

電気回路のシミュレーションと学校教育での利用

三 仲 啓
(1986年10月15日 受理)

Computer Simulation of the Electric Circuit for School Education

Akira MINAKA

1. はじめに

電池と電球を任意に接続した電気回路をパーソナルコンピュータの画面上で作成し、各電球・電池に流れる電流を表示するシミュレータを紹介する。

現在の学習指導要領では小学校第4学年から電池と電球の接続法が登場するが、電気回路の理解には高度のパターン認識が必要で、簡単なルールを覚えることだけではすまない。ある回路に配線を1本追加するだけで様相が一変することも多く、学習者の理解を混乱させる。このため大学生でも直列・並列、ショート回路などが理解できていない者がかなりの割合みられるのが実情である。

この難関を克服するには、多数の実例を体験し、注意力が及ぶ範囲を広げることが必要であろう。ここで紹介する電気回路のシミュレータは、学校教育の中で適切に利用すれば上記の目的に有効と思われる。

次章では、プログラムの機能と仕様を紹介するが、文章と流れ図だけでは操作の実感が得られないので、興味のある方は是非実物に向かって頂きたい。

第3章では、回路分析のためのアルゴリズム、特に Kirchhoff の第2法則を適用する際に必要な独立な閉回路の作り方を紹介する。これは、電気回路に限らず一般の閉回路網の分析にも利用できる応用性の広いものであるから、やや詳細に説明してある。

最後の章で、理科教育におけるコンピュータシミュレーションの意義と本プログラムの教育利用に向けての問題点を議論する。

プログラムの詳細に興味が無ければ、2.2節と3章は省略できる。

2. プログラムの機能と仕様

2.1 プログラムの機能

ここでは、キー入力操作に応じて本プログラムの機能を説明する。

プログラムを起動すると、次の初期メニューが表示される。

[1] 説明	プログラムの機能と操作法の説明
[2] 練習	与えられた電池、電球に配線し結果をみる
[3] 作成／保存	任意の回路を作成し結果をみる（回路はフロッピーに保存できる）
[4] 再現	保存した回路を呼び出し結果をみる
[5] おわり	終了

数字キー1～5を押すことにより、それぞれの処理に移る。この外に、メニューには表示されない命令（TEACHER）を入力することにより、保存ファイルの各種処理ができる。隠しコマンドにしたのは、子供がこのプログラムを使用したときに、保存ファイルを破壊しないための用心である。

また、キー入力にミスはつきものであるので、以下に説明する各キー入力の際に、ESCキーを押すとひとつ前の状態に戻るようになっている。

[2] では、電池1個または2個、電球1個または2個を選択し、2個の場合にはさらに直列か並列かを選択する。画面に指定した個数の電池と電球が表示されるので、カーソルキーで配線をする。配線終了を示すリターンキーを押すと、正しい配線か否かが判断され、簡単なメッセージが表示される。また、正解の場合には電流に応じた明るさで電球が点灯する。

これはシミュレーションというより「ドリル型」と称されるプログラムに属するが、¹⁾ ここでは[3]の準備として、カーソルキーによる配線に慣れることを主たる目的としている。

[3] では、電池、電球の数は各7個以下、接続点の数は20個以下という制限はあるが、スイッチを含む任意の回路が作成できる。スイッチのON/OFFを設定すると、結果が、電球の点灯と各電球・電池の電流値で表示される。ショートした電池があれば、それ表示される。スイッチを入れ直して結果をみたり、配線を少し修正してみることも可能である。また、30文字以内の説明を付け回路をフロッピーディスクに保存することもできる。図2-1に全体の概略を示す流れ図を、図2-2には電池の配置と配線の部分の流れ図を示す。

[4] では、保存された回路の一覧表から再現したいものを選び出し、[3]と同様にスイッチを設定し、結果をみる。スイッチの入れ直しや別な回路をみることはできるが、配線の変更や回路の保存はできない。

TEACHERで呼び出される補助プログラムでは、

- ・保存されている回路の説明文の一覧表示
 - ・保存回路の並べ換え
 - ・説明文の修正
 - ・保存回路の消去
- 等ができる。

[3] 作成／保存 ― 基本構造 ―

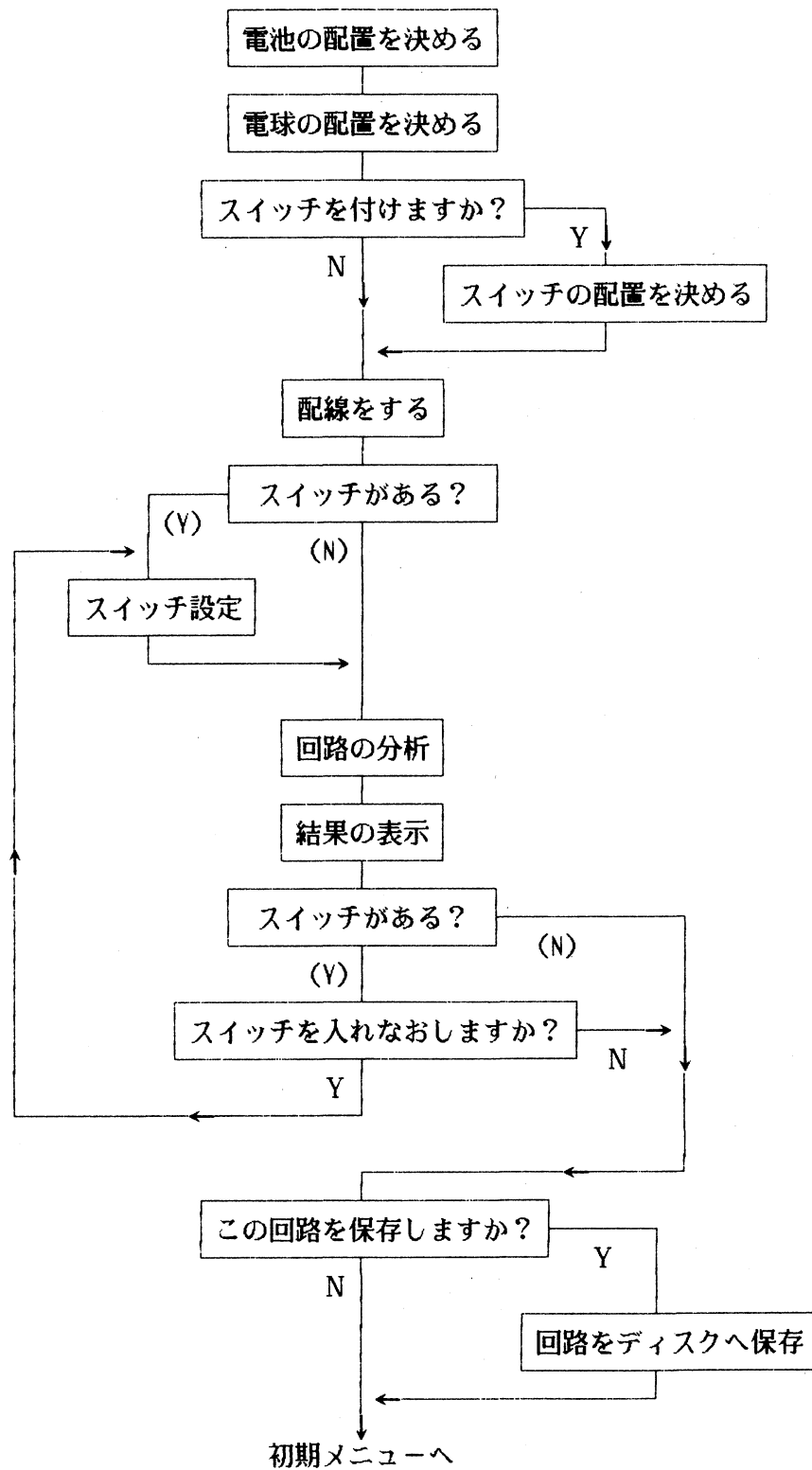
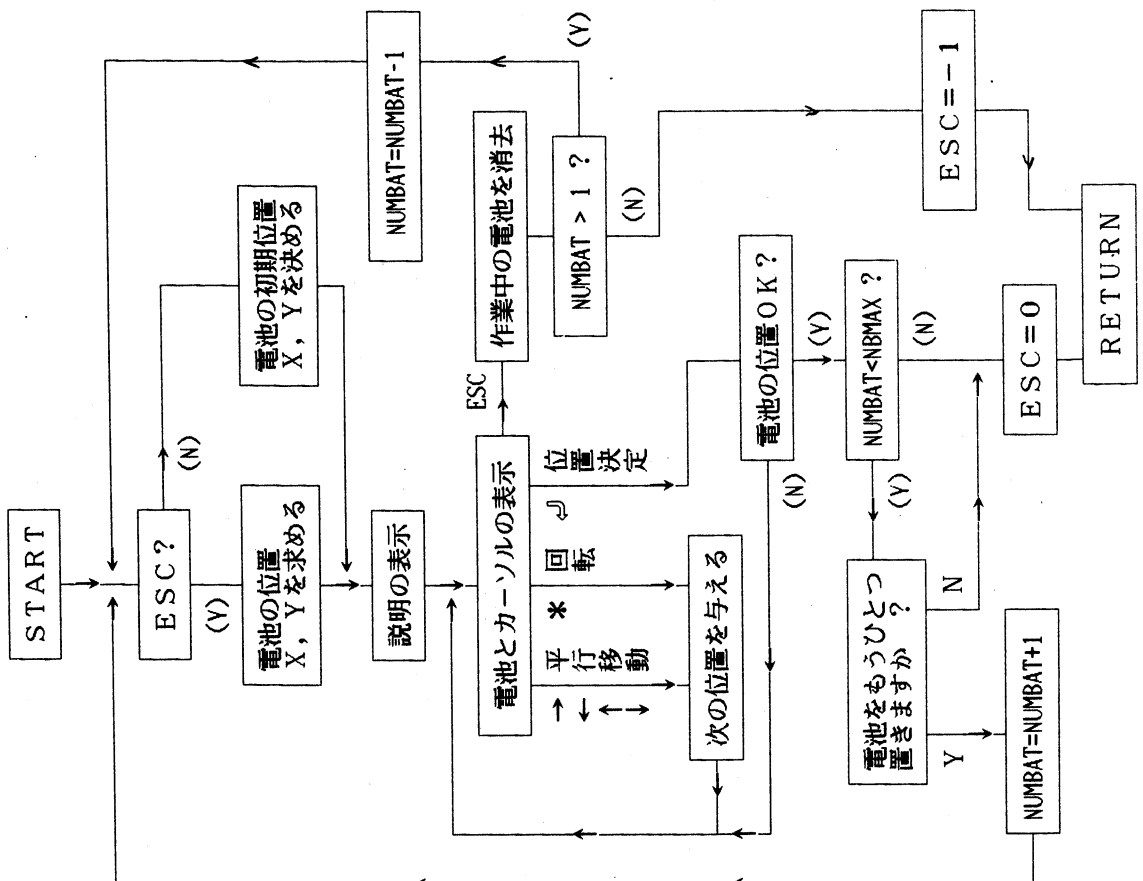


図2-1 作成／保存の処理の概略を表わす流れ図。この中の「電池の配置を決める」部分と「配線をする」部分の詳細な流れ図は、図2-2に示す。

(a) 電池の配置を決める [PUT. BAT]

NUMAX (許される電池の最大数)
NUMBAT (配置する電池の番号) を与える



(b) 配線をする [DRAW. LIN]

配線書き始めの位置を決める
(通常、電池1の+極)

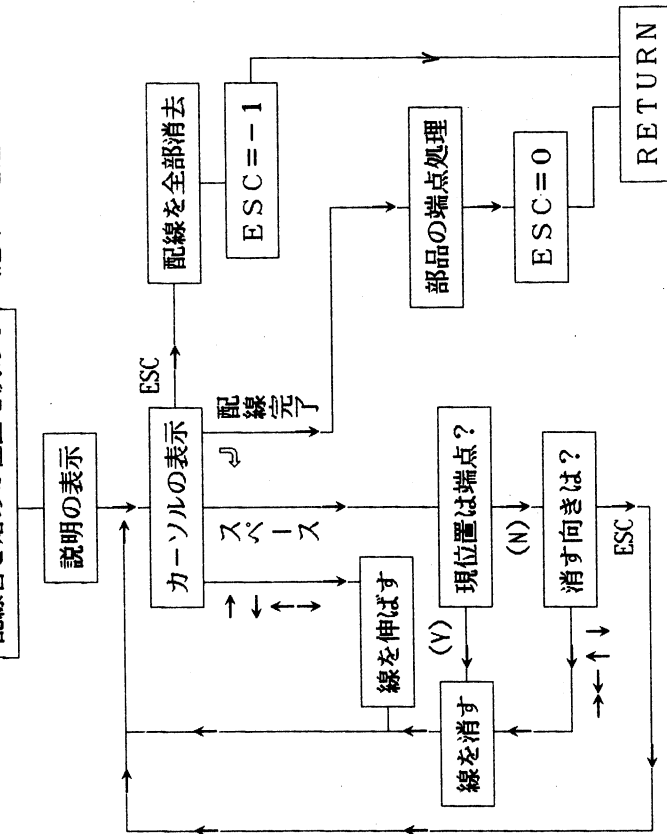


図2-2 (a)電池の配置を決める部分の流れ図。電球やスイッチの配置を決める部分もこれと同様。

(b)配線をする部分の流れ図。ESCで終了 (ESC=-1) したときは、最後のスイッチまたは電球の配置をするルーチンに戻る。

2.2 プログラムの仕様

(a) 必要なハード構成

本体はNECのPC-8801とPC-9801シリーズ(XAを除く)で、CRTはグラフィック画面で縦400ライン表示できる高解像のものが必要である。またフロッピーディスクドライブは1台あれば使用できる。

(b) 言語とOS

プログラム言語はBASIC, OSはNBS-DISK BASICであるが、PC-8801版は一部機械語を使用している。

(c) 回路図の構成

電気回路はグラフィック画面にスクリーンエディット方式で描かれるが、このとき画面は16×16ドットの細胞に分割されており、電池(3×5)・電球(3×3)・スイッチ(2×3)・配線要素(1×1)はそれぞれ()内に示した細胞数からつくられている。

配線要素には、端点(□, □, □, □), 結線(□, □, □, 等), 接続点(⊕, ⊕, ⊕, ⊕), 交差(⊗)がある。

(d) 回路図の配列変数への保存

画面上の各細胞の座標(X, Y)は、WIDTH 40, 25のときのキャラクタ座標で表わすことにし、回路図が描ける領域を(X_{min}, Y_{min}) ~ (X_{max}, Y_{max})とする。この領域に描かれた回路図は整数型配列変数TBL()に保存される。TBL()の引数(TBLP=table pointerと呼ぶことにする)は、処理の高速化を考え1次元としてあるが、画面上の座標(X, Y)と

$$TBLP = X - X_{\min} + (X_{\max} - X_{\min} + 1) \times (Y - Y_{\min})$$

という関係にある。

TBL(TBLP)の内容は次のように定義する。

- ・配線要素のとき、「向き」を上, 右, 下, 左の順に0, 1, 2, 3と決め,

$$\square = 2^0 = 1, \quad \square = 2^1 = 2, \quad \square = 2^2 = 4, \quad \square = 2^3 = 8$$

の和で表わす。つまり、空白 □ = 0, 端点は上記4つ, 結線□ = 3, □ = 5, 等, 接合点⊕ = 7, ⊕ = 14, 等, 交差⊗ = 15 まで 0 ~ 15の値をとる。

- ・電池のとき、電池が占めるTBLPについて、 $TBL(TBLP) = 2^4 \times (\text{電池No.})$.

ただし、+-極の位置には上記の端点の値を加える。

- ・電球のとき、電球が占めるTBLPについて、 $TBL(TBLP) = 2^8 \times (\text{電球No.})$.

ただし、電球からの配線の端点部には対応する端点の値を加える。

- ・スイッチのとき、スイッチが占めるTBLPについて、 $TBL(TBLP) = 2^{12}$.

ただし、端点部にはその値を加える。また、スイッチのON/OFF設定が終わり回路の分析が始まるときには、ONなら結線, OFFなら端点の値に置き換えられる。

(e) 回路図のディスクへの保存

シーケンシャルファイルで、次の順に保存される。

説明文 (30文字以内), X_{\min} , Y_{\min} , X_{\max} , Y_{\max} , TBL(),

電池の数 (NUMBAT), 各電池の-+極のTBLP (POSBAT()),

電球の数 (NUMLAM), 各電球の端点のTBLP (POSLAM()),

スイッチの数 (NUMSW), 各スイッチの端点のTBLP (POSSW())

3. 回路分析のアルゴリズム

3.1 Kirchhoff の法則を適用する際の問題点とその解決法

電池と電球だけからなる単純な回路であっても、全く任意の配線に対して各内線に流れる電流を求めるには、Kirchhoff の法則²⁾によらなければならない。この法則は、

〔第1法則〕 各接続点で電流が保存する。

〔第2法則〕 任意の閉回路につき、起電力の和は抵抗による電圧降下の和に等しい。

という2つの法則からなる。任意の閉回路網につき、第1法則から (接続点数-1) 個、第2法則から (独立閉回路数) 個の方程式が得られるが、多面体に関するオイラーの定理と同様な関係、

$$(\text{内線数}) = (\text{接続点数}) + (\text{独立閉回路数}) - 1$$

があるので、この連立方程式から各内線の電流を求めることができる。第2法則の独立な閉回路の選び方には任意性があるが、人間のパターン認識能力は優秀なので、少し慣れれば必要十分かつ最も簡単な閉回路を選ぶことができる。

しかし、コンピュータにこの作業をさせるには、独立な閉回路を選び出す明確なアルゴリズムが必要となる。これがプログラム上最大の問題であるが、ここでは Kirchhoff により示されている「生成樹」(generating tree) の考え方を利用する。^{2,3)} 図3-1(a)の閉回路網に対する生成樹の例を同図(b), (c)に示す。生成樹に点線で示した内線 (以下CL=cycle closing linkと略す) を1つ追加すれば1つの閉回路 (cycle) が得られ、CLが独立な閉回路と1対1に対応することがわかる。

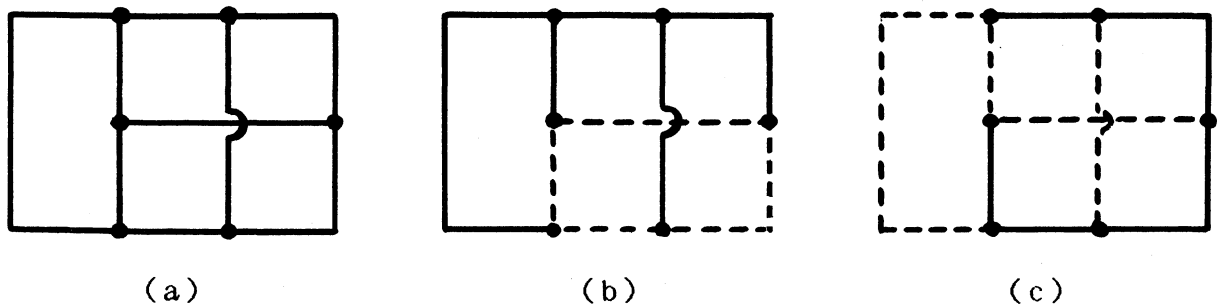


図3-1 (a)閉回路網の例。ここでは、電池、電球は省略されている。

(b)(c)回路網(a)に対する生成樹の例。実線の生成樹に、点線で示した内線を1本加える毎に独立な閉回路が1つ決まる。

本プログラムでは、接続点に集まる内線は3本に限られているので、生成樹とCLを作ることは比較的容易である。具体的なアルゴリズムは次節に示すが、その前にこれ以外の問題点とその解決法を列挙しておく。

- ・ショートした電池がある場合

電池には必ず内部抵抗を入れておく。

- ・途切れた線がある場合 (図3-2(a))

最初に途切れた線を消去しておく。これにともない接続点が消去されることもある。開かれたスイッチも途切れた線として扱われる。

- ・解が不定になる場合 (図3-2(b)) 図の内線ア, イ, ウに流れる電流は不定となる。

等電位となる接続点が閉回路を作るときは、その中の内線を1本消去する。

- ・連結されていない回路が複数ある場合 (図3-2(c))

各回路網ごとに生成樹を作る。

- ・接続点を持たない閉回路がある場合 (図3-2(d))

生成樹を作る際に参照されていない電池の+極から内線をたどって一極に達するかを調べる。

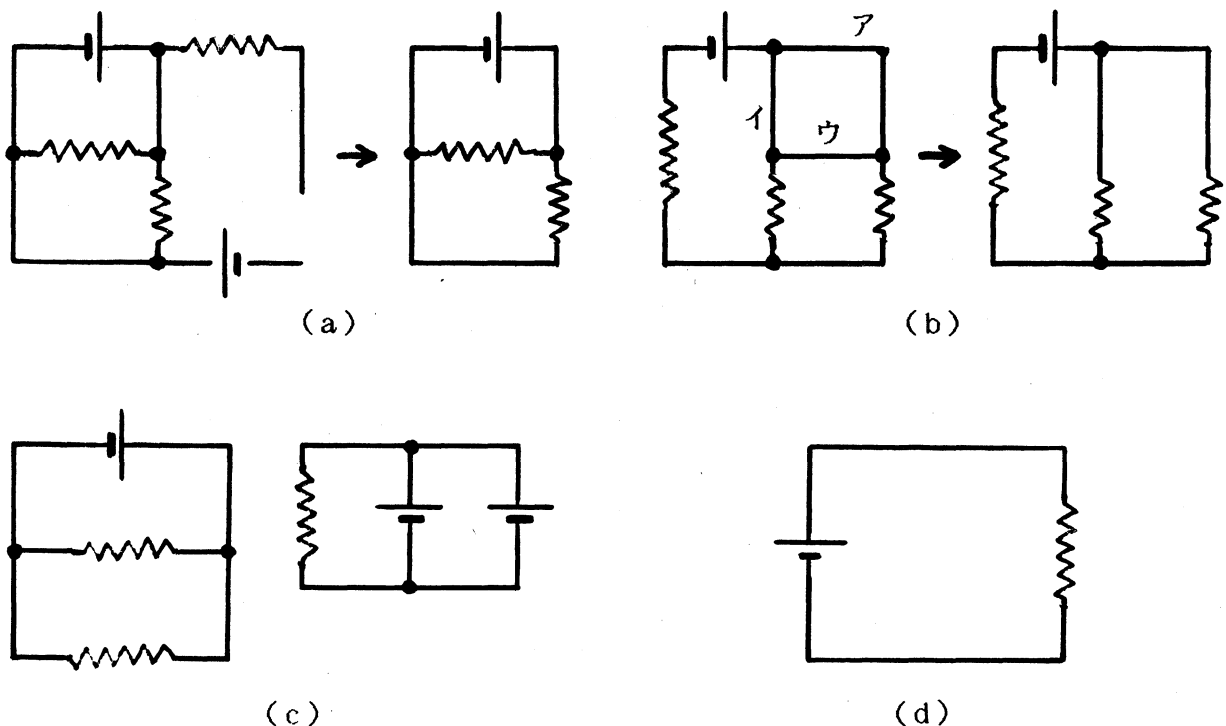


図3-2 (a)途切れた配線があれば、その線と接続点を消去する。
 (b)等電位になる閉回路があれば、内線と接続点を消去する。
 (c)非連結な閉回路網が2つある例。
 (d)接続点を持たない閉回路の例。

3.2 BASICによる回路分析の手順

ここでは、前節の考察に基づくアルゴリズムを示す。実際のプログラムでは処理の高速化のために細部に工夫がなされているが、煩雑になるので以下では省略してある。(各項目の後の[]内は、サブルーチン名)

(a) 途切れた線の消去 [OPEN. END]

途切れた線の端点から配線をたどり別な端点または接続点までを消去する。

終点が接続点の場合はTBL()の該当箇所を接続点から結線に置き換える。

途中で電池があれば、その電池は調査済みであることを記録する。

(b) 接続点への番号付け [NAME. VERTEX]

適当な順序で各接続点に番号をふってゆき、各接続点のTBLPと全接続点数を記録する。

全接続点数は2の整数倍になっている。これが0の場合は、(g)に跳ぶ。

(c) 接続関係表の作成と内線への番号付け [MK. REL]

接続点Vからある向き(D=0~3)に配線をたどると、接続点EVの向きEDに至るとき、接続関係表REL()に

$$REL(4 * V + D - 4) = 4 * EV + ED,$$

$$REL(4 * EV + ED - 4) = 4 * V + D$$

と記録する。同時に、内線に番号Lを付けてゆき、それが(V, D)と(EV, ED)を結ぶものであることを記録する:

$$V < EV \text{ のときは, } LINK(L) = 256 * (4 * V + D) + 4 * EV + ED$$

$$V > EV \text{ なら, } LINK(L) = 256 * (4 * EV + ED) + 4 * V + D$$

さらに、この内線の抵抗R!(L), 向きを考慮した起電力E!(L), および電池(BATNO), 電球(LAMNO)がこの内線上にあること

$$BATLIN(BATNO) = L : LAMLIN(LAMNO) = L$$

を記録してゆく。

(d) 不要な内線と接続点の消去 [MK. EQ. POT. TBL]

常に等電位となる接続点のグループをつくってゆく。等電位となる2つの接続点が途中で電池がない内線で結ばれるときは、その内線と両端の接続点を消去する。これに応じて、REL(), 内線番号, 内線上の抵抗, 起電力を書き変える。(図3-3)

有効接続点数(=全接続点数-ここで消去された接続点数)が0なら、(g)に跳ぶ。

(e) 生成樹と独立な閉回路を作る [MK. TREE]

再帰用法が困難なBASIC言語によるので、枝分れが比較的少ない生成樹を作ることを考える。この生成樹は、図3-4に示す例のように幹と枝からなる。枝は各接続点から高々1本出るものとし、各枝はそれが派生する接続点の番号(SV=start vertexと略)で区別する。幹のSVは0としておく。生成樹は、それを形成する接続点(TV=tree vertex)の列{TV}, 各TVのSV

の列 $\{SV\}$ ，生成樹を作る内線 (TL = tree link) の列 $\{TL\}$ で記録される。また，この生成樹を作る過程で閉回路を作る内線 CL も得られるので，その並び $\{CL\}$ も記録する。

図 3-5 に $\{TV\}$ ， $\{SV\}$ ， $\{TL\}$ ， $\{CL\}$ を作る流れ図を示す。

(f) 各生成樹に Kirchhoff の法則を適用し，連立方程式を解く

・第 2 法則 [KIRCH2]

$\{CL\}$ 中の各 CL につき閉回路を内線の列で表わすことが問題になるが，ここで作られた生成樹の次のような特性を利用する。CL の両端の接続点のうち， $\{TV\}$ の中で先に現われるものを V_1 ，後のものを V_2 とすると， V_2 は， V_1 と同じ世代か 1 つ後の世代の枝（または幹）にある。従って V_1 ， V_2 が同じ枝にないときには，まず V_2 をその SV までたどり，これを V_2 とする。 V_1 ， V_2 が同じ枝になれば，今度は V_1 をその SV までたどり，これを V_1 とする。以下同様に，両者が同じ枝になるまで繰り返す。

閉回路を形成する内線の列が得られたら，内線の向きに注意が必要だが，第 2 法則に対応する連立方程式を書き下すことは容易である。ここで，調査済みの電池は記録しておく。

・第 1 法則 [KIRCH1]

内線の向き (= 電流の正の向き) に注意すれば問題はない。

・連立方程式を，掃き出し法で解く [RENRI TSU]

(g) 未調査の電池をチェックする [CK, BAT]

未調査の電池の + 極から配線をたどり一極に至るまでの閉回路で電流を計算する。

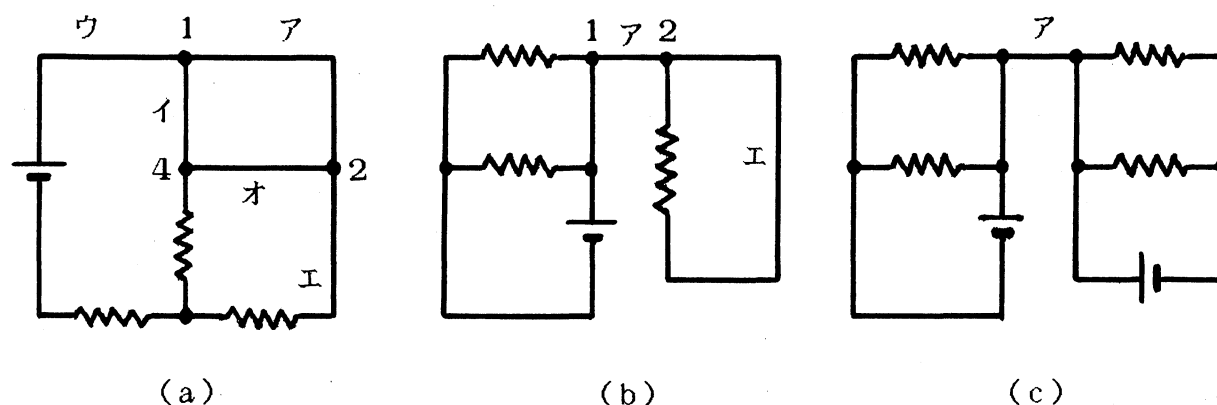


図 3-3 (a)内線ア，イをたどると接続点 1，2，4 が常に等電位となることがわかる。

内線を順次調べ，オに至ったとき，等電位の閉回路ができることがわかるので内線オとその両端の接続点 4，2 を消去する。これに伴い，内線アとエを改めて内線アと名付ける等の変更がなされる。

(b)起電力を含まず，自分自身で閉じる内線も消去される。内線エの消去に伴い内線ア，接続点 1，2 等も消去される。

(c)内線アには電流が流れないので，これを消去し，非連結な回路網 2 つに分解した方が分析は容易になるが，本プログラムではこの処理を行わず，連結な回路網として扱っている。

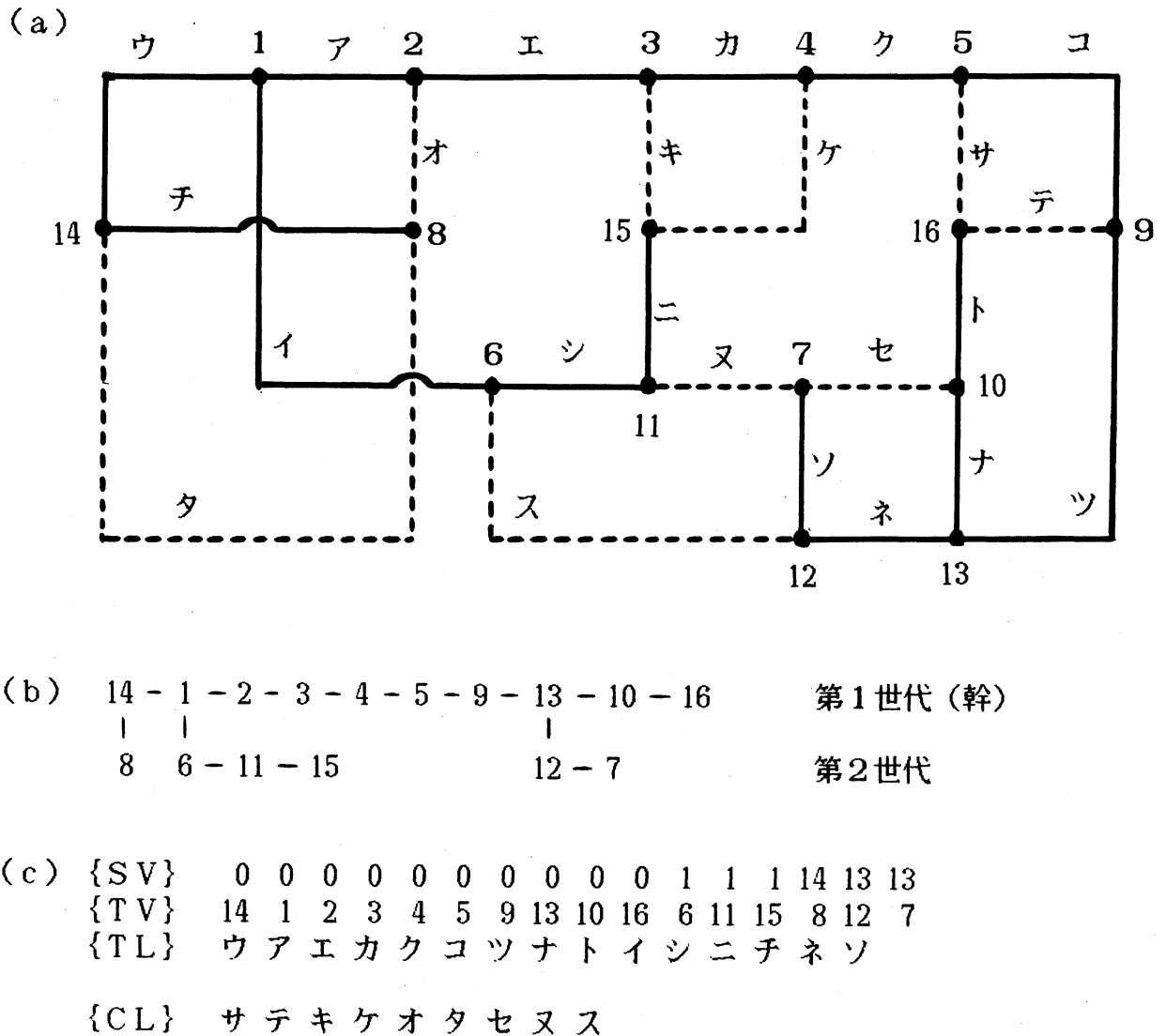


図3-4 (a)接続点16個、内線24本を持つ回路網の例。ここで、実線は生成樹、点線は生成樹に対し閉回路を作る内線 (CL) を表わす。

(b)生成樹における接続点の配置。この例では、幹の外に、同じ世代の枝が3本ある。

(c)図3-5の処理による生成樹とCLの記録法。最初は接続点1が幹の先頭であるが、1からの枝6~11~15が形成された後に、接続点14が幹の先頭に入れられる。

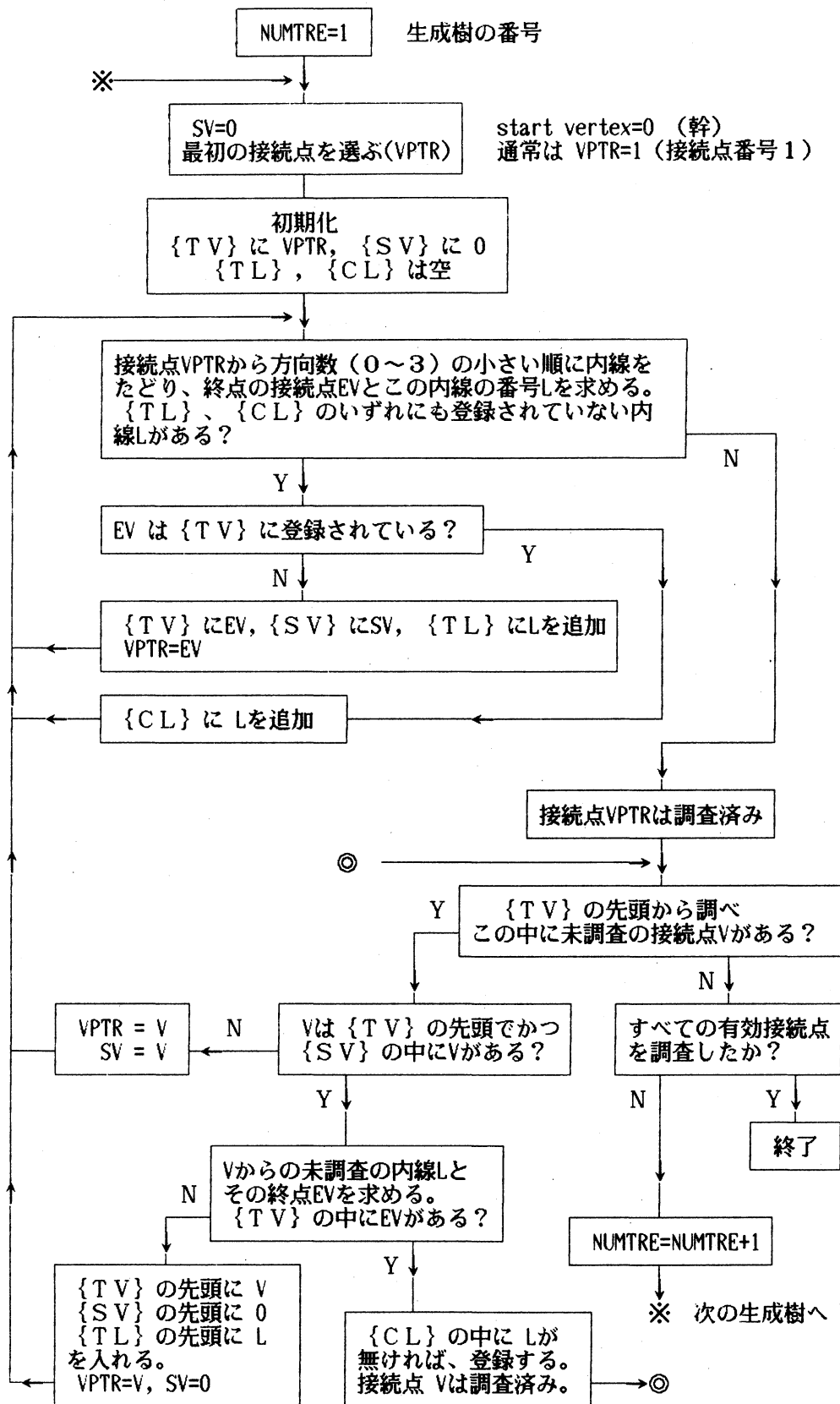


図3-5 生成樹作成のアルゴリズム。生成樹は{SV}, {TV}, {TL}に, CLは{CL}に記録される。これらは, 連結な回路網毎に定義される。

4. 教育利用に向けての問題点

4.1 理科教育におけるシミュレーションの役割

ここでは、パーソナルコンピュータの教育利用のうちシミュレーションについてのみ議論する。

シミュレーションとは実物のまねをすることで、広義には単なるアニメーションまで含まれるが、以下では「条件設定が変えられ、それに応じた結果が得られる」ものを念頭におくことにする。

さて、理科教育におけるシミュレーションを議論するとき大前提となってしまうのは「実際に実験できるものは実験させるのがよい」という点である。事実、学習者各自が実験や観察をする場合には、シミュレーションに比べ桁違いに多くの情報を得ることができる。小学校における電池と電球の例では、

- ・配線の色や形（ねじれ、もつれ等）には無関係なこと
- ・接触不良ということがある。一方、少しでも回路が閉じると瞬間に電球がつくこと
- ・数ボルトの電池の両極に触れても人体にショックは無いこと
- ・傾いた机の上では電池が転がること 等々

いずれもシミュレーションではまねのできない貴重な体験となる。

しかし一方では、教育的実験・観察では何に気付かせるかという目的が明確でなくてはならず、末節にとらわれ本質を見逃すことになることと失敗である。また、何かがわかるというとき、新しい概念が形成されるときには、多数の体験からの抽象化・シンボル化または帰納というステップが必要であるが、学校教育の実験・観察では、時間の制約のため、ごく限られた体験に一般的な結論を押しつけることになりがちである。（単一の実験で「わかる」者は、過去に類似の体験を多数もっており、またその場で多くの思考実験を行なっているのである。）

シミュレーションを有効に利用すれば、当面本質的ではない部分を捨象した対象について、「この場合は、こうなる」というパターンを多数体験させることができ、この多数の体験をもとに、「わかる」段階へ自然に導くことが可能になろう。このようなシミュレーションは、時間や手間の節約という消極的なものではなく、より積極的に教育のあり方を変える可能性を持っている。ただし、抽象化された対象を扱う前には準備が必要なこと、現状ではこの目的のシミュレーション・ソフトウェアが皆無に近いこと、学校教育のスタイルを大幅に変える必要があること等、問題が多くすぐには実現しそうにない。この型のシミュレーションを体験型と呼ぶことにしよう。

すぐに利用できかつ有効と思われるのは、グラフィック機能を用いて、目に見えないものを視覚化してみせることであろう。例えば、大きすぎまたは遅すぎて見えない天体の運動や地球のプレートの動き、小さすぎまたは速すぎて見えない振動や波動、そもそも見えない力やエネルギー、原子・分子の運動や化学反応等。これらについては、プログラミングも容易で、すでに実用に供されている。これらは、演示型シミュレーションと呼んでよいだろう。

演示型は、教室にコンピュータが1台あれば利用できる点でも導入が容易である。各種ビデオ機

器との結合も考えられ、近い将来教育用機器の一種として定着するだろうが、教育方法の大改革にまでは発展しないだろう。

一方の体験型は、現在の学校教育に導入するにはまだ問題が多く残されている。しかし、家庭や社会人教育にはすぐにでもはいり得るので、教育のあり方に大きな波紋を投げかける可能性も考えられる。

4.2 本プログラムの特長と教育利用における問題点

本プログラムの特長と、これを体験型シミュレーションとして学校教育に利用しようとしたときの問題点を考える。

まず特長が即ち問題点となる点を列挙しよう。

(a) 制限が少ない

電池・電球の数、接続点の数、作図領域、等の制限はあるが、配線・配置の組合せは全く自由である。このため、学校では、学習者が複雑すぎる回路を作ったり、教師にもわからない回路ができることもしばしばあるだろう。この点は通常の実験でも同様で、教師の適切な指導が必要である。また、これも実験と同様、発見的学習も可能となる。

(b) 対象を限定していない

キー操作は簡単なので、小学生から大人まで利用できる。

(c) 構成力が必要とされる

何も無い画面に自力で回路図を作るには、かなりの構成力が要求される。[2]の練習は配線のみを行なう「ドリル型」であるが、それでも解答は選択型ではなく構成型である。

要するにこのプログラム自身には、教育上の目標が最初から固定的には設定されていないので、特定の目的に使用する場合には、教師の適切な指導・工夫を必要とする。

改善すべき点としては、次の点がある。

(d) KR情報が不足なので、これを充実する

このプログラムでは、使用者が作成した回路図、その結果である各配線の電流がすべて記録されているので、明らかに無意味な配線や間違った配線に対しては、注意を促すメッセージを付けるよう改良することができる。

また、対象を高等学校生以上に限れば、配線をしながら電池・電球を配置したり、起電力・抵抗値を入力できるようにすることもできる。

現在いくつかの学校で試用中なので、その結果をもとに改版を行なう予定である。

参考文献

- 1) 学習用プログラムの分類については、例えば、
文部省社会教育審議会教育メディア分科会報告「教育用ソフトウェアの開発指針」
1985年12月11日 (「NEW教育とマイコン」No5, 1986年2月号, 学研 一部掲載)
また, C A I 一般についての用語・概念の解説, 実例報告については,
雨宮正彦「教育はコンピュータを必要とするか」エム・アイ・エー
産業能率短大教育工学センター「C A I のすべて」産業能率短大
等が参考になる。
- 2) G. Kirchhoff, Ann. Phys. Chem. (1847), 497.
- 3) F. Harary, "Graph Theory", Addison-Wesley (1968),
(池田訳「グラフ理論」, 共立出版 (1971))