Let $S = \{tl: \forall'a,'a\ list \to' a\ list, (::): \forall\ 'a,\ 'a \to' a\ list \to' a\ list, (=): \forall\ 'a,\ 'a \to' a \to bool\}$

Let $R = S \cup \{length: \forall\ 'a,\ 'a\ list \to list\}$

IF_THEN_ELSE		R |- "a": string	R|- []: list	
$R \cup \{lst: \forall'a,\ 'a\ list\}$|- if lst =[] then 0 else 1 + length (tl(lst)): int		R |- ("a"::[]) : list		R |- []: list
R|- fun lst -> if lst =[] then 0 else 1 + length (tl(lst)) : 'a list -> int		R |- ("a"::[])::[]) : list		
S |- (let rec length = fun lst -> if lst =[] then 0 else 1 + length (tl(lst)) in length (("a"::[])::[])) : int				

Let T = $R \cup \{lst: \forall'a,\ 'a\ list\}$

If_THEN_ELSE:

					T|- tl: $\forall'a,'a\ list \to' a\ list$	T|- lst: $\forall'a,'a\ list$
					T|- tl(lst): $\forall'a,\ 'a\ list$	
T|- lst: $\forall'a,\ 'a\ list$	T|- (=): $\forall'a,\ 'a \to' a \to bool$	T|- []: list		T|- 1: int	T|- length(tl(lst)):int	
T|- lst =[] : bool			T|- 0: int		T|- 1+ length(tl(lst)): int	
$R \cup \{lst: \forall'a,\ 'a\ list\}$|- if lst =[] then 0 else 1 + length (tl(lst)): int						