Alternative Functions

(1) AND. If a *nested-IF* expression satisfies the following conditions, it has the semantic of the AND function, as shown in Figure 1: first, the false branches of each condition are all identical; second, the true branches of each condition are all IF expressions, except for the last true value (i.e., $V1$). Such expressions can be replaced with $IF(AND(conditionlist), truebranch, falsebranch)$. For example, the expression with the first AST in Figure 1 can be replaced with $IF(AND(C1, C2, C3, C4), V1, V2)$.

(2) OR. If a *nested-IF* expression satisfies the following conditions, we infer that it actually has the semantic of the OR function, as shown in the second AST of Figure 1: first, the true branches of each condition are all identical; second, the false branches of each condition are all IF expressions, except for the last false value (i.e., $V2$). Such kind of expressions can be replaced with $IF(OR(conditionlist), truevalue, falsevalue)$. For example, the expression with the second AST in Figure 1 can be replaced with $IF(OR(C1, C2, C3, C4), V1, V2)$.

(3) CHOOSE. A CHOOSE function pattern [1] should have these features. First, all the conditions are number equality evaluations, with the corresponding numbers forming an arithmetic progression, which can be translated into natural sequences. Second, the false branches of each condition are all IF expressions, except for the last false value. Third, the true branch values are all strings. For example, $IF(A1 = 1, str1, IF(A1 = 2, str2, IF(A1 = 3, str3, IF(A1 = 4, str4))))$ could be transformed into $CHOOSE(A1, str1, str2, str3, str4)$; expression $IF(A1 = 2, str1, IF(A1 = 4, str2, IF(A1 = 6, str3, IF(A1 = 8, str4))))$ could be transformed into $CHOOSE(A1/2, str1, str2, str3, str4)$.

(4) MATCH. A *nested-IF* expression that matches the semantic of MATCH function [2] should have the following features. First, all the conditions are string equality evaluations. Second, the true branch values are all numbers that could form an arithmetic progression, which can be translated into a natural sequence. Third, the false branches of each condition are all IF expressions, except for the last false value. For example, expression $IF(A1 = str1, 1, IF(A1 = str2, 2, IF(A1 = str3, 3, IF(A1 = str4, 4))))$ could be transformed into $MATCH(A1, str1, str2, str3, str4, 0)$; expression $IF(A1 = str1, 2, IF(A1 = str2, 4, IF(A1 = str3, 6, IF(A1 = str4, 8))))$ could be transformed into $2 * MATCH(A1, str1, str2, str3, str4, 0)$.

(5) LOOKUP. A *nested-IF* expression that matches the semantic of VLOOKUP/HLOOKUP pattern [3, 4] should have the following features. First, all the conditions are equality evaluations of reference values. The references are cell neighbors vertically/horizontally. Second, all the true branches are references that referred to other cells. The references are cell neighbors vertically/horizontally, and have the same columns/rows as the references in the conditions. Third, the false branches of each condition are all IF expressions, except for the last false value. For example, as shown in Table **??**, expression $IF(A1 = C1, D1, IF(A1 = C2, D2, IF(A1 = C3, D3, IF(A1 = C4, D4))))$ can be transformed into $VLOOKUP(A1, C1 : D4, 2, FALSE)$.

The above patterns suit the circumstance that the values looked up can be found directly in other cells. For those that cannot be found directly, in this paper, we propose creating new tables in the worksheets to make ease for the look up function. Consequently, as long as the conditions are evaluating the value of a specific cell (doing look up based on this cell), we can perform transformation with the LOOKUP function. For example, for expression $IF(A1 = V1, V2, IF(A1 = V3, V4, IF(A1 = V5, V6, IF(A1 = V7, V8))))$, we create a table ranged $(E1 : F4)$, where $E1 = V1$, $F1 = V2$, $E2 = V3$, $F2 = V4$,$E3 = V5$, $F3 = V6$,$E4 = V7$, $F4 = V8$. In this way, the expression can be transformed into $VLOOKUP(A1, E1 : F4, 2, FALSE)$.

(6) MAX/MIN. A *nested-IF* expression that matches the semantic of MAX or MIN pattern should have the following features. The condition should do the comparison of two parts, e.g., $A < B$, $A <= B$, $A > B$, $A >= B$. The true branch and the false branch should be these two parts respectively. For example, expressions $IF(A < B, A, B)$, $IF(A <= B, A, B)$, $IF(B > A, A, B)$, $IF(B >= A, A, B)$ can all be transformed into $MIN(A, B)$; expressions $IF(A > B, A, B)$, $IF(A >= B, A, B)$, $IF(B < A, A, B)$, $IF(B <= A, A, B)$ can all be transformed into $MAX(A, B)$.

(7) IFS. The IFS pattern has the fewest conditions. As long as the false branches are IF expressions (except for the leaves), the expression can be transformed with the IFS function, as shown in Table **??**. Note that this pattern makes the fewest syntax changes comparing to the original syntax, and the number of conditions remain the same. However, the IFS function has the advantage of conciseness and readability, and there is also no need to worry about the IF statements and parentheses [5]. Additionally, there is no need to supply a value if the condition is false (unlike the *nested-IF* expression which needs another IF expression to serve as the false branch) [6].

REFERENCES

[1] CHOOSE. CHOOSE function. https://support.office.com/en-us/article/CHOOSE-function-fc5c184f-cb62-4ec7-a46e-38653b98f5bc.

[2] MATCH. MATCH function. https://support.office.com/en-us/article/MATCH-function-e8dffd45-c762-47d6-bf89-533f4a37673a.

[3] VLOOKUP. VLOOKUP function. https://support.office.com/en-us/article/VLOOKUP-function-0bbc8083-26fe-4963-8ab8-93a18ad188a1.

[4] HLOOKUP. HLOOKUP function. https://support.office.com/en-us/article/HLOOKUP-function-a3034eec-b719-4ba3-bb65-e1ad662ed95f.

[5] office. IF function – nested formulas and avoiding pitfalls . https://support.office.com/en-us/article/IF-function-%e2%80%93-nested-formulas-and-avoiding-pitfalls-0b22ff44-f149-44ba-aeb5-4ef99da241c8?ui=en-US&rs=en-US&ad=US, 2015.

[6] spreadsheeto. Let's take a look at IF, Nested IF and IFS. http://spreadsheeto.com/if/, 2015.

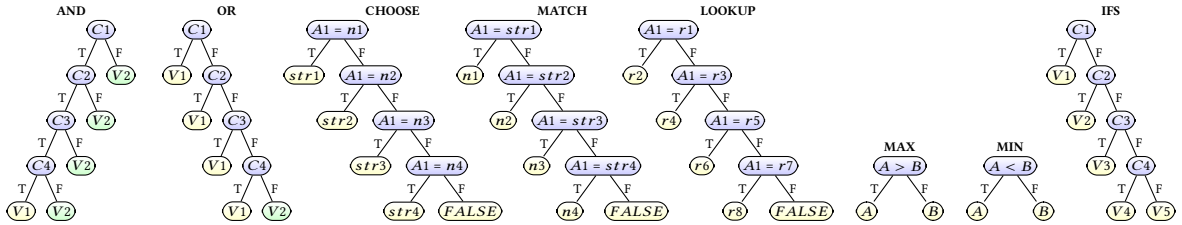

Figure 1: Typical AST of function AND,OR,CHOOSE,MATCH,LOOKUP, MAX,MIN, and IFS. *stri* represents a string; *ni* represents a number; *ri* represents a reference ($0 < i < 5$).