

## BNF clauses for state-dependent action costs

Most of the syntax is already provided by the PDDL 3.1 language. Only *sum* and *prod* are new language constructs.

Note: The compiler currently does not support real numbers and exists/forall

```
<sdac-term>      ::= <arithm-term>
                  | <logical-term>

<arithm-term>    ::= <number>
                  | (<binary-op> <sdac-term> <sdac-term>)
                  | (<multi-op> <sdac-term> <sdac-term>+)
                  | (- <sdac-term>)
                  | (sum (<typed list(variable)>) <sdac-term>)
                  | (prod (<typed list(variable)>) <sdac-term>)

<logical-term>   ::= <atomic formula(term)>
                  | (not <logical-term>)
                  | (and <logical-term>*)
                  | (or <logical-term>*)
                  | (exists (<typed list(variable)>) <logical-term>)
                  | (forall (<typed list(variable)>) <logical-term>)

<binary-op>      ::= - | /
<multi-op>       ::= * | +

<typed list (x)>  ::= x*
<typed list (x)>  ::= typingx+ - <type> <typed list(x)>

<variable>       ::= ?<name>
<atomic formula(t)> ::= (<predicate> t*)
<predicate>      ::= <name>
<term>           ::= <name> | <variable>

<name>           ::= <letter> <any char>*
<letter>         ::= a..z | A..Z
<any char>       ::= <letter> | <digit> | - | _
<number>         ::= <digit>+ [<decimal>]
<digit>          ::= 0..9
<decimal>        ::= .<digit>+
```