atomic formula

Canonical name	AtomicFormula
Date of creation	2013-03-22 12:42:54
Last modified on	2013-03-22 12:42:54
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	10
Author	CWoo (3771)
Entry type	Definition
Classification	msc 03C99
Classification	msc 03B10
Synonym	quantifier free formula
Related topic	TermsAndFormulas
Related topic	CNF
Related topic	DNF
Defines	literal
Defines	clause
Defines	quantifier-free formula
Defines	positive literal
Defines	negative literal

Let Σ be a signature and $T(\Sigma)$ the set of terms over Σ. The set S of symbols for $T(\Sigma)$ is the disjoint union of Σ and V, a countably infinite set whose elements are called *variables*. Now, adjoin S the set $\{=, (,)\}$, assumed to be disjoint from S. An *atomic formula* φ over Σ is any one of the following:

1. either $(t_1 = t_2)$, where t_1 and t_2 are terms in $T(\Sigma)$,
2. or $(R(t_1, ..., t_n))$, where $R \in \Sigma$ is an n-ary relation symbol, and $t_i \in T(\Sigma)$.

Remarks.

1. Using atomic formulas, one can inductively build formulas using the logical connectives $\vee$, $\neg$, $\exists$, etc... In this sense, atomic formulas are formulas that can not be broken down into simpler formulas; they are the building blocks of formulas.
2. A *literal* is a formula that is either atomic or of the form $\neg\varphi$ where φ is atomic. If a literal is atomic, it is called a *positive literal*. Otherwise, it is a *negative literal*.
3. A finite disjunction of literals is called a *clause*. In other words, a clause is a formula of the form $\varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_n$, where each φ_i is a literal.
4. A *qunatifier-free formula* is a formula that does not contain the symbols $\exists$ or $\forall$.
5. If we identify a formula φ with its double negation $\neg(\neg\varphi)$, then it can be shown that any quantifier-free formula can be identified with a formula that is in conjunctive normal form, that is, a finite conjunction of clauses. For a proof, see this `http://planetmath.org/EveryPropositionIsEquivalentToAP`