

CFGrammar

– derivationTree:Tree<const Symbol*>

+ CFGrammar()
+ CFGrammar(const Genotype &)
+ CFGrammar(const Phenotype &)
+ CFGrammar(const CFGrammar &)
+ ~CFGrammar(): virtual
+ readBNFFile(const char *): bool
+ readBNFFile(const string &): bool
+ readBNFString(const char *): bool
+ readBNFString(const string &): bool
+ addBNFString(const char *): bool
+ addBNFString(const string &): bool
+ outputBNF(ostream &): void
+ findRule(const Symbol &): Rule *
+ getDerivationTree(): virtual const DerivationTree* = 0
– isRecursive(vector<Symbol *>&, Rule*): bool
– updateRuleFields(): void
– clearRuleFields(): void
– genotype2Phenotype(): virtual bool = 0
– phenotype2Genotype(): virtual bool = 0