

新

組み込みソフトへの 数理的アプローチ

～形式仕様記述をどのように使うか～



第5回

AND-OR 木の形式化

—機能の取捨選択をグラフで表現する

藤倉 俊幸

はじめに

今回解説する AND-OR 木は木構造をしたグラフで、ノード間を結ぶ線に AND か OR の属性を付けたものである。ソフトウェア開発では、ゴール・ツリーとかフィーチャ・モデル、フォルト・ツリー、原因結果グラフなどが AND-OR 木である。

今回は AND-OR 木を命題論理で形式化してみよう。ソフトウェア開発で使ういろいろなグラフを形式化できると矛盾がないかどうか調べたり、一部を修正したりと使い回しがよくなる。

グラフにすると普通は読みやすくなるが、^{まんだら}のような大きなグラフになると、逆に隅から隅まで見られないので矛盾がないかどうか確認することが難しくなる。しかし、形式化すれば一発で確認できる。また、そのグラフに含まれる可能な条件の組み合わせが何通りあるかなども確認できる。しかも、グラフから論理式を機械的に作るので、直接論理式を書けない人でも形式化の御利益を受けることができる。つまり、論理式は書けないがグラフなら描けるという人にとっては、考えを形式化するツールとして利用できる。そして、形式化に慣れてきて、グラフなんかどうでもよくて、本質は論理式の集まりであるということに気付けば形式仕様記述の本当の便利さがわかるかもしれない。

グラフは見た目はわかりやすいが、実際の開発で出てくるような条件をグラフ化するとゴチャゴチャになって何だかわからなくなる。グラフを形式化し、検証して問題があったら、グラフではなく論理式を修正して結果を見る。OK になったら結果をほかの人に説明するためにグラフに

戻す。グラフはあくまでも必要な部分を切り出して説明するだけに使う。グラフを見ながら考えるアプローチにはサイズの限界がある。そのことを最後にフィーチャ・モデルによって示す。

可変性を表すフィーチャ・モデルでは木構造のほかに、ノード間の依存関係や排他的関係を記述する。これがグラフの近いノード間であれば問題ないが、遠くのノード間ともはや何の線かわからなくなってしまふ。つまり、ビジュアルな表現は現実の問題には役に立たないことが多いのである。一方、論理式の集まりとして記述すれば、ノードの距離などという問題は発生しない。また、データベース化して検索できるなどメリットも大きい。



1 AND-OR 木

● AND-OR 木のメリット

AND-OR 木を問題解決や問題分割に利用すると、複雑な条件をわかりやすく表現することができる。図1に簡単な AND-OR 木の例を示す。A, B, C, D, E はノードとか節点と呼ばれる。AND 条件を指定するところには扇のような弧を描くのが一般的である。弧のないところは OR 条件になる。ただし、これは人工知能分野で利用される場合で、ゴール・ツリーやフィーチャ・モデルでは流儀によっていろいろである^{注1}。

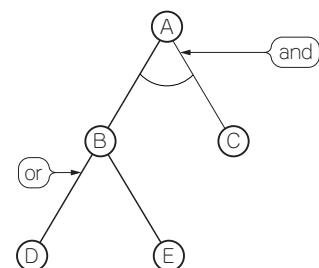


図1 AND-OR 木

注1：例えば、ゴール・ツリーの一種の NFR (Non Functional Requirement) フレームワークでは AND に弧を描くところは同じだが、OR には2重線の弧を描く。