# Copyleft - the economics of Linux and other open source software

Mikko Mustonen*

University of Helsinki, Department of Economics

**Abstract**

*A large population of programmers develops freely distributed software. A novel licensing scheme – copyleft – creates an incentive structure based on complementary income. We build a model in which the occupational choices of programmers determine the qualities 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 accepts the copyleft program in the market. Our model explains the simultaneous existence of commercial and free copylefted programs and also why commercial alternatives to copyleft programs may not exist.*

*Keywords:* Property rights; Copyleft; Linux; Open source software

*JEL Classification:* D23; L11; L15

(Submitted to *Information economics and policy*)

* P.O. BOX 54, FIN-00014, Finland. Tel. 358-400 200480. E-mail address: 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 incentives alternative to employment allowing them to signal their abilities and receive complementary income. As investments in computer software are becoming quite large, for example representing 11% of total national investments in 1998 in Sweden (Jagren and Morell, 2000), the efficiency of and incentives for software development work are an increasingly important 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.

We ask how copyleft affects the behavior of a monopolist who invests in the quality of his program at the development stage. A monopolist can influence the occupational choice of programmers through his wage policy. The programmers choose whether to be employed by the firm or to join the copyleft community to develop a copylefted program jointly 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. At the output stage, the monopolist supplies a program protected by copyright. It has a market of its own and there are no substitutes except the program eventually developed by the copyleft community. Consumers value quality and determine whether to buy the copyright program or acquire the copyleft program or not to use either of these. This creates a trade-off for the monopolist: paying a higher wage at the development stage increases costs but also increases 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 this 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 policy actions. The paper also contributes to the literature of vertical product differentiation. A novel industry structure arises: the decisions of a single firm, the monopolist, determine the qualities of all (two) products in the market. To bring out the effects copyleft licensing has on labour and product markets, 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. Program implementation costs determine 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 consumers preferring the copylefted program and this creates a constraint on profit maximization.

[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 then be the programmers working on the project initiated by Linus Torvalds and the Linux the copyleft product.

There is an intermediate interval for implementation costs, at which 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 the consumer market compared to the population of programmers is the more the monopolist employs programmers and the smaller the copyleft community is. The monopolist employs a large fraction of programmers if their complementary income from copyleft work is low or if the consumers' valuation of the quality of the programs is high. In the presence of copyleft, the consumer market has to be large compared to the total programming population. There is a threshold market size below which the monopolist cannot profitably develop a program. Casual empiricism suggests 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 also has policy implications. The incentives for copyleft programming are independent of the consumer market. Programmers do not care whether there are users of the copyleft program outside the copyleft community. It is possible that a copyleft substitute program may exist that consumers are not aware of. Informing consumers of such a program is likely to increase welfare with presumably low costs. 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 does not 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, beside copyrighting the program, also signs a General Public License (GPL) (GNU Project 2000b) granting everyone the right to use, modify and distribute the program *on the condition that the licensee also grants similar rights over the modifications he has made*. Under this arrangement, the program is simultaneously freely usable but protected from becoming someone's private intellectual property.[2] This allows decentralized program development because the enhancements and modifications accumulate to the basic program even if the programmers have no other affiliation with the project. For this analysis, however, we are interested in the other significant feature of copyleft as a device for 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.

Historically, copyleft licensing was created for ideological purposes by Richard Stallman and the Free Software Foundation (GNU Project 2000b). However, the

[2] Lerner and Tirole (2000) and Johnson (1999) provide detailed descriptions of the licensing schemes (GPL, Open Source Software, Debian Social Contract) that create the copyleft environment. The Open Source Initiative (2000) contains the definition of open source software and the GNU Project (2000a) a classification of free and non-free programs. Browne (2000) provides a practitioner's view of copyleft licensing.

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. Copylefted programs have gained significant market shares in a short time. 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 several million systems (Lerner and Tirole 2000). Having both commercial copyright and non-commercial copyleft programs present in the marketplace is a reality. However, the low cost of copyleft programs means that 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 at twice the speed of Windows, the turnover from the Linux business is projected to be only a few percent 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 for copyleft products. Perhaps 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 being an example. 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 for some existing copyleft programs like Linux. SUN Microsystems has also developed programs (Staroffice) that are substitutes for existing office applications and copylefted them (see Deckmyn 2000b). Lerner and Tirole (2000) and Subramanian (2000) discuss the intriguing 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 information as a public good and the incentives to create new information (Arrow 1962). Information, like a computer program, once it is created, is practically costless to reproduce. From society's point of view, it should be distributed freely. However, without incentives there will then be no creation of new information. 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 the inventor a temporary monopoly over the invention. Copyright protects the rights of the originator of a unique expression, for example a work of art, for a fixed period.[3]

From the intellectual property rights perspective, a computer program is a problematic object (Samuelson 1993, Dam 1995). It can be a unique expression like a poem, and copyright over it would seem an appropriate method of protecting the rights of the

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

originator. But a program, once it is running, also creates functions, quite like a machine. New and novel functions or uses for a function may seem to 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 creating and modifying algorithms. As a result of the first lawsuits over the ownership of property rights for software, copyright is generally the method of securing property rights. Presumably the original motivation of the Free Software Foundation to introduce the copyleft licensing scheme was based on this development. Copyright over programs was considered to restrict innovation in programming because of its uniqueness requirement. Recently, however, in the US patents have also been granted to programs and the EU considers presently a common policy on software patenting.

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 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 the suppliers' incentives are secured by property rights and copyleft software development where programmers give away their rights at 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 the 'Science' environment to the 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 also contains 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 the 'Science' and 'Technology' environments. For the same performance in the 'Science' environment, the level of expected direct monetary pecuniary elements is usually lower than in 'Technology' (see Stern 1999) and because of the complementary nature of income they may be linked to less interesting activities, like for example teaching or project management.

[4] Brooks (1994), Stephan (1996) and Stephan and Levin (1996) discuss similar issues.

However, the elasticity of the compensation with regard to the performance may be much higher in the 'Science' environment. An able scientist is almost certainly rewarded for his contributions because, according to the priority principle, they become public and receive the appreciation of peers. In the 'Technology' environment it is probable that the incentives in the firms do not take the contributions of individual employees fully into account, if at all. 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 in which 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 skill signaling is an essential factor in the incentives for copyleft work. Lerner and Tirole concentrate on the effectiveness, longevity and structure of decentralized open source software projects in comparison to the traditional hierarchical commercial projects. Johnson (1999) also focuses on the effects copyleft licensing has on program creation. He models copyleft activity as private provision of a public good: programmers receive utility from the 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. 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 copyleft activity as 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 program to users 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 for labor markets and software creation.

Copyright is an important method of securing intellectual property rights, and 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 protection is strong the authors face little competition from unauthorized copies in the market and profits are high. The trade-off is that a high level of copyright protection also inhibits authors of new works from utilizing existing works and the costs of creating new works are higher. Koboldt (1995) develops these ideas further by modeling a market in which 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 the cost 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. These models include 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 this paper is the following: in chapter 2 we first develop the cost function of the monopolist at the development stage based on the incentive structure implied by copyleft. We then 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 the paper.

# 2 The Model

The setup of our model is a two stage one. At the development stage a monopolist and a copyleft community engage in programming projects. At the output stage the consumers value programs and the monopolist sets a profit maximizing price for 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: at the development stage the monopolist anticipates the behavior of the program market at the output stage and maximizes profit by hiring the optimal number of programmers. The timeline of the model is in figure 1.

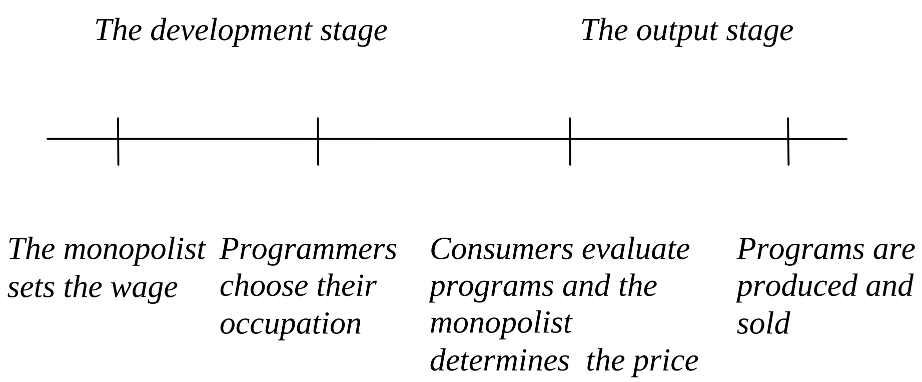


*Figure 1: Order of events in the model*

The monopolist invests in the quality of his product, a program, at the development stage. The quality is dependent on the programming output and the monopolist can determine this output by hiring programmers, whose heterogeneous ability is unobservable in employment. Copyleft licensing creates an alternative for programmers: to engage in copyleft work and receive 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 for 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.

At the output stage, consumers choose between buying the monopolist's program, acquiring the copyleft program developed by the copyleft community for free or not using any program. We assume that consumers face implementation costs for the programs. These costs include 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 implementation cost. Both technology and the skills of users influence these costs. While new programs have more complex properties, much of the programming effort is directed to enhancing 'ease-of-use'. A general trend in society is an increase in computer skills or 'literacy', both because of public education and user experience. This tends to decrease the implementation cost.

Copyright protection ensures that at the output stage the monopolist can price his product without the threat of illegal copies. Copyright protection is also essential 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. At 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.

### 2.1 Programmers' occupational choice at the development stage

We assume that a programmer has several career alternatives to choose from. Programmers can find employment in a software firm, become entrepreneurs by starting a business or engage in copyleft programming work. The combination of programming and entrepreneurial skills is likely to be rare, so we assume that the number of programmers choosing entrepreneurship 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. 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. This can be supported by casual empiricism 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.[5] The compensation for creating copyleft programs has to be of a 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 that able programmers can build a reputation that results in future complementary income, such as partnership in software ventures, grants or academic employment. Lerner and Tirole (2000) aggregate these career

[5] This simplifying assumption can be relaxed as long as the complementary income is more elastic than wage with regard to productivity.

concern[6] and ego gratification incentives "under a single heading: signaling incentives". They see these incentives as stronger the more visible the performance and the more dependent it is on effort and the more revealing it is about talent. The empirical analysis of Stern (1999) points out that the most able of the population tend to attach themselves to 'Science' community which corresponds to copylefting 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. The expected total complementary income for programmer $i$ is then $a_i P$. 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.[7] The number of copyleft programmers, $N_L$, and firm programmers, $N_R$, depends on the threshold productivity $a^*$,

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

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

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 more the smaller the community is. This assumption is in line with the analysis by Lerner and Tirole (2000) and Johnson (1999) 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 restrictive factor. To express the relation simply, we define the complementary income as 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

---

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

[7] 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.

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 conforms 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 $P$[8] be a function of the threshold productivity $a^*$ with the parameter $\beta > 0$ describing the level of complementary income (see figure 2),

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

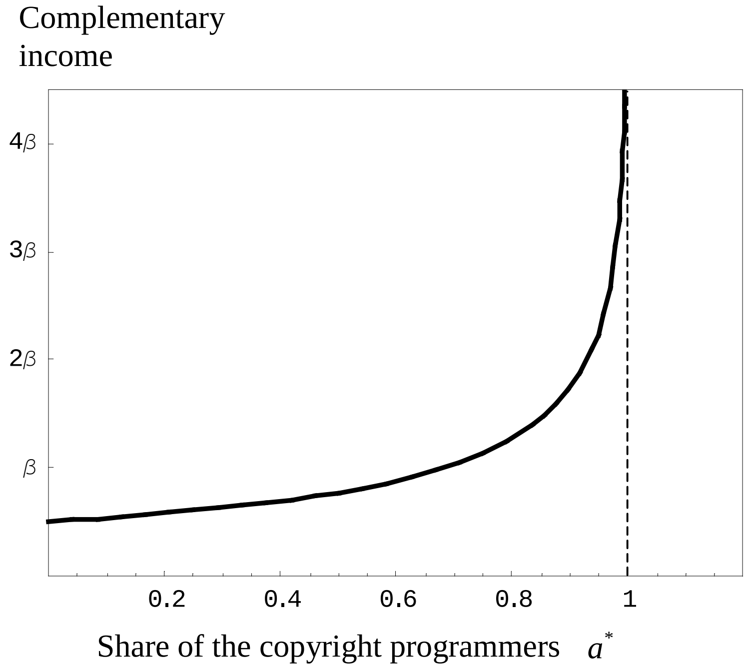


*Figure 2: Complementary income as a function of the relative shares of copyright and copyleft programmers*

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

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

---

[8] 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 that become obvious when analyzing the monopolist's profit-maximization: the solution becomes algebraically easy and intuitive and yields simple subsequent results. Replacing (3) by a linear or exponential function for complementary income decreasing in the number of copyleft programmers does not affect the qualitative results.

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 on 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 choices determine the programming outputs for the traditional, copyrighted program and the copylefted program. Our specification of the complementary income implies that the monopolist cannot suppress the copyleft community completely by his own actions. The ablest members always value copyleft work more than the firm's wage. The total development outputs $X_R$ of programmers employed in firms and $X_L$ of programmers in the copyleft community are then

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

$$X_L(a^*) = \frac{1}{2}\left(1 - a^{*2}\right) N \ , \qquad (5b)$$

respectively (see figure 3).

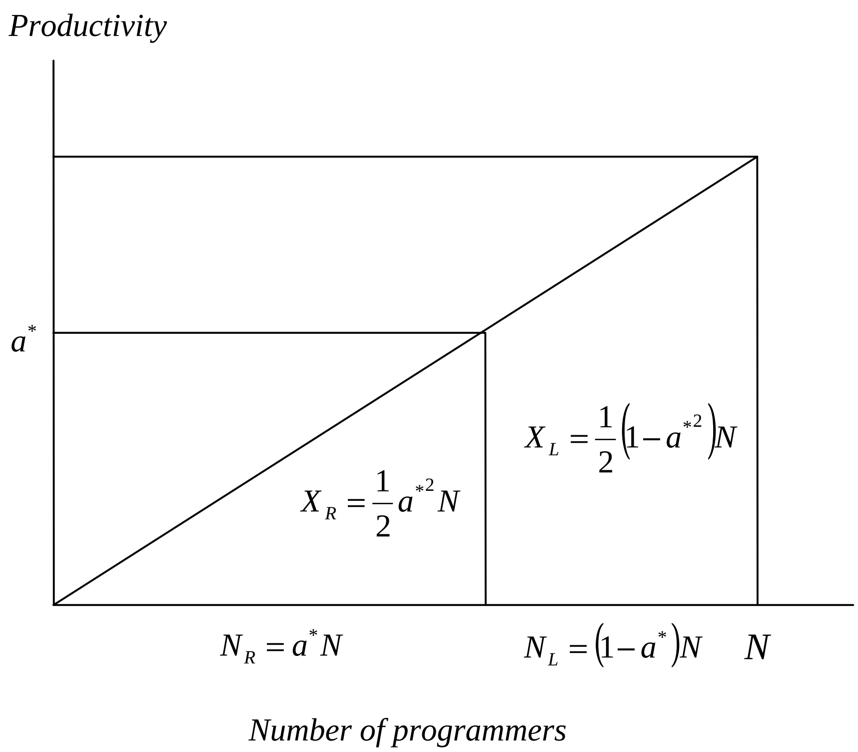


*Figure 3: The numbers and total programming efforts of copyright and copyleft programmers*

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

$$C(a^*) = w(a^*)a^* N = N\beta(-\ln(1-a^*) - a^*). \qquad (6)$$

### 2.2 Market demand at the output stage

We assume that there is a consumer market supplied by a monopolist. We denote the copyright program by $R$ and the eventual copyleft program by $L$ subscripts. There are $M$ consumers, who each buy at most one program product. Their valuation of the copyright program is evenly distributed on the interval $[0, V_R]$ and the valuation of the copyleft program on the interval $[0, V_L]$, respectively. The programs are substitutes and we assume that the ratio of the valuations is constant for all consumers. Each valuation depends on the properties of the program. We assume that these properties are proportional to the total programming output in creating the program. Let the highest valuations $V_R$, $V_L$ be simple linear functions with the parameter $\mu > 0$ of the total programming outputs

$$V_R = \mu X_R(a^*) \qquad (7a)$$

$$V_L = \mu X_L(a^*). \qquad (7b)$$

Let the price of the copyright program be $p$. We abstract from the differing skills of consumers and assume that all consumers face the same implementation costs for programs, $c_R \geq 0, c_L \geq 0$ respectively. The efforts of the copyleft community are directed to developing a program that has similar functions as the monopolist's product since we assume that the skills of programmers are industry-specific.[9] In the market the programs are (imperfect) substitutes. The marginal consumer $j$ buys the copyright program if the surplus accruing to him from it is larger than that from the copyleft program and naturally at least zero. To make the exposition clearer, we assume for the rest of the paper that the implementation costs are equal for both programs $c_R = c_L = c$. The condition reads

$$V_{Rj} - p - c > V_{Lj} - c \quad \Rightarrow V_{Rj} > \frac{p}{1 - \frac{V_L}{V_R}} \ (> 0). \qquad (8)$$

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

---

[9] Windows and Linux provide an example.

Whether any 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

$$V_{Lk} - c \geq 0 . \tag{9}$$

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 other possibility is that there are no consumers who 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 $L$ is 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. The monopolist can control the presence and impact of the copyleft program in the market. 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 optimal price. Combining condition (9) for the marginal consumer with the condition in (8) yields the condition for some consumers to use the copyleft program:

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

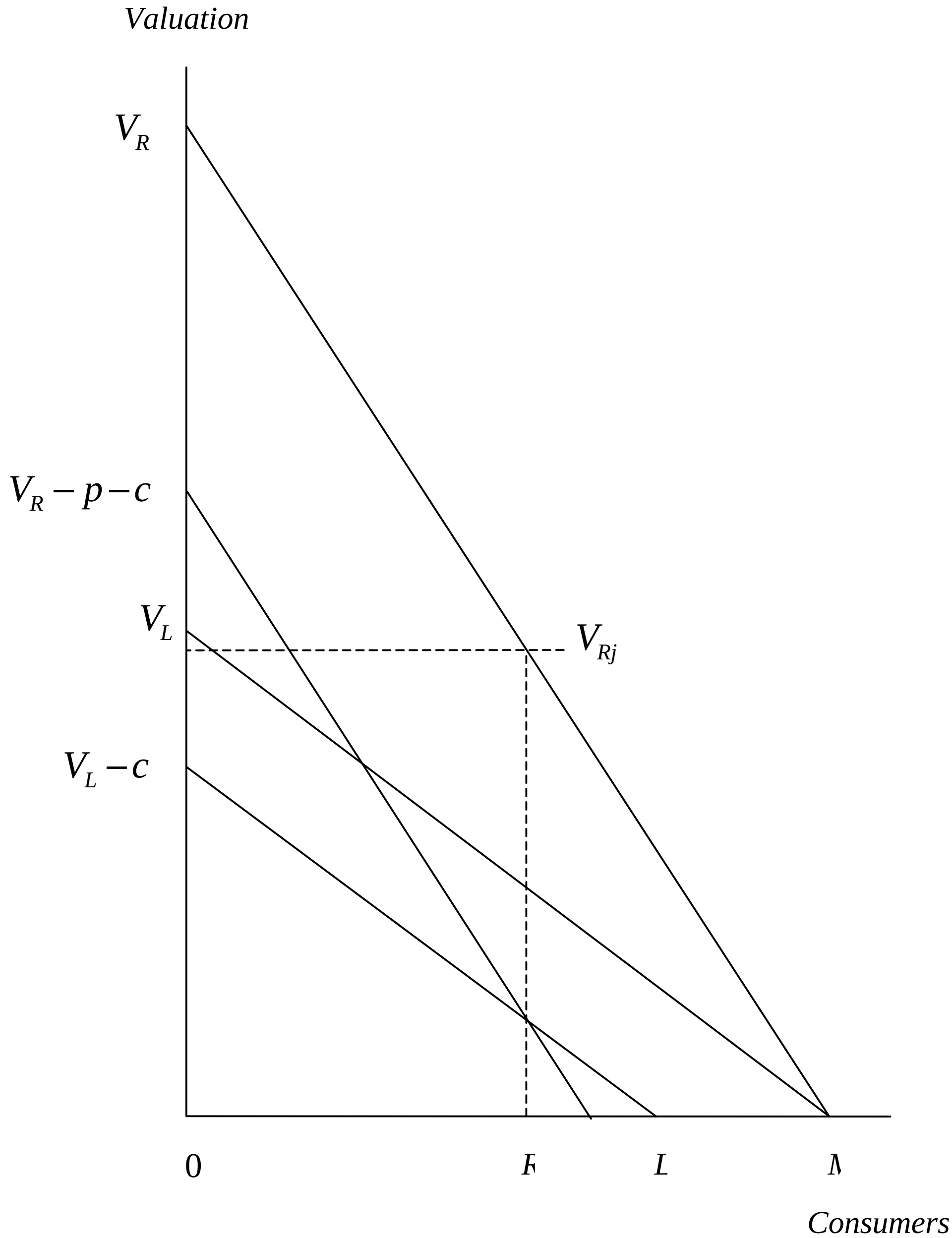


*Figure 4: The market for copyright and copyleft programs*

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

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

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

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

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

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

**Proposition 1: For the monopolist to participate in the market, the productivity of the best programmer employed, $a^*$, and the wage, $w$, must exceed threshold levels, $a^*_{\min} > \left(\sqrt{2}\right)^{-1}$ and $w_{\min} = w\left(a^*_{\min}\right) > 0{,}52\beta$, regardless of other parameters.**

**Proof**: From (8) we note 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. No consumers would then receive a 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 as:

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

Maximising the revenues in (12a,b) yields the optimal price with and without the copyleft product in the market. By inserting the valuations in (7a,b) we can express the prices as functions of the given productivity of the best programmer employed by the monopolist, $a^*$. The monopolist determined the value of this variable at the development stage by setting the wage for programmers. As the implementation cost is an interesting parameter in software markets, we solve the conditions in (12a,b) with regard to it. Inserting the respective optimal price and the valuations in the conditions in (12a,b) results in the following optimal prices for varying 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:** Inserting the threshold values of the implementation cost and the optimal prices in (14a,b,c) into the condition for the market presence of the copyleft program in (10) shows that it is satisfied for the price and condition in (14c). Q.E.D.

If the implementation cost is high enough, no consumers with a positive surplus from the copyright program receive a 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 choosing 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.[10]

## 3 Monopoly profit maximization

We solve the monopolist's profit-maximizing problem using backward induction. At the development stage, the monopolist anticipates that the program market behaves in a manner described in section 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. Consumer decisions depend on the qualities of programs and implementation cost. The level of implementation cost determines the market structure. In the analysis we concentrate on the scenario in which the copyleft program is present in the market.[11] Profit is the revenue at the output stage, (12), minus the labor cost at the development stage, (6). We assume that the monopolist has no other costs and that there is no discounting. Inserting in (12b) the optimal price in (14c) and the valuations of the programs in (7) leads to the following profit function in terms of the decision variable $a^*$:

$$\pi\left(a^*\right) = R\left(a^*\right) - C\left(a^*\right) = M\,\frac{\mu N\left(2a^{*2} - 1\right)}{8} - N\beta\left(-\ln\left(1 - a^*\right) - a^*\right) \qquad \text{when}$$

$$c \le \frac{\mu N\left(1 - a^{*2}\right)}{4} = \underline{c}\,. \tag{15}$$

[10] Using the terms of the literature on industry entry and exit, we could say that the copyleft program is either blockaded (14a), deterred (14b) or accommodated (14c) in the market.

[11] The profit functions for other scenarios are in the appendix.

**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 the consumer market $M$ is**
- **the more the consumers value quality (the higher $\mu$ is)**
- **the lower the programmers' complementary income is (the lower $\beta$ is)**

**Proof:** Solving the monopolist's profit maximization problem (15) yields the optimal productivity level of the best programmer employed by the firm, $a^{**}$, which in turn determines the optimal number of employed programmers $N_R^{**} = a^{**}N$.[12]

$$a^{**} = 1 - \frac{2\beta}{M\mu} \tag{16}$$

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 the programmers is higher**

- **the larger the consumer market $M$ is**
- **the more the consumers value quality (the higher $\mu$ is)**
- **the higher the programmers' complementary income is (the higher $\beta$ is)**

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

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

**Proposition 5: When the level of the implementation cost is sufficiently low $(c < \underline{c})$, the profit-maximising monopolist**
- **enters the industry if $\beta < 0{,}056M\mu$.**
- **then hires at least 89% of the programmers.**

**Proof:** Inserting the optimal solution of the decision variable, $a^{**}$, into the profit function (15) and setting it equal to zero yields[13] the following condition for the model parameters:

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

The monopolist decides to enter the industry at the development stage if the condition in (18) is satisfied. Applying condition (18) to the optimal solution of the decision

[12] Details of the maximization procedure are in the appendix.
[13] We use a numerical method to solve the equation.

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 $a^*N$ programmers at the development stage. If he enters the industry the profit-maximising 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 interpret it as a scenario in which the level of complementary income in the software application area is large compared to the market. The approach taken in Johnson's (1999) model – 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: If the implementation cost, $c$, is lower than $\underline{c}_{\max} = 0{,}21\mu N$ or less than 13% of the highest valuation of the copyright program, $V_R$, both the copyright and copyleft program can be present in the market. $\underline{c}_{\max}$ is 50% of the highest valuation of the copyleft program, $V_L$.**

**Proof:** The requirements for the simultaneous presence of the copyright and copyleft programs in the market are that the monopolist makes a profit and that the implementation cost is below the threshold value $\underline{c}$ defined in (15). It is decreasing in $a^*$ and reaches its maximum at the lowest possible value of $a^*$. Inserting the condition for industry entry in (18) into the equation for the lower limit of the implementation cost in (15) yields the proposition:

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

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 (15) is decreasing and the maximum valuation of the program in (5a, 7a) is increasing in the decision variable. The maximum ratio obtains in the minimum value of the decision variable.

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

Comparing the equations for the lower limit of the implementation cost in (15) and for the maximum valuation of the copyleft product (5b, 7b) we note that their ratio is a constant, always 50%. If the implementation cost is higher than the value indicated in (20) the market is served either by the monopolist or by the copyleft community. 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 of 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 $(c < \underline{c})$, informing consumers about an unknown substitute copyleft program increases welfare with presumably low costs.**

Let us assume two scenarios in the market. The first involves 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. This assumption is realistic since the copyleft programmers are indifferent as to whether the consumers use their product or not. The second scenario is the one analyzed in section 2.2 and defined by (15). 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 by $M$, the second by $CL$):

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

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

Comparison of (21a) and (21b) shows that the monopolist will always hire less programmers and the quality of the copyright product will be 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 for the monopolist when consumers learn about the substitute copyleft program. The market and cost parameters that result in profit when the monopolist is alone in the market may not fulfill the entry condition in proposition 5. This implies that when the monopolist anticipates a policy of informing consumers of copyleft programs the condition for commitment to the programming investment at the development stage is the one in proposition 5.[14]

[14] The Chinese government is promoting the use of Linux operating system. Its motives seem to coincide with the ones described here. (China joins the Linux Bandwagon, 2000)

**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 copyright is not perfect and that increasing the level of protection is costly. This means that authors, usually firms, with considerable resources can defend their intellectual property. Copyleft programmers or communities, however, do not usually possess such resources. As we noted earlier, copyleft programmers are indifferent as to whether consumers use the copyleft programs or not. 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 the increased opportunity to create derivative works and lower control costs when protection is low. The existence of copyleft communities is an additional variable in 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 promotes stronger 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 presently lack such institutions. The increased economic significance of copyleft software has already resulted in a discussion on the need for such institutions (see Deckmyn 2000a).

# 5 Discussion

The novel contribution of this paper is the simultaneous modeling of the impact of copyleft licensing on both the development environment and the consumer market for programs. The effect of copyleft on 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. Copyleft activity forces him to face 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 cost 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 in pricing his program. The presence of copyleft activity also increases the constraints on market participation for the monopolist. The larger the consumer market is compared to the programming population, the larger the share of the programmer population the monopolist hires and the smaller the copyleft community is. If the market size is small and consumer valuations are low, the monopolist may decide not to enter the market. Only the copyleft program is then available. This result coincides with some real-world phenomena: in certain markets, for example, for some network utility programs, the supply is entirely 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 does not 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 programmers join the copyleft community. There is parallel empirical evidence supporting this finding (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 programmers' incentives. 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 recoup the programming investment in the market imperfect competition of some degree is required. Relaxing the monopoly assumption to oligopoly does not change the qualitative results. In more general terms, the partial equilibrium 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 equilibrium 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 and what consequences it may have? In taxing suppliers of copyright programs 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. Our model does not address network effects for programs. These are an important property of programs and in the case of copyleft a new issue arises: the network for the copyright program is the consumers using it, while the network for the copyleft program is its consumers and the programmers in the copyleft community. This creates an environment in which 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[15] where potential employees signal their ability with the level of education it is assumed that the amount of education acquired does not affect the employer's profit. Copyleft programming, if it is used as a signal of ability, may have a negative impact on the employer's profit. The copyleft community develops a substitute program in a decentralized manner and this affects the market for the program of the employer. He will take this into account and this may change the results of the signaling analysis.

## 6 Acknowledgements

[15] see for example chapter 11 of Hirschleifer and Riley (1992).

I thank Vesa Kanniainen for advice and encouragement. The valuable comments of Vincenzo Denicoló and Tuomas Takalo at the FPPE March 2001 workshop are appreciated. The financial support of the Academy of Finland was vital for this work.

## 7 Appendix

**Profit functions of scenarios in (14a) and (14b)**

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

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

when $\bar{c} \geq c > \underline{c}$ (A.2)

**The monopolist's profit maximization (15)**

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

$$\frac{d\pi}{da^*} = \tfrac{1}{2}\mu N M a^* - \frac{N\beta a^*}{1-a^*} = 0\,. \quad \text{(A.3)}$$

This yields two solutions for the optimal $a^*$:

$$a_1^{**} = 0 \quad \text{(A.4)}$$

$$a_2^{**} = 1 - \frac{2\beta}{M\mu}\,. \quad \text{(A.5)}$$

The second-order condition for the maximum is

$$\frac{d^2\pi}{da^{*2}} = \tfrac{1}{2}\mu N M - N\beta\left(\frac{1}{\left(\left(1-a^*\right)^2\right.}\right) < 0\,. \quad \text{(A.6)}$$

From proposition 1 we know that the solution has to satisfy $a^{**} > \left(\sqrt{2}\right)^{-1}$. Inserting $a_2^{**} = 1 - \frac{2\beta}{M\mu}$ in (A.6) yields $\frac{2\beta}{\mu M} < 1$. This holds for possible values of $a^{**}$. The only profit-maximizing solution is $a_2^{**}$.

**Policy implication 1: Informing consumers**

The first-order condition to maximize profit in the absence of a copyleft product (A.1) with zero of implementation cost is:

$$\frac{d\pi_M}{da^*} = \frac{1}{4}\mu N M a^* - \frac{N\beta a^*}{1-a^*} = 0. \qquad \text{(A.7)}$$

This yields the optimum level of the best programmer employed,

$$a_M^{**} = 1 - \frac{4\beta}{M\mu}. \qquad \text{(A.8)}$$

The second-order condition and profit level analysis is similar to (A.6).

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

$$W_M = \frac{3}{16} NM\mu a_M^{**\,2} - N\beta\left(-\ln\left(1-a_M^{**}\right) - a_M^{**}\right). \qquad \text{(A.9)}$$

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

$$W_{CL} = NM\mu\left(\frac{1}{8}a_{CL}^{**\,2} + \frac{1}{16}\right) - N\beta\left(-\ln\left(1-a_{CL}^{**}\right) - a_{CL}^{**}\right). \qquad \text{(A.10)}$$

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

$$W_M > W_{CL} \Rightarrow \frac{1}{4}Z^2 - (1 - \ln 2)Z > 0. \qquad \text{(A.11)}$$

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

$$\pi_M > \pi_{CL} \Rightarrow Z\left(\ln 2 - \frac{1}{2}Z\right) > 0. \qquad \text{(A.12)}$$

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 $0 < Z < \frac{\sqrt{2}-1}{\sqrt{2}} \approx 0{,}297$. Welfare is thus 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.

# 8 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. and L. Raskind, 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).

China joins the bandwagon, 2000, Far Eastern Economic Review, February 24, p.8

Dalle J-M. and N. Juillien, 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. and P. David , 1987, Information disclosure and the economics of science and technology, CEPR Discussion Paper 73, also in: Feiwel D ed., Arrow and the ascent of modern economic theory (New York University Press, New York).

Dasgupta P. and P. David, 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. eds., Global Dimensions of Intellectual Property rights in Science and Technology ( National Academy Press, Washington, USA).

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., I. Jewitt and J. Tirole, 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).

Hirschleifer J. and J. Riley, 1992, The analytics of uncertainty and information (Cambridge University Press, Cambridge)

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

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

Johnson J., 1999, Essays in microeconomic theory, Dissertation, Massachusetts Institute of Technology.

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

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

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

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

Novos I. and M. Waldman, 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. eds., Global Dimensions of Intellectual Property rights in Science and Technology (National Academy Press, Washington, USA).

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

Shy O. and J.-F. Thisse, 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.

Subramanian P., 2000, Three essays in microeconomics: uncertain innovation in the presence of network externalities, copyleft: an R&D game with network externalities, imperfect recall in a model of search, Dissertation, Princeton University.

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).