VDM Syntax: Corrections and Improvements

The following list presents corrections and improvements to the VDM grammar in the VDM-10 Language Reference Manual. None of the changes modify the language defined by the grammar.

A.1

Delete definition of ‘any module’.
Replace ‘document’ with

document = module, {module} | definition block, {definition block};

A.1.1

Missing ‘,’ in ‘export functions signature’. Change to

export functions signature =
‘**functions**’, function signature, { ‘;’, function signature }, [‘;’] ;

Missing ‘,’ in ‘export operations signature’. Change to

export operations signature =
‘**operations**’, operation signature, { ‘;’, operation signature }, [‘;’] ;

A.2

‘,’ has higher precedence than ‘|’. Change to

document = (class | system), {class | system};

A.3.1

Misplaced ‘,’ in ‘inheritance clause’. Change to

inheritance clause = ‘**is subclass of**’, identifier, {‘,’, identifier} ;

A.4.1

Change ‘type definitions’ to:

type definitions = ‘**types**’, [access type definition, {‘;’, access type definition}, [‘;’]];

Change ‘access type definition’ to:

access type definition = ([access],[’**static**’] | [‘**static**’], [access]), type definition;

Definition of ‘type’; add option for ‘total function definition’.

A.4.1/A.4.2

‘invariant’ and ‘invariant initial function’ are defined in both of these sections, and are identical. Delete definitions from A.4.2.

A.4.3

Change ‘value definitions’ to:

value definitions = ‘**values**’, [access value definition, {‘;’, access value definition}, [‘;’]];

A.4.4

Change ‘function definitions’ to:

function definitions = ‘**functions**’, [access function definition,
{‘;’, access function definition}, [‘;’]];

‘extended explicit function definition’ is defined twice, once for VDM-SL and once for VDM++/VDM-RT. Both are identical. Delete one of the definitions.

Add to end of ‘extended explicit function definition’

[‘**measure**’, expression]

Change ‘identifier type pair list’ to

identifier type pair list = identifier type pair, {‘,’, identifier type pair}

A.4.5

Change ‘operation definitions’ to:

operation definitions = ‘**operations**’, [access operation definition,
{‘;’, access operation definition}, [‘;’]];

Correct the definition of ‘access operation definition’. Refer to https://github.com/overturetool/language/issues/40.

A.4.6

Change ‘access assignment definition’ to:

access assignment definition =
([access],[’**static**’] | [‘**static**’], [access]), assignment definition;

A.4.7

Change definition of ‘mutex predicate’ to

> mutex predicate = ‘**mutex**’, ‘(‘, (‘**all**’ | name list), ‘)’;

A.4.8

Change definition of ‘periodic obligation’ to

> periodic obligation = ‘**periodic**’, ‘(‘, (name | 4 * expression), ‘)’;

Change definition of ‘sporadic obligation’ to

> sporadic obligation = ‘**sporadic**’, ‘(‘, (name | 3 * expression), ‘)’;

A.4.9

Change ‘traces definitions’ to:

> traces definitions = ‘**traces**’, [named trace, {‘;’, named trace}];

Change ‘trace definition term’ to

> trace definition term = trace definition, {‘|’, trace definition};

Delete the spurious ‘;’ that appears after ‘trace binding definition’.

Change ‘trace repeat pattern’ to

> trace repeat pattern = ‘*’ | ‘+’ | ‘?’ | ‘{‘, numeric literal, [‘,’, numeric literal], ‘}’;

Change ‘trace let best binding’ to

> trace let best binding = ‘**let**’, multiple bind, [‘**be**’, ‘**st**’, expression], ‘**in**’, trace definition;

A.5

Add option ‘narrow expression’ to definition of ‘expression’, just after ‘lambda expression’.

Add option ‘result expression’ to definition of ‘expression’.

Add definition (probably in its own subsection of 5, like ‘undefined expression’).

> result expression = ‘**RESULT**’;

Strictly ‘RESULT’ is accommodated by the grammar since ‘name’ is one of the options for ‘expression’. However all other keywords are spelled out explicitly in the grammar; for consistency so should ‘RESULT’.

A.5.19

Change definition of ‘is expression’ to

is expression = ‘**is_**’, (name | basic type), ‘(‘, expression, ‘)’;

A.5.21

Change definition of ‘pre-condition expression’ to

pre condition expression = ‘**pre_**’, ‘(‘, expression list, ‘)’;

Note: ISO 14977 does not allow the character ‘-‘ in non-terminal names.

A.5.26

Change this section to

act expression = ‘**#act**’, ‘(‘, name list, ‘)’;

fin expression = ‘**#fin**’, ‘(‘, name list, ‘)’;

active expression = ‘**#active**’, ‘(‘, name list, ‘)’;

req expression = ‘**#req**’, ‘(‘, name list, ‘)’;

waiting expression = ‘**#waiting**’, ‘(‘, name list, ‘)’;

Note: ‘name list’ can be a single ‘name’.

A.7.2

Change definition of ‘multiple assign statement’ to

multiple assign statement = ‘**atomic**’, ‘(‘, assign statement, ‘;’, assign statement,
{‘;’, assign statement}, ‘)’;

A.7.6

Spurious ‘,’ at end of production for second ‘call statement’.

A.8.1

Replace ‘muinon’ with ‘munion’ (in ‘pattern’ and ‘map muinon pattern’).

B.2

Delete definition of ‘is basic type’.

Change definition of ‘decimal literal’ to

```
decimal literal = numeral, [‘.’, numeral], [exponent];
```

Change definition of ‘character literal’ to

```
character literal = `’`, (character | escape sequence), `’`;
```

Change definition of ‘escape sequence’ to

```
escape sequence = ‘\\’ | ‘\r’ | ‘\n’ | ‘\t’ | ‘\f’ | ‘\e’ | ‘\a’
                | ‘\x’, 2 * hexadecimal digit
                | ‘\u’, 4 * hexadecimal digit
                | ‘\c’, character
                | ‘\’, 3 * octal digit
                | ‘\”’ | `\’`;
```

Change name of ‘Single-line comment’ to ‘single line comment’. Note: ISO 14977 does not allow the character ‘-‘ in non-terminal names.

Change name of ‘Multiple-line comment’ to ‘multiple line comment’.