

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>;
```

```

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

```
| "≥"
| "≤"
| "≡"
| "≠"
| "min"
| "max"
| "member";
```