

phrase/3

Description

```
phrase(NonTerminal, Input, Rest)
```

True if the list `Input` can be parsed using the specified non-terminal `NonTerminal`. The list `Rest` is what remains of the list `Input` after parsing succeeded. This method also accepts grammar rule bodies in the first argument (with the limitation that cuts are only supported in local calls).

Template and modes

```
phrase(+callable, ?list, ?list)
```

Errors

NonTerminal is a variable:

```
instantiation_error
```

NonTerminal is neither a variable nor a callable term:

```
type_error(callable, NonTerminal)
```

Input is neither a partial list nor a list:

```
type_error(list, Input)
```

Rest is neither a partial list nor a list:

```
type_error(list, Rest)
```

The grammar rule non-terminal NonTerminal is private:

```
permission_error(access, private_non_terminal, NonTerminal)
```

The grammar rule non-terminal NonTerminal is protected:

```
permission_error(access, protected_non_terminal, NonTerminal)
```

The grammar rule non-terminal NonTerminal is not declared:

```
existence_error(non_terminal_declaration, NonTerminal)
```

Examples

```
| ?- sentence::phrase(noun_phrase, [the, girl, likes, the, boy], Rest).  
  
Rest = [likes, the, boy]  
yes
```

Copyright © Paulo Moura — Logtalk.org
Last updated on: October 26, 2005