

字は stroke を合成. 1字平均 900 バイト

4,000 字位用えている. 文字もいろいろ

1 ページ 組版のに

1 行 90sec. 組版のしやすさでペー
ジ組版と植字工より早い

C 3 汎用自動組み版システムの一例

昆野 誠司, 関谷 敏郎 (管理工学研究所)

望月 昭 (大倉商事)

1. はじめに

組み版作業は、ニュース性のある記事を扱う新聞、週刊誌と、単行本、辞典や、名刺などの「はもの」とでは、その工程が異なってくるが、ここでは 1 つ 1 つについて詳細に述べることを避け、現在、活版印刷で行なわれている、編集者と植字者の仕事のみ見てみる。

編集者と呼ばれる人は、原稿に、その文字の大きさ、書体を書き込み、その他、行の幅、行間隔、字下げ部分、強制的に改行する個所など植字に必要な情報を書き込む。そして必要とあればページのレイアウトを作成し、原稿とともに植字者にわたす。

植字者の作業を一口でいってしまえば、「ページという枠の中に原稿の 1 字 1 字を入れて行く」ということになる。ただしページレイアウトがない場合、またはあっても、原稿の文章があふれたり、スペースが余った場合、枠の調整の作業は、困難になり、植字者の熟練と、知能が要求されることになる。

この植字者の仕事を、自動植字装置と、これをコントロールする汎用計算機のソフトウェアとで代行させようとして開発されたものが、組み版ソフトウェアである。

OCS (大倉組み版システム) は VIDEO COMP という自動植字装置を働かせるために開発された組み版ソフトウェアである。

組み版ソフトウェアの中心になる組み版プログラムは HITAC 8400 (コア 128K) 用にコーディングされたものである。

OCS では、編集者が、植字者にわたす原稿と組版指定およびページレイアウトを OCL (大倉組み版言語) を使って表わし、この OCL で表現された組み版指定およびテキストを組み版プログラムが入力し、組み版指定を解釈しながらテキストを組み版処理することになっている。

この OCS を開発するに当たって、当然のことながら問題となった点は、編集者と植字者の間で取り交される組み版指定は、かなり植字者の熟練と知能を要求されている。そこで、植字者の代行としてそのまま組み版プログラムを設計することは非常に困難に思われたので、OCS なりに組み版に対する考え方を検討し、OCL と組み版プログラムの設計を行なった。OCS の開発は昭和 44 年に始まり、45 年 11 月に最初のシステムが出来上がった。その後編集者の作業を軽減させること、そしてさらに、もっと汎用性を求めて改良し、現在第 3 期の改良を行なっている。

2. OCS の組み版に対する考え方

2.1 OCS の構成

OCS は基本的に 3 つのプログラム群で構成されている。

(i) EDITER

これは各種漢字テレタイプ、紙テープパンチ、カードパンチ、磁気テープ出力装置などから出力された紙テープ、カード、磁気テープ上に記録されている文字を OCS 特有のコードに変換し、ファイルにまとめ、組み版プログラ

ムの入力を作成する。またゲラの校正によって得られた修正データによりファイルの修正を行なうという機能も持っている。

(ii) COMP…組み版プログラム

これは組み版そのものを行うプログラムであるが、2つに大きく分けられ、1つはOCLの解説と、もう一つは解説によって得られた組み版データによって文章の組み版をする。このプログラムは、自動植字装置の機種を限定しないように設計したもので、編集者の指定したページという枠の中で組み版する。

(iii) VCL…ビデオコントロールプログラム

これは組み版プログラムが編集者の指定したページといういわゆるロジカルな世界で組み版したものを、自動植字装置の出力の窓の大きさに分けたり、自動植字装置をコントロールするためのデータを追加したりする機能を持っている。

2.2 OCLの概略

OCL(大倉組み版言語)の第1の特徴は、組み版という仕事の植字のアルゴリズムを直接記述するものでないことである。

物の形、属性を記述するための言語といえる。またOCLは汎用を目的としている。したがって種々のページレイアウト、割付けを表現できるようにしている。OCLの中には現在約90種の指定があるが、これを大別すると、枠指定と、編集指定とに分けられる。

(i) 枠指定

枠指定はページレイアウトを記述するために用意した指定である。

(ii) 編集指定

編集指定は、文字の属性、割付けの形態などを記述するために用意した指定で、この中に特別4つの指定だけ、植字のアルゴリズムを記述するのに助けとなるように用意した。

OCLでは、従来の活版印刷、写植印刷では自明の事としていたような点についても新たなる定義を行なっているので、それらについてふれてみる。

2.2.1 文字の属性

組み版という仕事で文字を取り扱う場合に、文字には基本的に3つの属性があると考えられる。3つの属性とは、「字種」「書体」、「大きさ」である。

(i) 字種

いろいろの「い」とか、アルファベットの“A”といったもの。

(ii) 書体

日本語(漢字、かな文字)でいえば、「明朝体」、「ゴシック(呉竹)体」、「清朝体」、「教科書体」、…といったものがあるが、明朝体でも「XXの明朝」といったように多数の書体がある。

(iii) 大きさ

普通、印刷で使われている文字の大きさを示す単位には、「ポイント」と「号」がある。ポイントで言えば、4ポイント、5ポイント、……で、1ポイントは $1/72$ インチである。

活版印刷では、この3つの属性値がそろって1つの活字が定まる。たとえば、XXの明朝(単に明朝ともいう)で、8ポイントの「は」と言えば1つの活字が定まる。

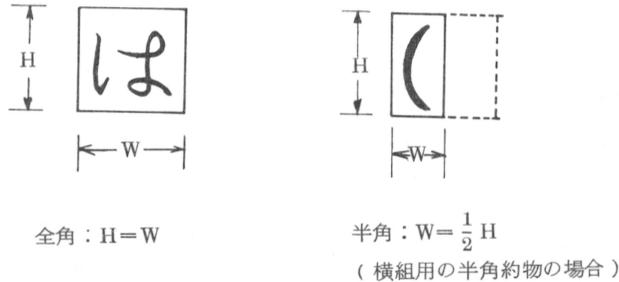
この3つの属性の他に、文章中その文字がどんな働きをするものであるかを示す、言ってみれば「機能」という属性がある。たとえば、ルビ、上つき、下つき文字、他の文字に重ね合わせる重畳文字、あるいは合成して1つの文字として扱い文字といったものは、本文の普通の文字と別の扱いをしなければならない。

OCLではこのような4つの文字の属性を表現できるような指定がある。

2.2.2 文字の大きさと形

活版印刷でも、写真植字印刷でも、記録媒体上に占める1文字の面積の形は、「矩形」である。

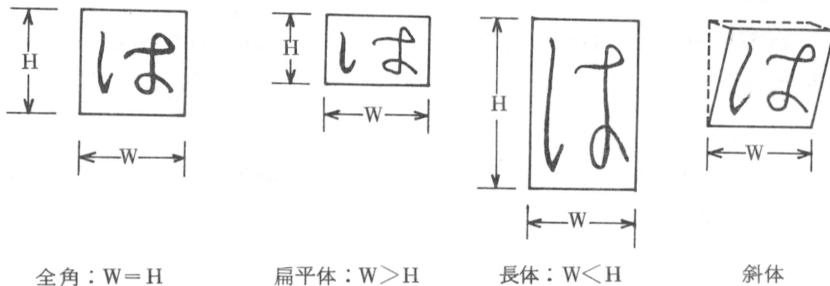
漢字やかな文字の原則は、「正方形」であり、これを「全角」と呼んでいる。全角に対して、日本字の約物などでは「半角」の文字がある。



第 2.1 図

第2.1図に示す通り、全角とは、文字の縦の長さ（高さ）と横の長さ（幅）が等しい場合で、半角、4分角は高さ、あるいは幅がそれぞれ他方の $\frac{1}{2}$ 、 $\frac{1}{4}$ の矩形である。

これとは別に、写真植字でよく使われる文字の描き方に、「扁平体」、「長体」、「斜体」がある。これらは、原形が全角である文字に光学的な歪みを加えたもので、第2.2図に示す通り。



第 2.2 図

以上のことから分かるように、ここに1つの仮定を設ける。

仮定： 文字の形は、どの書体に限らず、矩形とする。文字の高さと幅をもって矩形の「縦の長さ」と「横の長さ」とする。斜体の形を平行四辺形とせず、この平行四辺形を含む横長の矩形とする。

2.2.3 縦組と横組、縦書きと横書き

記録媒体上で、文章の文字を縦方向に並べることを「縦組」と呼び、横方向に並べることを「横組」と呼ぶ。

これに対して、文字の上下を互に隣接して並べることを「縦書き」と呼び、左右を互に隣接して並べることを「横書き」と呼ぶ。縦組はほとんどの場合、縦書きであり、同様に横組は横書きである。しかし、和文の組み版の場合、第2.3図に示すように、縦組で横書きの例がある。横組で縦書きの例を筆者は未だ見たことがない。

い
ろ
は
に
ほ

あいうえお

S
O
は
A
B
C
D

縦組，縦書き

横組，横書き

縦組，横書き

第 2.3 図

ここで定義した「縦書き」とおよび「横書き」という言葉は，OCL（大倉組み版指定言語）で使用している言葉で，一般に用いられている概念とは多少異なるかもしれない。

2.2.4 行とコラム，およびその大きさ

文章の文字を読む順番に 1 列に並べたものを「行」と呼ぶ。縦組の場合は，上から下へ，横組の場合は左から右へ並べる。

そして，行を 1 列に並べたものを「コラム」と呼ぶことにする。縦組の場合は，行を右から左へ，横組の場合は行を上から下へ 1 列に並べる。

ここで言う「コラム」とは，新聞，あるいは雑誌などで「段」と呼んでいるものと同じである。「段」という概念を拡張したために，コラムと呼ぶことにした。

文字の形を矩形と仮定したが，ここでさらに仮定を加える。

仮定： 行および，コラムの形を矩形とする。

組み版では，行の大きさを，そこに並べる文字の大きさから定め，そしてコラムの大きさを，そこに並べる行の大きさから定めるということを一般的には行なわない。むしろ，積極的にコラムの大きさを定め，行の大きさを定め，そして行の中に文字を納めるという考え方を取る。これら，行やコラムを，文字を納めるための「枠」と考える。

矩形の行の縦方向の長さ，と横方向の長さを次のように定義する。

行で，文字の並ぶ方向を「字並び方向」と呼ぶ。したがって，縦組の行では，縦方向が字並び方向，横組の行では，横方向が字並び方向となる。矩形の行の字並び方向の長さを，「行の長さ」と呼び，これに直交する他辺の長さを「行の幅」と呼ぶ。

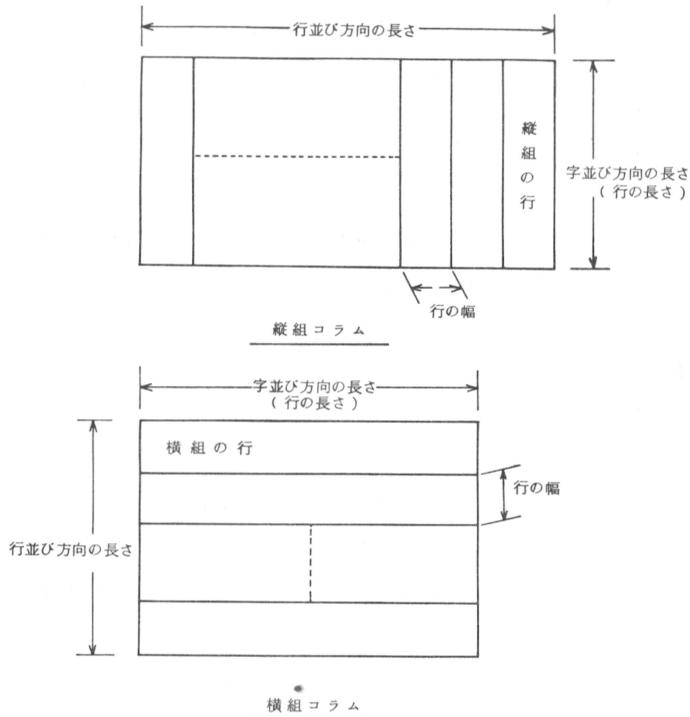
行が 1 列に並んだものをコラムと呼んでいるが，矩形のコラムの，行が並ぶ方向を「行並び方向」と呼ぶ。コラムで，縦組の行が 1 列に並ぶ場合は，コラムの横方向が行並び方向，横組の行が 1 列に並ぶ場合は，コラムの縦方向が行並び方向となる。行の字並び方向をもって，コラムの字並び方向とする。

コラムの字並び方向の長さを「コラムの長さ」，行並び方向の長さを「コラムの幅」と呼ぶこともある。

2.2.5 4 重の枠

OCL では，1 ページ内の文字は，基本的に 4 重の枠の中に位置きめされるものと仮定している。

一番外側の枠を「ページ」と呼ぶ。その内側に，任意の位置にある枠が「コラム」という枠であり，その中に「行」という枠があり，行の中に「マス」という枠があって，その中に文字が存在するものとしている。マスの中には文字 1 字，重畳した文字とされた文字，合成された文字が入る。従って，マスの大きさは，その中に入る文字 1 字，または重畳文字，合成文字の大きさによって定める。



第 2.4 図

2.2.6 コラムの中のコラム，行の中のコラム

OCLでは，基本的には4重である枠について，コラムの中のコラム，行の中のコラムという拡張を認めている。このコラムの中のコラム，行の中のコラムを，4重の枠のコラムと区別して「浮動コラム」と呼ぶ。

OCLでは，ページ内のどの文字も必ず行という枠の中に入っているものとしている。そして行もまた必ずコラム，または浮動コラムという枠の中に入っているものとしている。

そして，編集者がページ内で，文字，行の位置ぎめを直接できないようにしてある。

コラムと浮動コラムの考え方の相違は，コラムはページレイアウト上で，ページ内での位置を固定的に指定する枠であり，浮動コラムはコラム内で相対的な位置を指定するか，あるいは全く，その位置ぎめを組み版プログラムにまかせる枠である。

2.2.7 ページ内の文字の位置ぎめ

組み版処理の最終目的は，ページという枠内での文字の属性値を求めることと，位置ぎめをすることであるが，4つ以上の枠を通して，次のように文字の位置を定める。

- (i) 行の基点からの文字の基点の位置
- (ii) コラムの基点からの行の基点の位置
- (iii) ページの基点からのコラムの基点の位置

基本的にはこの3段階で1つの文字のページ内での位置が定まる。コラムの中にあるコラム(浮動コラムのこと)中の文字については，更に，コラムの基点からの浮動コラムの基点の位置が求められる。

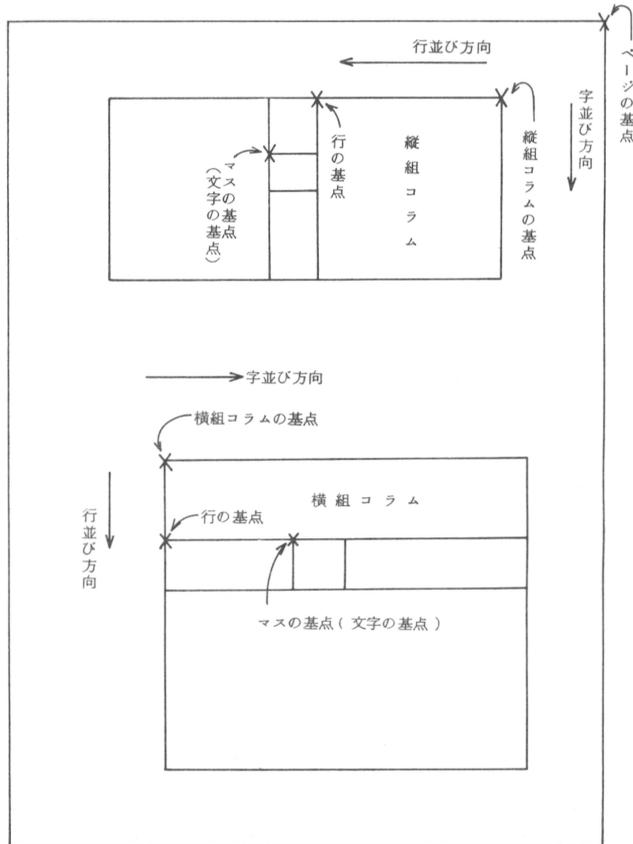
OCLでは，行という枠が，コラムという枠に含まれず，裸でページ内に存在することはない。必ずコラムか，浮動

コラム中に含まれている。同様にマスは必ず行に含まれている。したがって、1字、1行のコラム、または浮動コラムが存在することになる。

OCLでは、編集者が必ず指定しなければならない枠の基点は、コラムの基点と、特別な目的で使用する、コラムの外側に存在する浮動コラムの基点のみである。行の基点とマスの基点については陽に指定することができないようになっている。

第2.5図に示す通り、ページの基点は常に右上隅とする。

コラムの基点は、縦組コラムの場合はコラムの右上隅、横組コラムの場合はコラムの左上隅としている。



第 2. 5 図

2.2.8 シノニムとフォーマット

この「シノニム」と「フォーマット」とはOCL特有の呼び名でフォートランで言えば、文関数、外部関数に対応する概念である。

ある組み版指定とテキスト文字の文字列に名前をつけて定義しておき、組み版指定でその名前を指定することによって、組み版指定とテキスト文字の重複を避ける目的で考えられたものである。これによって、各ページの柱、ノンブル等で、あるいは表組などの見出し部分など、何回も同じ組み版指定とテキストを指定しなくてもすむ。

3. 組み版プログラムの自動処理

編集者が、ページレイアウト等で、ページ、コラム、行といった枠の大きさ、位置を指定し、組み版プログラムはそ

の枠に入れるべき文章の文字を1字1字その枠に納めて行くわけであるが、

- (i) 1行に納める文字の幅(行並び方向の長さ)が必ずしも行幅と一致しない場合がある。
- (ii) 1行に納める文字の長さ(字並び方向の長さ)の和が、行長と一致しない場合がある。
- (iii) 用意したコラムという枠に文章の文字が入り切れない場合、またコラムのスペースが余る場合がある。

といったように、ページレイアウトの段階で、文章が丁度納まる枠を設計することも、また枠に合わせて文章の長さを調整することも一般的に言うに困難である。ある定めたページレイアウトに対して、文章の文字を植字して見て、その結果、ページレイアウトを修正し、また植字するといったことを何回か繰り返さなければならない。

組み版プログラムの自動処理の主たる目的は、このページレイアウトの修正の回数を出来るだけ少なくすることにある。

OCSの組み版プログラムでは、この目的のために次のような自動処理をし、この自動処理の助けとなるような組み版指定をいくつか用意してある。

- (1) 行中で文字の行並び方向での位置ぎめ。
行内割付け指定(行並び方向)を用いる。
- (2) 行幅の調節
行に含まれる要素の行並び方向の長さの最大値を行の幅とする。
- (3) 字間調節, 語間調節, 任意の個所への空白の挿入。
1行に納める文字の長さ(字並び方向)の和が行長より小さい場合, 余ったスペースを, 字間, あるいは語間に均等に挿入する。行内割付け指定, 語間スペース指定を用いる。また余ったスペースを指定された場所に挿入する。これは可変スペース挿入指定を用いる。
- (4) 行頭, 行末への空白の挿入。
行端字空指定を用いる。
- (5) リーダ文字の挿入。
リーダー指定を用いる。
- (6) 自動的な改行。
1つの文章が複数の行を必要とする場合, 1行に納める文字の字並び方向の長さを見ながら, 自動的に改行させる際の, 行末, 行頭にかかる約物の禁則処理, 約物のぶらさげ処理, 英文, 欧文の場合の話のハイフネーション処理によるハイフン文字の挿入, 複数個連続する約物の平角扱い, 等の処理を行なう。
- (7) 浮動コラムの, 行並び方向の長さ, 字並び方向の長さの拡張, 短縮。
浮動コラムに入れる文章の長さより, 行並び方向の長さ, 字並び方向の長さを調節する。
- (8) 行間調節と任意の個所への行間空白の挿入
コラム, 浮動コラムに, 行並び方向で余分なスペースが出た場合, 余分なスペースを行間に挿入する。
行間割付け指定を用いる。
- (9) コラムの行並び方向の長さの短縮
コラムで行並び方向に余分なスペースが出た場合, (8)のような処理をせず, コラムの行並び方向の長さを短縮する。
- (10) コラムの行並び方向に直交してそのコラムを分割する。
- (11) 浮動コラムの行並び方向に直交して, また字並び方向に直交して, その浮動コラムを分割する。

(12) 追込み処理

もともと1つの文章，あるいは1つの表組の文字列のために設計したページレイアウトのページに，余分なスペースが生じたため，他のページレイアウトのページに割り当てべく予定していた文章をこのページに追込みといたことがよくある。これは2つのページのページレイアウトを重ね合わせる形になる。OCLでは，文章の途中にページレイアウト変更を示す枠指定を挿入し，(1)～(11)までの自動処理と枠の再指定を組み合わせ，追込み処理を行なう。

(13) コラム，浮動コラムという枠を囲むケイ線，行間ケイ線，タブ間のケイ線のページ内での位置，長さを自動的に決定する。

以上述べた自動処理は，主にページレイアウトと文章の長さが一致しない場合に行なうものでページレイアウトの自動修正ともいうべきものであり，文章の量，長さによって枠の大きさ，位置を適当に定めなおすものである。この自動処理を助けるためにいくつかの組み版指定を用意してある。

(A) 割付け処理指定

行内割付け（字並び方向）……行の要素の行頭よせ

…… // 行末よせ

…… // 中央揃え

……ジャスティファイ

行内割付け（行並び方向）……行の要素の右（上）よせ

…… // 左（下）よせ

…… // 中央揃え

行間割付け……コラム内での行の先頭よせ

…… // 末よせ

…… // 中央揃え

……ジャスティファイ

(B) 語間スペース指定

◎語間スペース挿入の位置を指定する指定

◎スペースの基本的な長さ（字並び方向）指定

◎ // 最大の長さ //

◎ // 最小の長さ //

(C) 可変スペース指定

◎可変スペース挿入の位置を指定する指定。

(D) リーダ指定

◎リーダ文字を余ったスペース分だけ複数個，作り出して挿入する位置を指定する指定。

(E) 約物のぶらさげ，ハイフネーション，約物の半角処理指定。

◎約物のぶらさげ処理をするか否かを指定する指定

◎ハイフネーション処理をするか否かを指定する指定

◎約物の半角処理をするか否かを指定する指定

(F) 行端字空指定

英文でも和文でも、文章の最初の行の行頭、あるいは強制的に改行した次の行の行頭、すなわちパラグラフの先頭の行の行頭をよく1字分下げしている。逆に第2行以下の行頭を何字分か下げしていることがある。また行末を何字分か上げしている場合もある。このような行端の字下げ、字上げを指定する。

またOCSの組み版プログラムでは次のような自動処理も行なり。

- (1) 本文のルビ付き文字に対するルビ文字の位置と大きさ、書体の決定.
- (2) 本文文字に対する上つき、下つき文字の位置、大きさ、書体の決定.

これらはすべてある約束に従って、決定するもので、編集者が別にルビ指定、上つき、下つき指定で位置と大きさ、書体を指定してもよい。それらの指定がない場合に、プログラムで決定する。

本 PDF ファイルは 1965 年発行の「第 6 回プログラミング—シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトの https://www.ipsj.or.jp/topics/Past_reports.html に下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載して、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思えます。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>