In the last decade the number of digital information sources has been risen exponentially. Most of these sources are not structured and consequently they are unsuited for automatic manipulation and reasoning. Therefore, a lot of effort has been devoted, in the area of Information Extraction (IE), for structuring these data automatically. Several approaches have been proposed but they (i) ignore the semantics of information they extract, (ii) are dependent on the document format (html, pdf, txt, ...), and (iii) are mainly syntactic. In this work we lay out the fundamentals of a logic-based approach for Semantic Information Extraction and we implemented the results in HiLeX, an advanced system for ontology-based information extraction, applied in relevant real-world applications such as: (i) Information extraction from balance sheets; (ii) Information extraction from news; (iii) Information extraction from clinical documents. In each case good results have been obtained. In particular, our approach uses ontologies for representing knowledge in a specific domain and formal grammars rules to discover objects in different type of documents. Ontologies are represented by the OntoDLP language because its object-oriented mechanism for describing knowledge perfectly fits with extraction rules. OntoDLP is a Disjunctive Logic Programming (DLP) language with object-oriented features including, besides the concept of relation, the object-oriented notions of class, object (class instance), object-identity, complex object, (multiple) inheritance, and the concept of modular programming by means of reasoning modules. In particular, the last feature (provided by the OntoDLV system) allows us to encode formal grammar rules in logic rules computed together with domain ontologies. Formal grammars rules extend classical grammars and regular expressions (massively used in IE as a convenient mean for describing search patterns) with semantic features. Regular grammars (RGs), indeed, offer a simple and declarative way to specify patterns to be extracted, and are suitable for efficient evaluation. However, there are simple extraction patterns that are relevant for IE while cannot be expressed by an RG. To express patterns of this kind, RGs can be enhanced by attributes, storing information at each application of a production rule. Unfortunately, the complexity of grammars with attributes is sensibly harder than in the attribute-free case; for instance, attributes on RGs lead to Exptime-hardness even in the simple case of deterministic grammars using only two attributes, if string concatenation is a possible attribute operation. Nonetheless, attribute grammars (combined with semantic domains, such as ontologies) offer a very natural and declarative way to describe object-oriented patterns for semantic IE. Thus, a careful complexity analysis, leading to the identification of tractable cases of attribute grammars where complexity and expressiveness are well-balanced, is of utmost importance, and is carried out in this work. We consider RGs with attributes ranging over integers. We analyze the complexity of the classical problem of deciding whether a string belongs to the language generated by an attribute grammar of a given class C (call it PARSE[C]), looking for efficiently computable subcases. We consider deterministic and ambiguous regular grammars, attributes specified by arithmetic expressions over {| |, +, -, \, % ,*}, and a possible restriction on the attributes composition (that we call "strict" composition). We single out many interesting tractable cases. Deterministic RGs with attributes computed by any arithmetic expression over {| |, +, -, \, %} are tractable. In particular, if the attribute composition is strict, then PARSE[C] is L-complete; otherwise it is P-complete. Problem PARSE[C] is NL-complete for general (possibly ambiguous) regular grammars with attributes strictly composed over {| |, +, -, \, %}. If strict composition is guaranteed, then PARSE[C] remains tractable (P-complete) even if the arithmetic operator '*' is allowed. Problem PARSE[C] becomes NP-complete for general regular grammars over {| |, +, -, \, % ,*} with strict composition and for grammars over {| |, +, -, \, %} without any restriction on the attribute composition. Finally, the problem is in PSPACE, in the most general case of a regular grammar with attributes specified by any expression over {| |, +, -, \, % ,*}. Since regular expressions are very important in IE, another developed task concerns the problem of decomposing a regular expression as composition of regular expressions. Such a task is useful for ``refactoring" regular expressions in order to classify them in an ontological way. Also here some interesting results have been obtained.