

BNF Specification of CoRE Planner 1.0

July 5, 2005

This appendix contains a complete Backus-Naur Form specification of the CoRE Planner language.

```
< letter >          ::= [ "a"-"z", "A"-"Z", "_", "-", "#" ];
< digit >           ::= [ "0"-"9" ];
< number >          ::= < digit > +;
< char >            ::= < letter > | < digit >;
< symbol >          ::= < letter > ( < char > * );
< alias >           ::= < symbol >;
< problem_name >    ::= < symbol >;
< domain_name >     ::= < symbol >;
< problem >         ::= "(defproblem " < problem_name > < domain_name >
                        < initial_state > [ < task_list > ] )";
< initial_state >   ::= "( " < literal > * ")";
< domain_rule >     ::= < axiom > | < operator > | < method >;
< domain >          ::= "(defdomain " < domain_name > "( "
                        < domain_rule > * ") )";
< operator >        ::= "(operator " < operator_head > < preconditions >
                        < del > < add > [ < number > ] )";
< preconditions >   ::= < conjunction >;
< del_list >        ::= < conjunction >;
```

< <i>add_list</i> >	::= < <i>conjunction</i> >;
< <i>method</i> >	::= "(:method " < <i>method_head</i> > (< <i>decomposition</i> > *) ")";
< <i>method_head</i> >	::= "(" < <i>symbol</i> > (< <i>term</i> > *) ")";
< <i>operator_head</i> >	::= "(!" < <i>symbol</i> > (< <i>term</i> > *) ")";
< <i>decomposition</i> >	::= "(" [< <i>alias</i> >] < <i>preconditions</i> > < <i>task_list</i> > ")";
< <i>task_list</i> >	::= " nil " "(" < <i>task_atom</i> > * ")";
< <i>task_atom</i> >	::= < <i>operator_head</i> > < <i>method_head</i> >;
< <i>axiom</i> >	::= "(:-" < <i>literal</i> > < <i>conjunction_list</i> > ")";
< <i>conjunction_list</i> >	::= [< <i>alias</i> >] < <i>conjunction</i> >;
< <i>conjunction</i> >	::= " nil " < <i>term</i> > "(" [" :first "] < <i>literal</i> > * ")";
< <i>literal</i> >	::= "(not " < <i>logical_atom</i> > ")" < <i>logical_atom</i> >;
< <i>logical_atom</i> >	::= < <i>term</i> > "(" < <i>symbol</i> > (< <i>term</i> > *) ")" "(~" < <i>term</i> > * ")";
< <i>term</i> >	::= < <i>variable</i> > < <i>constant</i> > < <i>list_term</i> > < <i>call_term</i> > < <i>function_term</i> >;
< <i>variable</i> >	::= "?" < <i>symbol</i> >;
< <i>constant</i> >	::= < <i>symbol</i> > < <i>number</i> >;
< <i>list_term</i> >	::= "(list)" "(list " < <i>term</i> > < <i>sublist</i> > ")";
< <i>sublist</i> >	::= "." < <i>term</i> > < <i>term</i> > < <i>sublist</i> >;
< <i>call_term</i> >	::= "(call " < <i>fctn</i> > < <i>term</i> > < <i>term</i> > ")" "(~ call " < <i>bool_fctn</i> > < <i>term</i> > < <i>term</i> > ")";
< <i>function_term</i> >	::= "(" < <i>symbol</i> > (< <i>term</i> > *) ") "(~" < <i>term</i> > * ")";
< <i>fctn</i> >	::= < <i>bool_fctn</i> > "*" "/" "+" "-" cat ;
< <i>bool_fctn</i> >	::= ">" "<"

```
| ">="
| "<="
| "="
| "! ="
| "min"
| "max"
| "member";
```