Advanced Microeconomics

Prof. Dr. Carsten Helm

Lecture 18: Contract, law, and enforcement in static settings

Essential reading:

- Watson (2013): Strategy – an introduction to game theory, chapter 13

The role of contracts

- Contracting institutions such as the legal system govern a wide variety of relationships in our society
  - – contracts for firms and workers that specify wages and other aspects of the employment relationship
  - – contracts of homeowners with building contractors
  - – contracts of firms with suppliers and customers
  - – contracts of nations about trade (WTO) and security (NATO) issues
- Contracting may help to
  - – alleviate strategic uncertainty
  - – avoid inefficient coordination
    - esp. when there are multiple equilibria
  - – align incentives
    - ie. mitigate conflicts between joint and individual decisions

The role of contracts

Definition: A **contract** is an agreement about behavior that is intended to be enforced.

- A contractual relationship consists of two phases:
  - Contracting phase, in which players set the terms of their contracts
    - Focus of lecture on Principal-agent model
  - Implementation phase, in which contract is carried out and enforced
    - Focus in this lecture: enforcement of contract
- Methods of contract enforcement: a contract is
  - **self-enforced,** if the players have the individual incentives to abide by the terms of the contract
  - **externally enforced,** if there is an external player, such as a judge or arbitrator, that enforces behavior in accordance with contract

An example

- Remodeling of a house requires effort of an architect and a building constructor
- Both decide simultaneously whether investing (costly) effort in the project (I) or not investing (N)
  - Investing by the architect facilitates the work of the constructor
  - Investing by the constructor improves quality of remodeling
    - We assume that architect is also the owner of the home; hence he benefits from quality of the remodeling
    - Game below is called the *underlying game*

Constructor

		I	N
Architect/Owner	I	z_1, z_2	y_1, x_2
	N	x_1, y_2	0,0

Example: self-enforcing

		Constructor	
		I	N
Architect/Owner	I	z_1, z_2	y_1, x_2
	N	x_1, y_2	0,0

- Assumptions:
  - $z_1 + z_2 > x_1 + y_2, y_1 + x_2, 0$
  - Parties can make monetary transfers
    - Hence (I, I) is the only efficient outcome of the game
- Question: can the parties enforce a contract specifying that (I, I) will be played?
- If $z_1 < x_1$ and/or $z_2 < x_2$, then an agreement to play (I, I) is not *self-enforcing*
  - i.e., it is not a Nash equilibrium

Example: external enforcement

- Thus (I, I) needs to be externally enforced
- Suppose a court imposes transfers α, β, γ if outcome is not (I, I)
  - This is called the *induced game*
- We distinguish two contractual settings
  - a) Court allows people to write *complete contracts* as they see fit and enforces them verbatim
  - b) Court puts constraints on the set of feasible contracts and parties write *incomplete contracts*

		Constructor	
		I	N
Architect/Owner	I	z_1, z_2	y_1, x_2
	N	x_1, y_2	0,0

		Constructor	
		I	N
Architect/Owner	I		
	N		

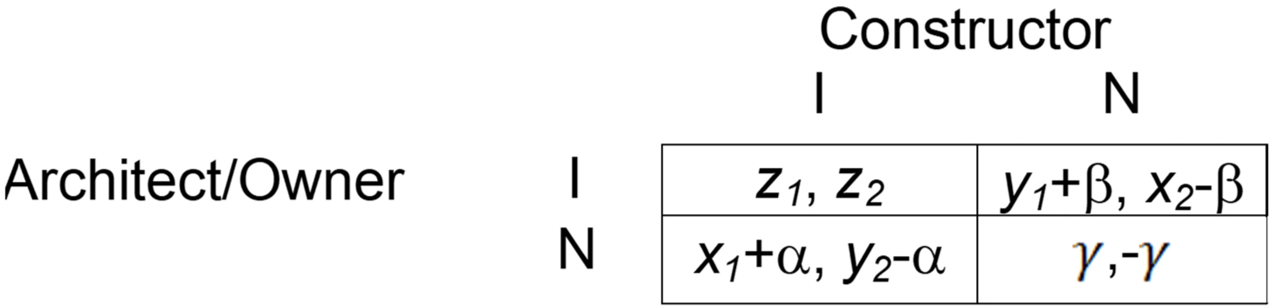
Example: external enforcement

- Thus (I, I) needs to be externally enforced
- Suppose a court imposes transfers α, β, γ if outcome is not (I, I)
  - $\alpha, \beta, \gamma \in \mathbb{R}$, i.e., transfers can be positive or negative
  - This is called the *induced game*
- We distinguish two contractual settings
  1. a) Court allows people to write *complete contracts* as they see fit and enforces them verbatim
  2. b) Court puts constraints on the set of feasible contracts and parties write *incomplete contracts*

		Constructor	
		I	N
Architect/Owner	I	z_1, z_2	$y_1+\beta$, $x_2-\beta$
	N	$x_1+\alpha$, $y_2-\alpha$	γ,$-\gamma$

Complete contracting in discretionary environments

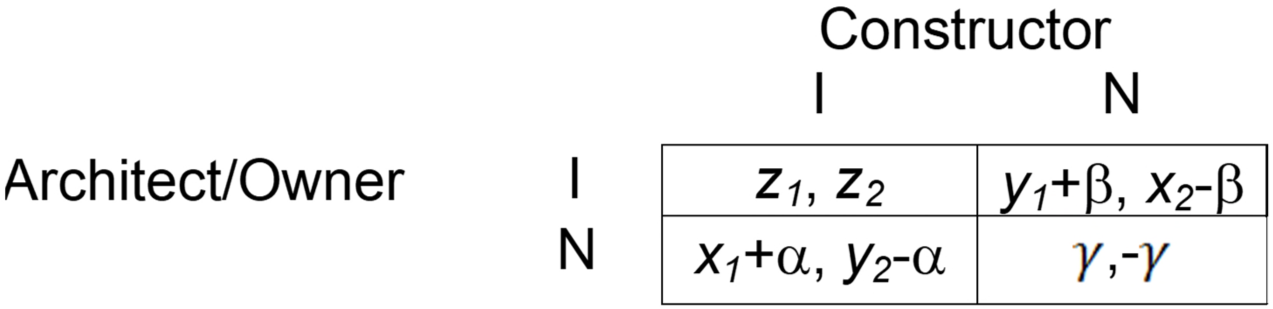
- Suppose the players can write a **complete contract** specifying α, β, γ
- The contract induces (I, I) if
  -
  - and the court enforces the contract,
  - which requires that the contract is **fully verifiable**
    - ie. the court must be able to verify whether the players have chosen I or N



		Constructor	
		I	N
Architect/Owner	I	z_1, z_2	$y_1+\beta, x_2-\beta$
	N	$x_1+\alpha, y_2-\alpha$	$\gamma,-\gamma$

Complete contracting in discretionary environments

- Suppose the players can write a **complete contract** specifying α, β, γ
- The contract induces (I, I) if
  - $z_1 \geq x_1 + \alpha$ and $z_2 \geq x_2 - \beta$,
  - and the court enforces the contract,
  - which requires that the contract is **fully verifiable**
    - ie. the court must be able to verify whether the players have chosen I or N



		Constructor	
		I	N
Architect/Owner	I	z_1, z_2	$y_1+\beta, x_2-\beta$
	N	$x_1+\alpha, y_2-\alpha$	$\gamma,-\gamma$

Contracting in discretionary environments

- But: full verifiability is more often the exception than the rule
  - E.g.: court may be able to judge quality of work, but not whether defects have been caused by failure of architect or constructor
  - In this example of **limited verifiability**, the court cannot distinguish between (I,N), (N,I) and (N,N), which yields game below
  - (I,I) a Nash equilibrium if and
  - Summation yields *necessary* condition
    - which may easily be violated (e.g. with payoffs as in prisoners‘ dilemma)

Constructor

		I	N
Architect/Owner	I	z_1, z_2	y_1, x_2
	N	x_1, y_2	0,0

Constructor

		I	N
Architect/Owner	I		
	N		

Contracting in discretionary environments

- But: full verifiability is more often the exception than the rule
  - E.g.: court may be able to judge quality of work, but not whether defects have been caused by failure of architect or constructor
  - In this example of **limited verifiability**, the court cannot distinguish between (I, N), (N, I) and (N, N), which yields game below
  - (I, I) a Nash equilibrium if $z_1 \geq x_1 + \alpha$ and $z_2 \geq x_2 - \alpha$
  - Summation yields *necessary* condition $z_1 + z_2 \geq x_1 + x_2$
    - which may easily be violated (e.g. with payoffs as in prisoners‘ dilemma)

		Constructor	
		I	N
Architect/Owner	I	z_1, z_2	$y_1+\alpha, x_2-\alpha$
	N	$x_1+\alpha, y_2-\alpha$	$\alpha,-\alpha$

Contracting with court-imposed breach remedies

- Often players write **incomplete contracts**
  - It may be expensive to list all contingencies
  - Players may count on court to “complete” a contract during litigation
- Often courts impose transfers on the basis of certain legal principles, rather than on the basis of the contract document
- Legal principle of **expectation damage**:
  - court imposed transfer of money from defendant to plaintiff so as to give the plaintiff the payoff he would have received had the contract been fulfilled
    - the players’ expectations are z_1, z_2

		Constructor	
		I	N
Architect/Owner	I	z_1, z_2	y_1, x_2
	N	x_1, y_2	0,0

Expectation damages

- **expectation damage**: court imposed transfer of money from defendant to plaintiff so as to give the plaintiff the payoff he would have received had the contract been fulfilled, i.e. z_1, z_2
- If architect breaches, it must pay constructor
- If constructor breaches, it must pay architect
- This yields induced game below

		Constructor	
		I	N
Architect/Owner	I	z_1, z_2	y_1, x_2
	N	x_1, y_2	0,0

		Constructor	
		I	N
Architect/Owner	I		
	N		

Expectation damages

- If architect breaches, it must pay constructor $z_2 - y_2$
- If constructor breaches, it must pay architect $z_1 - y_1$
- This yields induced game below
  - (I, I) a Nash equilibrium if and
  - Rearranging yields conditions for (I, I) to be efficient:

    and
- Result: under expectation damage (I, I) is enforceable if and only if (I, I) is efficient

Constructor

		I	N
Architect/Owner	I	z_1, z_2	y_1, x_2
	N	x_1, y_2	0,0

Constructor

		I	N
Architect/Owner	I	z_1, z_2	$z_1, x_2 - z_1 + y_1$
	N	$x_1 - z_2 + y_2, z_2$	0,0

Expectation damages

- If architect breaches, it must pay constructor $z_2 - y_2$
- If constructor breaches, it must pay architect $z_1 - y_1$
- This yields induced game below
  - (I, I) a Nash equilibrium if $z_1 \geq x_1 - z_2 + y_2$ and $z_2 \geq x_2 - z_1 + y_1$
  - Rearranging yields conditions for (I, I) to be efficient:

$$z_1 + z_2 \geq x_1 + y_2 \text{ and } z_2 + z_1 \geq x_2 + y_1$$

- Result: under expectation damage (I, I) is enforceable if and only if (I, I) is efficient

Constructor

		I	N
Architect/Owner	I	z_1, z_2	y_1, x_2
	N	x_1, y_2	0,0

Constructor

		I	N
Architect/Owner	I	z_1, z_2	$z_1, x_2 – z_1 + y_1$
	N	$x_1 – z_2 + y_2, z_2$	0,0

Reliance damages

- Hence expectation damages alleviate tension between individual and joint incentives
- But: the court needs a great deal of information
  - Who breached the contract
  - Exact payoff structure
- Legal principle of **reliance damages**: court imposes transfer that returns the plaintiff to the state in which she would have been but for the contract
  - Suppose we take the Nash equilibrium payoff $(0,0)$ as the “no contract” value
    - ie. for original game we assume $y_i \leq 0,\ i = 1{,}2$

		Constructor	
		I	N
Architect/Owner	I	z_1, z_2	y_1, x_2
	N	x_1, y_2	0,0

Reliance damages

- **reliance damages**: court imposes transfer that returns the plaintiff to the state in which she would have been but for the contract, here: $(0,0)$
  - yields induced game below
  - (I, I) is a Nash equilibrium in the induced game if and only if
  - In words: damages from not-investing, $y_i < 0$, must be sufficiently large
  - Advantage: easier to implement because court does not need to know payoff for (I, I)

		Constructor	
		I	N
Architect/Owner	I	z_1, z_2	y_1, x_2
	N	x_1, y_2	0,0

		Constructor	
		I	N
Architect/Owner	I		
	N		

Reliance damages

- **reliance damages**: court imposes transfer that returns the plaintiff to the state in which she would have been but for the contract, here: $(0{,}0)$
  - yields induced game below
  - (I, I) is a Nash equilibrium in the induced game if and only if

$$z_1 \geq x_1 + y_2 \quad \text{and} \quad z_2 \geq x_2 + y_1$$

  - In words: damages from not-investing, $y_i < 0$, must be sufficiently large
  - Advantage: easier to implement because court does not need to know payoff for (I, I)

		Constructor	
		I	N
Architect/Owner	I	z_1, z_2	y_1, x_2
	N	x_1, y_2	0,0

		Constructor	
		I	N
Architect/Owner	I	z_1, z_2	$0, x_2+y_1$
	N	$x_1+y_2, 0$	0,0