

Prolog #3 - Liste

📅 Tue, 18 Oct

Liste

Una **lista** è una **sequenza di elementi** separati da **virgole**.

```
[a, b, c]
```

Le liste sono l'unica struttura dati utilizzabile in Prolog.

Testa e Coda

Una lista può essere divisa in **due** parti, testa e coda.

La **testa** della lista è il primo elemento della lista.

La **coda** della lista è tutta la lista tranne la sua testa.

Esempi di testa e coda

<code>[a, b, c]</code>	<code>a</code> è la testa, <code>[b, c]</code> la coda
<code>[a, b]</code>	<code>a</code> è la testa, <code>[b]</code> la coda
<code>[a]</code>	<code>a</code> è la testa, <code>[]</code> la coda
<code>[[a]]</code>	<code>[a]</code> è la testa, <code>[]</code> la coda
<code>[[a, b], c]</code>	<code>[a, b]</code> è la testa, <code>[c]</code> la coda
<code>[[a, b], [c], d]</code>	<code>[a, b]</code> è la testa, <code>[[c], d]</code> la coda

Operatore barra |

L'operatore **barra |** divide la lista in due, ed è usato per distinguere la **prima parte** dalla **secoda**.

Esempi di unificazione che sfruttano l'operatore barra

```
1. ?- [X | Ys] = [mia, vincent, jules, yolanda]
```

```
True.  
  X = mia  
  Ys = [vincent, jules, yolanda]
```

```
2. ?- [X, Y | Zs] = [the, answer, is, 42]
```

```
True.  
  X = the
```

```
Y = answer  
Z = [is, 42]
```

```
3. ?- [X, 42 | _] = [41, 42, 43, foo(bar)]
```

_ è la variabile di indifferenza, ovvero indica che ciò che sostituisce non è di alcun interesse.

```
True.  
X = 41
```

```
4. ?- [X, 42, _] = [41, 42, 43, foo(bar)]
```

```
False.
```

In questo caso il compilatore restituisce `False` perchè senza l'operatore barra, la lista non viene divisa e dunque `_` dovrebbe essere unificato con un singolo elemento, mentre invece sono presenti sia `43` che `foo(bar)`. Nel caso precedente [3.] `_` veniva unificato invece con la lista contenente il resto della lista.
