AQA Computer Science A-Level

4.4.3 Context-free languages

Past Paper Mark Schemes

June 2012 Comp 3

2	(a)		Backus-Naur (Form); **A** Backus Normal (Form), BNF, Extended Backus-Naur (Form), Augmented Backus-Naur (Form), ABNF **A** Misspellings of Backus-Naur **A** Format for Form and the word "Notation" **R** BN	1

2	(b)		<table><tr><th>Statement Type</th><th>String</th><th>Valid (Yes/No)</th></tr><tr><td><number></td><td>129.376</td><td>No;</td></tr><tr><td><factor></td><td>23 + 17</td><td>Yes;</td></tr></table> **A** Alternative clear indicators of Yes or No e.g. Y/N, Tick/Cross, Valid/Invalid, True/False	2

June 2017 Paper 1

01	1	**Marks are for AO1 (understanding)** <table><tr><th>Real number</th><th>Valid? (Yes/No)</th></tr><tr><td>87.000</td><td>Yes</td></tr><tr><td>97+12</td><td>No</td></tr><tr><td>12.31E+12</td><td>Yes</td></tr></table> **A.** alternative indicators for Yes/No eg Y/N. **Mark as follows:** One mark per correct row	3

01	2	**Marks are for AO2 (apply)** `<natural> ::= <digit> | <digit> <natural>` **A.** alternative names for `<natural>` **A.** recursive and non-recursive cases swapped around **Mark as follows:** **1 mark:** correct recursive case **1 mark:** correct non-recursive case **MAX 1** if any errors in answer eg missing |	2

June 2013 Comp 3

1	(a)		One mark per correct response. **Construct** / **Example** / **Valid ?** *identifier* — `Player2name` — No; *parameter* — `x,y:bool` — Yes; *procedure-def* — `procedure square(s:real)` — No; *procedure-def* — `procedure rect(w:int,h:int)` — No; **A** alternative clear indicators of Yes/No such as Y/N, True/False and Tick/Cross.	**4**

1	(b)	(i)	The `<type>` rule has an extra type `char`; The `<procedure-def>` rule does not allow a procedure without parameters // cannot be just an identifier;	**2**

			Accept answers comparing the figures the other way around, i.e. • The type rule does not allow a char • The procedure does not have to have parameters / can be just an identifier		

1	(b)	(ii)	Required as there can be a list of parameters // required as there can be more than one parameter; BNF does not support iteration // BNF can only achieve iteration through recursion // would need infinite number of rules otherwise // recursion allows for more than one parameter; **MAX 1** **A.** Input for parameter **NE.** Rule needs to loop	**1**	

Specimen Paper 12

02	3	**Mark is for AO2 (apply)** **1 mark:** All values in the table have been completed correctly.	1

Rule number (given in Figure 2)	Could be defined using a regular expression
1	Y
2	Y
3	Y
4	N
5	N
6	Y

02	4	**1 mark for AO2 (analyse) and 1 mark for AO3 (design)** **1 mark for AO2 (analyse):** There is no non-recursive / base case; **1 mark for AO3 (design):** `<word> ::= <char><word>|<char>;`	2