

私

は、2008年からIPA（情報処理推進機構）に設置されたRuby標準化検討WGの主査を務めている。Rubyはまつもとゆきひろさんが開発し、世界で使われるようになった初めての日本発のプログラミング言語であるので、国際規格のお墨付きをつけてより広く使われるようにすることと、そのためにも言語仕様をきちんと記述するというのがこの標準化の目的である。Ruby処理系の熟知者が言語仕様の原案を作成しWGでそれを検討して修正する、その結果に対するWG外部からの意見も検討する、ということを繰り返して作成した言語仕様が2011年3月にはJIS（プログラム言語Ruby JIS X 3017）として制定され、その英語版を直ちにISO/IECに提案し、2012年4月にISO/IEC 30170として発行された。これで国産の言語が初めて国際規格になった。

ただし、規格になったのはRuby言語の核の部分と基本的なライブラリだけであり、しかも、いままで比較的広く使われていたRubyのバージョン1.8でも、最近使われるようになった1.9でもこの規格に適合していると言えるようにするために、バージョンで異なる機能は「実装依存である」として定義していない。したがって、多くのRubyプログラムはこの規格に適合しているとは言えるが、そのプログラムの正確な動作は必ずしもこの規格で規定されていない。今後標準的なバージョンとなるRubyについて、正確な動作を規定するフルセットの規格を作成する必要がある。

Rubyはオブジェクト指向のスクリプト言語であり、プログラマが楽しくプログラミングできるように設計されている、と言われている。そのためにいろいろな表現が可能になっている。しかし、そのような言語の仕様を正確に書くのは簡単ではない。言語仕様の表現の検討は主としてメールで行ったが、そのメールは8千件を超えている。それだけ最終

的な表現を得るまでに時間がかかった。Rubyの意味規則には動的に決まるものもあるので、抽象的なインタプリタが扱うものを「実行環境」として定義して、その実行環境の状態を使って動的な意味規則を与えている。構文規則も通常の生成規則だけでは表現できず、生成規則の適用条件などを示す特殊な記法も使っている。たとえば、Rubyでは空白や改行の意味の有無が文脈によって異なるので、それを特殊な記法で表現している。それらも含めた構文規則全体が正確に書かれていることを確認するのは容易

基
般

[シニアコラム]

IT好き放題



[No.19]

プログラミング言語 Ruby の国際標準化

ではない。そこで、それを確認するために、その記述に従ったパーサ（構文解析プログラム）を個人的に書いてみようと思った。しかし、このような特殊な記法を受け付けるパーサ生成系はないので、既存のあるパーサ生成系を改造してみたが成功しなかった。その後、パーサを書くのにパーサ・コンビネータを使う方法もあることを、千葉滋さんの著書「2週間できる！スクリプト言語の作り方」の原稿で知り、Scala言語のパーサ・コンビネータを使えば書けるのではないかと思って、ネットでその入門資料を公開していた水島宏太さんに教えてもらいながら、書いてみた^{☆1}。その結果、Rubyの規格文書の生成規則に従って書くことでパーサができることを確認した。またその過程で、生成規則のバグも発見した。

いろいろ苦労はしたが、WGの皆さんの協力でRuby言語の基本部分の仕様を厳密に記述することができて、それが国際規格となった。これでRubyがさらに広く使われるようになることを期待している。なお、先にも述べたように、今後Rubyプログラムの動作を正確に解釈できるようなより完全な規格を作成する必要がある。

（2012年4月16日受付）

中田育男 Ikuo NAKATA

[名誉会員] nakata@kamakuranet.ne.jp

筑波大学名誉教授。1958年東京大学理学部数学科卒業。1960年同大学院修士課程修了、同年日立製作所入社。1979年から2008年まで筑波大学、図書館情報大学、法政大学の教授歴任。コンパイラに関する著書が何冊かある。

☆1 https://github.com/inakata/ruby_scala