証明は計算できる
- 項書き換えシステムと私 -

外山 芳人

東北大学 情報科学研究科 情報論理学講座

大学に入学するまで

1952年 新潟県長岡市に生まれる。

小学時代: テレビのドキュメンタリー番組でウォルターのカメやシャノンのネズミなどの小型ロボットを知り感激。

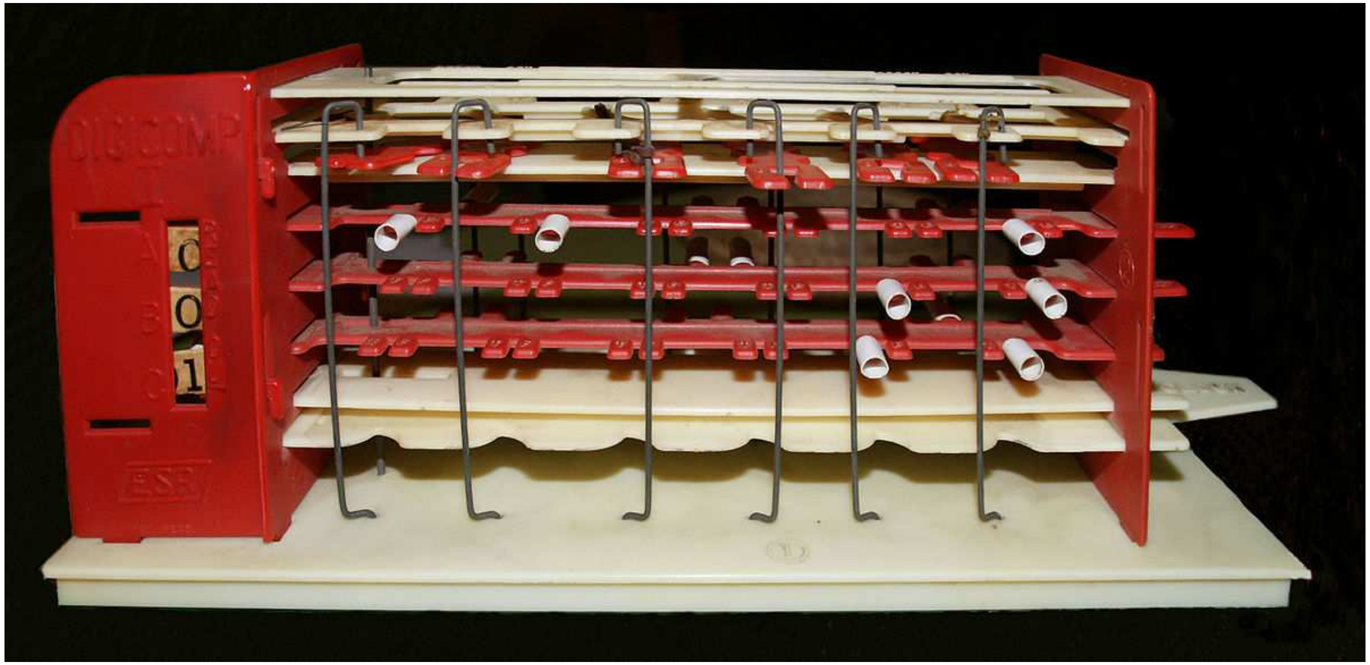
中学時代: サム・ロイドやマーチン・ガードナーの本で数学パズルの面白さを発見。石取りゲームの必勝法を考える。

高校時代: 論理回路の設計をパズルとして楽しむ。
プラスチック製コンピュータで石取りゲームのプログラムを作る。

デジコン (1971)

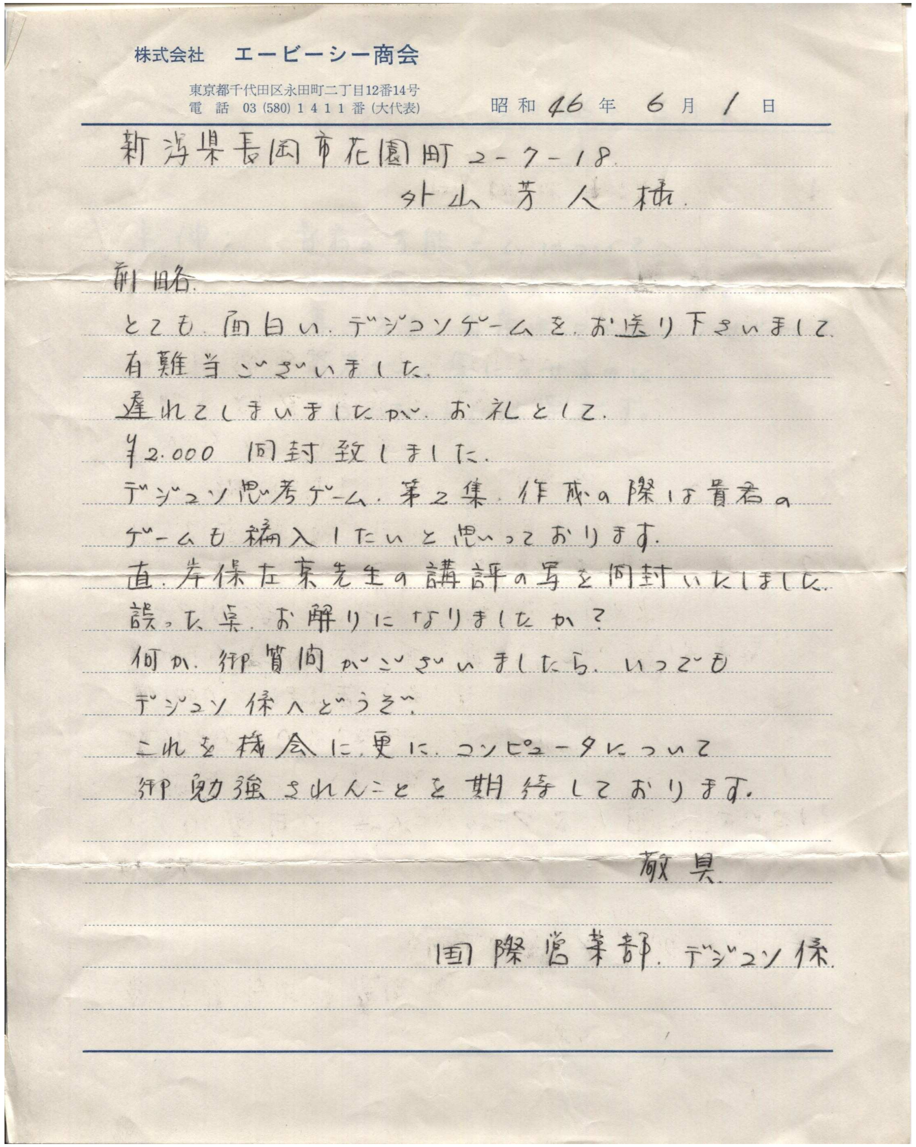
世界最初のオールプラスチック製デジタルコンピュータ。
ABC商会が輸入代理店となり書店経由で販売。

手動で**CLOCK**板を左端に押して右端に戻すと1 サイクル。
白いプラスチックの筒をさしこんでプログラムを組む。
2 進数の計算や簡単なパズルを解くことが可能。



石取りゲームのプログラム

お礼として **2,000** 円同封いたします。**(初めての原稿料)**


株式会社 エービーシー商会
東京都千代田区永田町二丁目12番14号
電話 03 (580) 1411 番 (大代表)
昭和 46 年 6 月 1 日

新潟県長岡市花園町 2-7-18
外山芳人様

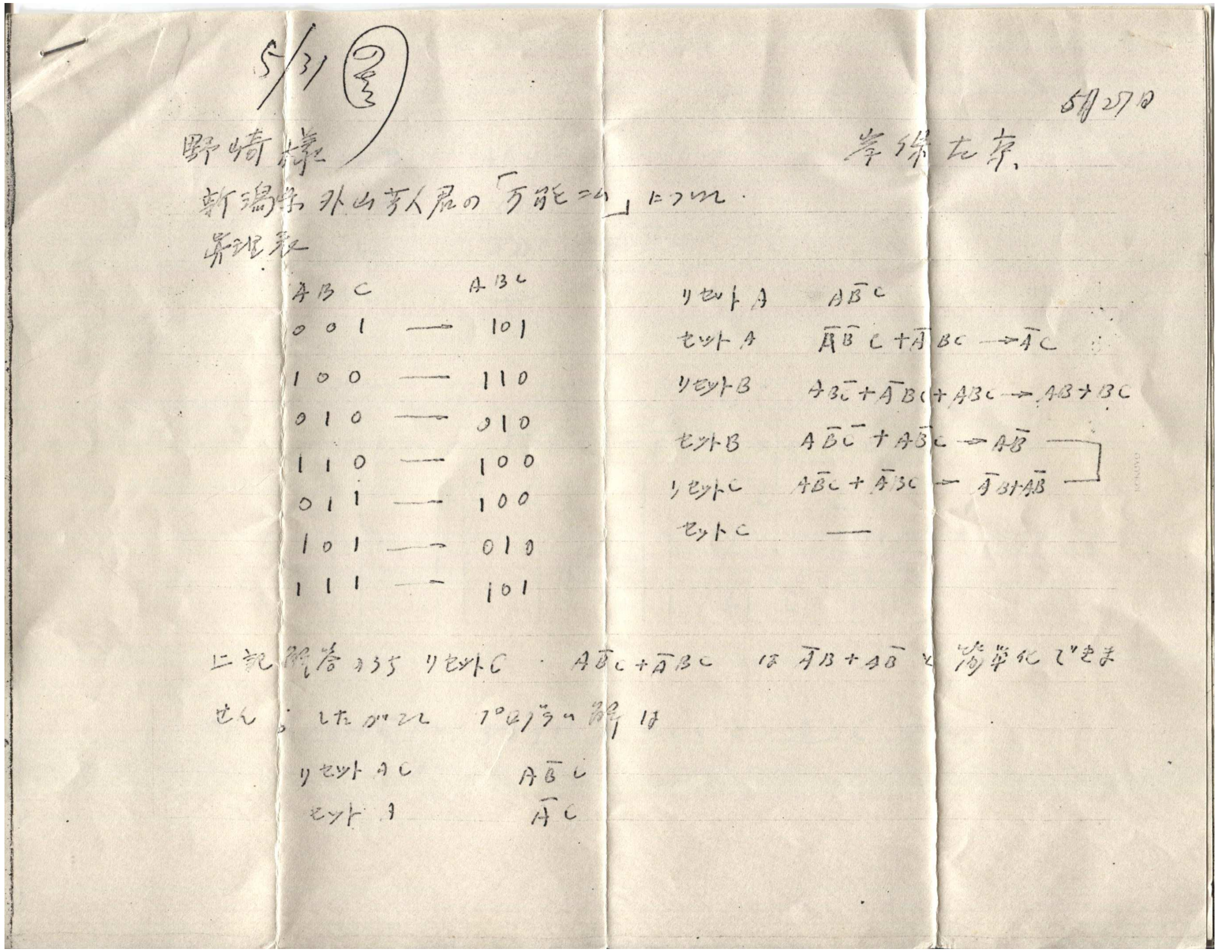
前略
とても面白いデジコンゲームをお送り下さいまして
有難うございました。
遅れてしまいましたが、お礼として
¥2.000 同封致しました。
デジコン思考ゲーム第2集作成の際は貴君の
ゲームも編入したいと思っております。
尚、岸保左京先生の講評の写を同封いたしました。
誤った点、お解りになりましたか？
何か御質問がございましたら、いつでも
デジコン係へどうぞ。
これを機会に更にコンピュータについて
御勉強されんことを期待しております。

敬具

国際営業部 デジコン係

石取りゲームのプログラム

新潟県 外山芳人君の「 万能ニム」について


5/31

5月27日

野崎様

新潟県 外山芳人君の「万能ニム」について

算理表

A B C		A B C
0 0 1	→	1 0 1
1 0 0	→	1 1 0
0 1 0	→	0 1 0
1 1 0	→	1 0 0
0 1 1	→	1 0 0
1 0 1	→	0 1 0
1 1 1	→	1 0 1

リセットA　$A\bar{B}C$

セットA　$\bar{A}\bar{B}C + \bar{A}BC \to \bar{A}C$

リセットB　$AB\bar{C} + \bar{A}BC + ABC \to AB + BC$

セットB　$A\bar{B}\bar{C} + A\bar{B}C \to A\bar{B}$

リセットC　$AB\bar{C} + \bar{A}BC \to \bar{A}B + A\bar{B}$

セットC　—

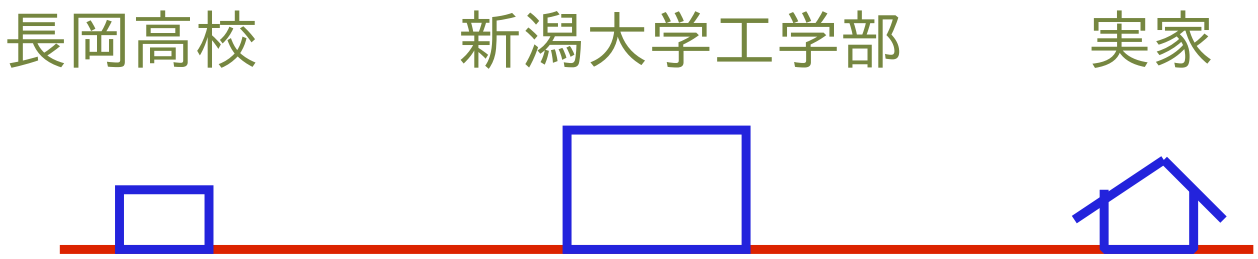
上記解答のうち リセットC・$A\bar{B}C + \bar{A}BC$ は $\bar{A}B + A\bar{B}$ と簡単化できません。したがってプログラム解は

リセットAC　$A\bar{B}C$

セットA　$\bar{A}C$

新潟大学に入学

新潟大学 工学部に入学 **(1971)**。実家から徒歩１０分。



長岡高校との比較すると通学時間**1/2**、学費**1/3**

情報工学はなかったので計算機関係が勉強できそうな電子工学を選ぶ。

計算の理論

オートマトン理論や計算の理論のパズル的面白さにのめり込む。

本多波雄, オートマトン・言語理論 (コロナ社 **1972**)
を読んで演習問題をすべて解く。

それ以外に読んだ本:

ミンスキー, 計算機の数学的理論 (近代科学社 **1970**)

デーヴィス, 計算の理論 (岩波書店 **1966**)

アービブ, オートマトン理論(日本経営出版会 **1971**) など

東北大学の大学院に進学

オートマトン・言語理論の著者の本多波雄先生に弟子入りすることを決心。東北大学の情報工学専攻の修士課程に進学 **(1975)**。

本多先生は名古屋大学へ移られる予定で弟子入りできず。

分野の近い木村正行先生の研究室に入る。

木村研究室

学生をひとりの研究者として尊重する自由な雰囲気。

Aho, Hopcroft, Ullman,
Design and Analysis of Computer Algorithms
(Addison-Wesley 1974)
を輪講し、再帰構造を理解。 **(帰納のことは帰納に聞け)**

バーコフ, マクレーン, 現代代数学概論 (白水社 **1967**) の自主輪講に参加して抽象数学の専門書の読み方を会得。**(一生の財産)**

数学がわかるとは

「 数学基礎論というのは最も厳密な数学であるからていねいにその論証を追ってゆけば明晰判明にわかるだろう、と思って張りきって読みはじめたのであるが、あいまい模糊としていてさっぱりわからなかった。ずい分一生懸命勉強したけれどもどうしてもわかったような気がしなかった。これにはがっかりした。**Kleene**の本も **Schoenfield**の本も大学院の1 年生のための教科書であるから、若いときに読めばわかったはずである。」 **(小平邦彦)**

S. C. Kleene, Introduction to Metamathematics (North Holland 1952)

J. R. Shoenfield, Mathematical Logic (Addison-Wesley 1967)

数学がわかるとは

計器飛行 $\Longleftrightarrow$ 有視界飛行

有視界飛行のコツがつかめるのは若いとき

有視界飛行はひとりひとり異なる

他人に伝えるためには計器飛行

数学がわかるとは

計器飛行 $\Longleftrightarrow$ 有視界飛行

有視界飛行のコツがつかめるのは若いとき

有視界飛行はひとりひとり異なる

他人に伝えるためには計器飛行

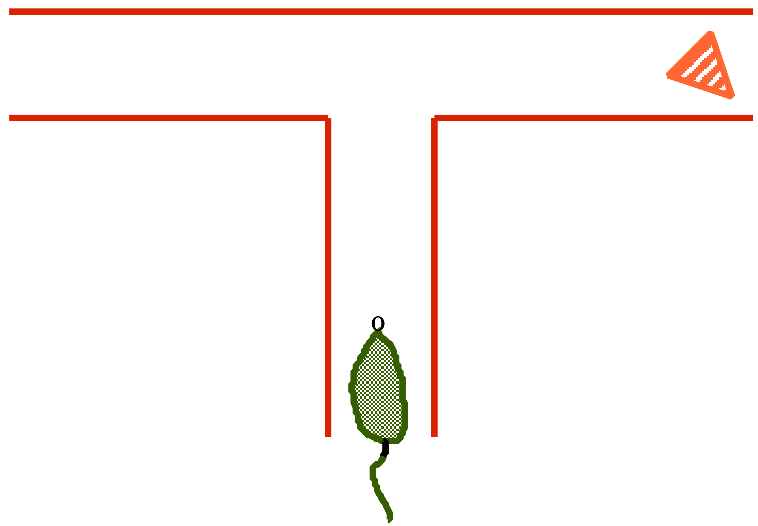
「本書を読むための予備知識はとくに必要ではない。高校程度の数学の素養があれば十分である。」(数学の専門書の「はしがき」より)

修士研究

学生は自分で研究テーマを決める。

研究分野の最前線まで到達する必要がある。**(研究の80%)**

学習オートマトンの振る舞いを理論的に解析することを決心。



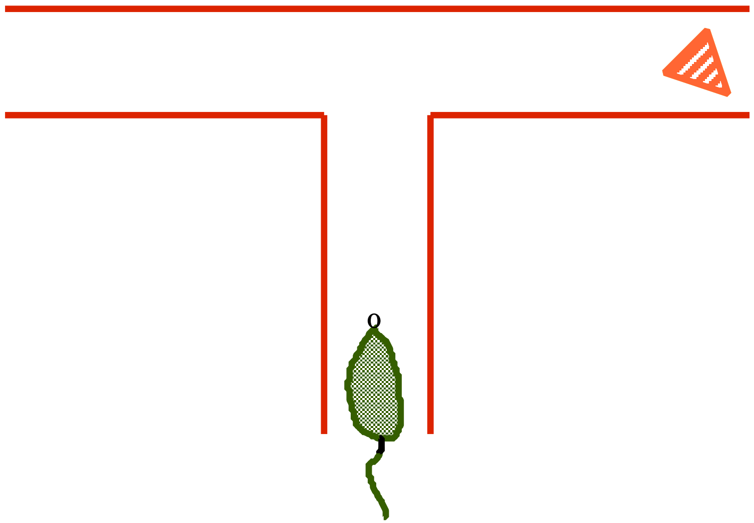
修士研究

学生は自分で研究テーマを決める。

研究分野の最前線まで到達する必要がある。**(研究の80%)**

学習オートマトンの振る舞いを理論的に解析することを決心。

実際はオートマトンというよりも連続状態の吸収マルコフ過程。



修士研究

いくら考えても解析の鍵となる不等式が解けない。

修士研究

いくら考えても解析の鍵となる不等式が解けない。

夏休みの帰省中に散歩をしていて証明が突然ひらめく。

修士研究

いくら考えても解析の鍵となる不等式が解けない。

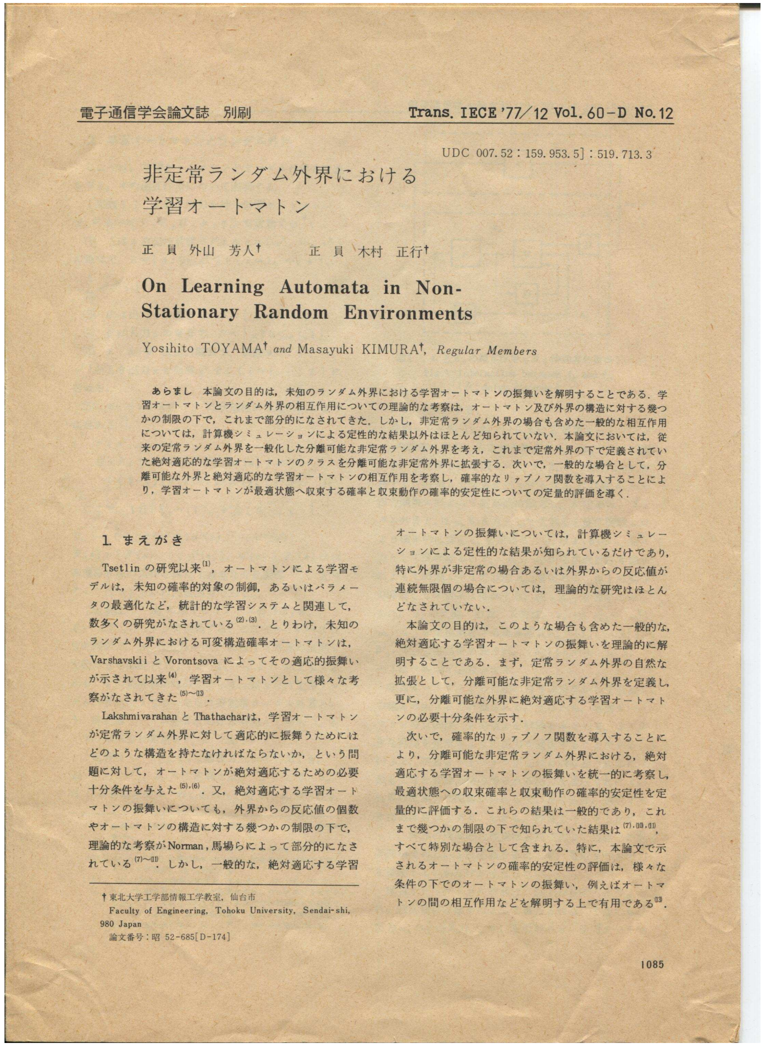
夏休みの帰省中に散歩をしていて証明が突然ひらめく。

答えが存在するか否かわからない問題に挑戦しつづけるスリルと
答えを発見して最前線を突破したときの高揚感を経験。

- 研究の醍醐味を知る -

修士研究

初めての学術論文 (後に IEEE の国際会議に発表)



電子通信学会論文誌 別刷 Trans. IECE '77/12 Vol. 60-D No. 12

UDC 007.52：159.953.5]：519.713.3

非定常ランダム外界における
学習オートマトン

正 員 外山 芳人† 正 員 木村 正行†

On Learning Automata in Non-Stationary Random Environments

Yosihito TOYAMA† *and* Masayuki KIMURA†, *Regular Members*

あらまし 本論文の目的は，未知のランダム外界における学習オートマトンの振舞いを解明することである．学習オートマトンとランダム外界の相互作用についての理論的な考察は，オートマトン及び外界の構造に対する幾つかの制限の下で，これまで部分的になされてきた．しかし，非定常ランダム外界の場合も含めた一般的な相互作用については，計算機シミュレーションによる定性的な結果以外はほとんど知られていない．本論文においては，従来の定常ランダム外界を一般化した分離可能な非定常ランダム外界を考え，これまで定常外界の下で定義されていた絶対適応的な学習オートマトンのクラスを分離可能な非定常外界に拡張する．次いで，一般的な場合として，分離可能な外界と絶対適応的な学習オートマトンの相互作用を考察し，確率的なリァプノフ関数を導入することにより，学習オートマトンが最適状態へ収束する確率と収束動作の確率的安定性についての定量的評価を導く．

1. まえがき

Tsetlin の研究以来[1]，オートマトンによる学習モデルは，未知の確率的対象の制御，あるいはパラメータの最適化など，統計的な学習システムと関連して，数多くの研究がなされている[2],[3]．とりわけ，未知のランダム外界における可変構造確率オートマトンは，Varshavskii と Vorontsova によってその適応的振舞いが示されて以来[4]，学習オートマトンとして様々な考察がなされてきた[5]〜[13]．

Lakshmivarahan と Thathacharは，学習オートマトンが定常ランダム外界に対して適応的に振舞うためにはどのような構造を持たなければならないか，という問題に対して，オートマトンが絶対適応するための必要十分条件を与えた[5],[6]．又，絶対適応する学習オートマトンの振舞いについても，外界からの反応値の個数やオートマトンの構造に対する幾つかの制限の下で，理論的な考察がNorman，馬場らによって部分的になされている[7]〜[11]．しかし，一般的な，絶対適応する学習オートマトンの振舞いについては，計算機シミュレーションによる定性的な結果が知られているだけであり，特に外界が非定常の場合あるいは外界からの反応値が連続無限個の場合については，理論的な研究はほとんどなされていない．

本論文の目的は，このような場合も含めた一般的な，絶対適応する学習オートマトンの振舞いを理論的に解明することである．まず，定常ランダム外界の自然な拡張として，分離可能な非定常ランダム外界を定義し，更に，分離可能な外界に絶対適応する学習オートマトンの必要十分条件を示す．

次いで，確率的なリァプノフ関数を導入することにより，分離可能な非定常ランダム外界における，絶対適応する学習オートマトンの振舞いを統一的に考察し，最適状態への収束確率と収束動作の確率的安定性を定量的に評価する．これらの結果は一般的であり，これまで幾つかの制限の下で知られていた結果は[7],[10],[11]，すべて特別な場合として含まれる．特に，本論文で示されるオートマトンの確率的安定性の評価は，様々な条件の下でのオートマトンの振舞い，例えばオートマトンの間の相互作用などを解明する上で有用である[13]．

† 東北大学工学部情報工学教室，仙台市
Faculty of Engineering, Tohoku University, Sendai-shi, 980 Japan

論文番号：昭 52-685[D-174]

電電公社に就職

電電公社・武蔵野研究所の池野信一先生が東北大学で講演。

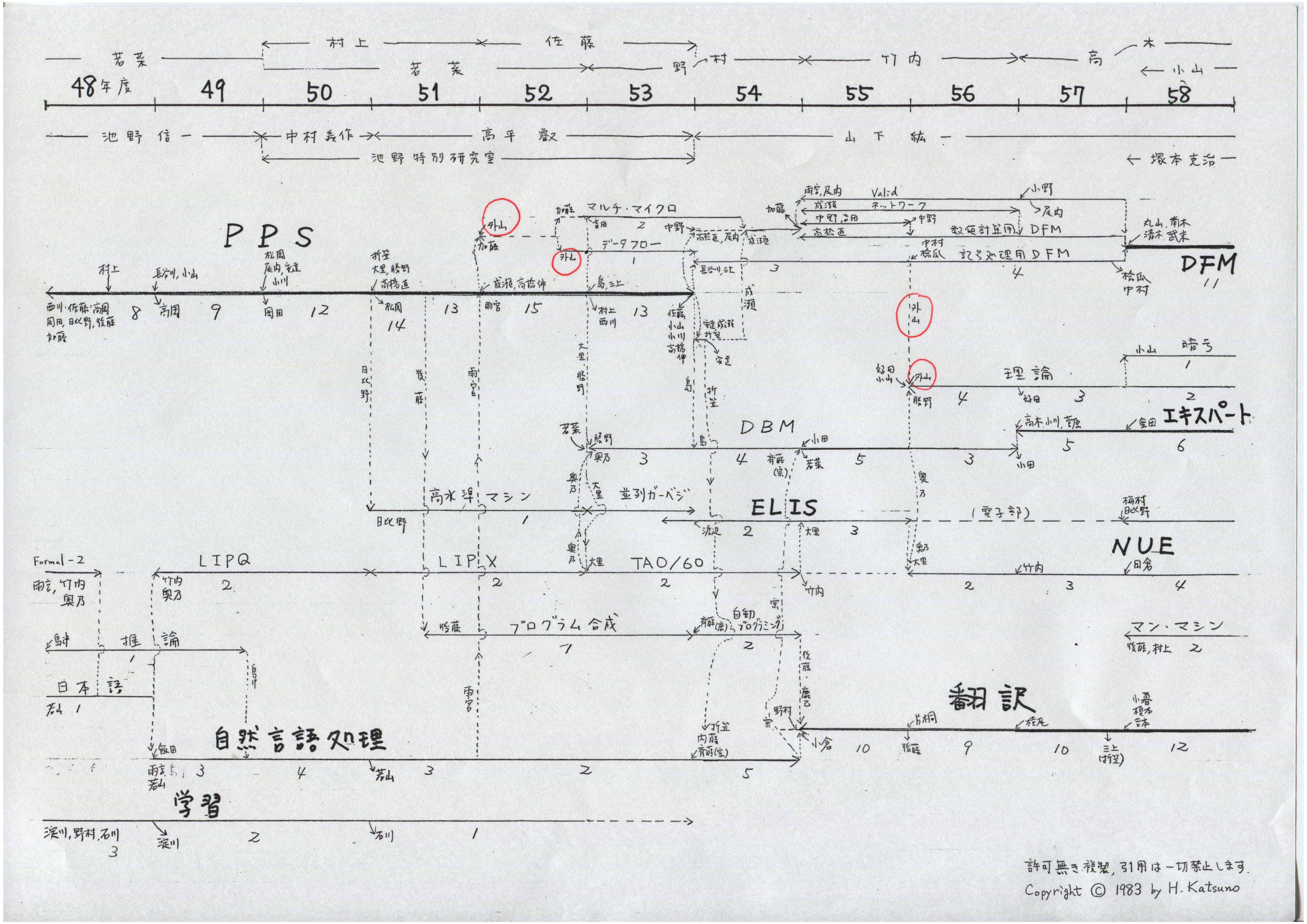
自作のコンピュータについて楽しそうに話す池野先生に弟子入りすることを決心。電電公社・武蔵野研究所に就職。

池野特別研究室に新人は配属しないということで今回も弟子入りできず。

基礎第1 研究室に配属。

並列計算機、データフロー計算機、**LISP**マシン、**LISP**、自動翻訳

研究室の歴史


48年度
49
50
51
52
53
54
55
56
57
58
村上
佐藤
若菜
野村
竹内
高木
小山
池野信一
中村義作
高平徹
山下紘一
池野特別研究室
塚本克治
PPS
マルチ・マイクロ
データフロー
DFM
理論
暗号
エキスパート
DBM
高水準マシン
並列ガーベジ
ELIS
(電子部)
NUE
LIPQ
LIPX
TAO/60
プログラム合成
自動プログラミング
マン・マシン
推論
日本語
自然言語処理
翻訳
学習
許可無き複製，引用は一切禁止します.
Copyright © 1983 by H. Katsuno

研究室の人たち

ソフトウェア：　竹内郁雄さん、奥乃博さん、齊藤康己さん

ハードウェア：　日比野靖さん

基礎理論：　後藤滋樹さん、勝野裕文さん

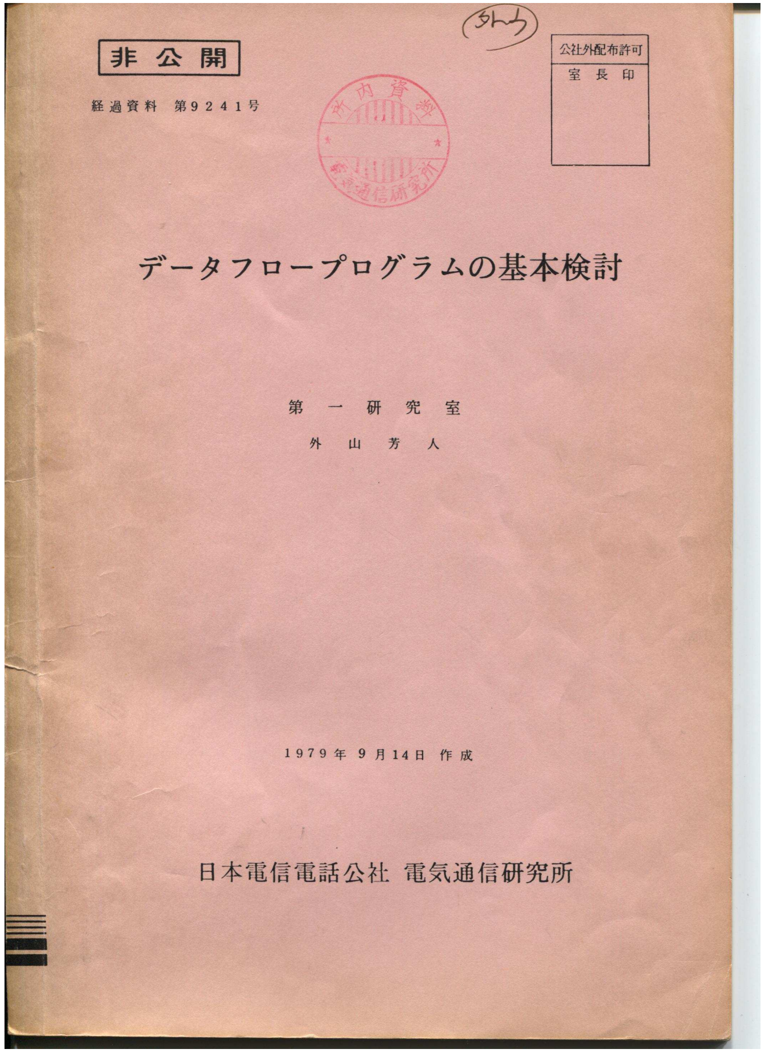
研究室の毎日

数学パズルやゲームを面白がる文化。仕事と遊びがごちゃまぜ。

世界初のコンピュータ同士の将棋対局 (大阪大学 v.s. 玉川大学)
着手は電話連絡。**1979**年**8**月**15**日開始、**10**月**22**日終了。

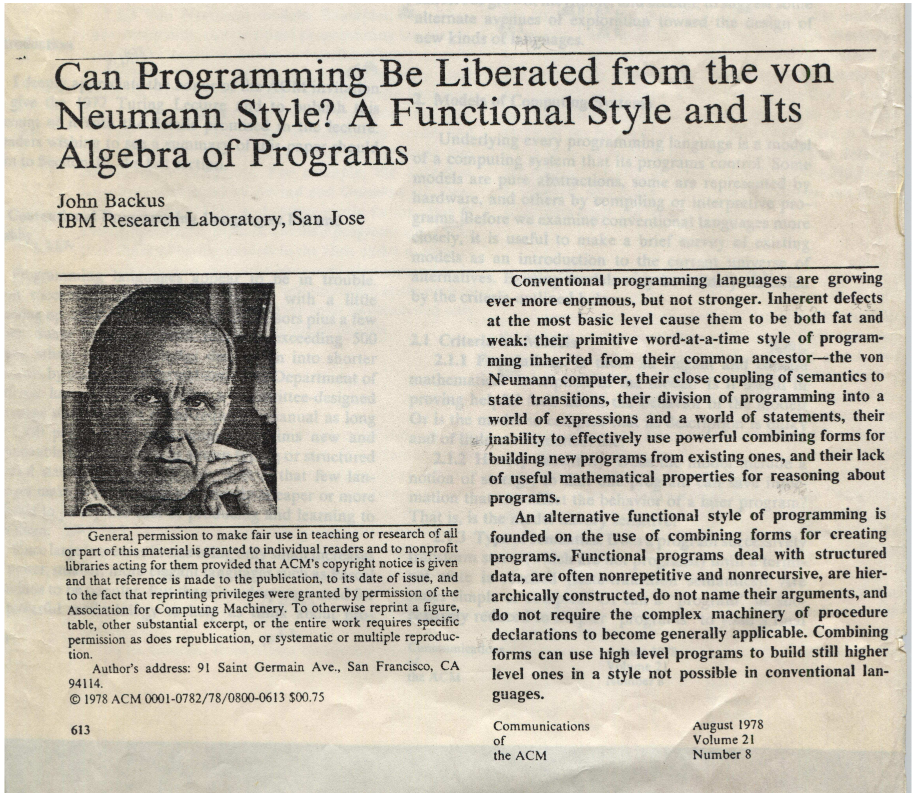
最初の仕事

次のプロジェクト のためのデータフロー計算機の調査



John Backus の記念講演の衝撃

プログラミングはフォン・ノイマン・スタイルから解放されるか?
関数的プログラミング・スタイルとそのプログラム代数
(CACM, 1978)



Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs

John Backus
IBM Research Laboratory, San Jose

Conventional programming languages are growing ever more enormous, but not stronger. Inherent defects at the most basic level cause them to be both fat and weak: their primitive word-at-a-time style of programming inherited from their common ancestor—the von Neumann computer, their close coupling of semantics to state transitions, their division of programming into a world of expressions and a world of statements, their inability to effectively use powerful combining forms for building new programs from existing ones, and their lack of useful mathematical properties for reasoning about programs.

An alternative functional style of programming is founded on the use of combining forms for creating programs. Functional programs deal with structured data, are often nonrepetitive and nonrecursive, are hierarchically constructed, do not name their arguments, and do not require the complex machinery of procedure declarations to become generally applicable. Combining forms can use high level programs to build still higher level ones in a style not possible in conventional languages.

General permission to make fair use in teaching or research of all or part of this material is granted to individual readers and to nonprofit libraries acting for them provided that ACM's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Association for Computing Machinery. To otherwise reprint a figure, table, other substantial excerpt, or the entire work requires specific permission as does republication, or systematic or multiple reproduction.

Author's address: 91 Saint Germain Ave., San Francisco, CA 94114.
© 1978 ACM 0001-0782/78/0800-0613 $00.75

613 Communications of the ACM August 1978 Volume 21 Number 8

フォン・ノイマン型計算モデルの束縛から抜け出せ。

新しい計算モデルの探求

- エレガントで簡潔な数学的記述で表現できる。
- モデルの振る舞いは理論的に解析できる。

論理的基礎の候補： ラムダ計算やコンビネータ論理を勉強。

新しい計算モデルの探求

- エレガントで簡潔な数学的記述で表現できる。
- モデルの振る舞いは理論的に解析できる。

論理的基礎の候補：　ラムダ計算やコンビネータ論理を勉強。

論理と計算を結びつける鍵は合流性！

論理と計算

問題 $\mathbf{1+2=?}$

論理と計算

問題 $\mathbf{1+2=?}$

答 $\mathbf{1+2=3}$

論理と計算

問題 $\mathbf{1+2=?}$

答 $\mathbf{1+2=3}$

答 $\mathbf{1+2=10\times 10-97}$

論理と計算

問題 $\mathbf{1+2=?}$

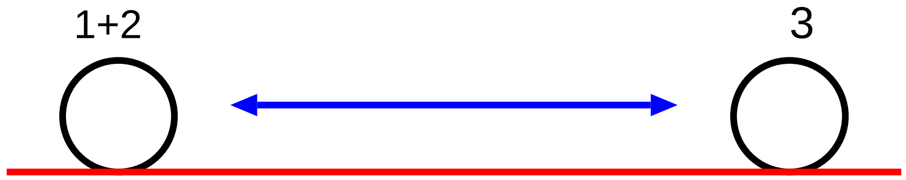
答 $\mathbf{1+2=3}$

答 $\mathbf{1+2=10\times 10-97}$

答 $\mathbf{1+2=1+2}$

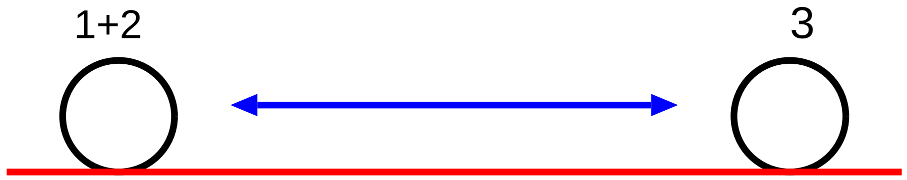
等式の意味

論理 $\mathbf{1 + 2 = 3}$

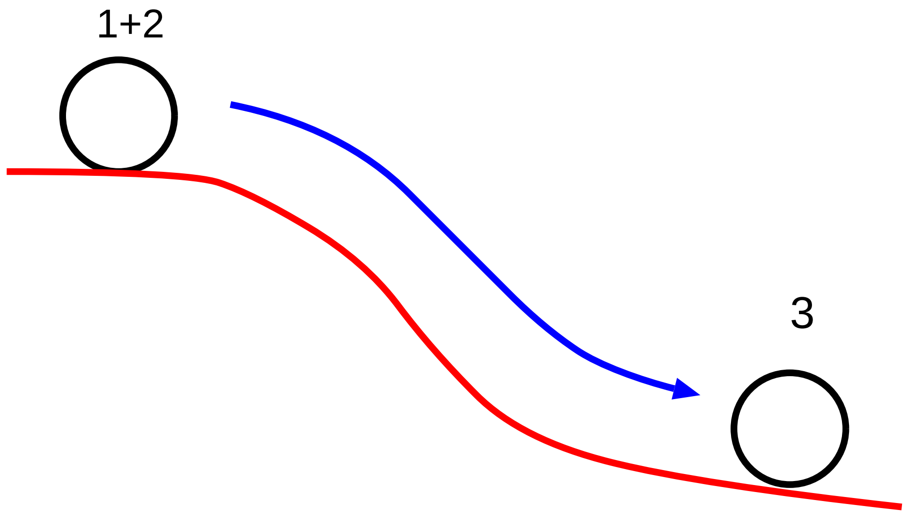


等式の意味

論理 $\mathbf{1 + 2 = 3}$



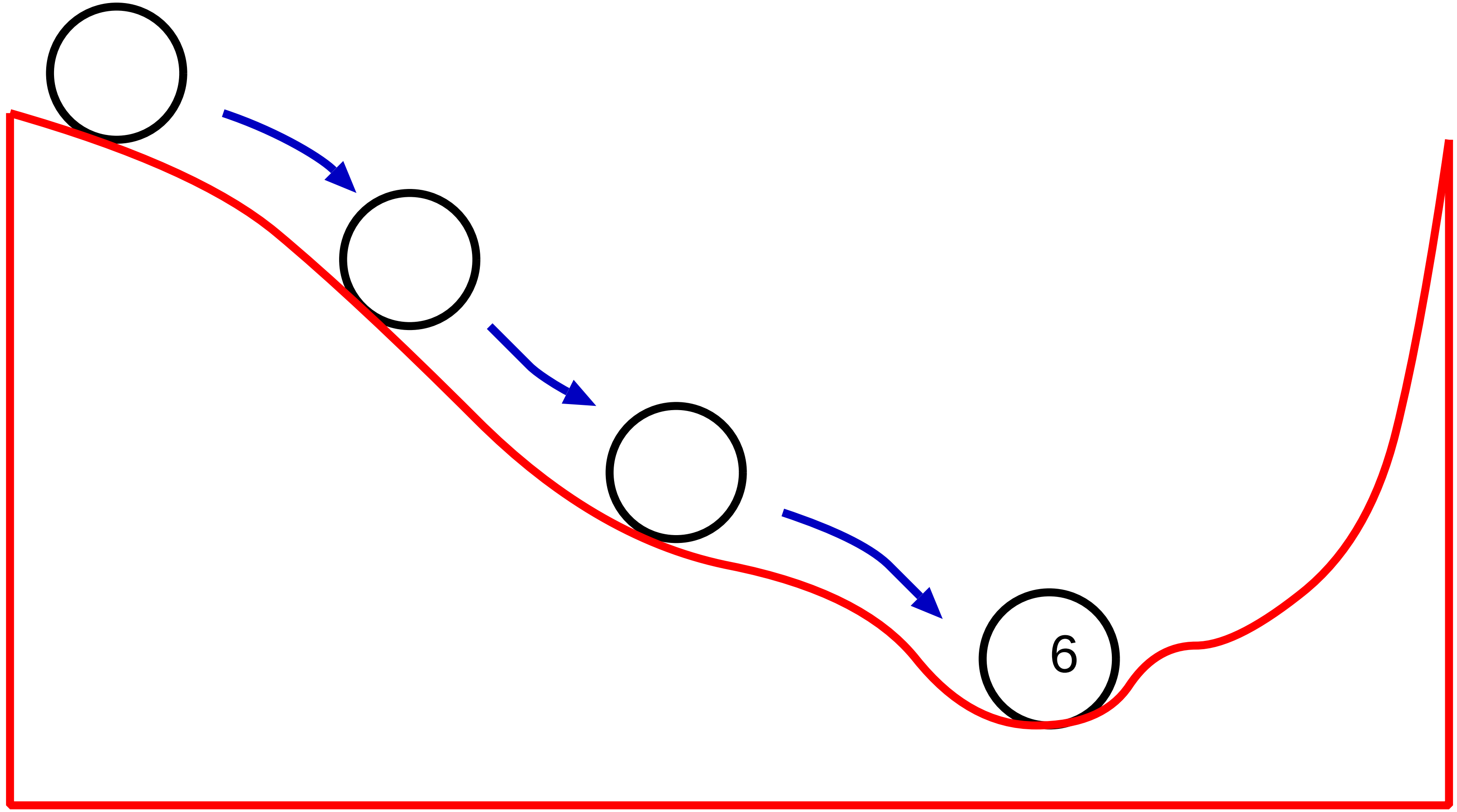
計算 $\mathbf{1 + 2 \to 3}$



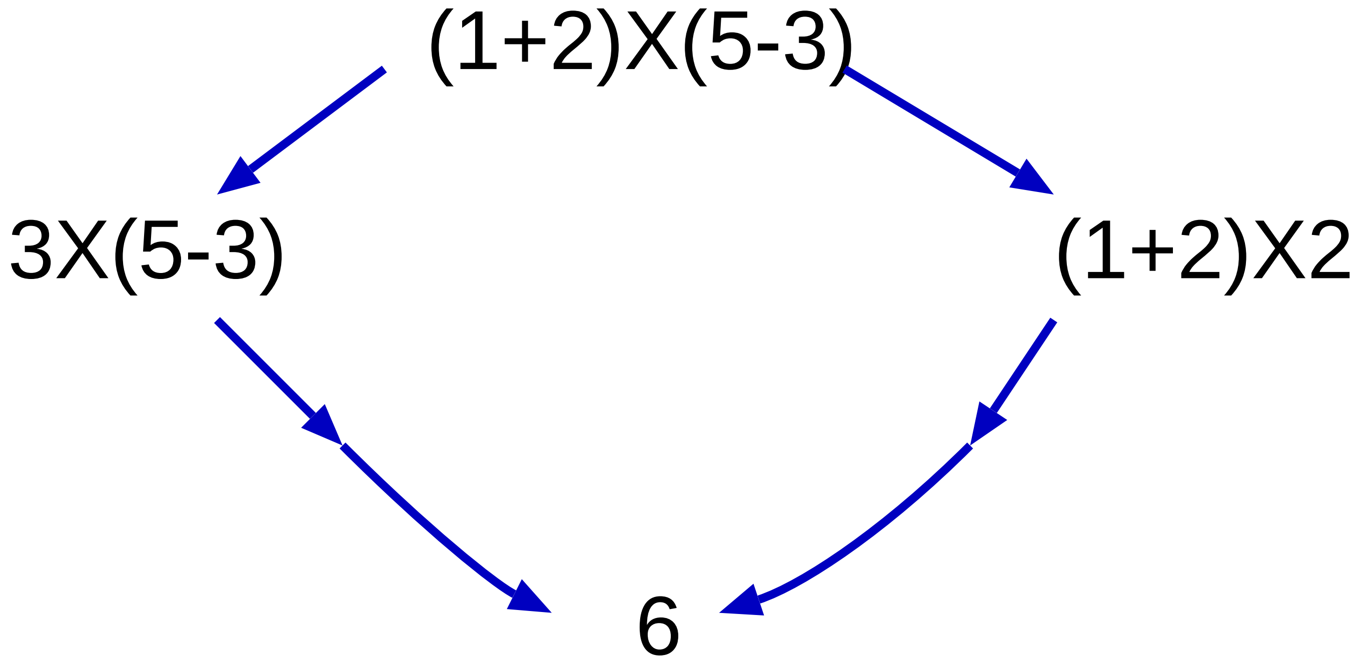
リダクション

「計算とはリダクションのことである」

$$\mathbf{(1+2)\times(5-3)\to 3\times(5-3)\to 3\times 2\to 6}\ \textbf{(正規形)}$$

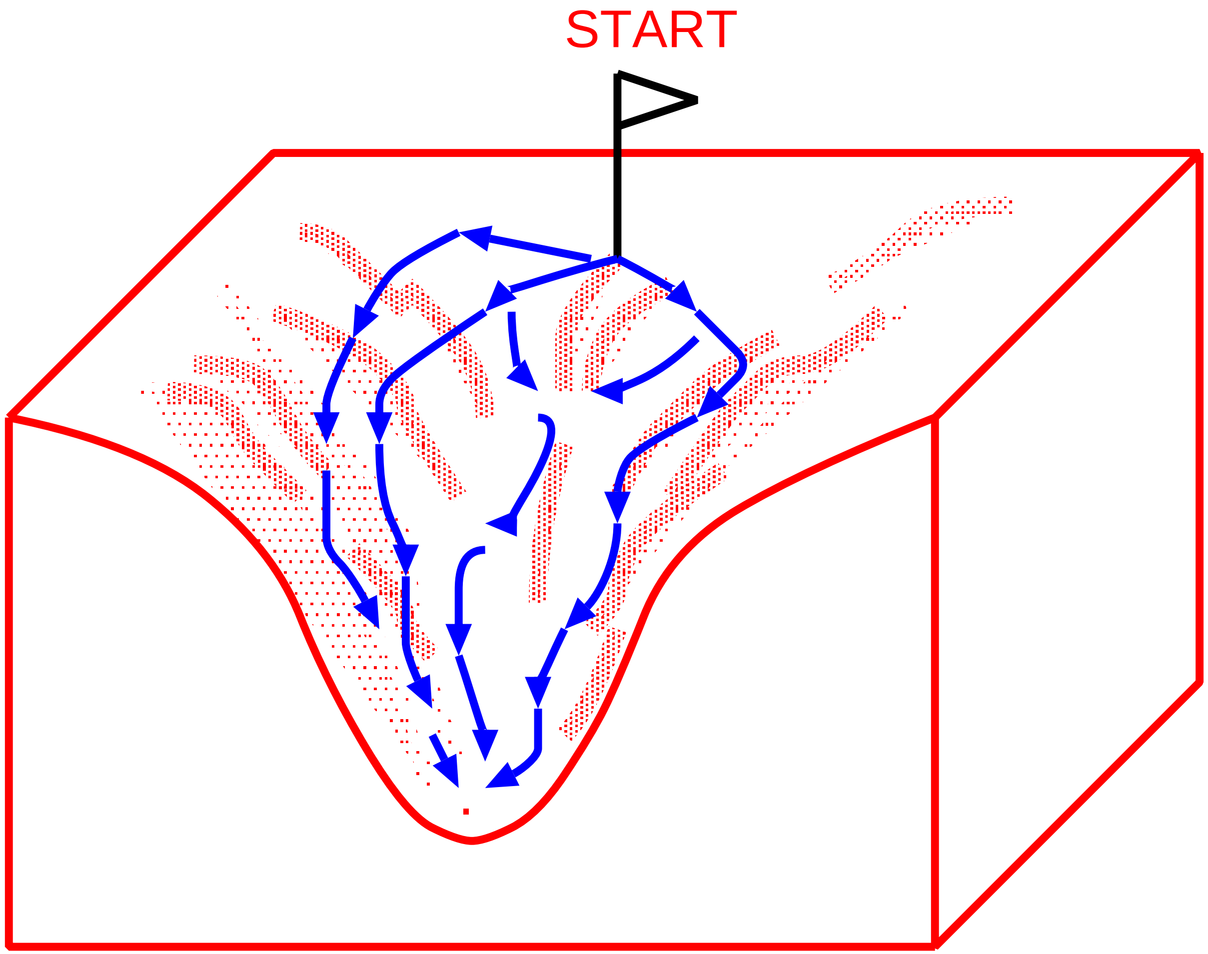


答の一意性



四則演算はどのように計算しても答は一意

合流性



正規形(答)が計算過程に依存しない

項書き換えシステム

証明 $\mathbf{1+2=3}$ をリダクションによる計算 $\mathbf{1+2 \rightarrow 3}$ とみなす計算モデル。

- エレガントで簡潔な数学的記述で表現できる。
- モデルの振る舞いは理論的に解析できる。

項書き換えシステム

証明 $\mathbf{1}+\mathbf{2}=\mathbf{3}$ をリダクションによる計算 $\mathbf{1}+\mathbf{2}\rightarrow\mathbf{3}$ とみなす計算モデル。

- エレガントで簡潔な数学的記述で表現できる。合格!
- モデルの振る舞いは理論的に解析できる。

項書き換えシステム

証明 $\mathbf{1} + \mathbf{2} = \mathbf{3}$ をリダクションによる計算 $\mathbf{1} + \mathbf{2} \to \mathbf{3}$ とみなす計算モデル。

- エレガントで簡潔な数学的記述で表現できる。合格!
- モデルの振る舞いは理論的に解析できる。 不明?

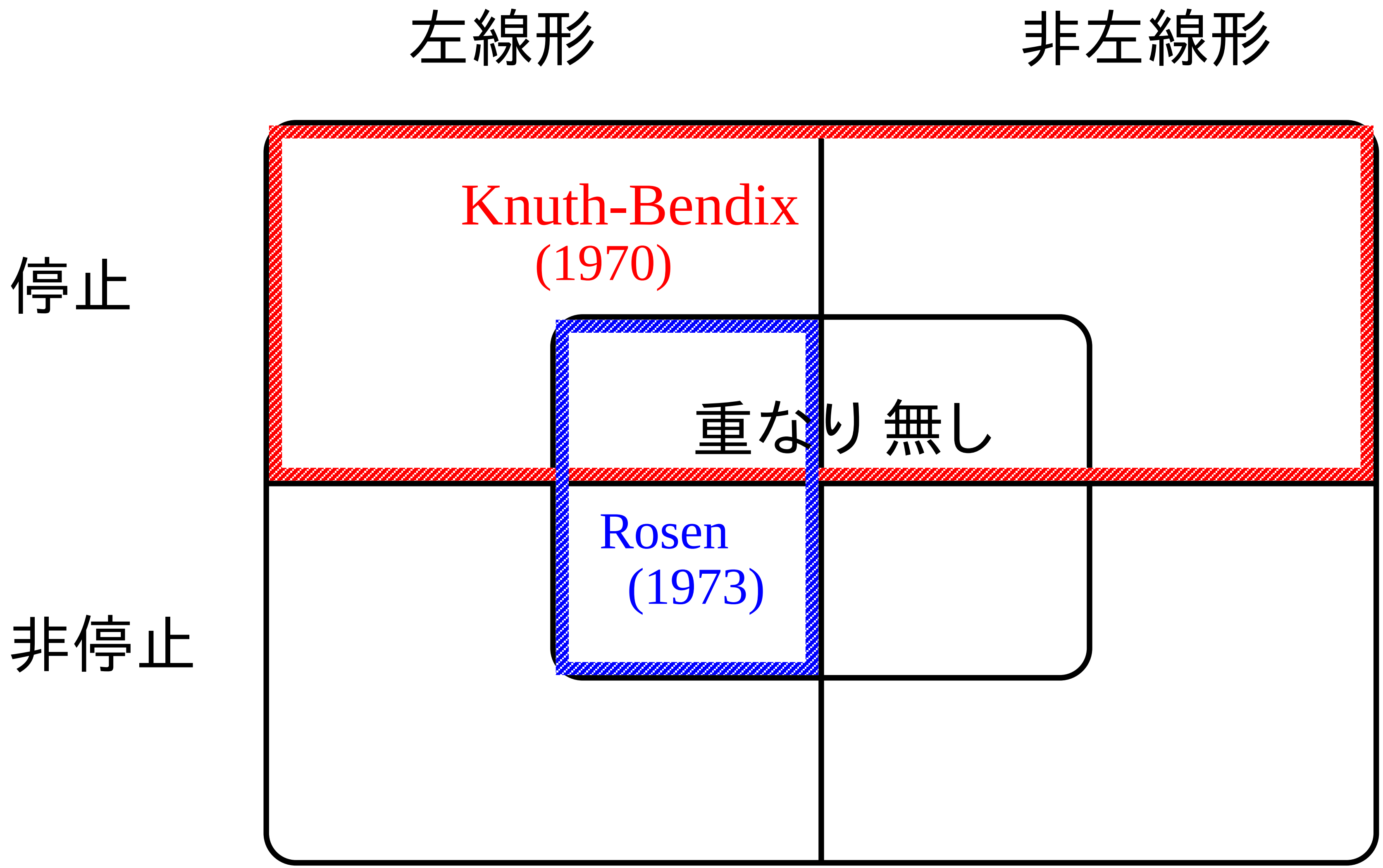
項書き換えシステム

証明 $\mathbf{1}+\mathbf{2}=\mathbf{3}$ をリダクションによる計算 $\mathbf{1}+\mathbf{2}\rightarrow\mathbf{3}$ とみなす計算モデル。

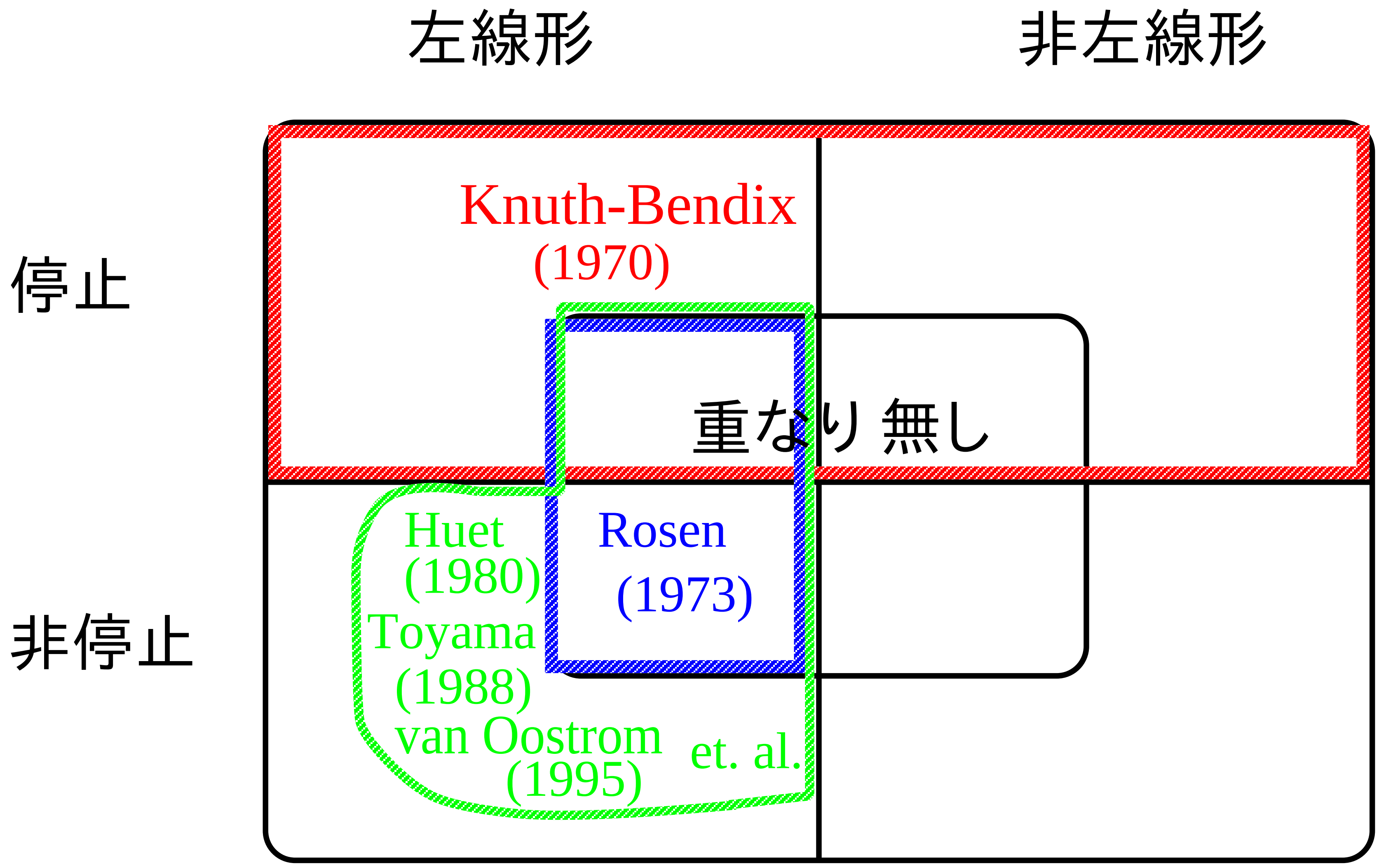
- エレガントで簡潔な数学的記述で表現できる。合格!
- モデルの振る舞いは理論的に解析できる。 不明?

合流性を理論的に解析できるか?

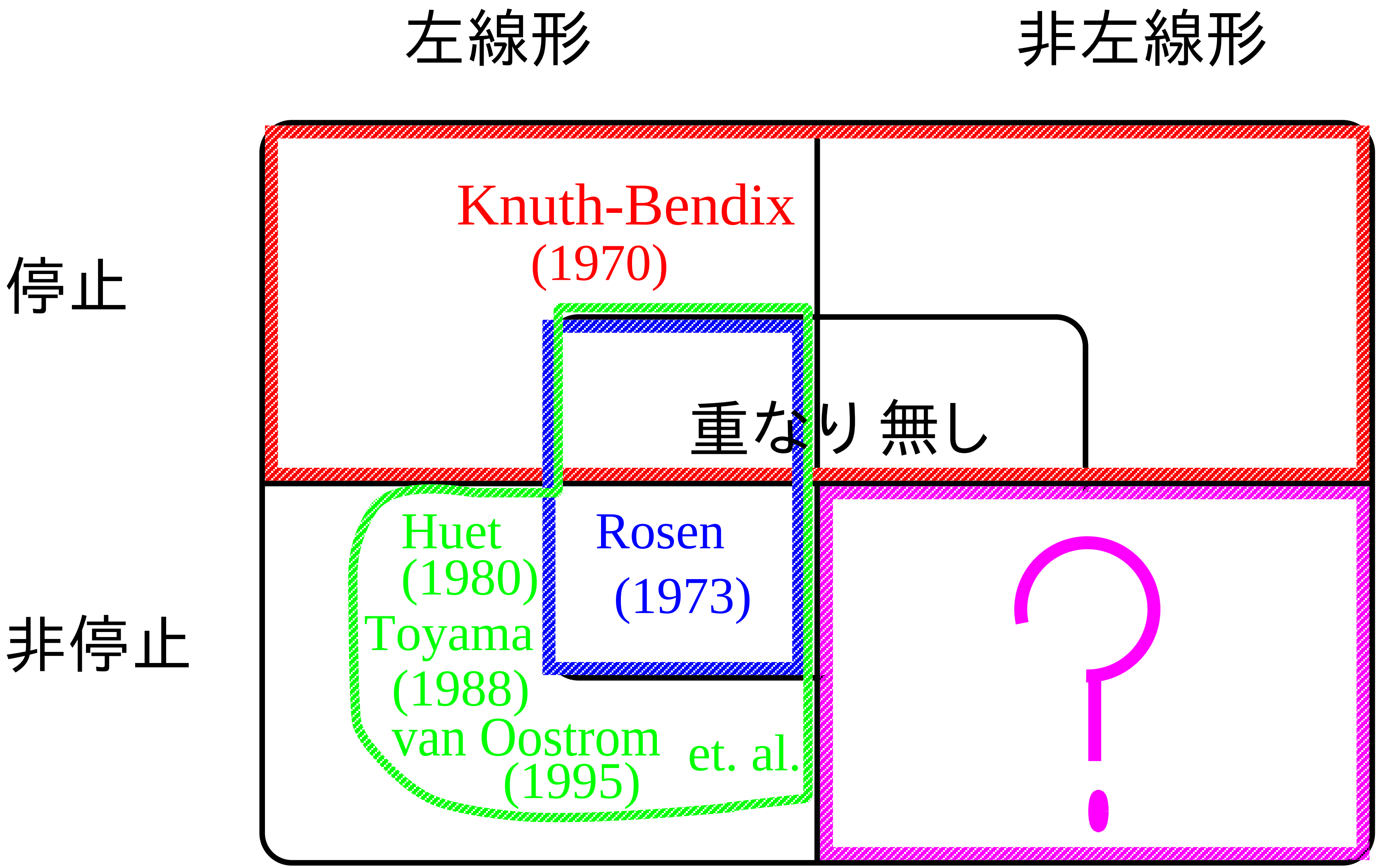
合流性の判定法



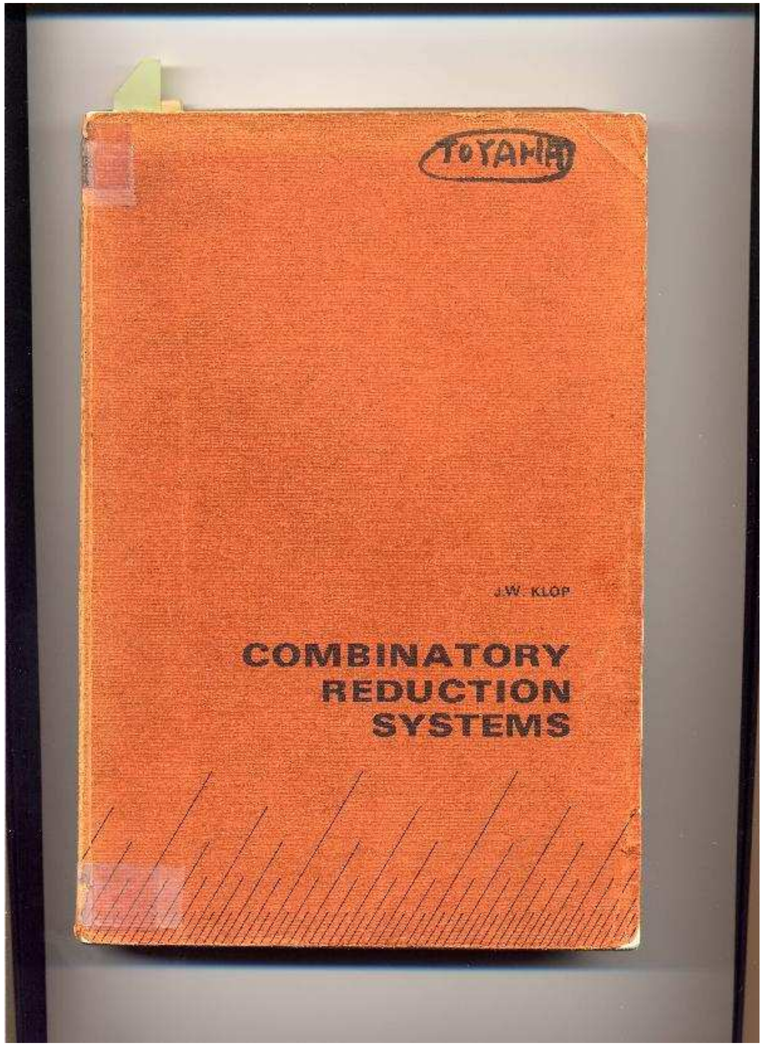
合流性の判定法


左線形
非左線形
停止
非停止
Knuth-Bendix
(1970)
重なり無し
Huet
(1980)
Rosen
(1973)
Toyama
(1988)
van Oostrom
(1995)
et. al.

合流性の判定法


左線形
非左線形
停止
非停止
Knuth-Bendix
(1970)
重なり無し
Huet
(1980)
Rosen
(1973)
Toyama
(1988)
van Oostrom
(1995)
et. al.
?

Jan Willem Klop (1980) の発見



Jan Willem Klop (1980) の発見

$CL + \{Dxx \to E\}$ は合流しない。
未解決問題を解決!

$CL + \{D(x,x) \to E\}$ は合流する。
合流性をもつ非左線形・非停止システムの発見!

Jan Willem Klop (1980) の発見

$CL + \{Dxx \to E\}$ は合流しない。

未解決問題を解決!

$CL + \{D(x,x) \to E\}$ は合流する。

合流性をもつ非左線形・非停止システムの発見!

問:

両者の違いは何か?

Jan Willem Klop (1980) の発見

$CL + \{Dxx \to E\}$ は合流しない。

未解決問題を解決!

$CL + \{D(x, x) \to E\}$ は合流する。

合流性をもつ非左線形・非停止システムの発見!

問:

両者の違いは何か?

答:

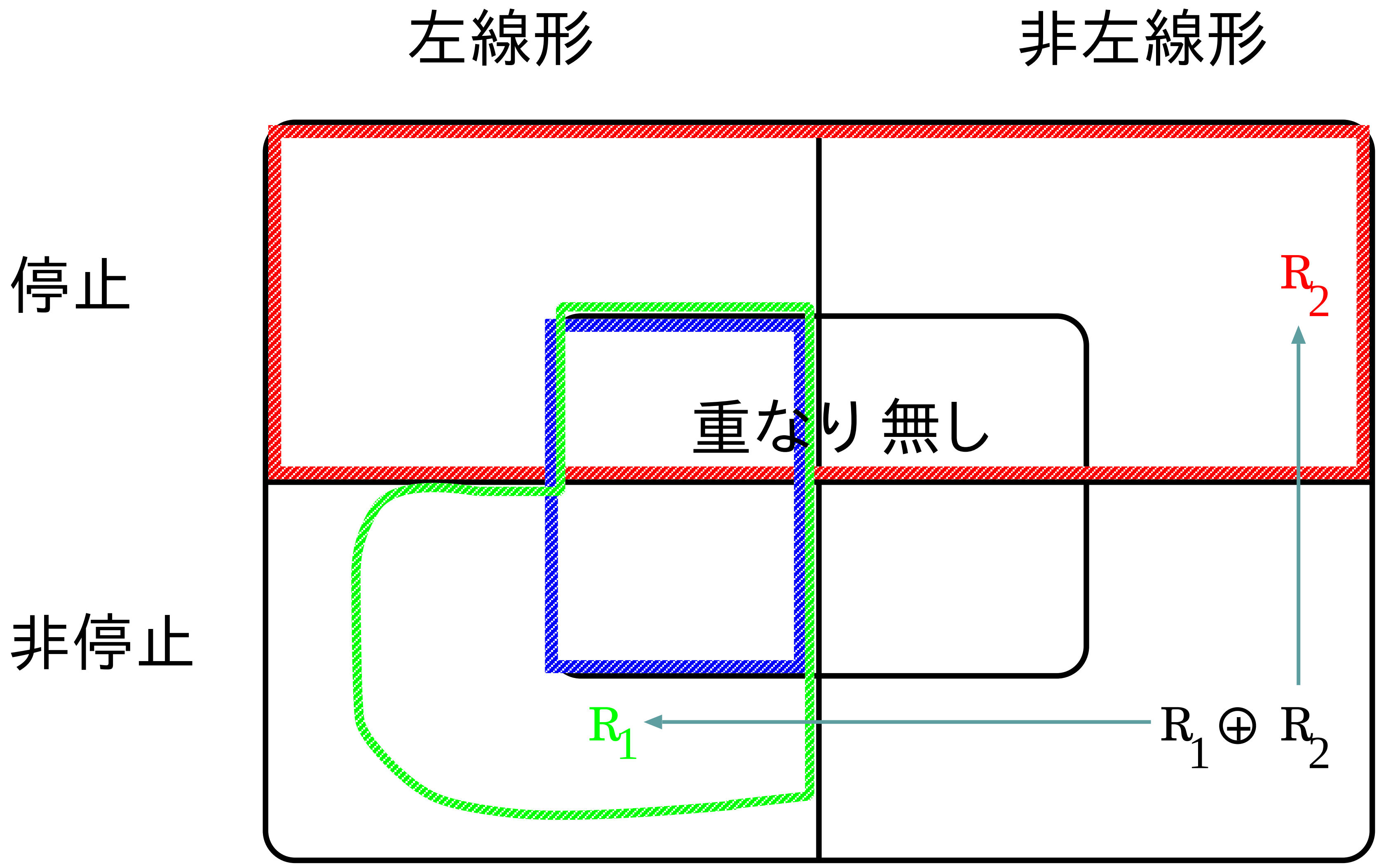
モジュラ性 **(Toyama 1987)**

合流性のモジュラ性

$$\boldsymbol{R_1} \text{ と } \boldsymbol{R_2} \text{ は合流} \iff \boldsymbol{R_1} \oplus \boldsymbol{R_2} \text{ は合流}$$

外山の定理 **(1987)**

モジュラ性に基づく合流性判定



停止性のモジュラ性

論文誌に投稿すると、レフリーから以下の質問を受ける。

“Can the author prove by his analysis of of the layer structure of $\mathcal{R}_1 \oplus \mathcal{R}_2$ - terms also the following:

$$R_1 \text{ and } R_2 \text{ are terminating} \iff R_1 \oplus R_2 \text{ is terminating?}$$

Maybe this fact, which would also be whorthwhile to have, can be obatained with relatively little extra effort.”

停止性のモジュラ性

論文誌に投稿すると、レフリーから以下の質問を受ける。

“Can the author prove by his analysis of of the layer structure of $\mathcal{R}_1 \oplus \mathcal{R}_2$ - terms also the following:

$$R_1 \text{ and } R_2 \text{ are terminating} \iff R_1 \oplus R_2 \text{ is terminating?}$$

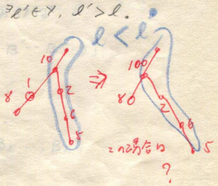
Maybe this fact, which would also be whorthwhile to have, can be obatained with relatively little extra effort.”

私の答えは完全に**YES**だった。 なぜなら ...

停止性のモジュラ性

すでに以下の定理も証明していたから：

$$\boldsymbol{R}_1 \text{ と } \boldsymbol{R}_2 \text{ は停止} \iff \boldsymbol{R}_1 \oplus \boldsymbol{R}_2 \text{ は停止.}$$



4

1.13 命題

$(\mathcal{M}(S), \gg)$ が well founded ordering

$\Leftrightarrow (S, >)$ が well founded ordering

1.14 補題

$R_1 \models SN$, $R_2 \models SN$ とする。ここで $\mathcal{T}(F_1 \cup F_2)$ 上の直和システム $R_1 \oplus R_2$ を考えると、$M \to N$ のとき、$D_M \gg D_N$。

証明

$P \in Part(M)$, $P(Q_1, \cdots, Q_m) \subseteq M$, $A_l \rhd A_r \in R_a$, A_l は P の中に出現しているものとする。$M \equiv C[P(Q_1, \cdots, Q_m)]$

$N \equiv C[R]$ とすると、$R \equiv P'(Q_{i_1}, \cdots, Q_{i_m})$, $Q_{i_j} \in \{Q_1, \cdots, Q_m\}$ と $R \equiv Q_i$ の2つの場合が考えられる。

i) $R \equiv P'(Q_{i_1}, \cdots, Q_{i_m})$ の場合 （$m=0$ の場合も含む。）

$X = \{l \mid d_p \in l \in W_M\}$,

$Y = \{l \mid d_{p'} \in l \in W_N\}$

とすると、$W_N = (W_M - X) \cup Y$. $d_p > d_{p'}$, $Q_{i_j} \in \{Q_1, \cdots, Q_m\}$ より $\forall l \in Y, \exists l' \in X, l' > l$ は明らか。

ii) $R \equiv Q_i$ の場合

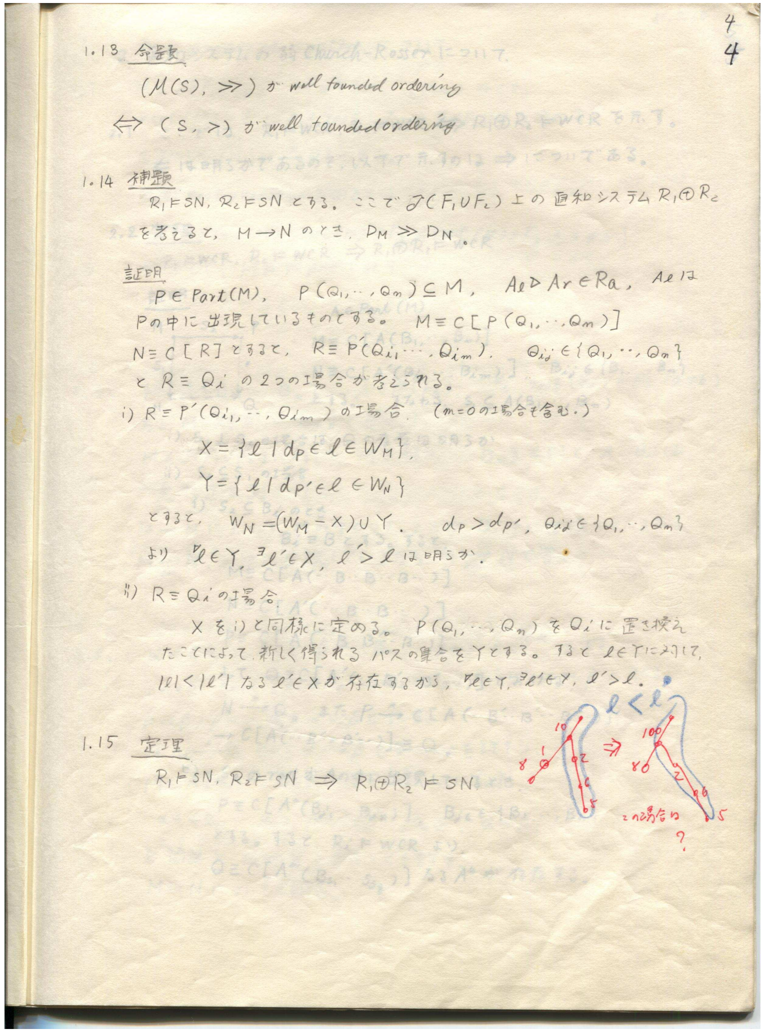
X を i) と同様に定める。$P(Q_1, \cdots, Q_n)$ を Q_i に置き換えたことによって新しく得られるパスの集合を Y とする。すると $l \in Y$ に対して、$|l| < |l'|$ なる $l' \in X$ が存在するから、$\forall l \in Y, \exists l' \in X, l' > l$.

1.15 定理

$R_1 \models SN$, $R_2 \models SN \Rightarrow R_1 \oplus R_2 \models SN$.

停止性のモジュラ性

証明のスケッチから完全な証明を完成させようとするが ...



4

1.13 命題

$(\mathcal{M}(S), \gg)$ が well founded ordering

$\Leftrightarrow$ $(S, >)$ が well founded ordering

1.14 補題

$R_1 \models SN$, $R_2 \models SN$ とする。ここで $\mathcal{T}(F_1 \cup F_2)$ 上の直和システム $R_1 \oplus R_2$ を考えると、$M \to N$ のとき、$D_M \gg D_N$。

証明

$P \in Part(M)$, $P(Q_1, \cdots, Q_m) \subseteq M$, $A_l \rhd A_r \in R_a$, A_l は P の中に出現しているものとする。$M \equiv C[P(Q_1, \cdots, Q_m)]$ $N \equiv C[R]$ とすると、$R \equiv P'(Q_{i_1}, \cdots, Q_{i_m})$, $Q_{i_j} \in \{Q_1, \cdots, Q_m\}$ と $R \equiv Q_i$ の2つの場合が考えられる。

i) $R \equiv P'(Q_{i_1}, \cdots, Q_{i_m})$ の場合 ($m=0$ の場合も含む。)

$X = \{l \mid d_p \in l \in W_M\}$,

$Y = \{l \mid d_{p'} \in l \in W_N\}$

とすると、$W_N = (W_M - X) \cup Y$. $d_p > d_{p'}$, $Q_{i_j} \in \{Q_1, \cdots, Q_m\}$ より $\forall l \in Y, \exists l' \in X, l' > l$ は明らか。

ii) $R \equiv Q_i$ の場合

X を i) と同様に定める。$P(Q_1, \cdots, Q_n)$ を Q_i に置き換えたことによって新しく得られるパスの集合を Y とする。すると $l \in Y$ に対して、$|l| < |l'|$ なる $l' \in X$ が存在するから、$\forall l \in Y, \exists l' \in X, l' > l$。

1.15 定理

$R_1 \models SN$, $R_2 \models SN \Rightarrow R_1 \oplus R_2 \models SN$

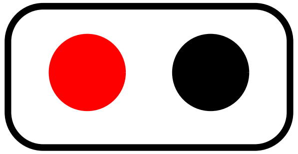
停止性のモジュラ性

ひとつ仮定を証明すると、証明すべき次の仮定が生まれ、いつまでたっても終らない。

停止性のモジュラ性

ひとつ仮定を証明すると、証明すべき次の仮定が生まれ、いつまでたっても終らない。

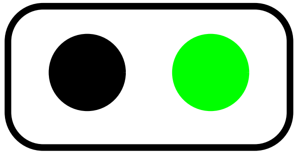
ある朝、横断歩道で信号が青に変わるのを待っていた。



停止性のモジュラ性

ひとつ仮定を証明すると、証明すべき次の仮定が生まれ、いつまでたっても終らない。

ある朝、横断歩道で信号が青に変わるのを待っていた。

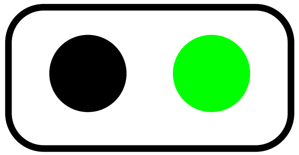


横断歩道を渡り始めたとき，

停止性のモジュラ性

ひとつ仮定を証明すると、証明すべき次の仮定が生まれ、いつまでたっても終らない。

ある朝、横断歩道で信号が青に変わるのを待っていた。



横断歩道を渡り始めたとき,

心にひとつの例が自然に浮かんだ。

外山の反例 (1987)

$$R_1 \left\{ f(0,1,x) \to f(x,x,x) \right.$$

$$R_2 \left\{ \begin{array}{l} g(x,y) \to x \\ g(x,y) \to y \end{array} \right.$$

R_1 は R_2 停止するが $R_1 \oplus R_2$ は停止しない:

$$f(g(0,1), g(0,1), g(0,1)) \to$$

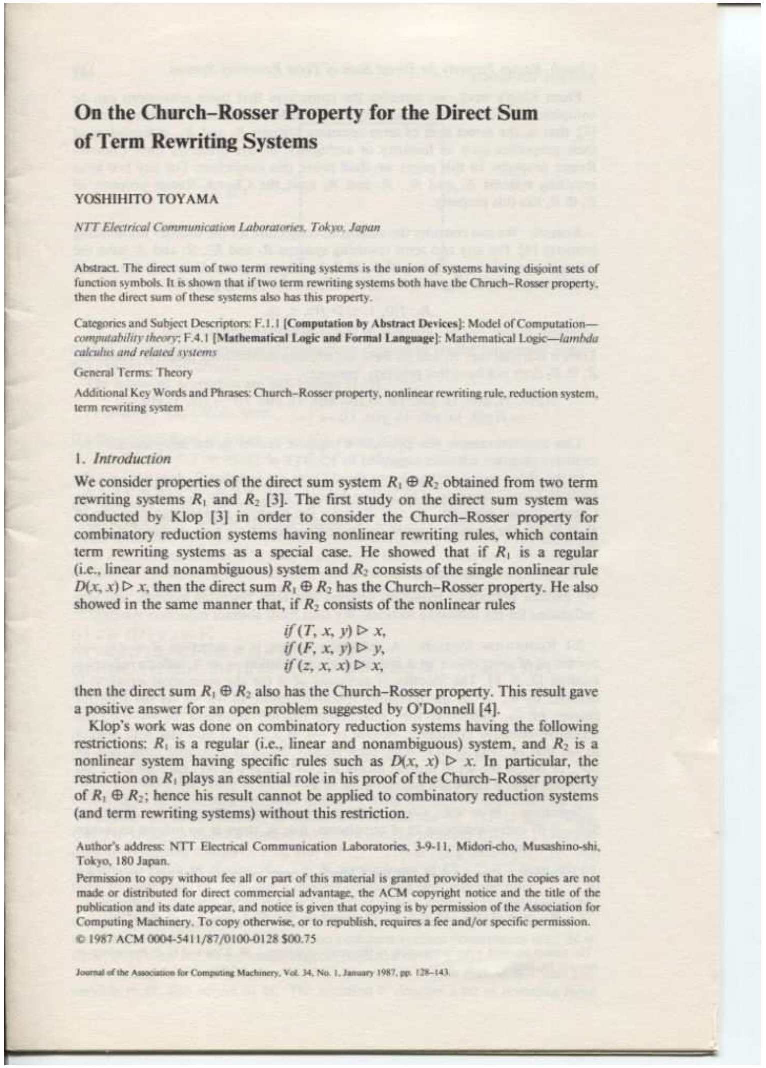
$$f(0, g(0,1), g(0,1)) \to$$

$$f(0, 1, g(0,1)) \to$$

$$f(g(0,1), g(0,1), g(0,1)) \to \cdots$$

停止性はモジュラ性をもたない!

外山の定理と反例



On the Church–Rosser Property for the Direct Sum of Term Rewriting Systems

YOSHIHITO TOYAMA

NTT Electrical Communication Laboratories, Tokyo, Japan

Abstract. The direct sum of two term rewriting systems is the union of systems having disjoint sets of function symbols. It is shown that if two term rewriting systems both have the Chruch–Rosser property, then the direct sum of these systems also has this property.

Categories and Subject Descriptors: F.1.1 **[Computation by Abstract Devices]**: Model of Computation—*computability theory*; F.4.1 **[Mathematical Logic and Formal Language]**: Mathematical Logic—*lambda calculus and related systems*

General Terms: Theory

Additional Key Words and Phrases: Church–Rosser property, nonlinear rewriting rule, reduction system, term rewriting system

1. *Introduction*

We consider properties of the direct sum system $R_1 \oplus R_2$ obtained from two term rewriting systems R_1 and R_2 [3]. The first study on the direct sum system was conducted by Klop [3] in order to consider the Church–Rosser property for combinatory reduction systems having nonlinear rewriting rules, which contain term rewriting systems as a special case. He showed that if R_1 is a regular (i.e., linear and nonambiguous) system and R_2 consists of the single nonlinear rule $D(x, x) \rhd x$, then the direct sum $R_1 \oplus R_2$ has the Church–Rosser property. He also showed in the same manner that, if R_2 consists of the nonlinear rules

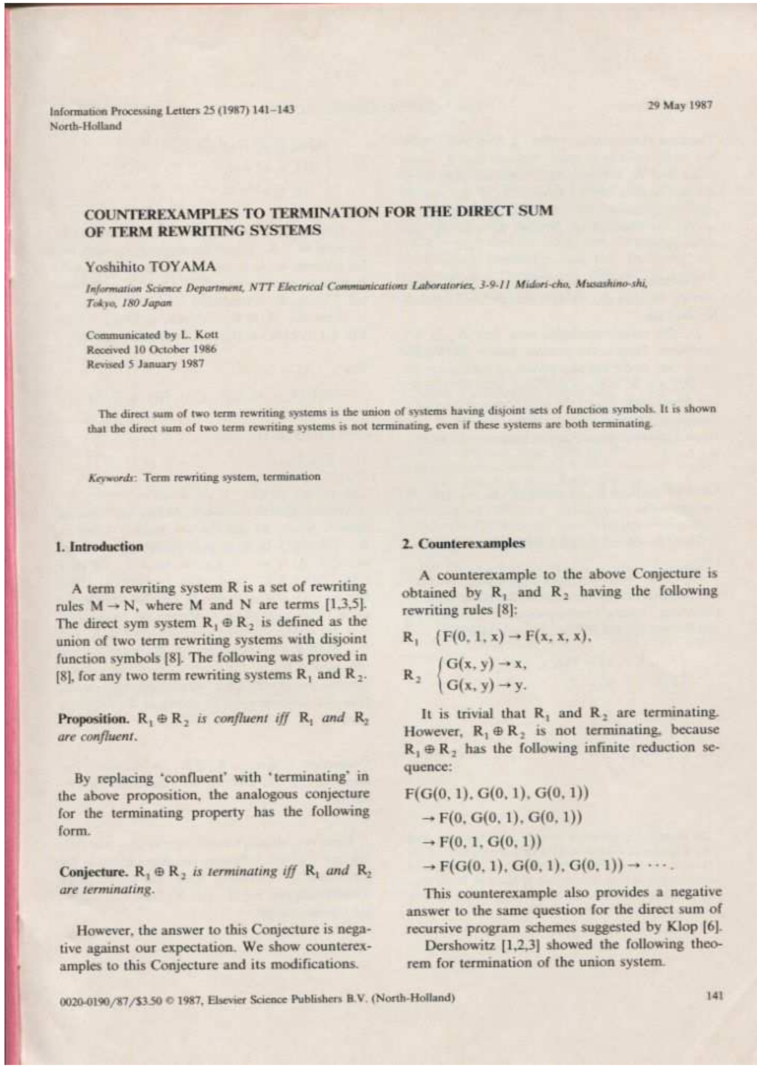
$$if(T, x, y) \rhd x,$$
$$if(F, x, y) \rhd y,$$
$$if(z, x, x) \rhd x,$$

then the direct sum $R_1 \oplus R_2$ also has the Church–Rosser property. This result gave a positive answer for an open problem suggested by O'Donnell [4].

Klop's work was done on combinatory reduction systems having the following restrictions: R_1 is a regular (i.e., linear and nonambiguous) system, and R_2 is a nonlinear system having specific rules such as $D(x, x) \rhd x$. In particular, the restriction on R_1 plays an essential role in his proof of the Church–Rosser property of $R_1 \oplus R_2$; hence his result cannot be applied to combinatory reduction systems (and term rewriting systems) without this restriction.

Author's address: NTT Electrical Communication Laboratories, 3-9-11, Midori-cho, Musashino-shi, Tokyo, 180 Japan.

Journal of the Association for Computing Machinery, Vol. 34, No. 1, January 1987, pp. 128–143.



Information Processing Letters 25 (1987) 141–143
North-Holland
29 May 1987

COUNTEREXAMPLES TO TERMINATION FOR THE DIRECT SUM OF TERM REWRITING SYSTEMS

Yoshihito TOYAMA

Information Science Department, NTT Electrical Communications Laboratories, 3-9-11 Midori-cho, Musashino-shi, Tokyo, 180 Japan

Communicated by L. Kott
Received 10 October 1986
Revised 5 January 1987

The direct sum of two term rewriting systems is the union of systems having disjoint sets of function symbols. It is shown that the direct sum of two term rewriting systems is not terminating, even if these systems are both terminating.

Keywords: Term rewriting system, termination

1. Introduction

A term rewriting system R is a set of rewriting rules $M \to N$, where M and N are terms [1,3,5]. The direct sym system $R_1 \oplus R_2$ is defined as the union of two term rewriting systems with disjoint function symbols [8]. The following was proved in [8], for any two term rewriting systems R_1 and R_2.

Proposition. $R_1 \oplus R_2$ *is confluent iff* R_1 *and* R_2 *are confluent.*

By replacing 'confluent' with 'terminating' in the above proposition, the analogous conjecture for the terminating property has the following form.

Conjecture. $R_1 \oplus R_2$ *is terminating iff* R_1 *and* R_2 *are terminating.*

However, the answer to this Conjecture is negative against our expectation. We show counterexamples to this Conjecture and its modifications.

2. Counterexamples

A counterexample to the above Conjecture is obtained by R_1 and R_2 having the following rewriting rules [8]:

$$R_1 \quad \{F(0, 1, x) \to F(x, x, x),$$

$$R_2 \quad \begin{cases} G(x, y) \to x, \\ G(x, y) \to y. \end{cases}$$

It is trivial that R_1 and R_2 are terminating. However, $R_1 \oplus R_2$ is not terminating, because $R_1 \oplus R_2$ has the following infinite reduction sequence:

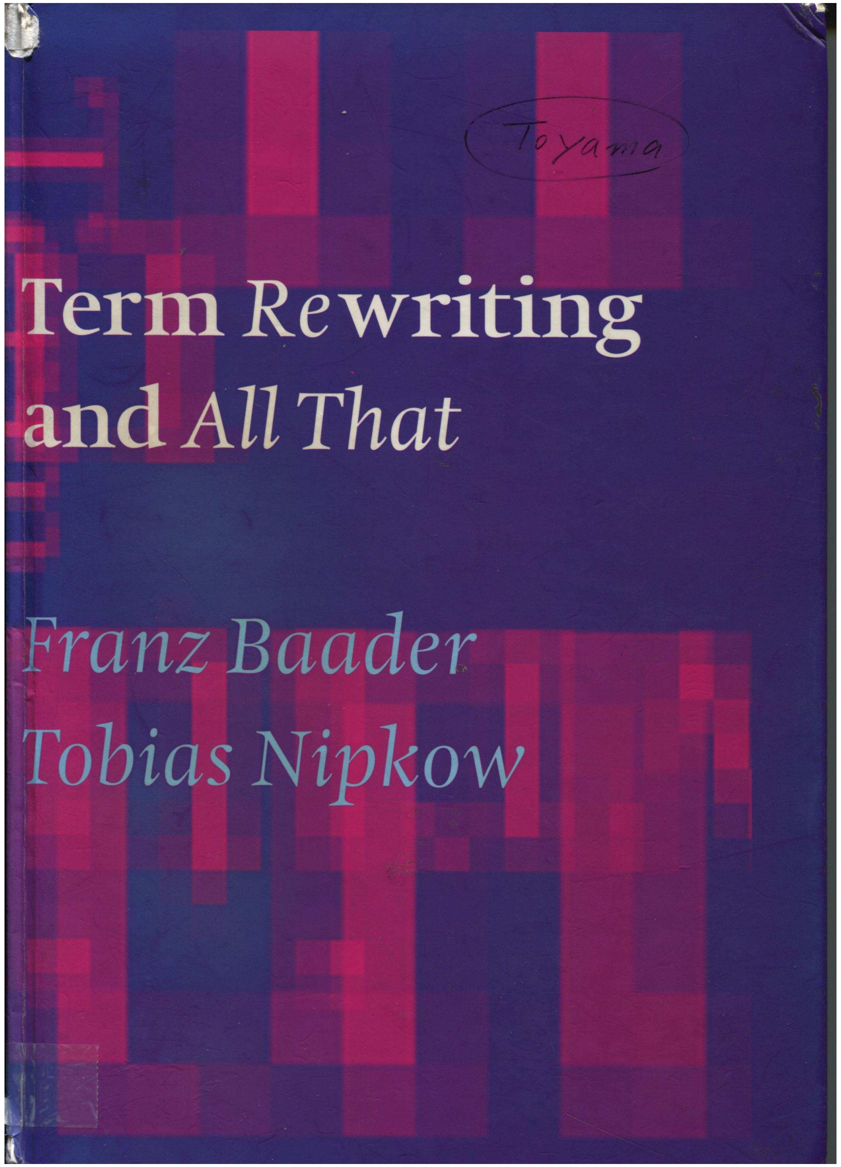
$$F(G(0, 1), G(0, 1), G(0, 1))$$
$$\to F(0, G(0, 1), G(0, 1))$$
$$\to F(0, 1, G(0, 1))$$
$$\to F(G(0, 1), G(0, 1), G(0, 1)) \to \cdots.$$

This counterexample also provides a negative answer to the same question for the direct sum of recursive program schemes suggested by Klop [6].

Dershowitz [1,2,3] showed the following theorem for termination of the union system.

Toyama's Theorem (1987) **Toyama's Counterexample (1987)**

Franz Baader and Tobias Nipkow, Term Rewriting and All That, Cambridge University Press 1998.



9 Combination Problems

9

Combination Problems

We have seen that properties like termination and confluence are in general undecidable and require sophisticated technology to solve interesting subclasses. Because the likelihood that a given TRS can be treated with a particular method decreases with the size of the TRS, it is desirable to modularize tests for confluence and termination. For example, the system $R := \{f(x,x) \to x,\ a \to g(a)\}$ cannot be shown to be confluent by any of the methods of Chapter 6 because R is neither left-linear nor terminating. However, $R_0 := \{f(x,x) \to x\}$ is terminating, has no critical pairs and is therefore confluent. Similarly, $R_1 := \{a \to g(a)\}$ is orthogonal and thus also confluent. Wouldn't it be nice if we could conclude that $R = R_0 \cup R_1$ must therefore also be confluent? A famous theorem by Toyama, which started the whole field of combination problems for term rewriting systems, asserts that this is the case because R_0 and R_1 do not share function symbols. This chapter studies under what conditions we can transfer confluence and/or termination from individual systems to their union.

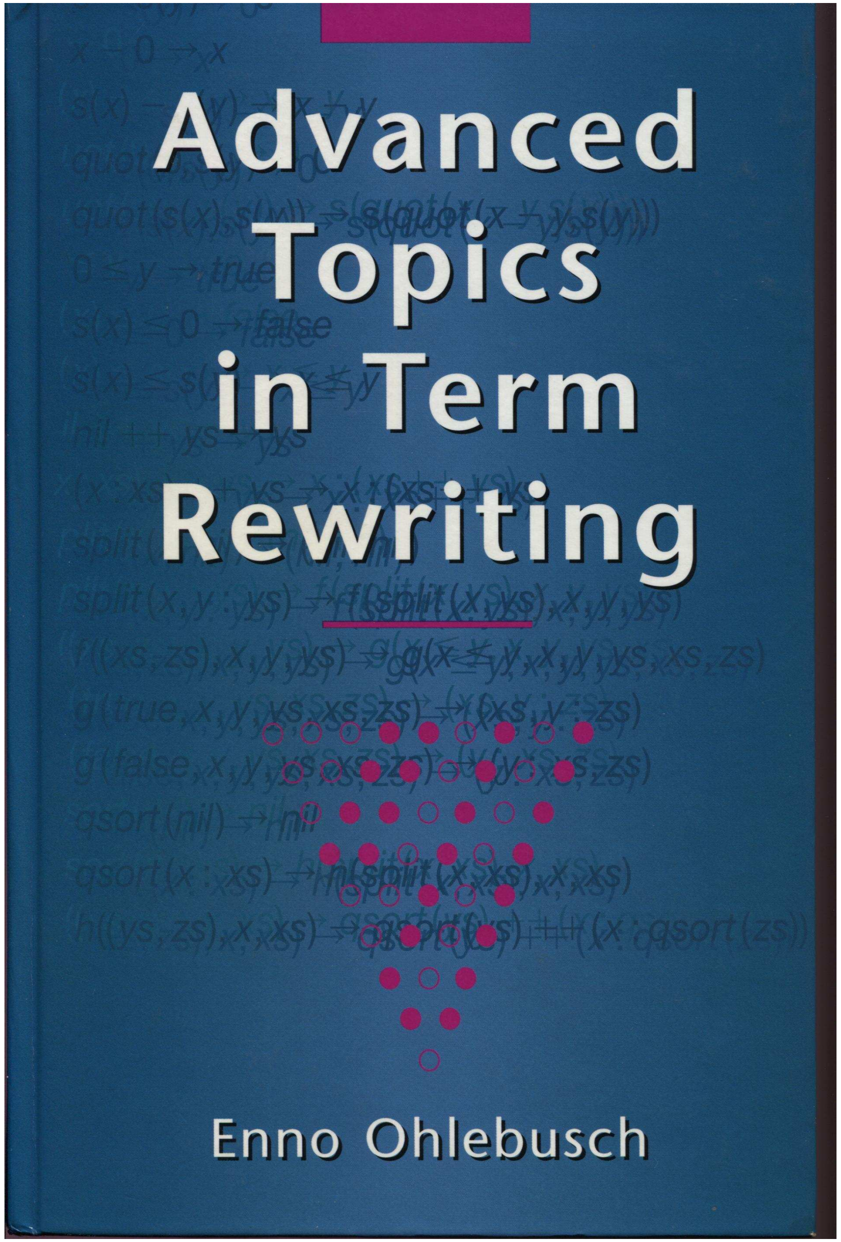
Computer scientists want to combine not just properties but also algorithms. Hence the final substantive section in this chapter is devoted to one particularly well-behaved instance, that of combining decision procedures for the word problem: given decision procedures for $\approx_{E_0}$ and $\approx_{E_1}$, how can we decide $\approx_{E_0 \cup E_1}$? Of course, for arbitrary E_0 and E_1 this is not possible, but if they do not share function symbols, it is.

9.1 Basic notions

It is obvious that the less interaction there is between two term rewriting systems R_0 and R_1, the easier combination problems become. Although most of the time we restrict ourselves to the case where R_0 and R_1 do not share function symbols, the problems are still far from trivial.

200

Enno Ohlebusch, Advanced Topics in Term Rewriting, Springer-Verlag 2002.



8 Modularity

8

Modularity

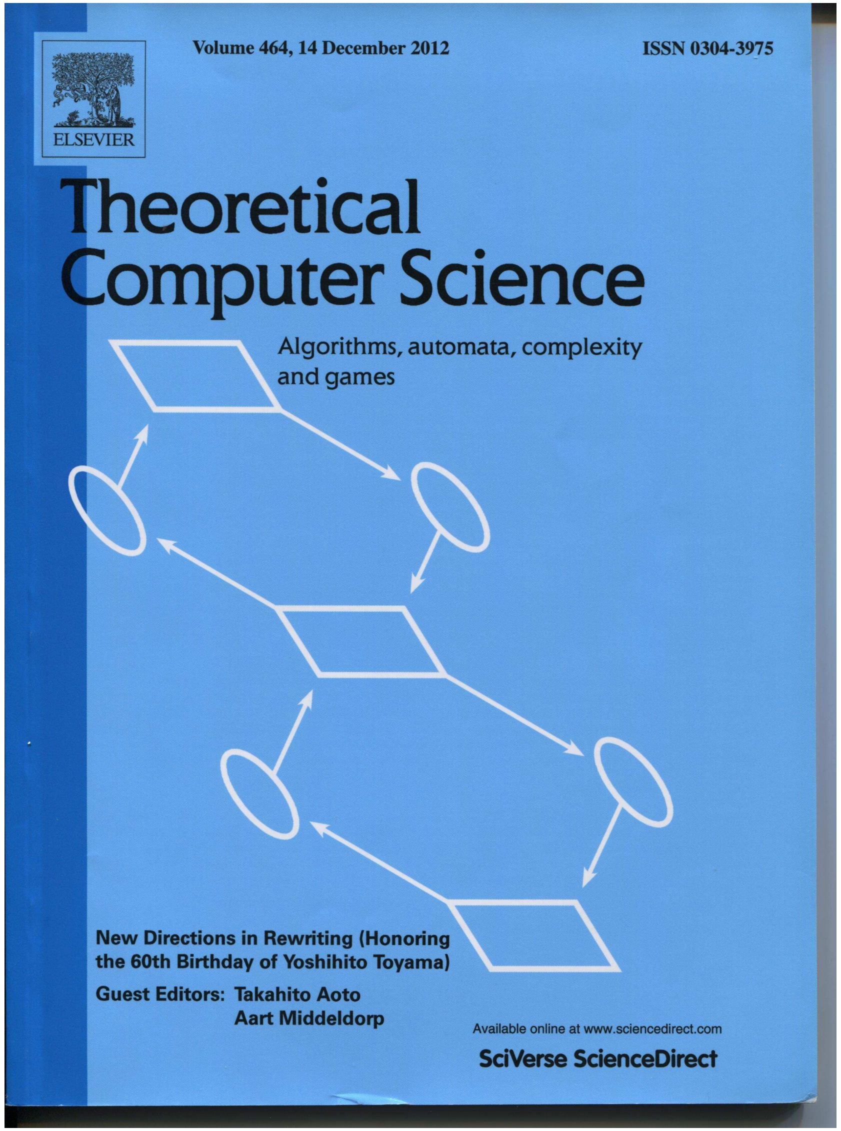
Modularity is a well-known programming paradigm in computer science. Programmers should design their programs in a modular way, that is, as a combination of small programs. These so-called modules are implemented separately and are then integrated to form the program as a whole. Because TRSs have many important applications in computer science, it is important (not only from a theoretical viewpoint but also from a practical point of view) to know under which conditions a combined system inherits desirable properties from its constituent systems. For this reason modular aspects of term rewriting have been extensively studied in the past decade. To render a detailed account of all known modularity results goes beyond the scope of this book. Instead, we will give an overview of the most important results for TRSs (Section 8.2) and CTRSs (Section 8.7). We will also present selected results in detail.

8.1 Different Kinds of Combinations

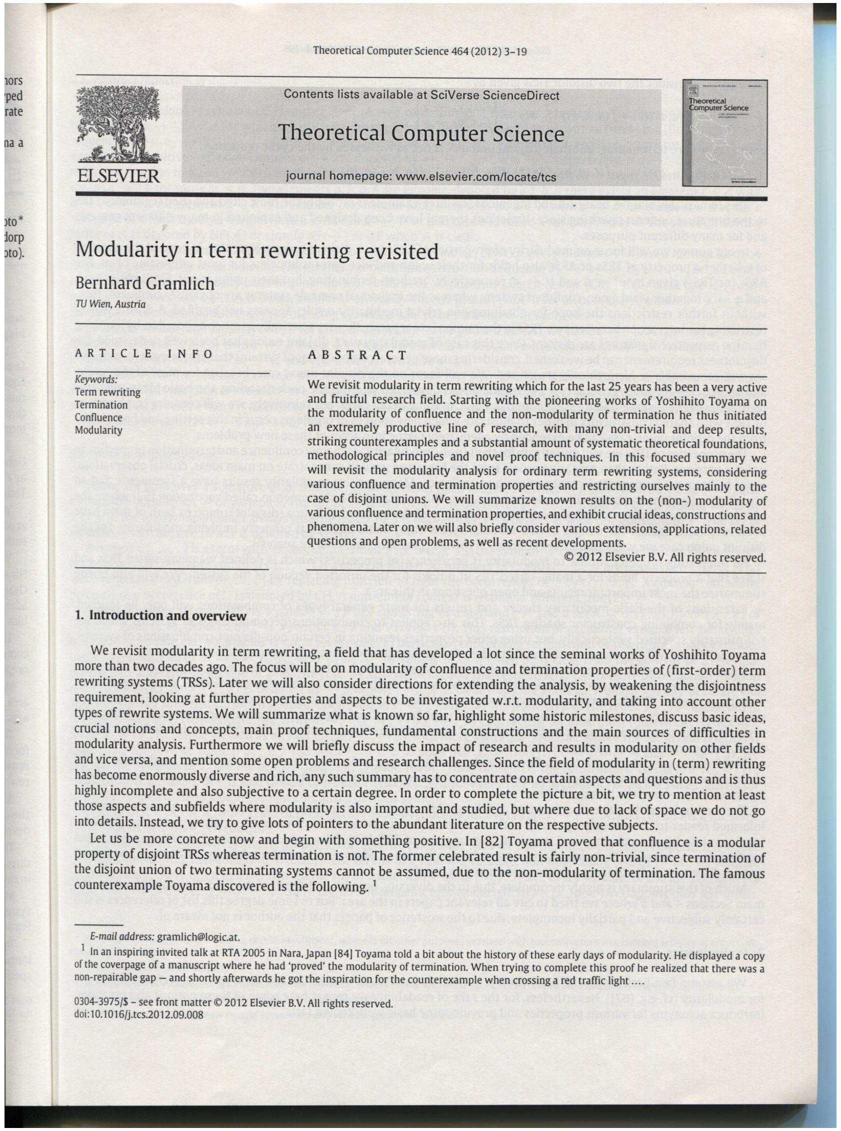
Definition 8.1.1 A property $\mathcal{P}$ is *modular* for TRSs if, for all TRSs $(\mathcal{F}_1, \mathcal{R}_1)$ and $(\mathcal{F}_2, \mathcal{R}_2)$ having property $\mathcal{P}$, their *combination* (union) $(\mathcal{F}, \mathcal{R}) = (\mathcal{F}_1 \cup \mathcal{F}_2, \mathcal{R}_1 \cup \mathcal{R}_2)$ also has the property $\mathcal{P}$.

The knowledge that (perhaps under certain conditions) a property $\mathcal{P}$ is modular allows an incremental development of programs. On the other hand, it provides a divide-and-conquer approach to establishing properties of TRSs. If one wants to know whether a large TRS has a certain modular

Theoretical Computer Science, Vol.464, 2012
New Direction in Rewriting
(Honoring the 60th Birthday of Yoshihito Toyama)



Bernhard Gramlich, Modularity in term rewriting revisited



Theoretical Computer Science 464 (2012) 3–19

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Modularity in term rewriting revisited

Bernhard Gramlich

TU Wien, Austria

ARTICLE INFO

Keywords:
Term rewriting
Termination
Confluence
Modularity

ABSTRACT

We revisit modularity in term rewriting which for the last 25 years has been a very active and fruitful research field. Starting with the pioneering works of Yoshihito Toyama on the modularity of confluence and the non-modularity of termination he thus initiated an extremely productive line of research, with many non-trivial and deep results, striking counterexamples and a substantial amount of systematic theoretical foundations, methodological principles and novel proof techniques. In this focused summary we will revisit the modularity analysis for ordinary term rewriting systems, considering various confluence and termination properties and restricting ourselves mainly to the case of disjoint unions. We will summarize known results on the (non-) modularity of various confluence and termination properties, and exhibit crucial ideas, constructions and phenomena. Later on we will also briefly consider various extensions, applications, related questions and open problems, as well as recent developments.

1. Introduction and overview

We revisit modularity in term rewriting, a field that has developed a lot since the seminal works of Yoshihito Toyama more than two decades ago. The focus will be on modularity of confluence and termination properties of (first-order) term rewriting systems (TRSs). Later we will also consider directions for extending the analysis, by weakening the disjointness requirement, looking at further properties and aspects to be investigated w.r.t. modularity, and taking into account other types of rewrite systems. We will summarize what is known so far, highlight some historic milestones, discuss basic ideas, crucial notions and concepts, main proof techniques, fundamental constructions and the main sources of difficulties in modularity analysis. Furthermore we will briefly discuss the impact of research and results in modularity on other fields and vice versa, and mention some open problems and research challenges. Since the field of modularity in (term) rewriting has become enormously diverse and rich, any such summary has to concentrate on certain aspects and questions and is thus highly incomplete and also subjective to a certain degree. In order to complete the picture a bit, we try to mention at least those aspects and subfields where modularity is also important and studied, but where due to lack of space we do not go into details. Instead, we try to give lots of pointers to the abundant literature on the respective subjects.

Let us be more concrete now and begin with something positive. In [82] Toyama proved that confluence is a modular property of disjoint TRSs whereas termination is not. The former celebrated result is fairly non-trivial, since termination of the disjoint union of two terminating systems cannot be assumed, due to the non-modularity of termination. The famous counterexample Toyama discovered is the following. [1]

E-mail address: gramlich@logic.at.

[1] In an inspiring invited talk at RTA 2005 in Nara, Japan [84] Toyama told a bit about the history of these early days of modularity. He displayed a copy of the coverpage of a manuscript where he had 'proved' the modularity of termination. When trying to complete this proof he realized that there was a non-repairable gap – and shortly afterwards he got the inspiration for the counterexample when crossing a red traffic light ....

0304-3975/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2012.09.008

オランダへ行く

オランダ国立数学·計算機科学研究所 (**CWI**) に **1990** 年 **8** 月より **1** 年間滞在。

- ヨーロッパの応用数学や計算機科学の理論研究の中心地のひとつ -

Jan Willem Klop 先生の研究グループや **Henk Barendregt** 先生の研究グループと共同研究。

オランダの博士論文審査会

大学内の行事というよりも、もっと公な儀式とみなされ、希望すれば誰でも参加可能。

オランダの博士論文審査会

大学内の行事というよりも、もっと公な儀式とみなされ、希望すれば誰でも参加可能。

特別なステッキをかざした先導係、「二人の乙女」に導かれた **Aart Middeldorp** さん、数珠を首にかけた議長、主査の **Klop** 先生、筆頭審査員の私、審査員である６人の教授達の順に入場。

オランダの博士論文審査会

Q「 学長の権威と私の権利にもとづいて質問をさせていただきたい。」

A「 非常に学識豊かな質問者である＊＊様にお答えいたします。」

オランダの博士論文審査会

Q「 学長の権威と私の権利にもとづいて質問をさせていただきたい。」

A「 非常に学識豊かな質問者である＊＊様にお答えいたします。」

舞台に向かってゆっくり進んできた先導係が突然「 時間である」とラテン語で叫びステッキで床をドスンとつくと質疑はピタリと打ち切り。

長距離徒歩旅行

Klopご夫妻、Aartさん、Fer-Janさん

長距離徒歩旅行

35km! ゴールは遠かった。

TRS ミーティング

オランダから帰国後、日本にも書き換えシステムに関する自由な研究交流の場をつくろうと決心。

- 参加者は必ず何か発表すること。
- 発表は英語で行うこと (第3回より)。

1991年**12**月に第**1**回を大山口通夫先生と三重大学で開催。

(TRS: Term Rewriting System)

TRS ミーティング

48: Sendai (2018) 47: Matsue (2017) 46: Shinojima (2017) 45: Obergurgl (2016) 44: Kanazawa (2016) 43: Morioka (2015) 42: Tokyo (2015) 41: Sapporo (2014) 40: Unazuki (2014) 39: Akita (2013) 38: Kofu (2013) 37: Sendai (2012) 36: Matsue (2012) 35: Nagoya (2011) 34: Aizu (2011) 33: Tsu (2010) 32: Sendai (2009) 31: Kaga (2009) 30: Sapporo (2008) 29: Tokyo (2008) 28: Osaka (2007) 27: Katayamazu (2006) 26: Sakunami (2006) 25: Tsu (2004) 24: Shimane (2004) 23: Nagoya (2003) 22: Yakushima (2003) 21: Noto Omakidai (2002) 20: Hitachi Daigo (2002) 19: Sendai (2001) 18: Sakunami (2001) 17: Amagasaki (2000) 16: Kiryu (2000) 15: Yufuin (1999) 14: Nara (1999) 13: Hokkaido (1998) 12: Nagoya (1997) 11: Tsukuba (1997) 10: NTT CS Lab (1996) 9: Hatoyama (1996) 8: Tsu (1995) 7: Nomi (1995) 6: Sapporo (1994) 5: Tsukuba (1994) 4: Gifu (1993) 3: Makuhari (1992) 2: NTT CS Lab (1992) 1: Tsu (1991)

大学で研究する

北陸先端科学技術大学院大学 (JAIST)
計算機言語学講座に1993年4月着任。

酒井正彦さんが助教授。
酒井さん離任後、青戸等人さん、鈴木太郎さんが助手。

博士号取得： 長谷崇、岩見宗宏、草刈圭一郎

東北大学
電気通信研究所 コンピューティング情報理論研究分野および
情報科学研究科 情報論理学講座(協力講座)に2000年4月着任。

鈴木太郎さん、草刈圭一朗さんが助手。
鈴木さん・草刈さん離任後、
青戸等人さんが准教授、菊池健太郎さんが助教。

博士号取得： 千葉勇輝

証明は計算できる

計算システムの効率性と証明システムの柔軟性をあわせもつ新しい計算・証明融合パラダイムを目指して研究。

理論的アプローチだけではなく、スタッフや学生と協力して実験的アプローチも試みる。

- 書き換えシステムの基礎理論 (理論的アプローチ)
- 書き換え理論に基づく定理自動証明 (実験的アプローチ)
- プログラムの自動変換・検証 (実験的アプローチ)

合流性の自動証明

世界初の合流性自動証明システム **ACP**
(Automated Confluence Prover)

青戸等人さんとの共同プロジェクト

- **2007** 年頃から開発
- モジュラ性に基づく分割統治判定
- **ACP** として国際会議 **RTA 2009** にて発表

Confluence Competition

- The completitions took part in IWC (Interntional Workshop on Confluence)
- YES/NO should be followed by a proof argument which is understandable by human experts.
- 100–150 problems (almost) from literature are considered.
- Timeout of 60 seconds for each problem.

ACPの成績

2012年 1位、2013年 1位、2014年 1位、2015年 1位
2016年 2位、2017年 2位

- 理論と実験の両輪が補いあって発展 -

なぜ研究するのか？

Q「 確かに面白いが、それが何の役に立つのかね？ 」

なぜ研究するのか？

Q「 確かに面白いが、それが何の役に立つのかね？」

「 前にはあんなに物理をやるのが楽しかったというのに、今はいささか食傷気味だ。なぜ昔は楽しめたのだろう？ そうだ、以前は僕は物理で遊んだのだった。いつもやりたいと思ったことをやったまでで、それが核物理の発展のために重要であろうがなかろうが、そんなことは知ったことではなかった。ただ僕が面白く遊べるかどうかが決めてだったのだ。」(ファインマン)

なぜ研究するのか？

Q「 確かに面白いが、それが何の役に立つのかね？」

A「 なんの役にも立たないよ。面白いからやってるだけさ。」

「 前にはあんなに物理をやるのが楽しかったというのに、今はいささか食傷気味だ。なぜ昔は楽しめたのだろう？ そうだ、以前は僕は物理で遊んだのだった。いつもやりたいと思ったことをやったまでで、それが核物理の発展のために重要であろうがなかろうが、そんなことは知ったことではなかった。ただ僕が面白く遊べるかどうかが決めてだったのだ。」(ファインマン)

なぜ研究するのか?

Q「 確かに面白いが、それが何の役に立つのかね? 」

A「 なんの役にも立たないよ。面白いからやってるだけさ。」

退職は新たな面白い遊びを発見するチャンス!

なぜ研究するのか？

Q「 確かに面白いが、それが何の役に立つのかね？」

A「 なんの役にも立たないよ。面白いからやってるだけさ。」

退職は新たな面白い遊びを発見するチャンス！

今日までの研究人生を支えて下さった皆様に
心より感謝いたします。