

1

a) $S \Rightarrow_{rm} (L) \Rightarrow_{rm} (L, S) \Rightarrow_{rm} (L, (L)) \Rightarrow_{rm} (L, (L, S)) \Rightarrow_{rm} (L, (L, \underline{a})) \Rightarrow_{rm} (L, (\underline{S}, a)) \Rightarrow_{rm} (L, (\underline{a}, a)) \Rightarrow_{rm} (\underline{S}, (a, a)) \Rightarrow_{rm} (\underline{a}, (a, a))$

b)		(a, (a,a))	shift
	(a, (a,a))	shift
	(a	, (a,a))	reduce
	(S	, (a,a))	reduce
	(L	, (a,a))	shift
	(L ,	(a,a))	shift
	(L, (a,a))	shift
	(L, (a	,a))	reduce
	(L, (S	, a))	reduce
	(L, (L	, a))	shift
	(L, (L ,	a))	shift
	(L, (L , a))	reduce
	(L, (L , S))	reduce
	(L, (L))	shift
	(L, (L))	reduce
	(L , S)	shift
	(L , S)		reduce
	(L)		reduce
	S		accept
	Stack	Input	Move

c)

2

State 2 has a shift/reduce conflict. When the next input token is else, the parser doesn't know whether it should reduce T according to rule 3 or shift else onto the stack. This is an example of the dangling else problem. For clarity, the state with a conflict is:

$S \rightarrow \text{if } S . T$ $T \rightarrow . \text{ else } S$ $T \rightarrow .$
