

5: ブロック・ダイヤグラム・シミュレータ

田 村 康 男 (早大電子計算室)

I 緒 言

“ブロックダイヤグラム・シミュレータ”は、自動制御系で代表される線型および非線型のブロック・ダイヤグラムを自動的に解く。

計数型計算機にかける問題が複雑化するにつれ、誤りのないプログラムを作製し、且つ準備時間の計算時間に対する比を下げるために、一般目的の自動プログラミングがまず必要になる。次いで、科学分野の、殊に工学分野の諸問題の解法に小数ケの命令から成る特殊目的の自動プログラミングを採用して、準備時間を更に短縮できると思う。

別の面から計数型計算機をながめると、Simulatorとして使用して便利ことが多い。標題の“ブロック・ダイヤグラム・シミュレータ”は、ブロック・ダイヤグラム計算用のサブルーチンと、特殊目的のCompiler “SWEEP”を組合わせたものである。

II プログラムの構成

ブロック・ダイヤグラム・シミュレータは下記の二種のプログラムからなる。

II-1 ブロックダイヤグラム・ルーチン (命令数 430)

ブロックダイヤグラムで表現された計算対象をそのままの形で受け入れ、所定の計算および印刷を行う。ブロック間の接続関係および関数計算等は代数式表現の“SWEEP” Compilerで書き下すことができる。

(1) 線型要素 $k/(T_2s^2 + T_1s + T_0)$

係数の与え方により2次おくれ、1次おくれ、積分、定数の4種の要素を表現することが可能である。

(A5)

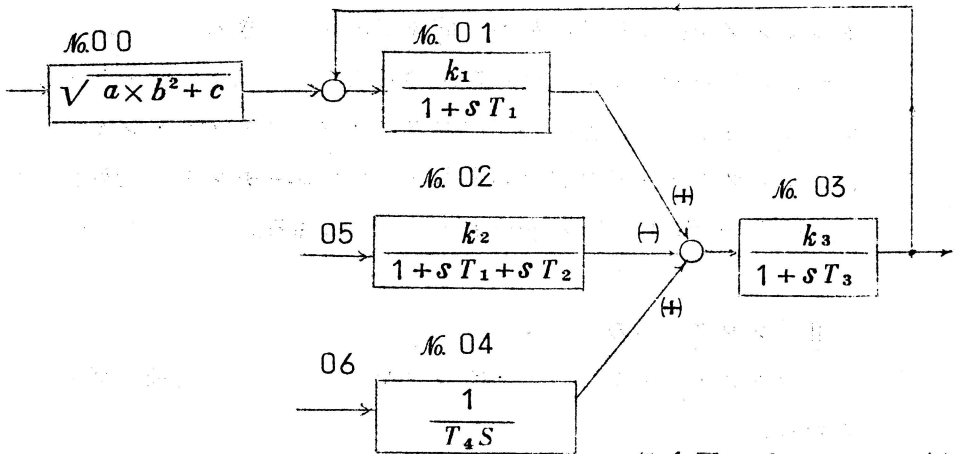
(ii) 関数計算要素

あらゆる線型、非線型計算および判断を行う。

取扱いうる問題の規模は、線型要素および関数計算要素のブロック数の合計は一応64コであるが、“SWEEP” Compilerの一部を変更すれば、記憶容量の許容する範囲内で適宜増加しうる。

II-2 特殊Compiler “SWEEP” (命令数384)

ブロック・ダイアグラム相互間の接続関係および任意の関数計算を代数式の形で受入れて、Compilingを行い、後日、直ちに計算機にかけられる形でpapertapeにPCH outする。これと、上述のブロック・ダイアグラム・サブルーチンの両者が組合わされて、ブロック・ダイアグラム・シミュレータが起きる。



第1図 ブロックの一例

“SWEEP” Statements の一例：

00 ; d01 × d03 × d03 + d02 r · fn”

01 ; 00 + 03 · 1 p”

02 ; 05.2 p’

04 ; 06.1 p”

03 ; 01 - 02 + 04.1 p”

(i) "SWEEP" の機能

(a) 代数式で表現された "Statement" を解読し、これに相当する floating pt. の命令を自動的に配列し、紙テープに punch out する。

(b) 下記の符号を識別する。

等号($=$), 終止符($.$), 加減乗除 ($+$, $-$, \times , $/$), $\sin(S)$,
 $\cos(C)$, $\tan^{-1}(A)$, $\text{Exp.}(H)$, $\ln(N)$, $\sqrt{\quad}(R)$,
 set sign plus(B), Set sign miners(T), Change sign
 (Y), Exit(E), Input(I), Print(P), STOP(Z) 等

(c) サブルーチン挿入命令 (R, U order) を 3 組用意してあるので、Fixed pt \leftrightarrow Floating pt. 間の出入、復帰改行 (Carriage Return), 等、任意の命令を随時挿入できる。ブロック・ダイアグラム・シミュレータと組合わして使う場合は、0 次, 1 次, 2 次の線型ブロックの数値積分サブルーチンへの往復に用いられる。

(d) DI_1I_2 (D : データの意) により中間値を一時貯えにすることが可能である。

(e) 通常のプリント命令以外に、"ブロック・ダイアグラム・シミュレータ" に便利なプリント記号が備えてある (Statement end にある P 或は N がそれで、 P があれば、その Statement の左辺に相当するブロックの Output はプリント and/or pCH される。

(f) 制限: (カッコ) は使用できない。代数式の右辺は "左から右へ" ただ 1 回の sweeping により解読されるので、記号の配列、特に加減乗除 ($+$, $-$, \times , $/$) の位置に考慮を払う必要がある。複雑な計算式の表現には、一時貯え DI_1I_2 を用いて原式を分解し、難点を除くことができる。

現在の所、繰返し演算の命令はない。しかし (c) 項の R.U. order により代行せしめることが可能である。

II-3 印刷サブルーチン

計算の前にカウンタの内容を指定し、且つコンソール・ボタン:

(A5)

(Transfer Control)を併用することにより、 n 回の演算に対し1回の印刷を行わせることができる。

II-4 ドラム上の番地割当て (LGP-30型計算機)

Compiling
Phase (SWEEP) 35~40 41~45 46 47

Program		"SWEEP"	S.R. 13.2	ス	Space for
Input			modi-	テ	"Target
Routine			fied	イ	<u>Program</u>
			for	ト	
			sweep	メン	
				ト	

"Source program"
from tape

Computing
phase
(simulator)

0~7 10~32 33~36 41 47 62 63

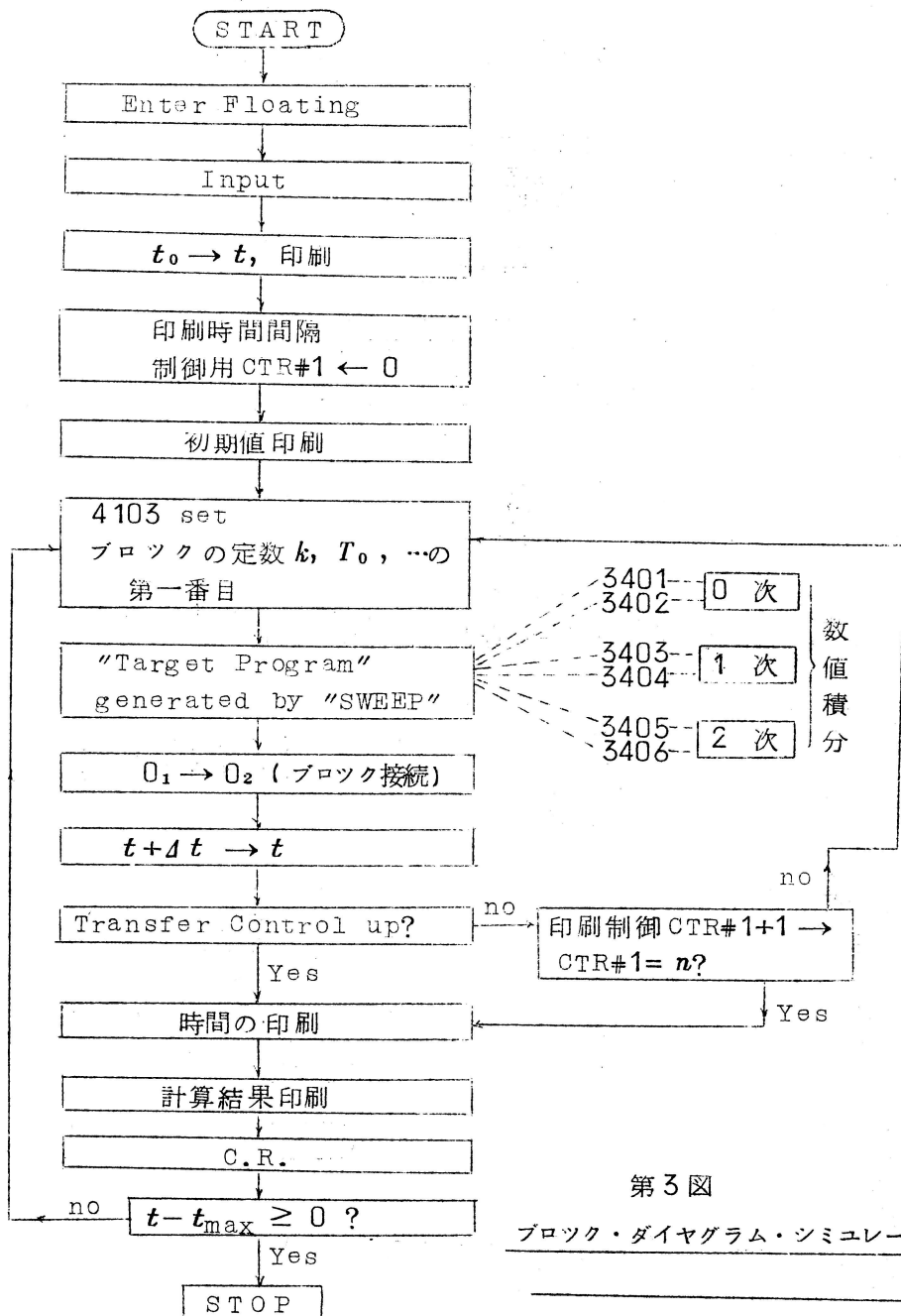
Program	Floating	Block	DO_1O_2	t_0	"Target
Input	pt.	Diagram		Δt	Program"
Routine	Interpre-	Simula-		t_{max}	genera-
	tive	tion		K	ted by
	Routine			T_0	"SWEEP"
				T_1	
				T_2	
				p	
				\vdots	
				\vdots	

一般命令部分
印刷命令部分

第2図 番地割当て

II Flow Diagrams

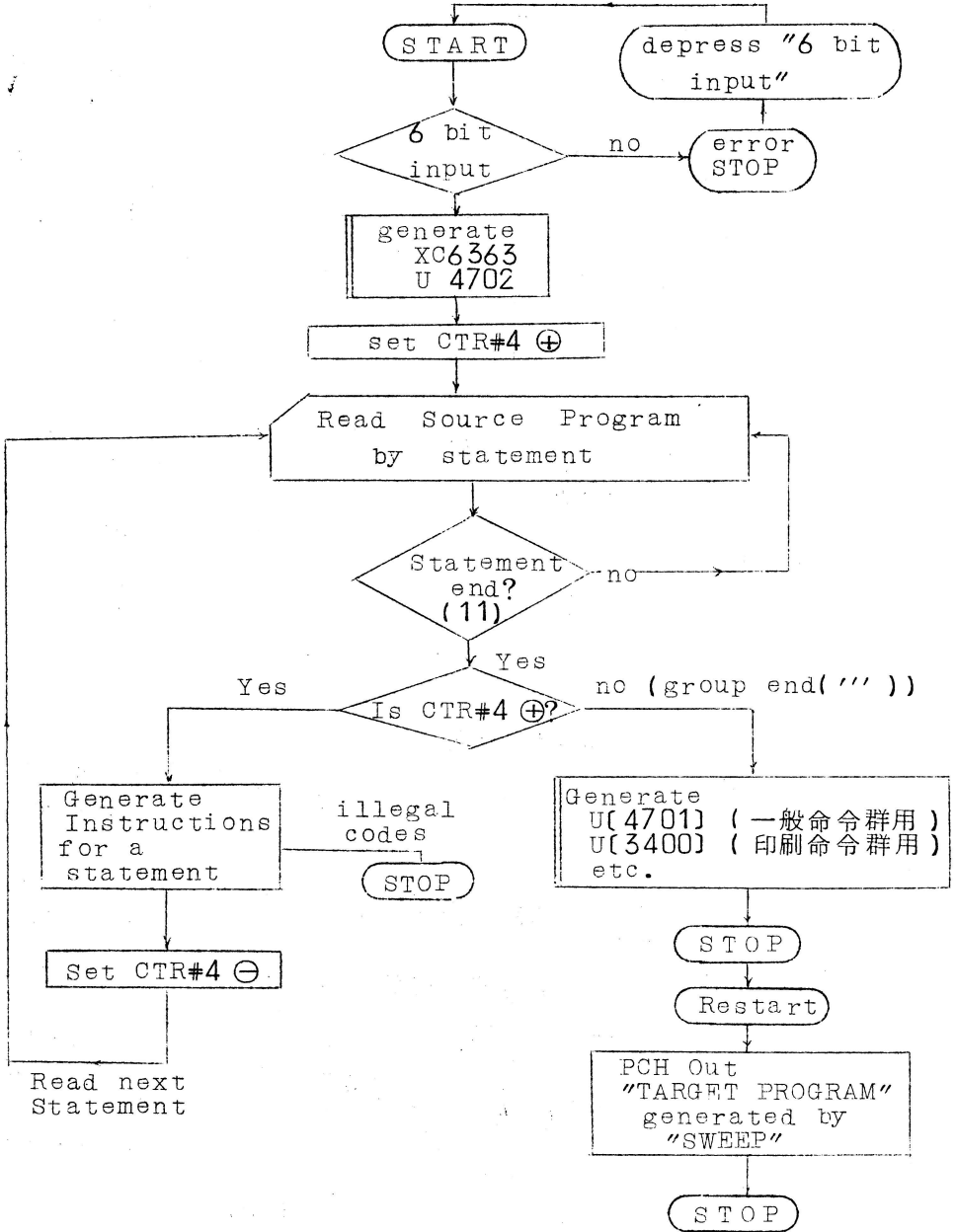
III-1 ブロック・ダイアグラム・シミュレータ



第3図

ブロック・ダイアグラム・シミュレータ

Ⅲ-2 特殊コンパイラ "SWEEP"



第4図 "SWEEP"のフロー・ダイアグラム

IV SWEEP とシミュレータの組合せ

第5図に SWEEPとブロック・ダイヤグラム・シミュレータの組合せを図示した。数値積分には Runge-Kutta 法を採用しているの、計算精度は極めて高い。

SWEEP とシミュレータの使用例

“ On-Off 制御による最小時間制御 ”

判定関数が一次式の場合の振動抑制を例に挙げる。

2階の微分方程式(1)の振動を最小時間で減衰させるのに、on-off 制御を用いる場合を考えよう。Switching 曲線に(2)式を採用するものとして、(1)式の振動系を解くことにする。

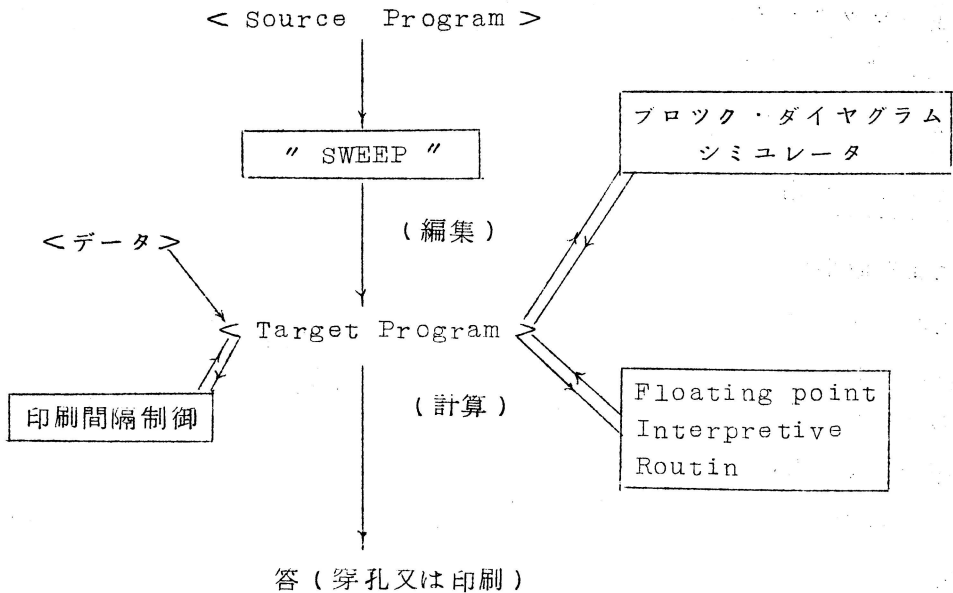
我々は第1表の Source Program を書き、データを準備すればよい。

$$\ddot{x} + 2k\dot{x} + x = 0 \dots\dots\dots (1)$$

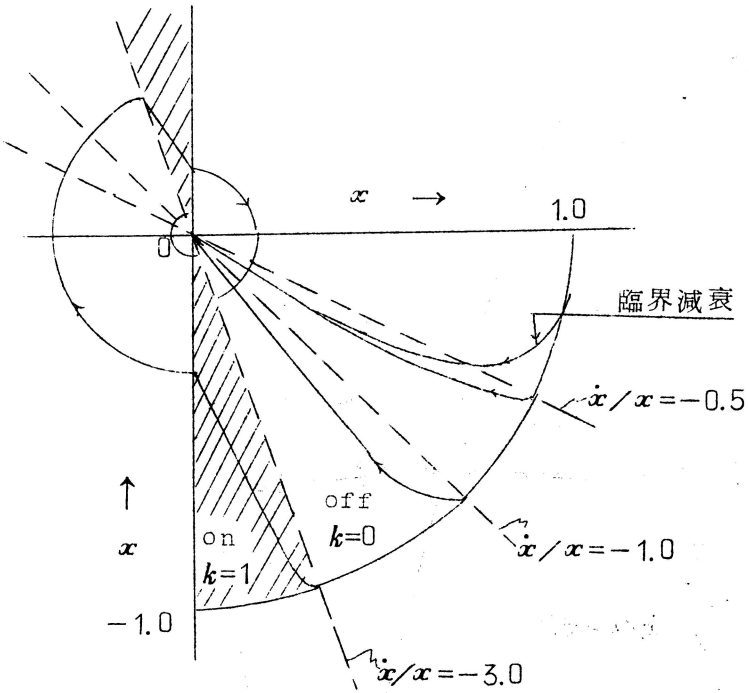
on 領域 ($\dot{x}/x < -a$) において, $k=1$
off 領域 ($x/\dot{x} \geq -a$) において, $k=0$ } (2)

パラメタ $a=0.5, 1, 3$

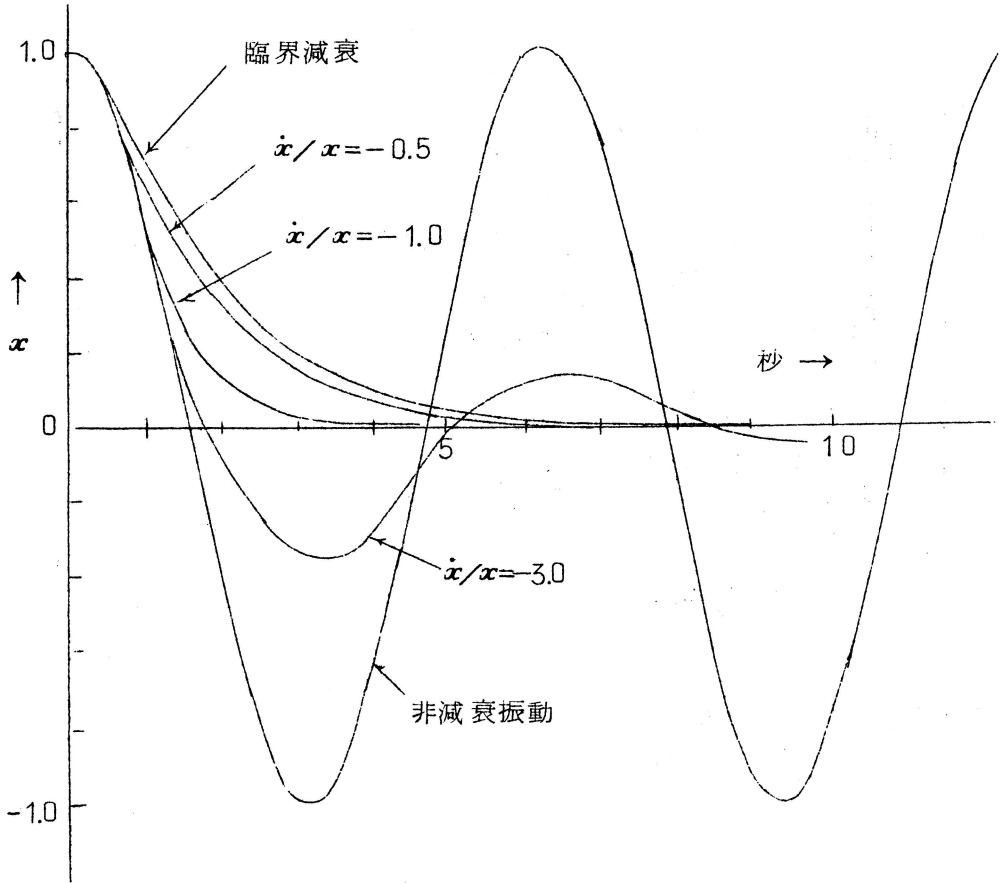
第1表には、Source Program から結果に至るまでの各段階を示す。第6図は Switching action による減衰の有様を位相面上に画いたもので、斜線の領域で $k=1$ 、それ以外で $k=0$ である。比較のため臨界減衰曲線をも画いた。制御結果の良否判定は第7図で行うと分かりやすい。第7図は同じ結果を時間に対して表現したもので、 $k=1$ (off のときは0) になるとき、on-offの境界面が $\dot{x}/x=-1$ (即ち $a=+1$) であれば、極めて速かな減衰が得られることがわかる。



第5図 SWEEP とシミュレータの組合せ



第6図 On-off 制御による
最小時間制御



第7図 On-off 制御による
最小時間制御

Source Program

d09;0'1/00+'d05.0'n0000'
 00;01'.1p00''
 01;'-d'02xd0'3x01-'00+d0'4.1p0''

Target Program

v0083100'8'f3108'12700'80000'1274'80000'
 f2200'8'
 0026f11j'
 v0193200'k3wwj'q2704'52700'q0814'32208'f2204'
 k0824'k3wwj'q2704'32210'f220j'k2600'
 k3wwj'w0808'f0000'7080j'f0000'72704'
 w2700'q0810'32210'f220j'k2604'f3104'
 72708'
 01061gg0'.0003300'

Data

+060803'2000000''3000000'-0000000'
 +063900'1000000''-0000000' (x_0, \dot{x}_0)
 +064100''500000'9900000'1000000'' ($t_0, \Delta t, t_{\max}, k, T_0$)
 1000000'1000000''1000000'-0000000'' (T_1, k, T_0, T_1)

01

Output

time		x		\dot{x}	
.0000000	00	.1000000	01	.0000000	00
.5000000	00	.8776041	00	.4791667-	00
.1000000	01	.5405882	00	.8410373-	00
.1500000	01	.1483937	00	.6892566-	00
.2000000	01	.8924998-	01-	.3477244-	00
.2500000	01	.2449441-	00	.2623987-	00
.3000000	01	.3406967-	00	.1129132-	00
.3500000	01	.3531011-	00	.6415742	01-

第1表 On-off 制御による

最小時間制御

(A5)

V 結 言

以上， SWEEP とブロック・ダイアグラム・シミュレータの主旨を説明したが，常微分方程式，自動制御系などの数値解法は，これらのプログラムに任せることができる。自動プログラム方式を採用しているので，スライド・ルールと同様の手軽さでデジタル計算機を利用することができる。計算対象如何では，解の迅速性，計算精度を考えて，アナログ計算機が適していることもあろうし，A/D，D/A変換器を仲介として両者の長所を生かし合うのが望ましい場合もあろう。これらの状況判断は当事者の良識にまつ外はないが，使用経験が豊かになるにつれ，おのずと中庸を得た判定基準が築かれるものと期待している。

本論文の作製に協力された電気工学科4年 久保田謙治君に謝意を表す。

本 PDF ファイルは 1960 年発行の「第 1 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>