Message Automata for Messages with Variants, and Methods for their Translation

Christian Boitet

GETA, laboratoire CLIPS,
385 rue de la bibliothèque - BP 53, 38041 Grenoble Cedex 9 - France
Christian.Boitet@imag.fr

Abstract. In messages with variables and variants, such as "the $n-[st|nd|rd|th] trial was successful, and the $p file[|s] found [is|are] satisfactory.", variable types are specific (cardinal, ordinal, politeness…) and induce different "variant cases" in each language. Controlled loop-free FSAs, called here "message automata" (MAs), are proposed to model such messages. To translate a MA, one generates an instance of it for each possible variant in the target language. After translation, the values used in the instances are discarded and a target language MA is built by factorization (not classical minimization), using an original dynamic programming algorithm. A library for handling catalogues of MAs, GetAMsg, has been implemented in C, and can be used from many usual programming languages. A still speculative idea is to use a UNL graph conform to the official specifications, but with some special conventions, to represent a message with variables, and generate the language-specific MAs from it.

1 Introduction

Software of all kinds needs to be localized in dozens of languages because of the increasing availability of PCs and multilinguality of the web. Elements to be translated are often dynamic, because they depend on the values of some variables. We will call them "variables with variables and variants" (MW). This includes not only "classical" short messages, such as "$n files have been processed", which give rise to singular/plural variants in English and other languages, but also more personalized and often longer messages, such as paragraphs in commercial offers, or short texts in games, or sentences from online documentation, where several variables and different kinds of variants (direct/polite, masculine/feminine/neutral, calendar…) can appear.

MWs are linguistically interesting, because variable types are specific (cardinal, ordinal, hour of day…) and induce different "variable cases" (classical, not grammatical cases) in each language. For instance, there are 3 cases in Russian for cardinals (1 год, 2 года, 5 лет, 21 год…), and 3 different cases (singular, dual, plural) in Arabic.

We will call *format* a string pattern like the example above, which may contain variables such as $n, and generate a potentially infinite set of message instances after variable substitution. A format may also contain formatting commands such as "%3i" (3 character place holder for an integer) or "\t" (tab) in C.