

ゴール選択基準に基づくゴール選択アルゴリズム：*gsl-dfs*

佐藤慎一[†]青山学院大学[†]

1 ゴール指向要求分析

ゴール指向要求分析 (Goal-Oriented Requirements Engineering; GORE) を用いた要求獲得は次の手順で行われる。①顧客のニーズに基づいて、初期ゴールを作成する。②実現可能なゴール (最終ゴール) が現れるまでゴール分解を繰り返す。③最終ゴールの中から要求として獲得するゴールを選択する。この手順により作成されるゴール指向要求分析の成果物をゴールグラフという。ゴールグラフは、AND/ORグラフかつ非循環有向グラフ (Directed Acyclic Graph; DAG) である。ゴールグラフの例を図1に示す。ただし、評価 (3節) に利用するため、個々のコネクタ^{※1} [大須賀 91] に貢献度 [Hayashi 12] が設定されている。また、個々のゴールに添字 (①～⑥) が設定されている。図1では、①が初期ゴール、④、⑤、⑥が最終ゴールである。

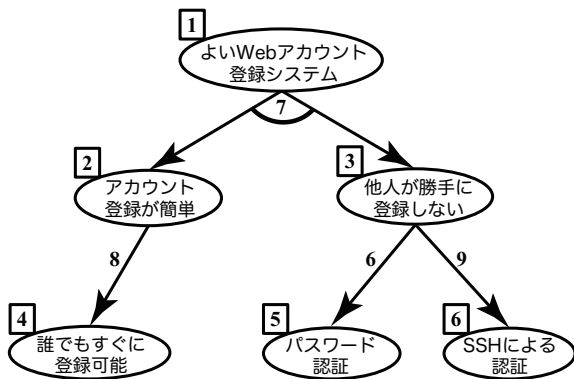


図1 ゴールグラフの例 [Hayashi 12]

2 *gsl-dfs*

単一のゴール選択基準に基づいて、ゴールグラフにおけるすべての^{※2}初期ゴールを達成するために最適な最終ゴールを選択するアルゴリズム*gsl-dfs* (goal selection algorithm - based on depth-first search) [佐藤 13] を図2に示す。ただし、 v , w はゴールを表す変数。 $mark[v]$ は、 v が未訪問であるか訪問済

であるかを表す、 $unvisited$ または $visited$ のブール値を要素とするリストである。 $L[v]$ は隣接リストであり、 v に隣接するゴールを要素として持つ。 $L_{sg}[v][w]$ は、 v が w を選択する場合1、そうでない場合0のブール値を要素とするリストである。 M には、対象とするゴールグラフ上の子ゴールの最大数以上の数 (初期ゴールを除くすべてのゴール以上の数) が設定されているものとする。 v の子ゴールが1つだけ ($n_{and} = 0$ and $n_{or} = 1$) または v がすべての子ゴールとAND分解で接続している ($n_{and} \geq 2$ and $n_{or} = 0$) 場合、すべての子ゴールが選択される (図2, 行13～16)。 v が2つ以上の子ゴールとOR分解のみで接続している ($n_{and} = 0$ and $n_{or} \geq 2$) 場合、または、少なくとも1つの子ゴールとAND分解で接続し、かつ、少なくとも1つの子ゴールとOR分解で接続している ($n_{and} \geq 1$ and $n_{or} \geq 1$) 場合、 v は少なくとも1つの子ゴールを選択しなければならない。この処理は、手続き*gsl*にまとめられている (図2, 行17～18)。手続き*gsl*の詳細を、図3に示す。 v とその子ゴールがこれら以外の接続関係である場合、 v に選択される子ゴールはないものとする (図2, 行19～20)。任意の初期ゴールを出発点として定めた*gsl-dfs*を実行後の $L_{sg}[v][w]$ において、 $L_{sg}[v][w] = 1$ 、かつ、 w が最終ゴールであるような w が、各々要求である。

```

1: procedure gsl-dfs (  $v$ : vertex );
2: var
3:    $w$ : vertex;
4:    $n_{and}$ ,  $n_{or}$ ,  $i$ : integer;
5:    $n_{and} := 0$ ,  $n_{or} := 0$ ;
6:    $L_{sg}$ : array [0.. $M+1$ , 0.. $M+1$ ] of boolean;
7: begin
8:    $mark[v] := visited$ ;
9:   for  $L[v]$ 上の各頂点 $w$  do
10:    if  $w$ が $v$ の子ゴール and  $w$ は $v$ とAND分解で接続 then
11:       $n_{and} := n_{and} + 1$ 
12:    else if  $w$ が $v$ の子ゴール and  $w$ は $v$ とOR分解で接続 then
13:       $n_{or} := n_{or} + 1$ ;

```

gsl-dfs: A Goal Selection Algorithm Based on a Goal Selection Criterion

[†] Sato Shin'ichi

[†] Aoyama Gakuin University

[†] Email: sato@ise.aoyama.ac.jp

```

14:  if (n_and = 0 and n_or = 1) or (n_and >= 2
    and n_or = 0) then
15:    for L[v]上の各頂点w do
16:      if wがvの子ゴール then
17:        L_sg[v][w] := 1
18:      else if (n_and = 0 and n_or >= 2) or
    (n_and >= 1 and n_or >= 1) then
19:        gsl(v)
20:      else
21:        L_sg[v][w] := 0;
22:      for L[v]上の各頂点w do
23:        if mark[w] := unvisited then
24:          gsl-dfs(w)
25: end; { gsl-dfs }

```

図2 *gsl-dfs*

```

1: procedure gsl ( v: vertex );
2: var
3:   w: vertex;
4:   max, flg: integer;
5:   flg := 1;
6:   L_vl: array [0..M + 1] of real;
7: begin
8:   for L[v]上の各頂点w do
9:     if wがvの子ゴール then
10:      L_vl[w] := { wのゴール選択基準値 }
11:     else
12:      L_vl[w] := 0;
13:   for L[v]上の各頂点w do
14:     if flg = 1 then
15:       max := L_vl[w];
16:       flg := 0
17:     else if max < L_vl[w] then
18:       max := L_vl[w];
19:   for L[v]上の各頂点w do
20:     if max = L_vl[w] then
21:       L_sg[v][w] := 1
22: end; { gsl }

```

図3 *gsl* (図2の18行目の処理)

要求がどのようにして獲得されたかというプロセスの明示性の基準である「追跡可能性」[IEEE Standards Board 98]の観点からは、単に獲得された要求を知るだけではなく、出発点として指定される初期ゴールから要求として採用される最終ゴールまでの経路も併せて知ることができればなおよい。この経路を得る手続き *pth* を図4に示す。ただし、*v* は初期ゴールを表す変数である。

```

1: procedure pth ( v: vertex );
2:   for L[v]上の各頂点w do
3:     if s[v][w] = 1 then begin
4:       { v->w };
5:       pth(w)
6:     end
7: end; { pth }

```

図4 *pth*

3 評価

手続き *gsl* (図2, 行18) は、*v* のすべての子ゴールに対して、そのゴール選択基準値を要素とするリスト *L_vl*[*w*] を作成し、そのうちの最大値をとる子ゴールを選択する (最大値をとる子ゴールが複数存在する場合、それらすべての子ゴールが選択される)。 *L_vl*[*w*] に代入される *w* のゴール選択基準値を定めるための何らかの関数が必要となり、この関数は、{ *w* のゴール選択基準値 } (図3, 行9) に置換される。ここでは、貢献度を定める関数が用意されていることを仮定し、*w* のゴール選択基準値として貢献度を採用する。図1に対して *gsl-dfs* (図2) を適用し、*pth* (図4) で表示した結果を図5に示す。この結果から、図1に対して *gsl-dfs* は適切に機能することが確かめられた。

一般に、ゴール選択基準は複数存在する。ゴール選択基準間のコンフリクトが存在する場合、ゴール選択結果が一意に定まらない [佐藤 13]。 *gsl-dfs* は、ゴール選択基準が複数の場合にも対応可能である。上述の関数を AHP [Saaty 80] などの多基準意思決定手法のアルゴリズムに基づいて設計し、それを組み込めばよい。しかる後、複数のゴール選択基準を持つ実用規模のゴールグラフに対して *gsl-dfs* の評価を行うことが、今後の課題である。

1: 1->2_2->4_1->3_3->6

図5 図1のゴールグラフに対する *gsl-dfs* の適用結果

※1一つの親ゴールとその子ゴールの集合を結ぶハイパーエッジ。

※2初期ゴールは複数作成可能である。 *gsl-dfs* (図2) は、初期ゴールが複数存在する場合にも対応している。

参考文献

- [Hayashi 12] Hayashi, S., Tanabe, D., Kaiya, H., and Saeki, M.: Impact Analysis on an Attributed Goal Graph, *IEICE Trans. Inf. & Syst.*, Vol. E95-D, No. 4, pp. 1012-1020 (2012)
- [IEEE Standards Board 98] IEEE Standards Board, : IEEE Recommended Practice for Software Requirements Specifications, IEEE Std. 830-1998 (1998)
- [Saaty 80] Saaty, T. L.: *The Analytic Hierarchy Process: Planning Setting Priorities, Resource Allocation*, McGraw-Hill (1980)
- [大須賀 91] 大須賀 節雄: 人工知能大辞典, 丸善株式会社 (1991)
- [佐藤 13] 佐藤 慎一: ゴール指向要求分析における形式的ゴール選択方法の構築, PhD thesis, 東京工業大学 (2013), <http://tdl.libra.titech.ac.jp/hkshi/xc/contents/pdf/300382932/1>