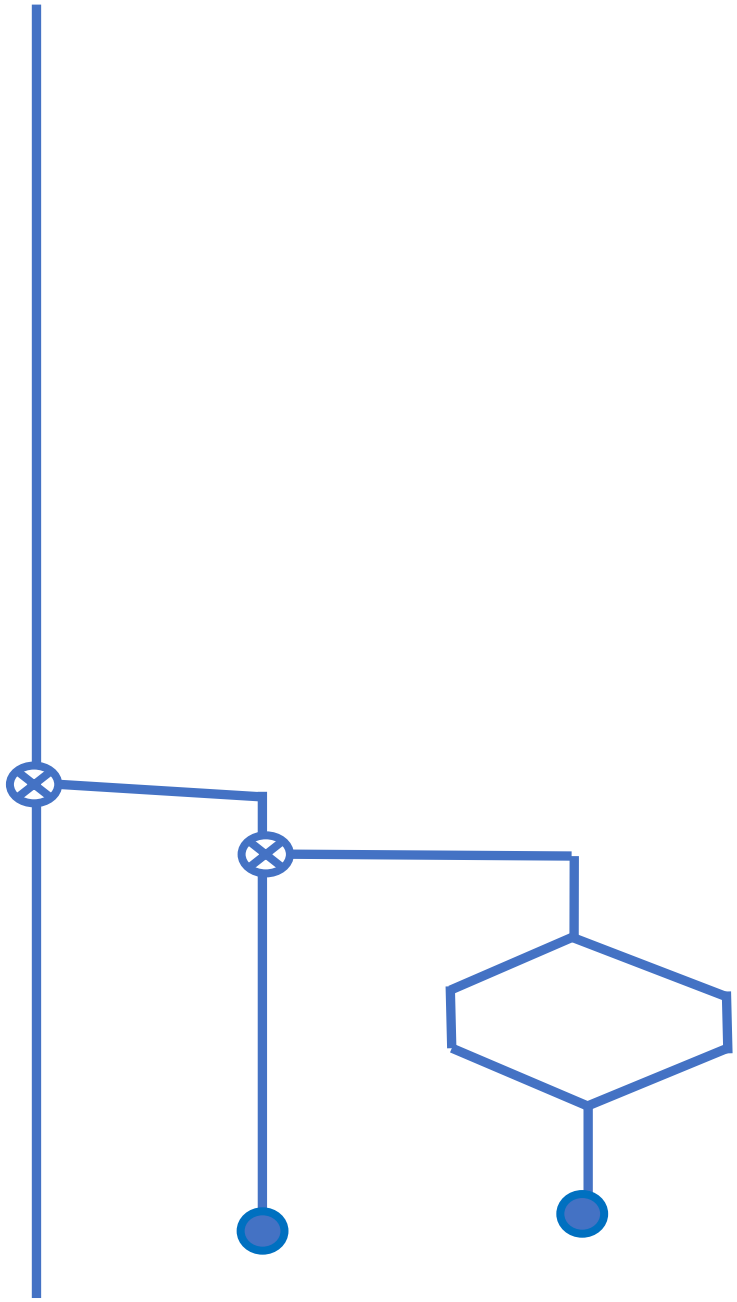


A Case Of Forth

SVFIG
April 24, 2021
Bill Ragsdale

Today . . .

Today we'll explore a common structure in programming: the N-way branch.

Today . . .

Today we'll explore a common structure in programming: the N-way branch.

The common IF-ELSE-THEN works well to make one to two decisions or selections. but. . .

Today . . .

Today we'll explore a common structure in programming: the N-way branch.

The common IF-ELSE-THEN works well to make one to two decisions or selections. but. . .

Another structure is better for a three or more way branch.

The Need

Say we want a three way branch: the selection of a label between inches, feet or yards.

1 should yield ‘inches’.

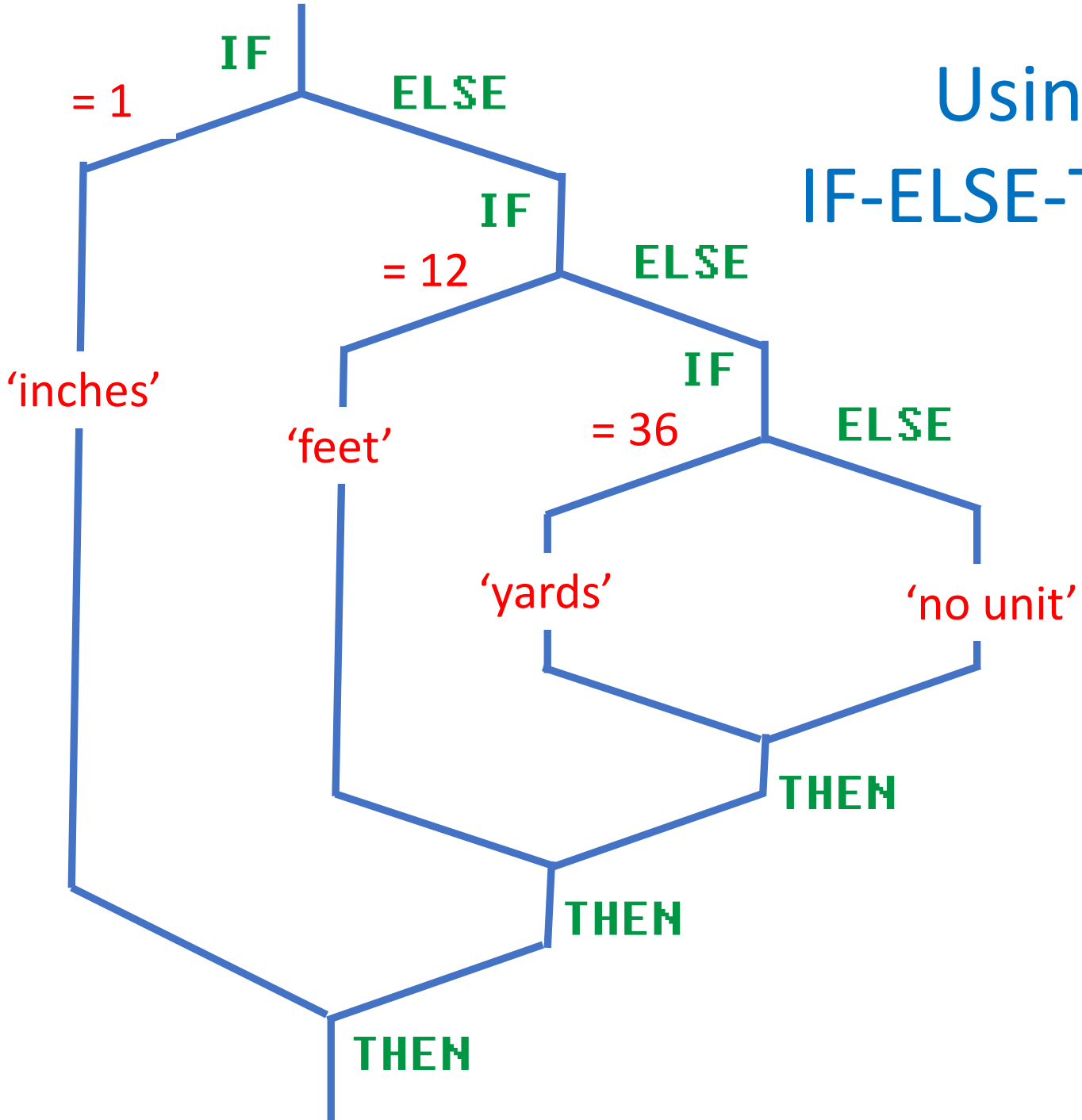
12 should yield ‘feet’

36 should yield ‘yards’.

Units Pseudocode

Input a number
Duplicate and compare to ‘1’.
IF equal, drop saved value. Show ‘inches’.
ELSE duplicate and compare to ‘12’.
IF equal, drop saved value. Show ‘feet’.
ELSE duplicate and compare to ’36’.
IF equal drop saved value. Show ‘yards’.
ELSE drop saved value. Show ‘no unit’.
THEN THEN THEN

Using IF-ELSE-THEN



Code for Units

```
: units
    1 over = if drop ." inches"  else
   12 over = if drop ." feet"    else
   36 over = if drop ." yards"   else
                drop ." no unit"
             then then then ;
```

Code for Units

```
: units
    1 over = if drop ." inches"  else
   12 over = if drop ." feet"    else
   36 over = if drop ." yards"   else
                drop ." no unit"
             then then then ;

cr 1 units cr 12 units cr 36 units

inches
feet
yard  ok
```

The CASE Statement

The CASE statement selects between an arbitrary number of execution paths.

Forths generally use the EAKER syntax published in Dr. Charles Eaker, "Just in Case", Forth Dimensions, Vol.II, No.3, (1980), pp.37.

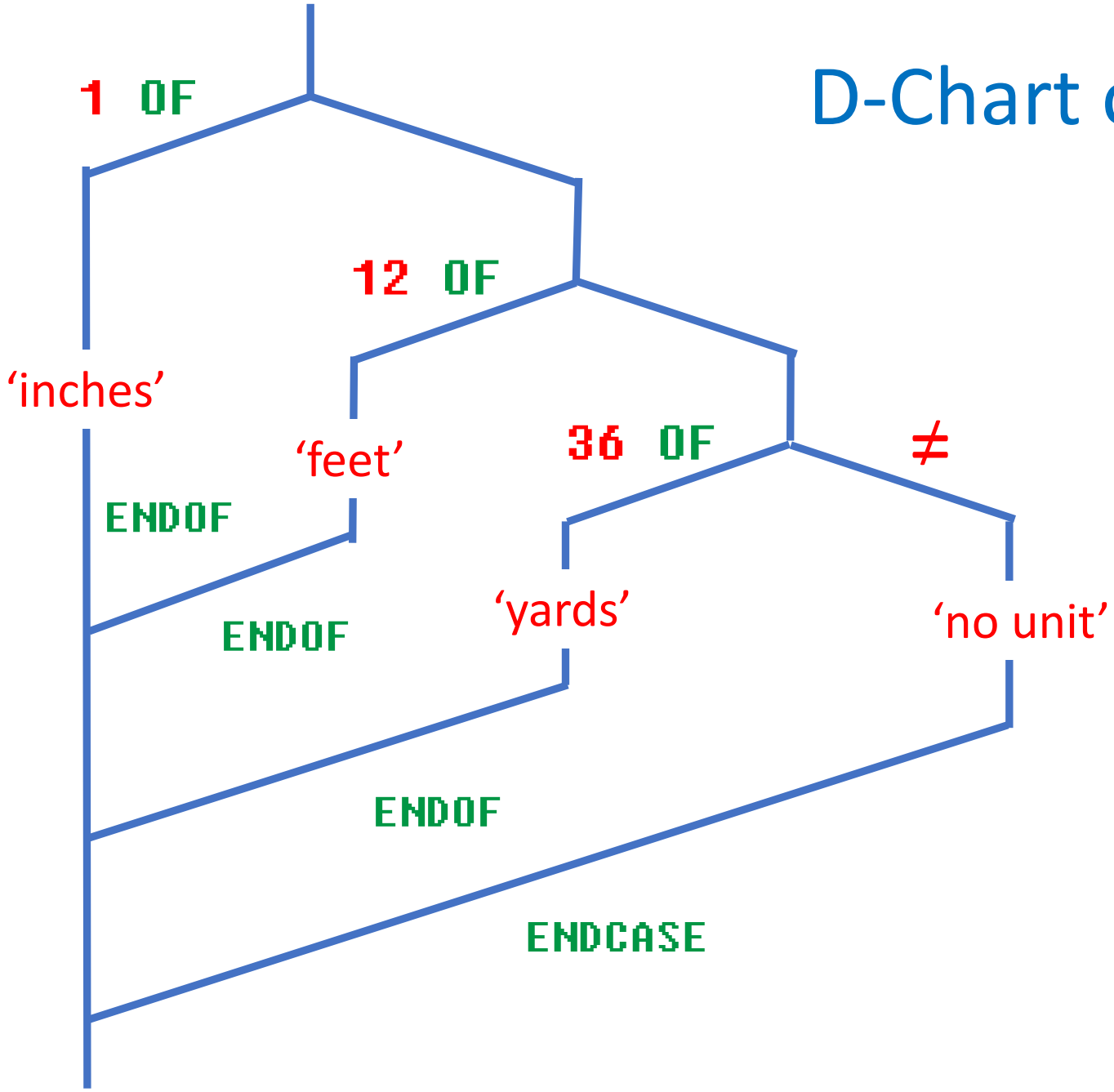
The CASE Elements

CASE: Mark the entrance point of the structure.

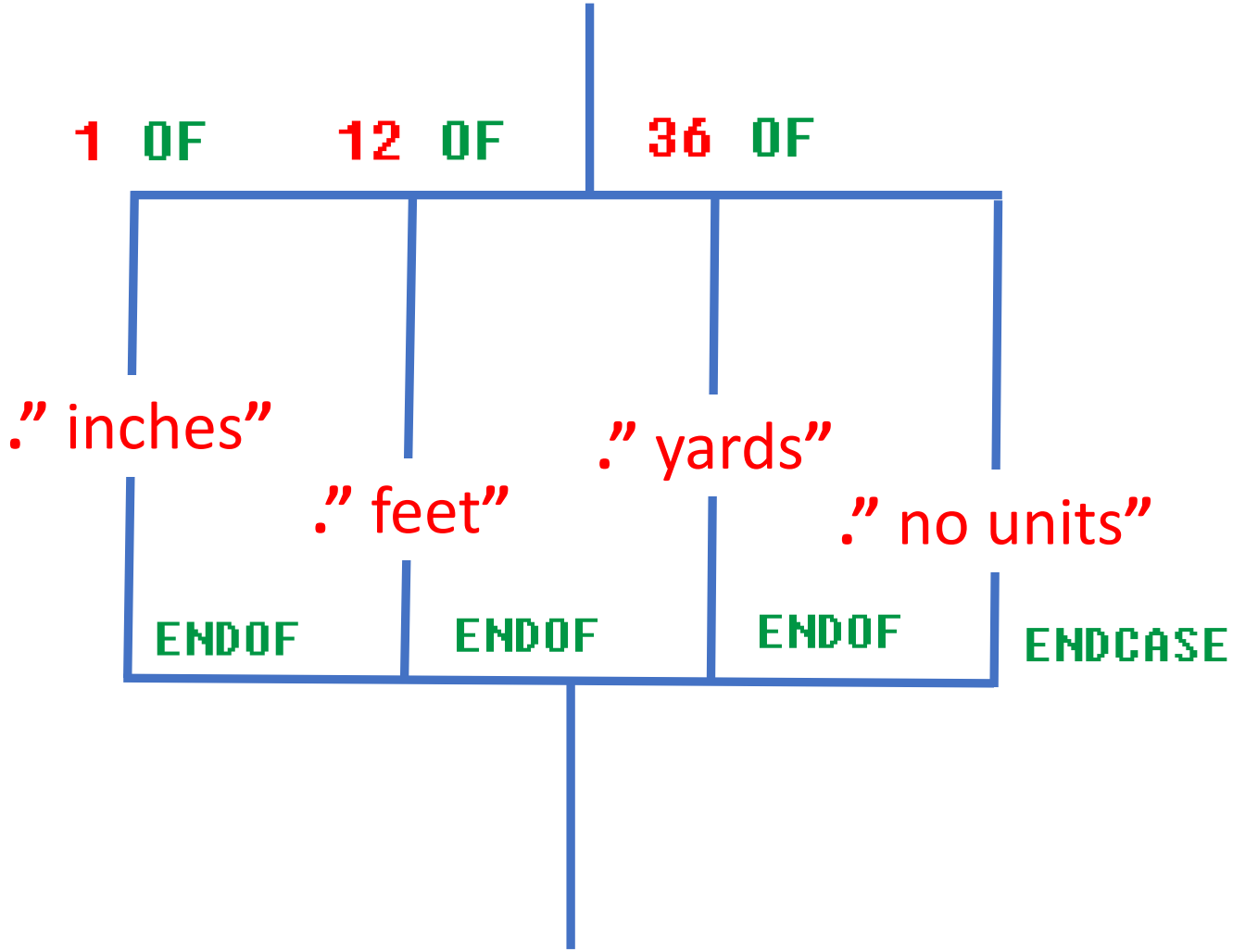
n OF: If stack value matches n, drop both and continue execution. At ENDOF jump to after ENDCASE. If no match, preserve n and continue after ENDOF.

Upon reaching ENDCASE with no selection taken, drop the preserved value n.

D-Chart of CASE



Simplifed D-Chart



Using CASE for Units

```
: units
  CASE
    1 OF  ." inches"     ENDOF
   12 OF  ." feet"       ENDOF
   36 OF  ." yards"      ENDOF
          ." no unit"    ENDCASE  ;
```

Using CASE for Units

```
: units
  CASE
    1 OF  ." inches"     ENDOF
   12 OF  ." feet"       ENDOF
   36 OF  ." yards"      ENDOF
          ." no unit"    ENDCASE  ;
cr 1 units cr 12 units cr 36 units

inches
feet
yards  ok
```

ASCII Case Selection

```
: A-char
case
    ASCII A  of ." alpha"    endof
    ASCII D  of ." delta"    endof
    ASCII G  of ." golf"     endof
                ." no match" endcase ;
```

ASCII Case Selection

```
: A-char
case
   ASCII A  of ." alpha"    endof
   ASCII D  of ." delta"    endof
   ASCII G  of ." golf"     endof
               ." no match" endcase ;

    cr 65 A-char   cr 68 A-char
    cr 71 A-char   cr 90 A-char

 alpha
 delta
 golf
 no match  ok
```

String Inputs For CASE

CASE statements expect a comparison between two numeric values.

If selecting based on strings the string comparison must be done separately.

If two strings match return TRUE TRUE.

If two strings do not match, return FALSE TRUE.

In both cases preserve the input string address

String Comparison

s== return two TRUE if strings match.

```
: s== ( addr1 addr2 --- addr1 bool1 true)
\  counted string comparison

over count rot count  compare 0=  true ;
```

String Comparison

s== return two TRUE if strings match.

```
: s== ( addr1 addr2 --- addr1 bool1 true)
\  counted string comparison

over count rot count  compare 0=  true ;
```

Pseudocode:

Duplicate input address. Count it.
Get reference address. Count it.
String compare. If a match generate TRUE TRUE.
Otherwise generate FALSE TRUE.

String CASE Pseudocode

Input the address of a counted string.
Compare to “Mercury”.
If matching, output “57,900,000 km“ otherwise
Compare to “Earth”.
If matching, output “149,600,000 km" otherwise
Compare to “Mars”.
If matching, output " 227,900,000 km“ otherwise
output “no match”.

String Case Selection

```
: planets  ( addr --- )
  case
   c" Mercury"  s==   of
          ."  57.900.000 km"  endof   drop
   c" Earth"    s==   of
          ." 149.600.000 km"  endof   drop
   c" Mars"     s==   of
          ." 227.900.000 km"   endof   drop
          ." no match"      1  endcase drop ;
```

Testing Planets

```
.( Mercury is at ) c" Mercury" planets  cr
.( Earth is at   ) c" Earth“   planets  cr
.( Mars  is at   ) c" Mars"    planets
```

Testing Planets

```
.( Mercury is at ) c" Mercury" planets  cr
.( Earth is at   ) c" Earth“   planets  cr
.( Mars  is at   ) c" Mars"    planets
```

And see:

```
Mercury is at  57,900,000 km
Earth is at   149,600,000 km
Mars  is at   227,900,000 km    ok
```

Summary

CASE statements simplify n-way selections based on numbers, ASCII values and strings.

Summary

CASE statements simplify n-way selections based on numbers, ASCII values and strings.

CASE statements clearly show program flow much better than nested IF-ELSE-THENs.

Summary

CASE statements simplify n-way selections based on numbers, ASCII values and strings.

CASE statements clearly show program flow much better than nested IF-ELSE-THENs.

CASE statements involving strings need added support for comparisons.

References

- https://github.com/BillRagsdale/Forth_Projects
- https://github.com/BillRagsdale/WIN32Forth-Guide

Questions?