

Predicates	Argument terms	What does it do	Notes		Used when?
=		Unification			Used in each predicate call
\+( \+ X = Y)		"Unifies" test. Unification with drops the MGU after success.			
\=		"Not-unifies" test. Implemented as \+( X = Y)			
@Term1 == @Term2	Arbitrary terms, may resolve to ground or nonground terms.	Term equivalence	Strong statement. If two terms are term-equivalent, they stay term-equivalent past the point of test.		Rarely used.
@Term1 \== @Term2		Term non-equivalence Implemented as \+(Term1 == Term2)	Weak statement. If two terms are term-non-equivalent, they can become term-equivalent after the point of test through unification.		== Can be used to compare a term to another fully ground term. Avoid.
-Number is +Expr	The RHS must resolve to a fully evaluable numeric expression at call time. The LHS is generally an unbound variable.	Numeric function evaluation: Evaluate RHS, unify with (not "set to") LHS.	This is weak sauce!  is should be a general-purpose function call, especially for obtaining stuff from the nonlogical environment (there already is "random/1" and "cputime/0"), not only a numeric evaluation call.	Confusingly functions and predicates have the same notation: functor/arity.  Bitwise operations (on integers) are also supported.	Whenever numeric evaluation needs to be performed.
+Expr1 := +Expr2	Terms must resolve to fully evaluable arithmetic expressions at call time.	Numeric equality	These predicates take away the hassle of type conversion, but are otherwise not very "logical" because LHS and RHS must resolve to fully ground terms at call time.	There are functions getting random numbers, cputime and IEEE 754 NaN.	In guard expressions.
+Expr1 \= +Expr2		Numeric non-equality			
>/2, </2, =</2, >=/2		Other numeric comparison predicates			
?X #= ?Y		Arithmetic equality constraint in CLP(FD)		Can't have logic programming if you don't at least have CLP(FD)	Use this as a proper logic programming alternative to is and := and \=
?X #\= ?Y		Arithmetic non-equality constraint in CLP(FD)			
#>/2, #</2, #=</2, #>=/2		Other numeric comparison predicates in CLP(FD)			
dif(@A, @B)		"Impossible-to-unify" constraint		Classed under "corouting"	