Hitotsubashi University

Global COE Hi-Stat Discussion Paper Series 112

Research Unit for Statistical and Empirical Analysis in Social Sciences (Hi-Stat)

Intellectual Property Right Protection in the Software Market

Yasuhiro Arai

February 2010

Hi-Stat Discussion Paper

Hi-Stat
Institute of Economic Research
Hitotsubashi University
2-1 Naka, Kunitatchi Tokyo, 186-8601 Japan
http://gcoe.ier.hit-u.ac.jp

Intellectual Property Right Protection in the Software Market*

Yasuhiro Arai†
Hitotsubashi University

January 2010

Abstract

We discuss the software patent should be granted or not. There exist two types of coping in the software market; reverse engineering and software duplication. Software patent can prevent both types of copies since a patent protects an idea. If the software is not protected by a patent, software producer cannot prevent reverse engineering. However, the producer can prevent the software duplication by a copyright. It is not clear the software patent is socially desirable when we consider these two types of coping. We obtain the following results. First, the number of copy users under the patent protection is greater than that under the copyright protection. Second, the government can increase social welfare by applying copyright protection when the new technology is sufficiently innovative.

Keywords: Copyright Protection; Intellectual Property Right; Software

JEL Classification: D42; K39; L86

*We thank Reiko Aoki, Dyuti Banerjee, Yongmin Chen, Taiji Furusawa, Hiroaki Ino, Kohei Kawamura, Keith Masukus, Akira Okada, Suzanne Scotchmer and seminar participants at Hitotsubashi University for helpful comments and discussions.

†yasuhiro68@gmail.com

1 Introduction

Patents have been used to reward inventors for their developments. In the USA, patent law grants right holders exclusive use only for inventions that are useful, new and nonobvious. Bessen and Hunt (2004) and Aharonian (2005) report that the United States Patent and Trademark Office (USPTO) grants more than 20,000 software patents a year. The number of software patents is growing rapidly in the USA. On the other hand, software patents are not granted by the European Patent Office (European Patent Convention Article 52). In July 2005, the EU rejected the patent proposal called the Computer Implemented Inventions Directive, and the European Patent Office announced that it would not grant software patents. USPTO gives weight to the software producer's incentive. The European Patent Office, by contrast, focuses on the welfare loss from exclusive use. It is not clear which policy is more socially desirable.

Many studies have investigated optimal patent protections. Gallini (1992) and Tandon (1982) discuss the optimal patent life by focusing on the basic tradeoff problem between the producer surplus and the social welfare loss. Klemperer (1990) and Gilbert and Shapiro (1990) consider the relationship between the patent length and the patent breadth. O'Donoghue, Scotchmer and Thisse (1998) and Scotchmer and Green (1995) discuss the optimal protection scheme with sequential innovation. However, it is difficult to apply such discussions to the software market because they do not consider specific properties of software. In the software market, there are two types of copying: reverse engineering and software duplication. Reverse engineering is copying by rival producers. For example, the rival producers can steal innovative technology by reading the source code of the software. They can improve the quality of their software by copying that technology. The literature on patents mainly focuses on the theft of ideas by producers.

When we consider the software market, we must also consider software duplication. With the emergence of computer technology, illegal copies of software are becoming increasingly easy to create and obtain. The Business Software Alliance (BSA, 2009) estimates that, in 2008, the illegal software market caused about US$53 billion in damages all over the world. This paper discusses both types of copying to ascertain the optimal form of intellectual property rights protection in the software market.

This analysis also focuses on the differences between patent and copyright since the software may be protected by both the patent and the copyright. From a legal viewpoint, there are many differences between these two forms of intellectual property rights. First, there are differences among those who make copies. For instance, most copyrighted products are copied by consumers. It is easy to copy a CD borrowed from a friend and enjoy it. However, it is difficult for a consumer to make a copy of a specific drug. Those who make copies of patented products require technology to capture the innovative idea of the product. Second, there are differences in the object protected by intellectual property rights. A patent protects an "idea," for example, how to make a specific medicine or innovative technology to improve computer throughput. On the other hand, a copyright protects "expression," such as films, books and video games.

Software can be protected by a patent if it contains an innovative "idea" to improve efficiency or productivity. Similarly, software also can be protected by copyright because it is an "expression" by a source code. We need to capture the differences to discuss the optimal form of intellectual property right protection in the software market. As shown above, a patent protects an "idea", and therefore a producer can prevent both reverse engineering by rival producers and software duplication by consumers. However, a copyright scheme cannot prevent reverse engineering as it does not protect an idea. It is not clear which is more beneficial from a social point of view.

Although there are many differences between copyright and patent from a legal viewpoint, they are treated in the same manner in economics. Economic analysis of intellectual property rights sheds light on the tradeoff between a producer's incentive and social welfare. All intellectual property rights grant the right holders exclusive use of their goods to protect producers' incentives. On the other hand, these exclusive rights decrease the social surplus owing to a monopoly situation. Many studies consider how to solve this trade-off problem. According to the economics literature, copyright is identical to patents because both forms of intellectual property rights share this tradeoff problem.

Over the past few years, a number of empirical studies have been conducted on software patents. For example, Lerner and Zhu (2007) and Mann and Sager (2007) reveal the impact of software patent on the software development empirically. However, few attempts have so far been made in theoretical research. Although some papers have considered software (Church and Gandal, 1992; Ellison and Fudenberg, 2000; Varian, 2000; Banerjee, 2003), they do not take into account differences between patent and copyright.

We obtain the following results. First, the number of copy users under the patent protection scheme is larger than that under the copyright protection scheme. Second, we compare two intellectual property right protection schemes for the software market: patent and copyright. When the degree of innovation is small, there is no difference between the two schemes because the rival producer does not steal the new technology. When the new technology is sufficiently innovative, governments can increase the quality of all software sufficiently by applying copyright protection. We show the effect of improving producer's quality and subsequent copying on protection. Recently, the necessity of software patents has been discussed. We indicate that the government should not protect software by patents. The government can increase social welfare by applying the appropriate copyright protection to provide sufficient incentive to producers.

This paper is organized as follows. Section 2 considers the optimal patent protection. In this section, theft of ideas by rival producers is prevented by a patent. We can obtain the socially optimal level of protection against software duplication and of social welfare in equilibrium. Section 3 discusses the optimal level of copyright protection against software duplication. In this section, we consider the case in which a rival producer can steal new technology and improve its software quality. The government attempts to set the optimal level of protection against software duplication in this situation. Section 4 then argues for the optimal intellectual property right protection scheme in the software market. We can compare social welfare in the equilibrium of the two schemes. This section reveals which protection scheme is better from viewpoint of society. Section 5 concludes the discussion. All proofs are provided in the Appendix.

2 Patent Protection in the Software Market

We discuss the optimal patent protection in the software market. In this case, the rival producer cannot copy new technology because of patent protection against reverse engineering. We consider two software producers in the market: producers 1 and 2. Both can produce software of the lowest level of quality $q_2 \geq 0$ [1] without innovation. Producer 1 can improve software quality to $q_1 = q_2 + \delta$ with new technology. δ denotes the degree of innovation. Producer 1 decides whether to produce the innovative software with development cost F. When producer 1 does not develop the new technology, producers set the price at zero and play a Bertrand competition in the software market. We also assume that there are two types of consumers: legal and illegal users. Legal users decide to purchase software from producer 1, producer 2 or neither. The consumer valuations of the software, each of which is denoted by v_i, are uniformly distributed on the interval $[0, 1]$. Each consumer wants to buy at most one unit. If consumer i purchases the software at its retail price p_j $(j = 1, 2)$, the utility is given by $q_j v_i - p_j$. Illegal users can make a perfect copy of the highest quality software at no cost and their utility is given by $q_j v_i$. The ratio of legal users is $0 \leq e \leq 1$, where e means the parameter of enforcement set by the government. We present a multistage game model

[1] We do not allow producer 2 to decrease its quality for simplicity. We can obtain the same qualitative conclusions even if we assume that the producer can decrease q_2.

to consider the optimal intellectual protection scheme. The four stages of the game have the following rules.

1. Government sets e to maximize social welfare.
2. Producer 1 decides whether to develop the new technology δ at development cost F.
3. Producers choose prices p_j simultaneously.
4. Legal users decide whether they will purchase the software from producer 1 or purchase nothing. Illegal users make copies of producer 1's software.

The government's goal is to maximize the social surplus, which is defined as the sum of the producers' surplus and the consumers' surplus. We analyze the subgame perfect equilibrium by backward induction. First, let us consider consumer behavior.

Lemma 1

Given e and price p_j, the optimal choice of legal consumers is to not obtain the good if and only if

$$v_i < \frac{p_1}{q_1}, v_i < \frac{p_2}{q_2}.$$

Legal users will purchase the software from producer 2 if and only if

$$v_i \geq \frac{p_2}{q_2}, v_i < \frac{p_1 - p_2}{q_1 - q_2},$$

and will purchase the software from producer 1 if and only if

$$v_i \geq \frac{p_1}{q_1}, v_i \geq \frac{p_1 - p_2}{q_1 - q_2}.$$

All illegal users will make a copy of producer 1's software.

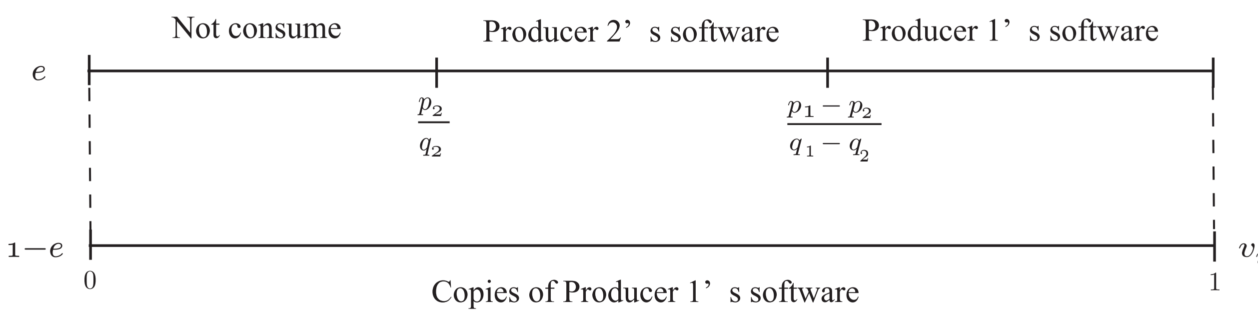

Figure 1: Consumer behavior when $p_1q_2 > p_2q_1$

Consumer behavior thus depends on their valuation of the software quality and price. In the first case, legal users ignore software when their valuation of the software is lower than the price of producer 2's software. In the second case, the utility of purchasing producer 2's software is positive and higher than the utility of purchasing producer 1's software. In the third case, consumers prefer producer 1's software to that of producer 2, because the utility of software 1 is positive and higher. Figure 1 shows consumer behavior when $p_1q_2 > p_2q_1$. In this class, consumers with valuations larger than $(p_1 - p_2)/(q_1 - q_2)$ purchase producer 1's software. Those with valuations between p_2/q_2 and $(p_1 - p_2)/(q_1 - q_2)$ buy the software from producer 2 and those with valuations less than p_2/q_2 do not consume. The legal users' demand for producer 1's software D_1 and for that of producer 2 D_2 when $p_1q_2 > p_2q_1$ are thus given by

$$D_1 = 1 - \frac{p_1 - p_2}{q_1 - q_2},\ D_2 = \frac{p_1 - p_2}{q_1 - q_2} - \frac{p_2}{q_2}. \tag{1}$$

From (1), we also obtain

$$\pi_1 = ep_1\left(1 - \frac{p_1 - p_2}{q_1 - q_2}\right) - F,$$

$$\pi_2 = ep_2\left(\frac{p_1 - p_2}{q_1 - q_2} - \frac{p_2}{q_2}\right).$$

Producers choose prices at the third stage. We consider their strategy in the next lemma.

Lemma 2

(1) If $0 \leq F < 4eq_1^2(q_1 - q_2)/(4q_1 - q_2)^2$, then prices of producers are given by

$$p_1^a = \frac{2q_1(q_1 - q_2)}{4q_1 - q_2}, \tag{2}$$

$$p_2^a = \frac{q_2(q_1 - q_2)}{4q_1 - q_2}. \tag{3}$$

The profits of producers are

$$\pi_1^a(q_1, q_2) = \frac{4eq_1^2(q_1 - q_2)}{(4q_1 - q_2)^2} - F, \tag{4}$$

$$\pi_2^a(q_1, q_2) = \frac{eq_1q_2(q_1 - q_2)}{(4q_1 - q_2)^2}. \tag{5}$$

(2) If $4eq_1^2(q_1 - q_2)/(4q_1 - q_2)^2 \leq F$, then producer 1 does not develop the new technology and as a result producers set $p_1^a = p_2^a = 0$.

We now consider the optimal level of patent protection against software duplication. The government chooses protection level e to maximize social welfare, which is defined as the sum of producer surplus and consumer surplus. If producer 1 develops new technology, the social welfare function is given by. The first term denotes the sum of producer surplus and consumer surplus from legal users. The second term represents the consumer surplus because of illegal users.

$$\begin{aligned} SW^a(e) &= e\left(\int_{\frac{p_1^a - p_2^a}{q_1 - q_2}}^{1} q_1 v dv + \int_{\frac{p_2^a}{q_2}}^{\frac{p_1^a - p_2^a}{q_1 - q_2}} q_2 v dv\right) + (1-e)\int_0^1 q_1 v dv - F \\ &= \frac{eq_1(12q_1^2 - q_1q_2 - 2q_2^2)}{2(4q_1 - q_2)^2} + \frac{q_1(1-e)}{2} - F. \end{aligned} \tag{6}$$

If producer 1 does not develop the new technology, social welfare is given by

$$SW^a(e) = e\int_0^1 q_2 v dv + (1-e)\int_0^1 q_2 v dv$$
$$= \frac{q_2}{2}.$$

The next lemma shows how changes in protection affect social welfare.

Lemma 3

(1) If $0 \leq F < 4q_1^2(q_1 - q_2)/(4q_1 - q_2)^2$, then

$$SW^a(e) = \frac{q_2}{2} \text{ for } 0 \leq e < \frac{F(4q_1 - q_2)^2}{4q_1^2(q_1 - q_2)},$$

$$\frac{\partial SW^a(e)}{\partial e} < 0 \text{ for } e \geq \frac{F(4q_1 - q_2)^2}{4q_1^2(q_1 - q_2)}.$$

(2) If $4q_1^2(q_1 - q_2)/(4q_1 - q_2)^2 \leq F$, then producer 1 does not develop the new technology and as a result $SW^a(e) = q_2/2$ for all e.

The implication of this lemma is clear. Social surplus is a decreasing function of the protection level e because the number of consumers who use the software decreases as protection increases. On the other hand, we can obtain the result that producer's profit is an increasing function of protection from equations (4) and (5), because the number of consumers who purchase the software increases as protection increases. If the government sets a low level of protection e, a producer may decide to not develop the new technology because it cannot compensate for its development cost. In such cases the social surplus will be $q_2/2$ under Bertrand competition. The next proposition shows the optimal level of patent protection e^a against software duplication.

Proposition 1

The optimal level of protection e^a against software duplication under the patent scheme is given by

$$e^a = \frac{F(4q_1 - q_2)^2}{4q_1^2(q_1 - q_2)} \text{ for } 0 \leq F < \frac{4q_1^2(q_1 - q_2)}{(4q_1 - q_2)^2},$$

$$e^a \in [0, 1] \text{ for } F \geq \frac{4q_1^2(q_1 - q_2)}{(4q_1 - q_2)^2}.$$

Lemma 3 shows that the government desires to set the protection level e as low as possible. Producer 1 may decide not to develop a new technology if the protection level is too low because its profit is an increasing function of e. Figure 2 shows this proposition. In the first case, setting the protection to zero will result in a negative profit for producer 1 with the development. The government sets e to provide sufficient incentive for development. The level of protection is set just high enough to result in a nonnegative profit after the invention. In this case, producers set the prices at (2) and (3). In the second case, producer 1 will never develop the new technology because the development cost is too high. If producer 1 does not develop it, consumers can use the software without cost because producers set the price at zero and play Bertrand competition. In this case, the social welfare does not depend on the level of protection against software duplication because all software is provided at zero price. Protection against software duplication increases as the development cost increases. In the next section, we consider the optimal level of protection against software duplication when the government applies a copyright protection scheme.

3 Copyright Protection in the Software Market

We consider how reverse engineering affects the protection level e and social welfare because copyright cannot prevent reverse engineering. When producer 1 develops the innovative technology δ, producer 2 can decide whether to steal it by reading the source code. We assume that the cost of reverse engineering is zero. The timing of the game is changed

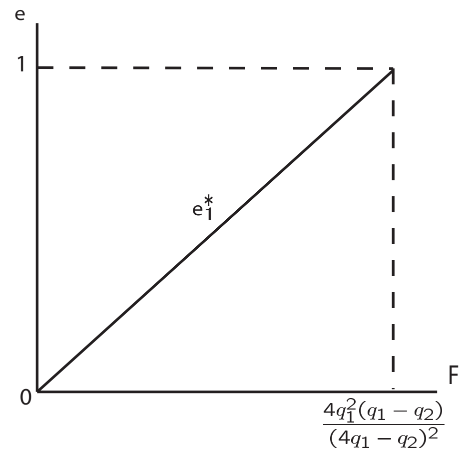

Figure 2: Patent protection against software duplication

as follows.

1. Government sets e to maximize social welfare.

2. Producer 1 decides whether to develop the new technology δ at a fixed cost $F > 0$. If producer 1 decides to develop it, producer 2 chooses a level of quality $q_2 + \gamma$ $0 \leq \gamma \leq \delta$ by reverse engineering.

3. Producers choose prices p_j simultaneously.

4. Legal consumers decide whether they will purchase software from producer 1, producer 2 or neither. Illegal users make copies of producer 1's software.

In this section, producer 2 can increase its software quality to maximize its profit by reverse engineering. We must consider how producer 2 applies the new technology. The next lemma shows how changes in producers' software quality affect their surplus.

Lemma 4

If producer 1 develops the new technology δ, producer 2 decides its strategy as follows.

(1) Producer 2 chooses $\gamma = (4\delta - 3q_2)/7$ when $\delta > 3q_2/4$.

(2) Producer 2 does not improve its quality when $\delta \leq 3q_2/4$.

The relationship between the quality and profit of producer 2 depends on the degree of δ. When δ is large, producer 2 may copy the new technology. Producer 2 can increase the software price and obtain higher profit by reverse engineering. On the other hand, when the degree of innovation is small, producer 2 has no incentive to steal the technology. In this case, the profit of producer 2 is decreased by reverse engineering because of severe price competition. In this section, we focus on the case where $\delta > 3q_2/4$. If δ is small, the outcomes are the same as discussed in Section 2. We can obtain the software producers' strategies in the next lemma.

Lemma 5

(1) If $0 \leq F < 7e(q_2+\delta)/48$, the prices of producers are given by

$$p_1^c = \frac{(q_2+\delta)}{4},$$

$$p_2^c = \frac{(q_2+\delta)}{14}.$$

The profits of producers are

$$\pi_1^c(q_1, q_2+\gamma) = \frac{7e(q_2+\delta)}{48} - F,$$

$$\pi_2^c(q_1, q_2+\gamma) = \frac{e(q_2+\delta)}{48}.$$

(2) If $7e(q_2+\delta)/48 \leq F$, then producers set $p_1^c = p_2^c = 0$.

The interpretation of this lemma is clear. In the first case, producer 1 develops the new technology and producer 2 copies it to maximize its own profit, because the degree of innovation is large and the development cost is low. When $7e(q_2+\delta)/48 \leq F < 4eq_1^2(q_1 -$

$q_2)/(4q_1 - q_2)^2$, the development cost F is so large that producer 1 cannot obtain sufficient incentive to develop the new technology because producer 2's copy decreases producer 1's profit. When $F > 4eq_1^2(q_1 - q_2)/(4q_1 - q_2)^2$, producer 1 does not develop the new technology because the development cost is too high. Consequently, producer 1 does not develop the new technology when $F \geq 7e(q_2 + \delta)/48$.

When the new technology is sufficiently innovative ($\delta > 3q_2/4$) and the development cost is small ($0 \leq F < 7e(q_2 + \delta)/48$), the social welfare function $SW^c(e)$ is as follows:

$$\begin{aligned} SW^c(e) &= e\left(\int_{\frac{p_1^c - p_2^c}{q_1 - q_2}}^{1} q_1 v dv + \int_{\frac{p_2^c}{q_2}}^{\frac{p_1^c - p_2^c}{q_1 - q_2}} \frac{4(q_2 + \delta)}{7} v dv\right) + (1 - e)\int_0^1 q_1 v dv - F \\ &= \frac{(q_2 + \delta)(12 - e)}{24} - F. \end{aligned} \tag{7}$$

The first term means the sum of the consumer surplus from legal buyers and the producer surplus. The second term means consumer surplus from illegal use. In these cases, the quality of both producers' products is increased by the new technology. The following lemma shows the impact of protection against software duplication on social welfare when there is reverse engineering in the market.

Lemma 6

(1) If $0 \leq F < 7(q_2 + \delta)/48$, *then*

$$SW^c(e) = \frac{q_2}{2} \text{ for } 0 \leq e < \frac{48F}{7(q_2 + \delta)},$$

$$\frac{\partial SW^c(e)}{\partial e} < 0 \text{ for } \frac{48F}{7(q_2 + \delta)} \leq e.$$

(2) If $7(q_2 + \delta)/48 \leq F$, *then* $SW^c(e) = q_2/2$ *for all* e.

This lemma can be interpreted in the same manner as Lemma 4. The following proposition discusses the optimal level of protection against software duplication e_2^*.

Proposition 2

When $\delta > 3q_2/4$, the optimal protection level e^c is given by

$$e^c = \frac{48F}{7(q_2+\delta)} \textit{ for } 0 \leq F < \frac{7(q_2+\delta)}{48},$$

$$e^c \in [0,1] \textit{ for } F \geq \frac{7(q_2+\delta)}{48}.$$

This result can be interpreted in the same manner as Proposition 1. Figure 3 shows this proposition. The government desires to set the protection as low as possible to maximize social surplus. However, it must set at a high enough level of protection to prevent producer 1's profit from being negative. In the first case, producer 2 applies the new technology to maximize its profit. The government takes into account producer 2's copying to set the protection level. In the second case, producer 1 will never develop the new technology because of the high development cost and producer 2's copying. When the degree of innovation is not large, producer 2 does not have an incentive to copy the new technology. In this case, the optimal protection level is the same as in Proposition 1.

4 Patent Protection vs. Copyright Protection

Thus far, we have considered two intellectual property right protection schemes, one that does not consider a producer's reverse engineering and another that does. The government can decide the level of protection against software duplication to maximize social welfare. However, the government also has the option of preventing reverse engineering by producer 2. As discussed above, software is protected by both patents and copyright. Patents can

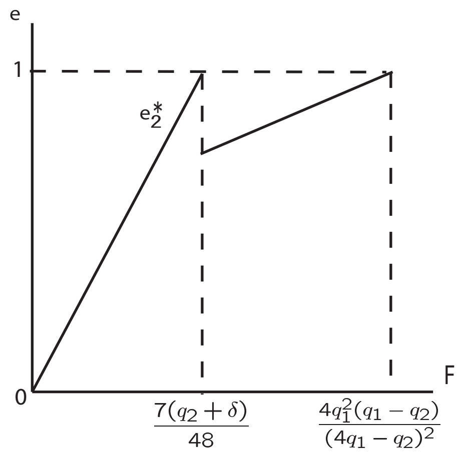

Figure 3: Copyright protection against software duplication

prevent reverse engineering; however, copyright cannot. In this section, we consider which protection scheme is better from the viewpoint of society: copyright or patent protection. The next proposition compares the levels of protection against the software duplication.

Proposition 3

When $F < 7(q_2+\delta)/48$, the number of copy users in the market under a copyright scheme is smaller than that under a patent scheme.

The intuition of this proposition is clear. When the degree of innovation is not large, the level of protection against software duplication is the same for a patent and copyright because producer 2 does not copy. When the degree of innovation is large and the development cost is small ($F < 7(q_2+\delta)/48$), the protection levels differ (Figure 4). If the government adopts a copyright protection scheme, producer 1's profit may decrease as a result of producer 2's reverse engineering. In this case, the government must increase protection against software duplication to provide sufficient incentive for producer 1. If the development cost is not small ($7(q_2+\delta)/48 \leq F$), the government cannot compensate producer 1 for its development cost under the copyright protection scheme. The next proposition shows how the software should be protected.

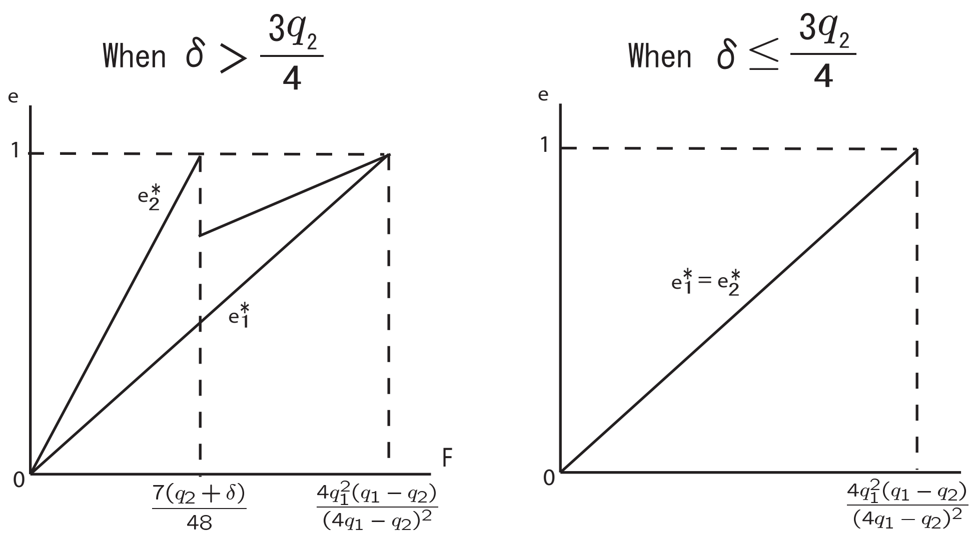

Figure 4: Protection against the software duplication

Proposition 4

From the viewpoint of society, the government should apply the copyright protection scheme in the software market when $F < 7(q_2 + \delta)/48$. *Otherwise, the government should apply a patent protection scheme.*

Figure 5 compares the social welfare under both schemes. When the degree of innovation is small, there are no differences between the two schemes, because producer 2 does not steal the new technology. When the new technology is sufficiently innovative and the development cost is low, the government can increase the quality of all software by relaxing protection against software duplication. In addition, if the government applies patent protection, producer 1's market power becomes very strong. This is not desirable from the viewpoint of social welfare. However, producer 1 does not obtain enough profit to develop the new technology under the copyright protection scheme when $F \geq 7(q_2 + \delta)/48$. In this case, the government should prevent copying by producer 2 with a patent protection scheme. This proposition also shows the relationship between the protection scheme and the idea. If we define the idea as a combination of the degree of innovation δ and the development cost F, we can show the optimal protection scheme for the idea (δ, F) as Figure 6. Figure 6 shows

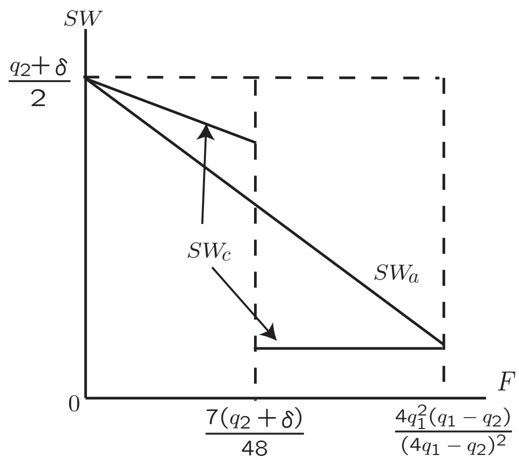

Figure 5: Patent vs. copyright protection

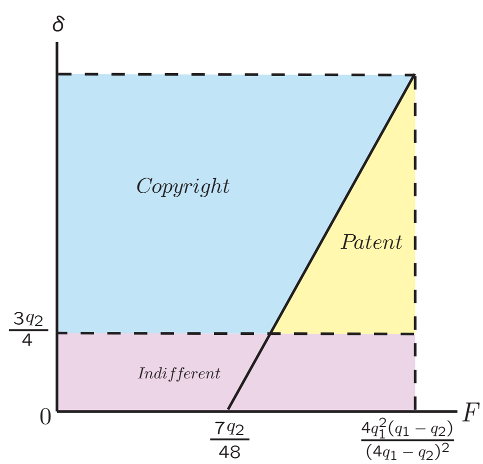

Figure 6: Optimal protection scheme

that the innovative idea, for example where δ is large and F is small, should be protected by a copyright scheme. Proposition 4 describes the effect of producer 1 improving quality and the subsequent copying on protection. In the USA, there are debates over protection of software by patents. In the software market, the quality of software is improved incrementally with each update. If the original technology q_2 is large and the degree of innovation δ is not, the range of development costs within which the patent protection scheme is socially desirable becomes small. Therefore, this proposition posits that the government should not use patents to protect software, but should provide more stringent protection against software duplication than for other copyrighted products.

5 Conclusion

We have considered whether software should be protected by patents. To discuss this problem, we need to take into account the differences between patent and copyright protection because software is protected by both forms of intellectual property rights. In this paper, patent protection and copyright protection are simply distinguished by changing the identity of the player who copies. We discussed intellectual property right protection in a model wherein (a) the government controls the level of protection against software duplication and against reverse engineering to maximize social welfare and (b) a software company can develop a new technology by incurring certain costs. We obtain the following results.

First, the level of protection against software duplication under the patent scheme is smaller than that under the copyright scheme when the development cost is not large. If there is reverse engineering in the market, protection against software duplication becomes strong. Consequently, the number of copy users under the copyright protection scheme is less than that under the patent protection scheme. Second, we show the optimal intellectual property right protection scheme in the software market. When the degree of innovation is small, there is no difference between the two schemes because the rival producer does not steal the technology. When the new technology is sufficiently innovative and the development cost is low, the government can increase social surplus by adopting copyright protection against software duplication. On the other hand, the government must apply a patent protection scheme when the development cost is high because producer 1 cannot obtain sufficient profit to develop software under a copyright protection scheme. We show the effect of improving the quality of software from a producer and subsequent copying on protection. Our analysis suggests that changes should be made with regard to the direction of modern copyright and patent policy. We indicate the importance of copyright protection in the software market.

We close this paper by pointing out some extensions of this model. First, we have assumed that the government can prevent copying by producer 2 perfectly under a patent protection scheme and that the government does not prevent reverse engineering at all under a copyright protection scheme. However, in reality the government can control the degree of reverse engineering by setting the breadth of the software patent. If we consider the case in which the government can set the upper limit of reverse engineering $\overline{\gamma}$, we can discuss the optimal intellectual property right protection scheme in the software market. We can obtain a similar result under this setting. When the development cost is low, copyright is more socially desirable than patent protection in the software market. The optimal level of protection against software duplication in this case is given by Figure 7. The intuition of this figure is clear. When the development cost is high, the government can decrease the degree of reverse engineering to permit development by producer 1. Consequently, the level of protection against software duplication becomes lower than in the first case.

Second, in this paper, the government can control the protection level e directly. However, the government controls this protection level through other policies such as a penalty for copyright infringement. It would be interesting to endogenize e. Third, producer 1 has the option of granting an exclusive license to producer 2. Patent protection may be better than copyright protection from viewpoint of society. In this case, the government does not have

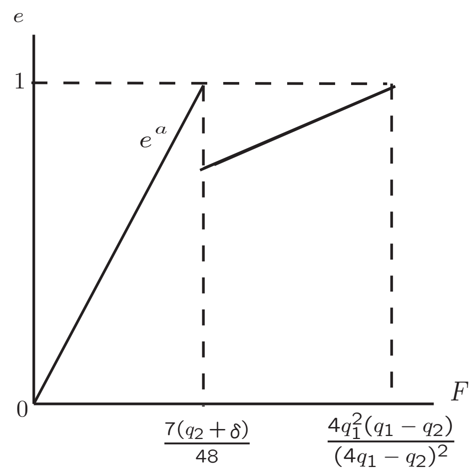

Figure 7: Optimal protection against software duplication

to set high levels of protection against software duplication because producer 1 can obtain a license fee from producer 2. Fourth, we assume that the cost of reverse engineering is zero in this model. In reality, it is difficult to acquire all information from reverse engineering. The rival producer must pay some costs to do that. Considering this cost, the range within which patent and copyright protection are indifferent from the viewpoint of society becomes large.

In addition, we have attempted to capture specific properties of software. However, this is not enough. As Scotchmer (1991) states, cumulative innovation is a very important topic with regard to software. For example, the quality of software improves sequentially as it is updated. Ellison and Fudenberg (2000) argue that monopolists have incentives to provide upgraded versions of the software. The arguments presented for patent protection for sequential innovation (Scotchmer and Green, 1990; O'Donoghue, 1998; O'Donoghue, Scotchmer, and Thisse, 1998; Bessen and Maskin, 2000) are helpful when considering this problem. These extensions will be the basis for future research. We also assume that the quality without the innovation is common to both producers. However, there are differences in technical capabilities among software producers. If we assume that one producer's software quality without the innovation is higher than that of the other producer's, we obtain a similar result to that in this paper.

References

[1] Aharonian, G. "Critiques of Software Patent Examination," *http://www.bustpatents.com/* (2005).

[2] Banerjee, D. "Software piracy: a strategic analysis and policy instruments," *International Journal of Industrial Organization,* Vol. 21 (2003), pp. 97–127.

[3] Bessen, J. and Hunt, R. "The Software Patent Experiment. Patents, Innovation and Economic Performance", ed. Organization for Economic Co-Operation and Development (2004).

[4] Bessen, J. and Maskin, E. "Sequential Innovation, Patents, and Imitation," *Working Paper* (2000), Department of Economics, MIT.

[5] Business Software Alliance. "BSA Global Piracy Study," *http://www.bsa.org/usa/research*. (2009).

[6] Church, J. and Gandal, N. "Network Effects, Software Provision, and Standardization," *Journal of Industrial Economics*, Vol. 40 (1992), pp. 85–103.

[7] Ellison, G. and Fudenberg, D. "The neo-Luddite's lament: excessive upgrades in the software industry," *RAND Journal of Economics*, Vol. 31 (2000), pp. 253–272.

[8] Gallini, N. "Patent Policy and Costly Imitation," *RAND Journal of Economics*, Vol. 23 (1992), pp. 52–63.

[9] Gilbert, R. and Shapiro, C. "Optimal Patent Length and Breadth," *RAND Journal of Economics*, Vol. 21 (1990), pp. 106–112.

[10] Klemperer, P. "How Broad Should the Scope of Patent Protection Be?" *RAND Journal of Economics*, Vol. 21 (1990), pp. 113–130.

[11] O'Donoghue, T. "A Patentability Requirement for Sequential Innovation," *RAND Journal of Economics*, Vol. 29 (1998), pp. 654–679.

[12] O'Donoghue, T., Scotchmer, S. and Thisse, J.-F. "Patent Breadth, Patent Life, and the Pace of Technological Progress," *Journal of Economics & Management Strategy*, Vol. 7 (1998), pp. 1–32.

[13] Scotchmer, S. "Standing on the Shoulders of Giants: Cumulative Research and the Patent Law," *Journal of Economic Perspectives*, Vol. 5 (1991), pp. 29–41.

[14] Scotchmer, S. and Green, J. "On the Division of Profit in Sequential Innovation," *RAND Journal of Economics*, Vol. 26 (1995), pp. 20–33.

[15] Tandon, P. "Optimal Patents with Compulsory Licensing," *The Journal of Political Economy*, Vol. 90 (1982), pp. 470–486.

[16] Varian, H. "Buying, sharing and renting information goods," *Journal of Industrial Economics*, Vol. 48 (2000), pp. 473–488.

6 Appendix

Proof of Lemma 1

In this lemma, we consider the optimal consumer behavior. In the first case, consumers choose to not consume the product because the utility of buying is negative. We obtain the equations

$$0 > q_1 v_i - p_1,\ 0 > q_2 v_i - p_2.$$

In the second case, when consumers use producer 2's software, they obtain a higher utility than when buying producer 1's software or when not using it at all. We therefore obtain the equations

$$q_2 v_i - p_2 > q_1 v_i - p_1,\; q_2 v_i - p_2 \geq 0.$$

In the third case, when consumers use producer 1's software, they obtain a higher utility than when consuming producer 2's software or using none at all. We therefore obtain the equations

$$q_1 v_i - p_1 \geq q_2 v_i - p_2,\;\; q_1 v_i - p \geq 0.$$

The lemma follows from these equations. Q.E.D.

Proof of Lemma 2

We define the producers' strategy as $S = \{(p_1, p_2) \mid p_1 \geq 0, p_2 \geq 0\}$, where $p_j (j = 1, 2)$ is the retail price of software. For convenience of analysis, we divide the strategy space S into two subclasses: $S_1 = \{(p_1, p_2) \mid p_1 q_2 \geq p_2 q_1\}$ and $S_2 = \{(p_1, p_2) \mid p_1 q_2 < p_2 q_1\}$.

When producers employ strategies in subclass S_1, the consumer behavior is illustrated in Figure 1. Consumers with valuations larger than $(p_1 - p_2)/(q_1 - q_2)$ purchase producer 1's software; those with valuations between p_2/q_2 and $(p_1 - p_2)/(q_1 - q_2)$ buy the software from producer 2 and those with valuations less than p_2/q_2 do not consume. The legal users' demand for producer 1's software D_1 and the demand for producer 2's software D_2 when $p_1 q_2 > p_2 q_1$ are thus given by

$$D_1 = 1 - \frac{p_1 - p_2}{q_1 - q_2},\;\; D_2 = \frac{p_1 - p_2}{q_1 - q_2} - \frac{p_2}{q_2}.$$

From these equations, we also obtain

$$\pi_1 = ep_1(1 - \frac{p_1 - p_2}{q_1 - q_2})$$

$$\pi_2 = ep_2(\frac{p_1 - p_2}{q_1 - q_2} - \frac{p_2}{q_2}).$$

Producers decide on a price to maximize their profit in subclass S_1 simultaneously. The equilibrium strategies in subclass S_1 are given by

$$p_1^* = \frac{2q_1(q_1 - q_2)}{4q_1 - q_2}$$

$$p_2^* = \frac{q_2(q_1 - q_2)}{4q_1 - q_2}.$$

The profits are thus given by

$$\pi_1^*(q_1, q_2) = \frac{4eq_1^2(q_1 - q_2)}{(4q_1 - q_2)^2}$$

$$\pi_2^*(q_1, q_2) = \frac{eq_1q_2(q_1 - q_2)}{(4q_1 - q_2)^2}.$$

When producers employ strategies in subclass S_2, producer 2's payoff becomes zero because there are no consumers who purchase producer 2's software in the market. In this class, producer 2 has an incentive to decrease its prices to a level that satisfies $p_1q_2 \geq p_2q_1$. Consequently, the strategies in subclass S_1 become the equilibrium outcomes over the whole strategy space S. Q.E.D.

Proof of Lemma 3

The social welfare when the new technology is developed by producer 1 is given by

$$SW^a(e) = \frac{eq_1(12q_1^2 - q_1q_2 - 2q_2^2)}{2(4q_1 - q_2)^2} + \frac{q_1(1-e)}{2} - F.$$

We thus obtain

$$\frac{\partial SW^a}{\partial e} = -\frac{q_1(q_1 - q_2)(4q_1 - 3q_2)}{2(4q_1 - q_2)^2} < 0.$$

The social welfare is a decreasing function of the protection level e when the new technology is developed. The technology will not be developed, however, if the profit is negative. The producer's profit depends on the degree of δ and F, and is an increasing function of e.

In the case of $0 \leq F < 4q_1^2(q_1 - q_2)/(4q_1 - q_2)^2$, if the protection is so low that the producer's profit is negative, producer 1 will not develop the new technology and play Bertrand competition. In the last case, the development cost is larger than the maximum profit of the producer. In this case, the new technology will not be developed for any e. Q.E.D.

Proof of Proposition 1

From Lemma 3, the social welfare is a decreasing function of the protection level e if the new technology is developed. In the first case, the protection should be chosen at the minimum level that provides an incentive for the producer to develop, because the new technology is socially desirable in this range of development cost. In the second case, the producer cannot develop the technology for any e. The government's optimal penalty is therefore unconstrained. Q.E.D.

Proof of Lemma 4

Producer 2's profit when producer 1 develops a new technology is given by

$$\pi_2^*(q_1, q_2) = \frac{eq_1q_2(q_1 - q_2)}{(4q_1 - q_2)^2}.$$

We can obtain

$$\frac{\partial \pi_2}{\partial q_2} = \frac{e q_1^2 (4q_1 - 7q_2)}{(4q_1 - q_2)^3}.$$

Producer 2's profit is maximized when $4q_1 = 7q_2$. Producer 2 can increase its profit by reverse engineering when δ is larger than $3q_2/4$. When δ is larger than the limit, producer 2 chooses γ to be $4(q_2 + \delta) = 7(q_2 + \gamma)$. Therefore, producer 2 chooses $\gamma = (4\delta - 3q_2)/7$ to maximize its profit. Q.E.D.

Proof of Lemma 5

From Lemma 4, producer 2 improves its quality to $4q_1/7$ when producer 1 develops the technology. We can obtain the following result by substituting $q_2 = 4q_1/7$ into the results of Lemma 2.

$$p_1^c = \frac{(q_2 + \delta)}{4},$$

$$p_2^c = \frac{(q_2 + \delta)}{14}.$$

Under these strategies, the profits of producers are

$$\pi_1^c \left(q_1, \frac{4(q_2 + \delta)}{7} \right) = \frac{7e(q_2 + \delta)}{48} - F,$$

$$\pi_2^c \left(q_1, \frac{4(q_2 + \delta)}{7} \right) = \frac{e(q_2 + \delta)}{48}.$$

In the second case, producer 1 will not develop the technology because the development cost is too high. Q.E.D.

Proof of Lemma 6

When the new technology is sufficiently innovative ($\delta > 3q_2/4$) and the development cost is smaller than producer 1's profit ($0 \leq F < 7e(q_2 + \delta)/48$), social welfare when producer 2 steals the new technology that is developed by producer 1 is given by

$$SW^c(e) = \frac{(q_2+\delta)(12-e)}{24} - F.$$

We obtain

$$\frac{\partial SW^c}{\partial e} = -\frac{(q_2+\delta)}{24} < 0.$$

The social welfare is thus a decreasing function of e when the technology is developed by producer 1. Producer 1 will not develop the new technology if its profit is negative. The profit depends on the magnitude of δ and F, and is an increasing function of e. In the first case, producer 1 does not develop the new technology because e is too small to compensate for the development cost when producer 2 copies it. When the development cost is large ($F \geq 7(q_2+\delta)/48$), producer 1 does not develop the new technology because its profit becomes negative even if $e = 1$. Therefore, producers play Bertrand competition. Q.E.D.

Proof of Proposition 2

From Lemma 6, the social surplus is a decreasing function of e when the technology is developed. In the first and second cases, protection is chosen at the minimum level that provides an incentive for a producer to work. In the last case, the producer cannot afford to develop the technology for any e. The optimal government protection is therefore unconstrained. Q.E.D.

Proof of Proposition 3

We compare the optimal protection level e^a and e^c.

When $0 \leq F < 7(q_2+\delta)/48$

$$e^a - e^c = \frac{F(4q_1-q_2)^2}{4q_1^2(\delta-1)(q_1-q_2)} - \frac{48F}{7(q_2+\delta)} = -\frac{F(21q_2+20\delta)(4\delta-3q_2)}{28(q_2+\delta)^2\delta} \tag{8}$$

This equation becomes negative when $\delta > 3q_2/4$.

From equation (8), the level of protection against software duplication under copyright protection is more severe than that under patent protection when the development cost is low. Q.E.D.

Proof of Proposition 4

We compare social welfare under each penalty scheme. We obtain the social surplus when reverse engineering does not exist in the market by substituting e^a into equation (6):

$$SW^a = \frac{4(q_2+\delta)^2 - F(9q_2+12\delta)}{8(q_2+\delta)} \text{ for } 0 \le F < \frac{4q_1^2(q_1-q_2)}{(4q_1-q_2)^2}, \tag{9}$$

$$SW^a = \frac{q_2}{2} \text{ for } F \ge \frac{4q_1^2(q_1-q_2)}{(4q_1-q_2)^2}.$$

We then obtain the social welfare when there is reverse engineering by substituting e^c into equation (7):

$$SW^c = \frac{7(q_2+\delta) - 18F}{14} \text{ for } 0 \le F < \frac{7(q_2+\delta)}{48}, \tag{10}$$

$$SW^c = \frac{q_2}{2} \text{ for } F \ge \frac{7(q_2+\delta)}{48}. \tag{11}$$

We compare equations (9) and (10).

$$\frac{4(q_2+\delta)^2 - F(9q_2+12\delta)}{8(q_2+\delta)} - \frac{7(q_2+\delta) - 18F}{14} = -\frac{3F(4\delta - 3q_2)}{36(q_2+\delta)}. \tag{12}$$

This equation becomes negative when $\delta > 3q_2/4$.

We compare equations (9) and (11):

$$\frac{4(q_2+\delta)^2 - F(9q_2+12\delta)}{8(q_2+\delta)} - \frac{q_2}{2} > 0. \tag{13}$$

Therefore, social welfare under copyright protection when F is large is smaller than that under patent protection. Q.E.D.