論　説

プログラムのリバースエンジニアリングの法的課題

－著作権法・不競争法・独禁法からの検討－

明治学院大学法学部教授　高田　寛

Ⅰ　はじめに

種子島久時の鉄砲記(注 1) によれば、天文 12 年 8 月 23 日（1543 年 9 月 23 日)、一艘のポルトガル船が種子島に漂着し、わが国に初めて火縄銃が伝えられたという。いわゆる鉄砲伝来である。当時のわが国は戦国時代であり、戦国諸大名は、競って火縄銃を買い漁り、またそれを分解・分析・改良することにより、以前にも益して威力のある高性能な火縄銃を考案・製造し、また大量生産した。これと似たようなことは、現代社会の製造業では、広く一般に行われている。このように、製品や部品を入手して分解や解析などを行い、その動作原理や製造方法、設計や構造、仕様の詳細、構成要素などを明らかにすることをリバースエンジニアリング（Reverse Engineering）(注 2)という。

現代社会では、発明や製品の製造に関して、知的財産権に関する各種の法整備がなされているが、特許法上、リバースエンジニアリングという行為自体は違法ではないと解されている。その根拠となる条文が、特許法 69 条 1 項である。同法 69 条 1 項は、「特許権の効力は、試験又は研究のためにする特許発明の実施には及ばない。」と規定する。すなわち、特許法では、試験又は研究のためにするリバースエンジニアリングは違法ではない。違法性が認められるものは、主にその情報を使用する一部の場合のみである(注 3)。

一方、著作権法で規定するプログラム(注 4)の場合、著作権法には、特許法 69 条 1 項に相当する明文規定がなかったため、プログラムに対するリバースエンジニアリングが認められるかどうかについては、従前から議論されてきたものの、その解釈は曖昧であった。特に、プログラムのリバースエンジニアリングが、権利制限規定（特に改正前著作権法 47 条の 7）との関係で、著作権侵害にならないかが問題となっており、改正前においては、プログラム著作物の情報解析と評価できない以上、改正前著作権法 47 条の 7 に該当せず、著作権侵害（複製権ないし翻案権侵害）となるおそれがあった(注 5)。

そのため、ソフトウェア開発者（ライセンサー）は、自身が作成したプログラムの使用許諾契約書に、必ずと言ってよいほど、リバースエンジニアリング禁止条項を入れ、ソフトウェア使用者（ライセンシー）（以下「ユーザ」という。）に使用許諾したプログラムを、勝手にリバースエンジニアリングされることを禁止してきた。

ところが、平成 30 年の著作権法改正(注 6)により、新たに「柔軟な権利制限規定」が整備され、著作権法 30 条の 4 が改正された(注 7)。これにより、著作権法においても、リバースエンジニアリングが合法化された(注 8)。この主な理由は、ソフトウェアのセキュリティ対策や、今後の人工知能（Artificial Intelligence：AI）の利用促進、技術革新や古いプログラムの修正等にとって必要不可欠な技術と判断されたからである。しかし一方で、リバースエンジニアリングによるソフトウェア開発者のビジネスリスク、著作権法 30 条の 4 とソフトウェア開発者が従来使用してきて使用許諾契約書のリバースエンジニアリング禁止条項との関係、また独占禁止法上の不公正な取引方法との関係、さらにプログラムのアルゴリズムやノウハウが開示されることによる営業秘密の非公知性の喪失など、いくつか懸念事項がある。

本稿では、最初にプログラムのリバースエンジニアリングにおける必要性と法改正の趣旨を整理し、特許法と著作権法との対比から、開発者のリスクと契約書との関係、また独占禁止法との関係、さらに不正競争防止法の観点から、営業秘密の非公知性の喪失を検討するとともに、これらについて海外の法規制との対比を通じて考察を加え、若干ながら実務上の提言を行いたい。

Ⅱ．プログラムのリバースエンジニアリング

プログラムのリバースエンジニアリングは、新しいプログラムの研究や開発、既存のプログラムの脆弱性対策や障害対策、運用互換性の研究・調査などを主な目的とするが、これらに共通しているのがプログラムの解析と精度向上である。従前から、「プログラムにはバグ(注 9) はつきもの」と言われ、大規模なソフトウェアに安全性及び完全性を求めることは困難であった。その最大の理由は、プログラミング技術が属人的なものであり、現在は一部、プログラムの自動生成が実現化されてはいるものの、家内制手工業的な作成方法が中心だからである。

しかし、今や本格的な AI 時代を迎え、またすべての機器がネットワークで接続される IoT（Internet of Things）時代を迎えると、「プログラムにバグはつきもの」という常識は通用せず、各々のプログラムに安全性及び完全性が求められることになる(注 10)。例えば、自動車の自動運転に搭載する AI 機器のプログラムにバグが存在することは許されない。もしバグが一つでもあれば重大事故を引き起こしかねないからである。そのためには、プログラムの安全性及び完全性を検証し、システム障害を防ぐためには、どうしてもプログラムのリバースエンジニアリングによる解析・研究が必要不可欠となる。

特に喫緊の課題としては、ソフトウェアのセキュリティ対策がある。ネットワークに依存している現代社会において、コンピュータウイルスは脅威となっているが、コンピュータウイルスが蔓延しても、プログラムのリバースエンジニアリングが法的に曖昧で、事実上禁止されていたため、コンピュータウイルスそのものを解析することができず、その対策が不十分であった。そのため、プログラムのリバースエンジニアリングに関し、ソフトウェア業界

に委縮効果をもたらし、その結果、ソフトウェアセキュリティ対策が思うように進まなかったという経緯がある。

その次に大きな問題としては、既存のプログラムの障害対策がある。特に古いプログラムは、ソフトウェアベンダの消滅・不明、プログラム開発者の退職・死亡等により、保守もされないまま使い続けられているものが多い。このようなプログラムのソースコードは散逸し、仕様書もないものもある。これらの古いプログラムにいったん障害が発生すると、プログラムのリバースエンジニアリング無しで修復することは、ほとんど不可能に近い。このような古いプログラムは、数えきれないほど多く、現代社会は、古いプログラムの障害という目に見えない爆弾をいつも抱えた状態となっている。

このような背景の下、プログラムのリバースエンジニアリングについては、①技術的アイデアや情報を習得して、技術革新の手段を提供する、②互換性や相互接続性を保証する手段を提供する、③プログラムの補修やバグ除去の手段を提供する、④自社開発のプログラムの著作権を他社のプログラムが侵害しているか検証する手段を提供する、などからその必要性が議論されてきた(注 11)。

この問題に関して、プログラムのリバースエンジニアリングが適法か否かを早くから議論してきたのが米国である。米国は、プログラムのリバースエンジニアリングを適法とするために、米国著作権法 107 条を使用してきた。いわゆるフェアユース（公正利用）規定である。

米国著作権法 107 条（注 12）は、「批評、解説、ニュース報道、教授（教室における使用のために複数のコピーを作成する行為を含む）、研究又は調査等を目的とする著作権のある著作物のフェアユース（公正利用）は、著作権の侵害とならない。」と規定する。その判断基準は、①使用の目的および性質（使用が商業性を有するか又は非営利的教育目的かを含む）、②著作権のある著作物の性質、③著作権のある著作物全体との関連における使用された部分の量および実質性、及び④著作権のある著作物の潜在的市場または価値に対する使用の影響、の 4 つの考慮要素である。これら 4 つの考慮要素を各事案にあてはめ、総合的にフェアユース（公正利用）かどうかを判断することになる。

これら 4 つの考慮要素から実際にフェアユース（公正利用）かどうかを判断するのはユーザではなく裁判所である。過去の裁判例からの判断、いわゆる判例法主義（コモン・ロー）の先例拘束性に基づいた判断となる。プログラムのリバースエンジニアリングに関し、先例拘束性を有する代表的な過去の裁判例としては、Sega Enterprises Ltd. v. Accolade, Inc.事件（977 F.2d 1510 (9th Cir. 1992)（注 13)、Atari Games Corp. v. Nintendo of America Inc. 事件（975 F. 2d 832 （Fed. Cir. 1992)）(注 14)、DSC Communications Corp. v. DGI Techs. 事件（81 F. 3d 597 （5th Cir. 1996)）(注 15)などがある(注 16)。これらの事件は、いずれもプログラムを含む科学技術の進歩を促進するためのフェアユース（公正利用）として、プログラムのリバースエンジニアリングを認めた。

一方、わが国では、従前から、著作権法に米国著作権法 107 条に類似したフェアユース

（公正利用）規定を導入してはどうかという議論もあったが、わが国は大陸法系（シビル・ロー）の国であることから、そのまま米国著作権法 107 条に類似したフェアユース（公正利用）規定を導入することは難しく(注 17)、個別権利制限規定を定める条文の改正により対処せざるを得なかった(注 18)。そのため、平成 30 年の著作権法改正において、「柔軟な権利制限規定」を新設することとなり、その一つとして「著作物に表現された思想又は感情の享受を目的としない利用」（著作権法 30 条の 4）が整備され、著作権法においても、リバースエンジニアリングの適法性が明文化された。

ちなみに、法改正によって新設された「柔軟な権利制限規定」とは、「著作物に表現された思想又は感情の享受を目的としない利用」（著作権法 30 条の 4）のほか、「電子計算機における著作物の利用に付随する利用等」（同法 47 条の 4）及び「新たな知見・情報を創出する電子計算機による情報処理の結果提供に付随する軽微利用等」（同法 47 条の 5）の 3 つがある。これらは、いずれも「AI による深層学習」(注 19)、「所在検索サービス」(注 20) や「情報解析サービス」(注 21) 等の新たなニーズに対応するためのものである(注 22)。特に、著作権法 30 条の 4 の改正は、デジタル化・ネットワーク化の進展に対応したものとなっている。

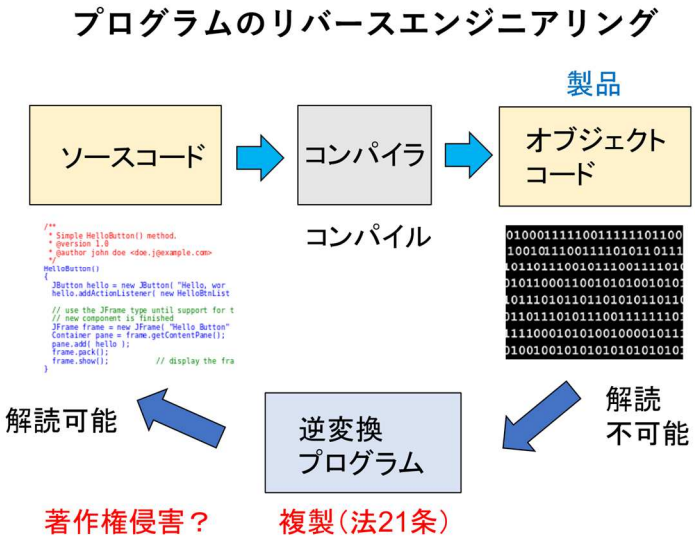
改正著作権法 30 条の 4 は、「①著作物利用に係る技術開発・実用化の試験、②情報分析、又は③その他人の知覚による認識を伴わない利用、に該当する場合、又はその他の当該著作物に表現された思想又は感情を自ら享受し又は他人に享受させることを目的としない場合には、その必要と認められる限度において、いずれの方法によるかを問わず、利用することができる。」と規定している。ただし、著作権者の利益を不当に害するものであってはならない。

この条文の中には、リバースエンジニアリング又はこれに類似する文言は見当たらないものの、リバースエンジニアリングによってオブジェクトコードを解析することは、著作物に表現された思想又は感情を自ら享受し又は他人に享受させることを目的としない場合に該当すると解釈され、リバースエンジニアリングは、プログラムの機能の享受を目的としない行為であるとした(注 23)。

しかし、プログラムのリバースエンジニアリングには、通常のハードウェア製品と異なり固有の問題が存在する。すなわち、プログラムのリバースエンジニアリングで問題となるのは、オブジェクトコードからソースコードへの変換（逆コンパイル又は逆アセンブル）が、プログラムの違法な複製（著作権法 21 条）又は翻案（同法 27 条）に当たる行為か否かである(注 24)。

一般に、ソフトウェア開発者は、ユーザにプログラムを使用許諾する場合、オブジェクトコードの形式でしか提供しない。近時では、インターネットを利用したクラウドコンピューティング(注 25)が一般的となってきているが、ユーザのサーバーにプログラムのオブジェクトコードをインストールするオンプレミス型(注 26)の使用形態は、未だに数多く存在する。特に、銀行の勘定系のような大規模ソフトウェアでは、オンプレミス型が一般的である。

プログラムは、人間が理解できるプログラム言語で記述されている。これをソースコードというが、このままではコンピュータは直接解読することができず、コンパイラと呼ばれる特殊なプログラムで、ソースコードをオブジェクトコードに変換する。オブジェクトコードは、「0」と「1」の 2 進による形式（機械語）であり、コンピュータはそれを読み取りプログラムを実行することができる。しかし、人間にとって、オブジェクトコードは「0」と「1」の羅列でしかなく、人間がこれを解読することは不可能である。そのため、当該プログラムが何をしているのか、どのように動作しているのかを知るために、オブジェクトコードからソースコードに逆に変換することが必要である。これがソフトウェアのリバースエンジニアリングである(注 27)。【図表】は、その概念図である。



【図表】プログラムのリバースエンジニアリングの概念図（筆者作成）

対象製品がハードウェアの場合、特許法 69 条 1 項により、リバースエンジニアリングは合法であることは明白であるが、平成 30 年の著作権法の改正前は、リバースエンジニアリングに関する明文規定がなかったため、オブジェクトコードからソースコードへのリバースエンジニアリングが、著作権法 21 条に規定する複製又は同法 27 条に規定する翻案に当たるかどうか長年議論されてきた(注 28)。

プログラムのリバースエンジニアリングを適法化する考え方の一つが、著作権は、元来表現を保護するものであり、表現の基礎にあるアイデアを保護するものではないため、プログラムのアイデア（例えば、アルゴリズム）を利用して別のプログラムを作成することは、著作権法違反とはならないという基本概念である。そのため、プログラムをリバースエンジニアリングして、プログラムのアイデアを入手し、それを基に新しいプログラムを研究・開発することは問題とはならない。ただし、許諾なく複製又は翻案をすることは著作権法上侵害行為となる。

ちなみに、著作権法には、プログラム著作物の複製又は翻案を一定の場合に認めた著作権法 47 条の 3 があるが、これは、プログラムを自分のコンピュータにインストールする場合など、プログラムを利用するために必要な複製又は翻案を予定したものであり、プログラムのリバースエンジニアリングのように、プログラムを調査・分析・研究するための複製又は翻案を予定したものではない(注 29)。

なお、特許法では、プログラムも「発明」の対象とされており、技術的思想の創作のうち高度なものについては、特許を出願・登録した場合には、特許法上の保護を受けることができる（特許法 2 条 1 項、3 項)。このため、プログラムが特許法の対象となっていれば、特許法 69 条 1 項により、プログラムのリバースエンジニアリングも適法となる。

しかしながら、ソフトウェアベンダとしては、苦労して作成したプログラムの内部構造やアルゴリズムを勝手に盗用されたくないというのは至極当然の願望であり、そのためユーザに対する使用許諾契約書にリバースエンジニアリング禁止条項を入れてきたという長年の経緯及び慣習がある。いわゆるプロプライエタリソフトウエア(注 30)である。

プログラムのリバースエンジニアリングの適法性に関して、過去の代表的な事件として昭和 62 年のマイクロソフト秀和事件(注 31)がある。この事件では、裁判所は、「本件オブジェクトプログラムを逆アセンブルしたうえ、解読して、ラベル及びコメントを付したこと」につき、「被告秀和が本件オブジェクトプログラムを逆アセンブルして、解読したものにラベル及びコメントを付した行為は、本件著作物の複製行為であると評価することができる」と判示した。

ただし、この判決は、「解読したものにラベル及びコメントを付した行為」、つまり、ラベル及びコメントを付したソースコードを記載した出版物の作成行為について判断したものであり、プログラムのリバースエンジニアリングが著作権法 21 条に該当し、オブジェクトコードからソースコードに変換する行為を複製と認め、プログラムのリバースエンジニアリングを違法だとは明言しておらず、プログラムの調査・解析行為それ自体が複製行為に該当するか否かについて判断したものとは言いがたい(注 32)。

その後の裁判例として、リバースエンジニアリングが複製・翻案にあたるとしても、具体的事情のもとでは、複製・翻案という著作権侵害に基づく損害賠償請求が、民法上の権利の濫用（民法 1 条 3 項）として許されないとした判決もある(注 33)。これは、権利の濫用という一般規定によって解決を試みたものであり、これもリバースエンジニアリングが著作物の複製・翻案に当たるか否かについて明確な基準を示したものではない(注 34)。

このような議論及び背景の下、プログラムのリバースエンジニアリングが必要不可欠として、著作権法 30 条の 4 が改正され、プログラムのリバースエンジニアリングが合法化された。

Ⅲ．著作権法 30 条の 4 とリバースエンジニアリング禁止条項

平成 30 年に著作権法が改正され、著作権法 30 条の 4 によりプログラムのリバースエン

ジニアリングが適法となったが、プログラムの使用許諾契約書の中のリバースエンジニアリング禁止条項は有効であろうか。いわゆる「著作権オーバーライド」の問題である。この問題は、改正著作権法 30 条の 4 が強行規定なのか任意規定なのかという問題と等価である。

この問題に関して、わが国では、特許法 69 条 1 項を類推適用するという強行規定説、独占禁止法からの公正競争阻害説（強行規定説に該当）、契約は認めるが著作権法 20 条 2 項 3 号(注 35)は効力がないとする説など各種の説があるが、わが国では、プログラムの使用許諾契約書の中のリバースエンジニアリング禁止条項は有効という説（任意規定説）が多数説を占める(注 36)。

このように、この問題に対し未だ確定した解釈はないものの、経済産業省は、プログラムのリバースエンジニアリング禁止条項を有効と考えるような判断を示している。平成 30 年 6 月に公表された「AI・データの利用に関する契約ガイドライン（AI 編）」(注 37)の中に、人工知能（AI）に関する「ソフトウェア開発契約書」がモデル契約書として掲載されているが、このモデル契約書 19 条(注 38)に、リバースエンジニアリング及び再利用等の生成の禁止条項がある。

当該モデル契約書 19 条は、「【ユーザ／ベンダ】は、本契約に別段の定めがある場合を除き、本件成果物について、次の各号の行為を行ってはならない。」とし、第 1 号で「リバースエンジニアリング、逆コンパイル、逆アセンブルその他の方法でソースコードを抽出する行為」を挙げている。この条項の存在は、明らかに著作権法 30 条の 4 の強行規定説を否定するもので、契約書上のリバースエンジニアリング禁止条項が有効であることを暗に示している(注 39)。

海外はどうであろうか。ソフトウェア開発の先進国である米国では、2003 年に重要な判決が出された。Bowers v. Baystate Technologies, Inc 事件（320 F. 3d 1317（Fed. Cir. (Mass.) 2003））(注 40)である。この事件は、米国著作権法でフェアユース（公正利用）として判断されたリバースエンジニアリングが、使用許諾契約において、それを禁止することが適法か否かが争われた事件である(注 41)。

この事件では、裁判所は、契約上のリバースエンジニアリング禁止条項が著作権法に優先するとした。ところが、ここで問題となったのが、「連邦法が州法である契約法に優先する」という一般原則である。米国では、知的財産に関しては、アメリカ合衆国憲法 1 条 8 節（注 42）において、連邦政府の権限とされており、米国著作権法も連邦法である。一方、契約法は元々コモン・ロー（判例法）であり、州の権限によって州法として契約法が整備されている。よって、本判決は、「連邦法が州法である契約法に優先する」という一般原則に反するのではないかという問題が生じた(注 43)。

最終的に裁判所は、「すべての連邦法が契約の内容に優先するのではなく、目的によって優先適用の可否を判断する。」として、リバースエンジニアリング禁止条項を有効とした。さらに、米国著作権法には、リバースエンジニアリングを禁止する規定は存在しないという

理由もあげられる。これ以降、米国では、プログラムの使用許諾契約書の中のリバースエンジニアリング禁止条項は、適法と解釈されている。

これと全く対照的なのが EU である。EU では、一律に、契約書の中のリバースエンジニアリング禁止条項を無効としている。その根拠は、2009 年の「コンピュータ・プログラムの法的保護に関する EU 指令」(注 44)（European Union Communities Council Directive of 23 April 2009 on the Legal Protection of Computer Programs）（Directive2009/24/EC）6 条（Decompilation）と 8 条（Continued application of other legal provisions）である。同指令 6 条は、逆コンパイル、すなわちリバースエンジニアリングについて規定しており、同指令 8 条第 2 文で、”Any contractual provisions contrary to Article 6 or to the exceptions provided for in Article 5(2) and (3) shall be null and void.”と、リバースエンジニアリングを禁止する契約条項を明確に無効としている(注 45)。

このように米国と EU とでは、プログラムのリバースエンジニアリング禁止条項に対して相反する立場をとっている。この理由は、ソフトウェアの先進国である米国と、ソフトウェア開発に遅れをとっている EU との産業政策上の違いがその背景にあると思われる。EU としては、プログラムのリバースエンジニアリングによって、米国の先進的なソフトウェア技術を取り入れたいと思う反面、米国は、ソフトウェア開発者が苦労して作成したプログラムの内部構造やアルゴリズムを盗用されたくないという思惑があるのではないだろうか。

Ⅳ．独占禁止法とリバースエンジニアリング

プログラムのリバースエンジニアリング禁止条項に関して、米国は有効、EU が無効としているが、わが国は、多数説において有効としている。これに対し、独占禁止法の観点からは、ユーザに対して契約によってリバースエンジニアリングを禁止することが、不公正な取引方法に該当するかどうかが問題となる。

独占禁止法に規定する不公正な取引方法は、公正取引委員会が昭和 57 年に告示した「不公正な取引方法」(注 46)に記載されているが、不公正な取引方法全 15 項目のうち、契約書のリバースエンジニアリング禁止条項が「一般指定 12 項：拘束条件付取引」に該当するかどうかである。「拘束条件付取引」とは、相手方とその取引の相手方の事業活動を不当に拘束する条件をつけて、当該相手方と取引をすることをいう。すなわち、リバースエンジニアリング禁止条項が、取引の相手方の事業活動を不当に拘束する条件となっているかどうかが問題となる。

この問題に関して、公正取引委員会の「ソフトウェアと独占禁止法に関する研究会」で議論を重ねた結果、平成 14 年に「ソフトウェアライセンス契約等に関する独占禁止法上の考え方－ソフトウェアと独占禁止法に関する研究会中間報告－」(注 47)が公表された。この報告では、「当該ソフトウェアとインターオペラビリティを持つソフトウェアを開発するためには、①当該ソフトウェアのインターフェース情報が必要であり、②ライセンサーが当該インターフェース情報を提供しておらず、③ライセンシーにとって、リバースエンジニアリン

グを行うことが、当該ソフトウェア向けにソフトウェアやハードウェアを開発するために必要不可欠な手段となっているような場合においては、リバースエンジニアリングを禁止することは、ソフトウェアにノウハウが含まれる場合があり、また、仮に外形上又は形式的には著作権法上の権利の行使と見られる行為であるとしても、著作権法上の権利の行使と認められる行為とは評価されず、独占禁止法が適用されるものと考えられる。」としている(注 48)。

このように公正取引委員会は、プログラムのリバースエンジニアリング禁止条項を、不公正な取引方法（拘束条件付取引）としている。この根拠は、近時の通説的学説である「再構成された権利範囲論」(注 49)である。知的財産権と独占禁止法がどのような関係にあるかを明示的に規定した条文が、独占禁止法 21 条であるが、同法 21 条は、「この法律の規定は、著作権法、特許法、実用新案法、意匠法又は商標法による権利の行使と認められる行為にはこれを適用しない。」と規定している。これは、独占禁止法の適用除外制度(注 50) の一つとして設けられたものである(注 51)。

知的財産権の行使を本来的行使（知的財産法による権利の行使と認められる行使）と非本来的行使（知的財産法による権利の行使と認められない行使）に分けると、独占禁止法 21 条の条文を反対解釈すれば、本来的行使については、独占禁止法の適用が除外されるが、非本来的行使については、独占禁止法が適用されるという見解が成り立つ（権利範囲論（注 52)）。これを発展させたものが、近時の通説である「再構成された権利範囲論」である。すなわち、プログラムのリバースエンジニアリングを契約上で禁止する行為は、知的財産権の本来的行使ではなく、非本来的行使とし、独占禁止法が適用されるとしている。

さらに、平成 19 年の「知的財産の利用に関する独占禁止法の指針」（最終改正：平成 28 年 1 月 21 日）(注 53) は、研究開発活動の制限として、「ライセンサーがライセンシーに対し、ライセンス技術又はその競争技術に関し、ライセンシーが自ら又は第三者と共同して研究開発を行うことを禁止するなど、ライセンシーの自由な研究開発活動を制限する行為は、一般に研究開発をめぐる競争への影響を通じて将来の技術市場又は製品市場における競争を減殺するおそれがあり、公正競争阻害性を有する(注 54)。したがって、このような制限は原則として不公正な取引方法に該当する（一般指定第 12 項)。」とし、研究開発を制限するような行為（リバースエンジニアリング禁止条項）は、著作権の本来的行使とは認められないとしている。

このように、平成 30 年の著作権法改正前ではあるが、公正取引委員会の各種の報告書において、契約上のリバースエンジニアリング禁止条項の有効性については疑問視されており、独占禁止法に規定する不公正な取引方法（拘束条件付取引）に該当するとしている。公正取引委員会の見解は、「プログラムのリバースエンジニアリング禁止条項は有効である。」という多数説や、経済産業省のプログラムのリバースエンジニアリングに対する考え方と相反するものであり、この問題の奥深さを物語っていると言えるであろう。

このような議論の基に、改めてプログラムのリバースエンジニアリング禁止条項が、不公

正な取引方法である拘束条件付取引かどうかを考える場合、プログラムのリバースエンジニアリングが著作権の本来的行使であるかどうかを検討する必要がある。すなわち、プログラムのリバースエンジニアリングが、①ライセンシーの自由な研究開発活動のためのものか、②当該著作物に表現された思想又は感情を自ら享受し又は他人に享受させることを目的としないものであるかどうか、また③著作権者の利益を不当に害しないものかどうか、を総合的に判断し、プログラムのリバースエンジニアリングが著作権の本来的行使であるかどうかを個々の事案ごとに検討するほかないのではないだろうか。

Ⅴ．営業秘密の非公知性との関係

ソフトウェア開発者が作成した高度なプログラムには、様々なアルゴリズムやノウハウが含まれており、プログラムの処理能力を向上させている。特に、コンピュータの処理スピードは、ハードウェアの処理能力だけでなく、プログラムの作成方法に大きく依存する。このように、プログラムの作成方法に関わるプログラミング技術は、不正競争防止法上の営業秘密の要件を満たしていれば、営業秘密として保護され得るものである。

ところが、著作権法 30 条の 4 によりリバースエンジニアリングが適法になったことで、営業秘密の要件の一つである非公知性の喪失が懸念される。不正競争防止法では、①秘密管理性、②有用性、③非公知性、の 3 つが営業秘密の要件であるが(注 55)、非公知性の喪失により、ユーザに使用許諾されたプログラムが営業秘密に該当せず、いくら他の要件である秘密管理性及び有用性を満たしていたとしても、不正競争防止法上、営業秘密として保護されない。

非公知性が認められるためには、一般的に知られておらず、又は容易に知ることができないことが必要であるが(注 56)、非公知性の基準について、不正競争防止法上の営業秘密の要件は、特許法 29 条の「公然知られた発明」と異なる。情報を開示された者の違いとしては、特許法では、守秘義務のある特定の者が知っている場合は「非公知」となるが、守秘義務のない特定の者が知っている場合は「公知」となる。一方、不正競争防止法では、特定の者が知っている場合であっても「非公知」となり、「非公知」の範囲は広い。また、判断基準の「時期」の違いとしては、特許法では、特許出願時であるのに対し、不正競争防止法では、不正行為が行われた時である。このように、同じ「公知」・「非公知」であっても、特許法と不正競争防止法では、その基準は異なる(注 57)。

このような非公知の基準であるが、1990 年の「営業秘密逐条解説　改正不正競争防止法」(注 58) は、リバースエンジニアリングの非公知性について、「リバースエンジニアリングといっても誰でも簡単に製品を解析することによって営業秘密を取得できるような場合には、当該製品を市販したことによって営業秘密自体を公表したに等しいと考えられることから、非公知性を失った情報となると考えられる。」と言及している。さらに、「これに対して、リバースエンジニアリングによって営業秘密を取得することができるといっても、特殊な技術をもって相当な期間が必要であり、誰でも容易に当該情報を知ることができない場合に

は、製品を市販したことをもって営業秘密が公知化することにはならない。」としている。すなわち、①特殊な技術を要するかどうか（特殊技術基準）、②相当な期間が必要かどうか（相当期間基準）、③誰でも容易に知ることができるかどうか（情報取得容易性基準）の 3 つが公知又は非公知性の判断基準と考えられている。

このような判断基準から、公知と判断されたものに、ブロワ用サイレンサ事件(注 59)、糸半田供給機事件(注 60)、広告宣伝カー事件(注 61)、錫合金組成事件(注 62)などがある。また、非公知と判断されたものに、アルミナ繊維事件(注 63)、婦人靴木型事件（注 64)、セラミックコンデンサー事件(注 65)などがある。これらは、いずれも、裁判所が個々の事案に、特殊技術基準、相当期間基準及び情報取得容易性基準を用いて判断したものである(注 66)。

特に、この問題に関しての先駆的な判決であるセラミックコンデンサー事件では、原告のセラミックコンデンサー積層機及び印刷機の設計図（合計約 6,000 枚）に係る情報の非公知性について、裁判所は、「原告のセラミックコンデンサー積層機及び印刷機のリバースエンジニアリングによって、本件電子データと同じ情報を得るのは困難であるものと考えられ、また仮にリバースエンジニアリングによって本件電子データに近い情報を得ようとすれば、専門家により、多額の費用をかけ、長時間にわたって分析することが必要であるものと推認される。したがって、本件電子データは、原告のセラミックコンデンサー積層機及び印刷機の相当台数が秘密保持契約なしに販売されたことによって公知となったとは言えない。」と判示した(注 67)。

これをプログラムのリバースエンジニアリングに当てはめるとどうなるであろうか。まず、特殊な技術を要するかどうか（特殊技術基準）であるが、前述のように、オブジェクトコードからソースコードに変換する技術が、特殊技術か否かである。これに関しては、プログラムをリバースエンジニアリングするツールは、すでに多く出回っており、一般に市販されている。

例えば、Java というプログラム言語で書かれたプログラムのオブジェクトコードをソースコードにリバースエンジニアリングするプログラム・ツールとしては、JProfiler がある。価格も 10 万円以下で購入できる。リバースエンジニアリングすることが難しいと言われている C/C++というプログラム言語であっても、Imagix4D というリバースエンジニアリング用のプログラム・ツールを使うことによって、かなりの正確さでオブジェクトコードをソースコードに変換することができる。また、2019 年 3 月、米国家安全保障局（National Security Agency：NSA）は、組織内で 10 年以上使用してきたプログラムのリバースエンジニアリングツール「Ghidra」を無償で公開した。このようにプログラムのリバースエンジニアリングは一般化しており、今後プログラムのリバースエンジニアリングが一気に進むと思われる。

次に、リバースエンジニアリングで復元されたソースコードの解読であるが、これは解読するエンジニアの技術力によって大きく異なる。ベテランのエンジニアは、プログラムのデバッグ等でソースコードの解読は、日常的に行っているので、さほど問題とはならないであ

ろう。ソースコードを、フローに落とし込むことさえできれば、どのような内部構造で、どのようなアルゴリズムを使用しているかがわかる。

次に、相当な期間が必要かどうか（相当期間基準）であるが、これはリバースエンジニアリングするオブジェクトコードの大きさやコンピュータの処理速度によって異なる。ただし、昨今のコンピュータの処理能力をもってすれば、さほど時間がかかるものではない。少々時間がかかったとしても、コンピュータがリバースエンジニアリングしている間、人間の作業をほとんど必要としないことを考えれば、リバースエンジニアリング用のプログラム・ツールさえ入手できれば、特殊技術基準と相当期間基準をともにクリアすることができ、オブジェクトコードが市販された時点で、プログラムの非公知性が喪失すると考えることができよう。

三つ目の、誰でも容易に知ることができるかどうか（情報取得容易性基準）であるが、上記の特殊な技術を要するかどうか（特殊技術基準）、及び相当な期間が必要かどうか（相当期間基準）をクリアすることができれば、容易に知ることができると考えられる。ただし、特殊技術基準及び相当期間基準をクリアするためには、ある程度経済的な余裕及び技術力が必要となる。経済的に余裕のない者にとって、プログラムをリバースエンジニアリングするための高価なリバースエンジニアリングツールを入手することは難しく、またそれを使ってリバースエンジニアリングする技術力もない者が、「誰でも容易に」の「誰でも」と言えるかどうか疑問である。

これらの問題は、事案ごとに判断が異なる可能性がある。もし仮に、非公知性を喪失した場合、いくらソフトウェア開発者が、プログラムの使用許諾契約書の中に、リバースエンジニアリング禁止条項を入れたとしても、ユーザがリバースエンジニアリングを行えば、契約上の違反とはなるが、不正競争防止法上の営業秘密に該当しなくなり、不正競争防止法による保護を受けることが難しくなるのではないだろうか。

VI. 今後の実務的対応

ソフトウェア開発者は、処理能力の優れたプログラムを作成するために、自らの技術を駆使してアルゴリズムを考え、それをプログラムに実装する。そこには、各種のノウハウも含まれる。これら苦労して考案したプログラミング技術を盗まれないため、ソースコードを非公開とし、ユーザにはオブジェクトコードのみを提供している。

しかし、著作権法 30 条の 4 の改正により、プログラムのリバースエンジニアリングが適法となり、プログラムのリバースエンジニアリングによりオブジェクトコードからソースコードを入手することが可能となった現在、ソフトウェア開発者がソースコードを非公開にするというメリットは少なくなった。

また、プログラムの使用許諾契約書にリバースエンジニアリング禁止条項を入れたとしても、この禁止条項が、独占禁止法上の拘束条件付取引とみなされ、無効となる可能性もある(注 68)。さらに、リバースエンジニアリングのしづらいプログラム言語を使用したとして

も、高度なプログラムリバースエンジニアリングツールが市場に出回っている状況下では、その効果も限定的である。このように、著作権法 30 条の 4 の改正により、ソフトウェア開発者は、ビジネス上不利な立場に置かれたといえるであろう。

これを解決するための方法は、二つ考えられる。一つは、オンプレミス型の提供を止めて、すべてクラウドコンピューティングに変更することであり、もう一つは、作成したプログラムについて特許を取得することである。これには一長一短があるものの、どうしても開示したくない、又は模倣されたくないプログラムの場合には、有効な手段となるであろう。

プログラムによる処理をすべてサービスとして提供するクラウドコンピューティングは、ユーザのサーバーにオブジェクトコードをインストールする必要がないので、実質的に、ユーザがリバースエンジニアリングすることができない。なぜなら、ユーザ自身のサーバー等にオブジェクトコードがインストールされていないからである。これによって、ソフトウェア開発者はソースコードの開示を免れる。ただし、クラウドコンピューティングを行うには、ソフトウェア開発者の方で、サーバー等のハードウェアやネットワークを準備しなければならない。近時、このようなクラウドコンピューティングは盛んに導入されているが、銀行の勘定系システム等の大規模なシステムを、すべてクラウドコンピューティングに置き換えるのは、多大な労力、時間及びコストがかかる。しかしながら、このような方法は、今後益々進展する傾向にあり、これによってプログラムのリバースエンジニアリングに対抗できることになろう。

もう一つの方法は、プログラムを特許出願するという方法である。ただし、特許を取得したとしても、プログラムのソースコード及び内部構造が公開されるので、リバースエンジニアリングを阻止するということにはならない。却って、自らソースコードを公開することになる。しかし、この技術を使ったプログラムの独占権が一定期間得られるため、不正盗用や模倣を防止することができる。ただし、通常の特許出願と同じ手続きを踏むので、時間、労力及びコストがかかる。この方法は、クラウドコンピューティングに比べると、費用対効果から考えても、あまり勧められないが、ソフトウェア開発者のプログラムを法的に保護するという点では有効な方法であると思われる。

一方、ユーザにとってみれば、著作権法 30 条の 4 の改正により、プログラムのリバースエンジニアリングが適法になったことで、使用許諾を受けたプログラムのアルゴリズムやノウハウが適法に入手できることになる。仮に、ソフトウェア開発者が、その内容を営業秘密であると主張しても、リバースエンジニアリングが容易であれば、ユーザは、営業秘密の要件の一つである非公知性が喪失していると主張することが可能となる。このように、著作権法 30 条の 4 は、ユーザにとって大きなメリットである。

ただし、ユーザが気を付けなければならないのが、著作権法 30 条の 4 の但書である「当該著作物の種類及び用途並びに当該利用の態様に照らし著作物の利益を不当に害することとなる場合」である。これに該当した場合には、明らかに著作権法違反となるので注意が必要である。

さらに、著作権法 30 条の 4 によって、プログラムのリバースエンジニアリングが可能になったことにより、セキュリティ対策ができるようになった。特に、ウイルス対策には大きな力を発揮できることが期待される。このセキュリティ対策に関し、令和元年 5 月、経済産業省は、「情報セキュリティ早期警戒パートナーシップガイドライン」(注 69)を公表した。ちなみに、当該ガイドラインは、情報処理推進機構（Information-Technology Promotion Agency：IPA (注 70)）と JPCERT/CC（JPCERT コーディネーションセンター）が「情報システム等の脆弱性情報の取扱いに関する研究会」での検討結果を踏まえて作成したものである。

これによると、ユーザがプログラムのリバースエンジニアリングによって、プログラムの脆弱性を発見した場合、情報処理推進機構 (IPA) にこれを届け出なければならない(注 71)。情報処理推進機構（IPA）は発見者のプライバシーが保護されるように、匿名で相手方にその内容を伝える。これは無用な訴訟リスクを回避するためである。このように情報処理推進機構（IPA）にプログラムの脆弱性情報を届けることによって、安心してプログラムのリバースエンジニアリングを行い、関連プログラムのセキュリティ対策を講じることができるようになった。

ソフトウェア開発者によるクラウドコンピューティングが進展していけば、ソースコードからリバースエンジニアリングによりソースコードを入手されることはなくなるが、それによりプログラムがどのサーバーにインストールされているのかが、ユーザから見えなくなり、セキュリティ対策はクラウドコンピューティングベンダ（ソフトウェア開発者）に全面的に委ねられることになる。そのため、クラウドコンピューティングベンダのセキュリティに対する責任が増大することも、ソフトウェア開発者の大きなリスクとも言えるであろう。

また、ソフトウェア開発者にとって注意を要することは、海外のユーザに対する対応である。特に、使用許諾契約の中のリバースエンジニアリング禁止条項の有効性については、米国と EU とでは異なることである。現地での取り扱い方法や、契約書上での準拠法については注意が必要であろう。

いずれにせよ、プログラムのリバースエンジニアリングが合法化されたことにより、ソフトウェア開発者が不利な立場に置かれる反面、セキュリティ対策等にとっては、非常に有効な手段となる。ソフトウェア開発者は、これらを踏まえた上で、ソフトウェアビジネスを展開する必要があろう。

Ⅴ．おわりに

世はまさにコロナ禍の時代であり、これを機に益々在宅勤務が増え、コンピュータネットワーク及びプログラムに依存した時代となろう。著作権法は、もともと表現を保護するものであり、表現の基礎にあるアイデアを保護するものではないことを思えば、プログラムのリバースエンジニアリングによって、プログラムに含まれるアイデア（アルゴリズムやノウハ

ウ）を入手することは、法的には問題がないことのように思うが、ソフトウェア開発者にしてみれば、自身の技術力の結晶であるプログラムの内部構造を開示することにはためらいもあろう。

著作権法 30 条の 4 により、プログラムのリバースエンジニアリングが適法となったが、それを契約で禁止する禁止条項の有効性や、営業秘密の要件の一つである非公知性の喪失など、法的に未整備な部分が存在する。これらは、産業政策的な側面もあることから、時代の要請とともに変化することが考えられるが、これらを曖昧にすることによって、ソフトウェア業界に新たな混乱を招くことにも繋がりかねず、明確な対応が望まれる。

今回の著作権法の改正により、ソフトウェア開発者にとっては、不利な立場に置かれ、盗用による経済的不利益を被るなどのビジネスリスクが増大することになったが、クラウドコンピューティングや特許制度の利用により、ソースコードの開示や、プログラムの模倣を回避することができる道も残されている。一方で、これによりセキュリティ対策は大幅に改善され、古いプログラムの障害対策にも力を発揮することになろう。この現実をどう評価するかは、今後の IT 社会の進展にかかってくると言えよう。

いずれにせよ、リバースエンジニアリングに関しては、ソフトウェアもハードウェアと同じ土俵に立ったと言えるが、解決すべき問題点はそのままであり、無用な混乱を招かないためにも、統一的な一定の指針が必要ではないだろうか。

（脚注）

(注 1) 慶長 11 年（1606 年）、種子島久時が薩摩国大竜寺の禅僧南浦玄昌に編纂させた鉄砲伝来に関する書。

(注 2) 逆行工学。リバースアナリシス（Reverse Analysis）。一般に、リバースエンジニアリングという用語が使用されていることから、本稿でも、リバースエンジニアリングという用語を用いることとする。

(注 3) リバースエンジニアリングで得られる情報には、特許、著作権、意匠権などの知的財産権で保護されているものが多く存在する。これらを、そのまま使用すると違法となる可能性がある。

(注 4) 著作権法 10 条 1 項 9 号。なお、「プログラム」に対し「ソフトウェア」という用語があるが、本稿では法律用語である「プログラム」を使用する。ただし、一部「プログラム」の集合体として「ソフトウェア」という用語を使用する。

(注 5) 小倉秀夫＝金井重彦編『著作権法コンメンタールⅡ』（第一法規、2020 年）63～64 頁。

(注 6) 「著作権法の一部を改正する法律」が、第 196 回通常国会（平成 30 年 5 月 18 日）にて成立し、同年 5 月 14 日に平成 26 年法律第 35 号として公布された。本法律は、一部の規定を除いて、平成 31 年 1 月 1 日に施行された。

(注 7) 文化庁「著作権法の一部を改正する法律案　概要説明資料（AI の利活用促進関係）」（平成 30 年 4 月）4～5 頁
（http://www.kantei.go.jp/jp/singi/titeki2/tyousakai/kensho_hyoka_kikaku/2018/contents/dai4/siryou6.pdf）（as of Dec 29, 2020）。

(注 8) 著作権法 30 条の 4 には、リバースエンジニアリングを適法とするという文言はないが、条文全体の解釈から、リバースエンジニアリングを適法とすることについて争いはない。

(注 9) バグ（bug）は、本来「虫」の意味であるが、プログラムのエラー（不具合）を指す。
(注 10) 平成 30 年に、工業標準化法（JIS 法）が大幅に改正され、新たに産業標準化法がスタートした。また、これに伴い、「日本工業規格（JIS)」は「日本産業企画（JIS)」に変わった。プログラムも新たに標準化を一層進めることになったが、平成 30 年の著作権法改正も、これに連動したものであると思われる（https://www.meti.go.jp/policy/economy/hyojun-kijun/jisho/jis.html）(as of Dec 28, 2020)。
(注 11) ソフトウェアに対するリバースエンジニアリングの必要性については、文化庁「コンピュータ・プログラムに係る著作権問題に関する調査研究者会議報告書－既存プログラムの調査・解析等について－」(平成 6 年) (https://www.cric.or.jp/db/report/h6_5/h6_5_main.html)（as of Nov 2020)、知的財産戦略本部「デジタル・ネット時代における知財制度の在り方について」(平成 20 年) (https://www.kantei.go.jp/jp/singi/titeki2/houkoku/081127digital.pdf)（as of Nov 30, 2020)、知的財産戦略本部「知的財産推進計画 2008－世界を睨んだ知財戦略の強化－」（平成 20 年）(https://www.kantei.go.jp/jp/singi/titeki2/2008keikaku.pdf)（as of Nov 30, 2020）などにおいて、長年議論されてきたという経緯がある。
(注 12) 17 U.S.C.§107. 著作権侵害に対する一つの抗弁として利用されている。
(注 13) Sega Enterprises, Ltd. v. Accolade, Inc., 1992 U.S. Dist. LEXIS 4621.
(注 14) Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 1992 U.S. App. LEXIS 21817, 24 U.S.P.Q.2D (BNA) 1015, Copy. L. Rep. (CCH) P26,978, 1992-2 Trade Cas. (CCH) P69,969, 92 Cal. Daily Op. Service 7858, 92 Daily Journal DAR 12936.
(注 15) DSC Communs. Corp. v. DGI Techs., Inc., 81 F.3d 597, 1996 U.S. App. LEXIS 10017, 38 U.S.P.Q.2D (BNA) 1699, Copy. L. Rep. (CCH) P27,513.
(注 16) 鄭鎭根「コンピュータ・プログラムの保護に関する米・EU・日・韓の比較法的研究－プログラム リバース・エンジニアリングを中心に－」知財権紀要 15 巻（2006 年）102 頁。
(注 17) わが国の著作権法にフェアユース（公正利用）を導入することについて否定的な論文として、高田寛「デジタルコンテンツの流通とフェアユースについての一考察」国際商取引学会年報 13 号（2011 年）259～272 頁、がある。
(注 18) 日本版フェアユースは、「著作物の付随的な利用（A 類型)」、「適法利用の過程における著作物の利用（B 類型)」及び「著作物の表現を享受しない利用（C 類型)」の 3 類型に権利制限の一般規定を導入しようとするものであったが、最終的には、本来の意味でのフェアユース（公正利用）規定を導入することは見送られた。
(注 19) AI に大量の情報を入力して分析させ、人間のサポート無しにそれらの情報がなんであるか等を判断できるようにする学習方法。
(注 20) 広く公衆がアクセス可能な情報の所在を検索可能にするとともに、その一部を検索結果と併せて表示するサービス。
(注 21) 広く公衆がアクセス可能な情報を収集し、求めに応じて解析結果を提供するサービス。
(注 22) これらは「文化審議会著作権分科会報告書」（平成 29 年 4 月）（https://www.bunka.go.jp/seisaku/bunkashingikai/chosakuken/pdf/h2904_shingi_hokokusho.pdf）(as of Nov 30, 2020）を踏まえて整備されたものである。なお、最新の「文化審議会著作権分科会報告書」は、平成 31 年 2 月版のものである。
(注 23) 小倉＝金井・前掲注(5) 63～64 頁。
(注 24) プログラムのリバースエンジニアリングで作成されたソースコードが、元のソースコードと完全に一致しているとは限らないが、実質的に同じものと解されている。
(注 25) Cloud Computing. インターネットなどのコンピュータネットワークを経由して、コンピュータ資源をサービスの形で提供する利用形態。オンプレミス型と異なり、ユーザのサーバー等にプログラムをインストールすることはしない。
(注 26) 自社で用意したサーバーへプログラムをインストールし、それを利用する形態を指す。
(注 27) プログラムのリバースエンジニアリングを、逆コンパイル又は逆アセンブルともいう。
(注 28) 文化庁「リバース・エンジニアリングに係る法的課題について」（平成 20 年）（https://www.bunka.go.jp/seisaku/bunkashingikai/chosakuken/hosei/h20_07/shiryo_1.html）(as of Dec 28, 2020)。

(注 29) 吉田秀平「プログラム・リバースエンジニアリングの法律問題」(法務・税務・労働などの問題解決エンジン)(2014 年 7 月 7 日)(http://kasiko.me/プログラム・リバースエンジニアリングの法律問/)(as of Dec 27, 2020)。
(注 30) プロプライエタリソフトウエアとは、ソフトウェアベンダが、ユーザの持つ権利を制限的にすることで自身や利用者の利益およびセキュリティを保持しようとするソフトウェアをいう。
(注 31) 東京地判昭 62・1・30 判時 1219 号。
(注 32) 吉田・前掲注(29)。
(注 33) 大阪地判平 21・10・15 裁判所 HP (https://www.courts.go.jp/app/files/hanrei_jp/146/038146_hanrei.pdf)(as of Dec 28, 2020)。
(注 34) 吉田・前掲注(29)。
(注 35) 特定の電子計算機においては実行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実行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果的に実行し得るようにするために必要な改変。
(注 36) 福岡真之介『AI 開発のための法律知識と契約書作成のポイント』(清文社, 2020) 98 頁。
(注 37) 経済産業省「AI・データの利用に関する契約ガイドライン(AI 編)」(https://www.meti.go.jp/press/2018/06/20180615001/20180615001-3.pdf)(as of Dec 26, 2020)。
(注 38) 経済産業省・前掲注(37) 119 頁。
(注 39) 著作権法改正と「AI・データの利用に関する契約ガイドライン(AI 編)」の公表は、ほぼ同時期に行われている。
(注 40) Bowers v. Baystate Techs., 320 F.3d 1317, 2003 U.S. App. LEXIS 1423.
(注 41) 鄭・前掲注(16) 103 頁。
(注 42) アメリカ合衆国憲法 1 条 8 節 8 項は、連邦政府の権限として、「著作者又は発明者に、一定期間それぞれの著作及び発明に対し独占的権利を保障することによって、学術及び技芸の進歩を促進すること。」と規定している。
(注 43) 鄭・前掲注(16) 103 頁。
(注 44) https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32009L0024&from=EN(as of Jan 14,2021)。
(注 45) 鄭・前掲注(16) 104 頁。
(注 46) 公正取引委員会「昭和 57 年 6 月 18 日公正取引委員会告示第 15 号」(最終改正：平成 21 年 10 月 28 日　公正取引委員会告示第 18 号) (https://www.jftc.go.jp/dk/guideline/fukousei.html)(as of Dec 26, 2020)。
(注 47) 公正取引委員会『ソフトウェアライセンス契約等に関する独占禁止法上の考え方－ソフトウェアと独占禁止法に関する研究会中間報告－』(平成 14 年 3 月 20 日)(公正取引委員会、2002 年)。
(注 48) 公正取引委員会・前掲注(47) 28～29 頁。
(注 49) 根岸哲「知的財産権と独占禁止法」経済法学会 10 号(1989 年) 21 頁。根岸哲＝舟田正之『独占禁止法概説(第 3 版)』(有斐閣、2006 年)。根岸哲「『競争法』としての民法。知的財産法、独占禁止法」曹時 56 巻 1 号 9 頁。
(注 50) 適用除外制度の法的性格については、確認説と創設説に大きく分けられる(岩本章吾『知的財産権と独占禁止法－独禁法解釈論の再検討序説』(晃洋書房、2008 年) 10 頁。金沢良雄『独占禁止法の構造と運用』(有斐閣、1979 年) 192 頁)。
(注 51) 高田寛「ソフトウェアライセンスにおける著作権法と独占禁止法 21 条との関係について－販売代理店契約解除による地位移転契約を例に－」国士舘法学 41 号(2008 年 12 月) 37 頁。
(注 52) もともと、権利範囲論は、米国における特許権と反トラスト法の関係が問題となる領域では、反トラスト法の規制が及ばない固有の範囲が存在するという観念(特許の権利範囲論)に起因している(岩本章吾『知的財産権と独占禁止法－独禁法解釈論の再検討序説』(晃洋書

房、2008 年）（注 8）40 頁。稗貫俊文『知的財産権と独占禁止法』（有斐閣、1994 年）（はしがき）i 頁）。

(注 53) 公正取引委員会「知的財産の利用に関する独占禁止法の指針」（https://www.jftc.go.jp/dk/guideline/fukousei.html）（as of Dec 29, 2020）。

(注 54) プログラムの著作物については、当該プログラムの改変を禁止することは、一般的に著作権法上の権利の行使と認められる行為である。しかしながら、著作権法上も、ライセンシーが当該ソフトウェアを効果的に利用するために行う改変は認められており（著作権法 20 条 2 項 3 号、47 条の 2）、このような行為まで制限することは権利の行使とは認められない。

(注 55) 不正競争防止法 2 条 6 項。

(注 56) 経済産業省知的財産政策室『逐条解説　不正競争防止法（第 2 版）』（商事法務、2019 年）44 頁。

(注 57) 石本貴幸「リバースエンジニアリングによる営業秘密の非公知性判断と自社製品の営業秘密の考察」知財管理 68 巻 12 号（2018 年）1670 頁。経済産業省・前掲注(56) 44 頁。

(注 58) 通商産業省知的財産政策室『営業秘密逐条解説　改正不正競争防止法』（有斐閣、1990 年）151 頁。

(注 59) 知財高判 H30・7・30 裁判所 HP（https://www.courts.go.jp/app/files/hanrei_jp/870/087870_hanrei.pdf）（as of Dec 28, 2020）。

(注 60) 大阪地判 H30・6・7 裁判所 HP（https://www.courts.go.jp/app/files/hanrei_jp/813/087813_hanrei.pdf）（as of Dec 28, 2020）。

(注 61) 大阪地判 H29・8・24 裁判所 HP（https://www.courts.go.jp/app/files/hanrei_jp/169/087169_hanrei.pdf）（as of Dec 28, 2020）。

(注 62) 大阪地判 H28・7・21 裁判所 HP（https://www.courts.go.jp/app/files/hanrei_jp/090/086090_hanrei.pdf）（as of Dec 28, 2020）。

(注 63) 東京地判 H29・10・19 裁判所 HP（https://www.courts.go.jp/app/files/hanrei_jp/262/087262_hanrei.pdf）（as of Dec 28, 2020）。

(注 64) 東京地判 H29・2・9 裁判所 HP（https://www.courts.go.jp/app/files/hanrei_jp/520/086520_hanrei.pdf）（as of Dec 28, 2020）。

(注 65) 大阪地判 H15・2・27 裁判所 HP（https://www.courts.go.jp/app/files/hanrei_jp/281/011281_hanrei.pdf）（as of Dec 28, 2020）。

(注 66) 石本・前掲注(57) 1670 頁。

(注 67) 経済産業省・前掲注(56) 44 頁。石本・前掲注(57) 1671 頁。

(注 68) ユーザが、プログラムの使用許諾契約の中のリバースエンジニアリング禁止条項に違反したとして、プログラム開発業者がユーザを訴えても、ユーザは、抗弁としてリバースエンジニアリング禁止条項を独占禁止法上の不公正な取引方法（拘束条件付取引）であると主張することが考えられる。

(注 69) 経済産業省「情報セキュリティ早期警戒パートナーシップガイドライン」（2019 年）（https://www.ipa.go.jp/files/000073901.pdf）（as of Dec 28, 2020）。

(注 70) 情報処理推進機構（https://www.ipa.go.jp/index.html）（as of Dec 28, 2020）。

(注 71) 情報処理推進機構 HP（https://www.ipa.go.jp/security/outline/todoke-top-j.html）（as of Dec 28, 2020）。経済産業省・前掲注(69) 8～9 頁。