# Rule-based integration: An extensive system of symbolic integration rules

Albert Rich[1], Patrick Scheibe[2], and Nasser M Abbasi[3]

1 Independent researcher, co-author of the Derive computer algebra system **2** Leipzig University, Saxonian Incubator for Clinical Translation, Philipp-Rosenthal-Straße 55, 04103 Leipzig **3** UW-Madison, Madison, WI 53706, USA

Summary

Finding the antiderivative of expressions is often challenging and requires advanced mathematical skills even for simple looking problems. Computer algebra systems (CAS) like Mathematica (Wolfram Research, Inc., Champaign, IL), Maple (Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario), and Maxima (maxima.sourceforge.net) provide integrators to compute antiderivatives symbolically. However, these systems give no insight as to how an antiderivative is found or why it could not be computed. Also, they use advanced methods incomprehensible to humans that often result in huge antiderivatives unnecessarily involving special, or even hypergeometric, functions.

In this work, we present Rubi(The Rubi Organization, 2018a), a rule-based integrator and its implementation in Mathematica(The Rubi Organization, 2018b). Rubi systematically applies an extensive system of symbolic integration rules able to find the optimal antiderivative of a wide variety of mathematical expressions. It currently has over 6600 rules implemented in Mathematica's pattern-matching language. Some of these rules are based on integration *formulas* from (Abramowitz, 2012; Burington, 1973; Gradshteyn, 2014; Zwillinger, 2011) and they are transformed into *rules* that include precise instructions as to when a formula should be applied. Numerous other rules had to be derived during Rubi's development in order to integrate expressions not addressed by formulas in published tables.

The key to the success of Rubi is the rigorous definition of conditions for the integration steps that determine under which circumstances the application of a specific rule is correct and useful. Therefore, Rubi produces optimal antiderivatives that are often dramatically simpler than provided by the commercial CAS integrators.

Rubi is implemented as a Mathematica package that gives the user the option to inspect integration steps and application conditions in detail. An extensive test-suite of over 70,000 integrals with known, optimal antiderivatives is employed to thoroughly test the system before each new release(The Rubi Organization, 2018c). Since it is also useful for testing other symbolic integrators, the entire test-suite is available on Rubi's website(The Rubi Organization, 2018a) expressed in Axiom, Maple, Mathematica, and Maxima syntax. Also on the website are PDF files showing how Rubi and Mathematica's built-in integrator perform on the test-suite. All information about Rubi, i.e. integration rules, source-code, and test-files, are publicly available on its GitHub organization page(The Rubi Organization, 2018d).

However, the value of Rubi goes far beyond its Mathematica implementation. All integration rules are available in human readable form as PDF files or Mathematica notebooks

which contain additional details and references to the relevant literature. Since Rubi's rules in general only require a system for manipulating symbolic expressions by applying pattern-based rules, it is feasible to implement the integration rules in other systems.

That this is indeed possible was shown in the case of Symja(Kramer, 2013), an open-source Java system for symbolic math, where we assisted in the translation and testing of Rubi's integration rules. Now, Rubi is used as the main integration engine in Symja for everything but the most trivial problems. Furthermore, there are efforts to include Rubi into SymPy(Meurer et al., 2017), a Python library for symbolic mathematics, which would directly open the way to use it in Sage(The Sage Developers, 2018), a free and open-source CAS.

Ultimately, the rationale behind Rubi is to make rule-based integration publicly available to enable researchers and enthusiast to find optimal antiderivatives and to verify the correctness of each integration step in the context of their problems.

Short Example

Figure 1 shows the computation of

$$\int \frac{\sec(x)^2 + \sec(x)^2 \cdot \tan(x)}{(2 - \tan(x)) \cdot \sqrt{1 + \tan(x)^3}} \, dx$$

using Rubi when all integration steps are displayed. The specific rules that are applied are shown in red and it is possible to open the rule display to inspect the exact conditions that need to hold to make the transformation valid. In blue, the intermediate expressions are visible and at the end the final antiderivative is returned. It should be noted that the size of the found antiderivative is 25, counting the leafs in the expression tree of the result. In comparison, Mathematica' current version 11.3 returns an antiderivative that has a leaf count of 290 and contains complex terms.

Acknowledgements

References

Abramowitz, I. A., Milton AND Stegun. (2012). *Handbook of mathematical functions.* New York: Courier Corporation.

Burington, R. S. (1973). *Handbook of mathematical tables and formulas* (5th ed.). New York: McGraw-Hill.

Gradshteyn, I. M., I S AND Ryzhik. (2014). *Table of integrals, series, and products.* Amsterdam, Boston: Academic Press.

Kramer, A. (2013). Symja library – java symbolic math system. Retrieved October 25, 2018, from https://github.com/axkr/symja_android_library

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., et al. (2017). SymPy: Symbolic computing in python. *PeerJ Computer Science, 3,* e103. doi:10.7717/peerj-cs.103

In[·]:= `Steps @`
`Int[(Sec[x]^2 + Sec[x]^2 * Tan[x]) /`
`((2 - Tan[x]) * Sqrt[1 + Tan[x]^3]), x]`

Figure 1: Example of computing the antiderivative of an expression, showing intermediate results and applied rules.

Rich et al., (2018). Rule-based integration: An extensive system of symbolic integration rules. *Journal of Open Source Software*, 3(30), 1054. 3
https://doi.org/10.21105/joss.01054

The Rubi Organization. (2018a). Rule-based integration – an extensive system of symbolic integration rules. Retrieved October 25, 2018, from https://rulebasedintegration.org/

The Rubi Organization. (2018b). Mathematica implementation of rubi. Retrieved October 25, 2018, from https://github.com/RuleBasedIntegration/Rubi

The Rubi Organization. (2018c). Mathematica test suite. Retrieved October 25, 2018, from https://github.com/RuleBasedIntegration/MathematicaSyntaxTestSuite

The Rubi Organization. (2018d). Mathematica test suite. Retrieved October 25, 2018, from https://github.com/RuleBasedIntegration

The Sage Developers. (2018). *SageMath, the Sage Mathematics Software System (Version x.y.z).*

Zwillinger, D. (2011). *CRC standard mathematical tables and formulae* (32nd ed.). Boca Raton, Fla: CRC Press.