CERGE
Center for Economic Research and Graduate Education
Charles University Prague

Essays on Pricing, Product Quality, and Intellectual Property Rights Protection in the Software Market

Jiří Střelický

Dissertation

Prague, March 2011

Jiří Střelický

Essays on Pricing, Product Quality, and Intellectual Property Rights Protection in the Software Market

Dissertation

Prague, March 2011

DISSERTATION COMMITTEE:

AVNER SHAKED, University of Bonn and CERGE-EI (Chair)
KREŠIMIR ŽIGIĆ, CERGE-EI (Local Chair)
ANDREAS ORTMANN, The University of New South Wales and CERGE-EI

REFEREES:

MILAN HORNIAČEK, Comenius University in Bratislava
MARTIN PEITZ, University of Mannheim

Contents

Abstract

In this thesis, I explore the particular issues of pricing, product quality selection, and intellectual property rights (IPR) protection in the software market. In the first part of the thesis, I study price discrimination in a monopolistic software market. The monopolist charges different prices for the upgrade version and for the full version. Consumers are heterogeneous in taste for software that is infinitely durable and there is no resale. I show that price discrimination leads to a higher software quality but raises both absolute price and price per quality. This price discrimination decreases the total number of consumers compared to no discrimination. Finally, such discrimination decreases consumers' surplus but increases the developer's profit and social welfare that attains the social optimum in the limit. In the second part of the thesis, I focus on the interaction between a regulator's IPR protection policy against software piracy on the one side and the forms of IPR protection that software producers may themselves undertake to protect their IPR on the other side. Two developers, each offering a variety of different quality, compete for heterogeneous users who choose among purchasing a legal version, using an illegal copy, and not using a product at all. Using an illegal version violates IPR and is thus punishable when disclosed. If a developer considers the level of piracy as high, he can introduce a form of private protection for his product. I examine the above issues within the framework where the quality of each developer's product is exogenously given, and the developers compete in prices.

Abstrakt

V disertaci zkoumám specifické problémy cenotvorby, výběr kvality výrobku a ochranu autorských práv (IPR) na softwarovém trhu. V první části disertace studuji cenovou diskriminaci u softwarového monopolu. Monopol účtuje rozdílné ceny za upgrade verzi a za plnou verzi. Uživatelé jsou různorodí v užitku ze softwaru, který je nekonečně trvanlivý a jeho další prodej není možný. Ukazuji, že cenová diskriminace sice vede k vyšší kvalitě softwaru, ale zároveň zvedne jak absolutní cenu, tak cenu za jednotku kvality, a že tato cenová diskriminace sníží celkový počet uživatelů. Nakonec, tento druh cenové diskriminace snižuje spotřebitelské přebytky, zvyšuje zisk monopolu, avšak v limitě celkový společenský přebytek dosahuje sociálního optima. V druhé části disertace analyzuji ekonomické dopady interakce mezi ochranou duševních práv proti softwarovému pirátství regulatorní politikou na straně jedné a formou ochrany softwaru, kterou implementují k ochraně svých děl výrobci, na straně druhé. Dvě softwarové firmy, každá nabízející software odlišné kvality, soutěží o různorodé uživatele, kteří se rozhodují mezi nákupem legální verze, používáním ilegální kopie a celkovým nepoužívání softwaru. Používání ilegální verze porušuje autorská práva a vede k postihům v případě odhalení. Jestliže softwarová firma považuje úroveň pirátství za vysokou, může zavést vlastní specifickou formu ochrany softwaru. Daný problém analyzujme v konceptu, kdy kvalita softwaru obou softwarových firem je daná exogenně a firmy soutěží v cenách.

Acknowledgements

When I made a decision to do Ph.D. studies at CERGE-EI, I could have hardly imagined the effort that was necessary to successfully complete it. Especially, when I moved to a business career six years ago, I realized that accomplishing such a task became more than challenging. To do so, required enormous effort and patience from all people around me. Thus, to many of them I am more than grateful for standing at the end of my long Ph.D. journey.

My first and the deepest thanks go to my chair, Krešimir Žigić who provided me with strong guidance and friendly support during my whole studies. He provided me not only with excellent expertise and guidance, but he was able to challenge me all the time and raised my curiosity for the topic in countless discussions. He showed long patience for reviewing dozens of versions of the dissertation and contributed with countless numbers of ideas. But I am even more than grateful to him for continuously motivating me to complete the dissertation and not giving it up.

I strongly appreciate the huge support of Avner Shaked, who was my chair in the beginning of the dissertation and who set the overall direction that was shaped during much discussion that partially took place in Bonn during my one-year Marie Currie Fellowship. He helped me to set the direction of my studies and mainly encouraged my interest for Microeconomics and Game Theory.

My visit at UC Berkeley, where I learned a lot about information economics, had a strong impact on forming my research interests, so my special thanks belong to Yale Braunstein and Hal Varian.

At the same time, I would like to thank all professors who contributed to my accomplishment during my Ph.D. studies. I would like to especially mention professors like Jan Hanousek, Andrew Austin, Jan Švejnar, Štěpán Jurajda, Michal Kejak, Lubomír Lízal, Evžen Kočenda, Daniel Münich, and others. These thanks belong to them not only for providing me with very useful knowledge but also for creating an exceptional CERGE spirit that I felt all around.

During my studies, I met a lot of exceptional fellows at CERGE and during studies, who shaped me and many of them became my friends. At this point, I should mention Dr. Michael Kúnin, whose exceptional technical skills proved inevitable in shaping the very last version of my thesis.

I would like also to thank my opponents Milan Horniaček and Martin Peitz for their valuable comments as well as Paul Belleflamme for his insightful comments on the final version of my thesis.

I leave the most important thanks for the very end. It goes to my beloved and incredibly patient wife, Katka, who provided me with daily support and motivation for all those years. She showed unbelievable tolerance during all these years in which I devoted a countless number of weekends to my dissertation instead of spending time with family. Without her, I would never manage to combine work with dissertation studies and never with family life. Thanks a lot, dear Katka!

Introduction

The dissertation is composed from four essays focused on the software market. The thesis can conceptually be divided into two main parts each of which uses a related set-up and methodology. In the first part, I focus on a price discriminating software monopolist using upgrade discounts that compete with older versions, while in the second part, I analyze the interaction between the public and private intellectual property rights protection (IPR) in a software duopoly set-up.

This first part of the dissertation comprises two essays, one in which software users have imperfect foresight and the second, a more complex one, in which these users have perfect foresight. I assume that software is infinitely durable and any kind of resale is forbidden and that consumers are infinitely lived and heterogeneous in sensitivity to software quality. Given the type of user foresight, I study a particular type of price discrimination in which the monopolist charges a different price for the so-called “upgrade version” and for the “full version” of software. I show that price discrimination leads to a higher software quality on the one side, but raises both absolute price and price per quality unit on the other. In contrast to standard price discrimination, I show that this price discrimination does not increase sales. Moreover, this specific discrimination not only decreases the number of consumers who would qualify for the “full version” without discrimination but even the number of consumers who would qualify for the “upgrade version.” Finally, I demonstrate that such discrimination decreases the consumers’ surplus and social welfare, which is not offset by a higher developer’s profit.

In the second main part, I study the economic impacts of the interplay between a regulator’s IPR protection policy against software piracy on the one side and the forms of IPR protection that software producers may themselves undertake to protect their intellectual property on the other. Two developers, each offering a variety of different quality, compete for heterogeneous users who choose among purchasing a legal version, using an illegal copy and not using the product at all. Using an illegal version violates IPR and is thus punishable when disclosed. If a developer considers the level of piracy as high, he can introduce a particular form of IPR protection. The quality of each developer’s product is exogenously given and the developers compete in prices. The second part of the dissertation is also composed of two essays. In the first essay, I study the positive aspects of competition between developers when private IPR protection comes in the form of restricting support and other services to illegal users. In the second essay, I analyze the same issues but now software protection appears in the form of physical protection rather than in the form of restricting services. I examine the above issues within the framework of Bertrand and Stackelberg competition while a monopoly set-up serves as a point of reference in both essays.

Software Upgrades under a Monopoly[1]

Abstract

We analyze a particular case of price discrimination in a software market dominated by a monopoly that charges different prices for what is known as "upgrade version" and for the "full version" of software. We assume that the software is infinitely durable, and any kind of resale is forbidden. We compare software prices, quality, consumer surplus, and social welfare in two set-ups, a market where the developer sets a single price for both versions *versus* a market where the developer discriminates. Consumers are infinitely lived and heterogenous in sensitivity to software quality. We show that price discrimination leads to a higher software quality on the one side, but raises both absolute price and price per quality unit on the other. Contrary to standard price discrimination, we show that this price discrimination does not increase sales. This specific discrimination not only decreases the number of consumers who would qualify for the "full version" without discrimination but even the number of consumers who would qualify for the "upgrade version." Finally, we find that such discrimination decreases consumers' surplus yet increases the developer's profit and social welfare with the social optimum attained in the limit.

[1] All errors remaining in this text are the responsibility of the author.

1 Introduction

The software market is one of the largest and fastest growing markets where, for instance, the largest developer (Microsoft) earns a revenue of more than \$60 billion per year[2]. Other several hundred thousand developers operate on the market, so it appears that tough competition should be the norm. By a closer look at every particular sub-market of the software market (not only the operating systems), we could often identify a dominant developer with such an established position (e.g., Adobe, Symantec, Pinnacle, and of course Microsoft) that the whole sub-market could be treated as almost a monopoly. Such a monopolistic market structure may serve as an explanation for some high software prices (Katz and Shapiro, 1998). Moreover, there is a high range of possible price discrimination schemes that such a monopolist may undertake. Since software is "lent" to users, where the identity of both parties is often known, the developer could easily set marginal prices to different groups of users without incentive problems. In such a case, a user prefers to reveal the personal information that qualifies him for a lower price. In the real world, we could observe dozens of prices for identical software: a price for the standard retail user (often called the full version), the OEM version, the upgrade, and student or multi-license versions. On top of that, there are different prices for the university, the army, the public sector, or large corporations, and naturally, the goal is to set prices close to the reservation prices of the respective groups.

Besides the pricing policy, an even more important issue is whether and how to improve the *quality* of software that such a monopolist firm generates. By the vague term "quality," we understand not only software functionality, but foremost software stability, speed, compatibility with hardware, and nowadays quite often security. From this point of view, software evolution may be viewed as socially sub-optimal if a developer rashly introduces poorly tuned software[3] that leads to welfare losses caused by additional consumer costs[4].

Thus, the main focus of our paper is to analyze the pricing policy of a software monopolist and its impact on software quality evolution. More specifically, a monopolist can either set a single price for its products, or it can stick to price discrimination based on an upgrade scheme. As we shall see, a different approach to a pricing strategy leads, in turn, to a different evolution of software quality.

Software developers often motivate users to switch more frequently to a new version of the product by

[2]The Microsoft report for the fiscal year 2008.

[3]For example, more than 20 000 mistakes were known for Microsoft Windows 2000 at the time of the release, and a Service Pack for Microsoft Windows XP was in preparation even before the official XP release.

[4]Though most users may have in mind a frozen or "blue screen of death" window with the consequent loss of the latest document, software bugs can, however, cause more severe losses. United States losses due to software problems are estimated at USD 59.5 billion, which is 0.6% of GDP. According to the Stanford Research Institute, most of these problems are due to a lack of testing. For example, a software bug caused the death of 228 people in an airplane crash of Korean Airlines, or a bug in ABS software caused more than 38 000 trucks to be withdrawn from the market. If such a situation occurs, the regulatory authorities tend to force developers to improve quality, e.g., by undertaking broader testing and tuning before the official release.

Note that software developers usually bear no, or very limited, responsibility for user losses associated with software usage according to most *End-User License Agreements* (EULAs) that must be accepted by users before installing the software. In the EULA, the developer determines the rights and commitments between himself and an end user. The industry standard is that the developer assigns to himself most of the rights without any commitment. Most types of EULAs are often criticized and considered as a sign of monopoly behavior. According to the SCC (Sustainable Computing Consortium), it is necessary to change it in the future and to force software companies to bear partial responsibility for their products. Under the current market situation, full developer responsibility would not be acceptable since official authorities are worried that such user protection, which is common in other industries, would undesirably suppress software evolution and unacceptably increase software prices.

The complex problem of developer responsibility and the legal framework are analyzed, for example, in Werden (2001).

lowering the price in the case where users own the previous version. Thus, a user faces two options: to switch to every new version to get the upgrade price, or to switch only from time to time for the full price[5]. In the real world, upgrade prices are predominantly used for business software. The ratio between upgrade and full price varies according to the developer's position on the market, the necessity to have updated software, and innovation frequency[6].

Because the average time between releases of new versions is short[7], users have to familiarize themselves with new software almost permanently. When an old version becomes sufficiently outdated, the users' willingness to switch to new software grows, and software obsoleteness becomes crucial to both users' and developers' decision process.

In order to model the above phenomena, we put forward a dynamic model environment in which there is, on the one side, a software monopolist who introduces a new version (quality) of software all the time (or every period), and on the other side, there are consumers (or users) who are heterogeneous in their sensitivity to software quality. That is, they buy software with a different frequency depending on their currently possessed software version. We assume that there is no second-hand market for software and that the developer commits to its prices at the outset for the whole horizon. That is, he sets either a single price (in the absence of price discrimination) or a pair of prices whereby a lower price is charged to the consumers who buy new software every period. Since software is perfectly durable and developers keep introducing new versions all the time, the model has to deal with a growing number of software versions. We first assume that users have imperfect foresight in a sense that they are unable to predict how often they would switch to a new software version in the future. Later on, we extend our model to users with perfect foresight without changing the developer's problem. We shall show that the perfect foresight of users, if known to developers, increases the developers' opportunity to force users to switch more often.

The key model simplification is the absence of consumers' outside option. In reality, every user has outside options such as using similar software from non-dominant developers (including open-source software) or even using an illegal copy of the software. However, even though software prices have a significant impact on the users' decision whether to undertake piracy or not, what is more important for the extent of software piracy is the role of government[8] and the expected piracy punishment. Thus, our model well suits markets such as business software in developed countries, where software upgrading is standard, and a high expected punishment limits piracy. Banerjee (2003) analyzed the problem of switching from legal software to piracy, and its social welfare impact with respect to the government's incentives to tolerate piracy. We also assume away the network effects since their significance under monopoly is often suppressed to the role of an entry barrier. The concept of network effects under a monopoly was studied by Ellison and Fudenberg (2000) and Fudenberg and Tirole (2000).

As for the related literature, our approach shares some of the features of behavior-based pricing with

[5]Nevertheless, sometimes users need not be motivated to switch, e.g., game players are often willing to pay more for a new version when they own the previous version than when they have no experience with that particular game.

[6]E.g., the full price for MS Windows XP Home is \$199, and the upgrade price is only \$99. For MS Office XP Pro, these prices are \$499 and \$299 respectively. Very often the upgrade price is approximately 50%-70% of the full price.

[7]The major releases are usually between 12 and 24 months.

[8]Usually governments only provide a legal environment, whereas actual anti-piracy force is exerted by independent or public organizations.

multiple products (see, for instance, Fudenberg and Tirole, 1998, and Ellison and Fudenberg, 2000; see also Villas–Boas and Fudenberg, 2007 and Belleflamme and Peitz, 2010 for a survey of this literature). The key feature of this approach is that the monopolist may be able to use his information about the consumers' purchasing history to offer different prices and/or products to consumers with such different histories. The two most common information structures consistent with behavior-based pricing are the "identified consumers" and the "semi-anonymous consumers."While the first category is self-explanatory, semi-anonymous consumers are consumers who can prove that they purchased software in the last period if they wish to do so, but they can also pretend not to have bought it if this is in their interest. So the underlying assumption in our model in which consumers qualify for an upgrade if they buy the last version of the software can be thought to fit the "semi-anonymous consumers" assumption. Moreover, much like Fudenberg and Tirole (1998), and Ellison and Fudenberg (2000), we also study the provision of "upgrades" by a monopolist in a setting of vertical differentiation, where there is a consensus among customers that newer versions of the software are better than older ones. Our analysis, however, differs from the above literature in some important aspects. First, we focus on the software market, rather than on a broader class of durable goods. This, among others, implies that the existence of a second-hand market is not appropriate in our setting. We, unlike Fudenberg and Tirole (1998), and Ellison and Fudenberg (2000), use an infinite horizon model to study successive product generations in the software market, and in this sense, we focus more on the incentives to innovate and on the social optimal level of innovation rather than on the very pricing decisions. As for monopoly pricing, we do not deal explicitly with the commitment issue, and are mostly interested in how the introduction of price discrimination based on upgrades and the lock-in of consumers affect the software quality evolution. Moreover, we do it in the two set-ups of myopic and forward-looking consumers.

Another relevant paper that shares some similarity with our analysis is that of Fishman and Rob (2000), who consider a durable-good monopolist that periodically introduces new product models with each new model being an improvement over the preceding one. In this light, our focus of software evolution can be viewed as a particular case of their analysis of R&D in durable-good industries under monopoly. They, however, assume that consumers are homogeneous, and, with their focus on social optimum, their primary finding is that if the monopoly developer can neither shorten the lifetime of its products nor discriminate in prices, then the monopoly developer innovates less frequently and invests less than at the socially optimal level. Then they show that if either planned obsolescence or price discrimination based on the age of the product held by the consumer is available, then the developer can both increase his or her profit and implement the social optimum.[9].

A peculiar feature of Fishman and Rob (2000) is that the homogeneity of consumers leads to every new

[9] There are many papers that build upon or deal with other aspects of Fishman and Rob (2000). For instance, Atil et al. (2008) show that a competitive environment or after-market leads to a lower investment in R&D. Mehra and Seidmann (2008) focus on the life-cycle management of software products and show how the optimal upgrade changes throughout the life cycle based on whether there are heterogeneous customers, externalities, or product incompatibilities. Anton and Biglaiser (2009) study upgrades and their quality in a dynamic model with homogenous users, focusing on the pricing and taking innovation as exogenous. Nahm (2004) focuses on the interaction between inter-temporal pricing and R&D decisions based on the pricing regimes (net sales vs. buyback). Other papers investigating obsolescence include Echevarría (2005), Wang and Hui (2005), and Inderst (2008).

product being purchased by all consumers. However, real durable-good markets, and the software market in particular, are characterized by the simultaneous presence of both the newest and the previous products in the consumers' ownership, i.e., while there are consumers who use the newest product, there also are consumers who keep using previous ones. As each new product represents a quality improvement over the preceding one, vertical differentiation is inherent in such markets; so, as already stated above, we allow for consumer heterogeneity with respect to quality sensitivity that captures this property. In addition, while Fishman and Rob (2000) show that a price discriminating monopolist is able to fully implement the social optimum, they assume that the price discrimination is of the first-degree kind in the sense that the developer is able to freely change the price according to the age of the product currently owned by the consumer. In our approach, the price discount is only offered to those who own the immediately preceding version, which is a more realistic setting of behavior based on third-degree price discrimination. We show that if the developer can set different prices for the upgrade and the full version, then the social welfare resulting from the monopolist's optimal action in the limit approaches the social optimal level irrespective of whether the users have perfect or imperfect foresight. We also show that a popular notion according to which upgrade prices help to spread software and lead to lower price levels does not hold if different prices for the upgrade and the full version are used.

The paper is organized as follows. In the second chapter, we set up the model that serves as the general framework for our analysis. The first essay (the third, fourth, and fifth sections) deals with the choice of a software upgrade versus the full-price version in an environment where users have imperfect foresight. More specifically, in the third section, we analyze the developer's behavior without price discrimination, and in the fourth section, we focus on price discrimination based on a lower upgrade price. Finally, in the fifth section, we analyze the impact of price discrimination on prices, software quality evolution, and social welfare. Next three sections constitute the second essay, where users are now assumed to have perfect foresight. In sections six to eight, we replicate our analysis from sections three to five, but now with the assumption of perfect foresight. In addition, in section seven, we study the monopolist's tendency to initiate lock-in behavior. In the ninth section, we discuss the results obtained across all sections and point out the differences between an imperfect versus perfect foresight set-up. Section ten concludes.

2 The model

2.1 Basics

There is a monopoly software developer who releases a new version every period. The timeline starts in period 1, when the first version of the software is released, and continues indefinitely.

When a user buys software, he can use the version purchased forever without any additional fee. All versions of software are infinitely durable without depreciation. However, the developer sells only the latest version of the software every period, whereas older versions are not sold anymore. Besides that, the developer outlaws any resale by a license agreement with the user. We assume that users have no outside option like piracy or open-source software, so all users are fully dependent on the actual developer's offer.

2.1.1 Price discrimination

While only the current software version can be sold in every period, the developer is able to price discriminate by setting different prices (for the same version) along with the eligibility rules for a consumer to qualify for a given price. We assume that in every period it is impossible that there is a user who is completely ineligible to any price offered in the period. In general, the developer can discriminate based on the history of the user's previous actions, and the rules may change between periods. In this paper, we make the following two assumptions.

First, the only thing the developer can observe about a specific user in any period is the software version (if any) possessed by the user. Thus, the only way how the developer can price discriminate is based on the age (in periods) of the software version in the user's hands.

Second, we assume that the developer's pricing policy is stable over time. Namely, the prices offered in period t and the corresponding eligibility rules do not depend on t. We concentrate on the following two cases:

- The single price, when every version is offered at the same price p to all users in every period.
- The "new user" versus an "upgrade", when in every period t anyone can buy at the "new user" price p_2, but the users who purchased the software version immediately preceding the current one, i.e., the version of period $t-1$, and only those users, are entitled[10] to buy at the "upgrade" price p_1, with p_1 and p_2 constant across periods. Here those who own older versions, but not the immediately preceding one, are treated as "new users" just as those who own no version at all.

Remark 1 *One of the key assumptions we make is that the monopolist is able to precommit to its price(s). That is, the monopolist sets price(s) in the first period and keeps them unchanged thereafter. In other words, the famous Coase conjecture that a price setting monopolist in an inter-temporal set-up may not commit to its future prices is not an issue in our set-up. The reason for this may, for instance, be that (besides permanent upgrades), our monopolist has a reputation for sticking to his or her pre-announced price or has fixed production capacity that servers as a commitment device against future price decreases, etc.*[11] *(See more on the durable-good monopoly with commitment in Belleflamme and Peitz, 2010, chapter 10.2; see also Fudenberg and Villas-Boa on behavioral based price discrimination and Stole's survey on price discrimination, 2007.)*

2.1.2 Quality and R&D cost

Denote software quality in period t as Q_t. For simplicity, we treat quality as a one-dimensional variable that can be viewed as a weighted linear combination of all characteristics (performance, design, stability,

[10] Generally, it is possible to introduce more differentiated upgrade prices, e.g., for users who possess the version from the second latest period, the third latest period, and so on. Doing that would not substantially change our results in any way, but the structure of the model, as well as overall results, would become less transparent.

[11] There is, however, an alternative micro-foundation approach which explains the price rigidity by means of rational inattention (see, for instance, Sims, 2005). In the realistic case when this inattention appears on the consumers' side, a frequent price changes may require consumers to pay lots of attention to the price. That, in turn, may be irritating for the consumers so in the end, they could decide to consume less. As a result, it would be optimal for the seller to choose his pricing strategy in advance and commits to it (see Matějka, 2010). Moreover, this may also explain the empirical observations that real-life developers are observed to keep roughly constant prices over longer period of time.

and security, amid others) In fact, exact quality cannot be measured, so we measure the quality indirectly by the willingness to pay for a product.

As resale is outlawed, there is no outside option for the users, and any version is infinitely durable, the only trigger for new demand is an improvement in software quality that persuades users to replace their older version of software[12]. In our model, we introduce the developer's cost of software quality improvement ΔQ, which is in fact the cost of R&D. All other developer costs are normalized to zero. We assume that the cost of software quality improvement is increasing in quality $\frac{\partial C(\Delta Q)}{\partial \Delta Q} > 0$, and we assume that costs are convex $\frac{\partial^2 C(\Delta Q)}{\partial \Delta Q^2} > 0$. This condition means that gradual quality improvement (e.g., to improve quality by 1 every period for the next three periods) is cheaper than a significant quality jump (e.g., to improve quality by 3 within one period). Both conditions are satisfied for quadratic functions; so, we use

$$C(\Delta Q) = \bar{B} \cdot (\Delta Q)^2, \tag{1}$$

where parameter $\bar{B} > 0$ reflects R&D effectiveness.

Improving software is a longtime process based on cumulative activities like learning by doing. Thus, the developer cannot increase software quality simply by hiring a large number of new programmers in one period. To achieve significant progress in quality improvement, the developer must decide about the targeted future quality several periods in advance. The developer does so by deciding on the maximum evolution quality improvement in period t, K_t, at least n periods before period t. Then in period t, software quality improvement is limited to $\Delta Q_t = Q_t - Q_{t-1} \leq K_t$ and cannot be exceeded[13]. In reality, n may be several years. In our model, we assume $n \to \infty$; hence, the developer decides about the evolution capacity once and for all.

In line with the assumption of stable prices and eligibility rules, we assume that the evolution capacity is stable over time, $K_t = K$ for all t. The capacity is assumed to be fully utilized in every period. Thus, a monopolist releases new software versions with constant quality improvement at the maximum capacity level, that is, $Q = \Delta Q_t = Q_t - Q_{t-1} = K$ each period t. If the initial quality in period t is Q_t, then the quality in period $t + l$ is $Q_{t+l} = Q_t + l \cdot Q$.

As all developer choice variables, namely the prices and the quality jump, are stable, we assume that they are chosen in the very beginning. Note that in our framework, quality adjustment choice and capacity choice are basically equivalent since in selecting capacity K in the beginning, the developer intends to fully use this capacity to achieve quality improvement $Q = K$ in every period.

2.2 Users

Assume an infinite number of heterogenous users on the market. The users differ in their sensitivity θ to product quality, where θ is uniformly distributed and is normalized[14] to interval $\langle 0, 1\rangle$. Users with θ close

[12]The model can be generalized by introducing the probability of failure. Nevertheless, we then get total demand as a linear combination of demand stemming from technological improvement and demand coming from physical depreciation. The total result will be influenced by the weights put on each source without changing the core of the result.

[13]This is a simplification helping us to focus the paper on price discrimination.

[14]We could also explicitly introduce a "market size" parameter by assuming that the density of θ is a positive constant rather than exactly 1, but this is effectively captured by parameter $\bar{B}$ in the developer's cost function as demonstrated below.

to 1 primarily prefer quality, while users with θ close to 0 are very price sensitive. Users' discount factor between periods is $\beta \in [0, 1)$, which is assumed the same[15] for all users.

At the beginning of every period, a user may either own no software version at all or own a software version from some of the previous periods. Then the user faces the following question. If no version of software is owned, then the question is: "Buy the software now or wait for some next period when the consumer value of the new software, because of a higher product quality, will increase." If some previous software version is owned, then the question is: "Buy the new version now or wait until the consumer value is increased and meanwhile continue using the old version." We can consider owning no software as owning the "zero version," whose quality is $Q_0 = 0$.

Consider a user of type θ who owns a version of quality Q_t in period t. If this user decides not to buy the newest version, then this user's utility flow in period t is θQ_t. (This implies that users have zero utility flow before they buy software for the first time.) If this user buys the newest version, whose quality is Q_t, at price p_t, then this user's utility flow in period t is $\theta Q_t - p_t$. Note that p_t can differ among users due to the developer's eligibility rules. However, if a user happens to be eligible for more than one price in a given period, then the lowest of these prices will be used as p_t.

In the paper, we separately analyze two cases of user behavior:

- Users with *imperfect foresight* or myopic users, who make a decision whether or not to buy software based only on the comparison of the utility of having (or not having) the actual product at the current period, while ignoring their eventual switching to new software. Those users assume in calculating their utility that they would keep the actual version forever.

- Users with *perfect foresight*, who know exactly in which period they would switch to a new version; those users calculate their utility flow precisely in advance.

2.2.1 The decision of users with imperfect foresight

An imperfect foresight user of type θ who owns no software until period t, switches to software of quality Q_t at price p_t in period t, and who never switches again, has the following infinite utility flow from period t onwards:

$$U_t = \theta(Q_t + \beta Q_t + \beta^2 Q_t + \dots) - p_t = \theta \frac{1}{1-\beta} Q_t - p_t. \tag{2}$$

For simplicity, denote $q_t = \frac{1}{1-\beta} Q_t$ and the utility flow of a user θ from buying software of quality q_t for price p_t in period t as

$$U^\theta(q_t, p_t) = \theta q_t - p_t. \tag{3}$$

It is obvious that a new user θ strictly buys software at period t if and only if $U^\theta(q_t, p_t) > 0$. The marginal user who is indifferent[16] between buying and waiting for some next period has sensitivity parameter $\theta = \frac{p_t}{q_t}$.

The previous decision process, however, does not cover a user who already possesses some version of software. Consider now an imperfect foresight user of type θ who possesses the software of quality Q_l from

[15] In other words, we assume that all heterogeneity among users is captured by θ.

[16] Indifferent consumers can either buy or not, and it is useless to restrict them, for example, to buy because their measure is 0 even if their number goes to infinity as t tends to infinity.

period l. Such a user has already ensured utility flow $U^{\theta} = \theta Q_l$ at every period $t \geq l$. He decides to switch to new software that would bring him $U^{\theta} = \theta Q_t$ if and only if the difference in quality offsets the disutility from price p_t. It implies that he buys in period t if and only if

$$U^{\theta}(q_t, p_t) = \theta q_t - p_t \geq \theta q_l = U^{\theta}(q_l). \tag{4}$$

If the quality change is not sufficient to compensate for the disutility from the price (if $\theta(q_t - q_l) < p_t$), then the user does not buy now, uses his older version from period l, and waits until the next period $t+1$ when he enters into the decision process again and when he compares the utility flow θq_l with $\theta q_{t+1} - p_{t+1}$.

As can be seen from the user decision process described, the users do not foresee future software quality levels Q_t and prices p_t so that each period they look at the actual quality offered by the developer and decide based on the current price. In this case, the users cannot foresee the exact time of next switching to a new product. These users simplify their decision process by comparing the utility flow from using software forever ignoring their own future decision process.

2.2.2 The decision of users with perfect foresight

Perfect foresight means that every user can foresee future quality levels Q_t and prices p_t in every period t. Then a perfect foresight user faces the following problem in every period. Let U_t^l be the user's discounted utility flow in period t given that the version from period $l \leq t$ with quality Q_l is used during this period, and let U_t be a shortcut for U_t^t. Let p_t^l be the minimal price the user is eligible for in period t given that the version from period l is owned. If the currently owned version $l < t$ is used for n periods on and then a new version is bought in period $t+n$, then

$$U_t^l = \theta Q_l \left(1 + \beta + \cdots + \beta^{n-1}\right) + \beta^n U_{t+n}, \tag{5}$$

whereas if the new version t is purchased at price p_t^l and then used for n periods, when another new version is bought, then

$$U_t = -p_t^l + \theta Q_t \left(1 + \beta + \cdots + \beta^{n-1}\right) + \beta^n U_{t+n}. \tag{6}$$

In every period, the user chooses (i) between buying and not buying and (ii) for how long to keep the version, given the anticipated quality and price development, which can be generally written as a dynamic programming problem. Given our assumptions about the stability of prices, pricing rules, and quality improvement, U_t^l solely depends on the difference $t - l$ in the sense $U_{t+m}^{l+m} = U_t^l$ for any integer $m \geq 0$. An interpretation of this is that the consumer has a guaranteed utility level of θQ_l per period, which amounts to the discounted flow of $\theta \frac{Q_l}{1-\beta} = \theta q_l$, and decides on the basis of added utility so that equations (5) and (6) can be re-written as

$$U_t^l - \theta q_l = \beta^n \left(U_{t+n} - \theta q_l\right) \tag{7}$$

if the consumer does not switch in period t, and

$$\begin{aligned} U_t - \theta q_l &= -p_t^l + \theta \left(Q_t - Q_l\right) \left(1 + \beta + \cdots + \beta^{n-1}\right) + \beta^n \left(U_{t+n} - \theta q_l\right) && (8) \\ &= -p_t^l + \theta \left(q_t - q_l\right) + \beta^n \left(U_{t+n} - \theta q_t\right) && (9) \end{aligned}$$

if the consumer switches in period t.

As a particularly important example, consider a user with a high sensitivity θ, who buys every period and he knows that. Given the stability of prices ($p_t = p$ or $p_t = p_1$) and quality improvements ($Q_t = Q_{t-1} + Q$, so that $q_t = q_{t-1} + q$), and the last equation takes the form

$$U_t - \theta q_{t-1} = -p + \theta q + \beta \left(U_{t+1} - \theta q_t \right) \tag{10}$$

for every t. Infinite iteration of this equation yields (for $\beta < 1$)

$$U_t - \theta q_{t-1} = (-p + \theta q) \left(1 + \beta + \beta^2 + \cdots \right) = \frac{1}{1-\beta} \left(\theta q - p \right), \tag{11}$$

so that a necessary condition for a user with perfect foresight to switch every period is $\theta q - p \geq 0$.

Notation 1 *We denote the user who buys every period as a high-end user and a user who buys less frequently than every period as a low-end user.*

2.2.3 Regularity of upgrades

In general, if prices, quality improvement, and eligibility rules vary over time, so may vary the users' decision as to buy the new version or keep the currently held one for another period. However, the ensuing stability assumptions lead to the following result, which substantially simplifies further analysis.

Proposition 1 *Let prices, quality improvement levels, and price eligibility rules be constant over time, and let the developer's pricing policy be either "upgrade" price p_1 versus "new user" price p_2 or single-price $p_1 = p_2 = p$. Then it is optimal for a user with either imperfect or perfect foresight to switch regularly, i.e., there exist a period $T \geq 0$ and a natural number $n = n(\beta, \theta, p_1, p_2, Q)$ such that the user switches in periods T, $T + n$, $T + 2n$, ..., and in no other period. In addition, n is non-increasing in θ.*

Proof. The existence follows from the facts that the initial utility is zero, the per-period quality improvement is positive, and the prices are stable. Then $\exists T$ such that $Q_0 + T \cdot Q > \max\{p_1, p_2\}$.

Regularity under the single price stems from the fact that the user faces exactly the same problem in every period. Regularity under "upgrade" versus "new user" prices follows from the analysis in sections 4 (imperfect foresight) and 7 (perfect foresight).

In addition, the fact that n is non-increasing in θ follows from the "$\theta Q - p$" utility structure: as θ increases, the user will not decide to purchase new versions less frequently, which means that n will not decrease. ■

While we show in this proposition that it is an optimal solution to upgrade regularly, it is actually the optimal behavior for all users except for those indifferent between two switching frequencies. However, such users are of measure zero and can be thus neglected.

Detail descriptions of user decisions will follow in the dedicated sections. At this moment, just note that the distribution of products across perfect foresight users differs from the distribution of products across imperfect foresight users.

2.2.4 The participation constraint and foresight

From our analysis of the decision process of a user with imperfect foresight, it follows that a necessary condition for such a user to switch every n periods at price p is $\theta\left(q_{T+n}-q_{T}\right)=\theta n q \geq p$. Using the same approach as in the derivation of (11), we can show that the same necessary condition (which is thus the participation constraint in our model) applies for users with perfect foresight. In addition, a user with imperfect foresight will switch to a new product at the earliest possible moment, so that the condition $\theta n q-p \geq 0$ is also sufficient for the minimal n at which it is satisfied. However, a user with perfect foresight may decide to wait till the next period or even longer instead of buying the new version at the first opportunity when the participation constraint is met.

Consider again a high-end user with a high sensitivity θ, who buys every period and he knows that (perfect foresight), and let the developer use the single-price policy. While we have shown that the participation constraint for this user is $\theta q-p \geq 0$, the user has other possibilities, one of which is buying every two periods. Logically, the user will prefer buying every period to buying every two periods if the discounted utility flow over a span of two periods is higher in the former case. Assume the user decides between switching in periods T and $T+1$ *and* in period T alone. The discounted (at the beginning of period T) utility flow is then $\theta Q_{T}-p+\beta(\theta Q_{T+1}-p)$ in the former case and $\theta Q_{T}-p+\beta(\theta Q_{T})$ in the latter case. Therefore, switching every period is more profitable than switching every two periods if

$$\theta\left(Q_{T+1}-Q_{T}\right)=\theta Q \geq p, \tag{12}$$

which is equivalent to $\theta \geq p/Q$. Since $q>Q$ for $\beta>0$, condition (12) is stronger than the participation constraint. In the part of this paper dealing with perfect foresight, we show that this condition is sufficient and derive the corresponding condition for $n>1$.

Remark 2 *If we assumed that the user initially owns the version from period $T-1$ and chooses between updating in periods T and $T+1$, and updating in period $T+1$ alone, the result would be the same.*

Note that this decision rule has now capital Q rather than q and recall the difference between the two: Q is the actual quality of software while q is the user-discounted flow of quality, $q=\frac{1}{1-\beta}Q$.

Notation 2 *In the rest of the paper, we will refer to both variables q, Q as "quality" keeping in mind the difference between the two.*

2.2.5 Demand

As every user switches to new software versions regularly, let $d_n = d_n(\beta, p_1, p_2, Q)$ be the measure, in terms of θ, of users who switch exactly every n periods. In general, $d_n \geq 0$ (some switching frequencies can be absent) and $\sum_{n=1}^{\infty} d_n = 1$.

We assume that the developer cannot observe the exact distribution of software versions across users, but the developer knows the function $d_n(\cdot)$. This can be interpreted in the way that the developer cannot observe in which period t he actually finds himself, so he cannot observe software distribution across users and assumes that all possible distributions are equiprobable (each distribution appears in the whole infinite model just once).

2.3 The developer's problem

The model deals with heterogenous users over infinite number of periods, and as $t \to \infty$, the number of software versions the users can own goes to infinity. The distribution of software versions differs from period to period and never repeats. Thus, the developer faces a different demand function each period, which leads to different profit Π_i, so the total profit would be $\Pi_\infty = \sum_{i=1}^{\infty} \delta^i \Pi_i$, where δ is the developer's discount factor.

To simplify our analysis, we assume that the developer values profit from every period equally[17], so that the discounting factor is $\delta = 1$. Thus, the developer's infinite-time profit maximization is equivalent to maximizing the average profit per period. In the rest of the paper, we will assume that the developer maximizes the following profit function:

$$\Pi = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \Pi_i. \tag{13}$$

Remark 3 *Fixing quality jump forever and maximizing average profit per period are key simplifications in the model that would allow us to get analytical solution while keeping the rest of the assumptions very flexible (infinite number of different products, infinite heterogenous customer, infinite number of users groups with different decisions). These simplifications are not so far from empirical observations given the fact that real-life developers are observed to keep roughly constant prices over several periods and to improve their products gradually.*

Remark 4 *As we already argued, the simplification with maximizing average profit corresponds to the situation where developer cannot observe in which period he actually finds himself, thus he assumes that all possible version distributions across consumers are equiprobable. In other words, the developer views profits in different periods as independent and identically (due to price and quality adjustment stability) distributed random variables, so that the limit in (13) equals the average per-period profit by the law of large numbers, and maximizes the average profit across all distributions. In fact, this is the same problem as stated in equation (13) except the constant.*

More specifically, the developer knows the demand function d_n, so that in every period the share d_1 of users will buy the new version (at price p_1), whereas for each $n > 1$, only $\frac{1}{n}$ of those users who switch only in every n periods will buy on the average (at price p_2, note that $p_1 = p_2 = p$ for the single price policy). Then the average per-period revenue equals $p_1 d_1 + p_2 \sum_{n=2}^{\infty} \frac{1}{n} d_n$, whereas the developer's cost equals $\bar{B}Q^2$ per period. Thus, the developer's profit per period can be rewritten as

$$\Pi = p_1 d_1 + p_2 \sum_{n=2}^{\infty} \frac{1}{n} d_n - \bar{B}Q^2. \tag{14}$$

[17] A possible interpretation is that an increase in the user base due to network effects exactly offsets the discounting of future earnings.

Also denote $B = \bar{B}(1-\beta)^2$, and recall the notation $q = \frac{Q}{1-\beta}$, which allows the expression of the developer's cost as Bq^2, a form we will prefer with imperfect foresight[18].

Notation 3 *The number of high-end users, d_1, will be also denoted N_H. The average number of low-end users who switch in a given period, $\sum_{n=2}^{\infty} \frac{1}{n} d_n$, will be denoted N_L. The average number of users who switch in a given period will be denoted $N = N_H + N_L$.*

Remark 5 *Introducing discounting $0 < \delta < 1$ would not lead to analytical solution and would require additional restriction on users without any impact on key results.*

2.4 Welfare

To measure the efficiency of the monopolist developer, we traditionally use social welfare, which consists of consumer surplus and the developer's profit. We use the average per-period welfare. Recall that consumer surplus is calculated based on $q_t = \frac{1}{1-\beta} Q_t$, so it is already discounted, and profits are not discounted by assumption. According to Proposition 1 , every user starts buying at some finite period and then buys regularly. Then consider the user with quality sensitivity θ, and let $n(\theta)$ and $p(\theta)$ be the frequency and the price at which this user switches to new versions of the product. Then the additional utility accruing to this user at every purchase is $\theta n(\theta) q - p(\theta)$, which corresponds to the participation constraint, and the per-period additional utility equals $\theta q - \frac{p(\theta)}{n(\theta)}$. Therefore, the average per-period consumer surplus equals

$$CS = \int_0^1 \left(\theta q - \frac{p(\theta)}{n(\theta)} \right) d\theta = \frac{q}{2} - \int_0^1 \frac{p(\theta)}{n(\theta)} d\theta.$$

The developer's profit equals revenue minus cost, and the per-period revenue generated by a user with quality sensitivity θ, who switches to a new version every $n(\theta)$ periods at price $p(\theta)$, equals $\frac{p(\theta)}{n(\theta)}$, whence the per-period revenue equals

$$\int_0^1 \frac{p(\theta)}{n(\theta)} d\theta,$$

and the per-period cost equals $\bar{B} Q^2 = Bq^2$.

Thus, the per-period social welfare equals

$$W = CS + \Pi = \frac{q}{2} - \int_0^1 \frac{p(\theta)}{n(\theta)} d\theta + \int_0^1 \frac{p(\theta)}{n(\theta)} d\theta - Bq^2 = \frac{q}{2} - Bq^2. \tag{15}$$

Proposition 2 *The socially optimal quality adjustment is given by*

$$q_0 = \frac{1}{4B} \Longleftrightarrow Q_0 = \frac{1}{4\bar{B}(1-\beta)}.$$

Proof. *The claim directly follows from (15).* ■

[18] Parameter $\bar{B}$ also effectively captures market size, i.e., the density of θ being a constant other than exactly 1. In such a case, the demand d_n will be multiplied by that constant, which is equivalent (as far as profit maximization is concerned) to dividing $\bar{B}$ by the constant in question.

3 A single price model for imperfect foresight users

3.1 User decision and products distribution across users

In this section, we analyze the case when the developer does not discriminate users based on purchasing history and sets a single price p to all users. The developer sets the same price p for all periods and every new version is sold for this price. In the first period, when the developer starts to operate in the market, no user possesses any version of software. The initial quality of software at time $t = 0$ is $Q_0 = 0$, or, using the notation $q = \frac{Q}{1-\beta}$, $q_0 = 0$.

When the developer releases a software of quality q in the first period, $t = 1$, for price p, it attracts the users whose utility from the software is positive: $U^{\theta}(p, q) = \theta q - p \geq 0$. Those users have $\theta \geq \frac{p}{q}$ (we assume $\frac{p}{q} < 1$ and check this assumption later), other users do not buy and wait till the next period, see Figure 1 .

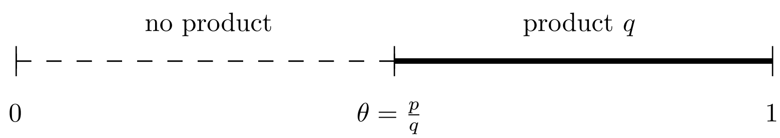

Figure 1: The distribution of products after the first period

In the second period, $t = 2$, the developer releases a new version of the software for the same price p, but with an additional quality improvement q (the quality is now $2q$), and offers this product to both groups of the users in the market—to those who already possess the version of software with quality q and to those who still do not have it. Users without software buy if $U^{\theta}(2q, p) \geq 0$ so from their utility function (3) their sensitivity θ must be higher than $\frac{p}{2q}$. Equation (3) implies that a user who already uses the software buys a new version if and only if $\theta \cdot 2q - p \geq \theta \cdot q$. Thus, after two periods, every user with sensitivity θ higher than $\frac{p}{2q}$ uses the software version of quality $2q$. See Figure 2 .

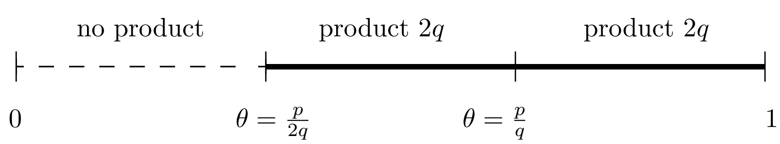

Figure 2: The distribution of products after the second period

In the third period, the developer releases a software version of quality $3q$ for the same price p. A user who does not own any version of software yet, buys the new version if his sensitivity is $\theta \geq \frac{p}{3q}$ according to (3). A user who is already using software decides according to equation (4). After the second period, users with $\theta \geq \frac{p}{2q}$ are using software of quality $2q$, so the new version of software $3q$ is bought only by users satisfying $\theta \cdot 3q - p \geq \theta \cdot 2q$ (by those with $\theta \geq \frac{p}{q}$). The distribution of software is depicted in Figure 3 .

When we look at the distribution of software version across users, we see that after the third period software of quality $3q$ is used by users with $\theta \in \langle \frac{p}{3q}, \frac{p}{2q} \rangle \cup \langle \frac{p}{q}, 1 \rangle$, software of quality $2q$ is used by users with $\theta \in \langle \frac{p}{2q}, \frac{p}{q} \rangle$ and no one is using version of quality q from the first period. Users with sensitivity to quality

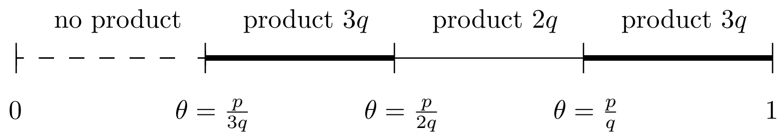

Figure 3: The distribution of products after the third period

$\theta \in \langle 0, \frac{p}{3q} \rangle$ are not using software at all and wait until the next periods.

For further analysis of software distribution across users, we have to look separately at each group of users $\theta \in \langle \frac{p}{(n+1)q}, \frac{p}{nq} \rangle$ where $n \in \langle 1, \infty)$. For each group, we look which version they are using and when they actually update (replace) the product. Users with $\theta = \frac{p}{nq}$ are marginal users who are, at selected periods, indifferent between switching to the newest version or staying with an older version (or not using the software at all). Continuing with the same approach, we obtain the following distribution of versions across users (see Figure 4).

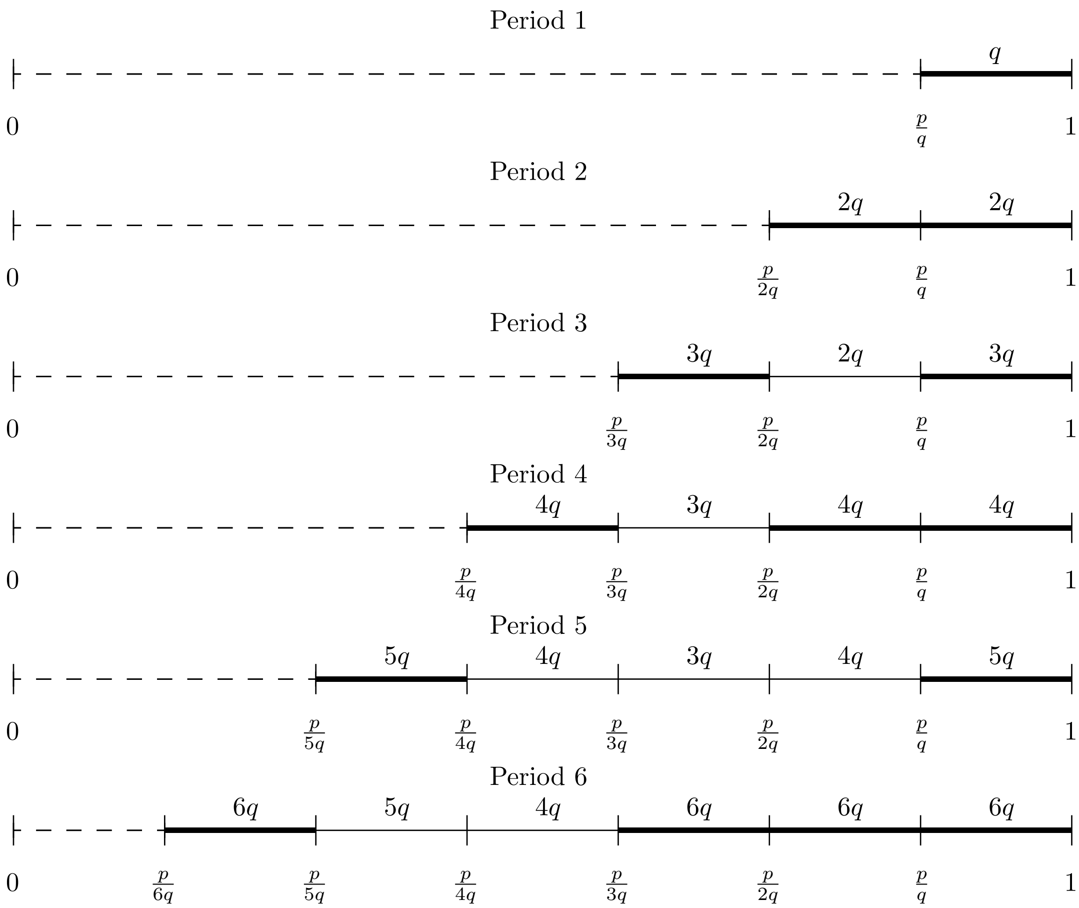

Figure 4: The distribution of products across users

The analysis presented can be summarized by the following proposition.

Proposition 3 *Users of the highest sensitivity to quality, that is,* $\theta \in [\frac{p}{q}, 1]$*, buy a new version of the*

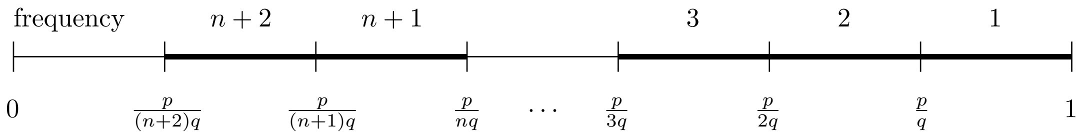

Figure 5: The distribution of switching frequencies across users

software every period and users with $\theta \in \langle \frac{p}{nq}, \frac{p}{(n-1)q} \rangle$ *buy a new version every* n*th period. As* $n \to \infty$*, the measure of those who do not use any version of software,* $\theta \in \langle 0, \frac{p}{nq} \rangle$*, goes to zero.*

It is interesting that the set of consumers who own a particular software quality is generally non-convex, and it is possible that in a given period a consumer with a higher quality sensitivity owns a lower quality version, as can be seen in Figure 4 . However, as we are interested in average per-period values, what really matters is how often consumers update. According to the proposition above, the set of consumers with the same updating frequency is convex, and (by the general result in Section 2) this frequency is non-increasing in quality sensitivity. Average frequency of switching is displayed in figure 5 .

This is equivalent to the following demand structure.

$$d_1 = 1 - \frac{p}{q},\ d_n = \frac{p}{(n-1)q} - \frac{p}{nq} = \frac{p}{q}\left(\frac{1}{n(n-1)}\right),\ n \geq 2. \tag{16}$$

3.2 The developer's problem

The developer's profit per period is generally given by (14), which, after taking into account the single price policy and substituting (16), takes the form

$$\Pi = p\left(1 - \frac{p}{q}\right) + p\sum_{n=2}^{\infty} \frac{p}{q}\frac{1}{n^2(n-1)} - Bq^2.$$

Denote

$$D = \sum_{n=2}^{\infty} \frac{1}{n^2(n-1)} = 2 - \frac{\pi^2}{6} \approx 0.355,$$

so that the profit can be written as

$$\Pi = \left(1 - \frac{p}{q}\right)p + \frac{p}{q}Dp - Bq^2. \tag{17}$$

The developer maximizes the profit by setting optimal p and q.

Proposition 4 *If the users have imperfect foresight and the developer uses the single price policy, then the developer's choice of price and quality, and the implied price-quality ratio, are the following.*

$$p^* = \frac{1}{16\,(1-D)^2\,B},\ q^* = \frac{1}{8B\,(1-D)},\ \frac{p^*}{q^*} = \frac{1}{2\,(1-D)}. \tag{18}$$

In addition, equilibrium numbers of users are

$$N_H^* = 1 - \frac{p^*}{q^*} = \frac{1-2D}{2-2D},\ N_L^* = \frac{p^*}{q^*}D = \frac{D}{2\,(1-D)},\ N^* = N_H^* + N_L^* = \frac{1}{2}, \tag{19}$$

so that and exactly half of the users are switching every period.

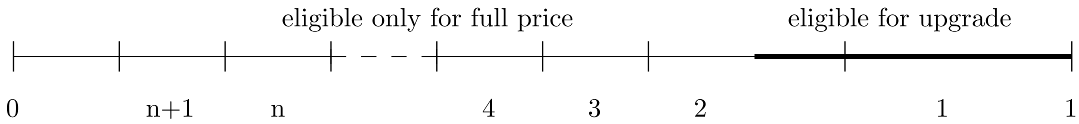

Figure 6: Market distribution in non-lock-in

Proof. The optimal price and quality are obtained from F.O.C.; S.O.C. are checked in Appendix A.3.1 . Equilibrium numbers of users are obtained directly from (18). ■

Substituting D into the equilibrium, we obtain:

$$p^* = \frac{9}{4B\left(\pi^2 - 6\right)^2} \simeq \frac{0.150262}{B}, \tag{20}$$

$$q^* = \frac{3}{4B\left(\pi^2 - 6\right)} \simeq \frac{0.193818}{B}, \tag{21}$$

$$\frac{p^*}{q^*} = \frac{3}{\pi^2 - 6} \simeq 0.775\,273. \tag{22}$$

(Note that the last equality also proves that the assumption $\frac{p}{q} < 1$ was correct.) Substituting back into (17), we obtain the developer's profit:

$$\Pi^* = \frac{1}{64B\left(1 - D\right)^2} = \frac{9}{16B\left(\pi^2 - 6\right)^2} \simeq \frac{0.037565}{B}. \tag{23}$$

Summary 1 *In this section, we derived monopoly equilibrium and distribution of user if only one price is allowed. As for the equilibrium price, this is in line with Stokey (1979, 1981) who shows that in a dynamic durable-good context when the developer can commit to the time path of prices, the monopolist precommits to the same price in all periods, which coincides with the static monopoly price. Like in many static models, the developer sells exactly to half of all users; so, equilibrium price and quality are set to reach those half of the users every period. Finally, note that the proportion of the users who buy software is independent on factor B.*

4 Imperfect foresight users and price discrimination

In this section, we analyze the situation when the developer offers a lower "upgrade" price p_1 to users who own the version from the previous period. The rest of the users, who are using older products, are not eligible for the lower price and could buy the new version only for a standard "full" price p_2. Denote the quality offered by the developer in this section Q_e, and $q_e = \frac{Q_e}{1-\beta}$.

If the upgrade price p_1 is only slightly lower than the full price p_2, some users who would buy every second period for price p_2 now prefer buying every period for price p_1; however, all users who buy every three periods do not change their decision and still buy every three periods. This leads to market coverage as in Figure 6 .

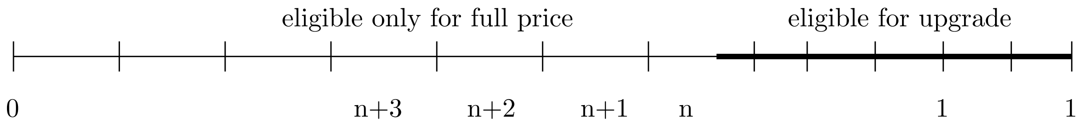

Figure 7: Market distribution in lock-in

In the case of a higher discount for upgrading users, not only those who would buy for price p_2 every two periods, but even users who would buy less frequently than every $n \geq 3$ periods would now switch every period. This market situation leads to market coverage as in Figure 7 .

We shall refer to the former case, where a user with all switching frequencies are present on the market, as to the "non-lock-in" case, while the latter case, where some frequencies of switching are out of the market, will be referred to as the "lock-in" case. We shall analyze both cases, and we start with the non-lock-in market.

4.1 The "non-lock-in" set-up

Assume in this part that the upgrade price is $p_1 \in \langle \frac{p_2}{2}, p_2 \rangle$. This condition will guarantee that all frequencies of switching are present in the market. (We show later that this condition is satisfied in equilibrium.) Let us separate the revenue generated by upgrade versions and by the full versions. We see that only users with sensitivity parameter θ greater than $\frac{p_1}{q_e}$ buy every period. These users generate revenue denoted as R_1. The rest of the users, who are not eligible for the upgrade price, generate revenue R_2. The full-price demand is similar to the single price case as in equation (16) with the exception that some users who would switch every two periods at p_2 now switch every period at p_1.The measure of those users is $\left(\frac{p_2}{q_e} - \frac{p_1}{q_e}\right)$ and since their frequency of switching was every second period in the case of a single price, we have to adjust the average demand per period for the term $\frac{1}{2}\left(\frac{p_2}{q_e} - \frac{p_1}{q_e}\right)$. Thus, the revenue functions are:

$$R_1 = \left(1 - \frac{p_1}{q_e}\right) \cdot p_1, \tag{24}$$

$$R_2 = \left(\frac{p_2}{q_e} D - \frac{1}{2}\left(\frac{p_2}{q_e} - \frac{p_1}{q_e}\right)\right) \cdot p_2,$$

and the developer's profit is:

$$\Pi = \left(\frac{p_2}{q_e} D - \frac{1}{2}\left(\frac{p_2}{q_e} - \frac{p_1}{q_e}\right)\right) \cdot p_2 + \left(1 - \frac{p_1}{q_e}\right) \cdot p_1 - B q_e^2. \tag{25}$$

Remark 6 *Imperfect foresight means that the users with θ only slightly higher than $\frac{p_1}{q_e}$, one period after buying, will find it profitable to buy the newest version at p_1, so they will buy. These users do not consider the possibility that they can be better off buying every two periods at p_2 rather than every period at p_1 whether it can actually happen or not.*

Proposition 5 *If the users have imperfect foresight, and the developer price discriminates as described, then the developer's choice of prices and quality is the following.*

$$p_1^* = 4 \frac{(1-2D)^2}{(16D-7)^2 B}, \; p_2^* = 2 \frac{1-2D}{(16D-7)^2 B}, \; q_e^* = \frac{2D-1}{B(16D-7)}. \tag{26}$$

The equilibrium numbers of users are

$$
\begin{aligned}
N_H^* &= \frac{3-8D}{7-16D} \simeq 0.120909, \\
N_L^* &= \frac{2D-1}{16D-7} \simeq 0.219772, \\
N^* &= 2\frac{2-5D}{7-16D} = \frac{5\pi^2-48}{8\pi^2-75} \simeq 0.340681.
\end{aligned} \tag{27}
$$

Proof. The optimal prices and quality are obtained from F.O.C.; S.O.C. as well as the condition $p_1 \in \langle \frac{p_2}{2}, p_2 \rangle$ are checked in Appendix A.4.1 . The numbers of users are obtained directly from these values.
■

Substituting D into the equilibrium, we obtain:

$$p_1^* = \frac{4}{B}\frac{\left(\pi^2-9\right)^2}{\left(8\pi^2-75\right)^2} \simeq \frac{0.193200}{B}, \tag{28}$$

$$p_2^* = \frac{6\left(\pi^2-9\right)}{B\left(8\pi^2-75\right)^2} \simeq \frac{0.333255}{B}, \tag{29}$$

$$q_e^* = \frac{\left(\pi^2-9\right)}{B(8\pi^2-75)} \simeq \frac{0.219772}{B}, \tag{30}$$

$$\frac{p_1^*}{q_e^*} = \frac{4\pi^2-36}{8\pi^2-75} \simeq 0.879090, \tag{31}$$

$$\frac{p_2^*}{q_e^*} = \frac{6}{8\pi^2-75} \simeq 1.516363. \tag{32}$$

substituting (26) into (25), we obtain the developer's profit:

$$\Pi^{D*} = \frac{1}{B}\frac{\left(2D-1\right)^2}{\left(16D-7\right)^2} = \frac{1}{B}\frac{\left(\pi^2-9\right)^2}{\left(8\pi^2-75\right)^2} \simeq \frac{0.048300}{B}. \tag{33}$$

Lemma 1 *If the users have imperfect foresight, and the developer uses either the single-price policy or "non-lock-in" price discrimination, then all equilibrium prices, qualities, and profits are dependent on the discount factor β, and all of them are increasing in β. On the other hand, the numbers of switching users N_H^*, N_L^*, N^* are independent on β.*

Proof. Can be seen immediately from the results above by recalling that $B = \bar{B}(1-\beta)^2$ and $q = \frac{1}{(1-\beta)}Q$.
■

Remark 7 *If we compare profit made by a single price developer and a developer who undertakes price discrimination based on upgrades, we notice that the profit for price discrimination is higher in the latter case. Clearly, the discriminating developer always has the option to set prices equally $p_1 = p_2$. The exact relationship between the profits is given by:*

$$\Pi^{D^*} - \Pi^* = \frac{1}{B}\frac{(2D-1)^2}{(16D-7)^2} - \frac{1}{64B(1-D)^2} \simeq \frac{0.010734}{B} > 0. \tag{34}$$

Remark 8 *In our analysis, we neglect the fact that in the first period all users should pay full price, including the users who will buy regularly every period afterwards. This simplification is in line with the basic form (13) for profit, which implies that any finite number of periods in the beginning can be neglected. Note* *again, that imperfect foresight users do not predict future switching to a new version so, for instance, they do not consider the situation that under certain prices p_1, p_2 they can buy a product for the full price p_2 with negative immediate utility and offset it later on with the future positive utility flow from upgrades. Such considerations will be analyzed in the case of perfect foresight.*

4.2 The "lock-in" set-up

In the case of lock-in, we deal with different optimization problems. There are only users who switch every period, and then there are users who switch less frequently than every two periods. Users switching exactly every two periods are not in the market. Consider now the general case, where only users switching every period and then every n or more periods are in the market (further referred to as n-lock-in). Assume $p_1 \in \langle \frac{p_2}{n}, \frac{p_2}{n-1} \rangle$. (Again, we will show later that this condition is satisfied in equilibrium.)

Remark 9 *Note that from a mathematical point of view, non-lock-in is a particular case of lock-in at $n = 2$ ("2-lock-in").*

Denote $\theta_{1,n}$ a user who is indifferent between switching every period and every n periods, and $\theta_{n,n+1} = \frac{p_2}{nq_e}$ is the user who is indifferent between switching every n and every $n+1$ periods. Denote again revenue from users switching every period as $R_1 = (1 - \theta_{1,n}) \cdot p_1$. From users utility function (3) in view of Remark 6 , we see that a user indifferent between switching every period and every n periods is a user with sensitivity to quality $\theta_{1,n} = \frac{p_1}{q_e}$, so the revenue from users switching every period is $R_1 = \left(1 - \frac{p_1}{q_e}\right) \cdot p_1$.

Revenue from users switching less frequently consists of revenue from users who switch every n periods: $\frac{1}{n}(\theta_{1,n} - \theta_{n,n+1})p_2$ and from users switching even less frequently $\frac{p_2}{q_e} D_{n+1} p_2$:

$$R_2 = \frac{1}{n}\left(\theta_{1,n} - \theta_{n,n+1}\right) p_2 + \frac{p_2}{q_e} D_{n+1} p_2,$$

where

$$D_{n+1} = \sum_{m=n+1}^{\infty} \frac{1}{m^2(m-1)} = \frac{1}{n} - \psi_1(n+1),$$

where $\psi_1(\cdot)$ is the polygamma function of order 1.

Remark 10 *As could be seen directly from the definition, D_n is decreasing and goes to zero as n goes to infinity.*

Summing up the revenue for both groups R_1, R_2, taking into account the developer's costs Bq_e^2, and re-arranging, we obtain the developer's profit function:

$$\Pi = \frac{p_2}{nq_e}\left(p_1 + p_2 - np_2\psi_1(n)\right) + \left(1 - \frac{p_1}{q_e}\right) p_1 - Bq_e^2. \tag{35}$$

Proposition 6 *If the users have imperfect foresight and the developer price discriminates so that no user switches every $2, \ldots, n-1$ periods, then the developer's choice of prices and quality is the following.*

$$p_1^* = \frac{n^2}{B} \frac{\left(n\psi_1(n) - 1\right)^2}{\left(4n^2\psi_1(n) - 4n - 1\right)^2},\ p_2^* = \frac{n^2}{2B} \frac{n\psi_1(n) - 1}{\left(4n^2\psi_1(n) - 4n - 1\right)^2},\ q_e^* = \frac{n}{2B} \frac{n\psi_1(n) - 1}{4n^2\psi_1(n) - 4n - 1},$$

and the average number of switching users per period is:

$$N^{D*} = \frac{n(1+2n)\psi_1(n) - 2(1+n)}{4n^2\psi_1(n) - 4n - 1}. \tag{36}$$

Proof. The optimal prices and quality are obtained from F.O.C.; S.O.C. as well as the condition $p_1 \in \langle \frac{p_2}{n}, \frac{p_2}{n-1} \rangle$ are checked in Appendix A.4.2 . The value N^{D*} is then derived by substitution. ■

Substituting p_1^*, p_2^*, q_e^* into the profit function (35), we obtain the equilibrium profit:

$$\Pi^* = \frac{n^2}{4B} \frac{(n\psi_1(n) - 1)^2}{(4n^2\psi_1(n) - 4n - 1)^2}.$$

Remark 11 *By substituting p_1^*, p_2^*, q_e^* into the definition of indifferent users, we see that in equilibrium the indifferent users are:*

$$\theta^*_{1,n} = \frac{p_1^*}{q_e^*} = \frac{1}{2}\left(1 + \frac{1}{4n^2\psi_1(n) - 4n - 1}\right), \ \theta^*_{n,n+1} = \frac{p_2^*}{nq_e^*} = \frac{1}{4n^2\psi_1(n) - 4n - 1}.$$

Lemma 2 *If the developer uses n-lock-in under imperfect foresight, then as n increases, the average number of switching users decreases, and equilibrium quality change, both prices, both prices per quality change, and the developer's profit increase. In addition, the equilibrium quality converges from below to the socially optimal value $q_0 = \frac{1}{4B}$ as n increases.*

Proof. Directly follows from the results above. ■

The last result implies that the best strategy for the developer in our model is to set n close to infinity. This result is caused by the set-up of our model, where the developer is maximizing average profit. Thus, the developer does not differentiate between profit realized in the first period and profit from a period close to infinity. Maximizing the average profit instead of the discounted flow of profit is a simplification to obtain an analytical solution[19]. However, remark that for those who switch every n periods, the developer starts receiving revenue from this group for the first time at period n; thus, it would be natural to assume that the profit from this group is discounted by β^n, which would immediately imply that for $\beta < 1$, the optimal n is finite.

5 Imperfect foresight users: comparisons and welfare analysis

5.1 A single price versus price discrimination

In this section, we contrast price discrimination with the single price equilibrium, so we compare changes in quality, prices, prices per unit of quality change $\frac{p^*}{q^*}$, the total number of switching users, and welfare for both cases. We do the comparison for the "non-lock-in" case and then extend the results to the general "lock-in" case via Lemma 2 . The proofs of the results stated here are made by direct arithmetical comparison of the relevant results above and are thus omitted.

[19]Introducing the discounted developer profit would require a numerical solution even for the simplest possible case of the single price and imperfect foresight.

Lemma 3 *The product quality in the case of a single price is lower than the quality set by the developer discriminating the users based on upgrades.*

This result is in line with the general view of the impact of price discrimination. From a purely quality point of view, the developer using price discrimination based on a lower price of the upgrade accelerates quality evolution more than a single price monopoly.

Lemma 4 *If price discrimination based on a lower upgrade price is not possible, the single price* p^* *is lower than the full price* p_2^**, and even lower than the price of an upgrade version* p_1^**.*

This result may be counter-intuitive at first sight as we should be aware that the quality $q_e^* > q^*$, so the higher price even for upgrades is justified by a quality change.

Another insightful comparison we obtain if we compare the price per quality[20], $\frac{p^*}{q^*}$.

Lemma 5 *The equilibrium price per quality* $\frac{p^*}{q^*}$ *is lower if possibility for discrimination does not exist, that is* $\frac{p^*}{q^*} < \frac{p_1^*}{q_e^*} < \frac{p_2^*}{q_e^*}$*. Consequently, the number of buyers is higher for a developer that charges the single price.*

Lemma 6 *The proportion of users buying in every period in the case of a single price is higher than in the case of price discrimination.*

This is a very interesting result. Generally, the motivation for price discrimination is to increase revenue by increasing the number of users. In standard price discrimination, the price for a more sensitive group is lower than for the other groups, and it is profitable to sell till the price reaches marginal costs.

On the other hand, a lower price for upgrades persuades some users to buy more frequently which, in turn, raises the number of users. The key factor of this price discrimination is that the discount is not offered to the most price sensitive users but to the most quality sensitive users. A possible interpretation of this is that price discrimination by using upgrades helps the developer to better separate the quality sensitive users buying in every period and to accelerate software evolution (recall that the quality change is higher under price discrimination) to fit their demand better. As for the low-end users, the full price for the software is set very high because the developer knows that even though their sensitivity for quality is lower, the fact that they possess an older version means that their valuation of the new software eventually becomes relatively high; thus, these users will buy sometimes as well. Accelerating software evolution enables the developer to charge a higher price to all of them. In a single-price model, the developer values low-end users relatively more, so he must attract a higher number of users by keeping both price and quality lower.

Remark 12 *Note, that in our comparison, when we compare single price and price discrimination equilibria, we have different equilibrium qualities* $q^* < q_e^*$*. Thus, the higher prices (and the lower number of users) in the price discrimination case is partially caused by a higher equilibrium quality. For a better understanding of this effect, we can decompose "moving" from the single price equilibrium to the price discrimination equilibrium into two steps. In the first step, we fix quality at the single price equilibrium level* q^* *while allowing price*

[20] Though in the literature the inverse, i.e., quality per price unit. $\frac{q}{p}$ is usually used for comparison, we use $\frac{p}{q}$ as it plays a key role in the distribution of switching customers in our model.

discrimination, and in the second step, we adjust quality (and prices) to their profit-maximizing levels. This analysis is performed in Appendix A.5.1 .

The results above apply to the "non-lock-in" case. It follows from Lemma 2 that if the developer decides to use n-lock-in, the results above are reinforced as n increases.

5.2 Welfare analysis

In this section, we analyze welfare changes by comparing price discrimination and single-price equilibria. We analyze consumer surplus (CS) for all, low-end and high-end users, as well as social welfare W, which is consumer surplus plus the developer's profit. For all these variables, we compare their average per-period values (in fact, all these differ from period to period). The values for a single-price developer can be found in Appendix A.5.2 , and the values for a price discriminating developer can be found in Appendix A.5.3 . In particular, the average CS per period under a single price equals

$$CS = \frac{1}{32}\frac{1-2D}{(1-D)^2 B} \approx \frac{0.021778}{B},$$

and under price discrimination with n-lock-in, the average CS per period equals

$$CS = \frac{1-\left(4n^2\psi_1(n)-4n-1\right)^{-2}}{32B}.$$

The following results are proved by direct comparison of the values derived in Appendices A.5.2 and A.5.3 .

Proposition 7 *The consumer surplus is higher in the case of a single-price monopoly than under non-lock-in price discrimination and decreases in n as n-lock-in is used.*

The consumer surplus for both high-end and low-end users is higher in the case of a single-price developer than under non-lock-in price discrimination, and decreases in n, and the same applies to CS per buyer for high-end users. As for consumer surplus per buyer for all users or for low-end users, they are higher under n-lock-in than under the single price except for $n = 2$ (non-lock-in) and $n = 3$, and increase in n.

Social welfare is lower in the case of a single-price developer than under non-lock-in price discrimination, increases in n as n-lock-in is used, and approaches the socially optimal level from below as n goes to infinity.

The different relation between CS per buyer for all or low-end users under a single price and under n-lock-in at $n = 2$ and $n = 3$ seems to be just a mathematical property of the model—recall that the developer prefers "infinite" lock-in so that in the end those CS are higher under price discrimination. See the discussion in section 8 for better exposition.

The intuition is that as the length of the lock-in period increases, the developer becomes "more precise" in the sense of extracting more and more consumer surplus by setting the appropriate price-to-quality ratio since the targeting group (upgraders) becomes more narrow. Thus, per-period CS in the lock-in case tends to zero as n increases to infinity. In the limit (which is mathematically unreachable), the monopolist makes all consumers pay a price of q^0 per period (i.e., p_2 has an asymptotic behavior of nq^0), where q^0 is also the socially optimal value of quality. Thus, much like in the case of the first degree price discrimination, the developer can in the limit extract the entire consumer surplus. Unlike in the case of the first degree price discrimination, however, this happens due to the dynamic nature of the model: as n increases, the size of the largest consumer group (in terms of switching frequency) in the market decreases, and it is easier to extract surplus from separate smaller groups. This can be interpreted as an example of first-degree price discrimination as the limiting case of third-degree one (see also the Remark before Proposition 11).

6 Perfect foresight users: single price developer

6.1 The user decision

In this section, we analyze the developer's behavior if all users have perfect foresight. The rest of the set-up is the same as in the previous part, which means that in every period t, the developer introduces a new version of software and sells only this new version. Every new version has a quality improvement Q over the previous version, so that at period t the software quality is $Q_t = tQ$. The decision process for perfect foresight users is the same as for users with imperfect foresight: users compare the utility flow from keeping the currently possessed version with the utility flow from switching to a new version. In the case of perfect foresight, however, every user can anticipate the optimal frequency of switching to a new version. A user of quality sensitivity θ with perfect foresight calculates the utility flow $U_n(\theta)$ from switching every n periods (for all $n \in \langle 1, \infty)\rangle$), and then he decides for such switching frequency n that brings him the maximal utility flow. Naturally, the optimal frequency of switching is fully dependent on the sensitivity to quality θ.

By perfect foresight user, we mean a user who calculates the utility flow and the optimal frequency of switching at the moment of buying a new version and later follows this decision. Alternatively, a user may calculate the utility flow every period and see whether switching would bring him higher utility or whether he should wait till the next period. When such a user calculates the utility flow every period, he takes into account that he will follow the same decision in the future so he incorporates into the calculation his future decisions that are aligned with his current decision. In the case of a single price, these two approaches are equivalent (see Remark 13).

According to the analysis presented in section 2, the necessary condition for a user with perfect foresight to switch every n periods at price p is $\theta nq - p \geq 0$, and while this necessary condition is the same as under imperfect foresight, it is no longer sufficient. As mentioned above, when a user decides for an optimal frequency of switching, he must compare the utility flow for all possible frequencies of switching n, where for each frequency he calculates the infinite-time utility flow. When doing so, the user must calculate the utility flow with respect to all possible future switching. To decide between two possible frequencies of switching, the user can just compare the utility flow between two periods when he would always switch to a new version, no matter which of the two frequencies of switching he would select. The simplest example is the decision between switching every period and every two periods, which yields the threshold of $\theta_{12} = \frac{p}{Q}$ in (12) as analyzed in Chapter 2.

For another example, consider a user with quality sensitivity θ who is comparing the utility flow from switching every two periods and every three periods. Assuming that the user decides to buy in period t, the periods when this user always switches to a new version, no matter whether he decides for frequency 2 or 3, are periods $t+6$, $t+12$, $t+18$, ... If the utility flow from one of those frequencies is higher for the next 6 periods, then this user naturally prefers this frequency in all of his future decision periods $t+6$, $t+12$, $t+18$, ... Thus, to compare which frequency of switching is better, it is enough in the example above to compare the utility flow from 6 periods since the decision will be regularly repeated every 6 periods.

Assume now that version available in the market in period t has quality tQ. Buying every three periods implies switching two times to a new version within six periods. For the first time, the user switches in period t to the version of quality tQ and pays the price p, and for the second time, he switches at period $t+3$ to the new version of quality $(t+3)\,Q$ and pays the price p again. Thus, at time t, the discounted utility flow from the next six periods is:

$$U_{n=3}(\theta) = \theta\left(tQ + tQ\beta + tQ\beta^2 + (t+3)\,Q\beta^3 + (t+3)\,Q\beta^4 + (t+3)\,Q\beta^5\right) - p - p\beta^3. \tag{37}$$

Similarly, switching every two periods requires switching at periods, t, $t+2$, $t+4$, so that the user obtains the following utility flow from the next 6 periods:

$$\begin{aligned} U_{n=2}(\theta) = \theta\left(tQ + tQ\beta + (t+2)\,Q\beta^2 + (t+2)\,Q\beta^3 + (t+4)\,Q\beta^4 + (t+4)\,Q\beta^5\right) - \\ - p - p\beta^2 - p\beta^4. \end{aligned} \tag{38}$$

Comparing the utility flows $U_{n=3}(\theta)$ and $U_{n=2}(\theta)$, the user sees which frequency of switching is better for him. From the above utility flows $U_{n=3}(\theta)$ and $U_{n=2}(\theta)$, we can immediately derive that the user indifferent between switching every two and every three periods has quality sensitivity

$$\theta_{23} = \frac{p}{Q}\frac{1}{(\beta+2)}, \tag{39}$$

where the user with quality sensitivity θ prefers switching every two periods to switching every three periods if $\theta > \theta_{23}$ and switching every three periods to switching every two periods if $\theta < \theta_{23}$.

Remark 13 *In this example, the decision process is equivalent to an alternative decision process when a user with quality sensitivity θ at period t possesses the product from period $t-2$ and decides whether to switch or to wait till the next period. The utility flows are then:*

$$U_{n=3} = \theta\left((t-2)\,Q + (t+1)\,Q\beta + (t+1)\,Q\beta^2 + (t+1)\,Q\beta^3 + (t+4)\,Q\beta^4 + (t+4)\,Q\beta^5\right) - \beta p - p\beta^4,$$

and

$$U_{n=2} = \theta\left(tQ + tQ\beta + (t+2)\,Q\beta^2 + (t+2)\,Q\beta^3 + (t+4)\,Q\beta^4 + (t+4)\,Q\beta^5\right) - p - p\beta^2 - p\beta^4.$$

Equating the two expressions above and solving for θ, we see again that the indifferent user is $\theta_{23} = \frac{p}{Q}\frac{1}{(\beta+2)}$. The same result can be derived for the case when the user owns a version from period $t-1$ and chooses between waiting one or two periods. Thus, the approaches to the decision process are equivalent, and it is not important in which period the user makes a decision about the optimal frequency of switching.

Remark 14 *From the derivation of the indifferent user θ_{23} and from the previous remark, we see that the user's decision about optimal switching to a new version is dependent only on β, Q, and p, but independent on the decision period t.*

Lemma 7 *Given the discount factor β, the software price p, and the per-period quality improvement Q, the user indifferent between switching every n and every $n+1$ periods has quality sensitivity*

$$\theta_{n,n+1} = \frac{p}{Q} \frac{(1-\beta)^2}{n(1-\beta) - \beta(1-\beta^n)}. \tag{40}$$

The users with $\theta > \theta_{n,n+1}$ strictly prefer switching every n periods to switching every $n+1$ periods, and the users with $\theta < \theta_{n,n+1}$ strictly prefer switching every $n+1$ periods to switching every n periods. This value does not depend on which version, if at all, is possessed by the user at any given time. Moreover, this threshold decreases in β, is not lower than the corresponding imperfect foresight value (with p and Q fixed), and the utility flow to the indifferent user is non-negative so that the participation constraint holds.

Proof. The derivation of the threshold (40) and the rest of the proof can be found in Appendix A.6.1 .

■

Remark 15 *Note that at $\beta = 0$, (40) yields $\theta_{n,n+1} = \frac{p}{nQ}$, which is the imperfect foresight value (at $\beta = 0$, $q = Q$), and the limiting value at $\beta \to 1$ is $\theta_{n,n+1} = \frac{p}{Q}\frac{2}{n+n^2}$.*

6.2 Product distribution across users

Users who switch every n periods are those whose quality sensitivities satisfy $\theta \in (\theta_{n,n+1}, \theta_{n-1,n})$; thus, substituting (40), we see that the users who switch every n periods are those with θ from the interval:

$$\theta \in \left(\frac{p}{Q} \frac{(1-\beta)^2}{n(1-\beta) - \beta(1-\beta^n)}, \frac{p}{Q} \frac{(1-\beta)^2}{(n-1)(1-\beta) - \beta(1-\beta^{n-1})} \right), \tag{41}$$

and the number of users switching with every n periods is

$$N_n = \theta_{n-1,n} - \theta_{n,n+1} = \frac{p}{Q} \frac{(1-\beta)^3 (1-\beta^n)}{(1-\beta^n - n(1-\beta))(\beta(1-\beta^n) - n(1-\beta))}. \tag{42}$$

The previous equations (40), (41), (42) are valid for all users who switch with frequency $n \in \langle 2, \infty)$. Users who switch every period are just from interval $\theta \in \langle \frac{p}{Q}, 1 \rangle$ (note that $\theta_{12} = \frac{p}{Q}$) and their number is $N_1 = 1 - \frac{p}{Q}$. Summing all switching users over all $n \in \langle 1, \infty)$ together, we obtain that, on the average, the number of switching users per period is:

$$\begin{aligned} N &= \left(1 - \frac{p}{Q}\right) + \sum_{n=2}^{\infty} \frac{1}{n} N_n \\ &= \left(1 - \frac{p}{Q}\right) + \frac{p}{Q} \sum_{i=2}^{\infty} \frac{1}{i} \frac{(1-\beta)^3 (1-\beta^i)}{(1-\beta^i - i(1-\beta))(\beta(1-\beta^i) - i(1-\beta))}. \end{aligned}$$

For simplicity, denote

$$L(\beta) = \sum_{i=2}^{\infty} \frac{1}{i} \frac{(1-\beta)^3 (1-\beta^i)}{(1-\beta^i - i(1-\beta))(\beta(1-\beta^i) - i(1-\beta))}. \tag{43}$$

Then, given the price p, the quality improvement Q, and the discount factor β, the number of switching users per period on the average, is

$$N = \left(1 - \frac{p}{Q}\right) + \frac{p}{Q} L(\beta). \tag{44}$$

Lemma 8 *$L(\beta)$ is increasing in β, with $L(0) = D = 2 - \frac{1}{6}\pi^2 \approx 0.355066$ and, in the limit, $L(1) = 7 - \frac{2}{3}\pi^2 \approx$ 0.420264.*

Proof. Equation (43) cannot be generally expressed analytically for β other than 0 and 1. To verify the statement, we used a numerical simulation where we showed that $L(\beta)$ increases in β. Selected simulation results can be found in Appendix A.8.1 . The limiting values can be derived using (42) and Remark 15 . ■

Remark 16 *It is clear that the introduction of perfect foresight has an impact on the distribution of indifferent users. If the prices and quality adjustment are fixed, then the threshold $\theta_{n,n+1}$ is higher under perfect foresight. This means that the proportion of users switching less frequently is higher than in the case of imperfect foresight.*

6.3 Equilibrium

From a mathematical point of view, the problem faced by the single-price developer when the users have perfect foresight is the same as in the imperfect foresight case. The only difference, in the single price framework, between perfect and imperfect foresight is in the numerical values of the equilibrium solution. The analytic form remains the same with $L(\beta)$ in the place of D, and the distribution of indifferent users is different. We can use all results from the single price case under imperfect foresight. Thus, using the profit function (17) derived for imperfect foresight users, we obtain:

$$\Pi(p, Q, \beta) = \left(1 - \frac{p}{Q}\right) p + p\frac{p}{Q} L(\beta) - \bar{B}Q^2. \tag{45}$$

Using the results from Chapter 4, we obtain:

$$p^*(\beta, B) = \frac{1}{16\left(1 - L(\beta)\right)^2 \bar{B}},\ Q^*(\beta, B) = \frac{1}{8\bar{B}\left(1 - L(\beta)\right)},\ \frac{p^*}{Q^*} = \frac{1}{2\left(1 - L(\beta)\right)}, \tag{46}$$

and the equilibrium profit:

$$\Pi^*(\beta, B) = \frac{1}{64\bar{B}\left(1 - L(\beta)\right)^2}.$$

As β increases, future present value of utility flow increases and for the same quality jump Q^* every user is willing to pay more; thus, the developer can increase price p^*. However from the developer's point of view, it is the same as raising the sensitivity to quality; thus, he can also raise produced quality, which in turn increases his profit. We can summarize this in the following lemma:

Proposition 8 *$p^*(\beta)$, $Q^*(\beta)$, $\frac{p^*}{Q^*}(\beta)$, and $\Pi^*(\beta)$ are increasing in discount factor β and decreasing in parameter $\bar{B}$, and the number of switching users in the case of the single-price is independent on β and is equal to $N^* = \frac{1}{2}$, which is the same number as in the case of an imperfect foresight set-up.*

Proof. $L(\beta)$ is increasing in β, and as can be immediately seen from the equilibrium, p^*, Q^*, $\frac{p^*}{Q^*}$, Π^* are decreasing in $\bar{B}$, and they are increasing in L, so that they are all increasing in β. The value N^* is then obtained directly. ■

7 Perfect foresight users and price discrimination by an upgrade version

7.1 General set-up

In the perfect foresight set-up with discrimination by upgrades, every user sees which frequency of switching is the best for him, and a user who switches to a new version every period for a lower upgrade price p_1 knows that not switching in one period means that in the next period he is not eligible for the upgrade price anymore and should pay the higher (full) price p_2. If the price p_2 is relatively high with respect to the upgrade price p_1, then there would be users who would rather switch every period for a lower price p_1 than switch for the full price more frequently than every n periods. In this case, the high difference between the prices p_1 and p_2 crowds out users with lower frequencies of switching, e.g. 2, 3, up to $n-1$.

Assume that the developer sets prices p_1 and p_2 in a way to crowd out users who would switch every $2, .., n-1$ periods, and so, only users switching every period and every n (and more) periods are on the market. Then denote the user indifferent between switching every period at the upgrade price p_1 and every n periods at the full price p_2 as $\theta_{1,n}$, and the developer's profit function combines the features of the profit function (45) from the perfect foresight set-up with single price as well as the profit function from imperfect foresight with price discrimination:

$$\Pi = (1-\theta_{1,n})\, p_1 + \frac{1}{n}(\theta_{1,n} - \theta_{n,n+1})\, p_2 + \frac{p_2}{Q} L_{n+1}(\beta)\, p_2 - \bar{B}Q^2, \tag{47}$$

where the profit function consist of four parts:

1.	$(1-\theta_{1,n})\, p_1$	Revenue generated by users switching every period;
2.	$\frac{1}{n}(\theta_{1,n} - \theta_{n,n+1})\, p_2$	Revenue generated by users switching exactly every n periods;
3.	$\frac{p_2}{Q} L_{n+1}(\beta)\, p_2$	Revenue generated by users switching less than every n periods; and
4.	$\bar{B}Q^2$	Cost of product development with quality jump Q.

Here $L_{n+1}(\beta)$ is analogous to $L(\beta)$ in (43) with the initial $n-1$ terms of the sum not included. The definition of $L_{n+1}(\beta)$ is thus the following:

$$L_{n+1}(\beta) = \sum_{i=n+1}^{\infty} \frac{1}{i} \frac{(1-\beta)^3 \left(1-\beta^i\right)}{\left(1-\beta^i - i(1-\beta)\right)\left(\beta\left(1-\beta^i\right) - i(1-\beta)\right)}$$

(note that $L_2(\beta)$ is the same as $L(\beta)$).

7.1.1 Necessary conditions for equilibrium existence

- **Condition 1:** There are users who would switch every period:

$$0 \leq \theta_{1,n} \leq 1. \tag{48}$$

- **Condition 2**: There are users who would switch every n periods, but no user would switch every $n-1$ periods:

$$\theta_{n,n+1} \leq \theta_{1,n} \leq \theta_{n-1,n}. \tag{49}$$

Remark 17 *The whole problem could be generally solved using Lagrangian multipliers and Kunh–Tucker conditions; however, we approach it by using an unconstrained maximization problem with the further identification of binding constraints that are later incorporated into the decision process. This approach gives more insights on model behavior.*

Remark 18 *Other conditions that must hold are participation constraints $\theta_{m,m+1}mq - p_2 \geq 0$ for $m \geq n$ and $\theta_{1,n}q - p_1 \geq 0$. However, these conditions hold by the construction of the respective thresholds as is shown in the corresponding propositions.*

7.1.2 A user indifferent between switching every period and every n periods

As compared to perfect foresight with a single price, the problem of the indifferent user is now more complicated. If a user owns a version which is more than one period older, then this user is only eligible for the full price p_2 so that the decision on the frequency of switching follows the rule (40) with p_2 instead of p, so that the optimal switching frequency *at the full price* $n = n(\theta, p_2, Q, \beta)$ is derived. However, if the user owns the version from the previous period, then the user is also eligible for the upgrade price p_1, so that two options are available. First, the user can exercise his upgrade price claim, and if this is optimal, then this will be done in every subsequent period as the same choice will be faced. Second, the user may wait $n-1$ periods so that the version at hand becomes n periods old and then switch every $n = n(\theta, p_2, Q, \beta)$ periods[21].

Lemma 9 *Given the discount factor β, the software prices p_1 and p_2, and the per-period quality improvement Q, the user indifferent between switching every period at the upgrade price p_1 and in every n periods at the full price p_2 has quality sensitivity*

$$\theta_{1,n} = (1-\beta)\,\frac{p_1\,(1-\beta^n) - p_2\,(1-\beta)\,\beta^{n-1}}{(1-\beta^n - n\beta^{n-1}(1-\beta))Q}. \tag{50}$$

The users with $\theta > \theta_{1,n}$ strictly prefer switching every period at p_1 to switching every n periods at p_2, and the users with $\theta < \theta_{1,n}$ strictly prefer switching every n periods at p_2 to switching every period at p_1. Moreover, if n is optimal at price p_2, then the utility flow to the indifferent consumer is non-negative so that the participation constraint is satisfied.

Proof. The derivation of the threshold (50) and the proof that the participation constraint is satisfied can be found in Appendix A.7.1 . ■

For the sake of convenience, denote

$$X = \frac{(1-\beta)^2}{n\,(1-\beta) - \beta\,(1-\beta^n)},\ Y = \frac{(1-\beta)\,(1-\beta^n)}{1-\beta^n - n\beta^{n-1}(1-\beta)},\ Z = \frac{(1-\beta)^2\,\beta^{n-1}}{1-\beta^n - n\beta^{n-1}(1-\beta)},$$

so that the thresholds can be written as $\theta_{n,n+1} = \frac{p_2}{Q}X$ and $\theta_{1,n} = \frac{p_1}{Q}Y - \frac{p_2}{Q}Z$.

7.2 Equilibria

The approach for calculating equilibria is the following. Given β, we fix n and look for an equilibrium assuming that conditions (48) and (49) are satisfied. If they indeed hold, we have an interior equilibrium.

[21] From the optimality of $n(\theta, p_2, Q, \beta)$, it follows that the user will not consider the options involving either switching (at p_2) at other frequencies or waiting for any other number of periods than $n-1$ and then switching every n periods.

In case some condition is not satisfied, we incorporate this condition into the profit function and calculate the equilibrium again. Then we look for $n \in (2, \infty)$ such that the developer maximizes his profit. Unfortunately, we cannot internalize n into a general solution and the only possibility is to verify the solution for the problem for all n. However, from the solution, it will be clear how the pattern of equilibria changes based on n and β.

As in the case of imperfect foresight, we will distinguish "lock-in" and "non-lock-in" equilibria. By a "lock-in" equilibrium we understand the developer's strategy when he sets the price difference between the upgrade price p_1 and the full price p_2 so high that there is no user switching every two periods (or even more). Consequently, a "non-lock-in" equilibrium is an equilibrium when prices p_1 and p_2 are in such a relation that there are users switching every two periods. As is shown later, the only condition that may be violated after an unconstrained optimization is $\theta_{1,n} \geq \theta_{n,n+1}$, which effectively means that it is optimal for the developer to increase n. The distribution of switching frequencies across users is qualitatively the same as in the imperfect foresight case, see Figures 6 and 7 , though the threshold values are different.

7.2.1 "Non-lock-in" equilibria

Necessary conditions for the existence of an interior non-lock-in equilibrium[22] are $\theta_{2,3}(p_2^*) \leq \theta_{1,2}(p_1^*, p_2^*) \leq \min\{\theta_{1,2}(p_2^*), 1\}$, and since equilibrium prices and the quality jump are dependent on β, the resulting necessary condition will be fully dependent on β too. As we have shown already in (50), the indifferent user between switching every period at p_1 and every two periods at p_2 has quality sensitivity

$$\theta_{1,2}(p_1, p_2) = \frac{p_1}{Q}(1+\beta) - \frac{p_2}{Q}\beta. \tag{51}$$

From equation (39), we know that a user who is indifferent between switching every second and every third period (at p_2) satisfies

$$\theta_{2,3}(p_2) = \frac{p_2}{Q\,(\beta+2)}. \tag{52}$$

Then after substituting $\theta_{1,2}$, $\theta_{2,3}$ from (51), (52) into the general profit function (47), simplifying for $n = 2$, and using F.O.C. (see Appendix A.7.2 for S.O.C. and condition tests), we obtain a profit function similar to the profit function of the non-lock-in case in the imperfect foresight set-up, see (25):

$$\Pi = \left(1 - \frac{p_1}{Q}(1+\beta) + \frac{p_2}{Q}\beta\right)p_1 + \frac{1}{2}\left(\frac{p_1}{Q}(1+\beta) - \frac{p_2}{Q}\beta - \frac{p_2}{Q\,(\beta+2)}\right)p_2 + \frac{p_2}{Q}L_3 p_2 - \bar{B}Q^2. \tag{53}$$

Using F.O.C. (see Appendix A.7.2 for S.O.C. and the necessary conditions), we obtain equilibrium prices p_1^*, p_2^* and quality improvement Q^*:

$$p_1^* = \frac{4\lambda_2^2}{\Lambda_2^2\bar{B}},\ p_2^* = \frac{2\,(2+\beta)\,(1+3\beta)\lambda_2}{\Lambda_2^2\bar{B}},\ Q^* = \frac{\lambda_2}{\Lambda_2\bar{B}}, \tag{54}$$

where

$$\lambda_2 = (1+\beta)^2 - 2(2+\beta)L_3,\ \Lambda_2 = 6 + 11\beta - \beta^3 - 16(1+\beta)(2+\beta)L_3,$$

[22]Here $\theta_{1,2}(p_1^*, p_2^*)$ is $\theta_{1,n}$ calculated at $n = 2$, i.e., the user who is indifferent between switching every period at p_1 and switching every two periods at p_2; whereas, $\theta_{1,2}(p_2^*) = p_2^*/Q^*$ is $\theta_{n-1,n}$ calculated at $n = 2$, i.e., the user who is indifferent between switching every period at the "full" price p_2 and switching every two periods at p_2.

and these values are positive at every $\beta \in (0,1)$. The only necessary condition that is not guaranteed to hold is $\theta_{2,3}(p_2^*) \leq \theta_{1,2}(p_1^*, p_2^*)$, which is violated at $\beta \geq B_2 \approx 0.325448$. The violation of this condition means that the developer's profit decreases in p_1 in the entire non-lock-in region, whence it is optimal for the developer to switch to 3-lock-in (and often further).

Substituting the equilibrium values back into the profit function as well as (51), (52), we obtain

$$\Pi^* = \frac{\lambda_2^2}{\Lambda_2^2 \bar{B}},\ \theta_{1,2}^* = \frac{2\left(2+4\beta-\beta^2-\beta^3-4(1+\beta)(2+\beta)L_3\right)}{\Lambda_2},\ \theta_{2,3}^* = \frac{2(1+3\beta)}{\Lambda_2},$$

so that the number of high-end users $N_1 = 1 - \theta_{1,2}$ and the per-period average number of low-end users $N_2 = \frac{1}{2}\left(\theta_{1,2} - \theta_{2,3}\right) + \frac{p_2}{Q} L_3$ in equilibrium are the following.

$$N_1^* = \frac{(1+\beta)(2+\beta+\beta^2-8L_3(2+\beta))}{\Lambda_2},\ N_2^* = (1-\beta)\frac{\lambda_2}{\Lambda_2}.$$

Corollary 1 *Equilibrium prices per quality rations, obtained directly from (54), equal*

$$\frac{p_1^*}{Q^*} = \frac{4\lambda_2}{\Lambda_2},\ \frac{p_2^*}{Q^*} = \frac{2\left(2+\beta\right)\left(1+3\beta\right)}{\Lambda_2}.$$

As β is increasing, the equilibrium number of high-end users is also increasing. Intuitively, the present value of the future utility flow is growing in β, so it becomes more profitable for the consumers to use the upgrade option. This also leads the developer to decrease p_1 with respect to p_2, thus attracting more consumers to the upgrade version, until β reaches B_2, when all consumers buying every two periods are crowded out.

7.2.2 "Lock-in" equilibria

In the previous part, we examined non-lock-in equilibria where users switching every two periods were always present on the market. Now, we will generalize the case in a way that users switching every two periods (or even less frequently) are not present on the market. Taking into account the previous calculation, we derive the following general profit function:

$$\Pi = \left(1 - \frac{p_1}{Q}Y + \frac{p_2}{Q}Z\right)p_1 + \frac{1}{n}\left(\frac{p_1}{Q}Y - \frac{p_2}{Q}Z - \frac{p_2}{Q}X\right)p_2 + \frac{p_2}{Q}L_{n+1}p_2 - \bar{B}Q^2, \tag{55}$$

where n is the lowest frequency of users switching at the regular price present on the market. Then the "lock-in" equilibrium is the following (see Appendix A.7.3 for S.O.C. and the necessary conditions).

$$p_1^* = \frac{n^2\lambda_n^2}{\Lambda_n^2 \bar{B}},\ p_2^* = \frac{n^2(Y+nZ)\lambda_n}{2\Lambda_n^2 \bar{B}},\ Q^* = \frac{n\lambda_n}{2\Lambda_n \bar{B}},$$

where

$$\lambda_n = X + Z - nL_{n+1},\ \Lambda_n = 2nY(2X+Z) - Y^2 - n^2Z^2 - 4n^2YL_{n+1},$$

and these values are positive at every n and $\beta \in (0,1)$. The only necessary condition that is not guaranteed to hold is $\theta_{n,n+1} \leq \theta_{1,n}$, which is violated at $\beta \geq B_n$, where B_n is increasing in n with $\lim_{n\to\infty} B_n = \frac{1}{2}$, tabulated in Appendix A.8.1 . The violation of this condition means that the developer's profit decreases in p_1 in the entire n-lock-in region, so that it is optimal for the developer to switch to $(n+1)$-lock-in. This implies that if $\beta \geq \frac{1}{2}$, then it is optimal in our model for the developer to crowd out lower frequencies of switching infinitely, so no price discrimination equilibrium exists. However, if we introduce discounting into the developer's profit and take into account that the consumers start buying only after the quality accumulated reaches their participation constraint (instead of neglecting this due to no discounting, infinite horizon, and hence per-period optimization), then lock-in equilibria would exist for high values of β. The range of β at which price discrimination equilibria exist would also expand if we assume that some consumers have perfect foresight and others have imperfect foresight as described in section 9.2. It should be also noted that high values of β are not very appropriate to software.

Substituting the equilibrium values back into the profit function (55) as well as into the thresholds, we obtain

$$\Pi^* = \frac{n^2\lambda_n^2}{4\Lambda_n^2\bar{B}},\ \theta_{1,n}^* = \frac{n\left(Y(2X+Z) - nZ^2 - 2nYL_{n+1}\right)}{\Lambda_n},\ \theta_{n,n+1}^* = \frac{nX(Y+nZ)}{\Lambda_n},$$

so that the number of users, who are switching are

$$N_1^* = 1 - \theta_{1,n}^* = \frac{Y\left(n(2X+Z) - Y - 2n^2L_{n+1}\right)}{\Lambda_n},\ N_2^* = \frac{1}{n}\left(\theta_{1,n}^* - \theta_{n,n+1}^*\right) + \frac{p_2^*}{Q^*}L_{n+1} = (Y - nZ)\frac{\lambda_n}{\Lambda_n}.$$

7.2.3 Comparative statics for price discrimination equilibria

The following results can be derived from the equilibria above. Recall that non-lock-in is mathematically 2-lock-in. We assume that β is valid, i.e., $\beta \leq B_n$ for the given n.

Lemma 10 *If the developer uses n-lock-in price discrimination under perfect foresight, then the average number of switching users is decreasing in n and increasing in β, whereas the equilibrium quality change, both prices, both prices per quality change, and the developer's profit increase in both n and β. In addition, the equilibrium quality change converges for $\beta < \frac{1}{2}$ from below to the socially optimal level $Q_0 = \frac{1}{4\bar{B}(1-\beta)}$ as n increases.*

7.3 Equilibria comparison under perfect foresight

Here we compare perfect foresight equilibria under a single price and under price discrimination. As under imperfect foresight, the developer's profit is always higher for price discrimination since the developer always has a possibility to set prices $p_1 = p_2$. The following result, which exactly parallels the imperfect foresight outcome, can be shown to hold.

Lemma 11 *In the case of price discrimination, a developer's quality Q and both prices are higher than in the case of a single price, that is $p^* \leq p_1^* \leq p_2^*$, and the price per unit of the quality jump is higher in the case of price discrimination, that is $\frac{p^*}{Q} \leq \frac{p_1^*}{Q} \leq \frac{p_2^*}{Q}$, no matter the β and B. This result is independent from the perfect or imperfect foresight set-up.*

From this result, it follows that the developer's cost of a higher quality improvement in the case of price discrimination is more than fully compensated for by the increased prices.

In comparing the developer profit for a particular β and n, we see that the best strategy for the developer in our model is to set n close to infinity, which is the same result as in the case of imperfect foresight. This result again stems from our model set-up, where the developer is maximizing the average profit per period. Thus, the developer does not differentiate between the profit realized in the first period and the profit from a very distant period "close to infinity." Note that maximizing the average profit instead of the discounted profit across each period is a simplification to obtain an analytical solution, and the introduction of discounting would lead to a finite optimal lock-in depth.

8 Welfare analysis: perfect foresight set-up

A welfare analysis for the perfect foresight set-up is analogous to the imperfect foresight set-up from section 5. All related results from the imperfect foresight remain valid for the perfect foresight at $\beta = 0$. The following result can be shown to hold[23].

Proposition 9 *(i) Consumer surplus per period is higher for the single price developer than for the price discriminating developer for all relevant* β*, increases in* β *and decreases in* n *as* n*-lock-in is used. This is valid for the total consumer surplus as well as for the consumer surplus for high-end and low-end users separately.*

(ii) Under perfect foresight, the equilibrium per-period CS per buyer (total, high-end, and low-end alike) increases in β*, which applies to both a single-price monopoly and price discrimination for every* n*. Per-period CS per buyer for high-end users is higher for the single price developer than for the price discriminating developer for all relevant* β *and decreases in* n*. Per-period CS per buyer for low-end users is lower for the single price developer than for the price discriminating developer for all relevant* β *except for* $\beta <\approx 0.097773$ *at* $n = 2$ *and* $\beta <\approx 0.004777$ *at* $n = 3$*, and increases in* n*.*

(iii) Under perfect foresight, total per-period CS per buyer is lower for the single price developer than for the price discriminating developer whenever $n \geq 8$ *or* $\beta >\approx 0.289509$*. It increases in* n *whenever* $n \geq 8$ *or* $\beta <\approx 0.099358$*.*

Part (i) exactly corresponds to the one from the imperfect foresight case, and part (ii) is similar to what happens under imperfect foresight (recall that under imperfect foresight, CS per buyer for high-end users follows the same pattern as CS itself; whereas, the total CS per buyer and CS per buyer for low-end users increase in n and are higher under price discrimination except for $n = 2$ and $n = 3$). Thus, part (ii) confirms that the different relation between a single-price and price discrimination CS per buyer for low-end users for low n is just a mathematical property.

Part (iii) shows that as for the total CS per buyer, the effect for low-end users eventually prevails (mainly because the CS for high-end users decreases more quickly than the CS for low-end users). (See Appendix A.8.2 for the behavior of the CS per buyer when n is small.) As it is optimal for the developer to "squeeze" non-updating consumers who switch relatively frequently, the CS per buyer eventually increases in n.

As for social welfare, according to Lemmata 10 and 11 , the equilibrium quality is higher under price discrimination and converges from below to the socially optimal level as n increases. Therefore, the same

[23] These results were derived similar to those presented in Appendices A.5.2 and A.5.3 , but with perfect foresight equilibria and thresholds. As these results are often mathematically cumbersome, we opted not to include them explicitly but rather leave the *Mathematica* file available upon request.

properties apply to the equilibrium social welfare (for $\beta < \frac{1}{2}$, when price discrimination equilibria exist).

9 The comparison between imperfect and perfect foresight

9.1 The comparison of equilibria

As we have already noted, the outcome of the model at $\beta = 0$ is identical under both perfect and imperfect foresight. While the results at $\beta > 0$ are different as described below, there is one notable exception. Namely, the average number of users who switch to a new version every period is $\frac{1}{2}$ no matter whether they have perfect or imperfect foresight. Therefore, the number of switching users per period under a single price is independent from the discount factor β, and costs $\bar{B}$.

Before comparing other results, recall that while perfect foresight results are expressed in terms of the developer's quality cost efficiency $\bar{B}$ and the quality choice Q, imperfect foresight results were, for convenience, expressed in terms of $B = \bar{B}(1-\beta)^2$ and $q = \frac{Q}{1-\beta}$, so the corresponding change has been made. The following result can be proved.

Proposition 10 *Assume that $\beta > 0$, the price setting is fixed at either a single price or price discrimination by upgrades, and that n is fixed with $\beta \leq B_n$ in the discrimination case. Then the following results hold.*

(i) The equilibrium prices, qualities, price-to-quality ratios, and per period profits are lower under perfect foresight than under imperfect foresight, and the average per-period number of switching users is higher under perfect foresight than under imperfect foresight.

(ii) Under price discrimination, the equilibrium per-period consumer surplus, whether total or per buyer, is higher under perfect foresight than under imperfect foresight, counted for all users or for high-end and low-end consumers separately. However under the single price, the CS is higher under perfect foresight if β is not too high ($\beta < \frac{1}{2}$ is sufficient in all cases) but becomes higher under imperfect foresight as $\beta \to 1$.

(iii) The social welfare is lower under perfect foresight than under imperfect foresight.

A useful insight into part (i) is provided by Lemmata 7 and 9 , where we show that the indifferent users have a positive utility flow under perfect foresight, and that flow is zero under imperfect foresight. In other words, the developer cannot exert as much monopoly power against users with perfect foresight as against users with imperfect foresight: hence, the result above. This is especially evident as $\beta \to 1$ when the imperfect foresight values (other than number of users) tend to $+\infty$; whereas, the perfect foresight values, while also increasing in β, have finite limits.

For part (ii)—See Appendix A.8.3 for the thresholds under a single-price monopoly—the lower degree of monopoly power in equilibrium also results in a higher consumer surplus under perfect foresight when β is relatively low, including all values of β such that a price discrimination equilibrium under perfect foresight exists. However, at high values of β (which are improbable as noted in the discussion of price discrimination equilibria under perfect foresight), when the imperfect foresight equilibrium values tend to infinity, CS becomes higher under imperfect foresight.

Part (iii), while following from the fact that equilibrium qualities are lower under perfect foresight, might be surprising as a weaker monopoly is usually associated with higher welfare. However, in our model, while the developer cannot extract as much CS from the users as under imperfect foresight, neither are the users able to capture the entire difference, which results in a loss in welfare. Note that as it is optimal for the price discriminating developer to "squeeze" infinitely, the welfare will tend to its optimal value in both cases.

9.2 The generalization of results

We have shown that the comparative static results are qualitatively independent from using a perfect foresight or an imperfect foresight set-up. Thus, we can generalize the comparative static results on the market set-up where we have mixed users with imperfect and perfect foresight. If the type of foresight of the user is statistically independent from the user's quality sensitivity, with the latter uniformly distributed on $\langle 0, 1\rangle$, then all qualitative results of this paper remain valid.

10 Conclusion

In this paper, we have analyzed monopoly price discrimination based on upgrades designed for users who switch every period. It was done in quite a general set-up (infinitely durable products, an infinite number of users with different sensitivity to quality), and our restrictions on the model set-up were only based on the empirically observed patterns of the developers' behavior. We assumed the prices to be the same in every period since in real markets, we often observe prices for new versions of a product set at almost the same level as for previous versions. Our second restriction was that the quality improvement from one period to another was fixed.

As for our main result, we showed that price discrimination does not lead to lower prices for any user group in the market. This is caused by the character of price discrimination in our model, when the target group of price discounts is not the most price sensitive group of the users, but the discount is provided to the users with the highest frequency of switching, i.e., to the users with the highest sensitivity to quality.

The second key result is that a price discriminating developer accelerates software development more than a single-price developer, which could be perceived as beneficial for users, but on the other hand, the number of switching users in the case of a price discriminating developer is always lower. Even if we split users into two groups, those who switch every period and those who switch less frequently, we see that in the case of a price discriminating developer, the number of switching users is lower in both groups.

We showed that all those comparative static results are valid no matter whether users have perfect or imperfect foresight.

We can also look at the results from a different perspective: A price discriminating developer can more effectively "separate" high-end users from low-end users and charge them a higher price by offering a higher software quality while at the same time being aware that low-end users become less important from a revenue point of view. Since the developer knows that low-end users would switch to a new version from time to time anyway, the developer sets the price for them relatively higher than in the case of a single price, which ensures for the developer that those users who may consider switching to a new version every second period at the full price would rather switch every period for the lower upgrade price. This effect is more visible for perfect foresight users.

As for social welfare, a price discriminating developer generates (under plausible assumptions) higher per-period consumer surplus per user, but at the same time the number of switching users is lower, and the overall consumer surplus is lower as well. While this also indicates that the developer's monopoly power strengthens, the social welfare increases as well and approaches the socially optimal level in the limiting case.

The model used in the essay was designed for the software market, but it can be easily applied more widely whenever users are heterogeneous and a resale market is outlawed. Infinite durability of the product is not necessary, but the absence of product depreciation eliminates demand generated by product replacement due to physical obsolescence, which simplifies the analytical solution.

References

Anton, J., and Biglaiser, G. (2009): “Quality, Upgrades, and (the Loss of) Market Power in a Dynamic Monopoly Model,” ERID Working Paper, No. 9.

Banerjee, D.S. (2003): “Software Piracy: A Strategic Analysis and Policy Instruments,” *International Journal of Industrial Organization*, Vol. 21, pp. 97–127.

Belleflamme, P., and Peitz, M. (2010): *Industriall Organization: Market Strategies*, Cambridge University Press.

Echevarría, R.C. (2005): “Technological and Physical Obsolescence and the Timing of Adoption,” Fundación BVVA, Working Paper.

Ellison, G., and Fudenberg, D. (2000): “The Neo-Luddite’s Lament: Excessive Upgrades in the Software Industry,” *RAND Journal of Economics*, Vol. 31, pp. 253–272.

Fishman, A., and Rob, R. (2000): “Product Innovation by a Durable-good Monopoly,” *RAND Journal of Economics*, Vol. 31, pp. 237–252.

Fudenberg, D., and Tirole, J. (2000): “Pricing a Network Good to Deter Entry,” *Journal of Industrial Economics*, Vol. 48, pp.373–390.

Fudenberg, D., and Tirole, J. (1998): “Upgrades, Tradeins, and Buybacks,” *RAND Journal of Economics*, Vol. 29, pp. 235–258.

Fudenberg, D., and Villas-Boas, J. M. (2006): “Behavior-based Price Discrimination and Customer Recognition,” in Hendershott, T.J., ed. *Handbook on Economics and Information Systems*, Elsevier.

Inderst, R. (2008): “Durable Goods with Quality Differentiation,” *Economic Letters*, Vol. 100, pp. 173–177.

Katz, L., and Shapiro, C. (1998): “Antitrust in Software Markets,”Freedom Foundation Conference.

Mehra, A., and Seidmann, A. (2008): “Optimal Timing of Upgrades over a Software Product’s Life Cycle,” The Bradley Policy Research Center Working Paper, No. FR 08-22.

Matějka, F. (2010): “Rigid Pricing and Rationally Inattentive Consumer,” CERGE-EI Working Paper Series, No. 409.

Nahm, J. (2004): “Durable-goods Monopoly with Endogenous Innovation,” *Journal of Economics & Management Strategy*, Vol. 13, pp. 303–319.

Sims, C.A. (2003): “Implications of Rational Inattention,” *Journal of Monetary Economics*, Vol. 50(3), pp. 665–690.

Shapiro, C. (2000): “Competition Policy in the Information Economy,” in Hope, E., ed. *Competition Policy Analysis*, Routledge.

Stokey, N., (1979): "Intertemporal Price Discrimination," *Quarterly Journal of Economics*, Vol. 93, pp. 355–371.

Stokey, N., (1981): "Rational Expectations and Durable Goods Pricing," *Bell Journal of Economics*, Vol. 12, pp. 112–128.

Stole, L.A. (2007): "Price Discrimination and Competition," in Armstrong, M., and Porter, R., eds. *Handbook of Industrial Organization, Volume 3*, Elsevier.

Wang, Q.H., and Hui, K.L. (2005): "Technology Timing and Pricing In the Presence of an Installed Base," EconWPA Working Paper, No. 0512013.

Werden, G.J. (2001): "Network Effects and Conditions of Entry: Lessons from the Microsoft Case," *Antitrust Law*, Vol. 69, pp. 87–111.

Internet Sources: www.tomshardware.com, www.idg.com, www.zive.cz.

A Appendix

Most of the calculations in this paper were performed using *Mathematica* and other similar software. The *Mathematica* file is available upon request.

A.1 The approach used for S.O.C. verification

In all cases, the objective function is the developer's profit, which can be either a function of a single price and a quality, $\Pi(p, Q)$, or a function of two prices and a quality, $\Pi(p_1, p_2, Q)$, where Q is replaced with q under imperfect foresight. In the single-price case, the form of the Hessian used in this paper is

$$H = \begin{pmatrix} \Pi_{pp} & \Pi_{pQ} \\ \Pi_{pQ} & \Pi_{QQ} \end{pmatrix}, \tag{56}$$

and in the two-price case we use

$$H = \begin{pmatrix} \Pi_{11} & \Pi_{12} & \Pi_{1Q} \\ \Pi_{12} & \Pi_{22} & \Pi_{2Q} \\ \Pi_{1Q} & \Pi_{2Q} & \Pi_{QQ} \end{pmatrix}, \tag{57}$$

where subscripts '1' and '2' stand for the derivatives with respect to p_1 and p_2. In the proofs below, we immediately proceed to listing the principal minors of the Hessians.

A.2 The approach used for consumer surplus calculation

A recurrent task in this paper is to calculate the average per period consumer surplus (CS) as an infinite sum of CS of the consumers who switch every n or more periods, $n \geq 2$. Recall that the structure of the consumers' utility is $\theta Q - p$, so that the structure of added consumer utility is $\theta q - p$, where $q = \frac{Q}{1-\beta}$, and the range of the consumers who switch every n periods is given by $\theta_{n,n+1} < \theta < \theta_{n-1,n}$, where $\theta_{n,n+1}$ strictly decreases in n and $\lim_{n\to\infty} \theta_{n,n+1} = 0$. In addition, the usual form of $\theta_{n,n+1}$ is

$$\theta_{n,n+1} = \frac{p}{Q} X_n, \ \lim_{n\to\infty} X_n = 0$$

(here p can be p_2, and Q can be q for imperfect foresight, but see below). Then the average demand per period from the group in question equals

$$\sum_{m=n}^{\infty} \frac{1}{m} \left(\theta_{m-1,m} - \theta_{m,m+1}\right) = \frac{p}{Q} \sum_{m=n}^{\infty} \frac{1}{m} \left(X_{m-1} - X_m\right) = \frac{p}{Q} L_n.$$

The value L_n depends on the consumers' discount factor β, and at $\beta = 0$, it turns into the imperfect foresight D_n. The subscript n is usually omitted when $n = 2$.

CS for the group in question is given by $CS_{n+} = \sum_{m=n}^{\infty} CS_m$, where

$$CS_m = \int_{\theta_{m,m+1}}^{\theta_{m-1,m}} \left(\theta q - \frac{p}{m}\right) d\theta = \int_{(p/Q)X_m}^{(p/Q)X_{m-1}} \left(\theta q - \frac{p}{m}\right) d\theta = \frac{p^2}{Q} \left(\frac{X_{m-1}^2 - X_m^2}{2(1-\beta)} - \frac{X_{m-1} - X_m}{m}\right).$$

Note that the infinite sum of the "subtracted" term inside the parentheses is L_n, and as $\lim_{m\to\infty} X_m = 0$,

$$\sum_{m=n}^{\infty} \left(X_{m-1}^2 - X_m^2\right) = \lim_{m\to\infty} \left(X_{n-1}^2 - X_m^2\right) = X_{n-1}^2,$$

so that the final expression for CS is

$$CS_{n+} = \sum_{m=n}^{\infty} CS_m = \frac{p^2}{Q}\left(\frac{X_{n-1}^2}{2(1-\beta)} - L_n\right).$$

Two particularly important cases are $n = 2$ (recall that $X_1 = 1$ for all β), and $n = n+1$ (when the summation starts at $n+1$). Then

$$CS_{2+} = \frac{p^2}{Q}\left(\frac{1}{2(1-\beta)} - L\right), \quad CS_{(n+1)+} = \frac{p^2}{Q}\left(\frac{X_n^2}{2(1-\beta)} - L_{n+1}\right).$$

For imperfect foresight the outcome is (note that here $X_n = 1/n$)

$$CS_{2+} = \frac{p^2}{q}\left(\frac{1}{2} - D\right), \quad CS_{(n+1)+} = \frac{p^2}{q}\left(\frac{1}{2n^2} - D_{n+1}\right).$$

A.3 The single price model for imperfect foresight users

A.3.1 S.O.C. verification

The profit function is

$$\Pi = \left(1 - \frac{p}{q}\right)p + \frac{p}{q}Dp - Bq^2,$$

and the principal minors of the Hessian are $\frac{2}{q}(D-1)$ and $\frac{4B}{q}(1-D)$, so that H is negative definite as $q > 0$, $B > 0$, and $D \approx 0.355$. Therefore, the solution to F.O.C. is a maximum.

A.4 Imperfect foresight users and price discrimination

A.4.1 S.O.C. and validity in non-lock-in

We have to maximize the profit

$$\Pi = \left(1 - \frac{p_1}{q}\right)\cdot p_1 + \frac{p_2}{q}D\cdot p_2 - B\cdot q^2 - \left(\frac{p_2}{q} - \frac{p_1}{q}\right)\cdot\frac{1}{2}\cdot p_2$$

with respect to the conditions $p_1 \in \left\langle \frac{p_2}{2}, p_2\right\rangle$ and $\frac{p_1}{q} \le 1$. Our approach is to start with unconstrained optimization and check the conditions afterwards. (It happens that the conditions are satisfied, so there is no need to re-calculate.)

F.O.C. result in

$$p_1^* = 4\frac{(1-2D)^2}{(16D-7)^2 B}, \; p_2^* = 2\frac{1-2D}{(16D-7)^2 B}, \; q_e^* = \frac{2D-1}{B(16D-7)},$$

so that $\frac{p_1^*}{p_2^*} = 2 - 4D$, and $1 - \frac{p_1}{q} = \frac{3-8D}{7-16D}$, which satisfies our assumptions that $p_1 \in \left\langle \frac{p_2}{2}, p_2\right\rangle$ and $\frac{p_1}{q} \le 1$. The principal minors of the Hessian equal $-\frac{2}{q}$, $\frac{7-16D}{4q^2}$, and $-\frac{B(7-16D)}{2q^2}$, so that H is negative definite as $q > 0$, $B > 0$, and $D \approx 0.355$. Therefore, the solution to F.O.C. is a maximum.

A.4.2 S.O.C. and validity in lock-in

The profit function in n-lock-in is

$$\Pi = \frac{p_2}{nq_e}\left(p_1 + p_2 - np_2\psi_1(n)\right) + \left(1 - \frac{p_1}{q_e}\right)p_1 - Bq_e^2.$$

As in the non-lock-in case, there are the validity conditions $p_1 \in \langle \frac{p_2}{n}, \frac{p_2}{n-1} \rangle$ and $\frac{p_1}{q} \leq 1$, which are assumed to hold before being checked.

F.O.C. result in

$$p_1^* = \frac{n^2}{B} \frac{(n\psi_1(n) - 1)^2}{(4n^2\psi_1(n) - 4n - 1)^2}, \; p_2^* = \frac{n^2}{2B} \frac{n\psi_1(n) - 1}{(4n^2\psi_1(n) - 4n - 1)^2}, \; q_e^* = \frac{n}{2B} \frac{n\psi_1(n) - 1}{4n^2\psi_1(n) - 4n - 1},$$

and the validity conditions can be shown to hold using the properties of the polygamma function.

The Hessian is calculated as in (57), and the principal minors equal

$$-\frac{2}{q}, \; \frac{4n^2\psi_1(n) - 4n - 1}{n^2q^2}, \; -\frac{2B(4n^2\psi_1(n) - 4n - 1)}{n^2q^2},$$

so that H is negative definite as $q > 0$, $B > 0$, and it can be shown that $4n^2\psi_1(n) - 4n - 1 > 0$, $\forall n \geq 2$. Therefore, the solution to F.O.C. is a maximum.

A.4.3 The number of users under imperfect foresight and price discrimination

The number of upgrading users in equilibrium:

$$N_H^* = 1 - \frac{p_1^*}{q_e^*} = \frac{1}{2}\left(1 - \frac{1}{4n^2\psi_1(n) - 4n - 1}\right). \tag{58}$$

The number of users who buy less frequently than every period is:

$$N_L^* = \frac{p_2^*}{q_e^*} D_{n+1} - \frac{1}{n}\left(\frac{p_2^*}{nq_e^*} - \frac{p_1^*}{q_e^*}\right) = \frac{n\psi_1(n) - 1}{4n^2\psi_1(n) - 4n - 1}. \tag{59}$$

Summing N_H^* and N_L^*, we obtain the average demand per period N^*.

A.5 Imperfect foresight users: comparison and welfare analysis

A.5.1 Single price versus price discrimination

A comparison using the single-price equilibrium quality q^* by the price discriminating developer In the equilibrium comparison in the main part, we see that prices and quality are higher in the case of a price discriminating developer. Assume now that the developer has already set the quality at a level of the single price equilibrium $q^* = \frac{1}{8(1-D)B}$, derived in (18), and now is allowed to price discriminate. We now compare how these prices, denoted as p_1^{cs} and p_2^{cs}, differ from the single-price equilibrium value p^*. This comparison will allow us to separate the pure effect of enabling price discrimination and the effect of higher quality in the case of price discrimination.

Proposition 11 *Optimal prices for discrimination for a developer who has already set $q = q^*$ are*

$$p_1^{cs} = \frac{1}{2B} \frac{1 - 2D}{(7 - 16D)(1 - D)}, \; p_2^{cs} = \frac{1}{4B} \frac{1}{(7 - 16D)(1 - D)}.$$

Proof. If the quality change q is fixed in the non-lock-in price discrimination problem, F.O.C. in prices result in

$$p_1 = \frac{4(1 - 2D)}{7 - 16D} q, \; p_2 = \frac{2}{7 - 16D} q,$$

the ratio of the prices is $\frac{p_1}{p_2} = 2 - 4D$, so that the non-lock-in condition $p_1 \in \left\langle \frac{p_2}{2}, p_2 \right\rangle$ is satisfied, and the principal minors of the Hessian equal $-\frac{2}{q}$and $\frac{7-16D}{4q^2}$ so that the solution is a maximum. Substituting $q = q^*$ yields the result. ■

By comparing with the optimal single price $p^* = \frac{1}{16(1-D)^2 B}$, we immediately see that $\frac{p_1^{cs}}{p^*} \simeq 1.133911$, so that $p_1^{cs}$is higher than p^*, and since p_2^{cs} is higher than p_1^{cs}, both prices are higher in the case of price discrimination.

Corollary 2 *Enabling price discrimination while fixing q^* from the single-price equilibrium decreases the average number of users who switch to the new product, raises the price for both the upgrade and full versions, and increases the price per quality value, $\frac{p^*}{q^*} < \frac{p_1^{sc}}{q^*} < \frac{p_2^{sc}}{q^*}$.*

Proof. The result is proven by direct comparison. ■

Thus, the key result from the comparison of the single-price and price discrimination equilibria comes from two effects: enabling price discrimination and increasing equilibrium quality. Further, these two effects reinforce each other as both result in higher prices and a lower number of users.

A.5.2 Welfare analysis: single price developer

Recall that the equilibrium price and quality improvement are

$$p^* = \frac{1}{16\left(1-D\right)^2 B}, \, q^* = \frac{1}{8\left(1-D\right)B} \Longrightarrow \frac{p^*}{q^*} = \frac{1}{2\left(1-D\right)}.$$

The asterisk superscript denoting the equilibrium values is implied where needed.

Consumer surplus for high-end users equals

$$CS_H = \int_{\frac{p}{q}}^{1} (\theta q - p)\, d\theta = \frac{1}{2q}\left(p-q\right)^2 = \frac{1}{64}\frac{(1-2D)^2}{\left(1-D\right)^3 B} \approx \frac{0.0048941}{B}. \tag{60}$$

Consumer surplus per buyer for high-end users is then

$$\frac{CS_H}{N_H} = \frac{CS_H}{1-\frac{p}{q}} = \frac{1}{2}(q-p) = \frac{1}{32}\frac{1-2D}{\left(1-D\right)^2 B} \approx \frac{0.021778}{B}. \tag{61}$$

Consumer surplus for low-end users equals

$$CS_L = \frac{p^2}{q}\left(\frac{1}{2} - D\right) = \frac{1}{64}\frac{1-2D}{(1-D)^3 B} \approx \frac{0.016884}{B}. \tag{62}$$

Consumer surplus per buyer for low-end users is then

$$\frac{CS_L}{N_L} = \frac{CS_L}{\frac{p}{q}D} = p\frac{(1-2D)}{2D} = \frac{1}{32}\frac{(1-2D)}{\left(1-D\right)^2 DB} \approx \frac{0.061335}{B}. \tag{63}$$

Total consumer surplus equals

$$CS = CS_H + CS_L = \frac{1}{32}\frac{1-2D}{\left(1-D\right)^2 B} \approx \frac{0.021778}{B}. \tag{64}$$

Consumer surplus per buyer is then

$$\frac{CS}{N} = \frac{1}{16} \frac{1-2D}{(1-D)^2 B} \approx \frac{0.043556}{B}. \tag{65}$$

Finally, social welfare under imperfect foresight with a single-price developer equals

$$W = \frac{1}{2}q - Bq^2 = \frac{3-4D}{64B(1-D)^2} = \frac{3}{16} \frac{2\pi^2 - 15}{B(\pi^2-6)^2} \approx \frac{0.059344}{B}. \tag{66}$$

A.5.3 A welfare analysis: price discrimination

Here the general n-lock-in case is analyzed. Recall that non-lock-in is mathematically 2-lock-in. Recall that the equilibrium prices and quality change are

$$p_1^* = \frac{n^2}{B} \frac{(n\psi_1(n)-1)^2}{\gamma_4^2},\ p_2^* = \frac{n^2}{2B} \frac{n\psi_1(n)-1}{\gamma_4^2},\ q_e^* = \frac{n}{2B} \frac{n\psi_1(n)-1}{\gamma_4},$$

where $\gamma_m = mn^2\psi_1(n) - mn - 1$, note that $\gamma_m > 0$ for $n \geq 2$, and $m = 2, 3, 4$. The asterisk subscript for the equilibrium values is implied where needed.

Consumer surplus for high-end users equals

$$CS_H = \int_{\frac{p_1}{q_e}}^{1} (\theta q_e - p_1)\, d\theta = \frac{1}{2q_e}(p_1 - q_e)^2 = \frac{n}{4B} \frac{(n\psi_1(n)-1)\gamma_2^2}{\gamma_4^3}. \tag{67}$$

Consumer surplus per buyer for high-end users is then

$$\frac{CS_H}{N_H} = \frac{1-\gamma_4^{-2}}{32B}. \tag{68}$$

Consumer surplus for low-end users consists of the surplus for those who buy with the frequency of exactly n periods ($\frac{p_2}{nq_e} = \theta_{n,n+1} < \theta < \theta_{1,n} = \frac{p_1}{q_e}$) and for those who buy even less frequently. The former term equals

$$\int_{\theta_{n,n+1}}^{\theta_{1,n}} \theta q_e - \frac{p_2}{n} d\theta = \frac{1}{2} \frac{(np_1 - p_2)^2}{n^2 q_e},$$

and the latter equals

$$\frac{p_2^2}{q_e}\left(\frac{1}{2n^2} - D_{n+1}\right).$$

Substituting the equilibrium values yields

$$CS_L = \frac{n^2}{2B} \frac{(n\psi_1(n)-1)^2 \gamma_2}{\gamma_4^3}. \tag{69}$$

Consumer surplus per buyer for low-end users is then

$$\frac{CS_L}{N_L} = \frac{n\left(1-\gamma_4^{-2}\right)}{16B}. \tag{70}$$

Total consumer surplus equals

$$CS = CS_H + CS_L = \frac{1-\gamma_4^{-2}}{32B},$$

and total consumer surplus per buyer is CS/N^{D*}, where

$$N^{D*} = \frac{n(1+2n)\psi_1(n) - 2(1+n)}{4n^2\psi_1(n) - 4n - 1}.$$

Finally, social welfare under imperfect foresight when the developer uses price discrimination with n-lock-in equals

$$W = \frac{1}{2}q_e - Bq_e^2 = \frac{n}{4B}\frac{(n\psi_1(n) - 1)\,\gamma_3}{\gamma_4^2}.$$

A.6 Perfect foresight: single price developer

A.6.1 The user indifferent between switching every n and every $n+1$ periods

Assume that there is a single price p, the user purchased the product in the current period and now decides on whether to switch every n or every $n+1$ periods. As the user will buy the product $n(n+1)$ periods from the current one in both cases, the decision is based on (the NPV of) the utility added between the current period and the period $n(n+1)-1$ from now. Note that if a user buys a version T periods from now and keeps it for S periods, then the utility added over the periods from T to $T+S-1$ equals

$$U = \beta^T \left(\theta T Q \frac{1-\beta^S}{1-\beta} - p \right).$$

The user switching every $n+1$ periods buys in periods $n+1$, $2(n+1)$, $\ldots$, $(n-1)(n+1)$, and keeps every version purchased for $n+1$ periods, so that the utility added equals

$$U_{n+1}(\theta) = \theta \sum_{i=1}^{n-1} \beta^{i(n+1)} \left(i\,(n+1)\right) Q \frac{1-\beta^{n+1}}{1-\beta} - \sum_{i=1}^{n-1} \beta^{i(n+1)} p.$$

The user switching every n periods buys in periods n, $2n$, $\ldots$, $n \cdot n$, and keeps every version purchased for n periods, so that the utility added equals

$$U_n(\theta) = \theta \sum_{i=1}^{n} \beta^{i \cdot n} \left(i \cdot n\right) Q \frac{1-\beta^{n}}{1-\beta} - \sum_{i=1}^{n} \beta^{i \cdot n} p.$$

From these two, we can derive the indifferent user $\theta_{n,n+1}$ after algebraic transformations:

$$\theta_{n,n+1} = \frac{p}{Q}\frac{(1-\beta)^2}{n\,(1-\beta) - \beta\,(1-\beta^n)}.$$

As for an alternative decision process, when the user has the version which is n periods old and chooses between buying now and then every n periods and buying in the next period and then every $n+1$ periods, the outcome is the same as the utilities to compare, $\beta^{-n}U_n(\theta)$ and $\beta^{-n}U_{n+1}(\theta)$ respectively.

Note that at $\beta = 1$, the term $\frac{1-\beta^S}{1-\beta}$ is to be replaced with S, and then $\theta_{n,n+1} = \frac{p}{Q}\frac{2}{n+n^2}$. Also note that the threshold can be expressed as

$$\theta_{n,n+1} = \frac{p}{Q}\frac{1}{\sum_{m=1}^{n} m\beta^{n-m}},$$

so that $\theta_{n,n+1}$ decreases in β.

Recall that the corresponding threshold for imperfect foresight is $\frac{p}{nq} = \frac{p}{Q}\frac{1-\beta}{n}$, and the ratio of the two equals

$$\theta_{n,n+1}\left(\frac{p}{Q}\frac{1-\beta}{n}\right)^{-1} = \frac{n(1-\beta)}{n(1-\beta) - \beta(1-\beta^n)} \geq 1$$

(equality holds only at $\beta = 0$), so that the perfect foresight threshold is not lower than the imperfect foresight one. Therefore, the utility received by a user indifferent between switching every n and every $n+1$ periods, which is zero in the imperfect foresight case, is non-negative, and positive if $\beta > 0$, under perfect foresight.

A.7 Perfect foresight: discrimination by upgrades

A.7.1 The user indifferent between switching every period at the upgrade price and every n periods at the full price

Assume that the user purchased the product in the previous period, so two options are considered. First, the user can exercise the right to buy at the upgrade price p_1, and if this is optimal, then the same decision will be made in all subsequent periods. Second, the user may decide to wait thus losing his eligibility for the upgrade price, so that the user would wait $n-1$ periods and then switch every n periods at the full price p_2, where n is determined as in Appendix A.6.1 . The easiest way to derive the threshold is to compare infinite discounted added utility flows from these two options. Note that if a user buys at price p a version T periods older than the previously possessed, then the infinite added utility flow at the moment of purchase equals

$$U = \theta T q - p = \frac{\theta T Q}{1-\beta} - p.$$

The user switching every period from the current one inclusive at price p_1 has an added utility flow of

$$U_1(\theta, p_1) = \left(\frac{\theta Q}{1-\beta} - p_1\right)\left(1+\beta+\beta^2+\cdots\right) = \frac{\theta Q}{(1-\beta)^2} - \frac{p_1}{1-\beta}.$$

The user switching $n-1$ periods from the current one and then every n periods, always at p_2, has an added utility flow of

$$U_n(\theta, p_2) = \left(\frac{\theta n Q}{1-\beta} - p_2\right)\left(\beta^{n-1}+\beta^{2n-1}+\beta^{3n-1}+\cdots\right) = \beta^{n-1}\left(\frac{\theta n Q}{(1-\beta)(1-\beta^n)} - \frac{p_2}{1-\beta^n}\right).$$

From these two, we can derive the indifferent user $\theta_{1,n}$ after algebraic transformations:

$$\theta_{1,n} = (1-\beta)\,\frac{p_1(1-\beta^n) - p_2(1-\beta)\beta^{n-1}}{(1-\beta^n - n\beta^{n-1}(1-\beta))Q}.$$

Note that as $\beta \to 0$, $\theta_{1,n} \to \frac{p_1}{Q}$, which is the imperfect foresight value of this threshold, and as $\beta \to 1$, $\theta_{1,n} \to \frac{2(np_1-p_2)}{n(n-1)Q}$.

If n is optimal at price p_2, then $\theta_{1,n}nq - p_2 \geq 0$ as is shown in Appendix A.6.1 , so that $\theta_{1,n}q - p_1 \geq 0$.

A.7.2 S.O.C. and validity in non-lock-in

The profit function is

$$\Pi = \left(1 - \frac{p_1}{Q}(1+\beta) + \frac{p_2}{Q}\beta\right)p_1 + \frac{1}{2}\left(\frac{p_1}{Q}(1+\beta) - \frac{p_2}{Q}\beta - \frac{p_2}{Q(\beta+2)}\right)p_2 + \frac{p_2}{Q}L_3 p_2 - \bar{B}Q^2,$$

and the validity conditions are $\theta_{1,2}(p_2) \geq \theta_{1,2}(p_1,p_2) \geq \theta_{2,3}(p_2)$ and $1 \geq \theta_{1,2}(p_1,p_2)$, which are assumed to hold before being checked. F.O.C. result in

$$p_1^* = \frac{4\lambda_2^2}{\Lambda_2^2\bar{B}},\; p_2^* = \frac{2(2+\beta)(1+3\beta)\lambda_2}{\Lambda_2^2\bar{B}},\; Q^* = \frac{\lambda_2}{\Lambda_2\bar{B}},$$

where

$$\lambda_2 = (1+\beta)^2 - 2(2+\beta)L_3,\; \Lambda_2 = 6 + 11\beta - \beta^3 - 16(1+\beta)(2+\beta)L_3,$$

so that the conditions $\theta_{1,2}(p_2) \geq \theta_{1,2}(p_1,p_2)$ and $1 \geq \theta_{1,2}(p_1,p_2)$ hold for every β, and the remaining condition $\theta_{1,2}(p_1,p_2) \geq \theta_{2,3}(p_2)$ holds for $\beta \leq B_2 \approx 0.325448$.

The minors of the Hessian equal

$$-\frac{2(1+\beta)}{Q}, \frac{\Lambda_2}{4Q^2(2+\beta)}, -\frac{\bar{B}\Lambda_2}{2Q^2(2+\beta)},$$

so that the solution is a maximum.

A.7.3 S.O.C. and validity in lock-in

The profit function is

$$\Pi = \left(1 - \frac{p_1}{Q}Y + \frac{p_2}{Q}Z\right)p_1 + \frac{1}{n}\left(\frac{p_1}{Q}Y - \frac{p_2}{Q}Z - \frac{p_2}{Q}X\right)p_2 + \frac{p_2}{Q}L_{n+1}p_2 - \bar{B}Q^2,$$

and the validity conditions are $\theta_{n-1,n} \geq \theta_{1,n} \geq \theta_{n,n+1}$ and $1 \geq \theta_{1,n}$, which are assumed to hold before being checked. F.O.C. result in

$$p_1^* = \frac{n^2\lambda_n^2}{\Lambda_n^2\bar{B}}, p_2^* = \frac{n^2(Y+nZ)\lambda_n}{2\Lambda_n^2\bar{B}}, Q^* = \frac{n\lambda_n}{2\Lambda_n\bar{B}},$$

where

$$\lambda_n = X + Z - nL_{n+1}, \Lambda_n = 2nY(2X+Z) - Y^2 - n^2Z^2 - 4n^2YL_{n+1},$$

so that the conditions $\theta_{n-1,n} \geq \theta_{1,n}$ and $1 \geq \theta_{1,n}$ hold for every β, and the remaining condition $\theta_{1,n} \geq \theta_{n,n+1}$ holds for $\beta \leq B_n$, where.B_n is tabulated in Appendix A.8.1 .

The minors of the Hessian equal

$$-\frac{2Y}{Q}, \frac{\Lambda_n}{n^2Q^2}, -\frac{2\bar{B}\Lambda_n}{n^2Q^2},$$

so that the solution is a maximum.

A.8 Numeric simulations

A.8.1 Functions $L(\beta)$ and B_n

The calculations were performed in *Mathematica*, where $L(\beta)$ was interpolated with step 10^{-3}.

β	$L(\beta)$
0	0.355066
0.01	0.355942
0.05	0.359394
0.1	0.363599
0.2	0.371653
0.3	0.379247
B_2	0.381107
0.4	0.386393
0.5	0.393105
0.6	0.399388
0.7	0.405247
0.8	0.410682
0.9	0.415691
0.95	0.418033
0.99	0.419827
1	0.420264

n	B_n
2	0.325448
3	0.377935
4	0.415119
5	0.441850
6	0.461048
7	0.474634
8	0.483997
9	0.490227
10	0.494211
11	0.496661
12	0.498115
13	0.498954
14	0.499427
16	0.499833
18	0.499953
20	0.499989

As for B_n, it increases in n, and its limit can be shown to be 0.5.

A.8.2 CS per buyer under perfect foresight for small n

In the following table, we report the ranges of β in which the behavior of the total price discrimination equilibrium CS per buyer is different from the general case, i.e., it is either lower than under single-price monopoly or decreases in n. No such behavior occurs for $n \geq 8$.

n	$CS_{PD} < CS_{SP}$	$CS_n > CS_{n+1}$
2	$[0, 0.220619)$	$(0.099358, B_2]$
3	$[0, 0.270516)$	$(0.211142, B_3]$
4	$(0.010498, 0.289509)$	$(0.303669, B_4]$
5	$(0.045734, 0.285588)$	$(0.379895, B_5]$
6	$(0.078875, 0.262546)$	$(0.442822, B_6]$
7	$(0.117158, 0.222414)$	–

A.8.3 CS comparison under a single-price monopoly

In the following table, we report the maximal values of β until which CS is higher under perfect foresight than under imperfect foresight when the developer charges the same price to all consumers. Note that CS is higher under perfect foresight in all cases when $0 < \beta < \frac{1}{2}$. The threshold for all users is the same because the number of switching users is the same ($N^* = \frac{1}{2}$) in both cases.

	All users	High-end users	Low-end users
CS	0.733526	0.797919	0.699392
CS per buyer	0.733526	0.875735	0.569339

Intellectual Property Rights Protection and Enforcement in a Software Duopoly[1]

(Coauthored with Krešimir Žigić)

Abstract

We study the economic impacts of the interaction between a regulator's Intellectual Property Rights (IPR) protection policy against software piracy on the one side and the forms of IPR protection that software producers may themselves undertake to protect their intellectual property on the other side. Two developers, each offering a variety of different quality, compete for heterogeneous users who choose among purchasing a legal version, using an illegal copy, and not using a product at all. Using an illegal version violates IPR and is thus punishable when disclosed. If a developer considers the level of piracy as high, he can either introduce a form of physical protection for his product or introduce a protection in the form of restricting support and other services to illegal users. The quality of each developer's product is exogenously given, and the developers compete in prices. We examine the above issues within the framework of Bertrand and Stackelberg competition while the monopoly set-up serves as a point of reference.

[1]All errors remaining in this text are the responsibility of the authors.

1 Introduction

During the last two decades violating Intellectual Property Rights (IPR) emerged as an important and hot economic and political issue since most of the world brands face a problem of illegal imitation of their products. The violation of IPR poses a threat to a wide range of products - from fashion such as Etro or Vuitton to Intel chipsets or Yamaha motorbikes. According to the WCO (World Customs Organization), 7% ($ 512 billion) of world trade takes place with fake merchandise[2] and most customers buy fake products unknowingly.

Barely anybody in Moscow believes that a $20 Rolex watch from a stallholder is an authentic one, but, on the other hand, hardly anyone assumes that a drug at a pharmacy, or car spare parts in dealer service centers are fake. In some cases, even professionals have difficulties to identify a particular product as a fake.

The key factor contributing to the creation of illegal imitations are low costs and low technical requirements. Based on such a view, the natural leaders among illegal imitations are "information" products or what are known as digital content products—software, movies, music, or e-books[3]. These products have two idiosyncratic attributes: imitations are 100% identical to the original and costs of copying are negligible. According to the report of the Business Software Alliance, the share of software that is pirated climbed to 41% of total units installed in 2008 and the global loss exceeded $50 billion[4]. Even in the US, where the rate of illegal usage is the lowest, it amounts to 20%, while in Western Europe about one-third of installed software is used illegally. The top of the list with 80% and more of illegal software installed is occupied by Georgia, Pakistan, Indonesia, and China[5].

The expansion of DVD burners accompanied with the penetration of broadband internet does not only increase the opportunity for illegal copying[6], but also eliminates mass illegal producers from market. Illegal copies are, nowadays typically made (installed) by the end users themselves who do it wittingly and only for themselves[7]. This attribute changes the essentials of the fight against IPR violation. While, say, in pharmaceuticals, luxury goods, or electronics markets, end users might be often perceived as victims, in "information" markets, end-users of illegal copies are predominantly the ones that actually carry out IPR violation[8]. Thus, the fight in digital content markets is now aimed mainly against end users (meaning both retail and corporate users)[9].

In this essay, we focus on such digital content markets (like the software market) where only the end

[2]Two thirds of illegal imitations come from China, the rest mainly from Ukraine, Russia, Vietnam, the Philippines, according to the WCO.

[3]For information about mp3, movie, or e-book protection and their illegal copying see www.ifpi.org, www.riaa.com, www.pro-music.org.

[4]According to the IDC Global Software Piracy Study.

[5]see also The Economist, May 16th, 2009.

[6]Most of the illegal copies of digital content are easily accessible using P2P networks (direct connect, torrent trackers) or data sharing (Rapidshare). Note that easy downloading could be accompanied by relatively complicated installation/usage of illegal versions.

[7]We omit in the essay the problem of the black market with DVDs/CDs or software in the suburbs. These kinds of piracy experienced a boom one decade ago and are now strongly declining especially in developed countries.

[8]However, companies try to distinguish between intentional piracy and the unconscious usage of an illegal version, e.g., Microsoft replaces fake versions with legal ones to users who bought a fake version of its software in good faith.

[9]A well-known examples aimed at end-users is suing students at US/EU universities for sharing software on university servers. Note that these actions are often accompanied with legal actions against the means of sharing e.g. closing Napster as the first famous case or the current hot suit against torrent tracker The Pirate Bay with the intention to close it.

users violate property rights. More specifically, we study the strategic interactions among software developers that may undertake various forms of product protections (developer IPR protection) and also analyze the impact of regulator (or government) IPR protection on such developer IPR protection. In particular, we put forward a dynamic two-stage duopoly model, where the last stage competition is in prices, and where each developer competes for users with different price sensitivity on the same market. That is, we rely on a quality competition model (see, for instance, Shaked and Sutton, 1984, and Tirole, 1988[10]). In the first stage of the game, each developer has an option to choose a particular form of IPR protection. The government[11] only sanctions illegal use of the product by means of imposing a penalty so those end users who illegally appropriate the software will be punished, if discovered.

We consider the two most common forms that the developers use to protect their products: a) decreasing product value to illegal users by, say, eliminating updates in antivirus or tune-up utilities[12] and b) physical product protection by means of special CDs (or encryption against cracking) like in games, where copies created on a standard DVD burner do not work anymore. These measures are known in the literature as "technical protection measures" which enhance copyright enforcement (see Scotchmer, 2004, for an excellent survey on this topic) and one of the most known ways of protecting the digital content is DRM (Digital Rights Management) system[13].

To capture the regulator's role in a simple manner, we assume that imposing a penalty is the only instrument for reducing or eliminating the illegal usage of the product that is under copyright protection. The government's reliance on taxes and subsidies as an instrument of IPR protection are not considered very realistic in the given context and is thus assumed away in the further analysis. Moreover, in order to focus on the impact of the penalty in different market configurations and its interaction with the private developers' enforcement, we do not assign the regulator a particular objective function such as maximizing social welfare, but we rather look at the penalty as exogenous and discuss its impact on developer equilibrium values and on the developers' choice of the form of IPR protection.

In most countries, governments are responsible for the creation of a legal environment for IPR enforcement and prevention from piracy. Nevertheless, a government's objective does not in general coincide with the developers'. First, if the costs of copying are negligible, the more users use the product, the higher social welfare is. Moreover, the original product can be an inspiration for other developers and its wide spreading may raise further product development and consequently, social welfare. On the other hand, in an environment where a product could be freely copied, the producers' incentives to develop new products

[10]Shy (1999) addresses the same problem using a Hotelling-type spatial competition model.

[11]In the essay, we do not distinguish the role (objective) of government from the regulator's role or from the role of private authorities executing any monitoring as e.g. the RIAA (Recording Industry Association of America), the MPA (Motion Picture Association), the BSA (Business Software Alliance).

[12]Illegal versions of some antivirus software, e.g., Symantec Antivirus, do not update their installed databases of viruses and thus the PC is more vulnerable in the case of the latest virus attack, or tune-up utilities do not update their internal list of supported problems, so some new errors cannot be corrected.

[13]DRM is an umbrella term for various technologies that limit the usage of digital content in an unintended way by the developer. DRM is used by a lot of major content providers such as Microsoft, Sony, Amazon, or Apple. DRM is sometimes considered a controversial approach to protecting the IPR since it often restricts the usage ways beyond the copyright laws (e.g., not only against illegal copying, but even the legal usage, such as using the legally bought e-book on only one device). DRM technologies, however, were effectively implemented in selected cases as, e.g., in the case of Apple (iTunes). Nowadays most content providers experiment with DRM-free alternatives mainly in music. In movies or e-books, DRM is still quite used. See Belleflamme and Peitz (2010).

are suppressed[14]. Thus, the regulator activity in setting IPR protection, exerting monitoring, and the scale of penalty for users convicted from illegal usage usually balances the trade-off between the dissemination of knowledge and products on the one side, and preserving the incentives to innovate on the other side. In setting the level of IPR protection, a government may favour one of the developers, e.g., in the case of a domestic dominant developer competing with a foreign developer, the government may, for instance, adapt IPR enforcement to favour the domestic developer or vice versa[15]. Thus, to introduce explicitly the regulator's objective function, we would have to put "more structure" into our model. As already noted, the choice of the optimal level of government IPR (or the optimal expected penalty) is out of the scope of our analysis. We, however, briefly discuss in the conclusion the possible extensions of our analysis to normative issues.

Since the legal environment as well as the regulator's activity are publicly observable, users can estimate the probability of being caught and then convicted for copyright violation and so correctly calculate the expected size of the penalty. Thus, if a user decides to use an illegal version, he can evaluate the expected penalty (EP), which can be considered as the cost of illegal usage.

The software market may distinguish itself from other digital content markets due to potentially high Network Effects (NEs) coming from software usage. NEs mean that increasing the base of users by, say, allowing the copying of a product to some other users, raises the utility of all users and thus adds extra value to the product. We, on the other hand, consider NE unimportant, but we nevertheless briefly discuss how NE can be easily incorporated in our set-up (see section 2.5).

It is important to stress at the outset that our approach is a bit different from the current literature on software piracy. To put our analysis into context, we follow the very recent comprehensive and influential survey of digital piracy by Belleflamme and Peitz, 2010 (see also Peitz and Waelbroeck, 2006). According to this, our approach belongs to the i) end-user piracy models that ii) includes the competitive effects meaning that there are two producers of substitutable and piratable digital products that directly compete with each other (see Belleflamme and Peitz, 2011, p. 20).

As clearly seen from the Belleflamme and Peitz (2011), there are indeed only a few articles that deal with the positive and normative issues of digital piracy while explicitly modeling direct firm competition. Moreover, all of these papers, in general, rely on the notion of horizontal product differentiation. The pioneering article seems to be the one of Shy and Thisse (1999), who analyze piracy in the Hotelling-type duopoly competition where users have exogenous preferences for a particular developer. They show that a developer's decision to introduce protection against illegal copying depends mainly on the NEs, and that under strong NEs, each developer decides not to implement protection in order to make his software more attractive and to raise the users base. Jain (2008) builds upon the model of Shy and Thisse (1999) and assumes that firms can choose a level of IPR protection so that only a proportion of consumers with low product valuations (who are, by assumption, the only consumers interested in copying) can copy its product.

[14] There are other effects, such as tax losses, raising unemployment etc., which could be studied separately.

[15] For illustration, we could use a comparison among countries that have strong developers (e.g., the US) and quite a severe protection of IPR with countries where no strong local developers exist, e.g., Finland, Sweden, or Norway, and their protection of IPR is moderate and more "open."

In the absence of network effects, Jain shows that in such a set-up piracy can change the structure of the market and, thereby, reduce price competition between firms. The reason is that copying by low, more price-sensitive types enables firms to credibly charge higher prices on the segment of consumers that do not copy. Furthermore, this positive effect of piracy on firms' profits can sometimes outweigh the negative impact due to lost sales. So, even in the absence of network effects, firms may prefer weak copyright protection in equilibrium.

Finally, there is a recent paper by Minnitti and Vergari (2010), who also rely on the Hotelling differentiated-product duopoly framework. They, however, deal with a rather specific form of piracy like a private file sharing community and study how its presence affects the pricing behavior and profitability of producers of digital products.

It is also important to note that digital developers' competition can also occur in a multi-product framework, where piracy can generate a kind of indirect competition between horizontally differentiated digital products as demonstrated by Belleflamme and Picard (2007). They show how the copying technology displaying increasing returns to scale can create an interdependence between the demands for digital products that would be unrelated otherwise. Moreover, the underlying demand is, much like in our approach, obtained in a vertical differentiation manner. However, the vertical differentiation does not, like in our set-up, arise from the different quality levels of the developers but from the existence of original and copied digital products in a market where the originals are assumed to be always of higher quality than the copies, and thus, all consumers unambiguously prefer the original product over the copy. In this set-up Belleflamme and Picard (2007) study how piracy affects prices and profits and, interestingly enough, they show that depending on the parameters of the model, prices can be either strategic substitutes or strategic complements. If the fixed cost of copying is low enough, there is no equilibrium in pure strategies. Firms may then randomize between several prices, leading to price dispersion.

Following the approach of Belleflamme and Picard (2007), Choi, Bae, and Jun (2010) use a Hotelling horizontal differentiation model as well and analyze the situation in which also the interdependence between the firms stems from their strategies against piracy rather than from direct competition on prices. They, like we do, consider the IPR efforts of the firms to be endogenous variables and study the interaction between public and private protection against piracy.

Besides the different focus (direct versus indirect competition), the other key difference between ours and the set-up of Belleflamme and Picard (2007) and Choi, Bae, and Jun (2010) is that in their settings the original products have the same quality, while in our set-up, the original products are vertically differentiated and thus have distinct qualities to begin with. Moreover, since we focus on the software market, we do not allow for a different copying technology as it is usual in the case with multiple, initially independent digital products. Thus, the cost of consuming illegal copies is constant in our setting, while it may be decreasing with the number of different originals copied in the settings of Belleflamme and Picard (2007) and Choi, Bae, and Jun (2010).

Finally, there are by now numerous scholarly articles that deal with the issue of digital piracy and private or public IPR protection in the monopoly set-up (see, for instance, Banerjee, 2003; King and Lampe, 2003; Kúnin, 2004; Takeyama, 2009). Banerjee (2003) demonstrates that the socially optimal level of IPR protection differs from a monopoly developer's optimum and stresses the role of NEs. King and Lampe (2003) show that the monopoly allows illegal users in the case when the network effect is present, while Takeyama (2009) shows that under asymmetric information about product quality, the copyright has to be imperfect in order to avoid adverse selection. Kúnin (2004) provides an explanation for why a software manufacturer may tolerate widespread copyright infringement in developing countries and often even offer local versions of their software. He showed that if NEs are present and there is an expected improvement in copyright, then software manufacturers enter the market even if they incur losses in the beginning when copyright enforcement is weak. For a deeper and systematic review of the literature on the piracy of digital products, the interested reader is advised to look at the two excellent and comprehensive surveys in Peitz and Waelbroeck (2006) and Belleflamme and Peitz (2011).

As already mentioned above, we focus our analysis on the developers' strategic interactions and the way how the size of the expected penalty affects market structure, market coverage, and the developers' IPR protection. We especially put the emphasis on the latter, meaning on the interaction between the government's (or public) and the developer's (or private) IPR protection. We show that when developers restrict services to illegal users (section 3), the government's and the developers' IPR are always substitutes in a sense that for the given developers' optimal protection, the public IPR protection could be substantially lower (compared to the situation with no private IPR protection) in order to fully eliminate illegal usage. Moreover, the government can by choice of its IPR protection (that is, via the size of the expected penalty) affect the market configuration and market coverage since the height of the expected penalty has an effect on equilibrium prices and profits and thus on the toughness of price competition. For instance, for the size of the expected penalty that falls between two prices, there might occur a market configuration with two unconnected segments of legal users. In this case, the high quality developer serves the upper part of the market and earns (constrained) monopoly profit, while the lower quality developer serves the lower tail of the market. In the middle of these two segments, there is a "buffer" composed of illegal users[16]. If on the other hand, the government sets penalty rather low so that both prices[17] are bigger than the expected penalty, then direct duopoly competition might be restored.

As for the situation when developers rely on the physical protection of their software (section 4), the government's and the developers' IPR could be either substitutes or complements in a sense that a marginal increase in the expected penalty can either decrease the optimal developer protection (implying substitutes) or increase it (implying complements). Moreover, the size of the expected penalty is a key in determining whether none, one, or both developers would introduce private IPR protection. If the expected penalty is so low that both prices exceed it, then both developers would introduce protection, and a small increase in the expected penalty would reinforce the developer protection indicating the complementarity of the two forms of protection. If on the other hand, the expected penalty exceeds the price of the low quality good but is still lower than the price of the high quality good, then only the high quality developer would introduce IPR

[16]A necessary condition for this case to arise is that an illegal copy of a high quality product has a higher value for users than the quality of a legal, lower quality product.

[17]Meaning equilibrium prices in standard Bertrand competition, where illegal usage is eliminated.

protection. Now, however, a marginal increase in the expected penalty would decrease the optimal developer protection implying substitutability between the two forms of protection.

The structure of the essay is the following: In the second chapter, we put forward our set-up that comprises three basic types of market conduct: monopoly, Bertrand duopoly, and the Stackelberg leader-follower model. We then analyze how the level of EP affects the developers conduct and market structure in the simplest case when there is no product protection from the developers' side whatsoever. In the third chapter, we allow the developers to introduce product protection in the form of a lower product value for illegal users by disabling them access to additional services. In the fourth chapter, we investigate the economic impacts of another form of product protection in which a developer implements a physical protection for his product. In both the third and fourth chapters, we study the effects of the particular form of developer's IPR protection within the three above mentioned market conducts. Chapter five concludes.

2 The basic model

We first analyze the cases where developers could eliminate the illegal usage of their products only by decreasing prices. Developers cannot introduce any product protection or restrict associated product services to the illegal users.

2.1 Model set-up

2.1.1 Industry set-up

Consider two developers A and B that compete in prices on a particular market and offer product varieties of different quality[18]. Developer A releases a product of quality q_A, while the quality of the second developer B is q_B and we assume, without loss of generality, in the rest of the essay that developer A offers higher quality ($q_A > q_B$). Product qualities q_A, q_B, in the whole essay are assumed to be exogenous and cannot be changed by developers[19] The unit variable costs are assumed to be constant and normalized to zero. One may think about developer A as an already established and known software producer that already operates on other markets. This fact is, in turn, reflected in the preferences of the consumers, who strictly prefer software A over software B if offered at the same price. Similarly, developer B can be thought of as a local developer offering lower quality. In other words, we assume that both developers already existed before meeting and competing on the market under consideration. Consequently, both developers are assumed to have already incurred set-up fixed costs and fixed costs associated with software development (R&D costs). These fixed costs are, from our perspective, general and not related to the developer's presence on the particular market under consideration, and therefore, we leave them out of the profit function. We, however, may allow for the fixed costs of entry to the particular market under consideration, so we denote as F_A and F_B these entry or set-up costs respectively (sinking these costs can be considered to take place at the first stage of the

[18]We will use the term "value" instead of "quality" when quality contains multiple dimensions.

[19]In the more elaborated versions of this kind of models, there is also a choice of qualities proceeding the pricing decision. In this case, it is standard to assume that the bulk of the costs of generating quality falls on fixed costs so that quality or R&D costs are in fact endogenously determined (see, for instance, Shaked and Sutton, 1982 and 1983; Kúnin and Žigić, 2006). For each case that we analyze, it should be clear how to relax the model and allow the developers to choose and compete in qualities too.

game). We will, however, omit these fixed costs from the profit functions for the purpose of transparency and assume that the developers' profits are positive net of these costs.

To summarize, we simply assume that:

1. Initially, both developers A and B already exist with established quality levels of their respective varieties.

2. The focus is on a particular software market, which is not interrelated with the other markets on which developers may operate ("segmented market hypothesis").

Thus, it is convenient to think that two developers compete (or may compete) in some third market (that is, the market that is not their home market). An important implication of these two assumptions is that in our set-up one or even both developers may not be active in the market under the considerations. The reason for this is that due to the absence of the developers' own IPR protection and the possible lack of IPR protection by the side of the regulator, it may not be profitable for the developer(s) to operate in the market under considerations. We, however, assume that even if a developer does not enter the market, the users are still able to obtain an illegal version via copying. This, in turn, makes entry deterrence not viable. We use a sub-game perfect equilibrium as a solution concept throughout this paper in all multi-period games under considerations.

2.1.2 The regulator's role

We introduce a very simple regulator whose role is limited to monitoring software usage and to the penalization of those users, who use products illegally and are disclosed. The probability of being caught using an illegal version is the same for all users, and the level of the penalty is fixed. The penalty and the probability of being caught is known and independent on used product and product prices[20], thus all users and both developers could calculate the expected penalty for using an illegal version, that we denote as X[21]. Moreover, while we implicitly assume that the regulator choice of optimal IPR is governed by an underlying objective function like the maximization of social welfare, we do not explicitly study the optimal choice of EP since we focus on the forms of the developers' IPR protection and their economic implications[22]. Thus, the whole regulator's framework[23] is very simple in our model and translates into one parameter: expected penalty X for illegal users.

2.1.3 The users' set-up

There is a continuum of risk neutral heterogenous users on a particular market under the considerations that differ in their personal value of product quality q_i, captured by parameter θ, where θ follows a uniform

[20] The penalty in the real economy is dependent on the price of software as well as the social impact from such behavior and real losses.

[21] We can expect that in a real economy, the fines for students compared to business companies differs as well as the regulator's monitoring rate among students/households is lower than in business. The extension of the model for different X for each product (different users) would lead to technical complication, that from our focus would add only limited further insight.

[22] For instance, if the government maximizes social welfare, we would need to know which of the developers is the domestic one and which is not in order to write down the objective functions. While these considerations are interesting per se, they are not the focus of the essay. For the analysis of the optimal IPR from the side of the regulator, see for instance Žigić (2000).

[23] Including government behavior, legal environment, and the real execution of property rights.

distribution over the interval $(0, \bar{\theta})$. Following Tirole (1988), utility for a user θ from consuming product q_i with price p_i and with expected penalty X from illegal usage is the following[24]:

$$U_\theta(p_i, q_i, X) = \begin{cases} \theta q_i - p_i & \text{... if a user buys software} \\ \theta q_i - X & \text{... if a user uses an illegal version} \\ 0 & \text{... if a user does not use the software at all} \end{cases} \tag{71}$$

From the users' utility function, we immediately see that for $p_i \neq X$, there are two groups of indifferent users. The users, who are indifferent between buying a product and not using it at all $(\theta q_i - p_i = 0)$, denote them as $\theta_{0A} = \frac{p_A}{q_A}$ (respectively $\theta_{0B} = \frac{p_B}{q_B}$) , and users who are indifferent between illegal usage and not using it at all $(\theta q_i - X = 0)$, denote them as $\theta_{0P} = \frac{X}{q_i}$.

2.2 Monopoly

We start with the simplest set-up, where only developer A with a product of quality q_A is present on the market, and there is perfect IPR enforcement. This set-up means that the expected penalty X is higher than monopoly the equilibrium price for the legal product p_M^* (72), and thus, nobody uses the product illegally. Developer A maximizes profit without any restriction, and from utility function (71), we see that $\theta_{0A} = \frac{p_M}{q_A}$ is the user indifferent between buying software and not using it at all. That leads to the following demand for product A:

$$D_M(p_M, q_A) = \bar{\theta} - \theta_{0A} = \bar{\theta} - \frac{p_M}{q_A},$$

so that the monopoly equilibrium price p_M^* and profit $\pi_M^* = p_M^* D_M(p_M^*, q_A)$ are:

$$\pi_M^* = \frac{1}{4} q_A \bar{\theta}^2, \; p_M^* = \frac{1}{2} \bar{\theta} q_A. \tag{72}$$

With such a price, developer A captures half of the market, $\frac{1}{2}\bar{\theta}$ (see Figure 8).

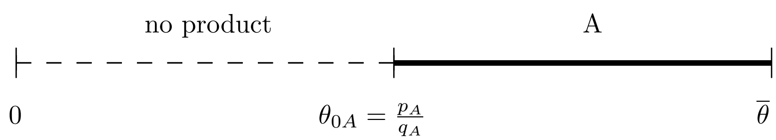

Figure 8: Monopoly market

Now for a while, assume that developer A operates on the monopoly market where IPR enforcement is lower than would be desirable for him, that is, $X < p_M^* = \frac{1}{2}\bar{\theta} q_A$. In this case, users compare the expected penalty X with price p_M, so all users with $\theta \geq \frac{X}{q_A}$ prefer illegal software usage. The only possibility for developer A to capture some legal users on the market is to lower the price to the level $p_M^* = X$. This is the case when developer A has to adjust the price to the level "set" by the regulator (that is, to the the level of X); otherwise, the developer is out of the market. Under such a situation, the regulator could effectively influence the developer's price p_M, which results in the following profit for developer A:

$$\pi_M^* = \left(\bar{\theta} - \frac{X}{q_A} \right) X, \tag{73}$$

[24]To avoid the problem of a fully satiated market, we do not follow distribution over (θ_l, θ_h) as in Banerjee (2003). Allowing the presence of users with $\theta \to 0$ ensures that for any price $p > 0$, there is a group of users who do not consume any product.

and the market share that the developer gains is $\left(\bar{\theta} - \frac{X}{q_A}\right)$, which is higher than $\frac{1}{2}\bar{\theta}$ from equilibrium (72).

2.3 Bertrand Competition

After the analysis of the basic case with a monopoly developer, consider now Bertrand competition in prices between developers A and B. In this case, low X might not be a constraint only for developer A but even for developer B. Thus, three basic cases of competition exist based on the presence of developer A in the market and on the level of the expected penalty X with respect to equilibrium prices p_A^o, p_B^o, where these prices come from pure Bertrand competition in (74):

1. $p_A^o, p_B^o \leq X$ ... High X: none of developers are limited in price setting by the level of X.
2. $\frac{X}{q_A} \leq \frac{p_B^o}{q_B} \leq \frac{p_A^o}{q_A}$ ... Low X: both developers are limited in price setting by the level of X
3. $\frac{p_B^o}{q_B} \leq \frac{X}{q_A} \leq \frac{p_A^o}{q_A}$ ... Medium X: only developer A is limited in price setting by the level of X

Notation 4 *In indexing indifferent users, P will always refer to illegal usage, 0 to not using any product at all, and A, B always refer to using (legally) the product A, B respectively. Moreover, we follow the rule that the first index refers to a user "on the left-hand side" and the second index will refer to a user "on the right-hand side." For instance, θ_{PB} means that a user with a lower θ than θ_{PB} uses the product illegally, while a user with θ higher than θ_{BP} legally uses product B. As for θ_{BP}, the same applies the other way around.*

2.3.1 Case 1: Bertrand competition under a high expected penalty $p_A^o, p_B^o \leq X$

This first basic case, where all piracy is eliminated, corresponds to the pure Bertrand competition[25], where both developers could freely compete in prices. Denote a user who is indifferent between product A and B as $\theta_{BA} = \frac{p_A - p_B}{q_A - q_B}$. Then profit functions for both developers are:

$$\pi_A = \left(\bar{\theta} - \theta_{BA}\right) p_A, \ \pi_B = \left(\theta_{BA} - \theta_{0B}\right) p_B.$$

This situation corresponds to the market coverage as in Figure 9 .

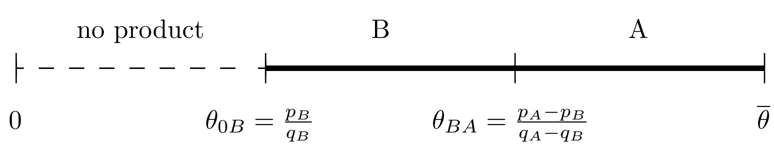

Figure 9: Standard Bertrand Competition

From F.O.C. and S.O.C., stated in Appendix A.3.1 , we obtain equilibrium prices and profits for both developers:

$$p_A^o = 2\bar{\theta} q_A \frac{q_A - q_B}{4q_A - q_B}, \ p_B^o = \bar{\theta} q_B \frac{q_A - q_B}{4q_A - q_B}. \tag{74}$$

$$\pi_A^o = 4\bar{\theta}^2 q_A^2 \frac{q_A - q_B}{\left(4q_A - q_B\right)^2}, \ \pi_B^o = \bar{\theta}^2 q_B q_A \frac{q_A - q_B}{\left(4q_A - q_B\right)^2}. \tag{75}$$

[25]It might be quite an unrealistic case that does not mimic an Operating System or Office Packages sub-markets (or other, on retail focussed, markets), nevertheless specific business software markets, e.g. CAD systems, are close to such situation. In such cases, the illegal usage of software often precedes official buying and exists mainly because of testing purposes.

Corollary 3 *Equilibrium with $p^o_A, p^o_B \leq X$ exists if and only if $X \geq 2\bar{\theta} q_A \frac{q_A - q_B}{4q_A - q_B}$. This could be immediately seen from (74).*

Obviously, both the prices and the profits of each developer increases when value of his own product increases or when a competitor's product value decreases. We see directly from the equilibrium that the relationships between prices and profits are the following:

$$\frac{p^o_A}{p^o_B} = 2\frac{q_A}{q_B}, \ \frac{\pi^o_A}{\pi^o_B} = 4\frac{q_A}{q_B}.$$

We refer to this case as the pure Bertrand competition.

2.3.2 Case 2: Bertrand competition under low expected penalty: $\frac{X}{q_A} \leq \frac{p^o_B}{q_B} \leq \frac{p^o_A}{q_A}$

Now we focus on the second case when the expected penalty X is lower than the level that would allow for the pure Bertrand competition stated above (which means $\frac{X}{q_A} \leq \frac{p^o_B}{q_B} \leq \frac{p^o_A}{q_A}$). We assume here that developer A cannot decrease the price at the level of (or below) X due to large entry costs, for instance, and developer B cannot react on it as well. (The case when both developers could adjust the price accordingly to X is analyzed as a special case in the next sub-section—Case 3). Putting p^o_B from (74) into $\frac{X}{q_A} \leq \frac{p^o_B}{q_B}$, we see immediately that X must be lower than $\bar{\theta} q_A \frac{q_A - q_B}{4q_A - q_B}$. In this case, the expected penalty is so low that all users prefer to use the product illegally. Since the expected penalty X is the same no matter which product is used, all users illegally use the product of higher quality q_A from developer A. It corresponds to the market coverage as in Figure 10 .

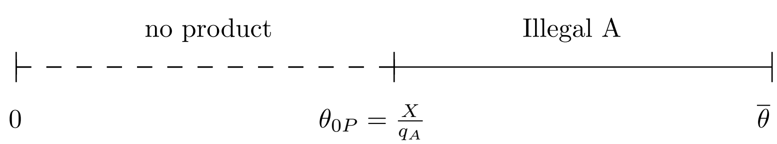

Figure 10: No legal version on the market

Remark 19 *Though this situation might seem implausible because none of the developers may generate profit unless they set their prices such that $p = X$, we could, nevertheless, find such a situation at which IPR practically does not exist (X goes to zero), and a price reduction from the developers would not lead to significantly higher sales*[26]*. At such markets, developers officially do not operate and are active on other markets, so we observe only the demand side of the market. Alternatively in dynamic models, developers may operate in those markets anticipating an improvement in IPR and so expecting to achieve profit in the future based on the established market position today, see Kúnin (2000).*

2.3.3 Case 3: Bertrand competition under medium expected penalty $\frac{p^o_B}{q_B} \leq \frac{X}{q_A} \leq \frac{p^o_A}{q_A}$

In this case, we assume that the expected penalty X influences only developer A since his equilibrium price p^o_A is higher than X (while still we have $p^o_B < X$). From (74), we see that $\bar{\theta} q_B \frac{q_A - q_B}{4q_A - q_B} \leq X$, and we assume

[26]Those are developing/emerging markets with very low IPR protection, where the percentage of illegal versions can be higher than 95%—e.g., the illegal usage evaluated in Vietnam according to the BSA (Business Software Alliance).

for the moment that developer A is not present in the market due to, say, large set-up costs that exceed profit. Then there are users who prefer to buy product B rather than face the risk of being caught as a user of an illegal version, while potential users of product A prefer to use an illegal version of product A. In such a market, developer B competes with an illegal version of software A in a sense that the upper part of the market that belonged to developer A, as in Case 1, is now occupied by the illegal users of A. A user θ_{0B} who is indifferent between using product A illegally and product B legally satisfies $\theta_{BP}q_A - X = \theta_{BP}q_B - p_B$. A user indifferent between using product B legally and not using any product at all satisfies $\theta_{0B}q_B - p_B = 0$. Thus, $\theta_{BP} = \frac{X-p_B}{q_A-q_B}$ and $\theta_{0B} = \frac{p_B}{q_B}$. This market situation leads to product coverage across users as in Figure 11 .

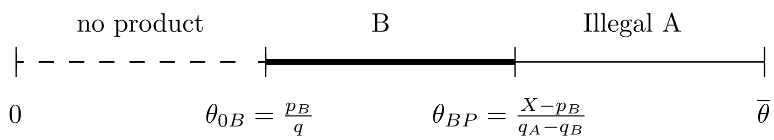

Figure 11: Only developer B is on the market with a legal version

Developer B captures market share $\theta_{PB} - \theta_{0B}$, while an illegal version of product A is used by $\bar{\theta} - \theta_{BP}$ users. In this case, only developer B makes some profit $\pi_B = (\theta_{BP} - \theta_{0B})\, p_B$. Equilibrium profits and prices are derived in Appendix A.3.2 :

$$p_B^* = \frac{q_B}{2q_A}X,\ \pi_B^* = \frac{q_B}{4q_A}\frac{X^2}{(q_A - q_B)}. \tag{76}$$

Developer A could decrease his price to a level $p_A = X$, and this behavior does not affect market share or the profit of developer B (see subcase 2.3.4 and Figure 12).

In real life, this market situation corresponds to the competition between a small local developer producing a lower quality product with a global developer, who may even not formally operate on the market. Close to this market situation, was the situation with system utilities and antivirus programs in Russia or in China around the year 2000 when global developers were waiting for improvement in IPR protection and making very negligible profit.

In this case where $\frac{p_B^o}{q_B} \le \frac{X}{q_A} \le \frac{p_A^o}{q_A}$, developer A could decrease the price to level $p_A = X$ and compete with developer B with this adjusted price, so we focus next on this interesting subcase.

2.3.4 Bertrand competition with the binding price $p_A^* = X$

Note that in Cases 2 and 3 where X is low, developer A has the possibility to decrease the price to $p_A^* = X$. The costs of this is that developer A has to incur the set-up costs F_A. Comparing it with Case 1, that price adjustment puts pressure on developer B to also lower his price p_B, so it results in a decreased profit for both developers. This market situation leads to the following equilibrium prices and profits (see Appendix

A.3.3):

$$p_A^* = X, \;\; p_B^* = \frac{1}{2}\frac{q_B}{q_A}X. \tag{77}$$

$$\pi_A^* = \frac{1}{2}X\frac{\left(X\left(q_B - 2q_A\right) + 2\bar{\theta}q_A\left(q_A - q_B\right)\right)}{q_A\left(q_A - q_B\right)}, \;\; \pi_B^* = \frac{q_B}{4q_A}\frac{X^2}{\left(q_A - q_B\right)}. \tag{78}$$

Nevertheless, this price adjustment forced by a lower X leads to a similar market distribution as in the first case, where $X \geq p_A^*, p_B^*$, but the total number of users is now higher. Market coverage is displayed in figure 12 .

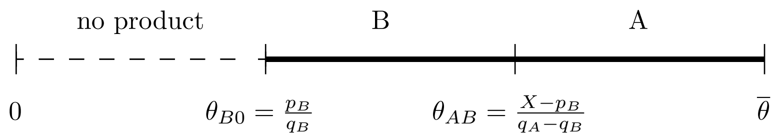

Figure 12: Competition with the adjusted price p_A

Lemma 12 *Relative prices are proportional to the corresponding quality levels and are the same as the ratio of the price derivatives with respect to X. As for the respective profits, note that the profit for developer A increases by a larger amount in X than the profit for developer B:*

$$\frac{p_A^*}{p_B^*} = \frac{\frac{\partial p_A^*}{\partial X}}{\frac{\partial p_B^*}{\partial X}} = 2\frac{q_A}{q_B},$$

$$\frac{\frac{\partial \pi_A^*}{\partial X}}{\frac{\partial \pi_B^*}{\partial X}} = \frac{2\bar{\theta}}{X}\frac{q_A\left(q_A - q_B\right)}{q_B} - 2\left(2\frac{q_A}{q_B} - 1\right).$$

Obviously an increase in X implies an increase in p_A by the same amount (that is $\frac{\partial p_A(X)}{\partial X} = 1$), while the induced rise in p_B is much lower $\left(\frac{\partial p_A(X)}{\partial X} = \frac{1}{2}\frac{q_B}{q_A} < \frac{1}{2}\right)$. Consequently, an increase in developer A's profit (due to a rise in X) is larger than the increase in developer B's profit despite the fact that market coverage of developer A shrinks at the expense of developer B. Note that developer B gains the lower tail of developer A's market, and this gain exceeds the loss of the lower tail of his own market. The latter occurs due to an increase in p_B induced by an increase in X. In other words, $\frac{\partial}{\partial X}\left(\theta_{BA} - \theta_{0B}\right) = \frac{1}{2(q_A - q_B)} > 0$.

2.4 Stackelberg competition in prices

In this part, we focus on Stackelberg (Leader–Follower) competition and particularly only on the situation when both developers are present on the market:

1. $X > p_A^*, p_B^*$ ... high expected penalty X, that no developer is restricted by X.
2. $p_A = X$ ... Developer A is limited in price setting by the expected penalty.

The other cases are equivalent to the cases analyzed in Bertrand competition and monopoly set-up in the previous part. Note, that in our analysis of the Stackelberg framework, we always assume that developer A, who releases a higher quality product q_A, is a price leader, while developer B is a price follower[27].

[27] Assuming that developer B is the price leader could be possible for selected sub-markets that correspond to the situation, where B is well established main stream player, while A is a niche player for a small proportion of high-end users, and A has to adjust his price according to the main stream. However, analyzing this market structure would not add value to this essay.

2.4.1 Stackelberg competition with a high expected penalty $p_A^*, p_B^* \leq X$

Since developer A is the price leader who knows the reaction function of developer B, he incorporates this reaction function of developer B (that is $p_B(p_A) = p_A \frac{q_B}{2q_A}$) into his profit function, and thus we obtain equilibrium prices and profits:

$$p_A^* = \frac{q_A - q_B}{2q_A - q_B} q_A \bar{\theta}, p_B^* = \frac{1}{2} \frac{q_A - q_B}{2q_A - q_B} q_B \bar{\theta}, \tag{79}$$
$$\pi_A^* = \frac{1}{2} \frac{q_A - q_B}{2q_A - q_B} q_A \bar{\theta}^2, \ \pi_B^* = \frac{1}{4} \frac{q_A - q_B}{(2q_A - q_B)^2} q_B q_A \bar{\theta}^2.$$

(For F.O.C. and S.O.C., see Appendix A.4.1 .)

Remark 20 *Compared with Bertrand competition, the profit of developer B is always higher in Stackelberg competition.*

In the next section, we will show a case (a buffer case), where developer B is indifferent between Stackelberg and Bertrand competitions since both frameworks bring him exactly the same profit.

2.4.2 Stackelberg competition with binding price $p_A^* = X$

Assume a market situation with a low expected penalty X which becomes binding (if $X \leq \frac{q_A - q_B}{2q_A - q_B} q_A \bar{\theta}$). Then, equilibrium prices and profits are the same as in the case of Bertrand competition with binding X as stated in 2.3.4 :

$$p_A^* = X, \ p_B^* = X \frac{q_B}{2q_A}, \tag{80}$$
$$\pi_A^* = \frac{1}{2} \frac{\left(2\bar{\theta} q_A (q_A - 2\bar{\theta} q_B) - X(q_A - q_B)\right) X}{q_A (q_A - q_B)}, \pi_B^* = \frac{1}{4} \frac{q_B}{q_A} \frac{X^2}{q_A - q_B}.$$

(For a derivation see Appendix A.4.1 .) In the Stackelberg case, the distribution of users on the market is equal to the Bertrand competition. The market coverage is the same as in Figure 12 .

2.5 Possible Network Effect Extension

Before concluding this section and approaching an analysis of product protection, we make small remarks on Network Effect since NE plays an important role on some software sub-markets. In these remarks, we will show a possible incorporation of the Network effect into the model.

Capturing the significant base of users may allow a developer to extract additional value from those users. In such a situation, the developer may tend to predatory behavior (entering the market with a low price to capture base, and increasing the price later), when he deliberately supports illegal copying to raise the user base and thus user value from a product and in the future, implement protection which results in "locking" the base of users (see Farrell and Klemperer, 2006)[28]. In non-predatory competition, the NE could raise the product's value substantially. Our set-up would allow for the capturing of the NE by internalizing it into the product quality. Assume now that q_i is composed from:

$$q_i = \beta_A \cdot Q_i + (1 - \beta_A) NE_i,$$

[28]Usually solved by a two-period model, where in the first period a developer raises his user base and in the second period, charges them for the additional value from the user base.

where Q_i is the quality of product itself, and NE_i is the value that a user puts into the user base, and $1 \geq \beta_A \geq 0$ is the weight of each component. In general, using the model with the NE we would search for the optimal value of each component Q_i, and NE_i, and the equilibrium would consist of p_i^*, Q_i^*, and NE_i^*. Such an equilibrium would be strongly dependent on weights β_A and the set-up cost functions $F_A(Q_A)$, $F_B(Q_B)$. Given the main focus of the essay, an extension of the set-up for the NE would not bring significant additional insights to our analysis.

Moreover, the NE starts to become a less important driving factor for most software submarkets recently, and especially, the NE is even already quite suppressed for industrial software. The more important factor becomes software compatibility among competitors and their mutual replacement ability (e.g. functions or layout)[29].

2.6 Key chapter results

From the analysis in Section 2, we see that imposing penalties has a strong impact on the resulting market coverage. In the case of the high expected penalty X, there is standard competition in prices (either Bertrand or Stackelberg), while in the case of the medium expected penalty X, the developer with the higher quality product has to either leave the market or decrease the price. In such a market situation, X fosters the competition and forces both developers to decrease prices, but at the same time, too low of an X could squeeze one of the developers out of the market since he may not be able to recover his set-up costs any longer. Thus, a very low X increases the toughness of price competition that in turn may result in a monopoly market structure. From the government's point of view, X may serve as an artificial price, set by the regulator, that must either be accepted by developer A, or he has to leave the market. In the case of a very low X, none of the developers would operate on the market.

3 Decreasing product value for illegal users

In the previous section, users did not perceive a quality (value) difference between the original product and its illegal version, and thus, users always chose the version with a lower "cost" per quality unit ($\frac{p_i}{q_i}$ in the case of a legal version and $\frac{X}{q_i}$ in the case of an illegal version). In this chapter, we assume that the legal and illegal versions are no more perfect substitutes. That is, the value of the legal version differs from the illegal version since a developer provides part of valuable services only to legal users (such as online help and technical support, live updates, a discount for upgrades or even free training, access to user manuals, etc.). Probably the most famous example of restricting services to illegal users, familiar to everybody, is the one with Microsoft Windows. Microsoft's Windows Genuine Program allows a user to run an illegal version of the product only up to a certain point. In order to install selected patches/updates, the user has to validate the originality of the program online. If a particular copy is identified as illegal, some functions are disabled, and the illegal user is irritated with constant messages about buying the legal version. If a user decides

[29] Thus, in the case of analyzing the Network Effects, we should always distinguish cases where products A and B are mutually incompatible, partially compatible, or fully compatible. The compatibility is then a factor that allows the competitors to exploit the user base and to suppress the NE advantage of a particular product.

not to validate the program online, he cannot update his Windows further for selected components (e.g. Windows Media Player or Internet Explorer)[30]. The implementation of such a restriction is technically easy since the developer could use the standard tools that restrain access to those services[31] that require user authorization based on personal information verification. In the case of automatic access to those services, a developer can use very reliable tools as authorization is based on the IP address or hard-locks.

3.1 Model set-up

3.1.1 Industry set-up

We now assume that developers cannot directly restrict illegal usage of the product itself but could restrict part of the services related to the product. This restriction lowers the product value for illegal users. Denote the value (perceived quality) of the legal version as q_i. The exclusive part of the product value that only the legal users can enjoy is $1-\alpha$, where $\alpha \in (\underline{\alpha}, 1)$, and $\underline{\alpha} > 0$ stands for technically the lowest possible level of restriction beyond which it is impossible to further restrain services [32]. Thus, the value of the product for the illegal users is decreased[33] to αq_i. As for the developers' costs of restricting services, it seems reasonable to assume that these costs are negligible given that the developers already exist and have chosen their quality levels and the accompanied level of consumer services previously. So, we assume these costs to be zero, but we do discuss the implications of non-zero costs for the optimal choice of α in section 3.4 . For simplicity, we assume that if both developers choose to restrict services, then they would choose the same α[34]. In a formal sense, adding the possibility for the developer to choose the degree of service restrictions can be considered as a two-stage game: In the first stage, one or both developers choose the degree of service restrictions α, and in the second stage, the developers compete in prices.

In what follows, we focus on the second stage of the game in which a developer chooses the prices and analyze the impact of different α on equilibrium prices and on the resulting market structure and coverage. In the last sub-section of this chapter, we briefly discuss the optimal choice of service restrictions.

Note that in the case of a low expected penalty, X, that results in a "protecting" action from the developer, we could perceive X as public protection while an action from a developer (in this case introducing α) as private protection.

Remark 21 *Note that developer B has somewhat limited incentives to restrain certain services to illegal users. Since the expected penalty is the same for whatever product is used illegally, the users would always prefer to use the illegal version of product A to illegal version of product B*[35]*. The only case when developer*

[30]Another examples is antivirus programs (e.g., Symantec Antivirus), when often after updates, the program recognizes that a particular installation is illegal and does not allow further updates of its virus database for new viruses. Finally, for many computer games, online playing is allowed only for the legal users.

[31]A separate question is whether a developer could provide enough services/online content that would bring a user enough additional value to offset the difficulties with accessing those services/content.

[32]Technically, the lowest possible $\underline{\alpha}$ represents such α where quality q_A to legal users is not affected. By decreasing α below $\underline{\alpha}$, we assume that it would require such strong protection/verification tools (e.g. manual online authorization), which become annoying even for legal users, and their value assigned to product q_A would drop.

[33]Banarjee (2003) explains the difference between an original and illegal version as the probability of an occurrence of a defect illegal version. In his framework, "not" defected illegal copy is the same for a user as a legal copy.

[34]Allowing for different α would in no way change qualitatively our analysis. It would make the results and analysis only less transparent since it would always require a comparison between illegal version values, see Bertrand competition in 3.3 .

[35]In real life, the developer of a product with a lower quality competes strongly with an illegal version of a better product

B would have the incentive to introduce the restriction of his services is, as we will see, when X is "low," and A introduces strict restrictions of his services (low α). In that case, the lack of developer B implementing protection would result in the illegal usage of product B (in this case, an illegal version of B has quality q_B, while an illegal version of A has αq_A, which is lower than q_B). In other words, no user would use product B legally unless developer B also implements protection (see Case 2 below).

3.1.2 Users set-up

As in the previous chapter, we assume that every user has access to all the versions: to both legal versions A, B and to the illegal versions of A, B and decides based on the product prices and values. Utility for a user θ is then:

$$U_P(\theta) = \begin{cases} \theta q_i - p_i & \text{... if he buys software.} \\ \theta \alpha q_i - X & \text{... if he uses software illegally.} \\ 0 & \text{... if he does not use software at all.} \end{cases} \tag{81}$$

The important difference in Section 2 is developer A cannot make any profit if he sets the price p_A higher than X because its legal product is eliminated from the market. In this section, on the other hand, there might be some users (top-end users with high θ) that may prefer to buy the legal version rather than the restricted illegal one even if both versions (legal and illegal) are available and even when X goes to zero. From utility function (81), we can identify 6 types of users indifferent between some two actions. Those users appear on the market under different levels of X, q_A, q_B, and α. Only some of the indifferent users exist on a particular market but never all of them. Here are the 6 types of indifferent users:

1. θ_{PA} ... The user indifferent between using legal product A and its illegal version.
2. θ_{0P} ... The user indifferent between using illegal version A and using nothing at all.
3. θ_{0A} ... The user indifferent between using legal product A and using nothing at all.
4. θ_{0B} ... The user indifferent between using legal product B and using nothing at all.
5. θ_{BP}, θ_{PB} ... The user indifferent between using legal product B and using illegal version A.
6. θ_{BA} ... The user indifferent between using legal product A and using legal product B.

In this chapter, we will again use the notation introduced in Notation 4 . As in the previous chapter, for a better illustration of the model behavior, we shall start with the monopoly case.

3.2 Monopoly

In the case of a monopoly market, developer A can compete only with an illegal version of his own product. Similarly to the previous chapter, if the expected penalty X is high enough that nobody is willing to use software illegally, we obtain the same market structure as in Section 1 (captured on Figure 8). This situation occurs when $\frac{1}{2}\bar{\theta}\alpha q_A \leq X$.

In the case where $X \leq \frac{1}{2}\bar{\theta}\alpha q_A$, there are users who prefer to use the illegal version and so setting α as low as possible is the right thing to do in order to increase the demand for the legal version In order to work out the monopolist's demand, we find user θ_{PA}, who is indifferent between the legal and illegal product, and so this user is described by $\theta_{PA} = \frac{p_M - X}{q_A - \alpha q_A}$. The demand for product A is then $D_A = (\bar{\theta} - \theta_{PA})$, and the monopolist profit is $\pi_M = (\bar{\theta} - \theta_{PA})p_M$, while the demand for the illegal version is $D_P = (\theta_{PA} - \theta_{0P})$. Equilibrium price and profit are:

developer. Both developers know that introducing sophisticated protection could only discourage legal users from their services, while illegal users would always prefer to use a better product.

$$p_M^* = \frac{X + \bar{\theta} q_A (1 - \alpha)}{2}, \ \pi_M^* = \frac{1}{4} \frac{\left(X + \bar{\theta} q_A (1 - \alpha)\right)^2}{q_A (1 - \alpha)}. \tag{82}$$

This results in the distribution of users on the market as captured in Figure 13 .

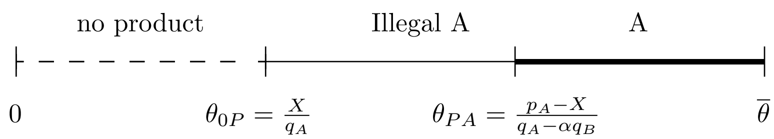

Figure 13: The decreased quality to illegal users on the monopoly market

Note that the monopolist that faces illegal usage but has an option to increase the number of legal users by restricting additional services generates uniformly higher profit than the monopolist that could only set $p_M^* = X$.

Clearly, now the profit of the monopolist increases not only in the level of expected penalty but also in the degree of restrictiveness to the additional services (that is, the lower the α, the higher the monopolist's profit). Thus, the maximal restrictions of services to the illegal users are optimal requiring the minimum possible level of α that we label as $\underline{\alpha}$.

3.3 Bertrand competition

A user's decision to use an illegal version now again depends on the user's sensitivity to product quality θ as well as on the expected penalty X. We first focus on the situation in which only a developer of a higher quality product uses the restriction in services[36]. If X is high enough such that $p_B < p_A < X$, then[37] illegal usage is fully suppressed, and the market is divided between both developers, which is in fact, the case we have already analyzed in pure Bertrand competition in Section 2, see Figure 12 . Assuming that illegal usage is not eliminated, and legal versions are on the market, then there are two interesting cases in which both developers operate on the market. The first one is $p_B < X < p_A$ and $q_B < \alpha q_A$ (implying $\frac{p_B}{q_B} \leq \frac{X}{\alpha q_A} \leq \frac{p_A}{q_A}$), and the second one is $X < p_B < p_A$ and $\alpha q_A \leq q_B$ (implying $\frac{X}{\alpha q_A} \leq \frac{p_B}{q_B} \leq \frac{p_A}{q_A}$). In all other cases, either the legal version of product B is eliminated so there is a monopoly for developer A, or the illegal usage of product A is eliminated and there is pure Bertrand competition.

In the first case when $\frac{p_B}{q_B} \leq \frac{X}{\alpha q_A} \leq \frac{p_A}{q_A}$ and $q_B \leq \alpha q_A$, developer A competes with an illegal version of his own product to capture users with relatively high θ, while developer B competes with an illegal version of product A to capture users with relatively low θ (See Figure 14). In the second case, when $\frac{X}{\alpha q_A} \leq \frac{p_B}{q_B} \leq \frac{p_A}{q_A}$ and $\alpha q_A < q_B$, developer A competes with developer B for users with high θ, while developer B competes with the illegal version of A for users with low θ (see Figure 15). The second case leads to a tougher competition between developers A and B, where q_A and q_B are relatively close, while the first case describes

[36]In the case that developer B also introduces a restriction of, say, α_B, then the product of developer B will not be used illegally unless α_B is significantly higher than α resulting in $\alpha_B q_B \geq \alpha q_A$ as any user who decides on using an illegal product would use the illegal product of the highest available quality since the expected penalty is X regardless of the product.

[37]In this part, whenever we write $p_B < p_A$ we mean $\frac{p_B}{q_B} < \frac{p_A}{q_A}$, which is a necessary condition for product B to be in the market.

a market where developer B produces a significantly less valuable product than developer A, and thus, he can hardly compete with his legal version[38].

3.3.1 Case 1: Bertrand competition when $p_B < X < p_A$ and $q_B < \alpha q_A$, second stage

$\left(\frac{p_B}{q_B} \leq \frac{X}{\alpha q_A} \leq \frac{p_A}{q_A}\right)$

This situation corresponds to a product distribution over the market in which there are three types of indifferent users:

1. A user indifferent between buying product A and its illegal usage: $\theta_{PA} = \frac{p_A - X}{q_A - \alpha q_A}$,
2. A user indifferent between the illegal usage of product A and buying product B: $\theta_{PB} = \frac{X - p_B}{\alpha q_A - q_B}$, and
3. A user indifferent between buying product B and not using any product at all: $\theta_{0B} = \frac{p_B}{q_B}$.

All users with $\theta \in (\theta_{BP}, \theta_{PA})$ use an illegal version of product A. The users of the illegal version split the market into two sub-markets and to put it roughly, the illegal users recruit themselves from the middle part of the market. The profit function for each developer is then $\pi_A = \left(\bar{\theta} - \theta_{PA}\right) p_A = \left(\bar{\theta} - \frac{p_A - X}{q_A - \alpha q_A}\right) p_A$, and $\pi_B = \left(\theta_{BP} - \theta_{0B}\right) p_B = \left(\frac{X - p_B}{\alpha q_A - q_B} - \frac{p_B}{q_B}\right) p_B$. Equilibrium prices and profits are the following:

$$p_A^* = \frac{\bar{\theta} q_A (1-\alpha)}{2} + \frac{X}{2},\ p_B^* = \frac{q_B}{2\alpha q_A} X \tag{83}$$

$$\pi_A^* = \frac{1}{4} \frac{\left(\bar{\theta} q_A (1-\alpha) + X\right)^2}{q_A (1-\alpha)}, \pi_B^* = \frac{1}{4} \frac{q_B}{\alpha q_A} \frac{1}{\alpha q_A - q_B} X^2. \tag{84}$$

(F.O.C. and S.O.C. are stated in Appendix A.3.1), and resulting market coverage is the following:

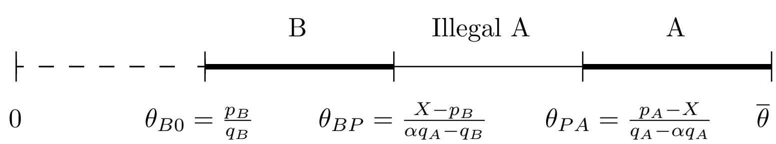

Figure 14: BC with illegal users in the middle of the market

Lemma 13 *The only necessary and sufficient condition with respect to X for this kind of equilibrium to exist is:*

$$0 < X < X_{\alpha 1} = \frac{\bar{\theta} \alpha q_A (\alpha q_A - q_B)(1-\alpha)}{(2-\alpha)\alpha q_A - q_B}.$$

Proof. see Appendix B.3.2

■

In this special case, only developer A has the incentive to choose service restriction in the first stage. Moreover, note that the developers do not directly compete against each other because users who are using product A illegally create a "buffer" between the legal users of products A and B. Thus, the profit of

[38] E.g., competition between the Microsoft Office 2010 package against small alternative developers such as 602 and its package known as "OpenOffice.org Software 602 Edition."

each developer is independent on competitor's price and the driving factors of the profit are the level of the expected penalty X, and the level of restricted services α. Moreover, note that the market coverage, equilibrium price, and, consequently, profit of developer A are the same as if he was a monopolist constrained by $X \leq p_A^*$ (implying that $X \leq \frac{1}{2}\bar{\theta}\alpha q_A$, see sub-section 3.2).

Remark 22 *Developer A's decision to implement α and then set the price to $p_A = X$ is never optimal in the given set-up.*

Lemma 14 *In the case of duopoly competition when $\frac{p_B}{q_B} \leq \frac{X}{\alpha q_A} \leq \frac{p_A}{q_A}$ and $q_B \leq \alpha q_A$, the equilibrium profit and price of developer A as well as developer B are decreasing in α as long as $q_B \leq \alpha q_A$ holds.*

Proof. The behavior of $p_A^*(\alpha)$, $p_B^*(\alpha)$, $\pi_B^*(\alpha)$ can be seen immediately from equilibrium (83) and (84), proof that $d\pi_A^*(\alpha)/d\alpha < 0$ could be also easily derived. ■

Intuitively as α decreases, the illegal usage becomes more costly and consequently shrinks. Since both developers compete directly only with an illegal version of product A, this improves their competitive advantage by making legally accessible quality more attractive compared to the illegal one allowing in turn both prices to increase in equilibrium.

3.3.2 Case 2: Bertrand competition when $X < p_B < p_A$ and $q_B > \alpha q_A$, second stage ($\frac{X}{\alpha q_A} \leq \frac{p_B}{q_B} \leq \frac{p_A}{q_A}$)

Note that in this set-up, developer B would also be forced to introduce the IPR protection of α in order to stay in the market. Otherwise the users who do not buy a legal version of product A, would prefer to use the illegal version of product B, whose quality would be $q_B > \alpha q_A$. As a consequence of IPR implementation by both developers, there would be a direct competition between the two developers, but their payoffs depend on the level of X and the developers' IPR protection α. A user indifferent between A and B is $\theta_{BA} = \frac{p_A - p_B}{q_A - q_B}$, and a user indifferent between illegal usage of A and buying B is $\theta_{PB} = \frac{p_B - X}{q_B - \alpha q_A}$. Users with $\theta \in \left(\frac{X}{\alpha q_A}, \frac{p_B - X}{q_B - \alpha q_A}\right)$ use an illegal version of product A. The profits for developers are: $\pi_A = \left(\bar{\theta} - \theta_{BA}\right) p_A$ and $\pi_B = \left(\theta_{BA} - \theta_{PB}\right) p_B$. This situation leads to the following distribution on the market:

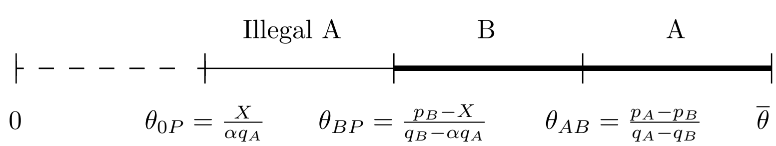

Figure 15: BC with illegal users at the low end of the market

This results in the following equilibrium prices and profits:

$$p_A^* = (q_A - q_B)\frac{\bar{\theta}2q_A(1-\alpha) + X}{4q_A - 3\alpha q_A - q_B},\; p_B^* = (q_A - q_B)\frac{\bar{\theta}(q_B - \alpha q_A) + 2X}{4q_A - 3\alpha q_A - q_B}. \tag{85}$$

$$\pi_A^* = (q_A - q_B)\left(\frac{2\bar{\theta} q_A (1-\alpha) + X}{4q_A - 3\alpha q_A - q_B}\right)^2, \tag{86}$$

$$\pi_B^* = (q_A - q_B)(1-\alpha) q_A \frac{\left(\bar{\theta}(q_B - \alpha q_A) + X\right)^2}{(q_B - \alpha q_A)(4q_A - 3\alpha q_A - q_B)^2}.$$

Lemma 15 *A necessary condition for the existence of an equilibrium is satisfied only for X and α such that:*

$$0 \leq X_{\alpha 2} = \frac{\bar{\theta}(q_A - q_B)(q_B - \alpha q_A)\alpha q_A}{4q_A q_B - q_B^2 - 2\alpha q_A^2 - \alpha q_A q_B}.$$

Proof. see Appendix B.3.3

■

Lemma 16 *The equilibrium profit and price of developer A as well as developer B are decreasing in α when $q_B > \alpha q_A$ holds.*

Recall that the competition in Case 2 is tougher than in Case 1 since developers now compete directly with each other, and the increase in market share of one developer automatically implies a decline in the share of the other developer.

3.3.3 Stackelberg competition

In Case 1 above, the illegal version of product A serves as a "buffer" between the two legal products, and the prices chosen by the developers do not depend on each other, i.e., the reaction functions $p_i(p_j)$ do not depend on p_j. Therefore, the Stackelberg outcome in this case is the same as the Bertrand outcome above. As for Case 2, the Stackelberg outcome is in Appendix B.4 .

3.4 Optimal service restrictions: the first stage

The optimal service restriction is rather simple in our set-up given the assumption of no costs for restraining services. (Recall that profit functions in both Cases 1 and 2 decrease in α.) Thus, the optimal service restriction is always such that $\alpha^* = \underline{\alpha}$ irrespective of the level of X (provided that the size of X is such that it requires the imposition of a service restriction by at least one developer, that is, $X < p_A$). What is more interesting here is to see how the levels of optimal α and X affect the emerging market structure and market coverage in the second stage equilibrium. We start with the buffer case: Case 1.

3.4.1 $\alpha^* = \underline{\alpha}$, and $p_B < X < X_{\alpha 1}$

This case appears in equilibrium when α is relatively large ($\alpha^* q_A > q_B$), and this is typically the case when the quality of the first developer is "substantially" larger than the quality of the second developer. The interesting (comparative static) question to ask here is what would happen if the regulator increases X to be at $X_{\alpha 1}$ or larger. If X exceeds $X_{\alpha 1}$, then piracy becomes too costly and, consequently, the buffer of illegal users is completely eliminated (that is, $\underline{\alpha} q_A \bar{\theta} - X < 0$ for all θ). Thus an expected penalty high enough (such that $X > X_{\alpha 1}$), restores pure Bertrand competition, and so, a pair of private and government protections

$\{\underline{\alpha},\ X_{\alpha 1}\}$ are able to restore the competition without illegal users. Recall that without private protection, the regulator would have to set a much higher excepted penalty to achieve the same outcome (that is, X has to be such that $X > p_A$).

An alternative way in which the pure Bertrand competition would appear in equilibrium is the situation when $\underline{\alpha} q_A = q_B$ (or $\underline{\alpha} q_A$ is "close" enough to q_B). The intuition is similar to the one above; the usage of an illegal version becomes non-attractive when α falls so low that a legal version of product two has the same (or only slightly lower) value for consumers but is offered at a lower price, $p_B < X$. Thus, again the illegal usage is completely eliminated.

3.4.2 $\alpha^* = \underline{\alpha}$, and $X < p_B < X_{\alpha 2}$

Clearly, this situation appears when $\underline{\alpha}$ is relatively small so that $\underline{\alpha} q_A < q_B$, and there is direct duopoly competition (Case 2 above) in which illegal usage occurs only at the lowest tail of the market. This would likely be the case when quality of the first developer is not "much" larger than the quality of the second developer. Note that it would be now optimal for both developers to introduce service restrictions. Moreover, both developers choose the technically minimal possible α, i.e., $\alpha_A^* = \alpha_B^* = \underline{\alpha}$. Let us assume again that the regulator sets the expected penalty that exceeds the critical value for Case 2 to occur, (that is, $X > X_{\alpha 2}$). In that case, $\frac{X}{\alpha q_A} > \frac{p_B}{q_B}$ (or $\theta_{0P} > \theta_{0B}$) implying that no one would use an illegal version and again, the competition would be back to the pure Bertrand. Thus, in this set-up too, a pair of $(\underline{\alpha}, X_{\alpha 2})$ restores a pure Bertrand competition and much like in the case of 3.4.1 above, X_{c2} is substantially lower than the expected penalty that would alone achieve complete elimination of illegal usage.

So in both cases, private and government IPR protection are substitutes in the sense that introducing private protection in the form of service restrictions, enables the regulator to eliminate the illegal usage of software with a much lower (and less costly) expected penalty.

Finally, if we alternatively assume that:

1. It would be costly for the developers to incur service restrictions,
2. Optimal α can be anywhere in the interval $(0, 1)$, and
3. The corresponding cost function for implementing α, $C(\alpha)$ is convex enough to generate an interior maximum, then α^* can be such that either $\alpha^* q_A > q_B$ or $\alpha^* q_A < q_B$ depending on the size of X and the shape of the cost function, $C(\alpha)$ such that $\frac{\partial}{\partial \alpha} C(\alpha) > 0$ and $\frac{\partial^2}{\partial \alpha \partial \alpha} C(\alpha) > 0$.

3.5 Key section results

In the case of the monopoly set-up, we show that developer A competes against an illegal version of his own product, and it is optimal for him to maximally restrict the additional software services for illegal users. He always operates in the market and is always better off than the monopolist that sets the price to an expected penalty when faced with illegal usage (as in Section 1).

In the case of Bertrand competition, the interaction between the developers' and the regulator's IPR (as well as the nature of the competition) depends critically on whether IPR protection, X, is "High," "Medium," or "Low."

1. A "High" expected penalty can be defined as the one in which $p_B \leq p_A \leq X$. In this case, the developers' IPR protection is redundant, and obviously the regulator's IPR protection acts as a substitute for the developers' IPR.

2. The regulator IPR protection is "Medium" if $p_B \leq X \leq p_A$. In that case, we see that the situation of the highest interest is the one where $\frac{p_B}{q_B} \leq \frac{X}{\alpha q_A} \leq \frac{p_A}{q_A}$ and $q_B < \alpha q_A$. We call it a buffer case. Developer A earns the same profit as a (constrained) monopoly that is positively affected by the level of the expected penalty X. The optimal service restriction is always such that $\alpha^* = \underline{\alpha}$ irrespective of the size of X.

3. Finally, there is a third case of "Low" IPR protection, in which $X \leq p_B \leq p_A$. The most interesting situation occurs when $\frac{X}{\alpha q_A} \leq \frac{p_B}{q_B} \leq \frac{p_A}{q_A}$ and $\alpha q_A < q_B$. In this situation developers, compete directly against each other, and the level of X positively affects the profit functions of both developers. Moreover, developer B has also to introduce the restriction of services to the degree α_B for the illegal users who would otherwise prefer to use product B illegally. The developers choose the maximal possible level of IPR protection ($\alpha = \underline{\alpha}$).

4. Marginal changes in X do not affect optimal choices of α given that the costs of the service restriction are zero. If, on the other hand, these costs are substantial yielding the inverse U-shaped profit function and interior maximum for α, then the marginal change in X does affect the optimal choice of service restriction.

5. Finally in both cases, private and government IPR protection are substitutes since the introduction of private protection enables the regulator to eliminate the illegal usage of software with a lower expected penalty.

6. In the case of the Stackelberg competition, we show that developer B has no advantage from setting his price as the second one in the "buffer" case. There is a "second mover advantage," only in the case of direct competition like in Case 2 above.

4 Physical product protection against copying

In this section, we focus on a situation where the developers can eliminate illegal usage by implementing physical protection against copying. By physical protection we understand that installing an illegal version of the software is more difficult either because of a low availability of the illegal version or because of the high requirement on the users' skill to install (or use) the illegal version. An example of such a protection is the DVD with games where a version coming from standard copying with a DVD burner cannot be installed on a PC any longer[39]. Another example is requiring users to authenticate their copy on the developer's

[39]The illegal copy is not working since the original DVD is intentionally produced with certain kinds of mistakes, and during copying, these mistakes are always corrected by the "burning" software. At the same moment, during the installation process, those mistakes are mandatory for the successful completion of the installation.

web pages during installation, which could be technically complicated to avoid (e.g., only by installing a "crack" to a particular directory and a set of steps to complete the installation). All such tools create obstacles in installing an illegal version, and thus limit its availability to common users. After installation, however, a user often may not distinguish an illegal version from the legal one. As already mentioned in the introduction, some forms of DRM can also serve as examples of such a protection. Thus, a user's perception of software quality is often intact.

As noted in the introduction, most of the research papers on IPR protection in software markets have analyzed the trade-off between the perfect protection and the costs of its implementation. In this section, we focus on what impact the size of the expected penalty X has on the developers incentives to introduce some kind of physical protection against illegal copying. We also study the impact of such protection on mutual competition between developers.

4.1 Model set-up

4.1.1 Developers' problem

We assume now that both developers have access to technology that allows product protection against copying and illegal usage [40]. The developers' decisions are dependent only on the profitability of such a step. The protection against copying is imperfect, which means that a fraction of the users still have access to the illegal version[41]. This fraction of users is uniformly distributed over the whole interval $\langle 0, \bar{\theta} \rangle$. We say that a developer implements protection at level c, if for each $\theta \in \langle 0, \bar{\theta} \rangle$ the fraction of users with the ability to use the illegal version is $(1 - c)$, and the remaining fraction of users (c) could only use the legal version. Protection c is from interval $\langle 0, 1 \rangle$, and if c tends to 1 we say that protection becomes perfect, while c tending to 0 represents the full public availability of an illegal version[42]. We further assume that both developers could implement this kind of protection, and that they could differ among themselves in the protection level c. Much like in the previous section, we can think about a two-stage game in which one or both developers choose the level of private protection in the first stage, and then they compete in prices in the second stage.

Unlike in the case of restricting services to illegal users, it is now reasonable to assume that implementing physical protection is costly, and that these costs rise more than proportionally as c increases tending to infinity as c approaches 1. Thus, the costs of implementing protection c, labelled as $C = h(c)$, possess the following properties:

1. $h(0) = 0$, $\lim_{c \to 1} h(c) = +\infty$;
2. $\frac{\partial}{\partial c} h(0) = 0$, $\frac{\partial}{\partial c} h(c) > 0$;
3. $\frac{\partial^2 h(c)}{\partial c^2} > 0$ and

[40] Neither legal nor licence restrictions are assumed for the developer in the case of implementing protection against copying.

[41] By eliminating public availability we mean both no access to an illegal version or access to an illegal version accompanied by the limited user's skill to install/use the illegal version.

[42] The availability of an illegal version and the ability to break it differs significantly among users and is more dependent on technical skill than on the sensitivity to price θ. The uniform distribution is an analytical simplification not harming the nature of the essay.

4. $\Pi_i^* = \pi_i^*(c_i) - h(c_i)$ is a concave function reaching its maximum at $c_i^* \in (0,1)$. (We use the symbol Π for net profit, when protection costs are accounted for, while π stands for the price-competition stage profit.)

Note that much like in the previous section with restricting services, we are not so interested in the actual optimal value of protection, c but rather in its interaction with the expected penalty X and, consequently, its impact on equilibrium prices, profits, and market coverage.

4.1.2 The consumer problem

Recall that in the previous sections, all users have access to illegal versions, and the user's decision to use an illegal version was always based on the expected utility coming from usage of such a version compared to the utility from using a legal version. In this section, we assume that only some users have access to both a legal and an illegal version, while some users have access only to a legal version. The users with access to both versions prefer the legal version only if the utility from it is higher and their proportion is $1-c$. The utility function of user θ is the following:

$$U_P(\theta) = \begin{cases} \theta q_i - p_i & \text{... if he buys the legal version of the software.} \\ \theta q_i - X & \text{... if he uses the software illegally.} \\ 0 & \text{... if he does not use the software at all.} \end{cases} \tag{87}$$

Users without access to the illegal version could compare only the expected utility from purchasing the legal version and not using it at all. Their proportion is c, and the utility function of user θ is:

$$U_P(\theta) = \begin{cases} \theta q_i - p_i & \text{... if he buys the legal version of the software.} \\ 0 & \text{... if he does not use the software at all.} \end{cases} \tag{88}$$

4.1.3 The market environment

As we already noted, both developers could implement physical protection for their product, and so three basic combinations of product protection could occur in the market :

1. None of the developers implement protection. This situation arises when X does not bind in the maximization problems of either A or B so that in the equilibrium, we have $p_B^* \leq p_A^* \leq X$.

2. Developer A implements protection while developer B does not. This situation occurs when pure Bertrand equilibrium is not possible because X would be binding for developer A since $p_B^* \leq X \leq p_A^*$.

3. Both developers implement protections.[43] Finally for low X, both developers would have to introduce protection since pure Bertrand equilibrium would result in $X \leq p_B^* \leq p_A^*$.

Before analyzing the above cases in more detail, we, as in the previous two sections, first start with the monopoly case that helps us to illustrate the flavor of the model.

[43] Note that the case in which only developer B implements protection never occurs. If B has to implement protection due to the low expected penalty X, then developer A must also implement physical protection because his product would be the primary target of illegal usage.

4.2 Monopoly

As in the previous two sections, we start with a monopoly case that will help us to illustrate the flavor of the model. Consider now developer A who introduces a level of protection at c for his product q_A and sets the price p_M. In analyzing monopolist behavior, we could focus only on the case when the expected penalty is such that $X < p_M$, since the case where $X > p_M$ is already described in the Section 2, and in this case, no user has the incentive to use an illegal version. Users' demand for the legal product of monopoly developer A is $D_A = c\left(\bar{\theta} - \frac{p_M}{q_A}\right)$ and leads to the following market coverage:

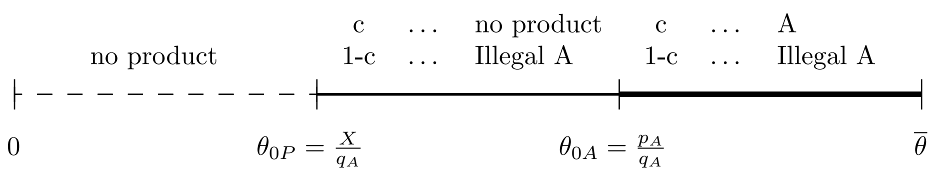

Figure 16: Monopoly market with product protection c

Monopoly equilibrium is analogous to the one in Section 1, as could be immediately derived from (72):

$$p_M^* = \frac{1}{2}\bar{\theta}q_A,\ \pi_M^* = c\frac{1}{4}\bar{\theta}^2 q_A. \tag{89}$$

Note that under the assumptions regarding $h(c)$, $\Pi_M^* = \pi_M^* - h(c)$ has a unique maximum, $c_M^* \in (0,1)$. A monopoly developer A always has an option to decrease the price to X instead of implementing protection c. By comparing developer A's profit in the case of lowering the price to X as in Section 2, see monopolist profit in (73), with his profit after implementing protection, we find out that developer A prefers physical protection as long as the expected penalty, X, is below a certain critical level. More specifically, even with protection costs $h(c) = 0$, it is more profitable to lower the price to X instead of implementing protection if $X > \bar{\theta}q_A\frac{1-\sqrt{1-c_M^*}}{2}$.

4.3 Bertrand competition

As in the previous part, we omit the case when the expected penalty X is high enough ($p_B^o \leq p_A^o \leq X$), and developers have no incentives to introduce physical protection against copying (this case we already analyzed in Section 2). In analyzing this set-up, first, we focus on the case where only developer A has the incentive to introduce protection $p_B^* \leq X \leq p_A^*$ and then, finally, on the case where both developers have such incentives, that is, $X \leq p_B^* \leq p_A^*$. Note that in our set-up, prices are as usually strategic complements (see Tirole, 1989, and Bulow et al., 1985), that is, $\frac{\partial^2 \pi_i}{\partial p_B \partial p_A} > 0$.

4.3.1 Only developer A implements protection c

In this case, where $p_B^* \leq X \leq p_A^*$, only developer A has the incentive to implement physical protection since the product of developer B would be used only legally. As we already mentioned in our model set-up, the illegal version of product A is available only to the fraction $1 - c$ of the users' base. Product A is used illegally only by users with $\frac{X}{q_A} \leq \theta$, while users with $\theta \leq \frac{X}{q_A}$ prefer not to use the product at all. The demand for product B consists of users with low sensitivity θ to purchasing product A, who, at the same time, have no access to an illegal version of A, but their θ is high enough to buy product B. These users have $\theta \in (\frac{p_B}{q_B}, \frac{p_A - p_B}{q_A - q_B})$, and their fraction is c. As for the users with access to an illegal version of product A, there are two sub-cases that could occur in equilibrium depending on the size of the expected penalty:

1. The first sub-case occurs when there are some users who have illegal access to product A but still want to buy product B, or more formally, the measure of these users is strictly positive with $\theta \in \left(\frac{p_B}{q_B}, \frac{X - p_B}{q_A - q_B}\right)$, and so, $\frac{X - p_B}{q_A - q_B} > \frac{p_B}{q_B}$. These users would like to purchase product B if X is not so "low" (in the sense that $X > p_B \frac{q_A}{q_B}$). The market coverage is given in Figure 17 .

2. The second sub-case occurs when illegal users always prefer an illegal version of A to the legal version of B, that is, when $\theta q_A - X > \theta q_B - p_B$ for all θ since illegal usage is then more profitable even for the consumer with the lowest valuation. So, X has to be "low" enough, that is, $\frac{X - p_B}{q_A - q_B} \leq \frac{p_B}{q_B}$ (or equivalently $X \leq p_B \frac{q_A}{q_B}$) given that $p_B^* \leq X$ still holds. The market coverage of this case is presented in Figure 18 .

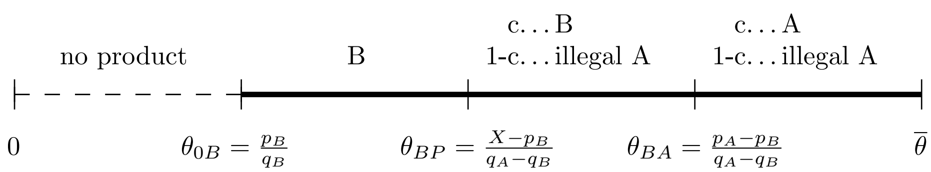

Figure 17: BC, when developer A introduces protection c (Case 1).

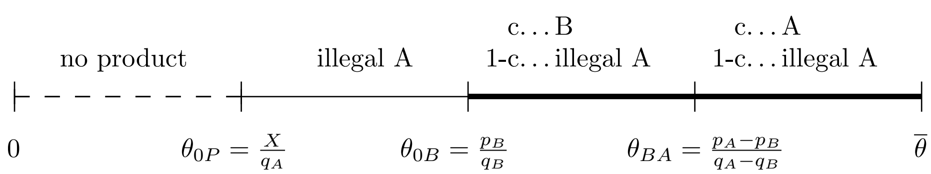

Figure 18: BC, when developer A introduces protection c (Case 2).

As for Sub-case 1, we obtain demand for legal versions of both products by putting all fraction of users together:

$$
\begin{aligned}
D_A &= c\left(\bar{\theta} - \frac{p_A - p_B}{q_A - q_B}\right), \\
D_B &= c\left(\frac{p_A - p_B}{q_A - q_B} - \frac{p_B}{q_B}\right) + (1 - c)\left(\frac{X - p_B}{q_A - q_B} - \frac{p_B}{q_B}\right) = \\
&= \frac{cp_A + (1 - c) X - p_B}{q_A - q_B} - \frac{p_B}{q_B}.
\end{aligned} \tag{90}
$$

In Sub-case 2, only the users without access to an illegal version of A buy product B so the demand functions are now:

$$D_A = c\left(\bar{\theta} - \frac{p_A - p_B}{q_A - q_B}\right),$$
$$D_B = c\left(\frac{p_A - p_B}{q_A - q_B} - \frac{p_B}{q_B}\right).$$

Note that Sub-case 2 is practically identical to the pure Bertrand case yielding the same equilibrium prices, see (74), and yielding the same market coverage as well as the equilibrium profits that are only sized down by factor c, see (75). Most importantly, small changes in the expected penalty have no impact on the size of optimal private protection nor on the other equilibrium values.

Thus, we focus on the more interesting Subcase 1. We start with determining the range of the expected penalty values X such that this sub-case is the Nash equilibrium in prices. Namely, sub-case 1 is not an equilibrium if (i) at least one developer's profit, given the other developer's price choice, does not have a local maximum in the relevant price range. Moreover, it is also not an equilibrium if (ii) there is a local maximum in the relevant range, but at least one developer is better off deviating to a price outside the range (e.g., developer A can be better off deviating to $p_A = X$). Intuitively for developer A to charge a high price $p_A > X$, the value of X should be small enough so that developer A prefers introducing protection than to simply lowering the price to X. As for developer B to charge a low price $p_B < X\frac{q_B}{q_A}$, X should be large enough so that developer B prefers charging a low price to both charging an intermediate price $X\frac{q_B}{q_A} \leq p_B \leq X$ and introducing protection or charging a high price $p_B > X$.

For (i), we show in Appendix C.3.4 that a necessary condition on X is $X_{cl} < X < X_{cu}$, where $X_{cl} = \frac{\bar{\theta} c q_A (q_A - q_B)}{2(1+c)q_A - c q_B}$, and $X_{cu} = 2\bar{\theta} q_A \frac{q_A - q_B}{4q_A - q_B}$; note that the upper bound X_{cu}, intuitively, coincides with the equilibrium price p_A^o from the case of the pure Bertrand equilibrium (74). Then both developers' profits reach the internal local maxima in the price ranges corresponding to our sub-case, with the prices equal to

$$p_A^* = \frac{X\left(1-c\right) q_B + 2\bar{\theta} q_A \left(q_A - q_B\right)}{4q_A - c q_B},\ p_B^* = q_B \frac{2X\left(1-c\right) + \bar{\theta} c \left(q_A - q_B\right)}{4q_A - c q_B}. \tag{91}$$

For (ii), we verify that neither developer has an incentive to unilaterally deviate given that the other developer sets the equilibrium price, p_i^*. For developer A, it can be profitable to deviate to $p_A = X$ (given that developer B sets p_B^*) if the decrease in price from p_A^* to X is more than compensated for by an increase in the number of consumers that is no longer confined to fraction c, and for X large enough, such a deviation would yield a higher profit even without protection costs, $h(c) = 0$. As for developer B, if p_B^* is close enough to $X\frac{q_B}{q_A}$, then it may pay off to jump to a higher price $p_B \in (X\frac{q_B}{q_A}, X)$ given that developer A sets p_A^* as in this case, the effect of such a price increase would more than offset the loss of the consumer base. The analysis in Appendix C.3.4 shows that for an equilibrium to exist, X should not be "too large" for developer A, so that $X < X_c^+ < X_{cu}$, nor should it be "too small" for developer B, so that $X > X_c^- > X_{cl}$. While values X_{cl} and X_{cu} always define a non-empty range, the condition $X_c^- < X < X_c^+$ defines a non-empty set only if $c > \frac{\sqrt{5}-1}{2} \approx 0.618034$, and if the quality ratio is not too high[44], then the lower bound on c can be

[44] Here "not too high" means that q_B/q_A is below the threshold value, which is itself above 0.9, so we can be almost sure that this is the case and consider it as the general situation.

improved to $c > \underline{c} \approx 0.704402$. If $X \in (X_c^-, X_c^+)$, then none of the developers have an incentive to deviate, and the prices above constitute an equilibrium.

As for the comparative statics analysis with respect to c, it is straightforward to show that equilibrium prices $p_A^*(c)$, $p_B^*(c)$ and the profit $\pi_B^*(c)$ increase as the level of physical protection c increases, so developer A acts strategically and softens the price competition and (in jargon) displays pacifistic "fat cat" behavior (see Fudenberg and Tirole, 1984).

We now focus on our key issue of how private and public protection interact. More specifically, we study the effect of the expected penalty X on the optimal developer's protection, c^*. The direction of this effect is determined by the impact of the expected penalty on the marginal profitability of private protection (more technically, on the sign of $\frac{\partial^2 \pi_A^*}{\partial c \partial X}$) and also by the existence of the interval (X_c^-, X_c^+). As we stated above, the necessary condition for that interval to be non-empty is that $c^* > \frac{\sqrt{5}-1}{2}$, and this, in turn, implies (or is sufficient for) $\frac{\partial^2 \pi_A^*}{\partial c \partial X} < 0$ entailing that the rise in the expected penalty decreases marginal profitability. This situation is described in jargon as "strategic substitutability" between c^* and X so that $\frac{dc^*}{dX} < 0$.

Proposition 12 *Private and public protection are always strategic substitutes, that is,* $\frac{dc^*}{dX} < 0$.

Proof. see Appendix C.3.5

■

With c^* being "large," the "cost effect" dominates the effect on revenue in the sense that the gains of additional public protection are lower than the ensuing private costs of protection. So developer A cuts back his private protection in response to the increased public protection, decreasing thus its private protection costs, and harming developer B (recall that $\frac{d\pi_B^*(c)}{dc} > 0$).

The nature of the interaction between the private and public IPR protection enables us to further study the comparative statics effects of X on equilibrium prices and profits.

Lemma 17 *The effect of X on $p_A^*(X)$ and $p_B^*(X)$ is a priori undetermined.*

Proof. Note that $\frac{dp_i}{dX}(c(X), X) = \frac{\partial p_i}{\partial c}\frac{dc}{dX} + \frac{\partial p_i}{\partial X}$. Straightforward differentiation shows that direct effect of X on prices is positive, that is, $\frac{\partial p_i}{\partial X} > 0$. From the analysis above, we know that $\frac{\partial p_i}{\partial c} > 0$, but $\frac{dc}{dX} < 0$. Thus, the indirect effect, $\frac{\partial p_i}{\partial c}\frac{dc}{dX} < 0$. ■

Lemma 18 *The effect of X is positive on $\pi_A^*(X)$ but the respective effect on $\pi_B^*(X)$ is a priori unclear.*

Proof. *Note that* $\frac{d\pi_A^*(X)}{dX}(c(X), X) = \frac{\partial \pi_A^*}{\partial X} > 0$. *Note further that* $\frac{d\pi_B^*(X)}{dX}(c(X), X) = \frac{\partial \pi_B^*(X)}{\partial c}\frac{dc^*}{dX} + \frac{\partial \pi_B^*}{\partial X}$, *where* $\frac{\partial \pi_B^*(X)}{\partial c}\frac{dc}{dX} < 0$ *since* $\frac{dc^*}{dX} < 0$ *and* $\frac{\partial \pi_B^*}{\partial X} > 0$. *Thus, the direct and indirect effects have a conflicting impact on developer B's profit.* ■

As we see, developer A reacts aggressively on an increase in X and cuts back in his private protection in response to increased public protection. As for developer B, if the net outcome of the above two conflicting (direct and indirect) effects is negative, the profit of developer B and equilibrium prices fall making price competition tougher. As a result, a "fat cat" strategy in this case becomes a bit diluted due to the enhanced public protection while, on the other hand, consumers of both goods benefit due to the decrease in equilibrium prices[45].

4.3.2 Both developers A and B implement protection

If the regulator sets up a very low expected penalty ($X \leq p_B^* \leq p_A^*$), then, naturally, both developers have to either implement physical protection or decrease prices to X; otherwise, they would be out of the market.

We denote protection used by developer A as c_A and protection used by developer B is c_B. Further, we assume that users may have access either to an illegal version of product A, an illegal version of product B, or to both illegal versions. Moreover, we assume that access to an illegal version of product A and B are mutually independent so there are users on the market that have access to illegal versions of product A but not to illegal versions of product B and vice versa. Then there are the following fractions of users on the market:

1. $c_A c_B$ ... The fraction of users with access only to legal products;
2. $c_A(1-c_B)$ ... The fraction of users with access to an illegal version of product B;
3. $(1-c_A)c_B$ ... The fraction of users with access to an illegal version of product A;
4. $(1-c_A)(1-c_B)$ ... The fraction of users with access to illegal versions of both products.

We have now the following types of users:

1. $\theta \in (\frac{p_A-p_B}{q_A-q_B}, \bar{\theta})$ ... Users who buy product A if they do not have access to any illegal version;
2. $\theta \in (\frac{p_B}{q_B}, \frac{p_A-p_B}{q_A-q_B})$ ... Users who buy product B if they do not have access to an illegal version of A;
3. $\theta \in (\frac{X}{q_A}, \bar{\theta})$ ... Users who use an illegal version of A if they have access to it;
4. $\theta \in (\frac{p_A-X}{q_A-q_B}, \bar{\theta})$ ... Users who buy A if they have access only to an illegal version of B.

Given the above set-up, it seems that two sub-cases could arise. The first one would be such that $\bar{\theta} \leq \frac{p_A-X}{q_A-q_B}$, implying that there is no user who would buy product A if he has illegal access to product B. This, however, never occurs since in the equilibrium, developer A sets the price low enough that users with θ close to $\bar{\theta}$ always prefer to buy the legal version of A (see Appendix C.4.1). The second situation appears when $\frac{p_A-X}{q_A-q_B} < \bar{\theta}$, implying that such users exist, and their number is higher than zero. So next, we discuss this only feasible sub-case.

Both competitors introduce physical protection and $\frac{p_A-X}{q_A-q_B} < \bar{\theta}$. In this case, there are users who prefer the legal version of the higher quality product q_A even though they have access to the illegal version of product B, but not of product A. This leads to the following market coverage:

From the distribution of users on the market, we obtain the following demand for the individual products:

$$
\begin{aligned}
D_A &= c_A c_B \left(\bar{\theta} - \frac{p_A - p_B}{q_A - q_B}\right) + c_A(1-c_B)\left(\bar{\theta} - \frac{p_A - X}{q_A - q_B}\right) \\
&= \frac{c_A\left(X(1-c_B) + \bar{\theta}(q_A - q_B) + c_B p_B - p_A\right)}{q_A - q_B}, \\
D_B &= c_A c_B \left(\frac{p_A - p_B}{q_A - q_B} - \frac{p_B}{q_B}\right).
\end{aligned} \tag{92}
$$

[45]It is straightforward to show that entry deterrence by means of c is not feasible in the set-up under consideration.

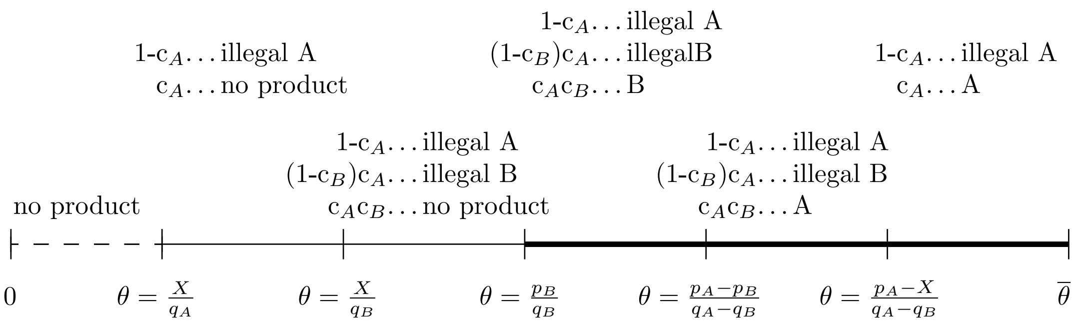

Figure 19: Both developers introduce protection, and $\frac{p_A - X}{q_A - q_B} < \bar{\theta}$

As in the previous section, we start with determining the range of the expected penalty values X such that this sub-case is a Nash equilibrium in prices. Recall that for the existence of a price equilibrium in the case when only developer A adopts protection, X has to be low enough from the perspective of developer A, but it has to be high enough from the view point of developer B. Now in the case under consideration, there are no such opposing requirements on X, since for both developers to charge high prices (above X), they both "need" X to be low[46]. Intuitively, if X is close to zero, then both developers would implement protection and charge prices above X rather than adjust their prices to X or below. We show in Appendix C.4.4 that a strictly positive $\underline{X} < \underline{X}_0 = \frac{\bar{\theta} q_B (q_A - q_B)}{4q_A - q_B}$ (note that $\underline{X}_0$ equals p_B^o of the pure Bertrand equilibrium) exists such that the following prices constitute an equilibrium:

$$p_A^* = 2q_A \frac{\bar{\theta}(q_A - q_B) + X(1 - c_B)}{4q_A - c_B q_B}, \tag{93}$$
$$p_B^* = \frac{\bar{\theta}(q_A - q_B) + X(1 - c_B)}{4q_A - c_B q_B} q_B.$$

As for a comparative statics analysis with respect to c_A and c_B, it is straightforward to show that equilibrium prices do not depend on c_A and increase in c_B. While the positive effect of c_B is not unexpected, the independence of the equilibrium prices on c_A might seem less intuitive. However, if both developers charge prices above X, any consumer not controlled by developer A would use an illegal version of product A, and a small change in c_A would only have a market size effect, i.e., both demands would change proportionally to the change in c_A. As there are no production costs, the change in marginal incentives will be also proportional to the change in c_A, so that the prices do not change. Note also that both developers prefer the good protection of a competitor's product, that is $\frac{\partial \Pi_A^*}{\partial c_B} > 0$ and $\frac{\partial \Pi_B^*}{\partial c_A} > 0$. The intuition is that an increase in either c_A or c_B increases the number of legal users for both developers, as it could be seen by visual inspection that $\frac{\partial D_A}{\partial c_B} > 0$ and $\frac{\partial D_B}{\partial c_A} > 0$ and also by looking at the market coverage in Figure 19 .

Before proceeding to the central issue of our analysis—the interaction between the private and public IPR protection—we make two additional assumptions: 1) $\left|\frac{\partial^2 \pi_i^*}{\partial c_i \partial c_i}\right| > \left|\frac{\partial^2 \pi_i^*}{\partial c_i \partial c_j}\right|$, and 2) $c_B \leq \frac{1}{2}$. Assumption 1)

[46] Certainly, if the developers could costlessly choose X, they would set it sufficiently high to exclude illegal use, so "need" is used in the sense of pure mathematical conditions for an equilibrium in the given range. Also note that since these mathematical conditions for both developers stipulate an upper bound, the analysis is to some extent simpler than in the case of developer A alone implementing protection as it is impossible that the intersection of conflicting requirements on X results in an empty set.

is a rather standard implying the uniqueness of the equilibrium values of c_A^* and c_B^* as well as its stability. As for assumption 2), we argue here that the most plausible optimal values of c_B are in the range of $\left(0, \frac{1}{2}\right)$. The reason for this is rather a tough price competition in the vertically differentiated market. Consequently, the lower quality producer charges a substantially lower price and usually earns only a small fraction of the high-quality developer's profit in equilibrium. Thus, developer B cannot afford to expand c_B much above zero due to the increasing marginal cost of private protection (recall that $\frac{\partial^2 h}{\partial c_i^2}(c_i) > 0$).

Proposition 13 *Let the protection cost function $h(c)$ be such that assumptions 1) and 2) above hold, then an increase in X leads to an increase in the optimal protection of both developers, that is, $\frac{dc_A^*}{dX} > 0$ and $\frac{dc_B^*}{dX} > 0$. Thus, private and public IPR protections are strategic complements.*

Proof. see Appendix C.4.6

■

The sign and the size of interaction between the public and private IPR protection, $\frac{dc_i^*}{dX}$, depends on the impact of the expected penalty, X, on the marginal profitability of both developers' private protection, or, more technically, on the signs of both $\frac{\partial^2 \pi_A^*}{\partial c_A \partial X}$ and $\frac{\partial^2 \pi_B^*}{\partial c_B \partial X}$. It turns out that $\frac{\partial^2 \pi_A^*}{\partial c_A \partial X} > 0$ for all permissible values, and $\frac{\partial^2 \pi_B}{\partial c_B \partial X} > 0$ for (at least) all values of c_B such that $c_B \leq \frac{1}{2}$ (see Appendix C.4.6).

So in the situation when the expected penalty is low (that is, $X \in (0, \underline{X}]$), there is strategic complementarity not only between the private and public protections but also between the two private protections that reinforce each other (recall that $\frac{\partial^2 \pi_i^*}{\partial c_A \partial c_B} > 0$). In this case, an increase in the private protection of one developer induces the increase in the optimal protection of the other developer. Thus, the "cost effect" is not dominant here unlike in the case when only the high-quality developer adopts protection (see section 4.3.1) because here an increase in X leads to an increase of both c_A and c_B causing an upward spiral in private protections until the new equilibrium is reached.

As before, the nature of the interaction between private and public IPR is the key ingredient in analyzing the comparative statics effects of X on equilibrium prices and profits.

Lemma 19 *An increase in X leads to a rise in both prices and profits for both developers.*

Proof. Directly from equilibrium prices (91) and from profit comparison (in Mathematica file). ■

Note also that as both protections c_A, c_B tend to perfect protections, the equilibrium prices and profits go to profit from pure Bertrand competition.

4.4 Key section results

In this section, we concentrated on how the change in expected penalty affects developers' equilibrium values when producers implement physical protection. Predictably, the initial size of the expected penalty plays the decisive role in shaping the behavior of the market participants. We concentrate on the cases where X has an impact on the optimal protection c^* at the margin.

Thus, if X zero or small, then both developers introduce protection, and an increase in X reinforces $c_A$and c_B, that is, $\frac{dc*_A}{dX} > 0$ and $\frac{dc_B^*}{dX} > 0$. This means the regulator's and developers' IPR are strategic complements. It is important to note that even for a zero or low expected punishment, it is never the case that all of the users that have access to the illegal versions would use only these illegal versions in equilibrium. (If this were the case, X would have no impact on the users' and consequently on the the developers' decisions on either c_A or c_B.) Thus, in an equilibrium with low X, some of the users with a high appreciation for quality who have illegal access to product B would still buy legal versions of product A. An increase in X would make product A more attractive for those users. As an optimal response, developer A would increase c_A that would in turn lead to larger profit. At the same time an increase in X would leave more room for developer B to increase his prices and profit via an increase in c_B.

For some intermediate values of X, only A introduces protection. Here, however, an increase in X leads to a direct increase in competitor B's demand, and thus has a substantially larger impact on B's price and profit than on A's corresponding values. So it is optimal for A to decrease c^* as a response to an increase in X harming competitor B and improving A's profit by lowering his protection costs, $h(c)$. So, the regulator's and the developers' IPR are strategic substitutes, that is $\frac{dc*}{dX} < 0$ and this case, as we showed, appears only for a large enough c^*.

When $X \geq p_A^o$, there is no need for protection by any developers, so the regulator's IPR protection is in a sense an effective full substitute for the private developers' IPR protection.

Finally, we omit the Stackelberg competition as it happens to produce no new insights than those of the Bertrand competition.

5 Conclusion

In this essay, we study the interaction between the two instances of IPR protection in a duopoly software market. The first instance is associated with the level of a government's or regulator's protection that comes in the form of an expected penalty for violating IPR. The second instance represents the private IPR protection at the level of the developer. The latter appears in two forms: i) a restriction of additional consumer services for the illegal users and ii) in the physical protection of software. While i) discourages illegal usage and makes it less attractive, ii) makes illegal usage harder. Thus, we examine the market equilibria with the above two forms of developer protection. Before that, we considered as a benchmark case the situation when developers do not use any form of IPR.

We show that the expected penalty may affect both the market coverage and the corresponding market equilibria in all considered set-ups. In the benchmark case, for instance, when the developers do not implement any protection and the level of the expected penalty is low enough, the expected penalty serves as a price regulation instrument putting the cap on the price. Furthermore, the low expected penalty may force one of the developers, mainly the one with the lower product quality, to leave the market and establish the second one as the monopolist. In the case of a high expected penalty, where no user has the incentive to use a product illegally, it does not play any role, no matter whether the developers use IPR protection or not.

In the case of medium and low levels of an expected penalty when developers implement some form of protection, the resulting effect of the expected penalty crucially depends on the framework under consideration.

Thus, if the protection based on restriction of services happens to be the developer's optimal choice, we show that the illegal users of the product may recruit themselves either from price sensitive users (the low-end of the market) or from the middle part of the market. In the latter case, the illegal users create a "buffer" between the two groups of legal users, the one with the highest valuation for quality and the other with the lowest preference for quality. In this case, a marginal price change of one developer does not affect the profit of the other developer and, moreover, the high-quailty developer generates the same profit as if he were a monopolist constrained only by the size of the expected penalty. In any case, when firms protect their IPR by means of service restrictions, the expected penalty has an impact on market conduct and the developer's IPR protection only if it exceeds or goes below a certain threshold.

In the case where the protection comes in the form of physical protection, however, the very marginal change in the expected penalty in general affects the developers' optimal choice. Furthermore, when there is an implementation of physical protection against copying, the expected penalty, depending on its size and on the particular set-up, can be either a complement or a substitute to the developers' IPR protection.

We did not explicitly compare the two forms of private protections nor was it the aim of our analysis. It is clear, however, that the decision whether to implement physical protection or protection based on service restrictions, depends on the respective profitability of these two forms that in turn depend on the cost of implementing physical protection, the respective levels of such protection, and the height of the expected penalty for illegal usage. In the case of restricting product services, the high-quality developer seems to target better the users with the highest sensitivity to quality. More specifically, implementing physical protection instead of implementing a service restriction (or decreasing prices to an expected penalty), leads to losing some of the high-end (quality sensitive) users since a fraction of these users have access to an illegal version. (Note that the users with the highest quality are usually the most important source of a developer's profit). Moreover, implementing physical protection involves direct costs unlike the two other options. Thus, it seems that the physical protection would be optimal only if a developer can relatively cheaply achieve a high fraction of users who could use the product only legally and, when at the same time, the expected penalty is low enough, and the protection via additional services is not very effective.

As for the possible extensions of our analysis, the normative considerations would seem to be the most natural ones. In other words, the optimal regulator's choice of IPR protection and its economic impacts would be an issue. This would, in turn, require putting "more structure" in our model and consequently specifying the regulator's objective function. Since in our context, it was suitable to think of the two foreign developers competing on a third host market, the simplest case would be that the host regulator maximizes the consumer surplus net of the costs of implementing a particular level of expected penalty. This would further mean that the regulator would prefer to induce the most competitive set-up by means of the expected penalty, given the costs of reaching a particular level of expected penalty (whereby the costs of reaching a particular level are convex, that is above proportionally increasing in it). However, in our set-up where the users have access to an illegal version of the product, the choice of an optimal expected penalty seems to be trivial; in order to maximize the consumer surplus, the regulator will simply set the expected penalty to zero (or to some minimal level if zero is not feasible due to, say, an international standard and requirements for a minimal IPR protection). Thus, the set-up in which one or both developers are the domestic ones would be surely more interesting to analyze.

Another interesting extension would be to allow for the explicit trade-off between the increased developer IPR protection and the decreasing functionality of the product and to study the social welfare consequences and policy implications of such a trade-off.

References

Banerjee, D.S. (2003): "Software Piracy: A Strategic Analysis and Policy Instruments," *International Journal of Industrial Organization*, Vol. 21(1), pp. 97–127.

Belleflamme, P. (2001): "Oligopolistic Competition, IT Use for Software Differentiation and the Productivity Paradox," *International Journal of Industrial Organization,* Vol. 19, pp. 227–248.

Belleflamme, P., and Peitz, M. (2010) "Digital Piracy: Theory," CESifo Working Paper Series, No. 3222.

Belleflamme, P., and Picard, P.M. (2007): "Piracy and Competition," *Journal of Economics and Management Strategy*, Vol. 16, pp. 351–383.

Boom, A. (2001): "On the Desirability of Compatibility with Software Selection," *Journal of Industrial Economics*, Vol. 49, pp. 85–96.

Bulow, J., Geanakoplos, J., and Klemperer, P. (1985): "Multimarket Oligopoly: Strategic Substitutes and Strategic Complements," *Journal of Political Economy*, Vol. 93, pp. 488–511.

Choi, P., Bae, S.H., and Jun, J. (2010): "Digital Piracy and Firms' Strategic Interactions: The Effects of Public Copy Protection and DRM Similarity," *Information Economics and Policy,* Vol. 22, pp. 354–364.

Economides, N.(1992): "Competition and Integration among Complements and Network Market Structure," *Journal of Industrial Economics,* Vol. 40, pp. 105–123.

Economides, N. (1996): "The Economics of Networks," *International Journal of Industrial Organization*, Vol. 16, pp. 673–699.

Farrell, J., and Klemperer, P. (2006): "Coordination and Lock-In: Competition with Switching Costs and Network Effects," CPC UC Berkeley Working Paper, No. 179157.

Fudenberg, D., and Tirole, J. (1984): "The Fat Cat Effect, the Puppy Dog Ploy and the Lean and Hungry Look," *American Economic Review*, Vol. 74, pp. 361–368.

Jain, S. (2008): "Digital Piracy: A Competitive Analysis," *Marketing Science*, Vol. 27, pp. 610–626.

King, S.P., and Lampe, R. (2003): "Network Externalities, Price Discrimination and Profitable Piracy," *Information Economics and Policy*, Vol. 15, pp. 271–290.

Kúnin, M. (2004): "Why do Software Manufacturers Tolerate Piracy in Transition and Less Developed Countries?" CERGE-EI Working Paper, No. 231.

Kúnin, M., and Žigić, K. (2006): "Strategic Trade Policy and Vertical Product Differentiation: Intra-industry Trade between Developed and Developing Countries," CERGE-EI Working Paper, No. 230.

Minnitti, A., and Vergari, C. (2010): "Turning Piracy into Profits: A Theoretical Investigation," *Information Economics and Policy*, Vol. 22(4), pp. 379–390.

Peitz, M. and Waelbroeck, P. (2006): "Piracy of Digital Product: A Critical Review of the Theoretical Literature," *Information Economics and Policy*, Vol. 21(4), pp. 449–476.

Quélin, B.V., Abdessemed, T., and Bonardi, J.P. (2001): "Standardization of Network Technologies: Market Processes or the Result of Inter-firm Co-operation," *Journal of Economic Survey*, Vol. 15, pp.543–569.

Scotchmer, S. (2004): *Innovation and Incentives*, The MIT Press.

Shaked, A., and Sutton, J. (1984): *Natural Oligopolies and International Trade: An Introduction*, Oxford University Press.

Shapiro, C. (2003): "Antitrust Limits to Patent Settlements," *RAND Journal of Economics*, Vol. 34, pp. 391–411.

Shy, O., and Thisse, J. (1999): "A Strategic Approach to Software Protection," *Journal of Economics & Management Strategy*, Vol. 8(2), pp. 163–190.

Stavins, J. (1995): "Model Entry and Exit in a Differentiated-Product Industry: The Personal Computer Market," *The Review of Economics and Statistics*, Vol. 77, pp. 571–584.

Sutton, J. (1991): *Sunk Cost and Market Structure*, The MIT Press.

Takeyama, L.N. (2009): "Copyright Enforcement and Product Quality Signaling in Markets for Computer Software," *Information Economics and Policy*, Vol. 21(4), pp. 291–296.

Tirole, J. (1988): *The Theory of Industrial Organization*, The MIT Press.

Werden, G.J. (2001): "Network Effects and Conditions of Entry: Lessons from the Microsoft Case," *Antitrust Law*, Vol. 69, pp. 87–111.

Žigić, K. (2000): "Strategic Trade Policy, Intellectual Property Rights Protection, and North-South Trade," *Journal of Development Economics*, 2000, Vol. 61(1), pp. 27–60.

Internet Sources: www.tomshardware.com, www.idg.com, www.zive.cz, www.ifpi.org, www.ria.com, www.pro-music.org.

APPENDIX

A Basic Model

A.1 General notes for all appendices

Most of the calculations in this paper were performed using *Mathematica* and other similar software. The *Mathematica* file is available upon request.

In almost all model situations here, profit functions are concave (quadratic, or, in singular cases, linear) in the respective choice variables, so that an interior solution is always a (local) maximum. In the remaining situations, profit functions are explicitly assumed concave in the main text. Thus, second-order conditions always hold in equilibrium, so they are omitted everywhere below.

A.2 Indifferent users

From the user utility function it follows that indifferent users are characterized by the following quality sensitivities. The notation θ_{YZ}, where Y and Z can be one of $\{0, A, B\}$ implies that the users with $\theta < \theta_{YZ}$ strictly prefer Y to Z, and the users with $\theta > \theta_{YZ}$ strictly prefer Z to Y. Then

$$\theta_{0A} = \frac{p_A}{q_A},\, \theta_{0B} = \frac{p_B}{q_B},\, \theta_{BA} = \frac{p_A - p_B}{q_A - q_B}.$$

For the situations wherein developer B competes with either developer A's product priced at X or the illegal version thereof, also priced at X, we use the threshold $\theta_{BP} = \frac{X - p_B}{q_A - q_B}$.

A.3 Bertrand competition

A.3.1 Pure Bertrand competition

Profit functions are $\pi_A = \left(\bar{\theta} - \theta_{BA}\right) p_A$, and $\pi_B = \left(\theta_{BA} - \theta_{0B}\right) p_B$, and from F.O.C., it follows that

$$p_A^o = 2\bar{\theta} q_A \frac{(q_A - q_B)}{4q_A - q_B},\; p_B^o = \bar{\theta} q_B \frac{(q_A - q_B)}{4q_A - q_B},$$

so that the equilibrium profits are

$$\pi_A^o = 4\bar{\theta}^2 q_A^2 \frac{q_A - q_B}{\left(4q_A - q_B\right)^2},\, \pi_B^o = \bar{\theta}^2 q_A q_B \frac{q_A - q_B}{\left(4q_A - q_B\right)^2}.$$

A.3.2 Bertrand competition, where only developer B makes profit

The profit function of developer B is $\pi_B = \left(\theta_{BP} - \theta_{0B}\right) p_B$, so that

$$p_B^* = \frac{q_B}{2q_A} X,\, \pi_B^* = X^2 \frac{q_B}{4q_A\left(q_A - q_B\right)}. \tag{94}$$

A.3.3 Bertrand competition with binding price p_A equal to X

Developer A is limited to setting the price $p_A^* = X$. Thus, the profit functions are $\pi_A = \left(\bar{\theta} - \theta_{BP}\right) X$, and $\pi_B = \left(\theta_{BP} - \theta_{0B}\right) p_B$, so that p_B^*, π_B^* are the same as in (94), and

$$\pi_A^* = X \frac{2\bar{\theta} q_A\left(q_A - q_B\right) - X\left(2q_A - q_B\right)}{2q_A\left(q_A - q_B\right)}.$$

A.4 Stackelberg competition in prices

A.4.1 Stackelberg competition in prices

First assume that the condition $p_A \leq X$ is not binding. Then the profit functions are $\pi_A = \left(\bar{\theta} - \theta_{BA}\right) p_A$ and $\pi_B = (\theta_{BA} - \theta_{0B})\, p_B$, and developer B's reaction function is $p_B(p_A) = p_A \frac{q_B}{2q_A}$. Substituting this into π_A and maximizing, we obtain

$$p_A^* = \bar{\theta} q_A \frac{q_A - q_B}{2q_A - q_B},$$

so that

$$p_B^* = \frac{\bar{\theta}}{2} \frac{q_A - q_B}{2q_A - q_B} q_B,\ \pi_A^* = \bar{\theta} q_A^2 \frac{q_A - q_B}{2\,(2q_A - q_B)},\ \pi_B^* = \bar{\theta} q_A q_B^2 \frac{q_A - q_B}{4\,(2q_A - q_B)^2}.$$

Recall that if $p_A \leq X$ is binding, then the Stackelberg outcome coincides with the Bertrand outcome.

B Lower quality to illegal users

B.1 Indifferent users

As usual, the notation θ_{YZ}, where Y and Z can be one of $\{0, A, B, I\}$ implies that the users with $\theta < \theta_{YZ}$ strictly prefer Y to Z, and the users with $\theta > \theta_{YZ}$ strictly prefer Z to Y. Throughout this appendix, "product P" refers to the illegal version of product A.

As in the basic model, for thresholds not involving the illegal version of product A,

$$\theta_{0A} = \frac{p_A}{q_A},\ \theta_{0B} = \frac{p_B}{q_B},\ \theta_{BA} = \frac{p_A - p_B}{q_A - q_B}.$$

For thresholds involving product P but not involving product B,

$$\theta_{0P} = \frac{X}{\alpha q_A},\ \theta_{PA} = \frac{p_A - X}{q_A - \alpha q_A}.$$

As for the threshold between B and P, two cases have to be distinguished. First, the quality reduction to illegal users can be relatively low so that P is still better than B, i.e., $q_B < \alpha q_A$. Second, the quality reduction to illegal users can be relatively high so that illegal A becomes worse than B, i.e., $q_B > \alpha q_A$. (If $q_B = \alpha q_A$, then it is impossible that both B and P are in the market, and we concentrate on the cases where all three products are present.) In the first case, users with their sensitivity below the threshold use B whereas those above use P, so we use notation θ_{BP}. In the second case, the situation is the opposite so we use notation θ_{PB}. These are equal to

$$\theta_{BP} = \frac{X - p_B}{\alpha q_A - q_B},\ \theta_{PB} = \frac{p_B - X}{q_B - \alpha q_A}.$$

(Mathematically, these two are identical.)

B.2 Monopoly

The relevant thresholds are θ_{0A}, θ_{0P}, and θ_{PA}. Two cases are possible. First, if $p_A \leq \frac{X}{\alpha}$, then $\theta_{PA} \leq \theta_{0A} \leq \theta_{0P}$ (equality holds everywhere or nowhere) so that P is out of the market and users buy either A or nothing. Second, if $p_A > \frac{X}{\alpha}$, then $\theta_{PA} > \theta_{0A} > \theta_{0P}$ so that both P and A are in the market as in Figure 13 .

The monopolist's profit can be shown to be unimodal, and three outcomes can be distinguished.

First, if $X \geq \frac{1}{2}\bar{\theta}\alpha q_A$, then the unconstrained monopoly price is such that the illegal product is ousted, so that

$$p_A^* = \frac{\bar{\theta} q_A}{2},\ \pi_A^* = \frac{\bar{\theta}^2 q_A}{4}.$$

Second, if $X < \bar{\theta}\alpha q_A \frac{(1-\alpha)}{2-\alpha}$, then both A and P are present so that

$$p_A^* = \frac{X + \bar{\theta} q_A (1-\alpha)}{2},\ \pi_A^* = \frac{1}{4}\frac{\left(X + q_A \bar{\theta}(1-\alpha)\right)^2}{q_A(1-\alpha)}.$$

Third, if $\bar{\theta}\alpha q_A \frac{(1-\alpha)}{2-\alpha} \leq X < \frac{1}{2}\bar{\theta}\alpha q_A$, then while the monopolist has to lower the price due to the possibility of illegal use, this illegal use is still eliminated at the optimum, namely

$$p_A^* = \frac{X}{\alpha},\ \pi_A^* = \frac{X}{\alpha}\left(\bar{\theta} - \frac{X}{\alpha q_A}\right).$$

B.3 Bertrand competition

B.3.1 Market structure

The following user distributions across products are possible depending on the prices.

Remark 23 *Unless the fixed costs are prohibitive, the developers can always choose their prices so that both legal products are in the market, so that we neglect the price combinations such that either A or B (or both) are out.*

If $p_A \leq \frac{X}{\alpha}$, then P is out of the market and the outcome is the same duopoly as in the basic model.

If $p_A > \frac{X}{\alpha}$, then P can be in the market, and it is necessary to distinguish between the general cases of $q_B < \alpha q_A$ and $q_B > \alpha q_A$ (we neglect the equality as singular). Let

$$p_B^P = \frac{X}{\alpha}\frac{q_B}{q_A},\ p_B^T = \frac{X(q_A - q_B) - p_A(\alpha q_A - q_B)}{q_A - \alpha q_A},$$

and note that $p_B^T \lesseqgtr p_B^P$ iff $q_B \lesseqgtr \alpha q_A$.

Case $q_B < \alpha q_A$: In this case, if $p_B \leq p_B^T$, then P is out of the market, and if $p_B^T < p_B < p_B^P$, then all three products are present and the market structure corresponds to Figure 14 , i.e., the relevant thresholds are θ_{0B}, θ_{BP}, and θ_{PA}.

Case $q_B > \alpha q_A$: In this case, if $p_B \leq p_B^P$, then P is out of the market, and if $p_B^P < p_B < p_B^T$, then all three products are present and the market structure corresponds to Figure 15 , i.e., the relevant thresholds are θ_{0P}, θ_{PB}, and θ_{BA}.

Remark 24 *In this paper, we concentrate on the cases where all three products, both the legal ones and illegal A, are in the market. Thus, we only consider equilibria such that $p_A^* > \frac{X}{\alpha}$, and p_B^* is strictly between p_B^T and p_B^P.*

B.3.2 Case $q_B < \alpha q_A$

The profit functions are $\pi_A = \left(\bar{\theta} - \frac{p_A - X}{q_A - \alpha q_A}\right) p_A$ and $\pi_B = \left(\frac{X - p_B}{\alpha q_A - q_B} - \frac{p_B}{q_B}\right) p_B$, so that

$$p_A^* = \frac{X + \bar{\theta} q_A (1-\alpha)}{2}, p_B^* = \frac{X q_B}{2\alpha q_A},$$
$$\pi_A^* = \frac{1}{4} \frac{\left(\bar{\theta} q_A (1-\alpha) + X\right)^2}{q_A (1-\alpha)}, \pi_B^* = \frac{1}{4} X^2 \frac{q_B}{\alpha q_A (\alpha q_A - q_B)}.$$

The conditions $p_A^* > \frac{X}{\alpha}$ and $p_B^T < p_B^* < p_B^P$ hold iff $X > 0$, and

$$X < X_{\alpha 1} = \frac{\bar{\theta} \alpha q_A (\alpha q_A - q_B)(1-\alpha)}{(2-\alpha)\alpha q_A - q_B}.$$

Both profits are decreasing in α when $0 < X < X_{\alpha 1}$.

B.3.3 Case $q_B > \alpha q_A$

The profit functions of the developers are $\pi_A = \left(\bar{\theta} - \frac{p_A - p_B}{q_A - q_B}\right) p_A$, and $\pi_B = \left(\frac{p_A - p_B}{q_A - q_B} - \frac{p_B - X}{q_B - \alpha q_A}\right) p_B$, so that

$$p_A^* = \frac{(q_A - q_B)\left(\bar{\theta} 2 q_A (1-\alpha) + X\right)}{(4q_A - 3q_A\alpha - q_B)},$$
$$p_B^* = (q_A - q_B) \frac{\bar{\theta}(q_B - q_A\alpha) + 2X}{(4q_A - 3q_A\alpha - q_B)},$$
$$\pi_A^* = (q_A - q_B) \left(\frac{2\bar{\theta} q_A (1-\alpha) + X}{4q_A - 3q_A\alpha - q_B}\right)^2, \text{ and}$$
$$\pi_B^* = (1-\alpha) \frac{(q_A - q_B) q_A}{(q_B - \alpha q_A)} \left(\frac{\bar{\theta}(q_B - \alpha q_A) + 2X}{4q_A - 3q_A\alpha - q_B}\right)^2.$$

The conditions $p_A^* > \frac{X}{\alpha}$ and $p_B^P < p_B^* < p_B^T$ hold iff $X \geq 0$, and

$$X < X_{\alpha 2} = \bar{\theta} \frac{(q_A - q_B)(q_B - q_A\alpha)\alpha q_A}{4q_A q_B - q_B^2 - 2\alpha q_A^2 - \alpha q_A q_B}.$$

Both profits are decreasing in α when $0 \leq X < X_{\alpha 2}$.

B.4 Stackelberg competition

The only relevant case is $q_B > \alpha q_A$. Developer B's reaction function is

$$p_B(p_A) = \frac{(X q_A - X q_B + p_A (q_B - \alpha q_A))}{2q_A - 2\alpha q_A}. \tag{95}$$

Substituting this into the profit function of developer A and solving for p_A, we obtain

$$p_A^* = \frac{1}{2} \frac{2\bar{\theta} q_A \alpha - 2\bar{\theta} q_A - X}{\alpha q_A - 2q_A + q_B} (q_A - q_B),$$

so that

$$p_B^* = \frac{1}{2} \frac{(q_A - q_B)\left(X (q_B - 4q_A + 3\alpha q_A) + \bar{\theta} 2 q_A (\alpha - 1)(q_B - \alpha q_A)\right)}{(2q_A - 2\alpha q_A)(q_B - 2q_A + \alpha q_A)},$$
$$\pi_A^* = \frac{1}{8} (q_A - q_B) \frac{\left(2\bar{\theta} q_A (1-\alpha) + X\right)^2}{(2q_A - \alpha q_A - q_B) q_A (1-\alpha)}, \text{ and}$$
$$\pi_B^* = \frac{1}{16 q_A} \frac{\left(2\bar{\theta} q_A (q_B - \alpha q_A)(1-\alpha) + X (4q_A - 3q_A\alpha - q_B)\right)^2 (q_A - q_B)}{(-q_B + \alpha q_A)(-1+\alpha)(-2q_A + \alpha q_A + q_B)^2}.$$

The conditions $p_A^* > \frac{X}{\alpha}$ and $p_B^P < p_B^* < p_B^T$ hold iff $X \geq 0$, and

$$X < X_{\alpha S} = \bar{\theta}\frac{2\left(q_A - q_B\right)\left(q_B - q_A\alpha\right)\alpha(1-\alpha)q_A}{(8-7\alpha+\alpha^2)q_Aq_B - (4-3\alpha)(q_B^2+\alpha q_A^2)}.$$

Both profits are higher than under Bertrand competition and decrease in α when $0 \leq X < X_{\alpha S}$.

C Developers implement physical protection

C.1 Indifferent users

As usual, the notation θ_{YZ}, where Y and Z can be one of $\{0, A, P, B, I\}$ implies that the users with $\theta < \theta_{YZ}$ strictly prefer Y to Z, and the users with $\theta > \theta_{YZ}$ strictly prefer Z to Y. Throughout this appendix, "product P" refers to the illegal version of product A, and "product I" refers to the illegal version of product B.

As in the basic model, for thresholds not involving the illegal products,

$$\theta_{0A} = \frac{p_A}{q_A},\ \theta_{0B} = \frac{p_B}{q_B},\ \theta_{BA} = \frac{p_A - p_B}{q_A - q_B}.$$

For thresholds involving product P, note that all consumers prefer P to I, and the decision between P and A is made on the basis of prices alone. The remaining thresholds are

$$\theta_{0P} = \frac{X}{q_A},\ \theta_{BP} = \frac{X - p_B}{q_A - q_B}.$$

For thresholds involving product I, note that the decision between I and B is made on the basis of prices alone. The remaining thresholds are

$$\theta_{0I} = \frac{X}{q_B},\ \theta_{IA} = \frac{p_A - X}{q_A - q_B}.$$

Also recall that the illegal products are available only to the fractions of consumers not controlled by the corresponding firms.

C.1.1 The price-quality ratio rule

The following general result can be easily shown to hold.

Lemma 20 *If there is a good of quality q_A available at price p_A and a good of quality $q_B < q_A$ available at price p_B, then a necessary condition exists for consumers to buy good B, namely the price per unit of quality is strictly lower for the lower quality good, i.e., $\frac{p_B}{q_B} < \frac{p_A}{q_A}$.*

Proof. The claim directly follows from $\theta_{BA} - \theta_{0B} > 0$. ■

This result was implicitly used in previous chapters, and the equilibrium prices complied with it. However, in this chapter, profit functions are not unimodal, and an analysis of deviations requires the Lemma above explicitly.

Corollary 4 *No consumer with access to P prefers B to P if $p_B \geq X\frac{q_B}{q_A}$.*

Corollary 5 *No consumer with access to I prefers I to A if $p_A \leq X\frac{q_A}{q_B}$.*

C.2 Duopoly: general notes

Recall that the physical protection settings imply that every consumer is controlled by firm A with probability c_A, and independently by firm B with probability c_B. Thus, four groups of consumers exist.(In all cases, it is assumed that $\bar{\theta}$ is high enough.)

1. Consumers controlled by both firms, $c_A c_B$: These consumers view the market as a standard duopoly, so that the following applies according to the price-quality ratio rule.

   (a) If $\frac{p_B}{q_B} < \frac{p_A}{q_A}$, then the consumers with $\theta < \theta_{0B}$ use nothing, those with $\theta_{0B} < \theta < \theta_{BA}$ buy product B, and those with $\theta_{BA} < \theta < \bar{\theta}$ buy product A.

   (b) If $\frac{p_B}{q_B} \geq \frac{p_A}{q_A}$, then the consumers with $\theta < \theta_{0A}$ use nothing, and those with $\theta_{0A} < \theta < \bar{\theta}$ buy product A.

2. Consumers controlled by firm A alone, $c_A(1 - c_B)$: If $p_B \leq X$, then product I is irrelevant, and the outcome is a standard duopoly as in group 1. If $p_B > X$, then these consumers choose between A and I so that the following applies.

   (a) If $p_A > X\frac{q_A}{q_B}$, then the consumers with $\theta < \theta_{0I}$ use nothing, those with $\theta_{0I} < \theta < \theta_{IA}$ use product I, and those with $\theta_{IA} < \theta < \bar{\theta}$ buy product A.

   (b) If $p_A \leq X\frac{q_A}{q_B}$, then the consumers with $\theta < \theta_{0A}$ use nothing, and those with $\theta_{0A} < \theta < \bar{\theta}$ buy product A.

3. Consumers controlled by firm B alone, $(1 - c_A)c_B$: If $p_A \leq X$, then product P is irrelevant, and the outcome is a standard duopoly as in group 1. If $p_A > X$, then these consumers choose between P and B so that the following applies.

   (a) If $p_B < X\frac{q_B}{q_A}$, then the consumers with $\theta < \theta_{0B}$ use nothing, those with $\theta_{0B} < \theta < \theta_{BP}$ buy product B, and those with $\theta_{BP} < \theta < \bar{\theta}$ use product P.

   (b) If $p_B \geq X\frac{q_B}{q_A}$, then the consumers with $\theta < \theta_{0P}$ use nothing, and those with $\theta_{0P} < \theta < \bar{\theta}$ use product P.

4. Consumers controlled by neither firm, $(1 - c_A)(1 - c_B)$: The outcome in this group is the same as in group 3 due to the price-quality ratio rule. Namely, all consumers not controlled by firm A have access to a good of quality q_A at a price of no more than X. Then no such consumer will be interested in a product of quality q_B if offered at a price above $X\frac{q_B}{q_A} < X$, so it is irrelevant whether these consumers are controlled by firm B.

Thus, the last two groups can be united into a single group of those not controlled by A, with the total measure of c_A. Also note that if $p_A \leq X$, then the outcome is that of a standard duopoly as both illegal products are dominated by product A.

Note that in this model, the duopoly is always viable in the sense that the low-quality developer can always set a price such that the demand for B is strictly positive, e.g., $p_B = \frac{\min\{p_A, X\} q_B}{2 q_A}$. Therefore, situations such that developer B is out of the market, e.g., $p_B \geq p_A$, can be neglected except in reaction functions.

From the above, it follows that every consumer depending on the firms controlling and the relative position of the prices w.r.t. X, faces one of the following three situations.

- Case I: a standard duopoly, the choice between A at p_A and B at p_B.
- Case II: the choice between P at X and B at p_B.
- Case III: the choice between A at p_A and I at X.

The correspondence between these three cases, the consumer groups, and price settings, is the following ($p_B < p_A$ assumed).

	$p_A \leq X$	$p_B \leq X < p_A$	$X < p_B$
$c_A c_B$	I	I	I
$c_A(1 - c_B)$	I	I	III
$1 - c_A$	I	II	II[47]

The approach to equilibrium verification is the following. First, the reaction functions are investigated, where it is assumed that the other developer's price satisfies the given constraints, and then it is checked whether it is optimal for this developer to charge a price in the relevant range. Second, equilibrium prices are computed from the corresponding first-order conditions, and constraints on parameters are finalized. This approach is necessary as the profit functions feature discontinuity and non-unimodality.

C.3 Bertrand competition where only A implements protection $c_A = c$

As stated in Chapter 4, we are interested in the subcase $p_B < X \frac{q_B}{q_A}$, $X < p_A$.

C.3.1 Reaction function of developer A

Let $p_B < X \frac{q_B}{q_A}$. Then developer A's demand function is described by the following.

1. Case (D): If $X < p_A \leq p_B + \bar{\theta}(q_A - q_B)$, then the situation that we focus on in the main text takes place,
$$D_A = c\left(\bar{\theta} - \theta_{BA}\right).$$
2. Case (d): If $p_B \frac{q_A}{q_B} < p_A \leq X$, then the outcome is that of an unconstrained duopoly,
$$D_A = \bar{\theta} - \theta_{BA}.$$
3. Case (m): If $p_A \leq p_B \frac{q_A}{q_B}$, then developer A is unconstrained,
$$D_A = \bar{\theta} - \theta_{0A}.$$

Given the range of p_B, this demand function is continuous between cases (d) and (m) but not at $p_A = X$ unless $c = 1$. The resulting profit function $\pi_A = p_A D_A$ is unimodal between (d) and (m), and is discontinuous at $p_A = X$.

An interior solution in case (D) can occur only if

$$X < X_d = \frac{\bar{\theta}\left(q_A - q_B\right) q_A}{2q_A - q_B}.$$

(Note, however, that X_d is always larger than the pure Bertrand duopoly price, that is $X_d > p_A^o = X_{cu}$.)

In this case, the reaction function and the corresponding profit are given by

$$r_A\left(p_B\right) = \frac{\bar{\theta}\left(q_A - q_B\right) + p_B}{2},\ \pi_A\left(p_B\right) = \frac{c\left(\bar{\theta}\left(q_A - q_B\right) + p_B\right)^2}{4\left(q_A - q_B\right)},$$

and an interior solution in (D) implies here that the maximum outside (D) is reached at $p_A = X$. Therefore, the profit above has to be compared with the profit in case (d), which equals

$$\pi_A^d = X\left(\bar{\theta} - \frac{X - p_B}{q_A - q_B}\right).$$

While it is possible to make a direct comparison between $\pi_A(p_B)$ and π_A^d and obtain the conditions such that there is no deviation to (m), the calculation of it would be rather cumbersome, so we postpone it to the equilibrium analysis. However, it is immediately clear that the protection duopoly profit is higher at $X = 0$ unless $c = 0$.

C.3.2 Reaction function of developer B

Let $X < p_A$. Then developer B's demand function is described by the following.

1. Case (X): If $X\frac{q_B}{q_A} \leq p_B < X$, then no user not controlled by A buys B as all such users prefer P,

$$D_B = c\left(\theta_{BA} - \theta_{0B}\right).$$

2. Case (D): If $p_B < X\frac{q_B}{q_A}$, then the situation that we focus on in the main text takes place,

$$D_B = c\left(\theta_{BA} - \theta_{0B}\right) + (1 - c)\left(\theta_{BP} - \theta_{0B}\right).$$

Strictly speaking, this analysis should include situation $p_B < p_A - \bar{\theta}(q_A - q_B)$, but in equilibrium $p_A < \bar{\theta}(q_A - q_B)$, so this can be neglected.

This demand function is continuous; however, the resulting profit function $\pi_B = p_B D_B$ is generally non-unimodal between (X) and (D).

An interior solution in case (D) occurs if $p_A < \left(1 + \frac{1}{c}\right) X$, in which case the reaction function and the corresponding profit are given by

$$r_B\left(p_A\right) = \frac{q_B}{2q_A}\left(cp_A + (1 - c)X\right),\ \pi_B\left(p_A\right) = \frac{q_B\left(cp_A + (1 - c)X\right)^2}{4q_A\left(q_A - q_B\right)}.$$

However, in (X), where the reaction function is the pure Bertrand reaction function $r_B\left(p_A\right) = \frac{q_B}{2q_A}p_A$, the condition $X\frac{q_B}{q_A} \leq p_B < X$ means that an interior maximum occurs if $2X < p_A < 2\frac{q_A}{q_B}X$, so that π_B is not unimodal around $p_B = X\frac{q_B}{q_A}$ if $2X < p_A < \left(1+\frac{1}{c}\right)X$. If the constraint $p_B \leq X$ is neglected, then the global maximum of π_B is attained in (D) when $p_A \leq \left(1+\frac{1}{\sqrt{c}}\right)X$. Then it can be shown that if $\left(1+\frac{1}{\sqrt{c}}\right)X \leq 2\frac{q_A}{q_B}X$, i.e., if $c \geq \left(\frac{q_B}{2q_A-q_B}\right)^2$, then the condition $p_A \leq \left(1+\frac{1}{\sqrt{c}}\right)X$ for the global maximum in (D) is both necessary and sufficient. If $c < \left(\frac{q_B}{2q_A-q_B}\right)^2$, then the global maximum occurs in (D) for $p_A \leq \bar{p}_A^D$, where $\left(1+\frac{1}{\sqrt{c}}\right)X < \bar{p}_A^D < \left(1+\frac{1}{c}\right)X$ and

$$\pi_B\left(\bar{p}_A^D\right) = \pi_B^X\left(\bar{p}_A^D\right) = cX\left(\frac{\bar{p}_A^D - X}{q_A - q_B} - \frac{X}{q_B}\right),$$

which is the profit from deviation to $p_B = X$.

C.3.3 Equilibrium calculation

Assuming that all conditions on the prices hold, the equilibrium prices and profits are the following.

$$\begin{aligned}
p_A^* &= \frac{2\bar{\theta}q_A\left(q_A-q_B\right)+X\left(1-c\right)q_B}{4q_A-cq_B},\\
p_B^* &= q_B\frac{2X\left(1-c\right)+\bar{\theta}c\left(q_A-q_B\right)}{4q_A-cq_B},\\
\pi_A^* &= c\frac{\left(2\bar{\theta}q_A\left(q_A-q_B\right)+q_BX(1-c)\right)^2}{\left(4q_A-q_Bc\right)^2\left(q_A-q_B\right)}, \text{ and}\\
\pi_B^* &= q_Aq_B\frac{\left(2X\left(1-c\right)+\bar{\theta}c\left(q_A-q_B\right)\right)^2}{\left(4q_A-q_Bc\right)^2\left(q_A-q_B\right)}.
\end{aligned}$$

C.3.4 Derivation of bounds on X and c

All conditions for these prices and profits to be interior local maxima are met if

$$c\frac{\bar{\theta}q_A(q_A-q_B)}{2(1+c)q_A-cq_B} = X_{cl} < X < X_{cu} = 2\frac{\bar{\theta}q_A(q_A-q_B)}{4q_A-q_B},$$

where $X < X_{cu}$ follows from $p_A^* > X$, and $X > X_{cl}$ follows from $p_B^* < X\frac{q_B}{q_A}$, with the latter equivalent to $p_A^* < X\left(1+\frac{1}{c}\right)$. (Note that $X_{cl} < X_{cu}$.) It remains to be checked whether these maxima are global, i.e., that no developer prefers switching to a price corresponding to another market structure.

Developer A can be shown *not* to switch to $p_A = X$ given $p_B = p_B^*$ if

$$X \leq X_c^+ = \frac{2\bar{\theta}q_A\left(q_A-q_B\right)\left(4q_A-c(2-c)q_B-\sqrt{1-c}\left(4q_A-cq_B\right)\right)}{16q_A^2-8q_Aq_B+\left(3c-3c^2+c^3\right)q_B^2},$$

which is smaller than X_{cu} when $c<1$. It turns out that $X_{cl} \lesseqgtr X_c^+$ iff $c \gtreqless \frac{\sqrt{5}-1}{2} \approx 0.618034$, i.e., the (sub)case in question cannot occur if $c \leq \frac{\sqrt{5}-1}{2}$.

As for developer B, cases $c \geq \left(\frac{q_B}{2q_A-q_B}\right)^2$ and $c < \left(\frac{q_B}{2q_A-q_B}\right)^2$ are distinguished. In the former case, the condition to check is $p_A^* \leq X\left(1+\frac{1}{\sqrt{c}}\right)$, which is equivalent to

$$X \geq X_c^- = 2\frac{\sqrt{c}\bar{\theta}q_A(q_A-q_B)}{(1+\sqrt{c})(4q_A-\sqrt{c}q_B)},$$

which is bigger than X_{cl} when $c<1$. It can be shown that $X_c^- \lesseqgtr X_c^+$ iff $c \gtreqless \underline{c}$, where

$$\underline{c} = \frac{1}{3}\left(4-8\left(6\sqrt{33}-26\right)^{-1/3}+\left(6\sqrt{33}-26\right)^{1/3}\right) \approx 0.704402,$$

so the lower bound on c can be improved to $\underline{c}$ when $c \geq \left(\frac{q_B}{2q_A-q_B}\right)^2$. In the other case, $c < \left(\frac{q_B}{2q_A-q_B}\right)^2$, a direct comparison between π_B^* and $\pi_B^X(p_A^*)$ yields a lower bound on X located between X_{cl} and X_c^-, which translates into a lower bound on c located between $\frac{\sqrt{5}-1}{2}$ and $\underline{c}$. Note that given the lower bounds on c, case $c \geq \left(\frac{q_B}{2q_A-q_B}\right)^2$ occurs with certainty if $\frac{q_B}{q_A}$ is not too high, namely, if $\frac{q_B}{q_A} \leq\approx 0.912622$.

C.3.5 The effect of X on c

By the implicit function theorem,

$$\frac{dc}{dX} = -\frac{\frac{\partial^2 \Pi_A}{\partial c \partial X}}{\frac{\partial^2 \Pi_A}{\partial c \partial c}},$$

so that the sign of $\frac{\partial c}{\partial X}$ is the same as the sign of:

$$\frac{\partial^2 \Pi_A^*}{\partial c \partial X} = 2q_B \frac{2\bar{\theta} q_A (q_A - q_B)(4q_A + cq_B - 8cq_A) + Xq_B(1-c)\left((4-12c)q_A + (c+c^2)q_B\right)}{(q_A - q_B)(4q_A - cq_B)^3}.$$

The sign of this expression depends on the sign of $(4q_A + cq_B - 8cq_A)$ and $\left((4-12c)q_A + (c+c^2)q_B\right)$. As $q_B < q_A$, both of these expressions can be shown to be negative for $c \geq \frac{4}{7} \approx 0.571429$. Since it is shown above that the subcase in question can occur only if $c \geq \frac{\sqrt{5}-1}{2} > \frac{4}{7}$, both $\frac{\partial^2 \Pi_A^*}{\partial c \partial X}$ and $\frac{dc}{dX}$ are negative.

C.3.6 The impact of X on prices and profits

First observe that $\frac{d\Pi_A^*}{dX}$ is clearly positive since $\frac{\partial \Pi_A^*}{\partial c} = 0$ at the point of optimum. Thus,

$$\frac{d\Pi_A^*}{dX} = \frac{\partial \Pi_A^*}{\partial c}\frac{dc}{dX} + \frac{\partial \Pi_A^*}{\partial X} = \frac{\partial \Pi_A^*}{\partial X} > 0.$$

In the case of developer B, the impact of X on developer B's profit is

$$\frac{d\Pi_B^*}{dX} = \frac{\partial \Pi_B^*}{\partial c}\frac{dc}{dX} + \frac{\partial \Pi_B^*}{\partial X}.$$

Since the indirect effect is negative and the direct one is positive, it cannot be told *a priori* which effect dominates. The same applies to both equilibrium prices.

C.4 Bertrand competition where both developers implement protection

As stated in Chapter 4, this case occurs if $X < p_B < p_A$.

C.4.1 The non-existence of subcase $p_A \geq X + \bar{\theta}(q_A - q_B)$

In this subcase, only the users controlled by both developers buy any legal products, so that the demands for the products are constant multiples of the standard duopoly demands, $D_A = c_A c_B\left(\bar{\theta} - \theta_{BA}\right)$ and $D_B = c_A c_B(\theta_{BA} - \theta_{0B})$. Therefore, if the solution is interior, then the equilibrium prices are identical to the standard duopoly equilibrium prices. In particular,

$$p_A^* = 2\bar{\theta} q_A \frac{q_A - q_B}{4q_A - q_B} < \bar{\theta}(q_A - q_B) \leq X + \bar{\theta}(q_A - q_B),$$

which is a contradiction. Hence, the solution must be corner with $\frac{\partial \pi_A}{\partial p_A} < 0$ at $p_A = X + \bar{\theta}(q_A - q_B) + 0$. However, it can be shown that this implies $\frac{\partial \pi_A}{\partial p_A} < 0$ at $p_A = X + \bar{\theta}(q_A - q_B) - 0$ as well (see the analysis of the profit and reaction functions below), so that $p_A \geq X + \bar{\theta}(q_A - q_B)$ is never optimal.

C.4.2 The reaction function of developer A

Let $X < p_B < \frac{q_B}{q_A}\left(X + \bar{\theta}\left(q_A - q_B\right)\right)$. (The upper limit on p_B here follows from $p_A < X + \bar{\theta}(q_A - q_B)$ and the price-quality ratio rule.) Then developer A's demand function is described by the following.

1. Case (d): If $p_A \geq X + \bar{\theta}\left(q_A - q_B\right)$, then all users of product A are completely controlled,
$$D_A = c_A c_B \left(\bar{\theta} - \theta_{BA}\right).$$
2. Case (D): If $p_B \frac{q_A}{q_B} < p_A < X + \bar{\theta}\left(q_A - q_B\right)$, then the situation that we focus on in the main text takes place,
$$D_A = c_A c_B \left(\bar{\theta} - \theta_{BA}\right) + c_A \left(1 - c_B\right)\left(\bar{\theta} - \theta_{IA}\right).$$
3. Case (I): If $X \frac{q_A}{q_B} < p_A \leq p_B \frac{q_A}{q_B}$, then no one uses B,
$$D_A = c_A c_B \left(\bar{\theta} - \theta_{0A}\right) + c_A \left(1 - c_B\right)\left(\bar{\theta} - \theta_{IA}\right).$$
4. Case (M): If $X < p_A \leq X \frac{q_A}{q_B}$, then no one uses B or I,
$$D_A = c_A \left(\bar{\theta} - \theta_{0A}\right).$$
5. Case (m): if $X \geq p_A$, then developer A is unconstrained,
$$D_A = \left(\bar{\theta} - \theta_{0A}\right).$$

Given the range of p_B, this demand function is continuous between cases (d) and (M) but not at $p_A = X$ unless $c_A = 1$. The resulting profit function $\pi_A = p_A D_A$ is strictly decreasing in p_A in (d), unimodal between (d) and (M), and is discontinuous at $p_A = X$.

Denote $X^A = X\left(1 - c_B\right) + \bar{\theta}\left(q_A - q_B\right)$. For cases (d), (D), (I), and (M), an interior solution in case (D) can occur only if
$$X < X_D = \frac{\bar{\theta}\left(q_A - q_B\right) q_B}{2q_A - q_B},\ X < p_B < p_B^D = \frac{q_B}{2q_A - c_B q_B} X^A.$$

In this case, the reaction function and the corresponding profit are given by
$$r_A\left(p_B\right) = \frac{X^A + c_B p_B}{2},\ \pi_A\left(p_B\right) = \frac{c_A\left(X^A + c_B p_B\right)^2}{4\left(q_A - q_B\right)}.$$

Now these values have to be compared with the monopoly profit in case (m). Since $X < X_D$ implies $X < \frac{\bar{\theta} q_B}{2}$ in the relevant case, the monopoly profit is maximized at the highest p_A in the range, i.e.,
$$\pi_A^m = X\left(\bar{\theta} - \frac{X}{q_A}\right).$$

While it is possible to make a direct comparison between $\pi_A(p_B)$ and π_A^m and obtain the maximal value $\bar{X}(p_B)$ such that there is no deviation to (m), the result is rather cumbersome. However, it is immediately clear that the duopoly profit is higher at $X = 0$.

C.4.3 The reaction function of developer B

Let $X\frac{q_A}{q_B} < p_A < X + \bar{\theta}\left(q_A - q_B\right)$. Then developer B's demand function is described by the following.

1. Case (D): If $X < p_B < p_A\frac{q_B}{q_A}$, then the situation that we focus on in the main text takes place,

$$D_B = c_A c_B \left(\theta_{BA} - \theta_{0B}\right).$$

2. Case (X): If $X\frac{q_B}{q_A} \leq p_B \leq X$, then no one uses I,

$$D_B = c_A \left(\theta_{BA} - \theta_{0B}\right).$$

3. Case (x): If $p_B < X\frac{q_B}{q_A}$, then there are consumers who prefer B to P (cf. the case when only A implements protection),

$$D_B = c_A \left(\theta_{BA} - \theta_{0B}\right) + \left(1 - c_A\right)\left(\theta_{BP} - \theta_{0B}\right).$$

Strictly speaking, this analysis should include situations $p_B < p_A - \bar{\theta}(q_A - q_B)$ and even $p_B < X - \bar{\theta}(q_A - q_B)$, but in equilibrium $X < p_A < \bar{\theta}(q_A - q_B)$, so these can be neglected.

This demand function is continuous between cases (X) and (x) but not at $p_B = X$ unless $c_B = 1$. The resulting profit function $\pi_B = p_B D_B$ is discontinuous at $p_B = X$ and can be non-unimodal between (X) and (x).

An interior solution in case (D) can occur only if $X < X_D$ (same as for developer A), in which case the reaction function and the corresponding profit have the same form as under a standard duopoly and are given by

$$r_B\left(p_A\right) = \frac{q_B}{q_A}\frac{p_A}{2},\ \pi_B\left(p_A\right) = c_A c_B \frac{p_A^2 q_B}{4q_A\left(q_A - q_B\right)}.$$

If the maximum in (D) is interior, then the maximum in (X) must be corner and the profit in (X) is maximized at $p_B = X$, i.e.,

$$\pi_B^X = c_A X\left(\frac{p_A q_B - X q_A}{(q_A - q_B)q_B}\right).$$

As for (x), the maximum is interior there if $p_A < X\left(1 + \frac{1}{c_A}\right)$, then $\pi_B^x = \frac{q_B(c_A(p_A - X) + X)^2}{4q_A(q_A - q_B)}$. It can be shown that if $p_A < X\left(1 + \frac{1}{c_A}\right)$ and $c_A > c_B$, then deviation to (x) from (D) is always profitable (note that deviation to (X) can be even more profitable). If $p_A \geq X\left(1 + \frac{1}{c_A}\right)$, then π_B strictly increases in p_B in (x), so that the maximal deviation profit is π_B^X above.

C.4.4 Equilibrium calculation

Assuming that all conditions on the prices hold, the equilibrium prices and profits are the following.

$$
\begin{aligned}
p_A^* &= 2q_A \frac{\bar{\theta}\,(q_A - q_B) + X\,(1 - c_B)}{4q_A - c_B q_B}, \\
p_B^* &= \frac{\bar{\theta}\,(q_A - q_B) + X\,(1 - c_B)}{4q_A - c_B q_B} q_B, \\
\pi_A^* &= 4c_A q_A^2 \frac{\left(\bar{\theta}\,(q_A - q_B) + X\,(1 - c_B)\right)^2}{\left(4q_A - q_B c_B\right)^2 (q_A - q_B)}, \text{ and} \\
\pi_B^* &= c_A c_B q_A q_B \frac{\left(\bar{\theta}\,(q_A - q_B) + X\,(1 - c_B)\right)^2}{\left(4q_A - q_B c_B\right)^2 (q_A - q_B)}.
\end{aligned}
$$

All condition for these prices and profits to be interior local maxima are met if

$$
X < \underline{X}_0 = \frac{\bar{\theta} q_B (q_A - q_B)}{4q_A - q_B}.
$$

It remains to check whether these maxima are global, i.e., that no developer prefers switching to a price corresponding to another market structure. As developer B will always switch to a price below $X\frac{q_B}{q_A}$ if $p_A < X\left(1 + \frac{1}{c_A}\right)$ and $c_A > c_B$, a necessary condition for no such deviation at $p_A = p_A^*$ is

$$
X < \frac{2c_A q_A (q_A - q_B)\bar{\theta}}{2(2 + c_A + c_A c_B) q_A - (1 + c_A) c_B q_B},
$$

which is below $\underline{X}_0$ when $c_A < \frac{q_B}{2q_A - q_B}$.

As for deviations to $p = X$ by either developer, let $\delta_A(X) = \pi_A^*(X) - \pi_A^m(X)$ and $\delta_B(X) = \pi_B^*(X) - \pi_B^X(X)$ be the differences between the duopoly and deviation profits. The functions $\delta_i(X)$ are positive at $X = 0$ and decreasing in X for $0 < X < \underline{X}_0$. If c_A is high enough, then it is possible that developer A does not switch for all applicable X; however, developer B always switches at $X = \underline{X}_0$, i.e. $\delta_B\left(\underline{X}_0\right) < 0$. From this, it follows that $\exists \underline{X}$, $0 < \underline{X} < \underline{X}_0$, such that the prices and profits above form an equilibrium.

C.4.5 The effect of protection on prices and profits

From the expressions for the equilibrium prices and profits, it is immediately seen that c_A has no effect on prices. By algebraic derivation it can be shown that if $X < \underline{X}_0$ (and recall that the actual boundary is $\underline{X} < \underline{X}_0$), then both equilibrium prices and the net profit $\Pi_A^* = \pi_A^* - h(c_A)$ increase in c_B, and that the net profit Π_B^* increases in c_A.

C.4.6 The effect of X on c_A and c_B

Applying the implicit function theorem, we obtain:

$$
\begin{aligned}
\frac{\partial \Pi_A^*}{\partial c_A}\left(c_A(X), c_B(X), X\right) \equiv 0 &\Longrightarrow \frac{\partial^2 \Pi_A^*}{\partial c_A \partial c_A}\frac{dc_A}{dX} + \frac{\partial^2 \Pi_A^*}{\partial c_A \partial c_B}\frac{dc_B}{dX} + \frac{\partial^2 \Pi_A^*}{\partial c_A \partial X} \equiv 0, \\
\frac{\partial \Pi_B^*}{\partial c_B}\left(c_A(X), c_B(X), X\right) \equiv 0 &\Longrightarrow \frac{\partial^2 \Pi_B^*}{\partial c_B \partial c_A}\frac{dc_A}{dX} + \frac{\partial^2 \Pi_B^*}{\partial c_B \partial c_B}\frac{dc_B}{dX} + \frac{\partial^2 \Pi_B^*}{\partial c_B \partial X} \equiv 0;
\end{aligned}
$$

or, in matrix form:

$$
\begin{pmatrix} \frac{\partial^2 \Pi_A^*}{\partial c_A \partial c_A} & \frac{\partial^2 \Pi_A^*}{\partial c_A \partial c_B} \\ \frac{\partial^2 \Pi_B^*}{\partial c_B \partial c_A} & \frac{\partial^2 \Pi_B^*}{\partial c_B \partial c_B} \end{pmatrix} \begin{pmatrix} \frac{dc_A}{dX} \\ \frac{dc_B}{dX} \end{pmatrix} = \begin{pmatrix} -\frac{\partial^2 \Pi_A^*}{\partial c_A \partial X} \\ -\frac{\partial^2 \Pi_B^*}{\partial c_B \partial X} \end{pmatrix}.
$$

For simplicity, denote the first matrix as H; thus, $H = \begin{pmatrix} \frac{\partial^2 \Pi_A^*}{\partial c_A \partial c_A} & \frac{\partial^2 \Pi_A^*}{\partial c_A \partial c_B} \\ \frac{\partial^2 \Pi_B^*}{\partial c_B \partial c_A} & \frac{\partial^2 \Pi_B^*}{\partial c_B \partial c_B} \end{pmatrix}$. Applying Cramer's rule:

$$\frac{dc_A}{dX} = \frac{|H_A|}{|H|} = \frac{1}{|H|} \begin{vmatrix} -\frac{\partial^2 \Pi_A^*}{\partial c_A \partial X} & \frac{\partial^2 \Pi_A^*}{\partial c_A \partial c_B} \\ -\frac{\partial^2 \Pi_B^*}{\partial c_B \partial X} & \frac{\partial^2 \Pi_B^*}{\partial c_B \partial c_B} \end{vmatrix},$$

$$\frac{dc_B}{dX} = \frac{|H_B|}{|H|} = \frac{1}{|H|} \begin{vmatrix} \frac{\partial^2 \Pi_A^*}{\partial c_A \partial c_A} & -\frac{\partial^2 \Pi_A^*}{\partial c_A \partial X} \\ \frac{\partial^2 \Pi_B^*}{\partial c_B \partial c_A} & -\frac{\partial^2 \Pi_B^*}{\partial c_B \partial X} \end{vmatrix}.$$

Differentiating the equilibrium profits yields $\frac{\partial^2 \Pi_A^*}{\partial c_A \partial c_A} = -h''(c_A) < 0$, $\frac{\partial^2 \Pi_A^*}{\partial c_A \partial X} > 0$, and $\frac{\partial^2 \Pi_A^*}{\partial c_A \partial c_B} > 0$ for $X < \underline{X}_0$, and by our assumptions $\frac{\partial^2 \Pi_B^*}{\partial c_B \partial c_B} < 0$ as well. As for $\frac{\partial^2 \Pi_B^*}{\partial c_B \partial c_A}$,

$$\frac{\partial^2 \Pi_B^*}{\partial c_B \partial c_A} = q_A q_B \left(\bar{\theta}(q_A - q_B) + X(1 - c_B)\right) \frac{\bar{\theta}(q_A - q_B)(4q_A + q_B c_B) - X\left(12 c_B q_A - c_B^2 q_B - 4q_A - q_B c_B\right)}{(4q_A - q_B c_B)^3 (q_A - q_B)},$$

which looks ambiguous, note that $\frac{\partial^2 \Pi_B^*}{\partial c_B \partial c_A} = \frac{\partial^2 \pi_B^*}{\partial c_B \partial c_A}$, and $\frac{\partial^2 \pi_B^*}{\partial c_B \partial c_A} = \frac{1}{c_A} \frac{\partial \pi_B^*}{\partial c_B}$; then, F.O.C. $\frac{\partial \Pi_B^*}{\partial c_B} = 0$ implies $\frac{\partial \pi_B^*}{\partial c_B} = h'(c_B)$, so that $\frac{\partial^2 \Pi_B^*}{\partial c_B \partial c_A} > 0$. Finally, for $\frac{\partial^2 \Pi_B^*}{\partial c_B \partial X}$,

$$\frac{\partial^2 \Pi_B^*}{\partial c_B \partial X} = -2 q_B q_A c_A \frac{\bar{\theta}(q_A - q_B)(4q_A(2c_B - 1) - q_B c_B) + (1 - c_B) X (4q_A(3c_B - 1) - q_B c_B(1 + c_B))}{(4q_A - q_B c_B)^3 (q_A - q_B)},$$

it can be shown that for $X < \underline{X}_0$ and $c_B \leq 1/2$, $\frac{\partial^2 \Pi_B^*}{\partial c_B \partial X} > 0$. While the condition $c_B \leq 1/2$ cannot be loosened, this is a typical situation that we expect to occur in equilibrium, in which clearly $c_B^* < c_A^*$. Thus, we postulate $c_B^* < 1/2$ so that $\frac{\partial^2 \Pi_B^*}{\partial c_B \partial X}(c_B^*) > 0$.

Now consider the matrix H and recall that $|H| = \frac{\partial^2 \pi_A^*}{\partial c_A \partial c_A} \frac{\partial^2 \pi_B^*}{\partial c_B \partial c_B} - \frac{\partial^2 \pi_A^*}{\partial c_A \partial c_B} \frac{\partial^2 \pi_B^*}{\partial c_B \partial c_A}$. The first term is always positive since $\frac{\partial^2 \pi_A^*}{\partial c_A \partial c_A} < 0$ and $\frac{\partial^2 \pi_B^*}{\partial c_B \partial c_B} < 0$. The second term is also always positive since $\frac{\partial^2 \pi_B^*}{\partial c_B \partial c_A} > 0$ and $\frac{\partial^2 \pi_A^*}{\partial c_A \partial c_B} > 0$. Thus, we make a standard stability assumption here that $\left|\frac{\partial^2 \pi_i^*}{\partial c_i \partial c_i}\right| > \left|\frac{\partial^2 \pi_i^*}{\partial c_i \partial c_j}\right|$, which ensures that $|H| > 0$. Given the above, the determinants $|H_A|$ and $|H_B|$ are positive, so that $\frac{dc_A}{dX} > 0$ and $\frac{dc_B}{dX} > 0$.

C.4.7 The effect of X on equilibrium prices and profits

As for the prices,

$$\frac{dp_A^*}{dX}(c_A(X), c_B(X), X) = \frac{\partial p_A^*}{\partial c_A} \frac{dc_A}{dX} + \frac{\partial p_A^*}{\partial c_B} \frac{dc_B}{dX} + \frac{\partial p_A^*}{\partial X},$$
$$\frac{dp_B^*}{dX}(c_A(X), c_B(X), X) = \frac{\partial p_B^*}{\partial c_A} \frac{dc_A}{dX} + \frac{\partial p_B^*}{\partial c_B} \frac{dc_B}{dX} + \frac{\partial p_B^*}{\partial X};$$

since $\frac{\partial p_A^*}{\partial c_A} = \frac{\partial p_B^*}{\partial c_A} = 0$, and the remaining terms are strictly positive (as is shown above or can be shown by direct differentiation), $\frac{dp_A^*}{dX} > 0$ and $\frac{dp_B^*}{dX} > 0$.

As for the profits,

$$\frac{d\Pi_A^*}{dX} = \frac{\partial \Pi_A^*}{\partial c_A} \frac{dc_A}{dX} + \frac{\partial \Pi_A^*}{\partial c_B} \frac{dc_B}{dX} + \frac{\partial \Pi_A^*}{\partial X},$$
$$\frac{d\Pi_B^*}{dX} = \frac{\partial \Pi_B^*}{\partial c_A} \frac{dc_A}{dX} + \frac{\partial \Pi_B^*}{\partial c_B} \frac{dc_B}{dX} + \frac{\partial \Pi_B^*}{\partial X};$$

by virtue of the envelope theorem, $\frac{\partial \Pi_A^*}{\partial c_A} = 0$ and $\frac{\partial \Pi_B^*}{\partial c_B} = 0$, and the remaining terms are again strictly positive, so that $\frac{d\Pi_A^*}{dX} > 0$ and $\frac{d\Pi_B^*}{dX} > 0$.