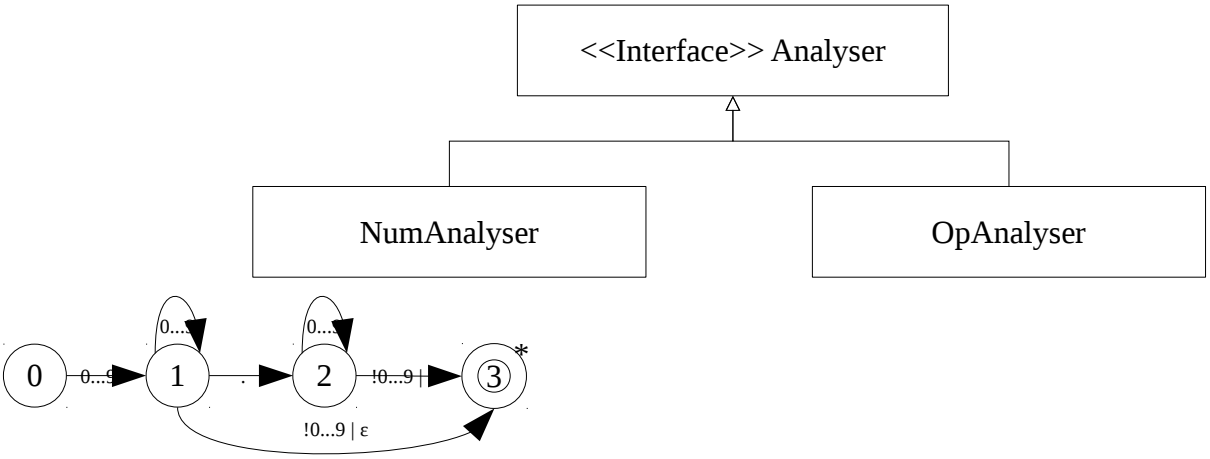

StartChar → Expr	{ StartChar.val = Expr.val; }
Expr → Term S	{ S.inh = Term.val; Expr.val = S.val; }
S → R S_1	{ R.inh = S.inh; S_1.inh = R.val; S.val = S_1.val; }
S → **ε**	{ S.val = S.inh; }
R → + Term	{ R.val = R.inh + Term.val; }
R → - Term	{ R.val = R.inh – Term.val; }
Term → Factor U	{ U.inh = Factor.val; Term.val = U.val; }
U → T U_1	{ T.inh = U.inh; U_1.inh = T.val; U.val = U_1.val;}
U → **ε**	{ U.val = U.inh; }
T → ***** Factor	{ T.val = T.inh * Factor.val; }
T → / Factor	{ T.val = T.inh / Factor.val; }
Factor → **digit** V	{ V.inh = **digit**.val; Factor.val = V.val; }
Factor → (Expr) V	{ V.inh = Expr.val; } Factor.val = V.val; }
Factor → -$Factor_1$ V	{ V.inh = -Factor.val; } Factor.val = V.val; }
Factor → +$Factor_1$ V	{ V.inh = Factor.val; } Factor.val = V.val; }
V → ^ Factor	{ V.val = pow(V.inh, Factor.val); }
V → **ε**	{ V.val = V.inh }

（注：*S* 即为 *ExprConcat*， *R* 即为 *ExprSuffix*， *U* 为 *TermConcat*， *T* 为 *TermSuffix*,*V* 为 *FactorSuffix*）
（再注：包含空符号的产生式不抛出 *NotMatch* 而直接返回调用者）