Copyleft - the economics of Linux and other open source software[♠]

Mikko Mustonen[*]

A large and increasing population of programmers is engaged in the development of freely distributed software. Its members receive little direct compensation for their work but instead a novel licensing scheme – copyleft – creates an incentive structure reminiscent of science based on complementary income. We build a model where the occupational choice of programmers determines the quality of programs in the consumer market. A monopolist, supplying the consumer market, has to take into account the impact the free software has on the market. When software implementation costs are low the monopolist will accept the presence of the copyleft program in the market. Our model explains the simultaneous presence of commercial and free copylefted programs in the market and also why there may not exist commercial alternatives to copyleft programs. The analysis provides policy implications showing for example that consumers' improved information of a substitute copyleft program increases welfare.

JEL Codes: D23, L11, L15
Keywords: Property rights, copyleft, Linux, open source software

(Published as Discussion Paper No. 493, Department of Economics, University of Helsinki)

[♠] I thank Vesa Kanniainen for advice and encouragement. The usual disclaimer applies.
[*] University of Helsinki, Department of Economics, P.O. BOX 54, FIN-00014, Finland. E-mail: mikko.mustonen@helsinki.fi

1 Introduction

Copyleft is a novel licensing scheme to facilitate open and decentralized software development. Its key feature is that once a program is licensed by the author, the subsequent programs based on the original must also be licensed in a similar manner. For programmers, copyleft creates an incentive structure alternative to employment allowing them to signal their abilities and receive complementary income. As investments in computer software are becoming quite large, for example in Sweden representing 11% of total national investments in 1998 (Jagren and Morell, 2000), the efficiency and incentives of software development work have become a new issue. In this paper we develop a positive economic model of software development in the presence of copyleft and analyze how it affects commercial software markets.

Copyleft licensing has profound implications in IT-industry. First, it creates an alternative incentive structure for programmers in program development. Second, the presence of a copyleft program that consumers can acquire for free interacts with the program product market. We ask how copyleft affects the behavior of a monopolist who invests in the quality of his program in the development stage. A monopolist can influence the occupational choice of programmers through his wage policy. The programmers make an occupational choice as to whether to be employed within the firm or whether to join the copyleft community to develop jointly a copylefted program and receive complementary income based on acquired reputation. These choices in turn affect the qualities of the firm's copyright program and the copylefted program. In the output stage, the monopolist is supplying a program protected by copyright. It has a market of its own and there are no substitutes except the program developed eventually by the copyleft community. Consumers value quality and determine whether to buy the copyright program or acquire the copyleft program or not to use any of these. This creates a trade-off for the monopolist: paying a higher wage in the development stage increases costs but increases also revenue by increasing the quality of the copyright program and decreasing the quality of the copylefted program. We analyze the monopolist's profit-maximizing behavior in relation to the consumers' implementation costs. These are the costs consumers face when installing and learning the programs.[1]

The contribution of the paper is the modeling of copyleft programming activity interacting both with the labor market of programmers and the product market for programs. We characterize the profit-maximizing behavior of the monopolist and the conditions for entering the industry. We also analyze the welfare consequences of society's actions resulting in policy implications of copyleft. To bring out the novel effects copyleft licensing has on labour and product markets in this paper we abstract from network effects for programs in consumer markets. Acknowledging their importance, we plan to address the issue in future.

The main results of the analysis are the following. The level of the implementation costs of programs determines whether the software firm, the monopolist, has to take into account the copylefted program in the market. If the costs are low, then there are

[1] The model is a general one and applicable to many markets but the reader may feel more familiar with it by imagining the monopolist as William Gates and his product as Windows. The copyleft community could be the programmers working on the project initiated by Linus Torvalds and the copyleft product could be Linux.

consumers preferring the copylefted program and this creates a constraint to profit maximization. There is an intermediate interval for implementation costs, where the optimal price for the monopolist just deters the marginal consumer from acquiring the copyleft product. When the implementation costs are high, the monopolist can apply the optimal monopoly price. The larger is the consumer market compared to the population of programmers the more the monopolist employs programmers and the smaller is the copyleft community. The monopolist employs a large number of programmers if programmers' complementary income from copyleft work is low or if consumers' valuation of the quality of the programs is high. In the presence of copyleft programmers, the consumer market has to be large compared to the total programming population. There is a threshold size of the market below which the monopolist cannot profitably develop a program. Casual empirism tells that the results coincide with real phenomena in software markets. The implementation costs of programs have no doubt decreased resulting in increased use of free copyleft programs like Linux. On the other hand, there are application areas, for example in networking, where the supply is almost exclusively copyleft programs.

Copyleft licensing has also policy implications. The incentive structure of copyleft programming is independent from the consumer market. Programmers do not care whether there are users of the copyleft program outside the copyleft community. In this environment it is possible that there exists a copyleft substitute program that consumers are not aware of. Informing consumers of such a program is likely to increase welfare with presumably low costs. An another policy implication concerns the enforcement of copyright. Securing the copyright can be costly and the copyleft community may not have the resources to defend the copyright vital to the incentives. If society doesn't support a high level of copyright protection the copyleft communities are likely to restrict the distribution of programs to consumers because the risk of copyright violations increases. This in turn decreases welfare.

1.1 Properties of copyleft

Copylefting a program means that the programmer, besides copyrighting the program also signs a General Public License (GPL) (GNU Project 2000b) granting everyone the right to use, modify and distribute the program *with the condition that the licensee also grants similar rights to the modifications he has made*. With this arrangement, the program is simultaneously freely usable but protected from becoming someone's private intellectual property[2]. First of all, this allows decentralized program development because the enhancements and modifications cumulate to the common program even if the programmers have no other affiliation with the project. For this analysis however, we are interested on the other significant feature of copyleft: it is a device of linking the programmer and his contribution permanently together while the contribution is publicly observable. This creates an environment where talented programmers have an incentive to signal their abilities via the copyleft community.

[2] Lerner and Tirole (2000) and Johnson (2000) provide detailed descriptions of the licensing schemes (GPL, Open Source Software, Debian Social Contract) that create the copyleft environment. Open Source Initiative (2000) contains the definition of open source software and GNU Project (2000a) a classification of free and non-free programs. Browne (2000) provides a practitioner's view on copyleft licensing.

Historically, copyleft licensing was created originally for ideological purposes by Richard Stallman and the Free Software Foundation (GNU Project 2000b). However, the functions and quality of some programs like Linux and Apache have reached and in some respects surpassed those of traditional copyright protected commercial programs and the population of programmers participating in the development work is so large that ideological motivations are inadequate to explain the phenomenon.

In a short time, copylefted programs have gained significant market shares. For example, the web server program Apache is used by 55% of web sites and the Linux operating system for personal computers and larger systems is installed in over two million systems (Lerner and Tirole 2000). The phenomenon of having both commercial copyright and non-commercial copyleft programs present in the marketplace is a real one. However, due to the low cost of copyleft programs the market shares for them in public market research reports are low even if the shares of users may be high. Deckmyn (2000c) reports that even though the number of users of Linux increases twice the speed of Windows, the turnover from Linux business is projected to be only a few percentages of that of Windows in five years.

Even though signing the license agreement means that the creator, the programmer, cannot receive any rents from the sale of his creation there still can be business activity based on copyleft programs. Varner (2000) categorizes the copyleft business actors into four groups: service sellers, loss leaders, widget frosting and accessorizing. A service seller provides installation and operation services to copyleft products. Maybe the best known company that packages and supports Linux at the moment is Red Hat. Loss leaders distribute a copyleft program to create demand for some other copyrighted product. Netscape web browsers are an example of that. Widget frosting refers to hardware suppliers that may enhance their product with some copyleft program. Accessorizing is essentially selling complementary, but remote services or products. Recently, large computer companies (IBM, HP, SUN) have announced their support to some existing copyleft programs like Linux. SUN Microsystems has also developed programs (Staroffice) that are substitutes to existing office applications and copylefted them (see Deckmyn 2000b). Lerner and Tirole (2000) discuss the troubling phenomenon of firms engaging in copyleft work.

1.2 The incentives to innovate

The economics of information tell us that there is a fundamental trade-off between the public good nature of information and the incentives to create new information. (Arrow 1962). Information, like a computer program, once it is created, is practically costless to be reproduced. From the society's point of view, it should then be distributed freely. However, without incentives there will then be no creation of new information. The society's solution to the missing incentives has been to secure various intellectual property rights to originators. In the case of a new, novel and non-obvious invention, a patent gives to the inventor a temporary monopoly to the invention. Copyright protects for a period the rights of the originator of a unique expression, for example a work of art[3].

[3] For a survey of the development of intellectual property rights, see David (1993)

From the intellectual property rights perspective, a computer program is a problematic object (Samuelson 1993, Dam 1995, Kemppinen 2000). It can be a unique expression like a poem, and copyright to it would seem an appropriate method of protect the rights of the originator. But a program, once it is running, also creates functions, quite like a machine. We could think that new and novel functions or uses for a function would fulfill the requirements for a patent. A disturbing detail is that algorithms and 'mathematics' are not eligible for patenting and programming in a sense is exactly creating and modifying algorithms. As a result of the first cases of law over ownership of property rights for software, copyright is generally applied as the method of property rights. Presumably the original motivation of the Free Software Foundation to introduce the copyleft licensing scheme was based on this development. Copyright to programs was considered to restrict innovation in programming due to the uniqueness requirement. Recently, however, in the US also patents have been granted to programs.

The rapid development of some originally copylefted programs (Linux, Apache, Mozilla, Sendmail, PERL) and the large number of participating programmers suggest that there must be powerful incentives present to create and further develop copyleft programs. The licensing scheme rules out direct appropriation of rents based on property rights. However, we see simultaneously traditional software development, where programmers' incentives are secured by property rights and copyleft software development where programmers give away their rights on the outset. Dasgupta and David (1987, 1994) present a framework of 'Science' and 'Technology' that applies well to this situation. They suggest that new knowledge is created in society under two distinctly different incentive structures[4].

In the 'Science' environment, peer recognition and the resulting reputation lead to complementary benefits, such as grants, positions in academic organizations or highly compensated future positions in firms. The combination of these is the incentive. Scientific recognition is achieved by making one's contribution public to peer review as quickly as possible and acquiring ***priority*** to the new knowledge. In the 'Technology' environment, the incentive structure is the traditional one in economics: maximization of profit by securing property rights. By definition, these rights keep the new knowledge private. Dasgupta and David, having studied several fields of research, conclude that these incentive structures in many cases appear simultaneously and that they both are present in the same research areas. The value of this analysis to the study of copyleft is the striking analogy of 'Science' environment to copyleft community and in turn 'Technology' to traditional software development. *We could say that the essential property of the copyleft licensing scheme is that it creates a particular incentive structure within the business environment. This structure has properties that are equivalent to the incentive structures of scientific communities.*

The framework of Dasgupta and David contains also the crucial element of the positive economic model of copyleft. They assert that the occupational choices of aspiring employees are the decisive factor in the relative shares of both incentive environments. Employees assess the benefits of 'Science' and 'Technology'

[4] Brooks (1994), Stephan (1996) and Stephan and Levin (1996) discuss similar issues. Stern (1999) provides empirical research that indicates that there exist significant incentives such as suggested by Dasgupta and David.

environment. For the same performance in ‘Science’ environment, the level of expected direct monetary pecuniary elements usually is lower than in ‘Technology’ and due to the complementary nature of income they may be linked to less interesting activities, for example teaching or project management. But on the other hand, the elasticity of the compensation to the performance may be much higher in the 'Science’ environment. An able scientist is almost certainly rewarded for his contributions because they, according to the priority principle become public and receive the appreciation of peers. In ‘Technology’ environment it is probable that the incentives in the firms do not take fully or at all into account the contributions of individual employees. Furthermore, the secrecy inherent in private research inhibits also the accumulation of reputation to individual employees.

The phenomenon of copyleft or open source software has been noticed in the literature. Lerner and Tirole (2000) provide an extensive survey of case studies of projects where the development mode is decentralized and the licensing of the programs is in copyleft fashion. They analyze the motivation and incentives of the programmers engaging in copyleft activity and conclude that the skill signaling is an essential factor in the incentives for copyleft work. Lerner and Tirole concentrate in their paper to the effectiveness, longevity and structure of decentralized open source software projects in comparison to the traditional hierarchical commercial projects. Johnson (2000) also focuses on the effects copyleft licensing has to program creation. He models copyleft activity as private provision of a public good: programmers receive utility from new code developed by them all. A single programmer can either contribute to the project or ‘free-ride’, receive utility from the work of others. The decisions of the programmers create a game, which according to Johnson has Bayesian Nash equilibria. Johnson shows that free-riding becomes more common and ultimately a constraining factor in program development as the size of the project increases. He also analyzes and compares the welfare implications of both the copyleft model and the traditional software development and finds that neither coincides with the welfare optimum. Johnson also acknowledges the signaling incentives for copyleft work but regards the public good nature of the open source software from copyleft activity to be dominant. Dalle and Juillien (1999) view copyleft licensing as an ‘anti-patent’ system enhancing creativity in society. They acknowledge the skill signalling incentives of copyleft programming but consider the expected profits from future programs created under traditional copyright protection to be an important motive for programmers. They also model the diffusion of a copyleft and a copyright to agents in an evolutionary adoption model. However, none of these authors consider the implications the copyleft programs have on the markets of copyright products. Their interest is focused on the implications of copyleft to labor markets and software creation.

Copyright is an important method of securing intellectual property rights and it has received considerable attention in the literature. Landes and Posner (1989) provide a seminal model of copyright protection. In their model, the level of copyright protection is endogenous. It is either a decision variable optimized by the social planner or is determined by decentralized markets. If the level of protection is high the authors face little competition from unauthorized copies in the market and profits are high. The trade-off to that is that a high level of copyright protection also inhibits authors of new works from utilizing present works and the costs of creation of new works are higher. Koboldt (1995) develops these ideas further by modeling a market where the original work is sold at a price and a substitute copy is sold at the cost of copying. These costs include the production cost and costs of being caught, violating the copyright. Our analysis of the market for a copyleft program draws on this model. Takeyama (1994,1997) and Shy and Thisse (1999) analyze the incentives of protecting the copyright of programs by costly technical methods. In these models there are both copyright programs and unauthorized cheap copies in the market. The authors all find, using different models and assumptions that when programs exhibit network externalities, the unauthorized copying may increase the profits of the authors.

The structure of the paper is the following: in chapter 2 we first develop the cost function of the monopolist in the development stage based on the novel incentive structure created by copyleft. Then we analyze the market for programs with the substitute copyleft program present. In chapter 3 we solve the profit function of the monopolist in the presence of copyleft and characterize the optimal behavior of the monopolist. Chapter 4 presents some policy implications and discussion in chapter 5 concludes.

2 The Model

The setup of our model is a two stage one. In the development stage a monopolist and a copyleft community engage in programming projects. In the output stage the consumers value programs and the monopolist sets a profit maximizing price to his program. There is also a copyleft program, developed by the copyleft community available for free. Consumers and the monopolist take the qualities of the programs as given. Programs are produced and consumers choose the monopolist's program, the copyleft program or neither of them. We solve the model using backward induction: in development stage the monopolist anticipates the behavior of the program market in the output stage and maximizes profit by hiring the optimal number of programmers. The timeline of the model is in figure 1.

The monopolist invests in the development stage in the quality of his product, a program. The quality is dependent on the programming output and the monopolist can determine the output by hiring programmers. The programmers' heterogeneous ability is unobservable in employment. Copyleft licensing creates an alternative for programmers, i.e. by engaging in copyleft work and receiving complementary income based on ability. To signal their skills the programmers in the copyleft community develop in a decentralized manner a program that is a substitute to the monopolist's program. In the labor market the monopolist faces convex total costs for output as a result of the presence of the copyleft community.

In the output stage, consumers choose between buying the monopolist's program, acquiring for free the copyleft program developed by the copyleft community or not using any program. We assume that consumers face implementation costs for the programs. These costs cover, for example, the effort of acquiring the physical media, the effort of installing the program in the computer, conversion and rearrangement of data and of course the learning costs of the program. In the model, we characterize the monopolist's behavior in relation to the level of the implementation costs. Both technology and the skills of users influence the implementation costs. On one hand new programs have more complex properties, on the other hand a large part of the programming effort is directed to enhance ‘ease-of-use’[5].

Copyright protection ensures that in output stage the monopolist can price his product without the threat of illegal copies. Copyright protection is essential also for the copyleft community. The copyleft license grants a copyright to the original programmer and the robustness of this property right is crucial to the incentives of programmers even though the copyleft license allows the free use and further development of the program. We assume that copyright protection in our model is perfect. With the risk of losing generality we have tried to choose simple functional forms and attribute distributions to bring out the effect of copyleft as clearly as possible.

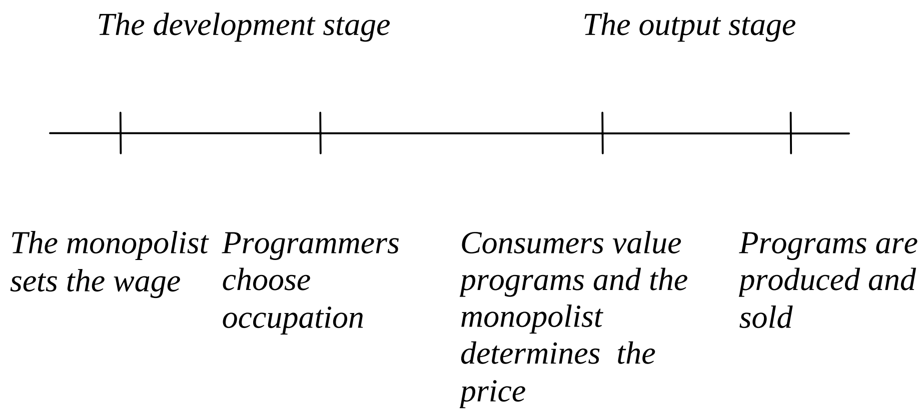


Figure 1: Order of events in the model

2.1 Programmers’ occupational choice in the development stage

We assume that a programmer has five career alternatives to choose between:

a) Employment in a firm producing software
b) Self-employment by creating a program and selling the rights to it
c) Entrepreneurship by creating a program and establishing a firm in a market
d) Entrepreneurship by creating a program and contracting with a venture capitalist with complementary skills to jointly establish a firm to sell and support the product
e) Copyleft work by creating a program and licensing it as copylefted with the GPL

We can abstract from some of these alternatives focusing into two. In well-developed labor and software markets the cases a) and b) are close to each other. The combination of programming and entrepreneurial skills is likely to be rare, so we assume that the number of programmers choosing c) or d) is small and constant. That leaves us with two career alternatives: either to be employed by a software firm or to join the copyleft community by developing programs that are copylefted.

[5] A general trend in society is an increase of computer skills or ‘literacy’, due to both public education and user experience. This may decrease the level of the implementation costs.

Based on the analysis of Dasgupta and David (1987,1994), we assume that these two careers have different incentive structures. Employment in the firm is compensated by an equal wage for all programmers.[6] The compensation of creating copyleft programs has to be of complementary nature. The programmer will not receive any rents from his creation after it has been copylefted. Instead, we assume that the public nature of the copyleft program encourages peer review and able programmers can build a reputation that results in expected complementary income, such as future partnership in software ventures, grants or academic employment. Lerner and Tirole (2000) aggregate these career concern[7] and ego gratification incentives "under a single heading: signaling incentives". According to them these incentives are the stronger the more visible the performance, the more it depends on effort and the more it tells about talent. The empirical analysis of Stern (1999) points out that the most able of the population tend to attach to the 'Science' community which corresponds to the copyleft programmers in our model.

To model the occupational choice of a programmer we make simplifying assumptions. There are N programmers with industry-specific skills, N_R employed by the firm in software production, N_L forming the copyleft community, $N = N_R + N_L$. We assume that programmers' productivity a_i is evenly distributed on the unit interval $[0,1]$. Let P be the expected complementary income for programmer contribution. Then the expected total complementary income for programmer i is $a_i P$. The firm pays a wage w to all employed programmers regardless of their productivity. Programmer i is indifferent between employment and copylefting if $Pa_i = w$. Given w and P, the level of productivity of the marginal programmer indifferent between employment and copylefting is

$$a^* = \frac{w}{P} \ . \tag{1}$$

Programmers with productivity greater than the threshold value a^* join the copyleft community and those whose productivity is below a^* are employed by the firm.[8] The number of copyleft programmers N_L and firm programmers N_R depends on the threshold productivity a^*,

$$N_L = \left(1 - a^*\right)N \tag{2a}$$

$$N_R = a^* N \ . \tag{2b}$$

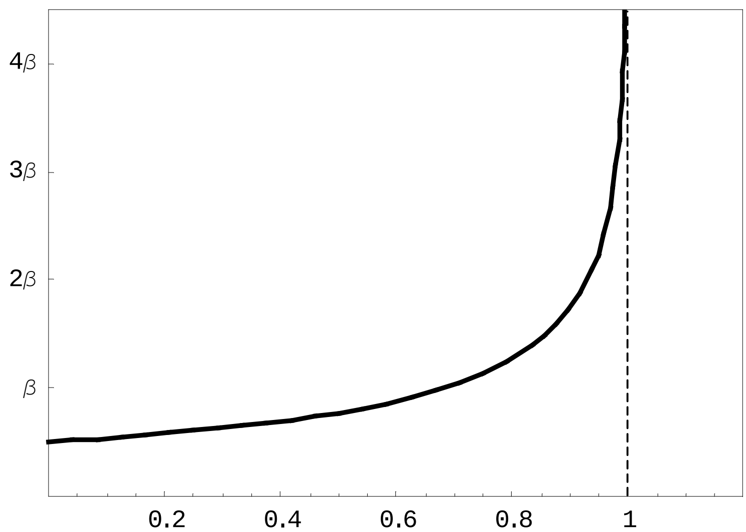
[6] This can be supported by casual empirism of software development environments, which indicates that the process of creating a competitive program in a centralized development environment requires sophisticated teamwork and project management. While the productivity of programmers varies greatly, in this environment it is difficult to assess individual contributions.

[7] Rigorous analysis of career concerns can be found in Holmström (1999) and Detriwapont et al. (1999)

[8] Obviously there is an outside employment option with a wage w_0 but we assume that it is below the wage levels present in this model.

We assume that the complementary income for contribution P has an inverse relation to the size of the copyleft community N_L. The members of the community value new contributions the more the smaller the community is. This assumption is in line with the analysis of Lerner and Tirole (2000) and Johnson (2000) on the inner dynamics of copyleft communities. Based on case studies, Lerner and Tirole assert that a copyleft community is prone to 'break' as the size increases. Johnson analyses the production process within the copyleft community and finds that as the size increases, free riding becomes a restricting issue. To express the relation in a simple way we define the complementary income to be a decreasing function of the proportion of programmers belonging to the copyleft community. One important feature of this specification is that as the number of copyleft programmers approaches zero, the complementary income goes to infinity. Several case studies of copyleft programming (Lerner and Tirole 2000) indicate that some individuals stay in the copyleft community even if alternative monetary benefits are large. The chosen functional form parallels with this notion. This assumption also captures the phenomenon of programmers that have in part ideological reasons to engage and stay in copyleft programming. Let P[9] be a function of the threshold productivity a^* with the parameter $\beta > 0$ describing the level of complementary income (see figure2)

$$P\left(a^*\right) = \beta\left(\frac{-\ln\left(1-a^*\right)-a^*}{a^{*2}}\right). \qquad (3)$$



Share of the copyright programmers a^*

Figure 2: Complementary income P as a function of the relative shares of copyright and copyleft programmers

[9] The parameterization looks complicated but satisfies $P\left(a^* \to 1\right) \to \infty, P\left(a^* \to 0\right) \to \frac{1}{2}\beta, P' > 0$ when $0 < a^* < 1$. It has merits, which become obvious when analyzing the monopolist's profit-maximization: the solution becomes algebraically easy and intuitive and yields simple subsequent results. Replacing (3) by i.e. a linear or exponential function for complementary income decreasing in the number of copyleft programmers doesn't affect the qualitative results.

As the firm wage is equal to the total complementary income for the indifferent programmer, combining (1) and (3) we have

$$w(a^*) = a^* P(a^*) = \beta\left(\frac{-\ln(1-a^*)}{a^*} - 1\right). \tag{4}$$

In (4), $w(a^*)$ is the wage level the monopolist has to offer to be able to hire $a^* N$ programmers. The wage level has a direct effect that affects the indifferent programmer and an indirect effect that changes the size of the copyleft community and through that the complementary income. In turn, the occupational choice determines the programming output of traditional, copyrighted programs and copylefted programs. Our specification of the complementary income implies that the monopolist cannot with his actions suppress the copyleft community completely. The ablest members always value copyleft work more than firm wage. The total development outputs X_R of programmers employed in firms and X_L of programmers in copyleft community are respectively (see figure 3)

$$X_R(a^*) = \frac{1}{2} a^{*2} N \tag{5a}$$

$$X_L(a^*) = \frac{1}{2}(1 - a^{*2}) N . \tag{5b}$$

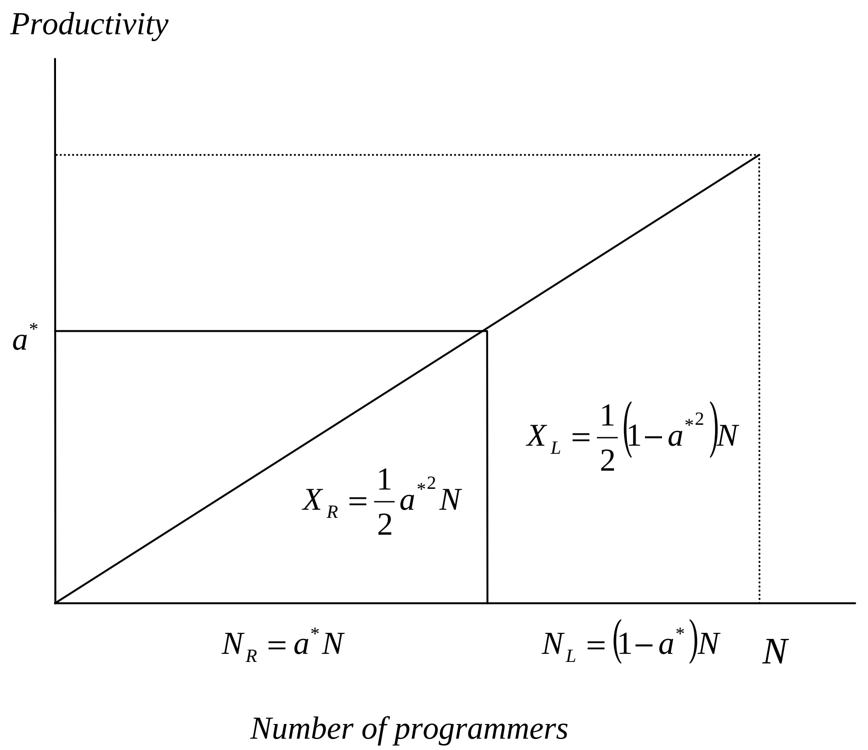


Figure 3: The numbers and total programming efforts of copyright and copyleft programmers

The total development cost to the monopolist, C, is simply the wage in (4) times the number of employed programmers (2b):

$$C(a^*) = w(a^*)a^* N = N\beta\left(-\ln(1-a^*) - a^*\right). \tag{6}$$

2.2 Market demand in the output stage

We assume that there is a software market supplied by a monopolist. The monopolist's program is traditionally protected by copyright. There is also a copyleft community in the sense that we assume that a GPL (General Public License) is available for programmers to license their contribution so the incentive structure discussed above is present. We denote the copyright program with R and the eventual copyleft program with L subscripts.

There are M consumers, who each buy at most one program product. We assume that their number is much higher than that of programmers, N. Their valuation of the copyright program is evenly distributed on the interval $[0, V_R]$ and the valuation of the copyleft program respectively on the interval $[0, V_L]$. The programs are substitutes so we assume that the ratio of the valuations is constant for all consumers. The valuation depends on the properties of the program. We assume that these properties are proportional to the total programming output in creation of the program. Let the valuation V_R for the program of the monopolist and the valuation of the copyleft program V_L be simple linear functions with the parameter $\mu > 0$ of the total respective programming outputs

$$V_R = \mu X_R(a^*) \tag{7a}$$

$$V_L = \mu X_L(a^*). \tag{7b}$$

Let the copyright program's price be p. We abstract from the differing skills of consumers and assume that all consumers face the same implementation costs for each program, $c_R \geq 0, c_L \geq 0$, respectively. The efforts of the copyleft community are directed to developing a program that has similar functions as the monopolist's product as we assume that the skills of programmers are industry-specific.[10] In this market the programs are imperfect substitutes. The marginal consumer j buys the copyright program if the surplus to him from it is larger than that from the copyleft program and naturally at least zero. This condition can be expressed as

$$V_{Rj} - p - c_R > V_{Lj} - c_L \quad => V_{Rj} > \frac{p + c_R - c_L}{1 - \frac{V_L}{V_R}} \; (> 0). \tag{8}$$

From (8) we can see that the presence of copyleft program affects the monopolist's behavior. If for some consumers the valuation of the copyleft program exceeds its

[10] Windows and Linux could be an example.

implementation cost, the monopolist has to take this into account when setting the profit-maximizing price. Consumers whose valuation fulfills the condition in (8) buy the copyright program. This is represented by distance OR in figure 4.

Whether consumers choose to acquire the copyleft program for free depends on the valuation of the marginal consumer k. The condition for consumer k to choose the copyleft program is

$$V_{Lk} - c_L \geq 0 \tag{9}$$

If the alternative surplus of buying the copyright program is negative for the marginal consumer k then the copyleft program is present in the market and the monopolist supplies the residual demand. Figure 4 illustrates this outcome. The distance OL-OR=RL represents the number of consumers acquiring the copyleft program and the distance OR represents the number of consumers buying the copyright product. The opposite possibility is that there are no consumers that would receive a positive surplus from the copyleft program and a negative one from the copyright program. In figure 4 this would mean that point L would be to the left of point R. In that case, while there is a programming effort by the copyleft community, no consumers use the developed program. Combining the condition (9) for the marginal consumer to the condition in (8) yields that the requirement for no consumers to use the copyleft program is

$$p < \left(\frac{V_R c_L - V_L c_R}{V_L} \right) . \tag{10a}$$

The monopolist can control the presence and impact of the copyleft program in the market. First of all, the price he sets determines whether consumers use the copyleft program. Apart from that, as noted earlier, the presence of the copyleft community and even the potential threat of a substitute program have an impact on the monopolist's pricing. To make the exposition clearer, we assume for the rest of the paper that the implementation costs are equal to both programs $c_R = c_L = c$. The condition (10a) for no market presence for the copyleft program in the market simplifies to

$$p < \left(\frac{V_R}{V_L} - 1 \right) c . \tag{10b}$$

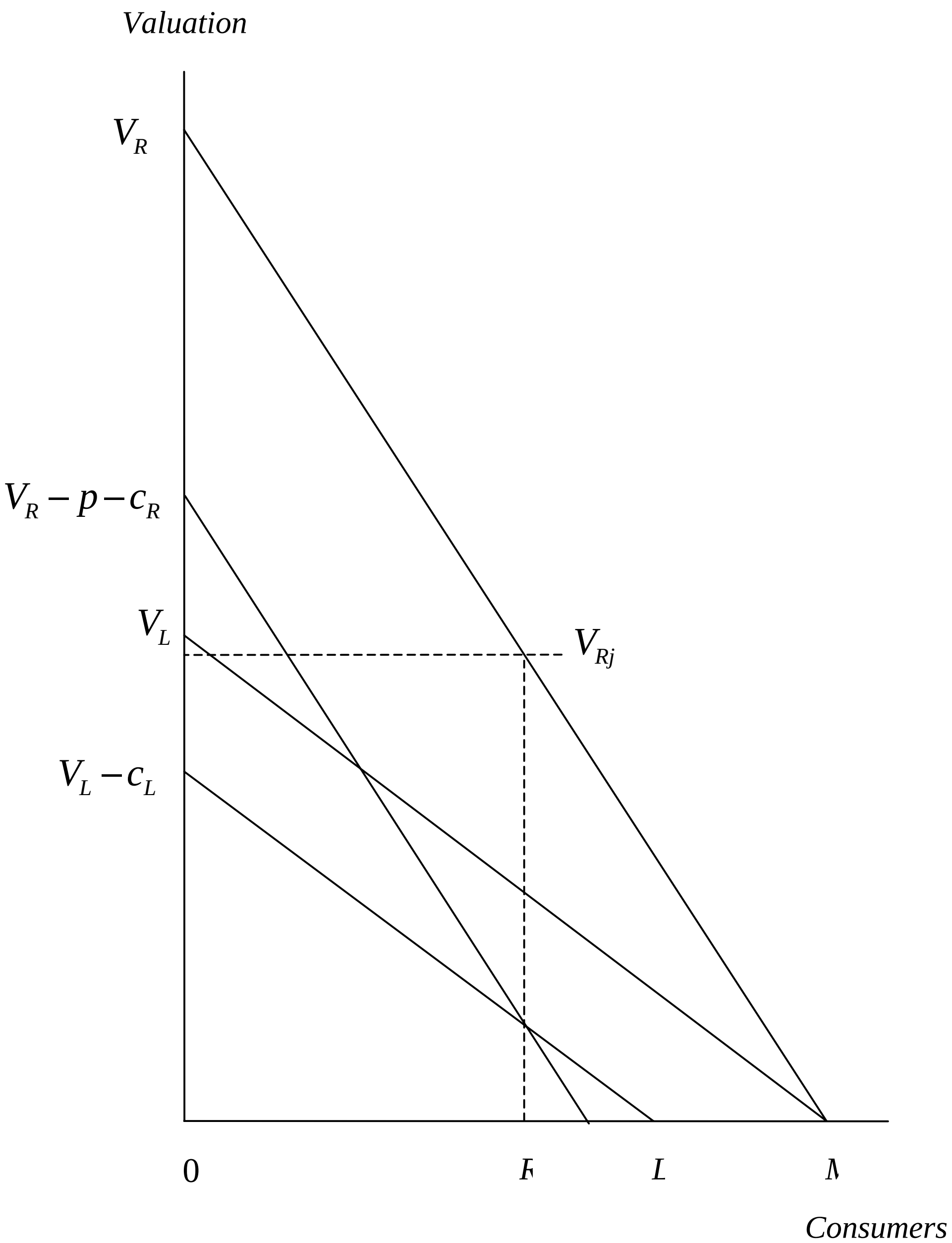


Figure 4: The market for copyright and copyleft programs

The presence of the copyleft community leads to the following kinked demand function for the monopolist:

$$q = \frac{V_R - p - c}{V_R} M , \qquad \text{when } p \le \left(\frac{V_R}{V_L} - 1 \right) c \tag{11a}$$

$$q = \frac{V_R - \frac{p}{1-\frac{V_L}{V_R}}}{V_R}, \quad \text{when } p > \left(\frac{V_R}{V_L} - 1\right)c. \tag{11b}$$

We assume that both of these information goods, programs, have zero unit costs. The revenue for the monopolist is

$$R = p\left(\frac{V_R - p - c}{V_R}\right)M \qquad \text{when } p \leq \left(\frac{V_R}{V_L} - 1\right)c \tag{12a}$$

$$R = p\left(\frac{V_R - \frac{p}{1-\frac{V_L}{V_R}}}{V_R}\right)M \text{ when } \quad p > \left(\frac{V_R}{V_L} - 1\right)c. \tag{12b}$$

The revenues in (12) are realized only if for all consumers the valuation of the copyright product is greater than the valuation of the copyleft product. In the opposite situation, all consumers choose the copyleft product. This leads to the following proposition regarding the monopolist's behavior in the development stage.

Proposition 1: (The participation constraint for the monopolist) The presence of the copyleft community determines an absolute lower limit for the productivity of the best employed programmer $\left(a^*_{\min} > \left(\sqrt{2}\right)^{-1}\right)$ and to the wage $\left(w^*_{\min} = w\left(a^*_{\min}\right) > 0{,}52\beta\right)$ the monopolist has to pay to the programmers in the development stage.

Proof: From (8) we note (assuming $c_R = c_L = c$) that to be present in the market the monopolist cannot allow the maximum valuation for the copyleft program to be higher than the maximum valuation for the copyright program. Then no consumers would receive positive surplus from buying the copyright program at a positive price since they can acquire the copyleft program of higher quality for free. The condition is satisfied if the total development output for the copyright program is larger than that for the copyleft program. The monopolist determines the outputs by setting the wage level. Combining (5) and (7) and inserting in (4) the condition can be expressed:

$$V_R > V_L \Rightarrow a^* > \left(\sqrt{2}\right)^{-1} = a^*_{\min} \Rightarrow w\left(a^*_{\min}\right) > 0{,}52\beta \quad \text{Q.E.D} \tag{13}$$

We can now solve the optimal prices for the monopolist's program as functions of the given productivity of the best programmer employed by the monopolist, a^*. The monopolist determined the value of this variable in the development stage by setting the wage to programmers. As the implementation cost is an interesting parameter in software markets, we solve the optimal prices for different levels of c:

$$p = \frac{\mu N a^{*2} - 2c}{4} \qquad \text{if} \quad c > \frac{\mu N a^{*2}\left(1 - a^{*2}\right)}{6a^{*2} - 2} = \bar{c} \tag{14a}$$

$$p = \frac{2a^{*2} - 1}{a^{*2}} c \qquad \text{if} \quad \underline{c} < c < \bar{c} \tag{14b}$$

$$p = \frac{\mu N\left(2a^{*2} - 1\right)}{4} \qquad \text{if} \quad c < \frac{\mu N\left(1 - a^{*2}\right)}{4} = \underline{c}\ . \tag{14c}$$

Proposition 2: If the level of the implementation cost is below a lower limit, $c < \underline{c}$, the copyleft program is present in the market.

Proof: The result follows from comparing (14a)-(14c) Q.E.D

If the implementation cost is high enough, no consumers with a positive surplus from the copyright program receive positive surplus from the copyleft program. The monopolist can price his program as if there were no copyleft program present. If the distribution cost is between the threshold values, the optimal price for the monopolist is the price that just deters the marginal consumer using the copyright program from receiving any positive surplus from the copyleft program. When the implementation cost is below a lower limit, $\underline{c}$, there are consumers choosing to use the copyleft program. The monopolist has to take this into account when setting the price. Ultimately, should the implementation cost be zero, all consumers use either the copyright or copyleft program.

3 Monopoly profit maximization

We solve the monopolist's profit maximizing problem using backward induction. In the development stage, the monopolist anticipates that the program market behaves in a manner described in chapter 2.2. He has to decide the ability of the best programmer he employs and this in turn determines the quality of his program and the quality of the program the copyleft community creates. Consumers' decisions depend on the qualities of programs and implementation costs. The level of implementation costs determines the market structure. The revenue in the output stage, (12), and the labor cost in the development stage, (6), lead to following profit functions. We assume that the monopolist has no other fixed costs than labor costs and that there is no discounting. Inserting the optimal prices in (14) and the valuations of the programs in (7) and the total development costs in (6) leads to the following profit functions in terms of the decision variable a^*:

$$\pi(a^*) = M\,\frac{\left(\mu N a^{*2} - 2c\right)^2}{8\mu N a^{*2}} - N\beta\left(-\ln\left(1-a^*\right)-a^*\right) \text{ when } c > \frac{\mu N a^{*2}\left(1-a^{*2}\right)}{6a^{*2}-2} = \bar{c} \quad \text{(15a)}$$

$$\pi(a^*) = M\left(\frac{\left[\left(2a^{*2}-1\right)\left(\mu N a^{*4} - 6a^{*2}c + 2c\right)\right]}{\mu N a^{*6}} c\right) - N\beta\left(-\ln\left(1-a^*\right)-a^*\right)$$

when $\bar{c} \geq c > \underline{c}$ (15b)

$$\pi(a^*) = M\,\frac{\mu N\left(2a^{*2}-1\right)}{8} - N\beta\left(-\ln\left(1-a^*\right)-a^*\right) \quad \text{when} \quad c \leq \frac{\mu N\left(1-a^{*2}\right)}{4} = \underline{c}. \quad \text{(15c)}$$

In the analysis we concentrate on the scenario where the copyleft program is present in the market (15c). We solve the monopolist's profit maximization problem and determine when the monopolist enters the industry.

Proposition 3: When the level of implementation cost is sufficiently low $(c < \underline{c})$ the monopolist employs a larger share of the programmers

- **the larger is the consumer market M**
- **the more the consumers value quality (the higher is μ)**
- **the lower the programmers' complementary income is (the lower is β)**

Proof: In the following we assume that the level of distribution costs, c, is low, fulfilling the condition in proposition 2. Solving the monopolist's profit maximization problem (15c) yields the optimal productivity level of the best programmer employed by the firm, a^{**} which in turn determines the optimal number of employed programmers $N_R^{**} = a^{**}N$.[11]

$$a^{**} = 1 - \frac{2\beta}{M\mu} \quad \text{(16)}$$

Inspection of (16) proves the proposition. Q.E.D

Proposition 4: When the level of implementation cost is sufficiently low $(c < \underline{c})$ the profit maximising wage the monopolist pays to the programmers is higher

- **the larger is the consumer market M**
- **the more the consumers value quality (the higher is μ)**
- **the higher the programmers' complementary income is (the higher is β)**

Proof: Inserting the optimal productivity of the best programmer employed, a^{**}, to the wage function (3) and differentiating wrt. the parameters yields the following comparative statics results:

[11] Details of the maximization procedure are in the appendix

$$\frac{dw(a^{**})}{dM} > 0, \ \frac{dw(a^{**})}{d\mu} > 0, \ \frac{dw(a^{**})}{d\beta} > 0. \qquad \text{Q.E.D} \qquad (17)$$

Proposition 5: (The entry condition for the monopolist) When the level of the implementation cost is sufficiently low $(c < \underline{c})$, the profit maximizing monopolist

- **enters the industry if $\beta < 0{,}056M\mu$.**
- **hires then at least 89% of the programmers.**

Proof: Inserting the optimal solution of the decision variable, a^{**}, to the profit function (15c) and setting it equal to zero yields[12] the following condition for the model parameters:

$$\pi(a^* = a^{**}) > 0 \Rightarrow \beta < 0{,}056M\mu. \qquad (18)$$

The monopolist determines to enter the industry in the development stage if the condition in (18) is satisfied. Applying the condition (18) to the optimal solution of the decision variable a^* in (16) allows us to characterize the employment decision of the monopolist.

$$\beta < 0{,}056M\mu \Rightarrow a^{**} > a^*_{entry} \approx 0{,}89. \qquad (19)$$

The monopolist hires in the development stage a^*N programmers. If he enters the industry the profit maximizing number of hired programmers is $a^{**}N$. The condition in (18) sets a lower limit to the level of employment in our model and it is 89% of all programmers. Q.E.D

When the condition in (18) is not satisfied we can interpreted it as a scenario where the level of complementary income in the software application area is large compared to the market. The approach taken in the model of Johnson (2000) – analyzing the copyleft activity as private provision of a public good – fits well with that situation. In his model the programmers are also the only users of the software and the consumer market is not present.

Proposition 6: There exists a maximum value of the lower limit of the implementation cost $\underline{c}_{\max} < 0{,}21\mu N$. If the implementation cost is higher than $0{,}21\mu N$ or more than 13% of the highest valuation of the program, V_R, the copyleft program is not present in the market. For the eventual consumers using the copyleft program $\underline{c}_{\max}$ is equal to 50% of the maximum valuation V_L.

Proof: The lower threshold for the implementation cost in (15c) is decreasing in a^* and reaches it's maximum at lowest possible value of a^*. Inserting the condition in (18) into the equation for the lower limit of the implementation cost in (15c) yields the proposition.

[12] We use a numerical method to solve the equation.

$$\underline{c}_{\max} < \underline{c}\left(a^* = a^*_{entry}\right) \approx 0{,}21\mu N \;. \tag{20}$$

To calculate the maximum ratio of the implementation cost to the maximum valuation of the monopolist's program we conclude that the lower limit in (15c) is decreasing and the maximum valuation of the program in (7a) is increasing in the decision variable. The maximum ratio obtains in the minimum value of the decision variable.

$$\frac{\underline{c}_{\max}}{V_R} < \frac{\underline{c}\left(a^* \approx 0{,}89\right)}{V_R\left(a^* \approx 0{,}89\right)} \approx 0{,}13 \,. \tag{21}$$

Comparing the equations for the lower limit for the implementation cost in (16c) and for the maximum valuation for the copyleft product (7b) we note that their ratio is a constant, always 50%. If the relative implementation costs of programs are above the level indicated in the condition (20) the copyleft program is not present in the market, regardless of other parameters. Q.E.D

In condition (20), we can see that the threshold level of the implementation cost is increasing in the size of the total resource for programming, N. Many determinants of the actual implementation cost are independent from the quality of the programs and more or less constant. This implies that consumers are more likely to use copyleft programs in software application areas where the programming resource is large.

4 Policy implications

Policy implication 1: When implementation costs are low $\left(c < \underline{c}\right)$, informing consumers about an unknown substitute copyleft program increases welfare with presumably low costs.

Let us assume two scenarios in the market. The first contains a monopolist supplying the market and a copyleft community of programmers. However, the consumers are not aware of the substitute copyleft product potentially available with nominal distribution cost. There is some realism in this assumption since the copyleft programmers are indifferent whether the consumers use or not their product. The second scenario is the one analyzed in 2.3 defined by (15c). In that scenario the consumers have full information on the substitute copyleft product. To make the analysis more tractable we approximate low implementation costs by assuming that they are zero, $c = 0$.

Maximization of profit in both scenarios yields the following optimal productivities of the best hired programmers (we denote the first scenario with M, the second with CL):

$$a^{**}_M = 1 - \frac{4\beta}{M\mu} \tag{21a}$$

$$a^{**}_{CL} = a^{**} = 1 - \frac{2\beta}{M\mu} \tag{21b}$$

Comparison of (21a) and (21b) shows that the monopolist will always hire less programmers and the quality of the copyright product is lower when consumers are unaware of the copyleft substitute. We compare the resulting welfare levels, that is the sums of firm profit and consumer surplus in the appendix. The comparison shows that welfare is always higher when consumers are aware of the copyleft product. The monopolist has no incentive to inform the consumers because when consumers are aware of the copyleft program his profit is always lower. An intriguing outcome is that the market may become unprofitable to the monopolist when consumers learn about the substitute copyleft program. The market and cost parameters that resulted in profit when the monopolist was alone in the market may not fulfill the entry condition in proposition 4. This implies that when the monopolist anticipates a policy of informing consumers of copyleft programs the condition for commitment to the programming investment in the development stage is the one in proposition 4.

Policy implication 2: Society's support for copyright enforcement is important for copyleft activity.

Copyleft incentives rely on copyright enforcement. The literature on copyright (eg. Besen and Raskind 1991, Landes and Posner 1989) generally considers the enforcement of the copyright to be the responsibility of the author. Landes and Posner assume in their model that securing copyright is not complete and increasing the level of protection is costly. This means that authors, usually firms, with considerable resources have the possibility to defend their intellectual property. Copyleft programmers or communities however do not usually possess such resources. As we noted earlier, copyleft programmers are indifferent whether consumers use or not the copyleft programs. However, if the distribution of the copyleft program outside the copyleft community results in violations of the 'collective' copyright of the program they may prefer not to allow it. The economics of copyright protection analyses the optimal level of copyright protection (Landes and Posner 1989, Koboldt 1995, Johnson 1985, Novos and Waldman 1984). There is a tradeoff between higher incentives to create new works when copyright protection is high and on the other hand the increased opportunity to create derivative works and lower control costs when protection is low. The existence of copyleft communities is an additional variable to this analysis. The incentives present in copyleft activity benefit from a high level of institutional copyright protection. As the consumption of programs created by the copyleft community seems to increase welfare this advocates higher copyright protection. We compared copyleft activity to 'Science' in the introduction. It is interesting to note that scientific publications are an institution that largely protects the intellectual property rights (in the sense of ***priority***) of the scientific community. Copyleft communities lack presently such institutions. The increased economic significance of copyleft software has already resulted in a discussion on the need of such institutions (see Deckmyn 2000a).

5 Discussion

The novel contribution of this paper is to model simultaneously the impact of copyleft licensing to both the development environment and the consumer market of programs. The effect of copyleft to the incentives and conduct of programmers has been noticed in the literature. In our model we extend the analysis by modeling a monopolist supplying copyright protected programs. Due to copyleft activity he faces constraints in the programmer labor market and competition from a substitute copyleft program in the consumer market. The presence of the copyleft program is dependent on the level of consumer implementation costs for programs. When the cost is sufficiently low, some consumers choose to use the copyleft program and the monopolist has to take this into account when pricing his program. The presence of copyleft activity also increases the constraints for market participation for the monopolist. He cannot allow the quality of the copyleft program to be higher than the quality of his program. The larger is the consumer market compared to the programming population, the larger share of the programmer population the monopolist hires and the smaller is the copyleft community. If the market size is small and consumer valuations are low the monopolist may decide not to enter the market. Then there is available only the copyleft program. This result coincides with some real-world phenomena: in certain markets, for example for some network utility programs the supply are solely copylefted programs. Our results imply that the monopolist may not be able to apply full monopoly power in the market if the copyleft program is present. Schmalensee (2000) analyses the personal computer operating system market in the US and finds that the market leader, Microsoft, is in practice a monopoly, but doesn't apply monopoly pricing. This deviation is a result of several factors but our analysis provides the explanation that the 'invisible' competition from Linux affects Windows pricing.

For programmers copyleft licensing creates an alternative incentive structure reminiscent of scientific research. The assumptions of our model mean that the ablest of programmers join the copyleft community. There is parallel empirical evidence supporting this result (see Stern 1999) but naturally our results rely on these assumptions. Furthermore, even if case studies (Lerner and Tirole 2000) of copyleft program projects imply that some programmers choose to engage in copyleft programming instead of highly paid copyright programming, modeling of the complementary income needs further empirical study of the incentives of programmers. We model the program market being supplied by a monopolist. This is clearly a simplification of the real world. The program business is sequential in nature: first there is the development stage and then the production of the program. For the firms to retrieve the programming investment in the market imperfect competition of some degree is required. Relaxing the monopoly assumption to oligopoly doesn't change the qualitative results. In more general terms the partial equlibrium nature of our model hides some important issues. Here we assume that the complementary income is an outside option from the model's point of view. Looking at the whole economy in general equlibrium terms raises the question of the resources for the complementary income. A copyleft community may be present in each sub-industry and copyleft products may dominate some markets. In this environment we can ask who provides the income for the copyleft programmers? In the taxation of suppliers of copyright programs the society has to take into account the substitute nature of the copyleft programming it may support from public funds.

There are numerous avenues for future research concerning this topic. First of all, our model doesn't address network effects for programs. They are an important property of programs and in the case of copyleft a new issue arises: The network for the copyright program are the consumers using it. We can think that the network for the copyleft program are the consumers using it and the programmers belonging in the copyleft community. This creates an environment where a copyleft program may enter a market dominated by copyright programs more easily if it has a large number of developers. The incentive structure of copyleft programmers may also have other implications. In the standard literature on signaling models where potential employees signal with the level of education it is assumed that the amount of education acquired doesn’t affect the employer’s profit. Copyleft programming, if it is used as a signal of skills may have a negative impact on the employer’s profit. The copyleft community develops in a decentralized manner a substitute program and this affects the market of the program of the employer. He will take this into account and this may change the results of the signaling analysis.

6 References

Arrow K. (1962) Economic welfare and the allocation of resources for invention, in Nelson R. ed. The Rate of Inventive Activity: Economic and Social Factors, Princeton University Press, Princeton.

Besen S., Raskind L. (1991) An introduction to the law and economics of intellectual property, Journal of Economic Perspectives 5(1), 3-27.

Brooks H. (1994) The relationship between science and technology, Research Policy 23(5), 477-486.

Browne C. (1999) The economics of free software, www.ntlug.org/~cbbrowne/freecon.html (accessed January 2000).

Dalle J-M., Juillien N (1999) Nt vs. Linux, or some explorations into the economics of free software, mimeo, ENS Cachan & IDHE-CNRS.

Dam K. (1995) Some economic considerations in the intellectual property protection of software, Journal of Legal Studies 24(2), 321-377.

Dasgupta P., David P. (1987) Information disclosure and the economics of science and technology, CEPR Discussion Paper 73, also in Feiwel D. Arrow and the ascent of modern economic theory, New York University Press, New York.

Dasgupta P., David P. (1994) Toward a new economics in science, Research Policy 23(5), 487-521.

David P. (1993) Intellectual property institutions and the panda's thumb: patents,copyrights and trade secrets in economic theory and history, in Wallerstein M., Mogee M., Schoen R. ed. Global Dimensions of Intellectual Property rights in Science and Technology, National Academy Press, Washington, US.

Deckmyn D. (2000a) Are open-source licenses legal? Computerworld July 24, 14.

Deckmyn D. (2000b) Sun goes open source with office software, Computerworld July 24, 14.

Deckmyn D. (2000c) Report: Linux won't beat Windows, Computerworld July 31, 14.

Detriwapont M., Jewitt I., Tirole J. (1999) The economics of career concerns I&II, Review of Economic Studies 66(1), 183-217.

GNU Project (2000a) Categories of free and non-free software, www.gnu.ai.mit.edu/philosophy/categories.html (accessed June 2000).

GNU Project (2000b) What is copyleft?, www.gnu.ai.mit.edu/copyleft/copyleft.html (accessed June 2000).

Holmström B. (1999) Managerial incentive problems. A dynamic perspective, Review of Economic Studies 66(1), 169-182.

Jagren L., Morell E. (2000) Hårda data om den nya ekonomin (in swedish), Industriförbundet, Sweden.

Johnson J. (1999) Economics of open software, mimeo, MIT.

Johnson W. (1985) The economics of copying , Journal of Political Economy, 93, 158-174.

Kemppinen J. (2000) Tietokoneohjelman oikeussuoja I-IV (in finnish), www.hut.cs.fi/~mam/DE.html.

Koboldt C. (1995) Intellectual property and optimal copyright protection, Journal of Cultural Economics, 19(2), 131-155.

Landes W., Posner R. (1989) An economic analysis of copyright law, Journal of Legal Studies 18(2), 325-363.

Lerner J., Tirole J. (2000) The simple economics of open software, NBER Working Paper 7600.

Novos I., Waldman M. (1984) The effect of increased copyright protection: An analytic approach, Journal of Political Economy 92(2), 236-246.

Open Source Initiative (2000) Open source definition, www.opensource.org/osd.html (accessed August 2000).

Samuelson P. (1993) A case study on computer programs, in Wallerstein M., Mogee M., Schoen R. ed. Global Dimensions of Intellectual Property rights in Science and Technology, National Academy Press, Washington, US, 248-318.

Schmalensee R. (2000) Antitrust issues in Schumpeterian Industries, American Economic Review 90(2), 192-197 .

Stephan P. (1996) The economics of science, The Journal of Economic Literature 34(3), 1199-1235.

Stephan P., Levin S. (1996) Property rights and entrepreneurship in science, Small Business Economics 8(3), 177-188.

Shy O., Thisse J.-F. (1999) A strategic approach to software protection Journal of Economics and Management Strategy 8(2), 163-190.

Stern S. (1999) Do scientists pay to be scientists? NBER Working Paper 7410.

Takeyama L. (1994) The welfare implications of unauthorized reproduction of intellectual property in the presence of demand network externalities, Journal of Industrial Economics 42(2), 155-166.

Takeyama L. (1997) The intertemporal consequenses of unauthorized reproduction of intellectual property, Journal of Law and Economics 40(2), 511-522.

Varner P. (2000) The economics of open source software. Open software economic models. www.cs.virginia.edu/~pev5b/econ.oss/econ_models.html (accessed August 2000).

Appendix

The monopolist's profit maximization (15c)

The first order condition for the monopolist's profit maximization is:

$$\frac{d\pi}{da^*} = \tfrac{1}{2}\mu NMa^* - \frac{N\beta a^*}{1-a^*} = 0 \,. \qquad \text{(A.1)}$$

This yields two solutions for the optimal a^*:

$$a_1^{**} = 0 \qquad \text{(A.2)}$$

$$a_2^{**} = 1 - \frac{2\beta}{M\mu} \,. \qquad \text{(A.3)}$$

The second order condition for maximum is

$$\frac{d^2\pi}{da^{*2}} = \tfrac{1}{2}\mu NM - N\beta\left(\frac{1}{\left((1-a^*)\right)^2}\right) < 0 \,. \qquad \text{(A.4)}$$

Inserting $a_1^{**} = 0$ yields $\frac{2\beta}{M\mu} > 1$ and inserting $a_2^{**} = 1 - \frac{2\beta}{M\mu}$ in turn yields $\frac{2\beta}{\mu M} < 1$. We assume that the latter inequality holds as we also assumed that the number of consumers is large compared to the number of programmers. Inserting a_1^{**} to the profit function results in negative profit so the only profit maximizing solution is a_2^{**}.

Policy implication 1: Informing consumers

The first order condition to maximize profit (15a) in the absence of implementation costs is:

$$\frac{d\pi_M}{da^*} = \frac{1}{4}\mu N M a^* - \frac{N\beta a^*}{1-a^*} = 0. \tag{A.5}$$

This yields the optimum level of the best programmer employed,

$$a_M^{**} = 1 - \frac{4\beta}{M\mu}. \tag{A.6}$$

The second order condition and profit level analysis are similar to (A.4).

Welfare, that is the sum of the firm profit and consumer surplus, when consumers are not aware of the potential copyleft program is

$$W_M = \frac{3}{16} NM\mu a_M^{**2} - N\beta\left(-\ln\left(1-a_M^{**}\right) - a_M^{**}\right). \tag{A.7}$$

If consumers are aware of the copyleft product some of them derive surplus from the use of it. The welfare measure is then

$$W_{CL} = NM\mu\left(\frac{1}{8}a_{CL}^{**2} + \frac{1}{16}\right) - N\beta\left(-\ln\left(1-a_{CL}^{**}\right) - a_{CL}^{**}\right). \tag{A.8}$$

Comparing welfare measures yields the following inequality (we denote $\frac{2\beta}{M\mu} = Z$)

$$W_M > W_{CL} \Rightarrow \frac{1}{4}Z^2 - (1-\ln 2)Z > 0. \tag{A.9}$$

The inequality holds if $Z >\approx 1{,}53$. Comparing the profits in the scenarios results in a following inequality

$$\pi_M > \pi_{CL} \Rightarrow Z\left(\ln 2 - \frac{1}{2}Z\right) > 0. \tag{A.10}$$

This inequality holds when $Z < 2\ln 2 \approx 1.39$ and according to our assumptions in the model the possible values of Z are $0 < Z < \frac{\sqrt{2}-1}{\sqrt{2}} \approx 0{,}297$. So welfare is always higher when consumers are aware of the existence of the copyleft program and the monopolist has no incentive to inform the consumers of the potential substitute copyleft program.