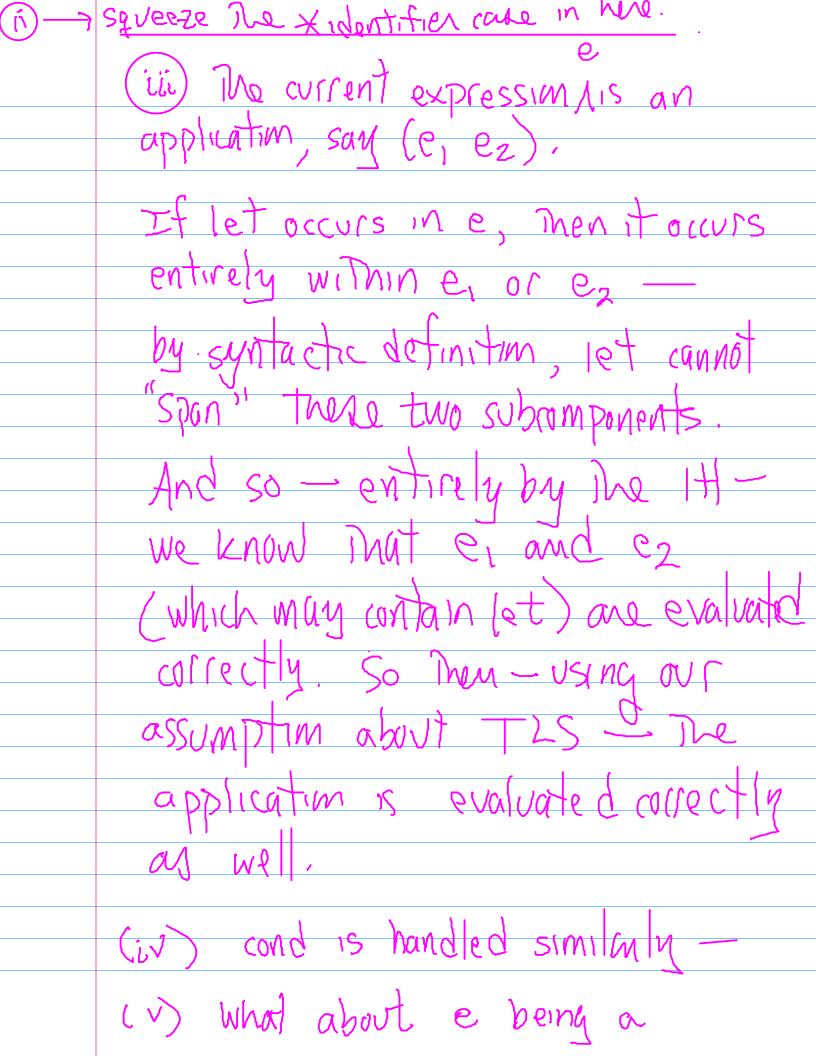
what does The rocrectness proof for 725-let look like? TLS WITH The let form I won't give details on coding the addition of let, except to sury that you need to add a new type — 1e — a new action function, Notos Realize That let "can occur anywhere" you may not base your code adaptations or your proof on the mistaken assumption That any top-level let needs to be dealt with. What is me standard of correctness? Basically, we'll punt: we will say That The good is to prove That TLS-let evaluates its (syntactically correct) inputs to give for more inputs.

	Do we have to give a proof for
	The entire TLS-let system? NO -
	we could - but instead we will assume
	that TLS (without let) works
	correctly [by The same standard]
Lo Du	The proof is structural induction
re	on TLS-18t.
1)1001	numbers
	Basis: booleans . Primitives
	1H: The conclusion is valid for all proper components of the current
	TLS-let expressim
	IS: as you expect - I'll fill in
	IS: as you expect. I'll fill in enough of the details, below.

We are to closed Dust experse TI S- lot out mere
We are to show mut every TLS-let expression is correctly evaluated by our (extended) interpreter.
int or who controlled by our controlled
MIGI hie 161.
Basis: numbers, booleans and primitives
all belong to TLS. So - by our assumptions are correctly evaluated.
They are correctly overland
1, 2) was coll & cital Exalogible.
1H; as above
IS: (By cases)
i The current exp. is (quote e),
Where e is in TLS-let. Referring
to The specific coding of the system,
ve argue mat (quote e) 15
reduced to (xquote e)
Which returns e - according to
Dag unaplamantation of the
The implementation of X quote.  And This is exactly what R5Rs would do.
THE INIS IS EXECTLY MANY IS IS 2 MINICI OF.



lambou expression? We step Through the evaluation to get to Xambon minich creates a closure and any occurrence of letis entirely in the body of the correct out at Dhis stage. So

For this, we'd to argue mut our xlet function is correct. Details lett for you