Part 1

L = | |

can be writen a\*,b\*

So L is RL.

Part2

W belongs to {a,b}\* .

Set L1 = {w | w{a,b}\*,|a|=|b| }

Assume L1 is RL.

L2 = { } is RL

L1 L2 = L1’ is RL , but L1’ ={ | n >=0 } is not RL.

So L is not RL.

Part3

Regular language is a subset of context-free grammar. Some languages cannot be expressed by regular language, but can be expressed by context-free grammar. What can be expressed by regular language must be expressed by context-free grammar. Regular Expressions can only use terminal symbols (characters in the alphabet), so it is easy to become complex and difficult to understand. In practice, context-free grammars are often used. Regular descriptions allow the use of non-terminal symbols to define expressions, much like EBNF, but It restricts the use of nonterminals until it is completely defined, that is, does not allow recursion or self-nesting. The important difference between regular definitions and context-free grammars is that recursive definitions are not allowed in regular definitions. For example, A → aA|b is not a regular definition, and the left A must be a new symbol. It is defined elsewhere, but its right-hand side requires every symbol to be defined, so this definition cannot be satisfied. The context-free grammar does not have this constraint, so A → aA|b is a production of a context-free grammar, but not a definition of a regular definition.

Bucher, W., & Hagauer, J. (1983). It is decidable whether a regular language is pure context-free. *Theoretical computer science*, *26*(1-2), 233-241

Jain, S., Ong, Y. S., & Stephan, F. (2010). Regular patterns, regular languages and context-free languages. *Information Processing Letters*, *110*(24), 1114-1119.

Part4

S->000A111B

A->A000|

B->B111|

Part5

abababa

S

A

S

A

a

A

S

b

A

a

S

S

A

b

a

A

S

A

S

A

a

There are two syntax derivation trees for abababa

So it is ambiguous.