

# 階層的科目体系における卒業単位チェックの判定アルゴリズム —動的パラメータの導入と再帰的アルゴリズムによる実現—

Algorithmic verification of graduation credit requirements for a multi-stage curriculum: Implementation using dynamic parameters and a recursive algorithm

松尾太加志\*

Takashi MATSUO

The purpose of this paper is to report on how recursive algorithms and dynamic parameters were used to create an academic credit requirement verification system to be used to determine graduation eligibility. The hierarchical curriculum has been represented using a tree structure with a pointer. The use of a recursive algorithm was necessary to allow curriculum components to be searched and total credits earned to be calculated. In the case of a system which does not cross-verify credit totals, a separate algorithm must be used to determine credit deficits within specific curriculum areas. For this reason, a recursive algorithm is required to allow for recalculation of previously accessed values (a kind of back-tracking method). Furthermore, by calculating prospective minimum credit totals and prospective maximum credit totals, dynamic parameters within each curriculum area may be compared with actual credits earned, thereby facilitating a three-step verification procedure.

Key words: recursive algorithm, dynamic parameter, graduation credit requirements

本論文の目的は、「卒業単位チェックシステム」の実装における動的パラメータと再帰的アルゴリズムについての報告である。階層的科目体系では、その構造はポインタによる木構造として表現され、その科目体系を探索して単位の合計を算出するには、再帰的アルゴリズムが必要である。また、合計不一致型の単位システムの場合、判定アルゴリズムにおいて、他の科目区分の不足単位数を計算する必要がある。そのため、一種のバックトラックによって再帰的アルゴリズムで実現しなければならない。さらに、各科目区分においての判定を3段階で行うのに、各科目区分における動的パラメータとして最大見込み単位数と最小見込み単位数を算出することによって、それらの動的パラメータと取得単位数との比較で判定が可能になった。

キーワード：再帰的アルゴリズム、動的パラメータ、卒業単位

## 1. はじめに

大学のカリキュラムの科目体系は科目区分が階層的な構造をしており、卒業に必要な単位は下位の科目区分の合計が上位の科目区分に算入されるようになっている。大学の学部学科によって、科目自体は異なるものの、このような階層的な構造は共通しており、卒業に必要な単位をチェックする際の基本的なルールや構造は同じである。そのため、卒業単位チェックの汎用的なプログラムを開発する

ことは困難な課題ではなく、そのようなシステムが利用できれば、大学の学務においても学生側にとっても有益である。現在、学務情報システムの多くは、履修成績や履修登録と連動させることによって有用なシステムとなっており、さらにコース管理システ

| 科目区分 | 算入科目 | 必要単位 | 認定単位 | 修得単位 |
|------|------|------|------|------|
| 1)教養 |      | 10   | × 8  | 8    |
| 2)語学 |      | 10   | ○ 10 | 10   |
| 3)専門 | 4+5  | 20   | × 18 |      |
|      | 4)基礎 | 10   | × 2  | 2    |
|      | 5)実践 | 0    | △ 16 | 16   |
| 6)合計 | 1~3  | 50   | × 36 |      |

図1 「卒業単位チェックシステム」<sup>3)</sup> の出力例。○は新たな取得不要、×は追加取得が必要、△は他の区分で取らなければ追加取得必要。○は緑、×は赤、△は黄で表示

ムとの連携が実現されている<sup>1)</sup>。しかし、カリキュラムに合わせたカスタマイズの必要性、成績と連動するためのセキュリティの確保<sup>2)</sup>の問題などが生じる。そのため、導入や運用コストは、金銭面だけではなくシステム構築や保守運用のための人的資源においてもかなり必要となる。しかし、卒業の単位チェックだけに限れば、卒業要件の条件だけを設計すればよい。コストはあまりかからない上に、成績との連動もないため、セキュリティ上の問題も考慮する必要がない<sup>3)</sup>。

同様の単位チェックシステムはいくつかみられるが<sup>4)</sup>、汎用的なものは存在しない。そこで、Webで利用できる卒業単位チェックの汎用的なシステム(図1)の開発を行った<sup>3)</sup>。汎用性をもたせるために、データ構造が木構造となるため、単位計算などの探索において、再帰的なアルゴリズムの導入が必要であった。さらに、科目区分で定められた最低単位数を取得しただけでは上位の科目区分の合計を満たさないことがあり、このような場合の判定に動的パラメータの導入が必要であった。本稿では、再帰的アルゴリズムと動的パラメータの導入によって実現された卒業単位チェックの判定アルゴリズムについて報告を行う。

## 2. 問題

### 2.1 階層的な科目体系の探索

「卒業単位チェックシステム」では、各大学のカリキュラムに合わせて、どの科目区分の単位がどの科目区分の合計に算入されるのかを表した科目表を作成するだけで、自動的にその大学に合った単位チェックプログラムが作成される<sup>3)</sup>。そのデータ構造は、図2に示したようなポインタによる木構造となる<sup>5)</sup>。各ノードには科目区分名や当該の科目区分における最低取得単位数がデータとして示される。卒業単位の全合計は根ノードとなり、最下位の科目区分は葉ノードとなり、ポインタは *null* で表される。

合計単位を計算するためには、この木構造を下に探索しながら単位数を合計していく必要がある。その際、合計単位は下位ノードの単位を合計するというアルゴリズムによって実現できる。下位の科目区分は部分木となるため、再帰的アルゴリズムによって実装することができる。

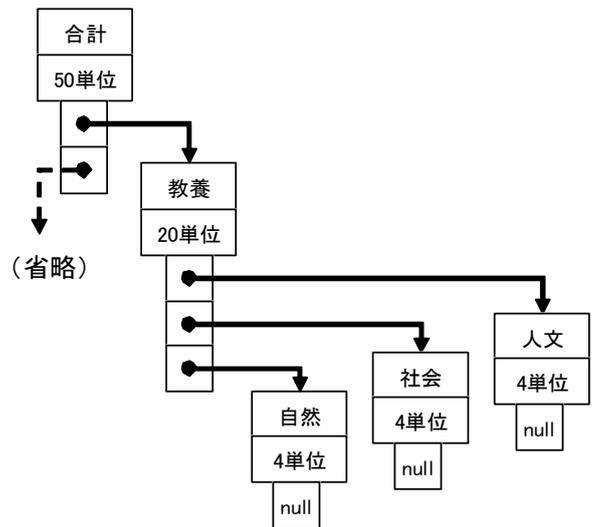


図2 階層的科目体系のポインタによる木構造。

### 2.2 単位の合計数が一致しないケースの判定

一般に、ある科目区分の必要単位数は、その下位にある複数の科目区分の必要単位数の合計となる。ただし、場合によっては、科目区分ごとに必要な単位の最低数が定められているだけで、それらの必要最低数を合計しても、上位で必要な単位数に満たないことが多い。上位の合計数を満たすには、各科目区分での必要単位数を満たした上に、さらに合計数を満たすように、どこかの科目区分から単位を取得しなければならない。

各科目区分での必要単位数の制約を低くすることによって、学生は自分が取りたい科目区分での科目を多く履修し、それらを卒業単位に反映できるため、学生の科目履修の自由度を高めたカリキュラム体系になっている。このようなカリキュラムの場合、学生によって、科目区分の取得単位数が異なるため、卒業に必要な単位の計算誤りが生じやすくなる。そのため、このようなカリキュラム体系の場合、「卒業単位チェックシステム」導入の有用性が高い。

本稿では、以下のような用語の使い方をする。必要最低単位数は各科目区分で最低取得しなければならない単位数とする。また、開講科目数などによって認定される単位数の上限が決まっている場合、それを最大認定単位とよぶ。そして、ある科目区分の必要最低単位数がそれよりも下位の科目区分の必要最低単位数の合計と一致する場合、合計一致型とよび、そうでない場合を合計不一致型とよぶ。表1に合計一致型と合計不一致型の例を示した。

合計一致型の場合、当該の科目区分で定められた

| 科目区分   | 必要最低<br>単位 | 取得単位 | 不足単位 |
|--------|------------|------|------|
| 教養(合計) | 20         | 14   | 6    |
| 人文     | 8          | 6    | 2    |
| 社会     | 6          | 6    | 0    |
| 自然     | 6          | 2    | 4    |

| 科目区分   | 必要最低<br>単位 | 取得単位 | 不足単位 |
|--------|------------|------|------|
| 教養(合計) | 20         | 14   | 6    |
| 人文     | 4          | 6    | 0?   |
| 社会     | 4          | 6    | 0?   |
| 自然     | 4          | 2    | 2?   |

単位を超えて取得しても、超過分は合計に算入されない。合計一致型でなくても、合計に算入される単位数を限定している場合も、超過分が合計に算入されないため、合計一致型と等価となる。このような場合を擬似合計一致型とよぶこととし、合計の算入の上限を算入上限とよぶ。

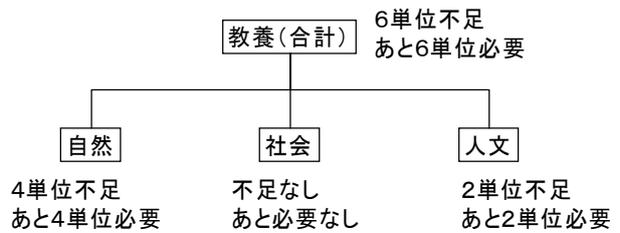
### 3. 再帰的アルゴリズムの必要性

「卒業単位チェックシステム」として有用なのは、最終的な卒業判定よりも、どの科目区分でさらに単位を取得する必要があるかを示すことである。合計一致型の場合、各科目区分での不足単位数がそのまま取らなければならない単位数となるため、単純に不足単位数を算出すればよい(表1上, 図3a)。

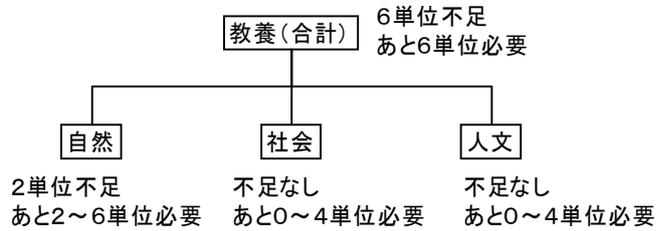
ところが、合計不一致型の場合、必要最低単位数を満たしていても、上位の合計を満たすために、さらに単位をとらなければならないことがあり、明確に何単位あと取得しなければならないか判定できないため、単純なアルゴリズムでは実現できない。表1下および図3bに示した例では、「人文」や「社会」では単位を充足しているが、「教養(合計)」を満たすためには、まだ単位を取る必要が残っている。ただし、「自然」で2単位不足しているため、2単位分

表2 下位科目区分まで探索しなければならない例

| 科目区分 | 必要最低<br>単位数 | 修得単位数 | 不足単<br>位数 |
|------|-------------|-------|-----------|
| 教養   | 10          | 8     | 2         |
| 専門   | 20          | 18    | 2         |
| 基礎   | 10          | 2     | 8         |
| 実践   | 0           | 16    | 0         |
| 合計   | 40          | 26    | 14        |



a) 合計一致型の場合. 不足単位分を取得すればよい.

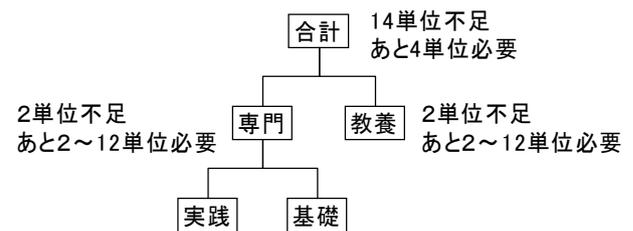


b) 合計不一致型の場合. 不足単位数以上の取得が必要

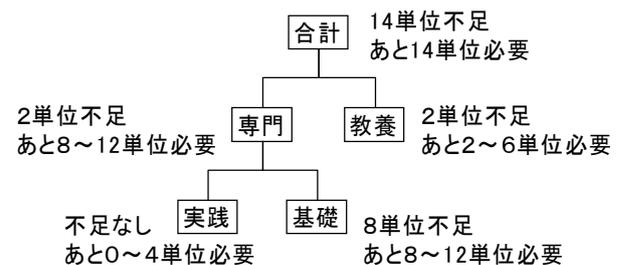
図3 各科目区分ごとの判定例

は確実に補うことができ、「教養(合計)」の不足分の6単位すべてを「人文」や「社会」で補う必要はなく、最大で4単位までとなる。しかも、その4単位までというの、「人文」、「社会」、「自然」のいずれかの科目区分で補ってしまえば、新たに取り必要がないため、「人文」や「社会」で取るべき単位は、4単位が必須ではなく、最大で4単位にすぎない。そのため、「あと0~4単位必要」という幅をもった判定になってしまう。「自然」についても同様で、「あと2~6単位必要」ということになる。

図3bで示した例では階層が2つで、同じ水準のノードの他の科目区分での必要単位数を計算すれば



a) 同一水準のノードだけで探索した場合



b) 下位科目区分まで探索した場合

図4 下位科目区分まで探索した場合との比較

よいが、階層が深くなると、その下位の科目区分に対しても探索が必要となる。そのような例を表2に示した。この例では「教養」から見た場合、「合計」の不足14単位を補うには、「専門」での2単位不足により2単位は確保できるため、同一水準のノードだけで見れば最大でも12単位でよい(図4a)。しかし「専門」の下位科目区分をみると、「基礎」で8単位不足しているため、その上位の「専門」でも8単位は確保できる。したがって「教養」で補うべき単位の最大は14から8を減じた6単位となる(図4b)。

このように、他の科目区分での不足単位数を計算しながら、当該の科目区分であと何単位必要なかの計算を行う必要がある。木構造で考えると、下位の部分木の探索が必要となり、一種のバックトラックによる再帰的アルゴリズムが必要となる。

#### 4. 単位充足判定の方法

##### 4.1 単位チェックの3段階判定

単位チェックの判定において、あと何単位とる必要があるかを範囲で示すと誤解を招いてしまう恐れがある。そのため、単位数は明示せず以下の3段階での判定を行うこととした。

- ・当該の科目区分において単位充足し、かつ、合計単位数を満たすために取得する必要がない。(○; 緑判定)
- ・当該の科目区分において単位が充足していないか、全体の合計単位数を満たすために必ず取得の必要がある。(×; 赤判定)
- ・当該の科目区分においては単位が充足しているが、全体の合計単位数を満たしてない。ただし、他の科目区分によっても合計単位数を満たすことが可能であるため、当該の科目区分で必ずしも取得する必要はない。(△; 黄判定)

合計一致型の場合、他の科目区分での単位取得状況とは関係なく単位の充足判定ができるため、当該の科目区分で単位が充足しているかどうかの判定になる。そのため黄判定が生じることなく、あらかじめ定められた最低必要単位数との比較で赤判定か緑判定かが決まってしまう。

主として問題になるのは、合計不一致型の場合、緑判定か黄判定のいずれになるかの判定である。判定には、他の科目区分の単位取得状況によって、ど

表3 p\_max, p\_min の計算と判定の例

| 科目区分   | 必要最低<br>単位 | 取得単位 | p_max | p_min | 判定   |
|--------|------------|------|-------|-------|------|
| 教養(合計) | 20         | 14   | 20    | 20    | ×赤判定 |
| 人文     | 4          | 6    | 10    | 6     | △黄判定 |
| 社会     | 4          | 6    | 10    | 6     | △黄判定 |
| 自然     | 4          | 2    | 8     | 4     | ×赤判定 |

こまで単位を取る必要があるかの情報が必要となる。

##### 4.2 動的パラメータの導入

そこで、各科目区分の単位の取得状況によって動的に変化する見込み単位数の最大と最小の値を算出することによって判定ができるようにした。前者を最大見込み単位数、後者を最小見込み単位数とよぶ。

- ・最大見込み単位数 p\_max

合計不一致型の場合、合計単位数を満たすためには、最低必要単位数以上の単位を取得しなければならない。それは、合計に算入される他の科目区分の単位取得状況によって変化する。他の科目区分で必要単位が多く残っていれば、そこで確実に取得単位が増えるため、当該科目区分で新たに取得する単位数は少なくてもよいが、そうでない場合は多く取得しなければならない。そのため、各科目区分において最大で何単位まで取得が見込まれるのかを算出する。たとえば、図3bで「社会」や「人文」から見ると、「自然」はまだ不足単位を2単位残しているため、2単位は確実に増える。そのため、合計6単位不足を満たすのに、「社会」や「人文」では、あと最大4単位とればよく、最大見込み単位数はすでにそれぞれで取得している6単位を加え10単位となる(表3)。一方、「自然」から見ると、「社会」や「人文」では不足単位がないため、「自然」では合計を満たすには最大であと6単位必要で、最大見込みは取得している2単位を加え8単位となる(表3)。

- ・最小見込み単位数 p\_min

各科目区分においての最低条件は、当該の科目区分で定められている最低必要単位数をまず満たすことである。したがって、最小見込み単位数は最低必要単位数を下回らない。図3bの「自然」では、最小見込み単位数は4単位となる(表3)。ただし、すでに最低必要単位数を超えている場合、見込みとしてはその取得単位を下回ることにはあり得ないため、取得単位数が最小見込み単位数となる。「社会」や「人文」では必要最低単位はいずれも4単位であるがす

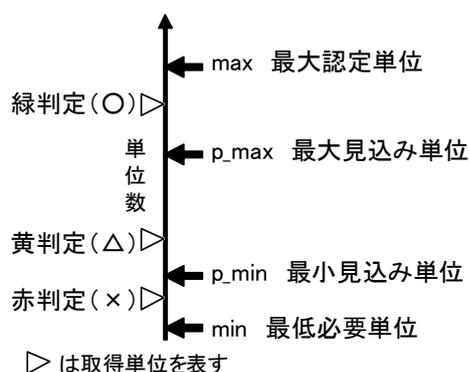


図5 2つの動的パラメータと単位充足判定ルールの関係。取得単位数によって判定が異

でに6単位取得しているため、最小見込み単位数は6単位となる(表3)。

ここで、最大認定単位(max)、最低必要単位数(min)、最大見込み単位数(p\_max)、最小見込み単位数(p\_min)の関係を図5に表した。合計一致型の場合、これら4つのパラメータは同じになるが、合計不一致型の場合、2つの見込み数はmaxとminの間で動的に変化することになる。

### 4.3 単位の充足判定のルール

2つの動的パラメータの導入によって、単位判定は以下のルールに集約される(図5)。

- ・ 緑判定 取得単位数が最大見込み単位数以上の場合
- ・ 赤判定 取得単位数が最小見込み単位数に満たない場合
- ・ 黄判定 取得単位数が最大見込み単位数未満で最小見込み単位数以上の場合。

取得単位数が最大見込み単位数以上であるということは新たに単位を取得する必要がないということである。また、最小見込み単位数に満たないということは、当該科目区分から必ず追加取得が必要であるため、赤判定になる。最大見込み単位数と最小見込み単位数の間に取得単位数がある場合は、上位の科目区分の合計を満たす必要があることを示しているが、それが当該区分でなくてもよく、他の科目区分で追加取得がなされれば、新たに取得の必要はなくなる。このような場合黄判定となる。

表3に判定例を示した。「人文」と「社会」は、取得単位6単位がp\_minとp\_maxの間にあるため黄判定となるが、「自然」と「教養(合計)」はいずれもp\_minを取得単位数が下回っているため、赤判定となる。このように2つの動的パラメータと実際に取得した取得単位を比較することによって、判定が可能

となる。

## 5. 動的パラメータの算出アルゴリズム

これまで述べてきたように、単位充足判定においては、動的パラメータの2つの値を算出することによって判定が可能となる。問題は動的パラメータをどのように算出するかである。それは以下の3つのフェーズで行われる。ここでは、表2に示した科目体系を用いて説明をする。p\_minやp\_maxの算出例を含めて、改めて表4に示した。なお、ここでは、取得単位という表現を認定単位と表している。科目体系によっては、取得単位すべてが卒業に必要な単位にならない場合があり、卒業に必要な単位として認定されるものだけを使って判定を行うため、認定単位としている。

### 5.1 第1フェーズ p\_minとp\_maxの仮設定

まず、p\_minとp\_maxの値を、他の科目区分を考慮せずに仮設定を行う。設定はいずれの値も、認定された単位と最低必要単位の大きいほうとする。p\_minは、最低でも最低必要単位をとることになるので、最低必要単位以上になる。ただし、先に述べたように、すでに最低必要単位以上の単位が認定されていれば、見込みとしては、それを下回ることではなく、すでに認定された単位をp\_minの値として設定する。

一方、p\_maxは見込みとして最低でも最低必要単位を取るが、すでにそれを越えた単位を取っている場合、見込みとしてはその値を下回ることがないため、p\_minと同様すでに認定された値を設定する。

### 5.2 第2フェーズ 他の科目区分での最小取得見込みを考慮したp\_maxの更新

p\_max更新を再帰的なアルゴリズムとして実装したのが、図7、図8である。この計算処理では、まず、他の科目区分で少なくとも何単位増えるのかを他の科目区分ごとに算出する必要がある。それが図7のアルゴリズムである。最小見込み単位数から認

表4 3つのフェーズにおけるp\_max, p\_minの更新例

| 科目区分 | 必要最低単位数 | 取得単位数 | 第1フェーズ p_max, p_min の仮設定 | 第2フェーズ p_maxの更新 | 第3フェーズ p_minの更新 |
|------|---------|-------|--------------------------|-----------------|-----------------|
| 教養   | 10      | 8     | 10                       | 14              | 10              |
| 専門   | 20      | 18    | 20                       | 30              | 26              |
| 基礎   | 10      | 2     | 10                       | 14              | 10              |
| 実践   | 0       | 16    | 16                       | 20              | 16              |
| 合計   | 40      | 26    | 40                       | 40              | 40              |

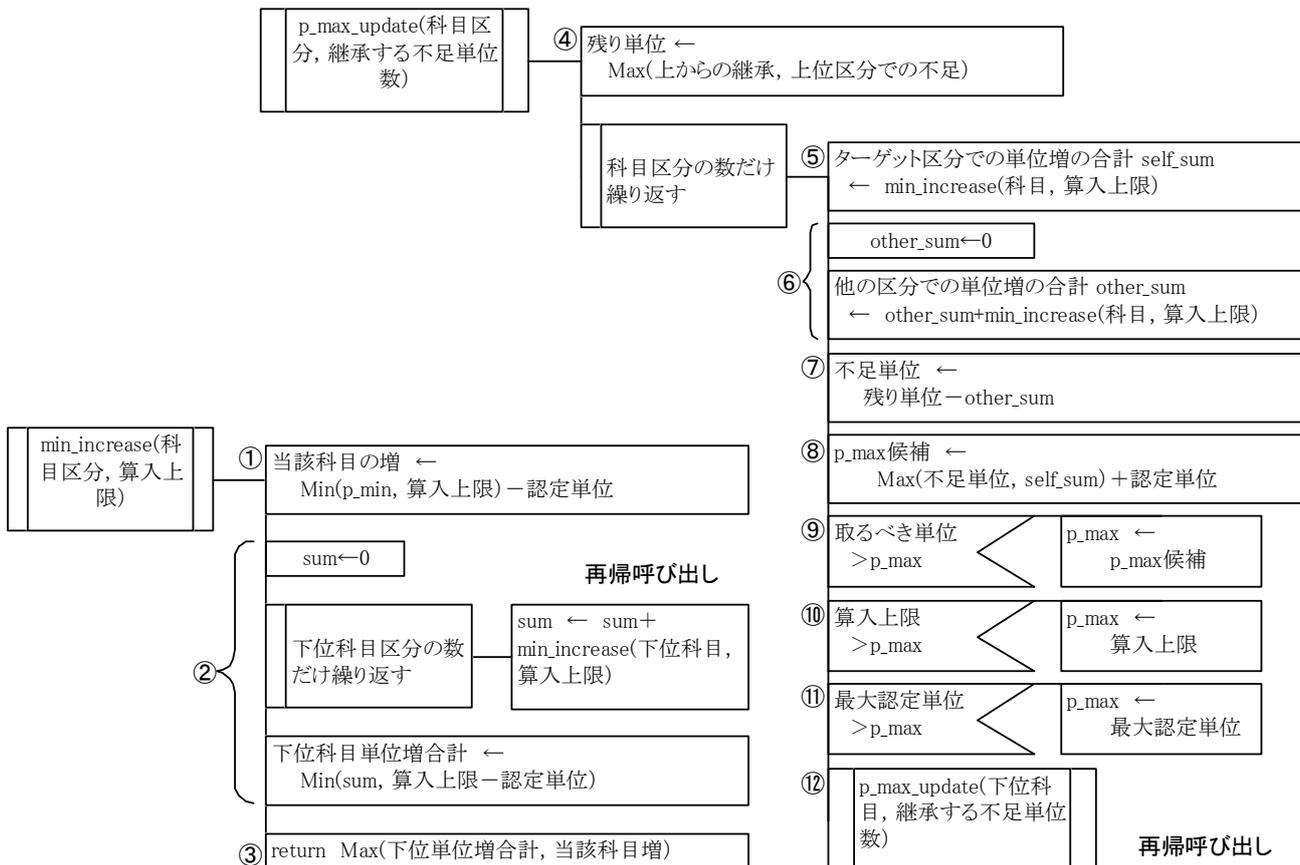


図7 少なくとも何単位増えるのかを合計するための再帰的アルゴリズム

図8 最大見込み単位数 (p\_max) 更新のための再帰的アルゴリズム

定単位数を減じれば何単位増えるのかがわかる (図7①)。ここで増える単位は、算入上限があれば、それを超えて上位の科目区分に算入されないため、それを考慮している。表4の例の「専門」で何単位増えるかは、第1フェーズでの  $p_{min}$  の値 20 から認定単位 18 を減じた値 2 単位となる。これが当該科目の増となる。

しかし、先に述べたように下位科目区分も探索する必要があり、下位科目区分での不足単位があれば、その分増えることになるため、下位科目区分での増の合計を再帰的に最下層までたどって合計を行う (図7②)。表4の例では、「基礎」と「実践」の科目の2つの下位科目区分の分だけ合計を算出する。

「基礎」は8単位増が見込まれ、「実践」では単位増無しで、この合計(sum)は8単位となる。ここでも算入上限を考慮して、下位科目単位増合計が算出される。この値と当該科目増の大きいほうの値がその当該科目区分での増分になる (図7③)。下位科目区分(「基礎」と「実践」)の合計8単位増と「専門」の2単位増を比較して、大きいほうの8単位が上位科目区分に増える単位数となる。

ここで再帰的に呼び出されている `min_increase` の関数は2つの引数を持っており、科目区分を指定する引数と算入上限の制約を指定することになっている。このアルゴリズムは実際には次に説明する  $p_{max}$  の更新のアルゴリズムで利用される。

$p_{max}$  の更新では、上記で述べた他の科目区分で算出される単位増の値を見込んで、最大何単位取得を見込めばよいかを算出する。そのアルゴリズムが図8である。まず、上位から残単位を継承し、同時に上の科目区分での不足分と比較し、大きいほうの値を残り単位としてまかなう必要がある (図8④)。ここでは「教養」と「専門」に関して注目すると、その上位区分は「合計」になる。「合計」では、14単位不足している。さらに「合計」の上位でもっと多く単位不足が生じている可能性があることも考慮して、その不足単位(上からの継承)も継承しなければならない。その単位数が多ければ、その単位数を下位でまかなうこととなる。ただし、表4の例の場合、「合計」の上は存在しないため、「合計」で不足している14単位を下位の「教養」と「専門」でまかなうことになる。

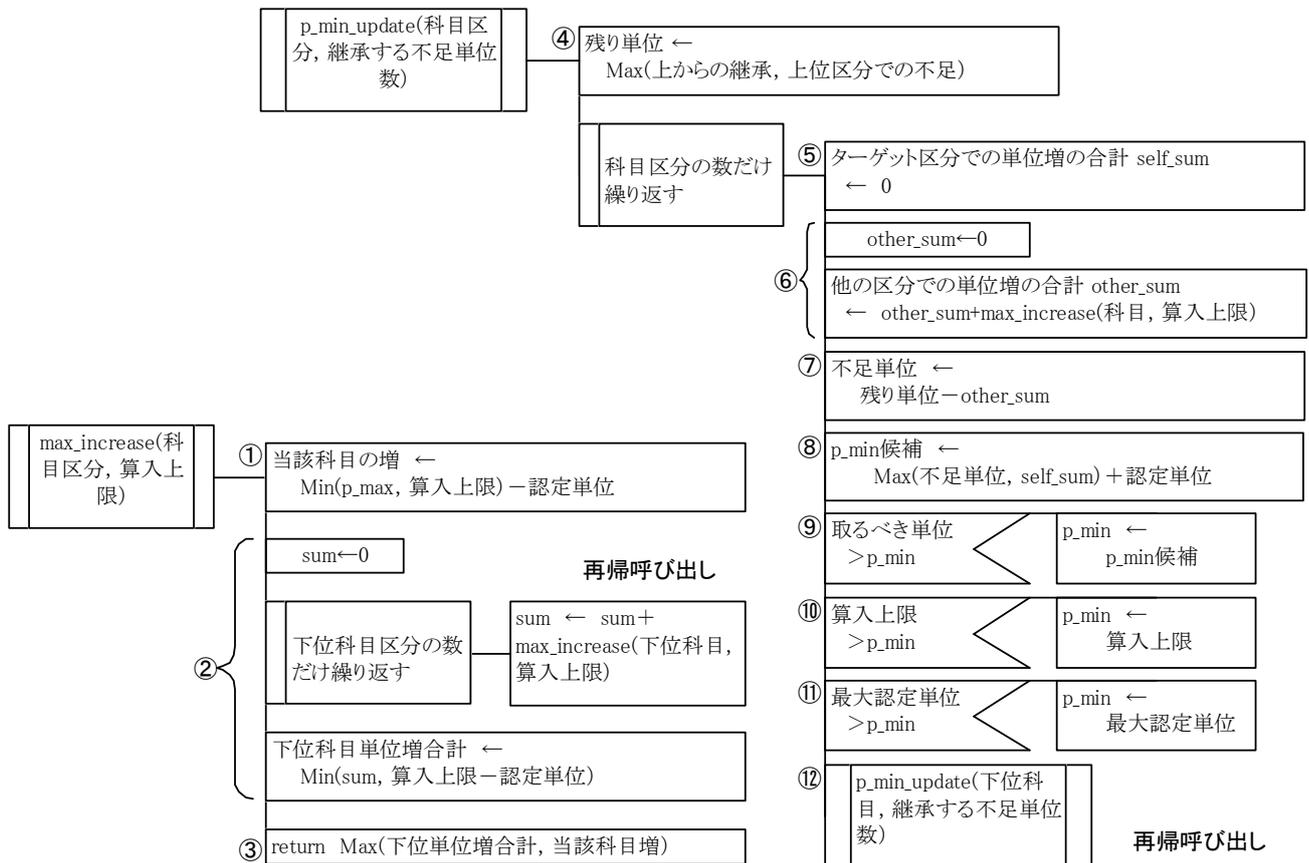


図9 最大何単位増えるのかを合計するための再帰的アルゴリズム

図10 最小見込み単位数 (p\_min) 更新のための再帰的アルゴリズム

次に、ここでの科目区分の「教養」と「専門」の2つについて、自分のところの下位科目区分での単位増と他の科目区分の単位増を計算して、それらで不足分をどこまでまかなえるかどうかを算出する。そのために、以下の処理を繰り返すことになる。

まず、ひとつの科目区分をターゲット（ターゲット区分）にして、その科目区分での不足単位（あと何単位増えるか）を計算する (self\_sum)。たとえば、「教養」について計算する。ここでは図7に示したアルゴリズムを使い、それよりも下位の科目区分も探索して何単位増えるかを算出する (図8⑤)。「教養」には下位科目は存在しないため、この場合下位探索は行わなくてよく、「教養」では2単位増だけとなる。

次に他の科目区分でどれだけ単位が補えるかを合計する。ここでは他の科目区分は「専門」だけであるが、複数あれば複数の科目区分の合計 (other\_sum) を算出する (図8⑥)。ここでも、図7に示したアルゴリズムを使って再帰的に下位科目区分まで探索をしていく。ここでの科目区分「専門」の計算過程は、図7のアルゴリズムを説明したときにすでに述べた通りである。Other\_sumは8単位となる。

この値は「合計」の科目区分の残り単位を減じるのに使え、その残り分が不足単位となる (図8⑦)。この場合、14単位から8単位を減じた6単位となる。この値がターゲット科目区分「教養」で取るべき最大の単位数となる。ただし、先ほど算出した「教養」自身で増える2単位 (self\_sum) がそれを上回っていけば、そちらのほうが取るべき最大単位数となる。そして、その値に認定された単位を加えた単位が p\_max の候補値となる (図8⑧)。つまり、6単位と2単位の大きい方の6単位にすでに認定されている8単位を加えて、14単位が p\_max の候補となる。そして、それがすでに設定されている p\_max よりも大きい場合、この候補値で更新することとなる (図8⑨)。後は、算入上限 (図8⑩) や最大認定単位 (図8⑪) を超えないようにする必要がある。そして、さらに、下位の科目区分でも同様の処理を行っていくために再帰的に行う (図8⑫)。

この一連の処理をターゲット科目区分を順々に変えて行く。ここでは、あとは「専門」をターゲット科目区分として処理するだけである。「専門」がターゲットとなると、さらにその下位科目区分「基礎」

や「実践」でも同様の処理を行う。その結果、表4に示した値で  $p_{max}$  の更新がなされる。

ここでの再帰的関数  $p_{max\_update}$  は科目区分の指定の引数と下に継承していく不足単位数を引数として指定することとなっている。

### 5.3 第3フェーズ 他の科目区分での最大取得見込みを考慮した $p_{min}$ の更新

第2フェーズでは、他の科目区分での単位の取得可能性の最低を考慮したが、第3フェーズでは、他の科目区分での単位の取得が最大の可能性を考慮することになる。他の科目区分で単位を最大限取得しても、まだ足りない可能性があった場合、足りない単位分は当該の科目区分で取得しなければならない。その単位数がどの程度であるのかを計算して、当該科目区分での最小見込み単位数  $p_{min}$  を更新する。

第3フェーズのアルゴリズムを図9、10に示したが、第2フェーズのアルゴリズムとほとんど類似しており。以下の2つの点のみが異なる。

まず、第2フェーズが他の科目区分での最小取得見込みを考慮するのに対して、第3フェーズでは、他の科目区分での最大取得見込みを考慮するため、何単位増えるかの算出において、第2フェーズで  $p_{min}$  と認定単位の差で計算していたのを  $p_{max}$  との差で計算するところが異なるだけである(図9①)。もうひとつは、ターゲット科目区分の合計を0とすることである(図10⑤)。これは、不足単位との比較において(図10⑧)、不足単位がマイナスを示すこともあり、0を超えないようにするためにすぎない。

表4における「専門」の  $p_{min}$  の更新のプロセスを追うと以下ようになる。残り単位は「合計」14単位で第2フェーズの場合と変わらない。他の区分での単位増の合計(図10⑥)は、対象となるのが「教養」だけであり、今回は、 $p_{max}$  と認定単位との差(図9①)で算出され6単位となる。そして、残り単位14との差の不足単位が8となる。この値に「専門」の認定単位18を加えたのが、 $p_{min}$  の候補値で26単位となる。第1フェーズで設定された  $p_{min}$  20単位よりも大きいため、26単位で更新されることになる。後の科目区分も同様の計算を行うことになるが、いずれも他の科目区分での単位増が残り単位を上回り、不足単位が負の値を示す。そのため、 $p_{min}$

の候補は、現在値を超えることはなく、更新されないこととなる(表4)。

## 6. 考察

2つの動的パラメータを導入し、その算出を再帰的アルゴリズムによって実装できた。そのため、3段階の判定が可能となった。図1に示した例は、表4に示したカリキュラムに語学の科目を加えたもので、語学は最大認定単位10単位となっており、これ以上単位を取得しても、上位の合計には貢献できず、すでに10単位取得しているため、緑判定(○)となっている。このように合計不一致型の体系であっても、どの科目区分でさらに単位取得が必要であるのかが簡単にわかるようになった。

ただし、他の科目区分に単位を流用するプログラムが十分に完成しておらず、今後さらに検討をする必要がある。また、現在のプログラムでは、科目区分ごとに、他の科目区分で何単位取得する可能性があるのかを計算しており、一度算出した値を再度計算し直す形になっており、無駄な計算を無くすような工夫も必要である。

## 参考文献

- 1) 梶田将司・角所考・中澤馬志・竹村治雄・美濃導彦・間瀬健二：高等教育機関における次世代コース管理システムの構築に向けて、日本教育工学会論文誌，31(3)，297-305，2007.
- 2) 吉牟田裕：Webによる履修登録システムの構築と運用—ウェブデザインからDBMS、運用体制まで—，富山国際大学地域学部紀要，4，111-116，2004.
- 3) 松尾太加志：汎用的な卒業単位チェックシステム，教育システム情報学会第34回全国大会講演論文集，168-169，2009.
- 4) 栗原和夫：Webアプリによる卒業要件単位計算，情報学研究(朝日大学経営学部電子計算機室年報)，16，21-28，2007.
- 5) ヴィルト，N.(浦昭二・國府方久史 訳)：データ構造とアルゴリズム，近代科学社，1990.