The aForth Forth compiler

Reuben Thomas

18th November 1999

1 Introduction

aForth is a Forth compiler which complies with the ANSI Forth standard [1]. It evolved from an earlier system which was designed to be a teaching tool and portable Forth compiler, but ended up fulfilling neither criterion well. It has been implemented on Acorn RISC OS. It is designed to be used as a teaching tool, and to this end is written mostly in standard Forth, so that the workings of the compiler can be examined and understood by students learning the language; the compiler itself can be used to illustrate the language and the ANSI standard. Some primitive functions are written in assembly code, and the compiler has a few environmental dependencies, such as requiring twos-complement arithmetic, which are exploited to make the system simpler.

Because it is designed to be easily understood and ported, the compiler is simple, using few optimisations, and with little error checking. It does not implement the whole of the ANSI standard, notably omitting floating point arithmetic.

2 Documentation required by the ANSI standard

Section 2.1 contains the ANS labelling for aForth; the other sections give the documentation required in [1, section 4.1], laid out like the corresponding sections in the standard.

2.1 Labelling

aForth is an ANS Forth System

> providing the Core Extensions word set (except `CONVERT`, `EXPECT`, `SPAN` and `UNUSED`),
>
> providing the Block Extensions word set,
>
> providing `D+`, `D.`, `D.R`, `D0=`, `D>S`, `DABS`, `DNEGATE`, `M+` and `2ROT` from the Double-Number Extensions word set,

providing the Exception Extensions word set,

providing `(`, `BIN`, `CLOSE-FILE`, `CREATE-FILE`, `OPEN-FILE`, `R/O`, `R/W`, `READ-FILE`, `REPOSITION-FILE`, `W/O` and `WRITE-FILE` from the File Extensions word set,

providing `.S`, `?`, `WORDS`, `AHEAD`, `BYE`, `CS-PICK`, `CS-ROLL` and `FORGET` from the Programming-Tools Extensions word set,

providing the Search-Order Extensions word set,

providing `-TRAILING`, `BLANK`, `CMOVE`, `CMOVE>` and `COMPARE` from the String Extensions word set.

2.2 Implementation-defined options

2.2.1 Core word set

- Aligned addresses are those addresses which are divisible by four.
- When given a non-graphic character, `EMIT` passes the code to the host environment's character output routine.
- `ACCEPT` allows the input to be edited by pressing the backspace key or equivalent to delete the last character entered (or do nothing if there are currently no characters in the input).
- The character set corresponds with one of the permitted sets in the range {32…126} but is otherwise environment-dependent.
- All addresses are character-aligned.
- All characters in any character set extensions are matched when finding definition names.
- Control characters never match a space delimiter.
- The control-flow stack is implemented using the data stack. All items placed on the stack are single cells except for *do-sys* elements, which occupy two cells.
- Digits larger than thirty-five are represented by characters with codes starting at the first character after "Z", modulo the size of the character set.
- After input terminates in `ACCEPT`, the cursor remains immediately after the entered text.
- `ABORT"`'s exception abort sequence is to execute `ABORT`.
- The end of an input line is signalled by pressing the return key or equivalent.
- The maximum size of a counted string is 255 characters.
- The maximum size of a parsed string is $2^{32} - 1$ characters.

- The maximum size of a definition name is 31 characters.
- The maximum string length for `ENVIRONMENT?` is 255 characters.
- Only one user input device (the keyboard) is supported.
- Only one user output device (the terminal display) is supported.
- There are eight bits in one address unit.
- Number representation and arithmetic is performed with binary numbers in twos-complement form.
- Types *n* and *d* range over $\{-2^{31}\ldots2^{31}-1\}$, types +*n* and +*d* over $\{0\ldots2^{31}-1\}$ and *u* and *ud* over $\{0\ldots2^{32}-1\}$.
- There are no read-only data-space regions.
- The buffer at `WORD` is 256 characters in size.
- A cell is four address units in size.
- A character is one address unit in size.
- The keyboard terminal input buffer is 256 characters in size.
- The pictured numeric output string buffer is 256 characters in size.
- The scratch area whose address is returned by `PAD` is 256 characters in size.
- The system is case-sensitive.
- The system prompt is “ok”.
- All standard division words use floored division except `SM/REM`, which uses symmetric division.
- When true, `STATE` takes the value 1.
- When arithmetic overflow occurs, the value returned is the answer modulo the largest number of the result type plus one.
- The current definition cannot be found after `DOES>` is compiled.

2.2.2 Block word set

- `LIST` displays “Block” followed by the block number in decimal, then the block as sixteen lines each of sixty-four characters, numbered from nought to fifteen in decimal.
- `\` discards up to the next multiple of sixty-four characters when used in a block.

2.2.3 Exception word set

– Exceptions -1, -2, -10, -11, -14 and -56 may be raised by the system. Exception values -256 to -511 are reserved for the environment executing aForth to raise exceptions. Value -512 is used by the word `(ERROR")`. Other exceptions in the range $\{-255 \ldots -1\}$ may be raised by the host environment.

2.2.4 File word set

The implementation-defined options depend on the host operating system.

2.2.5 Search-Order word set

– The search order may contain up to eight word lists.

– The minimum search order consists of the single word list identified by `FORTH-WORDLIST`.

2.3 Ambiguous conditions

The following ambiguous conditions are recognised and acted upon; all other ambiguous conditions are ignored by the System (although some of them may result in action being taken by the host machine, such as addressing a region outside data space resulting in an address exception). Dashes denote general ambiguous conditions which could arise because of a combination of factors; asterisks denote specific ambiguous conditions which are noted in the glossary entries of the relevant words in the standard.

2.3.1 Core word set

– If a *name* that is neither a valid definition name nor a valid number is encountered during text interpretation, the *name* is displayed followed by a question mark, and `ABORT` is executed.

– If a definition name exceeds the maximum length allowed, it is truncated to the maximum length (31 characters).

– If division by zero is attempted, `-10 THROW` is executed. By default this displays the message “division by zero” and executes `ABORT`.

– When a word with undefined interpretation semantics is interpreted, the message “compilation only” is displayed, and `ABORT` is executed.

– If the data stack has underflowed when the "ok" prompt would usually be displayed by QUIT, ABORT" is executed with the message "stack underflow". All other stack underflow conditions are ignored.

* If RECURSE appears after DOES>, the execution semantics of the word containing the DOES> are appended to that word while it is being compiled.

* If the argument input source is different from the current input source for RESTORE-INPUT, the flag returned is true.

* If data space containing definitions is de-allocated, those definitions continue to be found by dictionary search, and remain intact until overwritten, when the effects depend on exactly what is overwritten, but will probably include name lookup malfunction and incorrect execution semantics.

* If IMMEDIATE is executed when the most recent definition does not have a *name*, the most recent named definition in the compilation word list is made immediate.

* If a *name* is not found by ', POSTPONE, ['] or [COMPILE], the *name* is displayed followed by a question mark, and ABORT is executed.

* If POSTPONE or [COMPILE] is applied to TO, the compilation semantics of TO are appended to the current definition.

2.3.2 Block word set

– If a correct block read was not possible because the blocks file could not be opened, the message "blocks file not found" is displayed and ABORT is executed. Other reasons for a block read failing are not detected or handled.

* If a program alters BLK directly input is redirected to the block number stored in BLK; >IN retains its current value. If this is larger than the size of a block, other effects will occur.

* If there is no current block buffer, the buffer whose number is contained in the variable VALID is updated.

2.3.3 Double-Number word set

* If *d* is outside the range of *n* in D>S, the least-significant cell of the number is returned.

2.3.4 Programming-Tools word set

* If the compilation word list is deleted by FORGET, new definitions will still be added to the defunct word list; if the relevant data structures are subsequently overwritten, incorrect effects will probably occur.

* If FORGET cannot find *name*, *name* is displayed followed by a question mark, and ABORT is executed.

2.3.5 Search-Order word set

* Changing the compilation word list during compilation has no effect; changing the compilation word list before DOES> or IMMEDIATE causes the most recent definition in the new compilation word list to be modified; in the former case this may cause the next definition in memory to be partially overwritten.
* If the search order is empty, PREVIOUS has no effect.
* If ALSO is executed when the search order is full, the last word list in the search order is lost.

2.4 Other system documentation

2.4.1 Core word set

– No non-standard word provided uses PAD.

– The terminal facilities available are a single input (the keyboard), and a single output (the terminal display).

– The available program data space is dependent on the memory available in the host environment.

– 4096 cells of return stack space is available.

– 4096 cells of data stack space is available.

– The system dictionary space required depends on the implementation, and is typically under 32 kilobytes.

2.4.2 Block word set

– No multiprogramming system is provided, so there are no additional restrictions on the use of buffer addresses.

– The number of blocks available depends on the system configuration.

References

[1] American National Standards Institute. *ANS X3.215-1994: Programming Languages—Forth*, 1994.