by hand. The arbitrary three comparison examples are as follows:

Example 1:

Our method:

Simplify: $1 - \frac{1}{4}\sin^2(2x) - \sin^2(y) - \cos^4(x)$

Solution: $= -\cos^4(x) - \frac{1}{4}\sin^2(2x) - \sin^2(y) + 1$

$= 1 - \sin^2(y) - \cos^2(x)\sin^2(x) - \cos^4(x)$ (double angle formula)

$= -\cos^2(x) + \cos^2(y)$ (substitution of *sin* square)
$= \frac{1}{2}\cos(2y) - \frac{1}{2}\cos(2y)$ (lowering the degree of *cos* square)
$= \sin(x+y)\sin(x-y)$ (converting sum or difference to product)

Maple method:

simplify $(1 - 1/4 * sin(2*x)^2 - sin(y)^2 - cos(x)^4, trig)$;
result: $-\frac{1}{4} + \frac{1}{4}\cos(2x)^2 + \cos(y)^2 - \cos(x)^4$

Mathematica method:

FullSimplify $[1 - 1/4 * Sin[2*x]^2 - Sin[y]^2 - Cos[x]^4, Trig]$
result: $\frac{1}{2}(-Cos[2x] + Cos[2y])$

Maxima method:

Trigsimp$(1 - 1/4 * sin(2*x)^2 - sin(y)^2 - cos(x)^4)$;
result: $-\frac{4SIN^2(y) - COS^2(2x) + 4COS^4(x) - 3}{4}$

Example 2:

Our method:

Simplify: $\cos(\frac{\pi}{9})\cos(2\frac{\pi}{9})\cos(3\frac{\pi}{9})\cos(4\frac{\pi}{9})$
Solution:$= \cos(\frac{1}{9}\pi)\cos(\frac{2}{9}\pi)\cos(\frac{1}{3}\pi)\cos(\frac{4}{9}\pi)$
$= \frac{1}{2}\cos(\frac{1}{9}\pi)\cos(\frac{2}{9}\pi)\cos(\frac{4}{9}\pi)$ (values of special angles)
$= \frac{1}{2}\cos(\frac{1}{9}\pi)\cos(\frac{2}{9}\pi)\sin(\frac{1}{18}\pi)$ (induced formula)
$= \frac{1}{16} - \frac{1}{8}\sin(\frac{1}{18}\pi) + \frac{1}{8}\sin(\frac{1}{18}\pi)$ (converting product to sum or difference)
$= \frac{1}{16}$ (expanding and combination)

Maple method:

simplify $(cos(Pi/9) * cos(2*Pi/9) * cos(3*Pi/9) * cos(4*Pi/9), trig)$;
result: $\frac{1}{2}\cos(\frac{1}{9}\pi)\cos(\frac{2}{9}\pi)\cos(\frac{4}{9}\pi)$

Mathematica method:

FullSimplify $[Cos[Pi/9] * Cos[2*Pi/9] * Cos[3*Pi/9] * Cos[4*Pi/9], Trig]$
result: $\frac{1}{16}$

Maxima method:

Trigsimp $(cos(Pi/9) * cos(2*Pi/9) * cos(3*Pi/9) * cos(4*Pi/9))$;
result: $\cos(\frac{Pi}{9})\cos(\frac{2Pi}{9})\cos(\frac{Pi}{3})\cos(\frac{4Pi}{9})$

Example 3:

Our method:

Simplify: $\tan(7\frac{\pi}{18}) + \tan(5\frac{\pi}{18}) - \sqrt{3}\tan(5\frac{\pi}{18})\tan(7\frac{\pi}{18})$
Solution: $= -\sqrt{3}\tan(\frac{7}{18}\pi)\tan(\frac{5}{18}\pi) + \tan(\frac{7}{18}\pi) + \tan(\frac{5}{18}\pi)$
$= \cot(\frac{2}{9}\pi) + \cot(\frac{1}{9}\pi) - \sqrt{3}\cot(\frac{1}{9}\pi)\cot(\frac{2}{9}\pi)$ (induced formula)
$= \cot(\frac{1}{9}\pi)\cot(\frac{2}{9}\pi)\tan(\frac{1}{3}\pi) - \sqrt{3}\cot(\frac{1}{9}\pi)\cot(\frac{2}{9}\pi) - \tan(\frac{1}{3}\pi)$ (Inverse of product of *tan* or *cot*)
$= -\sqrt{3}$ (values of special angles, expanding and combination)

Maple method:

simplify $(tg(7*Pi/18) + tg(5*Pi/18) - 3(1/2) * tg(5*Pi/18) * tg(7*Pi/18), trig)$;
result: $\tan(7\frac{\pi}{18}) + \tan(5\frac{\pi}{18}) - \sqrt{3}\tan(5\frac{\pi}{18})\tan(7\frac{\pi}{18})$

Mathematica method:

FullSimplify $[Tan[7*Pi/18] + Tan[5*Pi/18] - 3(1/2) * Tan[5*Pi/18] * Tan[7*Pi/18]]$
result: $Tan[\frac{7\pi}{18}] + Tan[\frac{5\pi}{18}](1 - \sqrt{3}Tan[\frac{7\pi}{18}])$

Maxima method:

Trigsimp $(tg(7*Pi/18) + tg(5*Pi/18) - 3(1/2) * tg(5*Pi/18) * tg(7*Pi/18))$;
result: $(1 - \mathrm{SQRT}(3)tg(\frac{5Pi}{18}))tg(\frac{7Pi}{18}) + tg(\frac{5Pi}{18})$.

5. Conclusion

According to the comparisons in Section 4, it is clear that our method gets some better results than Maple, Mathematica and Maxima in the simplification of many trigonometric expressions. What's more, our process outputs explanations, which can be verified step by step, and which are helpful both in debugging and in understanding the process. Furthermore, our method can be extended easily by changing the rule lists, so that the simplification functionality can become more powerful.

In our work, we apply complete pattern matching to the rules, but there is still a difference from the manual method. It will greatly improve the capability and efficiency of our simplifications (for example, the flexibility of simplification) if partial pattern matching of rules can be introduced in the future.

Acknowledgements

The authors wish to express gratitude to Jon L. White for his kind help and valuable suggestions.

This work was partially supported by the China NKBRSF Project (Grant No. 2004CB318003) and the "Hundreds Talents Plan" of Institute of Computing Technology. Chinese Academy of Science (20044040).

References

[1] Jacques Carette, International conference on symbolic and algebraic computation, in: Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation, Santander, Spain, 2004, pp. 72–79.

[2] Ruqian Lu, Artificial Intelligence, Science Press, Beijing, 1996.

[3] Wen-tsun Wu, A Mechanization Method of Equations-Solving and Theorem-Proving, in: C.M. Hoffmann (Ed.), Advances in Computing Research, vol. 6, JAI Press Inc., Greenwich, USA, 1992, pp. 103–138.

[4] Shang-Ching Chou, Xiao-Shan Gao, Jing-Zhong Zhang, Machine Proofs in Geometry, World Scientific Publishing, 1994, pp. 101–165.

[5] Thomas Dean, James Allen, Artificial Intelligence: Theory and Practice, Addison Wesley Professional, 1994.

[6] John F. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks/Cole, 2000.