Semantics of Programming Languages

Consider the variant of untyped L1 with syntax as below and a standard small-step semantics \( \langle e, s \rangle \rightarrow \langle e', s' \rangle \) (this is identical to L1 except that it has equality testing \( e_1 = e_2 \) on integers instead of \( \geq \) and that here stores are total functions).

Booleans \( b \in B = \{ \text{true}, \text{false} \} \)
Integers \( n \in \mathbb{Z} = \{ ..., -1, 0, 1, ... \} \)
Locations \( \ell \in L = \{ l, l_0, l_1, l_2, ... \} \)
Stores \( s \), total functions from \( L \) to \( \mathbb{Z} \)
Values \( v ::= \text{skip} | n | b \)
Operations \( op ::= = | + \)

Expressions
\[
e ::= \text{skip} | n | b | e_1 \ op e_2 | \text{if} \ e_1 \ \text{then} \ e_2 \ \text{else} \ e_3 | \ell := e | !\ell | e_1; e_2 | \text{while} \ e_1 \ \text{do} \ e_2
\]

Define \( [e] \) to be the function that takes any store \( s \) and either is \( \bot \) (undefined), if \( \langle e, s \rangle \rightarrow \omega \), or is \( \langle v, s' \rangle \), if \( \langle e, s \rangle \rightarrow * \langle v, s' \rangle \).

Define (untyped) semantic equivalence \( e_1 \approx e_2 \) iff \( [e_1] = [e_2] \).

(a) State what it means for \( \approx \) to be a congruence. \hspace{1cm} [2 marks]

(b) For each of the constructs of the expression grammar, define an explicit characterisation of \( [e] \) in terms only of the semantics \( [e'] \) of its subexpressions \( e' \), without using the reduction relation. (For example, for \( n \) (which has no subexpressions) \( [n] = \lambda s. \langle n, s \rangle . \)) \hspace{1cm} [12 marks]

(c) Consider \( (\text{if} \ !\ell = 1 \ \text{then} \ e \ \text{else} \ e) \approx e \). Either prove it, using your answer to part (b), or exhibit a counterexample. \hspace{1cm} [3 marks]

(d) Consider \( (\text{while} \ e_1 \ \text{do} \ e_2) \approx (\text{while} \ e_1 \ \text{do} \ (e_2; e_2)) \) where \( e_1 \) does not read any store locations. State whether this is true or false, with an informal explanation of the possible cases. \hspace{1cm} [3 marks]