Lists as data structure (1)

Lists are one of the most common data structures in computational linguistics, especially in connection with recursive programs.

Term	Abbreviation	AVM
`e_list`	`[]`	$\langle\rangle$
`ne_list`	`[_|_]`	$\left[ne\_list\right]$

Lists as data structure (2)

Term	Abbreviation	AVM

`(ne_list, hd:a, tl:e_list)`
 	
`(hd:a, tl:  e_list)`
	`[a]`
	
	`[a|[]]`
		$\begin{bmatrix} \textit{ne\_list} & \\ \text{HD} & a \\ \text{TL} & \textit{e\_list} \end{bmatrix} \begin{bmatrix} \textit{ne\_list} & \\ \text{HD} & a \\ \text{TL} & \langle\rangle \end{bmatrix}$

`(ne_list, hd:a, tl:`
`   (ne_list, hd:b, tl:e_list))`
 	
`(ne_list, hd:a, tl:`
`            (hd:b, tl:e_list))`
 	
`(hd:a, tl:  (hd:b, tl:e_list))`
	`[a,b]`
	
	`[a|[b]]`
	
	`[a|[b|[]]]`
		$\begin{bmatrix} \textit{ne\_list} & \\ \text{HD} & a \\ \text{TL} & \begin{bmatrix} \textit{ne\_list} & \\ \text{HD} & b \\ \text{TL} & \textit{e\_list} \end{bmatrix} \end{bmatrix}$

Lists as data structure (3)

Term	Abbreviation	AVM
`ne_list, hd:a, tl:list`	`[a|_]`	$\begin{bmatrix} \textit{ne\_list} & \\ \text{HD} & a \end{bmatrix}$
`hd:a, tl:list`		
`hd:a, tl:X`		
`hd:a, tl:_`		
`hd:a`		

Lists as data structure (4)

Term	Abbreviation	AVM

`(ne_list, hd:X, tl: (X,list))`
`(hd:X, tl: (X,list))`
`(hd:X, tl:X)`
	`[X|X]`	$\begin{bmatrix} \textit{ne\_list} & \\ \text{HD} & \boxed{1} \\ \text{TL} & \boxed{1} \end{bmatrix}$

`(hd: (hd:a, tl:X), tl: X)`	`[[a|X]|X]`	$\begin{bmatrix} \textit{ne\_list} & \\ \text{HD} & \left\langle a \mid \boxed{1} \right\rangle \\ \text{TL} & \boxed{1} \end{bmatrix}$