

プログラム著作権の諸問題

——プログラム開発者の財産的利益と社会的利益の適切なバランスを求めて——

目次

- 一 プログラム著作権の基本的視点
 - 二 プログラムの表現の同一性の範囲（プログラムの保護範囲と侵害基準）
 - 二―一 現状
 - 二―二 解釈の明確化の指針
 - 三 プログラムに含まれる情報へのアクセス
 - 四 まとめ
- 一 プログラム著作権の基本的視点

高度情報化社会といわれる今日、社会基盤の情報化が、情報処理技術と情報通信技術の飛躍的發展をベースに進展しつつあり、これが同時に広範な市場を提供している。そして、こうした社会の基盤的な情報化という状況を背

景に、コンピュータ・システムにおいて情報処理や通信を司り、コンピュータにその処理の手順を指示するコンピュータ・プログラムの市場における重要性が認識されつつあるとともに、その模倣の容易さの故に、安易なただ乗りを誘発する傾向があることも否めない。こうした中でコンピュータ・プログラムの健全な市場の形成と維持のために、国際的にはプログラムを著作権法により保護することが趨勢となっている⁽¹⁾。一方、わが国ではかつてコンピュータ・プログラムの経済財としての特質に鑑み、独自立法 (sui generis) による保護が強く主張された経緯があったが⁽²⁾、結局は昭和六〇年の著作権法の一部改正 (昭和六〇年法律第六二号、昭和六一年一月一日施行) によりコンピュータ・プログラムを「プログラムの著作物」として著作権法により保護することが立法上明確にされた⁽³⁾。こうして、コンピュータ・プログラムは著作権法により保護されることには疑いを挟む余地はないが、その適正な保護を考えるのに際して、次のような基本的視点が念頭に置かれるべきであると考ええる。

まず第一点として、コンピュータ・プログラムは、コンピュータに対する指令をプログラム言語で記述する「表現」としての側面があるとともに、そこに記述された処理の手順に従ってコンピュータを稼働させるという「機能」を有しているということである。実際プログラムは、コンピュータ内で稼働している時には、電気的には回路の開閉の状態を表しているにすぎず、またプログラムはそれ自体技術製品であることから、一面ではハイブリッドな機能作品 (work of function) といわれる⁽⁴⁾。従って、コンピュータ・プログラムに余り広範な保護を付与することになると、それに応じて広範な機能及びその機能を実現する技術的手法の保護に至りかねず、またコンピュータ・プログラムにあっては、その機能作品としての特性の故に著作権法により保護される「表現」と保護されない

「アイデア」の区別は實際上非常に困難であり、これがために著作権法がプログラムの先行開発者に有利に働き、その者が類似プログラムを開発した後発者に対して権利侵害の主張を申し立て、この後発者を事実上市場より排除し、プログラムに含まれる技術的手法を独占するための武器を与えうる危険性があることには注意を払うべきである。

次に第二点として、コンピュータ・プログラムが従来⁽⁵⁾の著作物と大きく異なるのは、その本質が「技術」にあり、経済財・産業的所産としての性質を有するという点である。この点、従来⁽⁵⁾の小説、音楽や絵画等の著作物が「多様性」の追求の上に成り立っているため（伝統的著作物の遠心性）、これらについて広い保護範囲を設定しても特になんらの障害が生じることもなかった。これに対し「技術」の特性としては、まず①その性質上、先人の積み重ねに関わる既存の社会共通の技術的基盤に立脚して、個々の技術者が付加的な貢献をつけ加え、これにより社会的技術的共通基盤のより一層の拡大と、より高いレベルへの移行を経て、累積的に発展するものであり（技術の累積的発展性）、且つ、②「多様性」よりもむしろ「合理性」の追求を通じて、より良い技術へと収斂する傾向が強いことである（技術の球心性）。従って、こうした技術の産物に対して容易に独占的かつ広範な保護を与え、類似性の故に安易に侵害を認定し、また技術の進歩と発展に不可欠なプログラムに含まれた技術情報に対するアクセスを認めないような法律解釈をするということになると、技術競争を通じて技術の発展と進歩を阻害し、ひいては社会一般がより良い技術の恩恵に浴することを妨げる結果となる。そこで、その本質がこうした技術にあるコンピュータ・プログラムを、著作権法という主としてコピーから著作者の利益を保護する法制により保護することになった

根拠としては、プログラムの開発に多大な知的労力と投資がかかる反面、他の技術とは異なり模倣やただ乗りが容易になされうることと求め、且つこれを指針として、技術の発展と進歩を阻害することがないように、適切な保護範囲の探求がなされるべきである。

さらに第三点としては、情報処理技術と情報通信技術の結合により進行しつつある高度情報化社会にあっては、コンピュータは単体としてではなく複雑なシステムとして現われ、その中においてコンピュータ・システムを構成するハードウェア及びソフトウェア等の諸要素⁽⁶⁾の間においては、互換性(compatibility)、相互接続性(interconnection)が保証され、これを通じて次のようなユーザの利便性が確保されることが要請されるといえる。すなわち、①ユーザが用途に合わせて最新・最適のハードウェアを選択できること、②かつユーザが異なるメーカーの機種上でも多くの同一のプログラムを走らせることができ、このことがプログラムを供給する側に一層の競争効果を生じさせ、またその跳ね返りとしてユーザが一層安価で最適のプログラムをより広範な範囲から選択できるようになること、③ユーザが以前のものと異なるメーカーのシステムに乗り変えた時でも、既存のユーザ資産を受け継ぐことが出来ること、④複数の異なるベンダー、すなわちマルチベンダーによるコンピュータ・システムやネットワークを構築し、しかもシステムの広域化に対処できること。ただこれらは将来的にはコンピュータ・システムに関する技術仕様の「規格化」ないしは「標準化」(standardization)によりある程度達成されることが期待されるが、⁽⁷⁾現在ではそうした仕様の優劣は、差し当たっては自由競争による改良と淘汰に委ねられるべき点多くあることは否定できず、そうした中で、当面システム構成要素間の互換性や相互接続性等がユーザには保証されない状況

が多々存在することが考えられる。しかし、その場合であっても局部的に互換性や相互接続性を達成するための手だてを法的に保証することが、プログラムの適正な法的保護を議論するに当たっても考慮されるべきであり、且つそうすることがコンピュータ関連市場の競争を刺激し、また標準化を促す要因たりえるものと考ええる。

そこで以上の諸点を踏まえ、プログラムを著作権法により保護する場合に、プログラム開発者の利益と技術の進歩と発展に対する社会的利益のバランスを図るといふ観点から、プログラムの表現の同一性の範囲及びプログラムに含まれる技術情報に対するアクセスの可否について一定の解釈論的指針を示したいと思う。

(1) 一九七三年にはフィリピンがいち早くもコンピュータ・プログラムを著作権法によって保護することに乗り出したのを始め、米国も一九七六年の著作権法の全面改正及びこれに続く一九八〇年の著作権法の一部改正を通じてコンピュータ・プログラムを著作物として保護することを明確にしているほか、一九八三年にはハンガリーが、一九八四年にはインド及びオーストラリア、一九八五年に入って台湾、日本、西ドイツ、フランス及びイギリス、そして一九八六年には特別著作権法としての「コンピュータ・プログラム保護法」により韓国が、また一九八七年には著作権法の全面改正によりシンガポール及びマレーシアが、それぞれ次々とコンピュータ・プログラムを著作権法により保護することを明確に打ち出している。さらにヨーロッパでは、一九八八年にはイギリスが再度著作権法を改正することによってコンピュータ・プログラムの保護強化を図っているのを見るとともに、ECでは一九九二年の市場統合に向けて、著作権法の分野に置ける各国の立法のハーモニゼーションの動きが開始され、そうした動きの中で一九八八年に公表されたEC委員会のグリーン・ペーパー (Green Paper on Copyright and the Challenge of Technology-Copyright Issues Requiring Immediate Action, Commission of the European Communities, COM (88) 172 final, Brussels, 7, June 1988) の中でコンピュータ・プログラムの保護

が再検討されるところに、一九八九年四月にはこのグリーン・ペーパーに対して寄せられたさまざまな意見をもとに、EC 各国の著作権法によるコンピュータ・プログラム保護の統一を図るEC閣僚理事会指令のためのEC委員会案が発表されてゆく (Proposal for a Council Directive on the legal protection of computer program, Commission, COM (88) 816 final, SYN 186, Official Journal of the European Communities, 12, April 1989, No. C 91/4)。その他コンピュータ・プログラムを著作権法によって保護する国々としてチリ、ドミニカ共和国やスペイン等が挙げられる。

(2) 昭和四七年五月通産省重工業局ソフトウェア法的保護調査委員会中間報告、昭和五八年十二月産業構造審議会情報産業部会中間答申「ソフトウェアの法的保護の確立をめざして」——プログラム権法(仮称)の提案——、中山信弘「コンピュータ・ソフトウェアと著作権」ジュリスト七七八号四一頁、植松宏嘉「コンピュータ・ソフトウェアの法的保護と著作権法(七)」、(中)、(下)「NBL二九〇号一二頁、二九二号三四頁、二九三号三〇頁。特に学説が挙げる理由としては、ソフトウェアの保護の理由は、その開発に投下した資本の回収を容易にし、流通を促進して重複投資を防ぐためにあるところ、著作権法は、作者の気持ち、感情、良心等の保護という観点为基础に横たわっており、そこには著作物の流通を促進し重複投資を防止するという産業政策的契機はほとんど含まれておらず、またソフトウェアに著作権を適用したときには、人格権等の諸点につき種々の難点が生じうるとし、また実用一点張りの実用品であるコンピュータ・プログラムを著作物と解することを疑問とし、著作権法を適用することは妥当でなくむしろ独自立法によるべしとする。

(3) これは昭和四八年(一九七二年)年六月の文化庁の著作権審議会第二小委員会(コンピュータ関係)の報告書、および昭和五九(一九八四年)年一月の著作権審議会第六小委員会(コンピュータ・ソフトウェア関係)中間報告において、「プログラムの多くは、いくつかの命令の組み合わせ方にプログラム作成者の学術的思想が表現され、かつその組み合わせ方および組み合わせの表現はプログラム作成者によって个性的な相違があるので、プログラムは著作権法二条一項一号にいう『思想を創作的に表現したもので、学術の範囲に属するもの』として著作物に該当する」という見解に従ったものであり、

かつまた既に出されていたいくつかのビデオ・ゲーム・ソフトに関連する一連の判例にも沿ったものであった（東京地裁昭和五七年二月六日判決無体例集十四卷三号七九六頁、判例時報一〇六〇号一八頁、判例タイムズ四八二号六五頁、スペース・インベード・パートⅡ事件・横浜地裁昭和五八年三月三〇日判決判例時報一〇八一号一二五頁、判例タイムズ五〇六号二〇一頁、スペース・インベード事件・大阪地裁昭和五九年一月二六日判決無体集十六卷一号二六頁、判例時報一一〇六号一三四頁、判例タイムズ五三三六号四五〇頁、STRATEGY・X事件）。これは、ビデオ・ゲームをアセンブラ言語で表現したコンピュータ・プログラムは著作権法にいう著作物に当たるとし、そのプログラムの複製物であるROMに収納されたオブジェクト・プログラム（アセンブラ言語で記述されたプログラムを機械に可読な形態に転換したもの）を他のROMに無許諾で収納することは、著作物であるプログラムの複製物から更に複製物を作成したことに当たり、プログラムの有形的複製として複製に該当すると述べる。これらの判例以前には、ビデオ・ゲームの不正コピー事件の横行に対して、ゲームの受像機に映し出される映像及びゲームの進行に応じた映像の変化の様態が商品表示に当たるとして、不正競争防止法第一条第一項第一号により保護を図る東京地裁昭和五七年九月二七日判決（判例タイムズ四七七号八二頁）や大阪地裁昭和五八年三月三〇日判決（判例タイムズ四九五号一九六頁）があった。また、前記東京地裁昭和五七年二月七日判決等コンピュータ・プログラムの著作物性を認める一連の判例以後、これに次いで、ビデオ・ゲームの映像につき、それを生成するコンピュータ・プログラムの著作物性とは別に、「映画の著作物」に該当するとし、プログラムが無断複製されたビデオ・ゲーム機を喫茶店などに設置して映像を上映させることは「上映権」の侵害となるとする東京地裁昭和五九年九月二八日判決（無体集十六卷三号六七六頁、判例時報一一二九号一二〇頁、判例タイムズ五三四号二四六頁、バックマン事件）や、さらにはビデオ・ゲームのプログラムと映像の双方について著作物性を認める東京地裁昭和六〇年三月八日判決（判例タイムズ五六一号一六九頁、ディグダグ事件）や東京地裁昭和六〇年六月一日判決（判例タイムズ五六七号二七三頁、トッププレーサー事件）も見られるようになった。

こうしたコンピュータ・プログラムを著作権法により保護することに対して賛意を表する学説として、例えば、紋谷暢男「コンピュータ・プログラムの著作物性・保護」NBL二七五号六頁、同「コンピュータ・ソフトウェアの保護」法律時報五五巻七号一五頁、高石義一「ソフトウェアの法的保護」ジュリスト七八四号十九頁、三木茂「コンピュータ・プログラムの著作物性」NBL二八八号四三頁参照。これらの学説は、著作権法が保護するを目的とする文化的所産とは、今日では広く文化的現象すなわち生活形式の総体を意味し、従ってコンピュータ・プログラムを含む産業的所産をも包含する趣旨であり、またコンピュータ・プログラムが実用性を有するものであるとしても、中立的・没価値的な著作権法の立場からすれば、それが思想・感情の表現であり、文芸、学術、美術又は音楽の範疇に属するものとされる限りは保護を否定されないとされ、またコンピュータ・プログラムが著作権法により保護されれば、国際的著作権条約を通じてその国際的な保護が図られるという利点をも指摘して、積極的にコンピュータ・プログラムの著作権法による保護を支持した。

- (4) Intellectual Property Right in an Age of Electronics and Information, Office of Technology Assessment, Washington DC, U. S. Governmental Printing Office, April 1986, (Chapter 3). ハジビハ、エレクトロニクスと情報の時代における知的創作的成果の保護のあり方を検討するため「機能作品」(work of function)、「事実作品」(work of fact)及び「芸術作品」(work of art)という分析概念を提示している。その中で、コンピュータ・プログラムを機能作品品として捉え、その伝統的な著作権法のもとにおけるその保護につきさまざまな問題点を指摘し、その適切な保護のために将来取りうべきいくつかの政策的選択肢を提示している。

- (5) 中山信弘「ソフトウェアの法的保護」(新版)「有斐閣九六頁は、小説・絵画・音楽等においては、他人の作品と類似させなければならぬ必然性は全くないし、むしろ他人との差異を設けることに価値が見出されることが通常であり、これらの従来型の著作物は多様性の世界と言えるのに対して、技術というものは、効果や効率といった合理性の追求であり、自ずとある方向に収斂される傾向があると、また技術の本質は過去からの積上げにあり、ごく少数の画期的な大発明は別として、

ほとんどの技術は過去の技術に改良を加えて新技術とし、さらに次の人がそれに再改良を加えて新々技術とすることによって発展してきたと述べる(同書一二八頁)。また、Paul Goldstein, *Infringement of Copyright in Computer Program*, UFITTA 104 (1987) p. 69 (p. 72) は、科学 (science) と技術 (technology) は「球心的 (centripetal) であり、単一の最適結果 (a single optimal result) に向かうものである」とし、これに対して文芸 (literature) と芸術 (arts) は「遠心的 (centrifugal) であり、異なった趣向の広範囲の視聴者 (a wide variety of audiences with different tastes) を目標とする」とする。その Note, *Copyright Infringement of Computer Program: A Modification of the Substantial Similarity Test*, 68 *Minn. L. Rev.* p. 1264 (p. 1291-94) は「コンピュータ・ソフトウェア産業においては、各革新は改良製品を造り出すために過去の革新に依拠し、踏み石的な改良過程 (stepping-stone improvement process) により発展し、僅かながらの革新が累積することによって重大な意義のある技術的成長を生み出すものである」と述べる。

(6) 今日広義の意味でのコンピュータ情報処理システムの構成は概略以下のような要素からなるといえよう。

(A) ハードウェア資源——コンピュータ装置そのものを指すいわゆるハードウェアのほか、補助記憶装置、周辺・端末入出力装置、入力力チャネルおよび通信回線やネットワーク等を含む。

(B) ソフトウェア資源——①コンピュータ・プログラムとしてのソフトウェア。これには、①コンピュータ・システムのさまざまな資源の有効の利用を図るために設けられ、他のプログラムの実行とハードウェア資源の状態を監視・制御するプログラムとしてのオペレーティング・システム・プログラム(いわゆるOS)、②プログラミングしやすく自然言語に近い人言語で書かれたソース・プログラムを全体として一挙にコンピュータの実行できる形式である機械語に転換するコンパイラーや、ソース・プログラムの命令を逐次機械語に翻訳し実行するインタープリター等、ソースプログラムの翻訳の助けとなるプログラムたる言語処理プログラム、③ユーザのコンピュータ利用上の便宜を図るための各種のユーティリティ・プログラム、④統計、経営科学、数値解析など高度で複雑なデータ処理を、必要とされる特定の技法によって行うため、コンピ

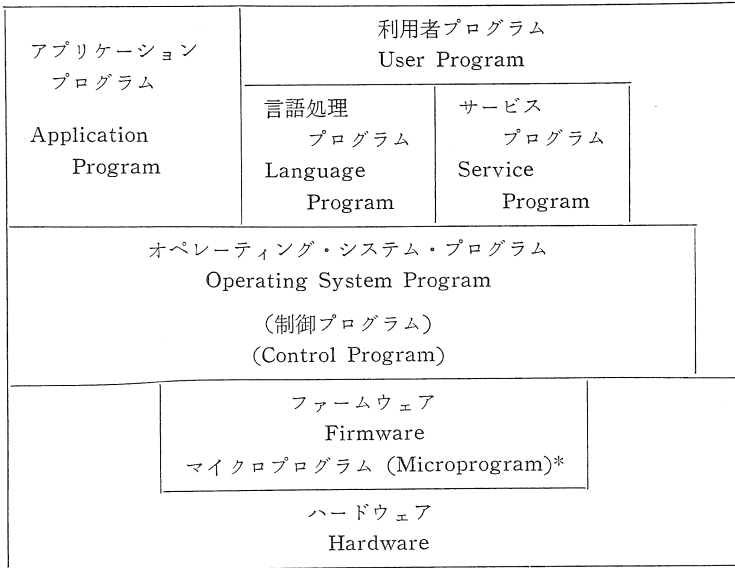
ユーザ・メーカーやソフトハウスがユーザに提供するアプリケーション・プログラム、さらには⑤ユーザ自身が作成し実行するユーザ・プログラム等が含まれる。(以上のプログラムの体系的構成については図一参照)。(二)ドキュメンテーションとしてのソフトウェア。プログラムの開発の過程で作られ、またプログラムの効率的な運用・改良のために必要な諸文書という。これには一般に、①プログラム設計書、フローチャート等のプログラム記述といわれるものと、②ユーザマニュアル等の附属資料とに分けられる。

(C) 人的資源——コンピュータの実際の運用やコンピュータによる情報処理システムの開発に携わる人々であるシステム運用管理者、システム・エンジニア、パンチャー、オペレーター、ハードウェア技術者等。そしてこれらの人的資源が上記諸資源を取り囲み、これら全体が有機的にうまく結び付き互に関係し合うことによって、効率のよいコンピュータ情報処理システムの運用が可能となっているのである。本稿でコンピュータ・システムの構成要素という場合は、(A)のハードウェア資源、および(B)のソフトウェア資源のなかでもコンピュータ・プログラムとしてのソフトウェアを指すものとする。プログラムやプログラム言語については、また注18を参照。

図1のマイクロプログラムについて——

* コンピュータ・プログラムは、コンピュータに実行可能な機械語(オブジェクト・コード)の形態のものが最終的にコンピュータ内の主記憶装置に格納され、中央処理装置(CPU; Central Processing Unit)に取り出されて逐次解読・実行されるが、この場合CPUは、解読した機械語の命令を処理するために諸種の微細な作動をなすように機械に働きかける。この一連の操作サイクルを命令サイクルという。これには機械語の命令をコンピュータ内の主記憶装置の中の一定のアドレス(番地)から取り出すサイクル(命令取り出しサイクル)と、その命令を解読し、それに対応する一定の操作を行う(命令実行サイクル)に分かれる。そして一つの機械語命令に対するサイクルは、コンピュータ内のクロック(時計)に合わせて、

—ハードウェア上のプログラムの階層的構造—



プログラム著作権の諸問題 辰巳

— 図 1 —

コンピュータのハードウェア内の各種のゲートの開閉等によって行われる一連の基本的操作によって構成され、このような基本的な各操作はマイクロ動作といわれる。ところで、従来このマイクロ動作の実行は論理回路の結線によって行われ、これをワイアード・ロジック (wired logic) と呼ぶ。これに対して最近では、各機械語命令に対する一連のマイクロ動作の内容を、各マイクロ動作に対応する二進数の電子信号の命令 (マイクロコード又はマイクロ命令) を組み合わせてマイクロプログラムとし、これを制御記憶という特殊なコンピュータ内の記憶装置に格納し、これを順次取り出してゲートを制御する方法が取られることがある。これをマイクロプログラムミング方式といい、この場合に用いられるマイクロプログラムの基本的な動作を制御しているという意味でハードウェアとソフトウェアとの中間的な存在とみることができ、これを特にファームウェア (firmware) ということがある。この著作物性については問題が多いが (中山前掲三三〇三七頁。植松宏嘉「プログラム著作権 Q & A (新版)」金融財政事情研究会六一頁)、米国ではマイクロプログラムについての著作

物性を肯定した判例がある (NEC Corporation and NEC Electronics Inc. v. INTEL Corp., No. C-84-20799-WPG)。本事件については R. Steinberg, *NEC v. INTEL: THE BATTLE OVER COPYRIGHT PROTECTION FOR MICROCODE*, *Jurimetrics*, Vol. 27, No. 2 (Winter 1987), p. 173. また R. Steinberg, *MICROCODE-IDEA OR EXPRESSION, COMPUTER/LAW JOURNAL* Vol. 9 (1989), p. 61 及び JOHN R. HARRIS, *LEGAL PROTECTION FOR MICROCODE AND BEYOND: A DISCUSSION OF THE APPLICABILITY OF THE SEMICONDUCTOR CHIP PROTECTION ACT AND THE COPYRIGHT LAWS TO MICROCODE, COMPUTER/LAW JOURNAL* Vol. 6 (1985), p. 187 参照。

(7) 標準化については、特に国際標準化機構 (ISO; International Organization for Standardization) と国際電信電話諮問委員会 (CCITT; Comité Consultatif International Télégraphique et Téléphonique=International Telegraph and Telephone Consultative Committee) の公的機関が共同で、メーカーの異なるコンピュータや端末等の異機種・異システム間の通信を可能とするため、通信プロトコルの標準化による推進している OSI (開放型システム間相互接続; Open Systems Interconnection) といわれるネットワークアーキテクチャがあり、また OSI については、企業グループではあるが X/OPEN や OSF (Open System Foundation) 等が UNIX の標準仕様の作成のための作業を進めている。

二 プログラムの表現の同一性の範囲 (プログラムの保護範囲と侵害基準)

二一、現状

わが国の現行著作権法はコンピュータ・プログラムを「電子計算機を機能させて一の結果を得ることができ、且つこれに対する指令を組み合わせたものとして表現したものをいう」と定義し (同法第二条第一項第一〇号の

二)、さらに「プログラムの著作物」(同法第一〇条第一項第九号)という新たな著作物の類型を設けてプログラムを保護することにしており、先に述べたようにコンピュータ・プログラムの著作権法による保護については全く疑いを挟む余地はない。⁽⁸⁾ また著作権法の一般原則として、①表現/アイデア(ないしは形式/内容)という二分法(dichotomy)に立ち、著作物の表現を保護し、そのアイデア(内容)を保護するものではないこと、⁽⁹⁾ ②著作権法による保護は独立創作を排除せず、独立して創作された二つの表現が結果的に同一であっても、その何れか一方が他方を排除して、その保護が否定されることはないこと、⁽¹⁰⁾ さらに③表現とアイデアが融合ないしは混同(merger)する場合には、その表現に保護を与えることはとりもなおさずアイデアの独占を与える結果になるので保護が与えられないこと(マージャー理論: merger doctrine)⁽¹¹⁾等は、一般に認められ又わが国において承認されているところでもあろう。

しかし著作権法の保護する表現とは如何なる範囲又は抽象度のものをいうのかについては必ずしも明確であるとはいえず、ことコンピュータ・プログラムにあってはいっそう不明確である。この点、コンピュータ・プログラムを「文芸の著作物」(literary work)と捉える米国においては、一九八五年頃を境として、それまではプログラムの外面的なコードの文字通りのコピー(literal copying)を問題にした一連のいわゆる第一世代(the first generation)⁽¹²⁾の判例に続いて、プログラムのコードの文字通りのコピーがない場合でも、ちょうど文学作品等においてその物語の展開やプロットが保護されるように、⁽¹³⁾ コンピュータ・プログラムについても、その外面的表現形式から内容にまで踏み込んで保護を論じる第二世代(the second generation)の判例が出始めた。そしてこれらの判例

の中で特に先例的なものが、プログラムのアイデアを、その果たす抽象的な目的 (purpose) 又は機能 (function) であると捉えた上で、それに不可欠 (necessary) でないものは全て表現に該当するとし、プログラムの外面的なコードのみならず、その全体的な構造、流れや構成 (the overall structure, sequence and organization) —— 所詮プログラムの SSO といわれるもの —— をも広くプログラムの表現であるとする一般論を展開するに及んで、大きな議論を呼んでいる。⁽¹⁴⁾

一方わが国において、これまで判例上ビデオ・ゲーム・ソフトの文字通りのコピーによる著作権侵害が問題になっても、⁽¹⁵⁾ それ以上にプログラムにおいて保護される表現とは如何なる範囲のものを言うのか、実務上これを明らかにするものは皆無と⁽¹⁶⁾ いうていい。特にコンピュータ・プログラムが著作権法によって保護されることになった結果、理論的にはプログラムの翻訳や翻案についての権利を、プログラムの著作者が専有することになったが (著作権法第二七条)、⁽¹⁶⁾ これとの関連で、外面的な表現の類似性がなくても表現内容が類似している場合や言語変換の場合にも侵害が成立し得るのかについては、米国の第二世代に当たる公式判例は未だなく、⁽¹⁷⁾ また著作権法がプログラムの著作物について保護の及ばないものとして「プログラム言語」⁽¹⁸⁾ のほか「規約」⁽¹⁹⁾ 及び「解法」⁽²⁰⁾ を明文上挙げている (著作権法第一〇条第三項) こととの関係においても解釈上議論の余地を多く残している。

もともとわが国では、コンピュータ・プログラムにおける問題解決のための処理の手順は一般にはアルゴリズム⁽²¹⁾ と言われ、これはアイデアであり、著作権法上「解法」に該当するものとして保護されないとされるが、⁽²²⁾ 他方、プログラムというものは、そもそもコンピュータにおける処理の手順をプログラム言語で記述したものであること

を考えると、わが国ではプログラムは、その外面的表現形式に即してその同一性の範囲を画することを意図していると思われる。加えて、わが国の著作権法は「翻案」という概念を認めており、これは著作物の表現に含まれる内面的形式を維持しながらその外面的形式に変更を加えることであるとされているが、この「翻案」がコンピュータ・プログラムの場合にどの範囲まで及ぶかという問題は、結局コンピュータ・プログラムの同一性の範囲の問題と表裏一体の関係にあり、またわが国では先の保護されないと規定されている「解法」との関連を考慮すれば、コンピュータ・プログラムにあっては、理論的には外面的形式にかなり近接したところに内面的形式が認められることになり、その結果翻案の範囲はかなり限定され、従って学術的な著作物一般についていえることでもあるが、基本的な筋・仕組みまで保護する小説の場合等と比較すると、プログラムの著作物の翻案権の及ぶ範囲は比較的狭いものと解されているように思われる。⁽²⁴⁾

一〇条第三項第三号の趣旨であるとの主張もなされてもいる。⁽²⁵⁾

(8) わが国の昭和六〇年の著作権法の一部改正の内容は、概ね次の通りである。①第一〇条第一項第九号にプログラムを「プログラムの著作物」として著作権法上保護される著作物として例示する規定を置くとともに、②第二条第一項第一〇号の二にプログラムの定義を設けた。また③第一〇条第三項において、プログラムの著作物についての著作権法による保護は、プログラムにおける「プログラム言語」、「規約」及び「解法」には及ばない旨明示し、これら各用語について定義を設けた。④第十五条第二項を新設し、プログラムの著作物について、従来の法人著作の要件を緩和し、法人等の名義における公表という要件を外した。⑤第二〇条第二項第三号において、著作人人格権の内の同一性保持権をプログラムの著作物について制

限り、特定の電子計算機においては利用し得ないプログラムの著作物を当該電子計算機において利用し得るようにするため、又はプログラムの著作物を電子計算機においてより効率的に利用し得るようにするための改変を許容した。⑥第四七条の二を新設し、プログラムの著作物の複製物の所有者は、自ら当該著作物を電子計算機において利用するために必要と認められる限度において、第一一三条第二項に該当する場合を除き、当該プログラムの著作物の複製又は翻案（これにより作成された二次的著作物の複製を含む）ができるとし（同条第一項。但し、こうして作成された複製物又は翻案物を目的外に使用し、頒布又は公衆に提示した場合は、第四九条第一項三号及び同条第二項第二号により、著作者の専有する複製又は翻案等の権利に該当する行為を行ったものとみなされ、侵害とされるようになった）、他方では、こうした複製物や翻案物の所有者が、そのいずれかについて滅失以外の事由により所有権を有しなくなった後には、著作権者の別段の意思表示がない限り、その他の複製物や翻案物を保存してはならないものとした（第四七条の二第二項。そして、この規定に違反して保存したものは、第四九条第一項第四号及び同条第二項第三号により、著作者の専有する複製又は翻案等の権利に該当する行為を行ったものとみなされ、侵害とされる）。⑦第一五条第二項の規定を受けた第五三条第三項により、プログラムの著作物について法人著作となる場合は、著作物の保護期間は公表後五〇年、公表されなかった時には創作後五〇年とした。⑧第七六条の二の規定の新設により、プログラムの創作年月日の登録の制度を設け、この規定を受けて昭和六一年五月に「プログラムの著作物に係る登録の特例に関する法律」（昭和六一年法律第六五号）が制定され（昭和六二年四月一日より施行。昭和六一年九月には、この法律の施行のため「プログラムの著作物に係る登録の特例に関する法律施行規則」（昭和六一年文部省令第三五号）が制定されている）、登録されたプログラムの概要を公示し、その開発に関する重複投資の防止と流通の促進を図ることとし、同法に基づき昭和六二年四月一日よりSOFTIC（ソフトウェア情報センター）が指定登録機関として、プログラムの著作物の創作年月日の登録事務を行うことになった。⑨第一一三条第二項により、プログラムの著作物の著作権を侵害する行為によって作成された複製物を、その使用権原を取得した際に情を知っていた場合は、業務上電子計算機において使

用することは、プログラムの著作権を侵害する行為とみなされる旨の規定が設けられた。以上の改正法の内容の解説については、坂東久美子「コンピュータ・プログラムに関する著作権法の一部改正について」『コピーライト』二九二頁、半田正夫・紋谷暢男編「著作権法のノウハウ」〔第三版増補〕有斐閣選書三三三頁。日本のプログラムの著作物の登録に関する外国における紹介としては、Kertheinz Pilny, Die Registrierung von Computer-Software in Japan—Ein neues Gesetz und seine Praxis, GRUR Int. 1988 S. 26ff.

(9) これを一番的確に表現しているのは、米国著作権法第一〇二(b)であろう。この規定は、「如何なる場合においても、オリジナルな著作物に対する保護は、当該著作物において記述、説明、例証又は具現されている形態を問わず、如何なるアイデア、手続、工程、体系、操作方法、構想、原理又は発見にも及ばない」としている。(In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work)。わが国において、著作権法によって保護対象とされる「著作物」とは、「思想又は感情を創作的に表現したものであって、文芸、学術、美術又は音楽の範囲に属するものをいう」(著作権法第一条第一項第一号)とされ、著作権法の保護対象が「表現」であることは明らかである。しかし、半田教授は、著作物は多くの要素によって構成されており、創作過程から眺めてみれば、たとえば、集められた素材、作者のアイデア、素材にアイデアが組み合わされて作者の頭脳に描き出された思想体系、および、この思想体系を外部に発表するために採られた表現形式などが考えられるとし、これらの要素のうち著作権によって保護を必要とするのはいったいかなる部分なのであろうかと問われ、この問題の解決はとりもなおさず著作者に留保され、したがってまた著作権の保護を受ける部分と、著作者から解放され万人の自由利用が許される公有の部分との識別に奉仕するばかりか、著作権侵害の有無を判断するに当たっての一つの重要な基準を提供するものだとされる。そして従来のドイツの学説の表現の形式／内容の二分法の展開を検討された後、最近の学説が、著作物の

「形式」と「内容」の峻別という方法を捨て、それぞれの中に著作者の独創的な個性の顕現を求め、これを著作物の本質と解して、著作権法上の保護資格を認める傾向にあるとされる。そして、こうした著作物の本質の捉え方は、従来ともすれば見失いがちであったことからの本質を再認識させる点で意義を認められつつ、その場合でも著作物ごとに、著作者の個性が認められ、したがって著作権によって保護される部分とは何か、また万人が自由利用が認められる公有の部分とは何か、検討しなければならぬであろうと述べられる(半田正夫「著作権法概説〔第四版〕」一粒社七九〜八四頁)。この見解は、ドイツの Ulmer の提唱に従い、従来からの形式(Form)と内容(Inhalt)との区分の代わりに、著作物の個人的な特徴(Den individuellen Zügen des Werkes)とそれに含まれてゐる公有物(denn in ihm enthaltenen Gemeingut)の区分により著作権保護の対象を考へる立場に立脚したものであり、注目に値する(Eugen Ulmer, Urheber- und Verlagsrecht, 3. Aufl. 1980, S. 119ff)。またこういふ見地に立って著作物についての著作権保護を再考してみると、表現であっても俳句等は保護されるが、スローガンや交通標語等は保護されないとされるのは何故か、また逆に小説等のストーリーやプロット等は内面的表現形式といわれ、実際は人間の視聴覚による感性では把握できないそれ自体は観念的なものであるにも拘らず、著作権法により保護され、それを他人がかってに利用して外面的表現形式を変えただけ(例えば小説のストーリーを映画化するだけ)では翻案による二次的著作物の作成に該当し、当初の著作者の権利が及ぶとされるが、それは何故かと考えさせられるものがある。そして、これにつき精神的成果を創作した著作者に、どの程度の保護をどのような素材に対して与えるかという著作権法のポリシーを探って見れば、結局そこには、個性的な知的精神的努力をした著作者に対して報い、新たな創作を奨励するという要請と、著作物に含まれる一定の要素に自由利用を認めることによる公共的利益との均衡の上に立って、著作権法により保護される対象や素材の決定がなされているのではないかということを思い至らしめる。

(10) この原則がわが国において、判例上実際の問題になった事例として、ワン・レイニー・ナイト・イン・トーキョー事件

がある（最高裁昭和五三年九月七日第一小法廷判決民集三二卷六号一一四五頁、判時九〇六号三八頁）。この事件は被告 Y_1 「ワン・レイニー・ナイト・イン・トーキョー」という楽曲を作成、公表し、音楽出版社 Y_2 に当該楽曲のレコード製作販売を許諾し、その結果そのレコードが市販されるようになったが、この曲のある小節の六割に当たたる部分が、ハリー・ウォーレン作曲の「ザ・ブルーバード・オブ・プロクン・ドリームズ」という曲のリフレインに当たたる部分と同一又は類似の旋律を使用しているので、「ワン・レイニー・ナイト・イン・トーキョー」は「ザ・ブルーバード・オブ・プロクン・ドリームズ」の複製物であるとして、後者曲の日本国内における著作権を一定の期限付きで譲渡されて管理していた音楽出版社 X が原告となつて Y_1 、 Y_2 に対して著作権侵害を申し立て、「ワン・レイニー・ナイト・イン・トーキョー」の著作権不存在確認及び損害賠償を請求したものである。最高裁は、無断で著作物を複製するときは偽作者として著作権侵害の責に任じなければならぬとしながら、「ここでいう著作物の複製とは、既存の著作物に依拠して再製されたものでないときは、その複製をしうるものを再製することをいうと解すべきであるから、既存の著作物に依拠して再製されたものではないときは、その複製をしたことにはあたらず、著作物侵害の問題を生ずる余地はないところ、既存の著作物に接する機会がなく、従つて、その存在内容を知らなかった者は、これを知らなかったことに過失があるか否とにかかわらず、既存の著作物に依拠した作品を再製するに由ないものであるから、既存の著作物と同一性のある作品を形成しても、これにより著作権侵害の責に任じなければならぬものではない。」とし、 Y_1 が「ザ・ブルーバード・オブ・プロクン・ドリームズ」の存在を知つて「ワン・レイニー・ナイト・イン・トーキョー」を作成したとは認められずとして X の控訴を認めなかつた原審（東京高裁昭和四九年一月二四日判決判夕三一九号一七八頁）の判断を支持した。判例評釈としては、齊藤博「同一性のある著作物の存在の不知と著作権侵害」民商法雑誌八一巻二号二二四頁。半田正夫「他人の著作物の不知と著作権侵害——同一性のある作品を作成した場合」昭和五三年重要判例解説（ジュリスト六九三号）二八八頁。東京大学判例研究会「最高裁判所民事判例研究」法学協会雑誌九八巻七号九八八頁。

(11) このパーチャー理論 (merger doctrine) は、もともと米連邦最高裁判所判決 Baker v. Selden 101 U. S. 99 (1879) に由来するものである。この事案は、一定の簿記システム (T-account system) に基づく会計帳簿票とその使い方を説明する書籍を Selden が出版したのを、Baker が同一の簿記システムにより利用出来る帳簿票を含んだ書籍を出版したので、Selden が、Baker は同一の簿記システムを記述した書籍の出版により自らの書籍に関わる著作権を侵害したとして訴えを提起したものである。裁判所は、「本の教示する技術が、その本の説明のために用いられた方法や図表、又はそれに類似するようなものを採用することなくしては利用することが出来ないときには、当該方法や図表はその技術に必然的な随伴物であり、それとともに公有に供されたものと考えられるべきである」(where the art it teaches cannot be used without employing the methods and diagrams used to illustrate the book, or such as are similar to them, such methods and diagrams are to be considered as necessary incidents to the art, and given therewith to the public; ……; 101 U. S. at 103) とした。わが国においては判例はないが、中山前掲一五頁以下参照。

(12) この時期は、おおよそ一九七〇年代の終わりから一九八五年頃迄で、米国ではこの時期においては、コンピュータ・プログラムの著作権侵害を訴えられた被告から、度々、ROM (読み出し専用メモリー: Read Only Memory) に格納されたオブジェクト・プログラムは機械の一部となり著作物とはいえないとか、オペレーティング・システム・プログラム(OS) は、米国著作権法第一〇二(b)によって明示的に著作権法の保護の及ばないものとされている工程 (process)、体系 (system) 又は操作方法 (method of operation) 等であって著作物性が認められないか、そうでなくとも表現とアイデアの融合 (merger) があり保護されるべきではない、という主張がなされたが、これに対して裁判所の判断としては、多少の揺れはあったものの、総じて①コンピュータ・プログラムはソース・プログラムであろうとオブジェクト・プログラムであろうと区別なく著作物性が認められること、また②たとえオブジェクト・プログラムがROMに格納されている場合でも、それが機械の一部になるのではなく、ROMはプログラムという著作物を固定する有形的な表現の媒体に過ぎないこと、③

アプリケーション・プログラムであろうとオペレーティング・システム・プログラムであろうと、区別なく著作物性が認められ、オペレーティング・システム・プログラムだからといって、それと同一の機能を表現する方法が多数ありうるので、表現とアイデアの融合があるとは直ちにいいえないこと等が判例上認められている。特にこれらの論点について総括して判示した判例として Apple Computer, Inc. v. Franklin Computer Corp., 714 F. 2d 1240 (3d Cir. 1983) *rev'g* 545 F. Supp. 812 (E. D. Pa. 1982) がある。また、わが国においては、コンピュータ・プログラムの種類には OS やアプリケーション・プログラムを始め種々の区分はあるものの、それらが一律著作権法によって保護されることについてはほとんど争いはない（著作権審議会第六小委員会（コンピュータ・ソフトウェア関係）中間報告昭和五九年一月三三頁。OS の位置づけについては注 5 さらに注 36 を参照）。ただ、マイクロプログラム（マイクロコード）については、これが著作権法上の「プログラム」の定義に該当し、また著作物といえるのか、著作物といえるとしても一般に創作性が否定されるのではないかという点につき、疑問を呈する向きが日本では強い（中山前掲三三〇～三七頁。植松前掲六一～六三頁は、マイクロ・プログラムは「プログラム」の定義に該当せず、またハードの構造に制約され選択の幅が非常に狭いため、著作権法によって保護するとアイデアの保護になってしまうので著作権法による保護を否定し、むしろ特許によるべきであるとする。マイクロプログラムの技術的位置づけについては注 5 を参照）。またソース・プログラムとオブジェクト・プログラムの関係につき、わが国ではソース・プログラムのオブジェクト・プログラムへの転換は複製に該当するとされ（前掲著作権審議会第六小委員会中間報告三三〇～三四頁。半田前掲一〇〇頁）、わが国の判例も著作権法改正以前から、同一の見解に立って ROM に収納されたオブジェクト・プログラムを取り出し、無権限に他の ROM に収納する行為は、著作物たるソース・プログラムの複製物（オブジェクト・プログラムを納めた ROM）から更に複製物（同一のオブジェクト・プログラムを収納した別の ROM）を作成することに当たるから、ソース・プログラムの著作権（複製権）を侵害するものと構成している（注 3 におけるビデオ・ゲームソフトについての関連諸判例参照）。しかしこれには異論があり、ソース・プログラムからオブジェクト・

プログラムへの転換は言語体系の変換を伴う限り普通の翻訳と同様に「翻訳」であり、二次的著作物の作成であるが、普通の翻訳の場合にはマニュアルでなされ創作行為が伴うのに対して、ソース・プログラムのオブジェクト・プログラムへの転換は機械的で創作行為は伴わないので、翻訳者には独立の権利は発生せず、オブジェクト・プログラムについての権利は全てソース・プログラムの権利者に帰属すべきものと考えるべきであるとする見解も主張されている(中山前掲三〇頁〜三二頁、植松前掲四二〜四四頁)。ただ直接的にオブジェクト・プログラムが作成された場合には、そこに創作行為があるのでそのオブジェクト・プログラムに著作物性があることについては見解の一致がある(前掲著作権審議会第六小委員会中間報告三三〜三四頁、半田前掲一〇〇頁、中山前掲三二頁)。またわが国の著作権法の解釈としては、コンピュータ・プログラムの記録され又は固定される媒体の如何に係わりなく、機械に可読とは言えないコーディング・シートに記述されたプログラムであろうと、機械に可読なパンチ・カードに穿孔されたプログラムであろうと、またROMに格納されたものももちろん、プログラムのプリントアウトや、さらには雑誌に掲載されたプログラムでもプログラムの著作物ないしはその複製物と考えることが出来るように思われる。ただコンピュータ・プログラムを実行する前提として、これをコンピュータ本体の内部記憶装置であるRAMにローディング(蓄積)することが複製に該当し、それ故、プログラムが内部記憶装置にローディングされた状態をプログラムの著作物の複製物と見ることが出来るかどうかについては解釈上争いがあり、昭和四八年の著作権審議会第二小委員会の報告書では「その蓄積は瞬間的かつ過渡的なものであって複製に該当しない」(同報告書二二頁)とし、昭和五九年の著作権審議会第六小委員会中間報告もこれを引用しているが(同中間報告四八〜五〇頁)、同時に複製と解し得るとする意見もあった旨付記しているほか、わが国では正面からあまり議論されていない。しかしROMへのプログラムの収納が複製と見られるならば、これと区別してRAMへのプログラムのローディングを過渡的という理由だけで複製に該当しないと結論づけることはできないとして、積極的にRAMへのローディングを複製とみる立場も表明されている一方(野一色勲「コンピュータにおける複製」著作権研究一六号六八頁)、電子計算機本体の内部記憶装置に蓄積する行為は、

過渡期的で、永続性、反復使用可能性のないものは複製と考えられないが、永続性、反復使用可能性のあるやり方で蓄積すれば複製となるとする見解もある（植松前掲八九頁）。外国では、コンピュータ本体の記憶装置（RAM）への蓄積を複製と見る見解が強いと思われる（マインツは、Kolle, GRUR 1982, S. 455.; Ulmer, Der Urheberschutz wissenschaftlicher Werke unter besonderer Berücksichtigung der Programme elektronischer Rechenanlagen, 1967, S. 19.; Kolle, GRUR 1974, S. 614 (S. 617); Kindermann, GRUR 1983, S. 157; Ulmer/Kolle, GRUR-Int. 1982, S. 499.; Sieber, BB 1983, S. 984.; GRUR-Denkschrift, GRUR 1979, S. 305.; Rupp, GRUR 1985, S. 147ff. また米国は、この「FINAL REPORT OF THE NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, July 31, 1978, pp. 12-13; Nimmer on Copyright § 8. 08.」。

(3) Nimmer on Copyright § 13. 03 [a] [1] 参照。この点、米国においては、二つの著作物が同一性の範囲に入るか否か、すなわち著作物の表現の範囲を決定する方法として、著作物の表現を順次抽象化していくながら比較する Abstraction test が主張されるとともに (Nimmer, p. 13-22)、比較の方法として、それぞれの抽象化の次元において現れるパターンを比較するという Pattern test が用いられる (Nimmer, p. 13-25)。著作権法の保護するマインツの表現 (expression of idea) には、作品の十分に具体的なパターン (sufficiently concrete pattern) が含まれるとする。そして文芸作品にあつては出来事の流れ (sequence of events)、登場人物の展開 (development of characters) や筋 (plot) も保護される表現に含まれるとみる (Nimmer, p. 13-28)。Abstraction test が、*Nimmer* (Judge Learned Hand) による提唱されたのは、Nichols v. Universal Pictures Corp., 45 F. 2d 119 (2d Cir. 1930) においてが始めてであり、ここにおいて次のようなことが述べられている。「如何なる作品についても、そしてとりわけ劇においては、より多くの挿話が取り除かれるにつれて、益々一般的といえる数多くのパターンが等しくうまく当てはまるようになる。最後のものは、たぶん、劇が何についてかの最も一般的な叙述に過ぎないであろうし、そして時としては、その表題に過ぎないこともありえよう。しかしながら、

一連の抽象化の過程においては、もはや保護されないという一点がある。なぜなら、さもなければ、劇作家は自らのアイデア——すなわちその表現とは異なって劇作家の財産権が決して及ぶことのないアイデア——の使用を禁じることができるところになるからである。ただたれも、この境界線を確定することが出来た者はいないし、またたれもそれを行うことが出来なかつてゐる。」(“Upon any work, and especially upon a play, a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may perhaps be no more than the most general statement of what the play is about, and at times might consist only of its title; but there is a point in this series of abstractions where they are no longer protected, since otherwise the playwright could prevent the use of his ideas, to which, apart from their expression, his property is never extended. Nobody has ever been able to fix that boundary and nobody ever can.”) のように、保護される表現を決定するために当初提唱された Abstraction test は、結局は保護される表現の範囲を決定するための「方法」を提唱したのに過ぎず、その「基準」を示すものではなく、結局この方法をとっても保護される表現の範囲の決定は、幾らか恣意的で個別的にならざるを得ないこととなる。そこで、この Abstraction test と Pattern test を接合して、保護される表現の範囲を決定する一応の一般的基準として示されたのが、先の Nimmer の見解であるように思われる。

(4) Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc. 727 F. 2d 1222 (3d Cir. 1986), *aff'd* 607 F. Supp. 1307 (E. D. Pa. 1985). 本件では、原告は、被告が IBM Series I コンピュータ用 EDL 言語 (Event Driven Language) で記述された原告の Dentalab System という歯科ラボラトリーのための在庫管理・貸し方勘定会計プログラムを IBM/PC 用 BASIC 言語で書き変え、Dentcom という商標を売り出したので、Whelan が著作権侵害を理由に訴訟を提起した。これについて一審のペンシルバニア連邦地方裁判所は、たとえコンピュータ・プログラムが異なった命令制御方式を持つ異種のコンピュータに適合させるために変更され、練り直されなければならないとしても、コンピュータ・プ

プログラムに具現したアイデアは著作権法により保護されるとし、また IBM/PC のソースコードが EDL と異なり BASIC で書かれているからといって複製の可能性または著作権侵害を排除するものではないとした。そして裁判所は、被告のプログラムを書いた者は、原告の Dentalab のソースコードにアクセスして、そのシステムを理解し、その全て同一の流れにおける処理 (all the same sequential operation) を利用したに違いないという原告側の鑑定人の意見を採用して著作権侵害を認め、被告に対し損害賠償の命令、プログラムのマーケティング・広告・販売・貸借またはライセンスングの禁止を言い渡した。これに対して被告は原審の判決は著作権法の保護の対象とならないアイデアを保護するものであると主張して控訴したが、第三巡回区控訴審裁判所は、プログラムのアイデアは歯科ラボラトリーの効率的な運用 (efficient administration) であるとし、それに不可欠でないものは全てプログラムの表現に該当するとし、プログラムの構造 (structure) (sequence) 及び構成 (organization) にも保護が及ぶと判断して原審を支持した。また、これより先、SAS Institute, Inc. v. S&H Computer Systems, Inc., 605 F. Supp. 812 (M. D. Tenn. 1985) 事件があるが、この事案においてそもそもその発端は、ナッシュビルにある Vanderbilt 大学の Charles Federspiel という教授が自らの研究のために、本件原告である SAS Institute が開発した IBM コンピュータ又は IBM コンピュータとコンパチブルなコンピュータにおいてのみ稼働可能な統計プログラム (SAS プログラム) に対比出来るプログラムを、自らの大学で利用できる VAX コンピュータ用に開発しようとしたことにある。そこで彼は自らの同僚及び S&H の従業員である者などと共に相談した結果当初 PASQUEL というパッケージを開発し市場に出すことを計画したが、それが困難であるとわかったので計画は放棄された。しかしその後、SAS プログラムを VAX コンピュータ用に転換することが決定され、このために Federspiel らに S&H が加わってテネシー州にパートナーシップ PASS が設立され、これ以降の作業は S&H に請け負われた。そしてこの結果出来たプログラムは INDAS と名づけられたが、これは S&H が、SAS にその意図を秘匿して、SAS からライセンスを受けて引き渡された SAS プログラムのソースコードを、ライセンス契約では禁じられたコンピュータにローディン

グして VAX コンピュータで稼働出来るように手を加えたものであった。そこで SAS が S&H を相手に著作権侵害を理由に INDAS または SAS プログラムの複製若しくは翻案である製品を市場に出すことの差止を求めた。これに対してテネシー州連邦地方裁判所は、審理で証明されたいくつかの形式上の複製を認めたほか、被告のプログラムが SAS プログラムの構成および構造の細部の複製を示している限りにおいて、かのような複製は S&H の製品全体に及ぶものである ([T]o the extent that it represents copying of the organization and structural details of SAS, such copying pervades the entire S&H product) などいふことがあり、また S&H の製品が S&H によるオリジナルな寄与を含まないとしてもそれは SAS に基づく翻案物ともいえるとして原告の勝訴としたのであった。こうして、一般には一九八五年頃を境として米国では、コンピュータ・プログラムの著作権法による保護に関しては、第二世代 (second generation) に入るに至ったといわれるようになった。もっとも、コンピュータの文脈では Whelan v. Jaslow 判決の立場を支持できないとし、たとえその立場に立ったとしても本件プログラムの構造・流れ及び構成は、綿花市場の外在的要因 (externalities) によって決定されているとして保護を否定した第五巡回区控訴審裁判所の Plains Cottonv. Goodpasture Computer 807 F. 2d 1256 (5th Cir. 1987) 事件もある。この事案は端末機を通じ綿花栽培者に対して綿花の市場に応じた綿花売買の価格決定を支援するという原告 Plains Cotton の Telecot というプログラムを、その製作に携わった Plains Cotton の元従業員達が、被告 Goodpasture Computer Service のためにパーソナル・コンピュータ用に作り変え、Goodpasture Computer Service がこれを GEM と名打って販売し出したので、Plains Cotton が自らのプログラムの著作権を侵害するとして訴訟を提起したものである。これに対して第五巡回区控訴審裁判所は、第三巡回区控訴審裁判所の Whelan v. Jaslow 事件判決と対立する先のような判断を下したのであった。また、これより先に Q-CO Industries Inc. v. Sidney Hoffman, Dilip Som and Computer Prompting Corp., 625 F. Supp. 608 (C.D.N.Y. 1985) に基づく原告 Q-CO Industries の元従業員が、テレビスクリーンに台詞を映し出し話手がそれを読みながら台詞を言うことが出来るよう

にした Atari 800-XL コンピュータ用のプロンプター (prompter) のプログラムで、BASIC と部分的にアセンブラ言語で記述された VPS-500 の開発に携わった者が、独立後 IBM/PC 用と PASCAL とアセンブラ言語を用いて同様なプログラム CPC-1000 を開発した事件につき、裁判所は先の SAS v. S&H 事件の判決に従い、コンピュータプログラムの構造や構成の類似性がコピーがあったか否かを決定する上で重要であると認め、本件で原告の主張するモジュールの構成の類似性は、自動車の特別なサスペンション・システムの複雑さよりはむしろ自動車の車の観念にたとえることが出来るとし、未だ被告が「表現」のコピーをしたという証明はなされていないとして被告の責任を否定している。しかしこれらに対して最近では更に Whelan v. Jaslow 判決に依拠し、システム設計レベルでの異なったデザインが可能であること、当事者のプログラムに見られるほとんど同一の構造はプログラムの作業に取って不可欠ではないことを認定し、被告の侵害責任を肯定す Pearl Systems, Inc. v. Competition Electronics, Inc., 8 U.S.P.Q. 2d 1520 (S. D. Fla. 1988) 事件及び原告の C 言語で書かれた医療研究所用プログラムを被告が BASIC に書き換えた事件で、当事者のプログラムはデザイン、データ構造、流れ及び構成を含め、実質的に類似しているとして、やはり被告の著作権侵害を認めたものとして Soft Computer Consultants, Inc., v. Lalehzarzadeh, No. CV 88-0756, slip op. (E.D.N.Y. 1988) 事件がある。米国の動向については、根山敬士「アメリカのコンピュータ・著作権法——判例の動向と分析」工業所有権法九四号一頁。

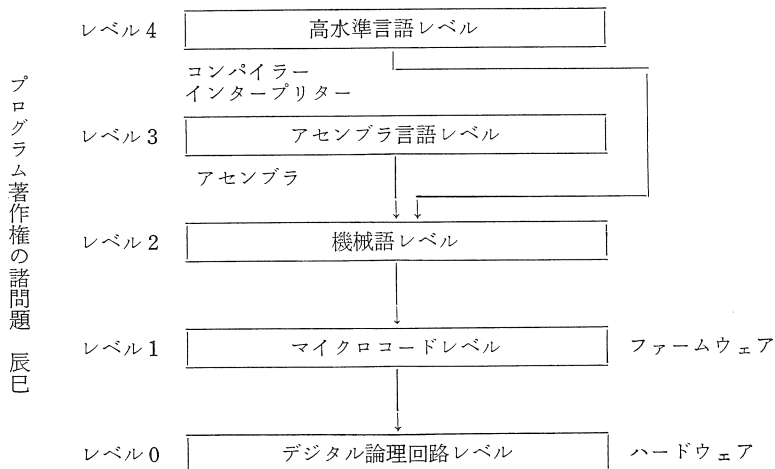
(15) ビデオ・ゲーム・ソフトのコードの文字通りの複製による著作権侵害事件については、注3を参照。

(16) 著作権法は著作者がその創作した著作物について専有する権利として、第二一条に著作物を複製する権利を認めるとともに、第二七条において著作者の改作利用権としての翻訳・翻案権を規定し、「著作者は、その著作物を翻訳し、編曲し、若しくは変形し、又は脚色し、映画化し、その他翻案する権利を専有する」とする。そして著作物を翻訳し、編曲し、若しくは変形し、又は脚色し、映画化し、その他翻案することによって創作された著作物を、著作権法は「二次的著作物」とし（同法第二一条第一号）、第一一条において「二次的著作物に対するこの法律による保護は、その原著作物の著作

の権利に影響を及ぼさない」とする一方、第二八条において、「二次的著作物の原著物の著作権者は、当該二次的著作物の利用に関し、この款に規定する権利で当該二次的著作物の著作権者が有するものと同一の種類の権利を専有する」との旨の規定を設けている。この場合、二次的著作物に著作権が発生し、著作権法上の保護を受けるためには、原著物の著作権者の許諾にもとづき二次的著作物が創作される必要があるか否かに関しては争いがあり、通説は必要としないとするが、必要とする見解も見られる。これについては、半田正夫「著作物の利用形態と権利保護」一粒社一二六頁以下。

- (17) 非公式判例ながら、わが国において、プログラムの「処理の流れ」自体は、アルゴリズム、すなわち著作権法第一〇条第三項第三号に規定されている「解法」であって著作物として保護されないとした裁判例として、東京高裁平成元年六月二〇日決定がある(工業所有権法研究一〇一号一五頁)。本決定については、梶山敬士「知的財産権の潮流二一 プログラム著作権判例の動向——NEC対インテル米国判例とシステム・サイエンス東京高裁決定を中心に」ジュリスト九四五号八〇頁。
- (18) 著作権法第一〇条第三項第一項は、「プログラム言語」を定義して、「プログラムを表現する手段としての文字その他の記号及びその体系をいう」としている。JIS情報処理用語(JIS X-0001-1987, 01.01.12)によれば、プログラム言語(programming language)とは「プログラムの作成又は表現するために設計された人工言語」となっている(JISハンドブック情報処理 用語・コード編 一九八九/五二 日本規格協会一八頁)。プログラム言語のうち、コンピュータが直接実行可能なのはコンピュータ内で電子信号に対応する1と0の機械語(machine language)の形態におけるプログラムで、このプログラムをオブジェクト・プログラム(object program)と称す。これに対し、機械語と一対一の関係にある記号言語としてはアセンブラ言語(assembler language)があり、またより日常言語に近い高水準言語としては、科学計算向き①FORTRAN (FORmular TRANslator) 事務計算向き②COBOL (Common Business Oriented Language) 科学・事務計算両向き③PL/I (Program Language One) 端末用対話型及びビジネス用言語④BASIC (Beginners All purpose Symbolic Instruction Code) 等がある。プログラムは、通常「プログラム言語」として記述され、

—現代の階層的プログラム言語構成—



— 図 2 —

プログラム著作権の諸問題 辰巳

これをソース・プログラムというが、このままではコンピュータにとって実行可能ではないので、このソース・プログラムを機械語の形態でのオブジェクト・プログラムに翻訳転換する必要があるが、アセンブラ言語で記述されたソース・プログラムならば、「アセンブラ」(assembler)という翻訳プログラムにより、その他の高水準言語で記述されたソース・プログラム全体を一旦一括して機械語の形態に翻訳転換し実行する場合には「コンパイラー」(compiler)という翻訳プログラム、また高水準言語で記述されたソース・プログラムの命令を各個逐次に機械語の命令に翻訳転換し実行する場合には「インタープリター」(interpreter)という翻訳プログラムを用いなければならない。

(19) 著作権法第一〇条第三項第二号は、「規約」を定義して、「特定のプログラムにおける前号のプログラム言語の用法についての特別の約束をいう」としている。

(20) 著作権法第一〇条第三項第三号は、「解法」を定義して、「プログラムにおける電子計算機に対する指令の組み合わせの方法をいう」としている。

(21) アルゴリズム (algorithm) とは、JIS 情報処理用語 (JIS X-0001-1987, 01.04.07) では「明確に定義された有限個の規則の集

まりであって、有限回適用することにより問題を解くもの。例… 2×3 を決められた精度まで求める算術的な手順をもれなく記述した文」としている。JISでは、アルゴリズムよりも「算法」という用語を先に挙げている(JISハンドブック情報処理用語・コード編 一九八九/五二 日本規格協会二〇頁)。代表的には、数学的な計算については計算法というものがあるが、数学的な計算に限らずより一般的に、明確な処理規則による問題解決のための手順のまともりは全てアルゴリズムと呼ぶことができる。従って、コンピュータはアルゴリズムによって情報を処理する機械であるということができる。この点アルゴリズムとコンピュータ・プログラムとはどのように異なるのが問題になるが、一応、アルゴリズムとはプログラムをどう組むかという考え方や手順を指し、プログラムとはそれをコンピュータが受け入れられる形式にプログラム言語を用いて具体化したものであるということができよう。従って、アルゴリズムは、問題を解決するための明確な処理規則による手順であり、プログラムの指令の組み方を指示するものであるとともに、一旦プログラムとして組まれた時には、プログラムはアルゴリズムを最も具体的に表現しているものといえる。牧野昇監修 平凡社「現代ハイテク事典」九四頁「アルゴリズム」の項参照。また植松宏嘉「プログラム著作権Q&A」新版「四八頁参照。なお、JIS情報処理用語(JIS X-0001-1987, 01.04.01)によれば、「(計算機)プログラム」(computer program)とは「処理に適した命令の順番付けられた列。注1. 処理には、プログラムの実行及びプログラムを実行する準備としての、アセンブラ、コンパイラ、解釈プログラム又はその他の翻訳プログラムなどの使用も含まれる。2. プログラムには命令及び所要の宣言文が含まれる。」としている(JISハンドブック情報処理 用語・コード編 一九八九/五二 日本規格協会一九頁)。

(22) 中山前掲四七頁。植松前掲四八頁、五〇〜五一頁。昭和五九年一月文化庁著作権審議会第六小委員会(コンピュータ・ソフトウェア関係)中間報告五一頁によれば、プログラムのアルゴリズムは保護しないという基本的な考え方に立つとする。

(23) 中山前掲四七頁及び一一頁。

(24) 前掲著作権審議会第六小委員会(コンピュータ・ソフトウェア関係)中間報告五一頁。

(25) 中山前掲四八頁。前掲著作権審議会第六小委員会(コンピュータ・ソフトウェア関係)中間報告「第三章 現行著作権法におけるコンピュータ・ソフトウェアの保護に関する問題点に対する対応策(提言)」の5は、プログラムの翻案権の及ぶ範囲をより明確化するために、アルゴリズムは保護しないという前提に立ち、必要な措置を講ずることが望ましいと提言しているが、これを受けて現行著作権法第一〇条第三項のプログラムにおける「解法」には保護が及ばないとする規定が設けられたものと思われる。

二二、解釈の明確化の指針

ところでこうした見解は、その本質が技術であるコンピュータ・プログラムを著作権法により保護する理由を、プログラム開発には多大な知的労力と投資がかかる反面、他の技術と異なり容易に模倣やただ乗りがなされ得ることに求め、累積的に発展していく技術の特性を鑑み、プログラム開発者の利益と技術に対する社会一般の利益とのバランスを図ることを指針とする限りは、大凡正しい方向を示すものであると考⁽²⁶⁾える。しかし、一方ではプログラムというものは、そもそもコンピュータ内における処理の手順をステップ毎にアルゴリズムとして記述するものであるにも拘らず、プログラムにおけるいかに詳細なアルゴリズムであっても、これを一律に全て著作権法により保護されないアイデアとして片付けてしまうことは、プログラムの外面的形式としての指令の組み合わせの表現を規定しているその個々のステップ毎の指令の組み合わせの方法さえも、これをアルゴリズムとして著作権法第一〇条第三項第三号にいう「解法」に該当するものとし、万人が自由にこれを利用することを許すことになる。しかしそうなれば、プログラムとしてある「指令の組み合わせの表現」がある場合でも、その「指令のステップ毎の組み合わせ

わせの方法”は万人が自由に利用することができることになるので、結局そこから必然的にもとのプログラムと同一の指令の組み合わせからなるプログラムを万人が作成することを承諾する結果になる。とすれば、全てのプログラムについて表現とアイデアとが融合ないしは混同（マージ）することを認めるのと同じことになり、結果的にはプログラムの著作権法による保護を否定するに至ってしまうことになる。ただ、こうした矛盾はハイブリッドな機能作品としてのプログラムの特性から由来するものであると考えるが、このことはプログラムの法的保護を考えるにおいては、何らかの程度のアルゴリズムを考慮に入れざるを得ないことを示唆するものでもある。というのも、実際のところコンピュータ・プログラムにおいて本質的なのはアルゴリズムであつて、それがコンピュータ言語で表現されていようが、その他の形態で表現されていようが、結局はそれをコンピュータに理解可能な形態に翻訳・解釈するコンパイラないしはインタープリター⁽²⁸⁾があるかないかの違いだけなのであり、それさえあれば如何なる形態のものでもコンピュータに対する指令を与えうるものとしてコンピュータ・プログラムといえるからである。従つてコンピュータ・プログラムの法的保護を考えるに当たつては、確かに現在ではコンピュータ・プログラムがプログラム言語で表記されるという現実は無視できず、またそれが著作権法により保護されることになつた結果として、著作権法の「表現」を保護するという原則を尊重しなければならないことは当然であるが、結局はその著作権法も翻案という概念を認めることにより実質的にはある程度アイデアをも保護するに至つている実状に鑑み、且つ冒頭に述べたような諸視点をも視野に入れて、いくらか政策的にコンピュータ・プログラムの表現の範囲及び保護範囲を決定せざるをえないものと思われる。

そこでプログラムにおけるアルゴリズムについていえば、一面では、これは問題解決のための処理の手順であり、それ自体としてはプログラムのアイデアであることは否定できない。しかし他面では、プログラム開発過程において、一定の固有の問題を解決するため、既存の基本的なアルゴリズムや新たに発見したアルゴリズムを幾度も選択・配列し、より複雑なアルゴリズムを構成しながら、その精緻化を図っていく過程が、ソフトウェア・エンジニアやプログラマーの最も多大な知的労働と個性の発揮される部分であり、またこのようにしてアルゴリズムが十分に精緻化された時点では、それが創作的な指令の組み合わせの表現としてのコンピュータ・プログラムの外面的形式に大きな影響を及ぼしうることは看過できないものと思われる。それ故、プログラムに表現されている限りにおいては、十分詳細に精緻化されたアルゴリズムは、プログラムの創作性やその指令の組み合わせの外面的形式を規定しているという意味で、著作権法上考慮しうるに足る特性を有しているものといえる。

従ってコンピュータ・プログラムにつき、著作権法による保護を考える際には、プログラムの外面的形式から出発することが正しいとしても、平均的プログラマーから見てプログラムの指令の組み合わせの外面的形式の本質的部分を規定している限りにおいては、プログラムから読み取れる十分詳細なアルゴリズムは、プログラムの外面的形式を保護する反射として実質的に保護されるか、あるいはもはやここに至っては実質的にプログラムの表現を構成するもの(内面的表現形式)と解しても差し支えないのではないかと考える。⁽³⁰⁾従って、他社開発のプログラムにおけるこうした十分詳細なアルゴリズムを利用してプログラムを作成し又は言語転換することは、外面的なステップ毎のコードの類似性がなくとも、実質的にもとのプログラムの指令の組み合わせを利用するものとして「翻案」

に該当するものと考えるべきであろう。さらに、あるプログラムからその十分詳細なアルゴリズムを記述したフローチャートを作成した場合に、そのフローチャートが平均的プログラマーから見てもとのプログラムの指令の組み合わせの本質的部分について特定できる程のかなり詳細なものならば、これに依拠して別のプログラムを作成するという程度では、同様の理由で著作権法上「翻案」に該当するものと考えられる⁽³²⁾。これに対し、プログラムにおける十分詳細なアルゴリズムとはいえない、それよりも抽象的なプログラムの構造、処理の流れや手順（これも見方によってはアルゴリズムであるが）は、まさに著作権法第一〇条第三項第三号の「解法」に該当するものとして保護されず、万人が自由に利用することができるものと考ええる。そしてこのあたりが政策的にみても、先行プログラムのただ乗りを防止するのに必要な境界であり、プログラム開発者の利益とプログラムに含まれる技術についての社会的利益とのバランスを図る接点であるように思われる。

しかし、以上のようにコンピュータ・プログラムの表現の同一性の範囲（ないしは保護範囲）を確定することができるとしても、具体的にはこれは裏返せば侵害訴訟の脈絡においては侵害基準との関連においても問題となり、その観点からはコンピュータ・プログラムにあっては、①先に述べたような保護対象の範囲内にあるコンピュータ・プログラムの表現に対してアクセス(access)があり、且つ②その表現と実質的類似性(substantially similar)のある表現のプログラムが作成・開発されたことが証明されれば、一般に、事実上侵害を推定させる一応の証明(prima facie proof)となりうる⁽³³⁾といえるであろう。しかし、その時にもコンピュータ・プログラムにあっては、プログラムの技術製品及び実用品の特性に鑑み、侵害を申し立てられたプログラムについて類似性を来した原因

として、①その開発に当たつての技術的な制約、②プログラムの経済性や効率性、さらには③技術一般の本性から由来するその収斂性等をも考慮しなければならぬであらう。⁽³⁴⁾ 実際、実体的な著作権法の観点からしても、他社開発のプログラムに依拠したとしても、その十分詳細とはいえない、より抽象的な構造又は処理の流れや手順、すなわち著作権法によっては保護されないプログラムのアイデアにまで遡つて、それのみを利用して別のプログラムを開発した場合に、独立創作ともいえるに等しい知的努力がなされた結果として実質的に類似するプログラムが作成・開発された時には侵害はないと考えられる。ただ、確かに現実の訴訟になれば、他社開発のプログラムに依拠してプログラムを開発した場合に、当初のプログラムと実質的に類似したものが開発された時には、まず原告となる当初のプログラムの著作権者が、被告が原告のプログラムの表現にアクセスし、原告のプログラムと実質的に類似するプログラムを開発したことを証明すること、先に述べたように、一応事実上侵害の推定がなされることにならう。しかしその場合でも、次に被告の側で、依拠したプログラムのアイデアまで遡り、且つ先に述べたような諸要因を考慮した上で独立創作にも等しい知的努力をなした結果、表現において実質的に類似するものが開発されたということを反証できる場合には、侵害責任は否定されるべきである。何故なら、当初のプログラムの表現に対するアクセスがあったとしても、アイデアにまで遡ったことが証明され、且つまたプログラムの開発に当たつての技術的制約、さらには効率性や経済性等の要因を考慮した上で、独立創作に等しい知的努力がなされた結果として、表現において実質的類似性のあるプログラムが開発されたと反証できる場合には、同一のアイデアから技術的・経済的な実際上の必然性あるいは技術的の収斂性の帰結として、実質的に類似するプログラムが開発されたといえ、

そこにはもはや当初のプログラムの表現の流用やただ乗りがあるとはいえないからである。ただ、こうした反証は往々にして困難なこともありうるが、プログラム開発に伴う諸判断、費用、労力や手続等が文書記録（いわゆる米国で paper trail といわれるもの）として残されている場合には、これについての専門家の鑑定に基づき、裁判官が独立創作といえるに等しい知的努力がなされたか否かを法的評価として判断し決定すべきであろう。また実際、ソフトウェア開発実務の現場では、既存のプログラムのアイデアを参考にしてプログラムを開発する場合に、結果的に実質的類似するプログラムが作成された時にも、単にアイデアを利用した独立創作であると主張せんがために、既存のプログラムの解析チームと新たなプログラムの開発チームを全く隔離し、解析チームの得た技術的アイデアの授受のみを開発チームに行い、表現の授受を遮断するというクリーン・ルーム方式又はアイソレーション・ブラス方式と呼ばれるプログラム開発方法が取られることがある。そこで、こうした方法が取られた場合に、その手続過程で I D カード等による各室への出入チェックと記録保存がなされ、またあらかじめ解析チームと開発チームとを地理的に離れた施設に置くというような厳格な分離がなされた上で、解析チームの抽出した技術情報で開発チームにトランスファー (transfer) されるものに技術情報抽出の対象となった他社開発プログラムの表現に当たるものが含まれていないことを法律専門家である弁護士等の検閲を受け、さらに解析チームによる技術情報抽出過程とその解析チームからトランスファーされた抽出技術情報を利用する開発チームのプログラム開発過程について文書記録が残されていれば、上述の意味での独立創作を証明する有効な証拠を残す手続となりうるものと考え⁽³⁵⁾。

またそもそも侵害訴訟においては、問題となるプログラムの形態や類型なども考えて侵害の有無を判断する必要

があり、例えばOS⁽³⁶⁾といわれる基本ソフトはアプリケーション・プログラムと比較するとさまざまな技術的制約が伴うことが考慮されるべきであり、また例えば小さなプログラム、ルーチンやモジュール⁽³⁷⁾、さらには現行の技術水準から見ても一般にありふれたプログラム又はもはや陳腐となったプログラム等は、その抽象的な構造や処理手続をもとに平均的プログラマーなら、およそ誰でもほぼ同一の表現に至りうるか、あるいは技術的に自明なほぼステップ毎のアルゴリズムを体现しているものとして、実際上はそのデッドコピーさえも問題にならないように思われる。⁽⁴⁰⁾

次に、わが国の著作権法第一〇条第三項第二号によれば、プログラムの保護はプログラムにおける「規約」に及ばないとされている。この規約とはインターフェースやプロトコルといわれるものを指し、その中でも異なった複数のプログラムあるいはプログラムとハードウェアとが互いに制御の受け渡しをしながら連動し作動するための規則（プログラム間インターフェース、ソフトウェア／ハードウェア間インターフェース⁽⁴¹⁾、および距離的に離れた異なった情報機器内のプログラム（通信プログラム）が互いにデータの送受信を行うための規則（プロトコル、通信規約⁽⁴²⁾）で、しかも特定のプログラムに含まれているものを意味するものと考えられる。これらは、交換機の製造やコンピュータ間の通信にとっては不可欠な技術仕様ともいえる情報であり、もしこれに独占権を与えようとすると、交換機の製造や周辺機器の他社による製造を不可能にし、さらには異機種間の相互接続や通信がなしえなくなるために、特定ベンダーのコンピュータ・システムを利用するユーザは長期にわたって同一ベンダーのシステムにロック・イン（Lock-in: 拘束）されることになる。また、そのベンダーが市場で独占的な地位を占めている時にはその独占は一層助長され、競争が阻害されることにもなる。⁽⁴³⁾従って、こうした情報それ自体にはコンピュータ・シ

システムの構成諸要素間における互換性や相互接続性を確保する社会的利益の必要上、及びそれ自体としては一般にアイデアに属するものとして保護が及ばないとするのが著作権法の立場と考えるべきである。ただそれ以上に、インターフェース又はプロトコル等の規約が特定のプログラムにおいて実現 (implement) されている場合に、一般論として、そのプログラムは著作権法上表現として保護されるといえるとしても、そのプログラムに含まれたインターフェース又はプロトコルが具体的でかなりの程度厳密な仕様を前提とするときに、その仕様がそれを実現するプログラムの指令の組み合わせを実質的に規定し、それ故にこうした規約がプログラムの表現を構成していると思われることもある。しかもその場合に、こうした具体的で厳密レベルでのインターフェースやプロトコルの仕様を利用しなければ、互換性又は相互接続性を図ることができないと見られるときには、こうした規約を実現し且つ互換性又は相互接続性を達成しうる別のプログラムを作成しようとするれば、そのプログラムのこうした規約を実現する部分については、不可避的にもとのプログラムと表現が同一になったり又実質的に類似せざるを得なくなる。そこで特に互換性・相互接続性を確保する社会的利益の見地より、プログラムにおける規約は如何なるものでもそれ自体としては保護しないとすることが著作権法一〇条三項二号の趣旨であると考え、その結果として、右のような場合には、もとのプログラムの形態における規約を表現した部分にも保護が及ばないとするのが著作権法の立場でもあると考えるべきである。しかし、どのような状況においてこうした結果になるかは、個々のに検討して決すべきであらう。

なお、インターフェースやプロトコルの開発には多大な費用と時間とを必要とし、最初にこれを開発した者に何

らか利益を与えることなく、これを万人が自由に無償で利用してもよいとすると、最初に開発した者だけが不利益を被りセカンド・ランナー以下が利益を受けることになり、不当であるばかりでなく開発意欲を減退せしめるので、独占権はともかく立法論としては対価（報酬）徴収権を認めることが望ましいとする見解がある。⁽⁴⁴⁾しかし、標準化のなされていない現在、先行者によるより良いインターフェース又はプロトコルの開発と改良は、結果的に、その良さの程度に応じコンピュータ機器や通信関連市場において採用されることになれば、それによる市場支配を通じて、それ相応の開発投資の回収と開発意欲の維持を確保できるものと思われる。むしろその者による独占を通じて、ユーザの囲い込みがなされることの方をより一層警戒すべきであろう。また、インターフェースやプロトコルそれ自体を保護するとすれば、これらには具体的レベルから抽象的レベルまでさまざまなものがありうるので、どのレベルまでのものを保護するのか問題になろうし、またあるインターフェースあるいはプロトコルが、他のインターフェースあるいはプロトコルとどの程度類似していれば同一のもので、どれだけ異なっていれば別物といえるのかという同一性確定の問題が、プログラムの表現の同一性の確定以上に困難な問題となりうることが予想される。さらに、インターフェース及びプロトコルについては、標準化が促進されることが望ましいが、これらに権利性を認めるのに比べ、権利性を認めない方が標準化作業が円滑に行われることが期待できる。そこで、私見としては、インターフェース又はプロトコルそれ自体には、差し当たっていかなる権利性を認める必要はないと考えるたい。

(26) 技術としてのコンピュータ・プログラム (computer program as technology) の特性に鑑み、その保護を考えるべきであるが、ただコンピュータ・プログラムがかって技術を保護したことのない著作権法によって保護されることにな

った政策決定においては、他の技術とは異なり、プログラムは模倣されることにより原プログラムと競争的に市場に出されることの容易さ——すなわち盗用 (piracy) の容易さ——に求められると主張するものとして、Dennis S. Kajara, COPYRIGHT, COMPUTER SOFTWARE, AND THE NEW PROTECTIONISM, Jurimetrics (Fall 1987) p. 33.

(27) アルゴリズムとプログラムの関係については注21。

(28) コンパイラー及びインタープリターについては注18。

(29) 中山前掲四七〜四八頁。

(30) ここで簡単に、プログラムの開発過程を辿ってみると、おおよそ次のような段階を経てプログラムが開発されることになる(括弧内は各段階で作成される文書類)。(1)要求分析 (requirement analysis): 何のために、何をやる情報システムが必要なかを明らかにし、その中に位置づけられるコンピュータ・システムの役割を決定する(問題記述、プロジェクト計画書)。(2)システム設計 (system design): コンピュータ・システムについて、データの出入力の方法と様式等を決め、作成する必要があるプログラムの仕様を明らかにする(システムチャート、プロセスチャート、プログラム仕様書)。(3)プログラム設計 (program design): プログラムの仕様として与えられた課題を確認し、その課題の解決のための処理の手順(アルゴリズム)を設計する。すなわち、プログラムが実現すべき課題の解決のためにアルゴリズムを考案し、これを基本的なものから次第にコンピュータに実行させる具体的な処理の手順として練り上げ、精緻化し、まとめていく(基本プログラム・フローチャート、詳細プログラム・フローチャート、プログラム設計書)。(4)コーディング (coding): コンピュータに実行させる処理の手順、すなわちアルゴリズムに合わせて、プログラムをプログラム言語で記述し、これをコンピュータに読み取れる媒体(パンチカード等)に移し、コンピュータに入力して、これをソース・プログラムとし、アセンブラやコンパイラー等の翻訳プログラムにかけて、コンピュータに実行可能な形式の機械語形態のオブジェクト・プログラムに転換する(コーディング・シートに書かれたプログラム等)。(5)テスト、デバッグ (testing, debugging): 出来上がったプログ

ラムをテストデータを与えて実行させ、実行結果に誤りや不都合があったり、コーディング上のミスがあれば、望ましい結果が出るようにプログラムを修正する。⑥保守・運用のための手当(maintenance): コンピュータ・システムのユーザが効率的に当該システムを利用できるように手当をする(ユーザ・マニュアル、メンテナンス・マニュアル)。プログラムの開発過程については、松田政行編著「コンピュータ・ビジネス・ロー」商事法務研究会一九〇二二三頁参照。

(31) ヴイツの学説におけるプログラムにおけるアルゴリズムの取扱を論じる代表的な学説的見解として Eugen Ulmer/Gert Kalle, *Der Urheberrecht von Computerprogrammen*, GRUR Int. 1982 S. 489ff. は、次のように述べる。「広く流布している見解とは反対に、プログラムの言語的及び外面的形式の付与においても(in der sprachlichen und äußeren Formgebung der Programme) 自然言語でなされた表現ほどは大きくないとしても、個性的な創作のためのかなりの自由度(ein erheblicher Spielraum für individuelles Schaffen)が既に存在している。すなわちこのことは、全く自由に選択し得るオペランド記号(Operandsymbolik)つまり定められた命令記号(オムレーション)に帰属させられる情報要素の表記(die Bezeichnung der den festgelegten Befehlsymbolen (Operationen) zugeordneten Informations-elemente) (たとえば、プログラムセグメント(Programmabschnitte)、データ領域(Datenfelder)、データファイル(Dataien)、記憶領域(Speicherbereiche)、分岐アドレス(Verzweigungsadressen)、副プログラム(Unterprogramme)等のための表記)についていえることである。個々のプログラム内における自由に選択されたオペランド記号及び所与の命令記号の組み合わせ、およびその並びを通じて個性的な刻印(die individuelle Prägung)は強められる。これに加え、実際上のすべてのプログラムにおいて見られるものとして同じく自由に選択され、大抵は日常言語でなされるコメント(Erklärungen)、説明(Erklärungen)及び指示(Hinweise)などがある。これらのとりわけプログラムの理解及び読み易さに役立つプログラム要素の総体によって基礎づけられる形成行為の自由度(Gestaltungsspielraum)は、既に、著作権法保護にとって要求されるプログラムの個性を肯定するに適切なものである。さらに個性的な形式付与のための自由度は、

プログラムの構造 (Programmstruktur) 〴〵まり全体としてのプログラム内において、プログラムセグメント及び副プログラムがいかに分割され且つ結び合われているか、つたさの態様や様式 (in der Art and Weise, wie Programm-Abschnitte und Unterprogramme innerhalb des Gesamtprogram aufgeteilt und verknüpft) の領域にわたつても開かれてゐる。

「ならに、個性的なプログラムのための数多くの可能性は、しかしまたプログラムの内容において (auch im Inhalt der Programme) 〴〵なわち素材の創作的な収集、選択、分類及び配列に関わる領域 (dem Bereich, in dem es um die schöpferische Sammlung, Auswahl, Einteilung und Anordnung geht) にわたつても存在する。内容的な態様に関わる創作的、個性的な決定の自由度 (Spielraum für schöpferische, individuelle Entscheidung inhaltlicher Art) が、このわけ、しばしば必要とされる入出力量の選択 (bei der oft erforderlichen Auswahl von Eingangs- und Ausgangsprößen) アルゴリズム的解決要素 (たとえこれらはそのものとしては既に知られ又自明であつたとしても) の選択、適応、変更又はその新たな開発 (bei der Auswahl, Anpassung, Abwandlung und Neuentwicklung von algorithmischen Lösungselementen (auch wenn diese oft als solche schon bekannt sind oder nahelegen)) 及びプログラムの流れの組織化、すなわち個々の命令・命令群及びサブルーチンの具体的な配列から帰結するプログラムの構成 (bei der Organisation des Programmablaufs, d. h. bei dem aus der konkreten Anordnung der einzelnen Befehle, Befehlsgruppen und Unterprogramme folgenden Programmaufbau) において存在するものとあふ。最後にプログラムは、その時々の最適化目的に照らしてなされなければならず、また部分的には同一のプログラム内においても度々変更される優先度の決定を通じても独自性の刻印 (eine eigentümliche Prägung) を保有する。強調すべきは、非常に創作的たりうる全てのこうした内容的な要素 (alle diese inhaltlichen Elemente) は、プログラムの組織 (das Gewebe des Programms) を構成するものであり、直接的にその具体的な形態に入り込む (unmittelbar in seine konkrete Gestalt einfließen) といふことであらう。

またにプログラミングに際しては、内容的な考慮及び決定のためのかなりの自由度を有し得るので、プログラムに個性的な精神的創作という特徴を付与するその適性は、それがシステム、原理又はアルゴリズムそれ自体の著作権的保护に帰結するところ根拠によつて否定されることは出来なう。(S. 495)

また「コンピュータ・プログラムに於て、結論としては、プログラムが依拠する抽象的要素 (die abstrakten Elemente, auf denen die Programme beruhen) 十分な数学的原理 (die mathematischen Prinzipien) 計算方法 (die Regelmethoden) 専門用語及び符号アルゴリズム (Algorithmus) とされるものな、それ自体として (als solche) は保護される著作物とはいへない。しかし既に述べたように、アルゴリズム的解決要素 (die algorithmischen Lösungselemente) は直接的にプログラムの具体的形態に入り込み (unmittelbar in die konkrete Gestalt des Programms eingehen) 其の結果プログラムの個性的な刻印に於て決定的である (mitbestimmend für das individuelle Gepräge des Programms sind) 。

それが個々のプログラムと具体的に融合するに於て (in ihrer konkreten Verschmelzung mit dem einzelnen Programm) 。

それ故にアルゴリズム的解決要素は、著作権的保护に於て取り込まれることになる (vom Urheberrechtsschutz miteinfaßt) 。」(S. 497) と述べた。同 Gert Kollé, Der Rechtsschutz der Computersoftware in der Bundesrepublik Deutschland, GRUR 1982, S. 443ff. (S. 454); Ulrich Loewenheim, Der urheberrechtliche Schutz der Computer-Software, ZUM 1985, S. 26ff. (S. 29-30); Die Denkschrift der Deutschen Vereinigung für gewerblichen Rechtsschutz und Urheberrecht über den Rechtsschutz der Datenverarbeitungssoftware, GRUR 1979, S. 300 (SS. 303, 304).

これに對して BGH Urteil vom 9. 5. 1985—Inkasso-Programmentscheidung—, BGHZ 94, S. 276ff.; CR 1985, S. 22ff.; NJW 1986, Heft 4, S. 192ff. の連邦通常裁判所 (BGH) の判決は、一九八五年六月二十四日のドイツ著作権法改正によりコンピュータ・プログラムが「言語の著作物」(Sprachwerke) として明文上著作権法により保護され

るものとなつたままに直前に下されたものであり、また依然唯一の連邦通常裁判所判決として先例的価値を有し続けているものとして興味深いものがある。この判決の骨子としては、①コンピュータ・プログラムは、ドイツ著作権法第一条という学術の範囲 (Bereich der Wissenschaft) に属する著作物であり、著作権法第二条第一項の掲げる七つの著作物のカテゴリーのうち、文書 (Schriftwerk) (第一号) か、あるいは「学術的又は技術的態様の描写」 (Darstellung wissenschaftlicher oder technischer Art) (第七号) として考慮され得ること、②プログラム開発過程の個々の段階において、システム分析者又はプログラマーにより精神的な態様の成果 (Leistungen geistiger Art) がもたらされるが、その精神的な思考内容 (der geistige Gedankeninhalt) は、表現された内容の思考形成や遂行において (in der Gedankenformung und-führung des dargestellten Inhalt) 、あるいは提示された素材の収集、分類及び配列に関わる特に精神性豊かな形式又は態様において (in] der besonders geistvollen Form und Art der Sammlung, Einteilung und Anordnung des darbotenen Stoffe) 、その沈澱と表現 (seinen Niederschlag und Ausdruck) を見出し得るが、学術的又は技術的著作物の著作権法的判断 (die urheberrechtliche Beurteilung wissenschaftlicher oder technischer Werke) においては、表現された内容の思考遂行や形成におけるその精神的・創作的な内実 (ein geistig-schöpferischer Gehalt) は、学説及び学問的成果は自由で誰にでも利用可能であることから、排除されること、③従つてコンピュータ・プログラムにあっては、その中で考慮され、所与の計算機に関係づけられた計算規則 (Rechenregel) (いわゆるアルゴリズム (der sogenannte Algorithmus)) は、プログラムの制作に際して引き合いにされたその他の数学的又は技術的学説や規則と同様に著作権法の保護を受けないこと、④それ故、コンピュータ・プログラムの保護については、素材の収集 (Sammlung) 、分類 (Einteilung) 及び配列 (Anordnung) の形式 (Form) 及び態様 (Art) のみが考慮されること、⑤しかもコンピュータ・プログラムが著作権保護を享受するためには、情報 (Informationen) 又は指令 (Anweisungen) の選択 (Auswahl) 、収集 (Sammlung) 、配列 (Anordnung) 及び分類 (Einteilung) における形成活動 (Gestaltungsfähigkeit) が一般平均的に可能なことに対して明

らかにぬきこんでいること(ein deutliches Übertreten ... gegenüber dem allgemeinen Durchschnittskönnen)を前提とすることを判示する。この判決は、アルゴリズムはいかにしてもプログラムの表現たりえず万人に自由であること、及びプログラムが著作権法によって保護をうけるためには、平均的に可能であることに對して明らかにぬきこんでいるという高い創作水準を要求するものであるといえる。この判決に對しては、コンピュータ・プログラムの著作権保護を認めるのにつき高い創作水準を要件とする場合、その要件を満たすかどうかについて、アルゴリズムを考慮しないとすると何を考慮して判断しうるのかと批判し、アルゴリズムもプログラムの表現となりうることを見解として、Helmut Haberstumpf, Grundsätzliches zum Urheberrechtsschutz von Computerprogrammen nach dem Urteil des Bundesgerichtshofs vom Mai 1985, GRUR 1986, S. 222ff.; Helmut Haberstumpf, Der urheberrechtliche Schutz von Computerprogrammen, in: Michael Lehmann (Hrsg.), Rechtsschutz und Verwertung von Computerprogrammen, S. 7ff. ことに反して B G H の判例を支持し、プログラムは著作権法に よつては例外的にしか保護をされないとし、むしろ独自立法による保護(Sonderschutz)によるべきだとする見解として、Gernot Schütze, Urheberrechtsschutz von Computerprogrammen- geklärte Rechtsfrage oder bloße Illusion?, GRUR 1985, S. 997ff. ことにドイツの判例の分析については、辻正美「西ドイツにおけるソフトウェア法の展開」日本工業所有権法学会年報第一二号六頁。

(32) (プログラム)フローチャートとは、プログラム開発の段階で、プログラムのコーディングをする前段階として、処理の手順を図・記号を用いて書き表したものである。処理の概要を基本フローチャート(general program flowchart)に書き表した後、詳細な処理手順を詳細なフローチャート(detail program flowchart)に分解・記述する。このフローチャートといつてもいろいろの程度のものであることは注意すべきである。JIS 情報処理用語(JIS X-0001-1987, 01.05.2)は、フローチャートという用語より、むしろ「流れ図」という用語を用い、流れ図(flow chart, flow diagram)を「問題の定義、分析又は解法の図的表現であつて、演算、データ、流れ、装置などを表現するために記号を用いたもの」としている

(JISハンドブック情報処理用語・コード編 一九八九/五二 日本規格協会二〇頁)。このフローチャートとプログラムの関係について、プログラムからフローチャートに遡る過程において、中山教授は、「ソース・コードとほぼ一対一に対応しているフローチャートは、ソース・コードの複製と見ることができよう。そのような場合は、プログラムをみて詳細なフローチャートを作成し、そこから新たなプログラムを書き下したとしても、著作権侵害を免れることはできない。これに対して、ジェネラル・フローチャートは、ソース・コードと一対一の対応関係はなく、処理の流れを示しており、ソース・コードの表現に含めるべきではない。この処理の流れは、まさに著作権法一〇条三項三号にいう『指令の組み合わせの方法』を指し、著作権法による保護を受けることはできない。従って著作権法の解釈としては、プログラムからジェネラル・フローチャートを導き、換言するならアイデアを抽出し、そこから独自にプログラムを作成すれば侵害の問題は生じない」とされる(前掲一一三頁)。このように、中山教授は、ソース・コードのプログラムからほぼ一対一の対応関係にある詳細なフローチャートは、ソース・コードのプログラムの「複製」とされるが、私見によればソース・コードのプログラムから厳密な「一対一の対応関係」になくても、平均的プログラマーから見ても、そのプログラムの指令の組み合わせの「本質的部分について特定できる」ほどの詳細なフローチャートを遡って作成すれば、そのフローチャートはソース・コードのプログラムの表現を含んでいるが、外面的表現形式が異なることから、こうしたフローチャートの作成はソース・コードのプログラムの「翻案」と考えたい。何れにしてもこうした見解は、多少の違いがあるとしても、プログラムの指令の組み合わせを特定しうる限りにおいては、フローチャートを実質的にプログラムの著作物とみることになる。そして、何も同一人がプログラムから遡ってフローチャートを作成し、そこからまた新たなプログラムを書き下すという場合だけでなく、ある者が、平均的プログラマーから見ても最終プログラムの指令の組み合わせの本質的部分について特定しうるほど詳細なフローチャートを作成し、これをもとに別人が最終プログラムのコーディングをした場合にも、この最終プログラムの作成は、もとになった詳細なフローチャートの「翻案」と考えるべきであろう。ちなみに、植松前掲三七頁は、「それではフローチャートの

著作物とプログラムの著作物とはどういう関係に立つか、いいかえれば、プログラムの著作物はフローチャートの著作物の翻案か、という問題については、著作権法が『プログラムの著作物』として独立に規定している以上、プログラムはフローチャートとは別個独立の著作物になるのが原則と考えられるでしょう。ただ、フローチャートが非常に詳細であって、あとは初心者プログラマーでも、機械的にコーディングしていくだけでプログラムが完成するような場合には、プログラムがフローチャートの二次的著作物となる場合もありえましょう。しかし、プログラムはフローチャートの複製物かといえは、フローチャートとプログラムは表現形式が違い、また法はコーディングすることによってプログラムとなることに著作物としての成立を認めて規定をおいたものであることを考えれば、プログラムがフローチャートの複製物になるとは考えられません」とする。これらいずれも、多少の見解の相違が見られるが、中山教授も植松弁護士の見解も、結局一定限度、詳細なフローチャートは最終のコーディングされたプログラムと法的評価において同一の表現を含みうることに帰着し、その限度で私見も同旨であるが、もしそうならばフローチャートとプログラムの表現形式の違いに関わらず、そこには表現としてなお同一のものがあるとする根拠は何か又それは何かを、両氏は示していないように思われる。それは、結局一定の程度十分詳細で精緻化されたアルゴリズムではないのか、もしそうでなければ何なのか明らかにならなければならないであろう。また前掲著作権審議会第六小委員会（コンピュータ・ソフトウェア関係）中間報告四八頁参照。

(33) これは、一般に米国で *substantial similarity test* といわれるもので、原告が被告の著作権侵害を問うためには、被告が原告の著作物の *copying* をしたことを証明する立証責任があるが、これは実際上不可能なので (See, e. g., *Bumcraft v. Newmann Bros.*, 373 F. 2d 905, 907 (9th Cir. 1976))、このような原告の証明の実際上の困難さを軽減するため、アクセスと実質的類似性の証明がなされれば *copying* の蓋然性が高いために、これら二点の証明を原告がなせば、被告の著作権侵害が事実上推定され、この推定を破ろうとすれば、次に被告の方で独立創作 (*independent creation*) を証明しなければならぬとするものである。この *substantial similarity test* において、アクセス (*access*) とは、著作物を実際

に見づく¹⁾の判決があるが (See, *Bradbury v. Columbia Broadcasting Sys., Inc.*, 287 F. 2d 478, 479 (9th Cir. 1961); *Christie v. Harris*, 47 F. Supp. 39, 40 (S.D.N.Y. 1942))²⁾ だが一般的な定義として、著作物を見る機会のある人々³⁾ (See, *Smith v. Little Brown & Co.*, 245 F. Supp. 451, 458 (S.D.N.Y. 1965), *aff'd* 360 F. 2d 928 (2d Cir. 1966); See also, *Nimmer on Copyright*, §13.01[A])⁴⁾。また、後半の実質的類似性の判断については、米国の *Sid & Marty Krofft Television Productions v. MacDonald Corp.*, 562 F. 2d 1157 (9th Cir. 1977) を先例として、二段階のテスト (bifurcated test) が適用される⁵⁾。第一段階のテスト (extrinsic test) によつては著作物間の「アイデア」の類似性があるかが判断される。この段階では著作物を個々の部分に解体して分析する⁶⁾ (analytic dissection) が許され、その分析に専門家の鑑定を得ることも可能であるが、これに対して、第二段階目のテスト (intrinsic test) では著作物間の表現の類似性が専門家の鑑定意見なしに通常の観察者 (ordinary observer) の視点から判断される (この判例の先駆者となったものは *Arnstein v. Porter*, 153 F. 2d 464 (2d Cir. 1946))⁷⁾。しかし、先の注14の *Whelan Associates v. Jaslow Dental Laboratories*, 797 F. 2d 1222, 1232-1233 (3d Cir. 1986) は、高度に技術的なコンピュータ・プログラムについては、実質的類似性の判断は、通常の観察者では無理であるとして、裁判所は専門家の鑑定意見のみで決定し得るとし、二段階の実質的類似性のテスト方法を放棄した。わが国において、このように米国で発展した *substantial similarity test* と同様な判断方法を明言する判例はないが、実質的な訴訟上の証明の配慮としてそのヒューセンスは適用されてもよいように思われる。わが国における適用については、中山前掲一〇〇頁は、「著作権侵害の有無の判断基準は模倣があつたか否かであるが、模倣の立証は著しく困難であるため、現実には、他人の著作物へのアクセスと類似性で判断されることになる」とする。また植松前掲九四〜九五頁は、実質的類似性の有無は、著作権侵害の有無の重要な手がかりとなり、実質的類似性とアクセス、この二つが著作権侵害の成立の要件となるとする。しかし正しくは、実質的類似性とアクセスの証明は著作権侵害の成立の要件ではなく、著作権侵害を事実上推定させる一応の証明

に過ぎないというべきであろう。さらにアクセスの意義については、植松同書九七〜九八頁参照。

(34) 中山前掲九六頁、一〇一頁。

(35) 原告の著作権侵害についての立証責任と被告の paper trail による独立創作の反証の関係については、カージャラ・デニス「著作権法の下におけるコンピュータ・プログラムの保護範囲」法とコンピュータ四号一〇四頁参照。また、クリーン・ルーム方式又はアイソレーション・ブース方式によるプログラム開発については、中山前掲一二五〜一二七頁及び植松前掲一九四頁。米国におけるプログラムの保護範囲の拡大がアイソレーション・ブース方式によるプログラム開発に及ぼす影響については、佐野稔「アイソレーション・メソッドによるプログラム開発と米国の裁判例」NBL三七二号一七頁。

(36) OS (オペレーティング・システム: operating system) とは、JIS 情報処理用語 (JIS X 0001-1987, 01. 04. 05) によれば、「プログラムの実行を制御するソフトウェアであって、資源の割振り、スケジューリング、入出力制御、データ管理等のサービスを提供するもの。注 オペレーティングシステムはソフトウェアが主体であるが、部分的又は全体的にハードウェア化することが可能である」とされている (JIS ハンドブック 情報処理 用語・コード編 一九八九/五二日本規格協会一九頁)。結局 OS とはプログラムの中でも、ハードウェアであるコンピュータと、ユーザが実際に具体的な処理を行うために利用するアプリケーションプログラムとを媒介し、コンピュータ・システムのさまざまな資源を有効的に活用するための基礎的な一連の機能を果たすプログラム群と見ることができ、今日ではメインフレームばかりではなく、パソコンでも広くその利用が行われているのを見る (世界市場の七十%を占める IBM のメインフレーム用の MVS や IBM/PC における採用によってパソコン用に普及した MS-DOS 等)。そして、OS の機能としては、入出力や外部媒体上のファイルの管理、プログラムの実行の監視、TSS (Time Sharing System) やマルチタスク・システム等の場合に複数のプログラムが並行して実行される際には、それぞれのプログラムについての CPU に対する実行の割当等がある。その意味で OS は、実際上はユーザにとっては全く見えない部分であり、どちらかと言うと機械に近い位置づけを有するといえる

ものであるが、OSはハードウェアであるコンピュータの上に乗る、またそれと一体になって、アプリケーションプログラムがコンピュータ・システムで実行されるに際しての基礎的な環境を作り出す働きをするという点で非常に重要な意義を有している。

ところがこうしたOSも、ある種のものが一定の機種ハードウェアの上に乗る、他の競争者より先行して市場に出されるに普及するようになると、そのハードウェアとOSの上に乗って稼働するアプリケーションも多く開発され、これがまた跳ね返ってその特定のハードウェアとOSの普及を加速度的に拡大させ、しかもこれと相乗的にその上で開発されるアプリケーションも広範に増大することで、結局市場はそのハードウェアとOSおよびその上に乗って動くアプリケーションで占められてしまい、結果としてコンピュータを利用しようとするユーザのハードウェアやOSの選択の余地の狭小化や特定のハードウェアとOSに対するロック・イン (lock-in: 拘束) を生じさせることとなる。またこのために、市場での競争が成り立たなくなり、ことによると市場を制覇した特定のハードウェアとOSのメーカーは、市場での独占に跌をかい改め努力を怠り、コンピュータ利用に関わるユーザの不利益や社会の全般的観点から見た経済的デメリットを引き起こしかねない。

そこで、こうして実際上市場を制覇し又は有力な地位を占めて事実上の標準 (de facto standard) となった特定の機種ハードウェアやOSに対しては、後発者が、それらの上に形成された膨大なアプリケーション群をそのまま受け継ぐことのできる互換性のあるハードウェアやOSを市場に出すことで、特定のハードウェアやOSにより制覇された市場に参入し競争するという「互換戦略」が取られることがあるが、ただこの場合、既にIBM対富士通の係争またはNEC対EPSONの係争(ここではBIOSといわれる基本入力システムが争いの対象となり、これも広義にOSというものと思われる)で周知のごとく、OSについて後発者による著作権侵害が重要な争点として持ち出されるのを見る。ところがこうした場合に、事実上の標準となったハードウェアやOSとの関係で、安易に後発者による侵害を認め、結果的に先発者に強い法的保護を与えることがあっては、ともすれば先に述べたようなユーザの不利益や社会的なデメリットを助長することにな

りかねないので特別の考慮を必要とする。特にOSの法的保護について、その特性と市場分析を踏まえて反省を迫るものとす。Peter S. Menell, Tailoring Legal Protection for Computer Software, Stanford Law Rev. Vol. 39, p. 1329.

(37) 小さなプログラムの技術的特性についての研究として渡辺勝、辻ヶ堂信「極小さいプログラムの性質の一面について」AIPPI (1988) Vol. 33 No. 10, 五七九頁、渡辺勝、辻ヶ堂信、吉岡明彦「小さいプログラムの性質の考察」AIPPI (1988) Vol. 33 No. 11, 六八五頁参照。

(38) JIS情報処理用語 (JIS X 0015-1987, 01.04.06) によれば、ルーチン (Routine) とは、「他のプログラムによって呼び出され又は頻繁に使用されるプログラム」となっている (JISハンドブック情報処理 用語・コード編 一九八九/五二 日本規格協会二〇頁)。

(39) JIS情報処理用語 (JIS X 0015-1987, 15.01.9) によれば、モジュール (module program unit) とは、「プログラム単位」ともされており、これは「手続きやデータ宣言からなる言語的構成要素であって、他の同様の構成要素と相互に作用しうるもの。例 Ada のパッケージ、FORTRAN のプログラム単位、PL/I の外部手続き」となっている (JISハンドブック 情報処理 用語・コード編 一九八九/五二 日本規格協会二二二頁)。また、JIS情報処理用語によれば (JIS X 0015-1987, 15.01.11)、「プログラム言語におけるプログラム (program in programming language) とは、「二以上」の互いに関連するモジュールの論理的集まり」とも定義されている (同JISハンドブック二二二頁)。

(40) 中山教授は、プログラムのような機能作品については、創作性の概念の中に、特許法における進歩性のような考え方を導入する必要がある、ドイツの連邦通常裁判所の Inkasso-Programm 判決 (注31参照) が参考になると述べられ、そのような考え方を採用しない限り、他の技術保護法との比較上、平仄のあわない極めて不都合な結果となるので、プログラムにおいても、通常のエンジニアであるなら容易に作成しうるようなものは保護すべきではないとされる (中山前掲一〇四頁)。私見も基本的には賛成であるが、プログラムの保護される創作性の要件として、著作権法にはない「進歩性」という

積極的要件を正面に出すことなく、プログラムの技術製品としての特性を鑑み、現行の技術水準から見て、一般に平凡あるいは陳腐なプログラムは、著作権法にいう創作性のある著作物とはいえず保護されないといえ十分であると考える。

また、こうした考えをさらに一歩進めて、創作時には平凡あるいは陳腐とはいえなかったが、侵害問題時において、現行の技術水準から見て既に平凡でありふれたものとなり陳腐化したプログラムについても、それらはもはや公有に属するに至ったと考え、保護されないと解釈することができるのではないかと考える。

(41) インターフェース (interface) とは、JIS 情報処理用語 (JIS X 0009-1987, 09.01.01) によれば、「二つ機能単位で共用される境界部分であって、機能特性、共通する物理的相互特性、信号特性及びその他の適当な特性により定義されるものをいう。注 この概念は、異なる機能を持つ二つの装置の接続を行う仕様を含む」と定義されている (JIS ハンドブック 情報処理 用語・コード編 一九八九/五二 日本規格協会七五頁)。

(42) プロトコルないしは通信規約 (protocol) は、また通信プロトコルともいわれ、JIS 情報処理用語 (JIS X 0009-1987, 09.05.01) によれば、「通信を行うに際し、機能単位の動作を決定する意味上及び構文上の規則の集合」と定義されている (JIS ハンドブック 情報処理 用語・コード編 一九八九/五二 日本規格協会七八頁)。

(43) 中山前掲四四〇四五頁。
(44) 中山前掲四四六頁。

三 プログラムに含まれる情報へのアクセス——リバース・エンジニアリング

ところでコンピュータ・プログラムは技術製品であり、従ってそれが前提として技術内容を知りうる手だてを保障することは、市場での先行者の技術を後発者が習得し、自らの技術力を向上させ、先行者と有効に競争する

ことを許し、ひいては社会全般の技術・産業の発展を促すために不可欠のことである。またコンピュータ・システムの構成要素間の互換性や相互接続性を確保するために他社開発のプログラムを調査解析し、そこに含まれているインターフェースやプロトコルの仕様を知る社会的・技術的要請もある。⁽⁴⁵⁾そこでコンピュータ・プログラムに含まれるアイデアである技術内容を抽出するための調査・解析行為である所詮リバース・エンジニアリング (reverse engineering)⁽⁴⁶⁾を、著作権法は明文をもってこれを許容していないために、いかに考えるかが問題になる。ただ、コンピュータ・プログラムに含まれる技術内容を抽出するためには、対象プログラムを実行させたり、プログラム付属のマニュアルを読んだり、さらにはテスト・データを使って通信回線のトレースを行ったり、またテスト・プログラムを対象プログラムとともに実行させたり等して、ある程度の情報は入手でき、しかもこうした行為は、通常プログラム著作権者の排他的権利の内容に該当する行為を伴わないといえるので、その限りでは、このような態様のリバース・エンジニアリングはそもそも著作権法の関知するところではないといえよう。しかしながら、より詳細で正確な情報が必要とする時には、適正な料金によるプログラム権利者の情報開示がない限りは困難な問題に直面する。すなわち、こうした場合、コンピュータ・プログラムは一般の著作物とは異なって、通常、人が読み理解することの出来ない0と1の羅列であるオブジェクト・コードの形態で、且つ磁気テープやフロッピー等の媒体に乗せられて市場に出されているために、プログラムに含まれている技術的内容を知ろうとすれば、プログラムをコンピュータ内のメモリー (主記憶装置) にローディングした上で、プログラムのダンプ (特にプリントアウト) をしたり、また逆アセンブル又は逆コンパイルによりソース・コードに転換しなければならないことになる。ところ

がこうした行為が著作権法上の複製、翻訳又は翻案に該当するものとする、著作者の権利を重視する余り競争を通じた技術や産業の発展を阻害しかねず、また高度情報化社会の要請に答えることも出来なくなる。そこで、他の典型的な技術保護法制である特許法（第六九条第一項）や実用新案法（第二六条）あるいは半導体集積回路の回路配置に関する法律（第一二条第二項）が、技術保護の要請と技術発展の要請とのバランスを図りつつ、権利者の排他的権利に該当するような行為を伴うリバーズ・エンジニアリングをも一般に認めていることとの関係で、著作権法上著作物としてのコンピュータ・プログラムのリバーズ・エンジニアリングが許容されるか否かが問題になる。

これについてわが国において、米国著作権法第一〇七条の公正使用（fair use）の概念に準じ⁽⁴⁷⁾又はこの要件を参考に⁽⁴⁸⁾して、プログラムのダンプ、逆アセンブル又は逆コンパイル等のように、プログラムの複製あるいは翻訳・翻案に当たたる行為を伴うリバーズ・エンジニアリングであっても許容されるとする見解もみられるが、必ずしも一般には明らかではない。そこで、現行の著作権法の解釈論としても、こうしたリバーズ・エンジニアリングを許容する方向で、次のように考えるべきであろう。まず、①著作権法は、その第一条の目的規定において明示されているごとく、その最終目標は著作者の権利保護と同時に、今や産業的所産であるプログラムを含む著作物の公正な利用にも留意した、文化一般の発展に寄与することにあること、②そこである文化には、今日では広く文化的現象すなわち生活形式の総体を意味し、従って広く技術・産業文化も含む趣旨であること、③著作権とはいえ一般の私権であり、その限りでは私権の基本的関係を規律する民法、特にその第一条の一般規定により、公共の福祉に遵い、その濫用は許されず、信義誠実にこれ行使する義務を伴うものであること、④コンピュータ・プログラムを保護す

ることの必要性は強く認められるが、それと同時に高度情報化社会におけるソフトウェア技術の普及と発展は同程度に要請されること、⑤このソフトウェア技術の普及と発展という要請は、まさに高度情報化時代における公共の福祉の内容に合致し、それは翻つては著作権法にいう文化の発展に寄与しうる要請でもあること、⑥リバー・エンジニアリングは、こうした高度情報化社会におけるソフトウェア技術の普及と発展という要請を実現するために必要不可欠な手段であり、その意味で著作物の公正な利用といえるものだが、これをプログラム著作権者がいかなる場合でも理由なく禁止しようとすれば、こうした社会的要請が実現される道が閉ざされ、これがひいては技術・産業文化の発展を阻害し、公共の福祉にも反する結果を来すことになること、しかも⑦リバー・エンジニアリングそのものは究極的には、著作権法第一〇条第三項第二号乃至三号に掲げられた著作権法により保護されない技術情報を抽出するためになされ、これを禁ずることになれば実質的に著作権法はアイデア保護法になってしまうこと、以上の解釈論的論拠より、⑧わが国においてはリバー・エンジニアリングは一般に、たとえ複製や翻案などプログラム著作権者の排他的権利に該当する行為を形式的に伴うものであっても、それは著作物の公正な利用に該当し（著作権法第一条）、著作権者がこれを著作権の侵害——ないしは著作人格権の侵害——として主張することは権利濫用（民法第一条）に該当すると解釈することで、許容されるべきものと考える。しかもこの場合リバー・エンジニアリングは、対象プログラムの複製物を市場より購入し、その所有権を有する所有者のみではなく、ライセンスによりライセンサーから提供を受け、プログラムの複製物について所有権を有しないライセンスンであることも許容されるべきである。また、調査対象プログラムの複製や翻案を伴うリバー・エンジニアリングであつ

ても許容されるべきものとすれば、その結果当然のこととして調査対象プログラムの複製物や翻案物を作成し得ることになるが、その部数もリバース・エンジニアリングの目的に照らして適正な範囲内であれば、複数作成されることも認められていいと考えられる。

ただ、一般にリバース・エンジニアリングが許容され、その結果抽出された技術情報やアイデアは、プログラムに含まれた著作権法上保護されないアイデアの利用としていかなる目的に利用しうるとしても、右に述べたようにその抽出過程では、リバース・エンジニアリングの対象となったプログラムの「表現」を含む複製物や翻案物が作成されるので、こうした複製物や翻案物を流用し、その表現をそのまま取り入れた別プログラムの開発や、リバース・エンジニアリングの目的の範囲内にあるとはいえない別著作物の作成は行われるべきではない。こうした場合、リバース・エンジニアリング対象のプログラムの表現を取り入れたこれら別プログラムや別著作物は、リバース・エンジニアリングの対象となったプログラムの著作権を侵害するものといえる。しかもさらに、このようにリバース・エンジニアリングの過程で作成された複製物や翻案物が流用されて違法結果物が作成された場合には、遡ってリバース・エンジニアリングの過程で行なわれた複製・翻案等の行為をも捉えて全体として著作権侵害行為が行われたと解し、リバース・エンジニアリングの対象となったプログラムの著作権者は、リバース・エンジニアリングの過程を含めて著作権侵害が有ったと主張しても、もはや権利濫用には該当しないと解することもできよう。

次に、リバース・エンジニアリングの過程で適法に作成された複製物・翻案物の目的外使用、すなわちそれらの「頒布」や「公衆への提示」は許容されるべきではない。⁽⁴⁰⁾この場合著作権法は、著作物の公正な利用として、第三

○条以下に私的使用の目的等のための複製物・翻案物の作成を著作権の制限として認める一方、こうして作成された複製物・翻案物の「頒布」又は「公衆への提示」を、著作者の排他的権利としての著作物の複製又は翻案に該当する行為とみなす旨の規定(第四九条第一項第一号及び同条第二項第一号)を設けていることが参考になる。そこで、著作物の公正な利用と解されるリバース・エンジニアリングについても、これが許容されるとしても、明文はないが、その過程で作成される複製物・翻案物を頒布又は公衆へ提示する行為がなされた場合には、特に先の規定を準用して、リバース・エンジニアリングの対象になったプログラムの著作者(著作権者)が専有する複製又は翻案をする権利に該当する行為を行ったものとみなして、侵害を構成すると解することができよう。また先と同様、リバース・エンジニアリングの過程で作成された複製物又は翻案物が、結果として頒布され又は公衆に提示されることになれば、ここに至っては、もはや調査・解析目的のためのリバース・エンジニアリングの正当な態様を逸脱するので、遡ってリバース・エンジニアリングの過程でなされた複製や翻案等の行為をも捉えて、全体的にプログラムの著作者(著作権者)の複製権や翻案権を侵害するものと解し、リバース・エンジニアリングの対象となったプログラムの権利者が著作権侵害を主張することは「権利濫用」に該当するものとはいえないと解することもできよう。

そのほか、日本の著作権法第四七条の二第一項によれば、プログラムの著作物の複製物の所有者は、自ら当該プログラムを電子計算機において使用するために必要な限度において、当該プログラムの複製物又は翻案物を作成することが認められる一方、同条第二項によれば、これらの複製物・翻案物の所有者は、そのいずれかの複製物又は翻案物につき滅失以外の事由により所有権を失った場合には、その他の複製物・翻案物を保存してはならないこと

になっている。この規定はプログラムの複製物の所有者としてのユーザが、自らの事務や電子計算機に合わせて、プログラムを使用することができるよう、また使用に当たってのプログラムの安全を確保しうるよう手だてを保証したものであり、その限度でプログラムのバックアップや変更を行い、そのためにするプログラムの複製物・翻案物の作成を第一項において認めるものであるが、同時に同条第二項はこうした複製物や翻案物について取扱いを慎重にさせ、とりわけ著作者の利益を保護するために、これらの複製物や翻案物が実質的に二人以上の者へ分属する可能性を認めないとするのがその趣旨であるように思われる。そこでリバース・エンジニアリングを許容する場合にも、その過程で作成されるリバース・エンジニアリングの対象となったプログラムの複製物や翻案物についての取扱いを考える時には、特に同条第二項の規定の実質的趣旨が類推適用されていいであろう。ただ、第四七条の第二項はプログラムの著作物の複製物の所有権を有する者に焦点を当ててプログラムの複製物及び翻案物の取扱いを規定しているが、プログラムの複製物について所有権のあるなしに拘らず、リバース・エンジニアリングを許容する前提に立って、リバース・エンジニアリングの対象となったプログラムの原複製物及びリバース・エンジニアリングの過程でそれより作成された複製物や翻案物の取扱いについて考える場合には、第四七条の第二項の趣旨をやや拡大して解釈し、リバース・エンジニアリングの遂行者が、リバース・エンジニアリングの対象となったプログラムの原複製物及びリバース・エンジニアリングの過程でそれより適法に作成した複製物や翻案物のいずれかにつき、滅失以外の事由により、所有権はもろんのこと、その他いかなる所持ないしは使用の権原を失った場合にも、手元に残った他の複製物・翻案物は廃棄などし、保存してはならないと解すべきである。しかもこれに反し

て、複製物又は翻案物を保存した場合には、やはり第四九条第一項第四号及び同条第二項第三号の規定の趣旨を類推適用して、第四七条の二第二項に違反して複製物又は翻案物を保存した場合と同様、リバース・エンジニアリングの対象となったプログラムの著作者（著作権者）が専有するプログラムの複製又は翻案をする権利に該当する行為を行ったものとみなして、侵害を構成するものと考えるべきであろう。こう解することによって、リバース・エンジニアリングの過程で作成されるリバース・エンジニアリングの対象となったプログラムの複製物や翻案物が、不用意に外部に流出することに対し管理・保管を嚴重にさせ、プログラムの著作者（著作権者）の利益が不当に損なわれないことに資するものと思われる。

さらに解釈論として一般にリバース・エンジニアリングを認めるとしても、著作権法第一一三条第二項の適用がある場合、すなわちプログラムの著作物の複製物について、その使用権原を取得した時に著作権を侵害する行為によって作成された複製物であるという情を知っていた場合には、当該プログラムの複製物を業務上電子計算機において使用することは、プログラムの著作権を侵害したものとみなされる結果、違法複製物であるということを知ってその使用権原を得たプログラムの複製物については、業務上コンピュータを用いてなすリバース・エンジニアリングは、いかなる態様のものであっても許容されないことになろう。

こうして、一定の限界のもとに、プログラムに含まれる技術内容や技術情報を取得する手段としてのリバース・エンジニアリングは許容されるべきであると考ええる。しかし勿論、さらには権利者との契約による方法、またインターネットフェース情報や通信プロトコル等の不開示が独禁法違反になる場合には、独禁法による方法なども考えられる

であろう。結局、プログラムに含まれる技術内容や技術情報の取得のためには、なるべく多数の道を開いておき、あとは情報が必要とする者が、情報入手のために取り得る諸方法について効率性やコスト等を考慮した上でなす経済的選択に委ねることが、技術保護と技術競争による技術発展、さらには技術取引とのバランスを図ることに寄与することになるのではないかと考える。ただ、リバース・エンジニアリングの許容性とその限界については、先に示したように現行著作権法の解釈として、その個々の条文の実質的趣旨を勘案しながら明確化できないことはないが、現行のままではどうしても不明確なところが残らざるを得ないので、これを除去し又不要な争いを避けるためにも立法的措置が取られることが望ましいと考える。

(45) 高度情報化といわれる今日状況の中、コンピュータはシステムとして現出し、その中であって、このシステムの構成単位である個々のハードウェアはもちろん、種々のソフトウェアさらにはデータ記憶媒体等が、他と互換性があり、また他の構成要素と容易に相互接続でき、これによりシステム構成要素間での自由な相互運用性と情報伝達の可能性が保証されていることが要請される。例えば、①A社コンピュータ及びそのOSの上で稼動する種々のアプリケーション・プログラムがB社コンピュータ及びそのOSの上でも稼動である。②B社のホスト・コンピュータに通信回線を介して、単にB社開発の端末又はB社開発のその他のコンピュータだけではなく、A社開発の端末又はA社開発のコンピュータが接続でき、相互に通信が可能である。③ユーザが、かつてA社コンピュータ(及びそのOS)で業務処理を行い、これの上で稼働するプログラム資産を保有していた場合に、仮に業務の拡張のため、より高性能なB社コンピュータ(及びそのOS)に変えた時、A社コンピュータで使用していたユーザのプログラム資産がB社のコンピュータの上でも稼働である。④A社コンピュータで利用できる媒体及びそこに書き込まれたデータが、B社コンピュータやその上で稼働するプログラムによっても利用できる。

このような状況があつてこそ、情報処理技術及び情報通信技術は高度情報化社会を支えるにふさわしい社会の基盤技術たりうるし、また同時にコンピュータ・システムの個々の構成要素をなすハードウェアやソフトウェア等の市場が競争的に形成される前提を提供するよう思われる。ただ、今日の状況ではベンダーによりシステムの仕様が区々に異なるため、必ずしもこうした理想とはほど遠い状態にあり、これも将来的には「標準化」(standardization)により実現されることをある程度期待できるものの、現在では、システム仕様そのものの優劣を巡っても自由競争によるその淘汰と改良に任すべきところが多くあることは否定できない。従つて、ハードウェアやソフトウェアの互換性、相互接続性及びこれらによる相互運用性等は、必ずしも容易に保証されるものではないと考えられるが、しかしその場合でも、現状におけるユーザの立場からは、例えば以下のようなことが認められる必要性が大きいといえよう。まず、①のような場合には、A社コンピュータ用OSと種々のアプリケーション・プログラムとが互いに制御の受け渡しをしながら連動し作動するための約束事(所詮「プログラム間インターフェース」)が、A社コンピュータ用OSについて調査・解析されることにより抽出され、これに合わせてA社コンピュータ用OSと互換性のあるB社コンピュータ用の新たなOSの開発や既存のB社コンピュータ用OSの手直しが必要とされること。②の場合には、A社開発の端末内の通信プログラムが用いているデータの送受信のための約束事(所詮「通信規約」又は「通信プロトコル」)について調査・解析がなされ、それに合わせてB社のホスト・コンピュータの通信プログラムがこれをもサポートできるようにされること。③の場合には、A社コンピュータ用の言語処理プログラムの仕様が調査・解析され、その上で作成されたユーザ・プログラム(ソース・コード)をB社コンピュータ用の言語処理プログラムの仕様に合わせるように手直しされること、又はそのような仕様に転換するプログラム(プログラム・コンバータ)が開発されること。④の場合には、A社コンピュータにおいて、通常はそのOSやその他のプログラムを通じてデータが記憶媒体上に記録される形式(データ形式)や媒体を使用しデータを読み書きするための手順の調査・解析がなされ、これにより抽出された情報をもとに、記憶媒体上のデータが、B社コンピュータ及びその上で動くプログラムにおいても利用できるよう

に、データ形式を転換するプログラム(データ・コンバータ)が作成されること。

このように、コンピュータ・プログラムについては社会の高度情報化という流れの中で、コンピュータ・システムの構成要素間の互換性、相互接続性等が保証されるべき要請があるにもかかわらず、これが必ずしも全面的に保証されない差し当たつての現状において、他社開発のコンピュータ・プログラムを調査・解析し、それを通じて局部的にでもユーザのコンピュータ・システムにおける互換性や相互接続性等の達成を保証する手だてが法的に確保される必要性が感じられるのである。そしてこうした必要性を満たす手だてを法的に確保することが、ひいてはコンピュータ関連市場におけるベンダー間の競争を刺激し、また一面では標準化を促す要因たりうるのではないかと思われる。

(46) リバース・エンジニアリングとは、一般に他社の開発した工業製品を調査・解析・研究し、そこに含まれている技術的アイデアや情報、又はその製品製造のためのノウハウ等を抽出することをいう。コンピュータ・プログラムのリバース・エンジニアリングについていえば、プログラムに含まれている技術情報やアイデアの調査・解析行為ならばすべての態様が広くこれに含まれ、これに該当するものを概ね列挙すると、次のような行為がこれに当る。

①プログラムのテスト・ラン…調査対象プログラムをコンピュータで実行し、用意されたコマンドを種々に指定することで、その機能や作動の仕方を調査し、またテスト・データを入力し、その処理結果を見るために、これをアウトプットとして出力してみる。

②マニュアル調査…コンピュータ・プログラムに付属して提供されるマニュアルその他のドキュメントを手がかりに、そのプログラムの作動の仕方、そのプログラムの処理したデータが記憶媒体に対して読み書きされたり又は通信回線を経て伝送される規則や手順を、また調査対象プログラムが言語処理プログラムなら、その提供するプログラミング言語文法その他の言語仕様等を、主としてこうしたマニュアルやドキュメントを読むことで理解する。

③テスト・プログラムの利用によるプログラム間インターフェースや言語仕様の調査…ある機種のコンピュータ用のOSが

その上でアプリケーション・プログラムが稼働するために用意しているプログラム間インターフェースがいかなるものか、またそのコンピュータ用の言語処理プログラムにより用意されているプログラミグ言語がいかなる仕様になっているかを、調査解析者が自らテスト・プログラムを作成し、そのメモリー・ダンプを取ったり、実際にそのテスト・プログラムを実行し、その実行結果を分析したりすることで調査する。

④回線トレースによる通信プロトコルの調査…例えば調査対象たる他社開発の端末の用いている通信プロトコルを調査するために、その端末に接続された通信回線にテスト・データを流して、端末との送受信を時系列的に採取して、その結果を解析する。

⑤記憶媒体のダンプ…調査対象のプログラムを利用してデータを作成し、これをフロッピーディスク等のデータ記憶媒体に保存し、その上でこれらの記憶媒体に記録されたデータや制御情報をプリンター等にアウトプットし、プログラムがデータを記憶媒体に記録する時に用いるフォーマットやデータの格納形式を解析する。

⑥プログラムのメモリーダンプ、逆アセンブルや逆コンパイル等によるプログラム・コードの調査…以上のほか、調査対象プログラムをコンピュータ内のメモリーにローディングした上ダンプしたり（特に紙上へのプリントアウトや画面へ出力する）、オブジェクト・コード形式のプログラムを逆アセンブル又は逆コンパイルによりソース・コードに転換したりしてプログラム・コードを調査し、さらには当初からソース・コードが入手可能な時には、そのソース・コードを調査・解析すること。

以上のように、コンピュータ・プログラムのリバース・エンジニアリングとされる行為には、技術的にはさまざまな行為が含まれるのであり、またその結果として抽出される技術情報もさまざまなものがありうる。そうした情報の中には、①プログラムの機能や性能、②プログラムがコンピュータ内において実行する問題解決の処理の手順たる解法ないしはアルゴリズムといわれるもの、③ハードウェア/ソフトウェア間及びプログラム間のインターフェース情報、④通信プロトコル、

⑤ プログラム言語の文法や言語仕様、⑥ 記憶媒体のフォーマット形式やデータの格納形式等が挙げられよう。そしてこれらの情報はいずれもそのものとしては著作権法上、アイデアといって差し支えないものと考えるが、これらの情報の利用目的としても、① 調査対象となったプログラムのバグや障害除去を始めとして、② 改良やバージョン・アップ、さらには③ 調査対象プログラムと同一又は類似の機能・性能を持つ互換プログラムの開発、あるいは全く異なった機能を持つプログラムの開発に利用される等、*さまざまでありうる*。

(47) 中山前掲一三二頁。

(48) 阿部浩二「ソースプログラム(オペレーティング・システム・プログラム)について著作物性を認めた例」判例評論三四五号五九頁(判例時報一二四七号二〇五頁)。米国におけるリバース・エンジニアリングの許諾の議論としては、**COMPUTER SOFTWARE AND COPYRIGHT PROTECTION, The "Structure, Sequence and Organization" and "Look and Feel Questions",** Last Frontier Conference Report, Arizona State University, College of Law, Center for the Study of Law, Science and Technology, pp. 9-10.; "Reverse Engineering and Intellectual Property Law by the Committee on Computer Law", The Record of the Association of the Bar of the City of New York, Vol. 44, No. 2, March 1989, p. 133. また米国の判例を見ると、中には、被告会社のエンジニアが原告のコードをダンプし、フローチャートにし、分析したことはそれ自体において盗用 (pirating) を証明するものではなく、競争者のコードをダンプし、分析することは産業界における標準的慣行 (a standard practice in the industry) であるとしてリバース・エンジニアリングそのものを認めるが、リバース・エンジニアリングの利用行為において原告のプログラムと実質的に類似する表現でプログラムを開発したことを違法としたものがある (E. F. Johnson Co. v. Uniden Corp. of America, 623 F. Supp. 1485 (D. C. Minn. 1985))。一方、原告がライセンス料に応じてプログラムにガバナ (governor) をかけて、機能上の制限を加えてライセンスに提供していたのを、被告がこのガバナを、被告のプログラムのライセン

シーのために除去するサービス(後に同一の機能のプログラムを販売)を行い、その過程で被告がカバナーを発見し、除去するために原告のプログラムをコンピュータのメモリーにローディングしたことを著作権の侵害とした判例がある(Hubco Data Products Corp. v. Management Assitence Inc. 219 U.S.P.Q. 450 (D. Idaho 1983))。しかしまた他方では、原告が、PROLOK というコピー防止装置をもったディスクレットで、その上のプログラムと一体になって、そのディスクレットの上に第三者開発プログラムがさらに乗せられたならば、そのプログラムを普通のディスクレットにコピーし稼働させることを防止する機能を有しているものを販売していたのを、被告がコピー防止機能を無効にする RAMKEY というプログラムに乗った Copy Write というディスクレット(このディスクレットに、原告のディスクレットに乗っている第三者開発のプログラムがコピーされると、正常に稼働する)を販売し、しかもその開発過程で、被告が原告のプログラムを、コンピュータのメモリーにローディングし、DISK Explorer や IBM Debug 等の市販されているプログラムで解析し、また逆アセンブルあるいは逆コンパイルをしていた事実があったのにつき、原告はこれを著作権侵害と主張した事件で、裁判所は、被告による原告プログラムのメモリー内へのローディング(RAMへの複製)は、たとえコピー防止プログラムのコピー防止機能が無効にするプログラムを開発するという目的であっても、当該複製は米国著作権法第一一七条(1)に規定されたプログラムの使用に不可欠なステップとして許容されると判断したのも見られる(Vault Corp. v. Quaid Software Ltd. 847 F. 2d. 255 (5th Cir. 1988))。

(49) わが国の著作権法によれば、「頒布」とは、有償であるか又は無償であるかを問わず、複製物を公衆に譲渡し、又は貸与することをいい(著作権法第二条第一項二〇号)又「公衆」とは、特定かつ多数の者を含む(著作権法第二条第五号)とされている。

四 まとめ

以上、コンピュータ・プログラムの同一性の範囲及びプログラムに含まれる情報に対するアクセスについて述べてきた。前者については、プログラムとアルゴリズムとの関係を示唆し、その実質は有力な学説の主張とそれほど差異はないものの、ここでは明確にされていない理論的及び政策論的側面の交差を明らかにし、私なりの解釈論を展開した。実際アルゴリズムを何程か考慮しない限りは、プログラムの適正な、いや全くその保護さえも問題にしないというのが、私の率直な感想であり、著作権法とは異なる独自立法によりプログラムの保護を図っていたとしても、アルゴリズムがどの程度保護の対象となりうるかという検討なしでは、結局は同様の基本的な問題に突き当たっていたのではないかと思われる。またリバース・エンジニアリングについては、それを認める方向で現行の著作権法の枠内での解釈論的論拠及びその限界を示唆してきた。従って、現行のままでもリバース・エンジニアリングについては妥当に結論を導き出すことができるものと考えているが、その本質が技術にあるコンピュータ・プログラムを著作権法により保護する限りは、立法によりリバース・エンジニアリングの許容性及びその限界について明確にすることが必要な争いを避ける意味において必要とされると考える。

* 追記 本稿は、一九八九年秋に開催された私法学会民法部会の個別報告として、私が発表した内容に加筆訂正を加えたものである。私法学会の個別報告に至るまでは、さまざまな先生方からご厚い示唆・鞭撻をいただいたことにつき、ここに心

から感謝する次第である。またコンピュータ・プログラムについての技術については、多くの技術者の方々からお教えをいただいた。合わせて感謝の念を表したい。