



The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search

<http://ageconsearch.umn.edu>

aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

No endorsement of AgEcon Search or its fundraising activities by the author(s) of the following work or their employer(s) is intended or implied.

TEL-AVIV

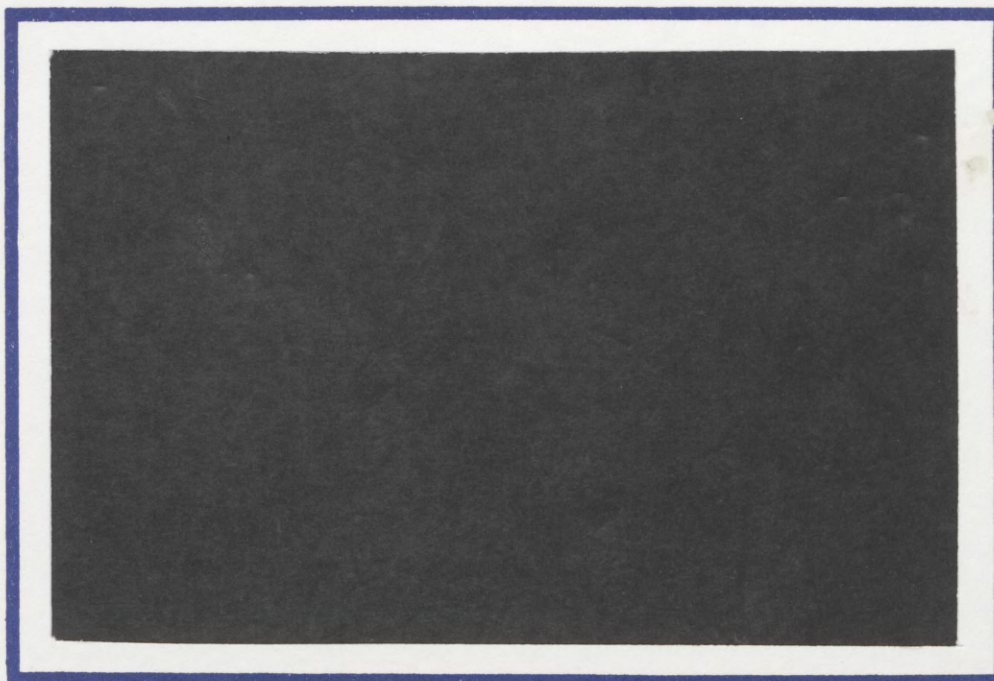
WP 5-91

WITHDRAWN

THE FOERDER INSTITUTE FOR ECONOMIC RESEARCH

TEL-AVIV UNIVERSITY

RAMAT AVIV ISRAEL



מכון למחקר כלכלי ע"ש ד"ר ישעיהו פורדר ז"ל

ע"י אוניברסיטת תל-אביב

COMPLEMENTARY NETWORK EXTERNALITIES
AND TECHNOLOGICAL ADOPTION

by

Jeffrey Church* and Neil Gandal**

Working Paper No.5-91

February, 1991

* University of Calgary
** Tel-Aviv University

This is a substantially revised version of our 1989 Boston University School of Management Working Paper #89-20, issued under the same title.

We wish to acknowledge the helpful comments of R.Gilbert, M.Katz, J.Perloff, S.Scotchmer, and two anonymous referees. We also wish to express our appreciation for the comments of seminar participants at Harvard, GTE labs, Boston College, University of Alberta, Boston University, University of Calgary, Canadian Economics Association Annual Meeting, Canadian Economic Theory Conference, and the Sixth World Congress of the Econometric Society. The authors gratefully acknowledge financial support provided by the Social Sciences and Humanities Research Council of Canada and partial financial support from a grant given to the Foerder Institute of Economic Research by the JOHN RAUCH FUND.

THE FOERDER INSTITUTE FOR ECONOMIC RESEARCH
Faculty of Social Sciences
Tel-Aviv University, Ramat Aviv, Israel

ABSTRACT

We address the adoption of technology when there are network externalities and networks are characterized by complementary products produced by different firms. We show that the market outcome is efficient if the software firms are monopolistic competitors. If the software firms are Bertrand competitors, a hardware technology with lower software development costs is adopted for many parameter values for which it is socially optimal to adopt the other technology. The over-adoption is due to a discrepancy between the private and social value of having a larger network: this divergence is increasing in the software development costs. We also examine various contractual arrangements between hardware firms and software firms which internalize the network externality.

1. Introduction

The benefit received from the consumption of a particular good often depends on the aggregate number of consumers who elect to purchase compatible goods. This positive consumption or network externality can be direct or indirect. If it is the former the utility of a consumer depends directly on the total number of subscribers to the same network. For instance, the value of access to a telephone network depends on the total number of consumers with similar access. Examples of goods where the network externality is indirect include many consumer electronic durables such as televisions, video cassette recorders, compact disc players, and personal computers. To be of value these durable or hardware goods require complementary software. The value of the hardware is enhanced as the variety of complementary software increases. Variety may in turn depend on the total number of consumers who purchase compatible hardware. The greater the sales of hardware, the greater the demand for software. This increases the profitability of supplying software. Entry by firms and the concomitant proliferation of software is the form of the network externality when it is indirect.

In this paper we formally model complementary network externalities. We show that with CES preferences over software, increasing returns to scale in the production of software, and free entry into the software industry the consumption benefit of hardware depends on the number of consumers purchasing the same hardware. Under the maintained assumption that software is not compatible between the two hardware technologies, we use this model to investigate how the interaction between the hardware and software firms affects the adoption prospects of two competing hardware technologies. Which technology prevails depends on the availability of software. If very little software is available even technically superior hardware will be relatively unattractive to consumers.

The compact disc digital audio system provides superior sound reproduction. Nevertheless, without the abundant provision of numerous recording artists on this new format by the record companies, compact discs would not be supplanting the phonograph. The success of compact discs can be contrasted with the failure of various quadraphonic audio technologies. The absence of a large variety of software halted the widespread adoption of these technologies.

Another example is personal computers. Despite its advanced capabilities and user-friendly interface, initial acceptance of the Macintosh computer was lukewarm. This was primarily attributable to the limited availability of software. All of Apple's new generations of

computers—the Apple III, the Lisa, and the Macintosh—were incompatible with the very large software base of the Apple II. This provided a window of opportunity for the IBM sanctioned MS-DOS based personal computer to supplant the Apple II as the dominant platform.

IBM ensured that the introduction of the PC would be matched with the availability of plenty of software. This was attained by disclosure of the technical details of the computer and by adopting the MS-DOS operating system. The adoption of this relatively unsophisticated operating system simplified the task, and hence the cost, of developing software. Presently Sun, Next, and IBM are all attempting to induce software developers to create software for their new generations of technical marvels.

Previous work in the literature has examined the implications of single product networks and the associated direct network externality. Farrell and Saloner in a series of papers (1985, 1986a, 1986b) explore the demand side coordination problems associated with network externalities. In order to realize network benefits consumers must correctly anticipate each others adoption choice. If there are alternative competitively supplied networks the possibility of multiple equilibria to the adoption game played by consumers results in coordination problems.

Katz and Shapiro (1985, 1986a, 1986b) are concerned with the behaviour of oligopolistic producers in a single product network. They note that if network externalities exist there are obvious benefits if all consumers are on the same network or a compatible network. The common theme of their work is an examination of the social and private incentives to attain compatibility, i.e. standardization.

Katz and Shapiro (1986a) consider the case where there are two incompatible technologies and investigate whether the market, by adopting only one of the competing technologies, establishes a de facto standard. In their paper, one technology enjoys a cost advantage in the first period and a second technology enjoys a cost advantage in the second period. They show that when both technologies are supplied by a monopolist, the second technology is adopted for many parameter values for which it is socially optimal to adopt the first technology. This adoption pattern occurs because the second technology has a second-mover advantage. A future cost advantage limits the ability of the rival firm to internalize the external benefits accruing to second-period consumers; it reduces the maximum price that the rival can charge in the second period and still have sales.

In this paper we examine networks which consist of both hardware and software products. The

essence of such a network is described by three characteristics. First, we rule out vertical integration. Hardware firms cannot provide their own software. While we do not explore the reason for this restriction, we note that it corresponds to the actual industrial organization of the markets in which we are interested.¹

Second, the production of software typically involves set-up or development costs which are large relative to marginal production costs. Our model shows how these development costs influence the adoption prospects of a technology.

Third, when faced with a choice between two networks consumers typically evaluate each on the basis of the software available. We capture this aspect of preferences by adopting a CES utility function.² We have thus implicitly assumed that both hardware technologies are capable of performing the same tasks or yielding the same services, provided the requisite software is available. Moreover, we have assumed that there are no technological reasons which make the provision of a type of software impossible. Thus it seems reasonable to adopt a specification of preferences which assumes that consumers value all available software products equally. Hence a consumer cares only about the number of software products provided, the price of software, and the price of hardware.

Within this "hardware-software" paradigm, we address the following four issues: Which technology does the market adopt? Does the nature of competition in the software industry affect which technology the market adopts? Is the technology adopted socially optimal? If the market pattern of adoption is inefficient, what kind of contractual arrangements between hardware and software firms internalize the network externality and redress the inefficient adoption pattern?

Using a simple model, we show that when competition in the software industry can be approximated by monopolistic competition, the market outcome is always efficient. The technology which provides the greatest benefit to consumers in the market is also the technology which provides the greatest social surplus. However, when software firms are Bertrand competitors, a hardware technology that has lower development costs is adopted for many parameter values for which it is socially optimal to adopt the other hardware. We establish that there is a systematic bias: the technology which provides the greatest benefit in the market and the technology which provides the greatest social surplus are not the same.

¹Church and Ware (1989) provide an explanation for this distribution of functions across firms.

²The CES utility function was first used to model variety by Dixit and Stiglitz (1977).

The magnitude of the network externality depends on both the number of compatible software products or varieties and the price of compatible software. By enabling a larger number of software firms to enter, lower software development costs increase the magnitude of the network externality. The private value of a larger network under Bertrand competition is both a greater variety of software and a lower price of software. The social value of a larger network, however, is only a greater variety of software. We show that the divergence between the private and social value of a larger network is greatest when software development costs are large or when the assumption of monopolistic competition is least appropriate.

We then entertain the possibility that through a number of contractual arrangements, the low-cost hardware firm might be able to internalize the externality. Successful internalization implies that the market surplus and the social surplus are identical and hence the pattern of adoption efficient. We find that the low-cost hardware firm can redress the inefficient adoption in the market if it can control access by software firms to its technology. The situation we have in mind is if the development of software requires either knowledge or a component which only the hardware firm can supply. Under these circumstances there are two types of agreements which led to an efficient pattern of adoption. The first is if the hardware firm is able to contractually specify the *price of software*. This type of re-sale price maintenance not only increases the profits of the hardware firm, but the surplus of consumers as well: it is unambiguously welfare improving.

Alternatively, we find that the low-cost hardware firm can increase the consumption benefits of its network if it can license access to software firms. An access fee effectively increases the fixed costs of software development which reduces the number of software firms and, *ceteris paribus*, decreases the attractiveness of the technology. However, the proceeds from the licensing fee can be used to subsidize sales of hardware, increasing the surplus associated with the technology. We show that through the optimum license fee, the hardware firm can provide consumers with the full social surplus of its technology, and hence the market pattern of adoption under these circumstances is no longer inefficient.

Two considerations appear to indicate that for successful internalization of the externality, the low-cost hardware firm should subsidize entry into the software industry. These are that the welfare of consumers is increasing in the number of varieties of software and that the market bias is towards the technology with the lower software development costs precisely because it has more software varieties. The optimal license has the opposite effect of a subsidy: it decreases the number of software varieties. This surprising result stems from the fact that relative to the social optimum,

the market outcome is characterized by too much variety.

In Section II, we describe the preferences of a representative consumer. We discuss the production technology in Section III. In Section IV we derive the equilibrium in the software industry under the assumptions of monopolistic and Bertrand competition. Section V is an examination of the market pattern of adoption. We characterize the socially optimal adoption of technology in Section VI. Section VII is a comparison of the market outcome to the social optimum. In Section VIII we extend the model and allow hardware firms the opportunity to either subsidize or restrict the entry of software firms. A conclusion and suggestions for further work appear in Section IX.

II. Consumer Preferences

In this section we specify a partial equilibrium model of the preferences of a representative consumer. We then derive the software demand functions of a consumer and the benefit or utility they receive from joining a network.

We assume that consumers can purchase one of two competing hardware systems, denoted A and B. Since the software is not compatible between the systems, a consumer that purchases a unit of hardware from firm A must purchase software written for that system. The benefit received from the hardware system depends on the number of software products available. The preferences for a representative consumer for software are given by

$$U(x_1, x_2, \dots, x_N) = \left(\sum_{i=1}^N x_i^{1/\beta} \right)^\beta, \quad (1)$$

where x_i is the amount of software good i consumed and

$$\left(\sum_{i=1}^N x_i^{1/\beta} \right)^\beta$$

is a network benefit function.³ We have assumed that hardware only facilitates the consumption of software and therefore it does not enter the utility function explicitly. No consumer purchases more than one unit of hardware and we assume that each system has the same benefit function.

The representative consumer who purchases the hardware for system h , will maximize (1) subject to the following budget constraint:

³To insure concavity, $\beta > 1$.

$$\sum_{i=1}^N \rho_i^h x_i = y - p_h, \quad (2)$$

where ρ_i^h is the price of software variety i on network h , y is total expenditure, and p_h is the price of a unit of hardware supplied by firm h .

The solution to this problem consists of the following system of demand equations:⁴

$$x_i[\rho_i^h, p_h, q_h] = \frac{(y - p_h) q_h^{1/(\beta-1)}}{(\rho_i^h)^{\beta/(\beta-1)}}, \quad \forall i, \quad (3)$$

where

$$q_h(\rho_1^h, \rho_2^h, \dots, \rho_N^h) = \left(\sum_{i=1}^N (\rho_i^h)^{-1/(\beta-1)} \right)^{1-\beta}. \quad (4)$$

The magnitude of the network externality for network h is given by q_h , where q_h is the shadow price of utility. The smaller q_h , the greater the external benefit provided by the network. The indirect utility of a consumer that purchases system h is found by substituting (3) and (4) into (1):

$$V(q_h, p_h) = \frac{(y - p_h)}{q_h}. \quad (5)$$

The consumption benefit of a network depends on the magnitude of the network externality (q_h) and the price of hardware (p_h). The former depends on software prices and the number of software products available for hardware system h . A salient case in the sequel is when the price of each brand of software is the same. If $\rho_i^h = \rho_j^h = \rho^h, \forall i$ and j , then

$$q_h = \frac{\rho^h}{N_h^{(\beta-1)}} \quad (6)$$

and the indirect utility of a consumer is

$$V(q_h, p_h) = \frac{(y - p_h) N_h^{(\beta-1)}}{\rho^h}. \quad (5a)$$

This representation highlights the importance of the price of software (ρ^h), the number of varieties of software (N_h), and the consumer's preference for variety (β). The greater β , the stronger the preference of a consumer for variety. As β rises, the elasticity of substitution, $\sigma = \frac{\beta}{\beta-1}$, falls.

Each consumer will purchase one of the competing hardware systems and some of each software variety available for that system. The system purchased will be the one which provides the greatest benefit, i.e. the one for which (5) is greatest. The number of software products supplied will

⁴This derivation is shown in the Appendix.

depend on the number of consumers that adopt the same hardware. Each consumer must form expectations about the adoption decisions of all other consumers. Consumers thus play an adoption game. We assume that consumers are homogeneous, with each willing to spend y on hardware and software. Hence there will be two Nash equilibria to the adoption game. All consumers will adopt one hardware or all consumers will adopt the other hardware. We assume that consumers play the Pareto superior Nash equilibrium.⁵

III. Production Technology

Typically the creation of software involves large development costs. These fixed costs are orders of magnitude larger than the marginal cost of production. The actual production cost for a unit of software consists only of the cost of duplication and packaging. We denote the fixed cost of developing a software product for hardware h by F_h , $h = A, B$. We assume that the fixed cost of developing software for the two technologies is different, but that this cost is the same for all products developed for a particular hardware. The constant marginal cost of software production, denoted by s , is identical for every product produced for either system.

There are only two hardware technologies, A and B. Each is provided by a monopolist. The constant marginal cost of producing hardware is denoted by c_h , $h = A, B$.

IV. The Software Industry

Figure 1 displays the timing and structure of the model. At the beginning of the period hardware firms set their prices. Consumers then purchase the hardware system that gives them a higher expected surplus. In order to determine the benefit derived from the adoption of hardware system h , consumers must, from (5), have expectations about q_h . Hence they must anticipate the software supply response—the number of software products and software prices—to their adoption decision. Based on the adoption decision of the consumers, software firms develop software. The total variety of software developed depends on which hardware technology is purchased by the cohort. The game is solved by backwards induction. We begin by describing the nature of the competition between software firms. We consider two possibilities. The first is that software firms are monopolistic competitors; the second that they are Bertrand competitors. The equilibrium price and number of software firms will depend on the nature of the price competition.

⁵For a detailed exposition of demand side coordination problems, see Farrell and Saloner.

Regardless of the nature of competition, however, the number of software firms is endogenous and depends on which technology is adopted. The analysis of this section will highlight the importance of the behaviour of the software industry in determining the magnitude of the network externality of a hardware technology and hence which hardware system is adopted in the market. The assumption of free entry into software underscores the critical role played by the development costs of software. Moreover, in this section we assume that software firms have unlimited or free access to the technology required to produce software. In Section VIII we consider the case when the hardware firm can control access to the technology required for the production of software.⁶ Finally, we restrict each software firm to the production of only one variety.

The gross profit of a representative software firm i depends on the hardware system adopted (h), the price it charges (ρ_i^h), the price charged by the hardware firm (p_h), and the price index of software (q_h), which summarizes all software prices:

$$\begin{aligned}\pi_i[\rho_i^h, p_h, q_h] &= m x_i[\rho_i^h, p_h, q_h] (\rho_i^h - s) \\ &= \frac{m(y - p_h) (\rho_i^h - s) q_h^{1/(\beta - 1)}}{(\rho_i^h)^{\beta/(\beta - 1)}},\end{aligned}\quad (7)$$

where we have substituted (3), the demand of each consumer for a variety of software. Software firm i takes the price of hardware, the price of all other software varieties, and the number of software firms as given and chooses ρ_i^h to maximize profits. The distinction between the two types of competition is based on whether or not the software firms internalize the dependency of the price index of software, q_h , on ρ_i^h . If the competition in the software industry is Bertrand, when firms select their profit maximizing price, they will take into account the indirect effect that changes in ρ_i^h have on $x_i[\rho_i^h, p_h, q_h]$ due to the dependence of q_h on ρ_i^h . If the competition in the software industry is monopolistic, firms ignore this dependency and treat q_h as exogenous.⁷

If the competition is Bertrand, the symmetric Nash equilibrium price, ρ_B^h , is,⁸

$$\rho_B^h = \frac{(\beta N_h - 1)s}{(N_h - 1)}.\quad (8)$$

If the competition is monopolistic, the symmetric Nash equilibrium price, ρ_m^h , is,

⁶Farrell and Gallini (1988) consider the case of a single hardware firm that controls access to the hardware technology.

⁷The equilibrium price of software and the equilibrium number of software firms under monopolistic competition correspond to those found in Dixit and Stiglitz (1977) when allowance is made for our exclusion of an outside good.

⁸This derivation is done in the appendix. We restrict the parameter space such that $N > 1$. Equation (11) and (11a) below shows that these restrictions are $m(\beta - 1)(y - c_h)/\beta > F_h$.

$$\rho_B^h = \beta s \quad (8a)$$

Under Bertrand competition the equilibrium price of software depends on the number of firms in the industry, the preference of consumers for variety, and the marginal cost of software production. Equation (8) shows that as the number of software firms increases, the equilibrium price of software under Bertrand competition falls. The equilibrium price of software under monopolistic competition depends only on the marginal cost of production and the preference of consumers for variety. The limit of (8) as N_h goes to infinity is (8a). As the number of software firms in the industry increases under Bertrand competition, the software price approaches the price under monopolistic competition.

In the symmetric Bertrand-Nash equilibrium the magnitude of the network externality is,

$$q_h = \frac{(\beta N_h - 1)s}{N_h^{(\beta-1)}(N_h - 1)}, \quad (9)$$

while in the case of monopolistic competition,

$$q_h = \frac{\beta s}{N_h^{(\beta-1)}}. \quad (9a)$$

If we substitute (8) back into (7), the equilibrium net profit of a representative software firm under Bertrand competition is,

$$\pi_i[p_h, N_h] = \frac{m(\beta - 1)[y - p_h]}{\beta N_h - 1} - F_h. \quad (10)$$

Under monopolistic competition,

$$\pi_i[p_h, N_h] = \frac{m(\beta - 1)[y - p_h]}{\beta N_h} - F_h. \quad (10a)$$

A free-entry equilibrium requires that in equilibrium the number of software firms be such that the profits of an additional entrant be non-positive. Thus the equilibrium number of firms for both types of competition is found by setting (10) and (10a) equal to zero and solving for N_h . The free-entry number of firms in the Bertrand case is

$$N_h^B[p_h] = \frac{m(\beta - 1)[y - p_h]}{\beta F_h} + \frac{1}{\beta} \quad (11)$$

Under conditions of monopolistic competition:

$$N_h^m[p_h] = \frac{m(\beta - 1)[y - p_h]}{\beta F_h} \quad (11a)$$

The equilibrium software price for any number of software firms is higher under Bertrand competition than under monopolistic competition; consequently profits will also be greater. For any price of hardware, the number of software firms required to drive profits to zero will be greater under Bertrand competition than under monopolistic competition.

For either type of competition in the software industry and regardless of which network is adopted, we have the following proposition regarding the existence of a complementary network externality.

Proposition 1. The welfare of a consumer on a network increases, ceteris paribus, as the aggregate number of consumers on that network increases.

Proof. From (11) or (11a) it is immediate that for any price of hardware, the number of software products provided increases as the size of the consumer cohort, m , increases. From (5a), the benefit of joining a network is increasing in the number of software products available for the network, N_h . QED.

For any hardware price, p_h , equations (8) or (8a) and (11) or (11a) determine the Nash equilibrium price of software and the free entry equilibrium number of software firms. Equation (9) or (9a) determines the extent of the network externality. In the next section we will derive the equilibrium pricing strategies for the hardware firms and the implied pattern of adoption by consumers.

V. The Market Pattern of Adoption

At the beginning of the period, the two hardware firms engage in price competition. Only the firm which can offer consumers the greatest surplus or benefit will have positive sales. The lowest possible price a hardware firm will be willing to charge is its marginal cost: a lower price would result in negative profits. In equilibrium the winning firm will offer its hardware at a price, which

in conjunction with the amount of software supplied at that price, gives consumers the same benefit they would receive if the other technology was priced at marginal cost.

From (11) and (11a), the equilibrium number of software firms depends on the price of hardware. The effect of the price of hardware on the number of software firms and the equilibrium price of software is a function of the competition in the software industry. The maximum number of software firms that can be induced to provide software if network h is adopted is, under Bertrand competition,

$$\hat{N}_h^B = \frac{m(\beta - 1)[y - c_h]}{\beta F_h} + \frac{1}{\beta}, \quad (12)$$

and under conditions of monopolistic competition,

$$\hat{N}_h^m = \frac{m(\beta - 1)[y - c_h]}{\beta F_h}. \quad (12a)$$

With some abuse of notation, we denote the magnitude of the network externality obtained from substituting (12) and (12a) into (9) and (9a), respectively, as \hat{q}_h .

Proposition 2. The standard adopted by the market is the technology of firm A if

$$\frac{[y - c_A]}{\hat{q}_A} > \frac{[y - c_B]}{\hat{q}_B}. \quad (13)$$

The technology supplied by firm B will always be adopted if

$$\frac{[y - c_B]}{\hat{q}_B} > \frac{[y - c_A]}{\hat{q}_A}.$$

Proof. From (5), the benefit received from joining network h is,

$$V[q_h, p_h] = \frac{y - p_h}{q_h}.$$

Consumers will adopt the network which generates the greatest benefit. The lowest price that hardware firm h is willing to charge is c_h . This price induces the greatest number of software firms to enter. The maximum surplus associated with network h is

$$V[\hat{q}_h, c_h] = \frac{y - c_h}{\hat{q}_h}.$$

The hardware firm whose network can offer the greatest surplus will always be adopted. Suppose that it is network A. Then the price it will charge is implicitly defined by

$$\frac{[y - p_A]}{q_A} = \frac{[y - c_B]}{\hat{q}_B}$$

or

$$p_A = y(1 - \frac{q_A}{\hat{q}_B}) - \frac{q_A}{\hat{q}_B} c_B$$

A lower price reduces profits. A higher price results in zero profits; the technology will not be adopted. Firm B would respond by charging a slightly lower price, thereby capturing the market and earning positive profits. QED.

Two technological aspects of a network are critical in the determination of its adoption prospects: the marginal cost of hardware production and the cost of software development. The difference between the income of consumers and the marginal cost of hardware production represents a direct surplus. It is the amount of the consumer's endowment which can be used to purchase software after paying the cost of production for the required unit of hardware. This is then multiplied by the inverse of the network externality, q_h . The magnitude of the network externality depends on the number of software firms. The number of software firms is determined in large part by the fixed cost of software development. The trade off between the fixed costs of software development and the production cost of hardware for the two different types of software price-setting behaviour is captured in the following corollary.

Corollary 1. Under conditions of Bertrand competition in the software industry, technology A is adopted in the market if:

$$F_A \left(\frac{m(\beta - 1)[y - c_A] + (1 - \beta)F_A}{\beta F_A} \right) \left(\frac{m(\beta - 1)[y - c_A] + F_A}{\beta F_A} \right)^{\beta - 1} > F_B \left(\frac{m(\beta - 1)[y - c_B] + (1 - \beta)F_B}{\beta F_B} \right) \left(\frac{m(\beta - 1)[y - c_B] + F_B}{\beta F_B} \right)^{\beta - 1} \quad (14)$$

Under conditions of monopolistic competition in the software industry, technology A is adopted in the market if:

$$[y - c_A] > \left(\frac{F_A}{F_B} \right)^{\frac{(\beta - 1)}{\beta}} [y - c_B]. \quad (14a)$$

Proof. To derive (14) and (14a), substitute (12) and (12a) into (9) and (9a), respectively, to find

\hat{q}_h . Substitute \hat{q}_h into (5a) to find the maximum surplus associated with each hardware technology. Substitute this maximum for each technology into (13) and rearrange.

QED

Equations (14) and (14a) delineate the market adoption pattern as a function of the hardware firms' marginal costs, the software development costs, the preferences of consumers for variety, and the size of the consumer cohort. The role of the fixed costs of software development in the Bertrand case can be understood if we let $\beta = 2$. Upon making the substitution (14) reduces to

$$[y - c_A]^2 > \frac{[y - c_B]^2 F_A}{F_B} - \frac{F_A(F_B - F_A)}{m^2}$$

It is apparent that the difference $F_B - F_A$ plays an important role. If it is positive, then firm A's technology is adopted for many cases where $[y - c_B] > [y - c_A]$, i.e. firm B has a marginal cost advantage. The network externality advantage provided by lower software development costs more than overcomes the disadvantage of not being able to offer as low a hardware price. Notice, however, that the degree of the network advantage provided by lower software development costs enters in two ways. The first is the slope coefficient, F_A/F_B , i.e. the ratio of the fixed costs of software development. This term picks up the effect of a differing number of software varieties, holding the price of software constant. The intercept depends on the absolute magnitude of the difference in the fixed development software cost, deflated by the square of the cohort size divided by F_A . The importance of this latter (m^2/F_A) term is easily understood when we consider that in equilibrium, the magnitude of the market network externality depends on both the number of software varieties and the price of software. If m^2/F_A is small then there are relatively few software varieties supplied for either network. If this is true then the network externality advantage of technology A is further enhanced by dint of it having a significantly lower price of software. If there are only a few software firms, then an increase in the number of software firms has a large negative impact on the price of software.

For the case of monopolistic competition in the software industry, (14a) shows that only the ratio of the development costs of software matters. Under monopolistic competition, the price of software does not depend on the number of software firms. It is therefore independent of the fixed costs of software development. It is only the relative number of software varieties available which determines the adoption prospects of a hardware technology.

VI. The Socially Optimal Adoption Pattern

We now derive the socially optimal pattern of adoption. In the optimal pattern of adoption, the technology which provides consumers with the greatest utility is provided. For each hardware technology we assume that the social planner can select the price of hardware, the price of software, and the number of software varieties. The problem of the social planner is to select the price of hardware, the price of software, and the number of software products to maximize the benefit of consumers (5a), subject to the break-even constraint,

$$m(\rho_h^h - s)x_i[\rho_h^h, p_h^s, q_h^s]N_h^s + m(p_h^s - c_h) - N_h^s F_h = 0,$$

where p_h^s , ρ_h^h , and N_h^s are respectively, the socially optimal price of hardware, software, and number of software products. The gross revenues from the sale of software and hardware must cover the fixed costs of software development.

If hardware system h is adopted the utility of the consumer cohort is maximized by providing⁹

$$N_h^s = \frac{m(\beta - 1)[y - c_h]}{\beta F_h} \quad (15)$$

and setting the price of hardware and the price of software such that

$$\frac{y - p_h^s}{\rho_h^s} = \frac{y - c_h}{\beta s} = v_h \quad (16)$$

The number of software products that maximizes the utility of consumers is uniquely determined. However, (16) defines a continuum of optimal hardware prices and the corresponding optimal software price.¹⁰ The maximized surplus of the consumer cohort if hardware h is optimally provided by the social planner is

$$V_h^s = m v_h (N_h^s)^{\beta - 1}. \quad (17)$$

Proposition 3. The adoption of technology A is socially preferred to the adoption of technology B if

$$[y - c_A] > \left(\frac{F_A}{F_B}\right)^{\frac{\beta - 1}{\beta}} [y - c_B] \quad (18)$$

⁹The derivation of these magnitudes is found in the Appendix.

¹⁰The first-order necessary conditions do not have full rank. The necessary conditions corresponding to p_h^s and ρ_h^h are identical. An examination of the utility function (5a) and the budget constraint, after substituting in the demand for software, shows that it is the factor $\frac{y - p_h^s}{\rho_h^s}$ that determines the utility of consumers.

Proof. To derive (18) substitute (15) and (16) into (17) for each hardware technology. QED.

Whether V_A^s or V_B^s is larger depends on the marginal cost of hardware and the ratio of the fixed costs of software development. The difference in hardware marginal costs is adjusted by the ratio of the fixed costs of software development. In the social optimum, the benefit from a larger network or equivalently, lower costs of software development, is an increase in the number of software varieties supplied. If $F_B > F_A$ then $c_A > c_B$ for technology B to provide the same social surplus as technology A. For social indifference, $v_B > v_A$, to compensate for $N_A^s > N_B^s$.

VII. Comparison Between the Market Pattern of Adoption and the Socially Optimal Pattern of Adoption

In this section we compare the pattern of adoption in the market with the socially optimal pattern. The optimality of the market adoption depends on whether or not the competition in the software industry is Bertrand or monopolistic. If the competition is monopolistic, then the maximum market surplus and the social surplus of a technology are identical and there is no adoption bias. However, if the competition in the software industry is Bertrand, then the technology with the lower software development cost is overadopted. The private benefit of the technology with the lower costs of software development exceeds the surplus provided by the other technology in the market, but the social benefit of the second technology is greater. The analysis of the preceding two sections gives rise to Proposition 4.

Proposition 4a. If the competition in the software industry is monopolistic, the market pattern of adoption is optimal.

Proposition 4b. If competition in the software industry is Bertrand, the technology which has the lowest fixed cost of software development is over-adopted.

Proof. Without loss of generality assume that $F_B > F_A$. To establish 4a note that from Corollary 1, the market pattern of adoption under conditions of monopolistic competition is given by (14a) and from Proposition 3, the social pattern, (18), is identical.

For 4b, the social planner is indifferent between the two technologies, if from (18),

$$[y - c_A] = \left(\frac{F_A}{F_B}\right)^{\frac{\beta - 1}{\beta}} [y - c_B]. \quad (19)$$

For technology A to be over-adopted, when the social surplus provided by the two technologies is

the same, the market surplus of A must be greater than the surplus provided by technology B. Thus if we substitute (19) into (14) and rearrange

$$m[y - c_B] \{ (m[y - c_B](\beta - 1) + (F_A)^{1/\beta} (F_B)^{(\beta-1)/\beta})^{\beta-1} - (m[y - c_B](\beta - 1) + F_B)^{\beta-1} \} > \\ F_A^{1/\beta} F_B^{(\beta-1)/\beta} [m[y - c_B](\beta - 1) + (F_A)^{1/\beta} (F_B)^{(\beta-1)/\beta}]^{\beta-1} - F_B [(m[y - c_B](\beta - 1) + F_B)^{\beta-1}]$$

for technology A to be adopted in the market. It is simple to show that this inequality is true since by assumption, $m[y - c_B] > F_B > F_A$. If $m[y - c_B] < F_B$ then $N_B < 1$.

QED.

The market bias towards technology A under conditions of Bertrand competition when $F_B > F_A$ can be intuitively understood by noting that the benefit from a larger network differs between the market and the social optimum. In the social optimum, the benefit of a larger network comes from the provision of more software products. Hence the importance of the ratio of the development costs of a software variety. In the market outcome, both the price of software and the number of software products determine the magnitude of the network externality. In the market, network B needs an even greater marginal cost advantage, than it does in the social optimum, to be an equally attractive alternative to network A. It is easy to show that the price of software is convex in the number of software firms. When there are only a few software firms in the market (m is small or F_h is large), the effect on the price of software of another software firm is substantial. When there are only a few firms in the industry the market benefit of the network externality will be much greater than the social value of the network externality and the technology with the lower costs of software development will be over-adopted. When the number of software firms is small is precisely when we would expect that the software firms would be Bertrand competitors.

The free-entry number of firms in the monopolistically competitive software industry when $p_h = c_h$ is identical to the social optimum number. The hardware firm by lowering its price to marginal cost can induce the socially optimum number of software firms to enter. Moreover, if $\rho_h^s = \beta s$ then from (16) the socially optimal price of hardware, p_h^s , is c_h .

In the Bertrand case, when $p_h = c_h$, the hardware firm induces too much entry: compare (12) and (15). Consumers would be better off with a lower price of software and fewer software firms. Of course, they would like a lower software price and the same number of software firms. The difference between the social surplus and the Bertrand surplus indicates that they would be willing to forgo some variety if it meant a lower price of software. The higher price under conditions of

Bertrand competition arises because the elasticity of demand is more inelastic than under monopolistic competition. The higher price means that more firms can enter in a symmetric free-entry equilibrium. Relative to the low fixed cost of software technology, the high fixed cost technology offers fewer software firms and a higher software price.

VIII. Hardware Subsidization of Software Development

In this section we entertain the possibility that the hardware firm can enter into contracts with the software firms and thus determine the conditions of entry. We consider three different kinds of contractual arrangements between hardware and software firms. The contracts considered here are applicable only to the case when competition in the software industry is Bertrand. If the software industry is monopolistically competitive the hardware firm can provide consumers with the greatest possible surplus simply by lowering its price to marginal cost.

Prima facie it would appear that the low-cost hardware firm, for those parameter values for which its technology is socially optimal but not adopted in the market, would have an incentive to subsidize entry into the software industry. After all in these circumstances the maximum surplus is provided by its technology. The availability of more software varieties might make a network more attractive. Lowering the fixed costs of software development paid by software firms would have this effect. To fund the subsidy, the hardware firm would have to raise its price above marginal cost.

To find if subsidization by either hardware firm raises the benefit provided by its technology, we assume that each hardware firm announces that it is willing to cover α percent of the fixed cost of software development for all software varieties. The optimal subsidy- the one that maximizes the surplus of technology h -is found by maximizing the utility of a consumer,

$$V(q_h, p_h) = \frac{(y - p_h) N_h^{(\beta-1)}}{\rho^h} \quad (5a)$$

subject to the budget constraint,

$$m[p_h - c_h] = \alpha F N_h \quad (20)$$

where the free-entry number of software firms is,

$$N_h[p_h, \alpha] = \frac{m(\beta-1)[y - p_h]}{\beta(1-\alpha) F_h} + \frac{1}{\beta} \quad (21)$$

Proposition 5. Subsidizing entry into the software industry decreases the surplus of a network. However, if the hardware firms are able to control access to the technology required to develop software, they can increase the surplus offered by a network by charging a licensing fee.

Proof. From (20) and (21) we can solve for p_h and N_h as functions of α :

$$p_h(\alpha) = \frac{m\alpha(\beta-1)y + m(1-\alpha)\beta c_h + \alpha(1-\alpha)F_h}{m(\beta-\alpha)}$$

$$N_h(\alpha) = \frac{m(\beta-1)[y-c_h] + (1-\alpha)F_h}{F_h(\beta-\alpha)}$$

Substituting these and the equilibrium software price under Bertrand competition (8), into (5a), we have utility as a function only of α :

$$V(\alpha) = k(1-\alpha) \frac{\{m(\beta-1)[y-c_h] + F_h(1-\alpha)\}^{\beta-1}}{(\beta-\alpha)^\beta}, \quad (22)$$

where k is a positive constant. It is simple to show that

$$\frac{dV(\alpha)}{d\alpha} = kz^{\beta-2}(\beta-\alpha)^{-\beta-1} \{(1-\alpha)\beta z - (\beta-\alpha)(z + (1-\alpha)(\beta-1)F_h)\}$$

where k is a constant and $z = m(\beta-1)(y-c_h) + (1-\alpha)F_h$.

If we impose the sensible restriction that $\alpha \leq 1$, this equals zero when

$$\hat{\alpha}_h = \frac{\beta F_h}{\beta F_h - m(\beta-1)[y-c_h]}$$

which is negative. The restriction on parameters which ensures that $N_h > 1$ imply that the denominator is negative.¹¹ The optimal subsidy is negative! Subsidization of entry into software in the case of Bertrand competition in the software industry does not raise the surplus offered by a technology, but restricting entry by offering a negative subsidy or charging a licensing fee which increases the fixed cost of the software firms does raise the surplus offered by a technology. QED.

In the previous section we showed that the unsubsidized maximum market surplus in the Bertrand

¹¹See footnote 8.

case was less than the social maximum because the market solution at $p_h = c_h$ had too high a software price and too many software products. By subsidizing entry the low-cost software firm is able to lower the price of software towards the optimal price by increasing the number of firms in the software industry. However, the negative consequences for utility from raising the price of hardware always more than offsets the positive effect on utility of reducing the price of software and increasing the variety of software products. Consequently, the low-cost hardware firm would never find it optimal to subsidize entry into software.

However, suppose that there is either some knowledge or component that the hardware firm must provide to software firms for them to be able to develop software. In these circumstances, the hardware firm can increase the surplus of its technology by controlling the terms of entry. Proposition 5 indicates that the hardware firm can increase the surplus of its network if it can charge software firms a licensing fee equal to $\hat{\alpha}_h F_h$. The revenues from the licensing fee permit the hardware firm to price below marginal cost. The reduction in the benefit to consumers from a higher software price and fewer varieties of software is more than offset by the lower price of hardware.

Alternatively each hardware firm could increase the benefit of its hardware by using re-sale price maintenance. In exchange for access to its technology, the hardware firm could stipulate that the software firms charge the monopolistically competitive price, βs . For both of these alternatives we have the following proposition.

Proposition 6. If the hardware firms can stipulate the terms of entry to the software firms and they either charge the optimal license $\hat{\alpha}_h F_h$ or they allow access on the condition that the software firms set their price at βs , then the market pattern of adoption is efficient.

Proof. If the price of software firms is βs , then from our previous analysis of a monopolistically competitive software industry, we know that when a hardware firm sets $p_h = c_h$, the surplus of the network will equal the socially optimal surplus and the market pattern of adoption will not be inefficient. It is straightforward to show that the market surplus for each hardware equals the social surplus when the license fee is $\hat{\alpha}_h F_h$. QED.

Under these circumstances both of these contractual arrangements have attractive welfare properties. They increase not only the profits of hardware firms, but also the surplus of consumers! They are unambiguously welfare improving.

IX. Conclusion

In this paper, we considered technologies with network externalities generated by complementary products. When the complementary products are incompatible between hardware technologies, market standardization depends not only on the difference in hardware costs, but also on software development costs. The role of the software development costs is in determining the number of software firms which can enter. The software development costs determine both the relative number of complementary products and the nature of the competition in the software industry. If the development costs of software are large vis-a-vis the consumer cohort, then the number of software products will be small and the appropriate competition in the software industry will be Bertrand. Under these circumstances the adoption pattern of the market is inefficient. The technology with the lower costs of software development is over-adopted by the market. Our analysis indicates that the socially efficient hardware firm is not able to redress this inefficiency by subsidizing entry into software. The inefficiency can be redressed via either re-sale price maintenance in the software market or by the imposition of a license or entry fee on the software firms. Both of these alternatives assume that the hardware firm is able to deny access to the software market to dissenting software firms.

In a previous version of this paper, we extended the model to two periods. This allowed us to address the feasibility and the desirability of intertemporal standardization. The results of the extension showed that the second-mover advantage identified by Katz and Shapiro (1986a) continues to hold in a model where the network externality is generated via complementary products. With complementary products, the second-mover advantage is created by both lower second period marginal costs and lower software development costs. That work also showed that the main effect of differing costs of software development was to give the low cost technology, similar to the one-period model presented in this paper, a stronger network effect.

We are presently working on two related papers. Both of these use the indirectly utility function introduced here to model consumer preferences over complementary products. In this paper, our results depend on free-entry into the software industry. This effectively negates any strategic behaviour on the part of the software firms in determining which technology is adopted. In another paper (Church and Gandal; 1990) we investigate the strategic role that software firms might play in determining which network emerges as the market standard. The second extension, which is work in progress, uses our framework to investigate the incentives for exclusive dealing and market foreclosure on the part of hardware firms.

APPENDIX

A. Demand For Software

We now derive the demand for software; by duality we can minimize expenditure subject to a fixed utility level. We minimize

$$E = \sum_{i=1}^N \rho_i^h x_i$$

subject to a fixed utility level, z . The Lagrangian is:

$$L = \sum_{i=1}^N \rho_i^h x_i + q_h \left(z - \left(\sum_{i=1}^N x_i^{1/\beta} \right)^\beta \right)$$

$$\frac{\partial L}{\partial x_i} = \rho_i^h - q_h \left(\sum_{i=1}^N x_i^{1/\beta} \right)^{\beta-1} x_i^{(1-\beta)/\beta}, \forall i$$

$$\frac{\partial L}{\partial q_h} = z - \left(\sum_{i=1}^N x_i^{1/\beta} \right)^\beta$$

Setting $\frac{\partial L}{\partial x_i} = 0$ implies that:

$$x_i^{1/\beta} = \left(\frac{q_h}{\rho_i^h} \right)^{1/(\beta-1)} z^{1/\beta}, \forall i.$$

Setting $\frac{\partial L}{\partial q_h} = 0$, and substituting the above expression for x_i yields:

$$q_h = \left(\sum_{i=1}^N (\rho_i^h)^{-1/(\beta-1)} \right)^{1-\beta},$$

the shadow price of utility.

Using the expression for x_i ,

$$\sum_{i=1}^N \rho_i^h x_i = q_h z = e(\rho, z), \text{ i.e. the expenditure function.}$$

Using the identity that $I = e(\rho, z(\rho, I))$,

$$z = \frac{I}{q_h}, \text{ where}$$

I is the disposable income spent on software, $y - p_h$. Of course z is the indirect utility function.

Using Roy's identity,

$$x_i = - \frac{\frac{\partial z}{\partial \rho_i^h}}{\frac{\partial z}{\partial I}} = (y - p_h) q_h^{1/(\beta-1)} (\rho_i^h)^{\beta/(\beta-1)}, \forall i.$$

B. Bertrand Profit Maximizing Software Price

We now derive the profit maximizing software price for the case of Bertrand competition.

The gross profit of a representative software firm, if technology h is adopted, is:

$$\begin{aligned}\pi_i[\rho_i^h, p_h, N_h] &= \max_i[\rho_i^h, p_h, q_h] (\rho_i^h - s) \\ &= m(\rho_i^h - s) (y - p_h) q_h^{1/(\beta-1)} (\rho_i^h)^{\beta/(\beta-1)}\end{aligned}$$

From which the first order condition

$$\rho_i^h + (\rho_i^h - s) \frac{\beta}{1 - \beta} + \frac{\rho_i^h - s}{(\beta - 1)N_h} = 0,$$

since

$$\frac{\partial q_h}{\partial \rho_i^h} = \left(\frac{q_h}{\rho_i^h} \right)^{\beta/(\beta-1)}$$

and in a symmetric equilibrium $\rho_i^h = \rho^h, \forall i$ and thus

$$q_h = \rho^h N_h^{-(\beta-1)}.$$

Substituting these into the first order condition yields

$$\rho^h = \frac{(\beta N_h - 1)s}{(N_h - 1)}.$$

C. Social Optimum

We now derive the socially optimal allocation when the technology of firm h is adopted.

$$L = m(y - p_h^s)(\rho_h^s)^{-1}(N_h^s)^{\beta-1} + \lambda(m[y - p_h^s](\rho_h^s - s)(\rho_h^s)^{-1} + m(p_h^s - c_h) - N_h^s F_h)$$

The first order conditions are:

$$\text{From } \frac{\partial L}{\partial p_h^s}, \quad \frac{-(N_h^s)^{\beta-1}}{\rho_h^s} - \lambda \frac{(\rho_h^s - s)}{\rho_h^s} + \lambda = 0$$

$$\text{From } \frac{\partial L}{\partial \rho_h^s}, \quad \frac{-(N_h^s)^{\beta-1}}{(\rho_h^s)^2} - \lambda \frac{(\rho_h^s - s)}{(\rho_h^s)^2} + \frac{\lambda}{\rho_h^s} = 0$$

$$\text{From } \frac{\partial L}{\partial N_h^s}, \quad \frac{m(\beta-1)(y - p_h^s)(N_h^s)^{\beta-2}}{(\rho_h^s)} - \lambda F_h = 0$$

$$\text{From } \frac{\partial L}{\partial \lambda}, \quad [y - p_h^s](\rho_h^s - s)(\rho_h^s)^{-1} + (p_h^s - c_h) - \frac{N_h^s F_h}{m} = 0$$

Setting the first order equations equal to zero and solving yields

$$\frac{(y - p_h^s)}{\rho_h^s} = \frac{(y - c_h)}{\beta_s}$$

and

$$N_h^s = \frac{m(\beta - 1)(y - c_h)}{\beta F_h}.$$

Bibliography

- Church, Jeffrey and Roger Ware. 1989.
 "Adverse Selection, Market Structure and Innovation."
 Mimeo: University of Calgary.
- Church, Jeffrey and Neil Gandal. 1990.
 "Network Effects and Software Provision."
 Boston University School of Management Working Paper.
- Dixit, Avinash and Joseph Stiglitz. 1977.
 "Monopolistic Competition and Optimum Product Diversity."
 American Economic Review
 67: 297-308.
- Farrell, Joseph and Nancy Gallini. 1988.
 "Second Sourcing as a Commitment: Monopoly Incentives to Attract Competition."
 Quarterly Journal of Economics
 103: 673-694.
- Farrell, Joseph and Garth Saloner. 1985.
 "Standardization, Compatibility, and Innovation."
 Rand Journal of Economics
 16: 70-83.
- . 1986a.
 "Standardization and Variety."
 Economic Letters
 20: 71-74.
- . 1986b.
 "Installed Base and Compatibility: Innovation, Production Preannouncements, and Predation."
 American Economic Review
 76: 940-955.
- . 1988.
 "Coordination Through Committees and Market."
 Rand Journal of Economics
 19: 235-252.
- Gilder, G. 1984.
 The Spirit of Enterprise.

New York: Simon and Schuster.

Katz, Michael and Carl Shapiro. 1985.

"Network Externalities, Competition, and Compatibility."

American Economic Review

75: 424-440.

---. 1986a.

"Technology Adoption in the Presence of Network Externalities."

Journal of Political Economy

94: 822-841.

---. 1986b.

"Product Compatibility Choice in a Market with Technological Progress."

Oxford Economic Papers

38: 146-165.

Matutes, Carmen and Pierre Regibeau. 1988.

"Mix and Match: Product Compatibility without Network Externalities."

Rand Journal of Economics

19: 221-234.

Spence, Michael. 1976.

"Production Selection, Fixed Costs, and Monopolistic Competition."

Review of Economic Studies

43: 217-235.

Tirole, Jean. 1988.

The Theory of Industrial Organization.

Cambridge: MIT Press.

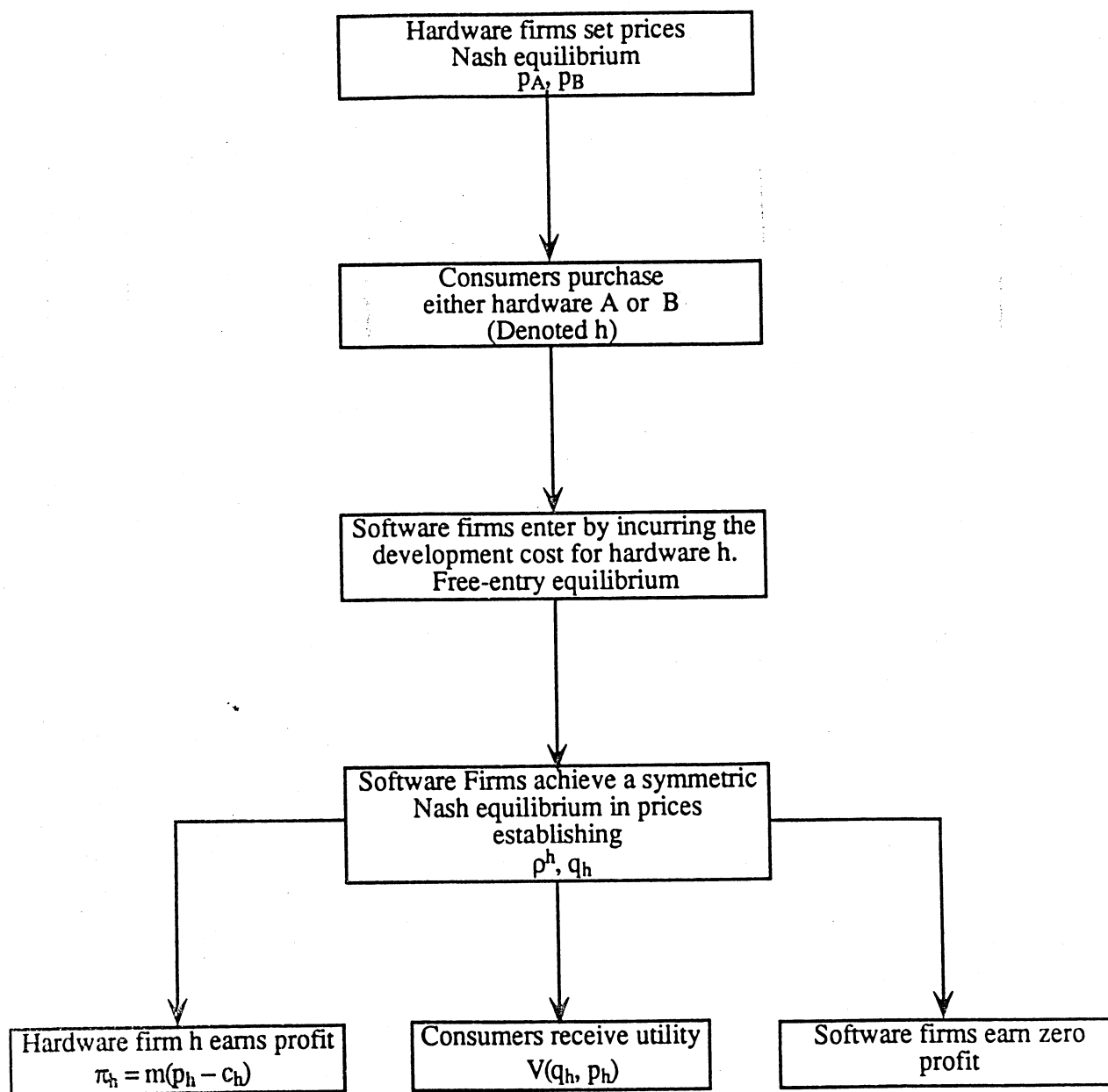


Figure 1
Timing of the Game

