March 23rd

→ Extensions

2.10 : Note that the book calls it more general "for the Agent", but in 2.10.1 it is more general for $\underline{P}$.

2.11 : Contracting before type revealed to A

2.12 : next time...

not on reading list:

2.13 : A few words regarding concavity or not ⟺ averaging or not cf 2.11

2.14 : A few words on # states of the world

2.15 : You might want me to elaborate on seminar problem #2 ?

2.10

2.10.1 Covered last time

2.10.2 Extension: type-dependent utility S_i $(i=0,1)$ for P.

→ problem as before, but with S_0, S_1 distinguished.

→ new feature: $S_0 \neq S_1$ *may* lead to $q_0 ? q_1$ (under S-M) being *binding*.

→ this *may* happen if $S_1' > S_0'$ in optimum

→ interpretation: type 0 is more quantity-efficient, but you prefer type 1's goods for quality.

(→ good explanation why both types exist)

Recall when $S_0 = S_1 = S$:

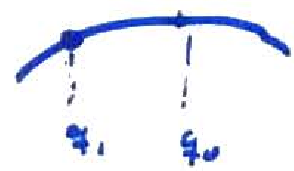
→ S-M: $C'_1 \geq C'_0$, which will imply $q_0 \geq q_1$

→ If $S'_0 > S'_1$, this effect is enhanced. Ex: put $C_1 = C_0 = C$

At q_0: $S'_0 = C'$

At q_1: $S'_1 = C' + 0$

At q_1, S'_0 is $> C'$ and by concavity: $S'_0 - C' > 0$ ⇒



→ But if $S'_1 > S'_0$, it will have the opposite effect: lowering $q_0 - q_1$.

→ If this effect is strong enough, we must pool $q_0 = q_1 = q$, because under S-M, $q_0 < q_1$ is not incentive feasible.

(Note: If $C'_1 \leq C'_0$ – with opposite constant sign – then the effects are reversed.)

The maths: Assume S-M applies.

Form the Lagrangian

$$\nu\big(S_0(q_0) - C_0(q_0)\big) + (1-\nu)\big(S_1(q_1) - C_1(q_1)\big) - \nu\Delta C$$

$$+ \underbrace{\lambda(q_0 - q_1)}_{\text{constr. } q_0 - q_1 \geq 0} + \mu q_1$$

$q_1 \geq 0$. (Recall compl. slackness)

1^{st} o.c's:

At q_0: $\quad S_0' - C_0' = -\dfrac{\lambda}{\nu}$

At q_1: $\quad S_1' - C_1' = \dfrac{\nu}{1-\nu}\Delta C' + \lambda - \mu$

Cases: • $\lambda = \mu = 0$ will be chosen if incentive feasible

• $\mu > 0 \Rightarrow q_1 = 0$, just as before

• $\lambda > 0 = \mu$: $\quad q_0 = q_1$. Insert:

$$\underbrace{\nu S_0 + (1-\nu) S_1}_{=:\ \tilde{S}} - C_1$$

Adapt when $\tilde{S}' = C_1'$

2.10.3 More goods.

q_0 vector in $\mathbb{R}^n$
q_1 — " —

Differs from book

Write D_k for partial derivative wrt the k^{th} component $q^{(k)}$.

Turns out, by inspecting calculations: need to consider the "$q_0 \geq q_1$" which in the single-good case was implied by S-M (which now is a comp.-wise property)

Compatibility leads to $\Delta C(q_0) \geq \Delta C(q_1)$.

Denote ΔC by Φ. Need $\Phi(q_0) \geq \Phi(q_1)$

Both q_i and $\nabla \Phi$ vectors. No "$\geq$".

But previous solutions suggest:

$$\nabla(S_0 - C_0) = 0 \text{ at } q_0$$

$$\nabla(S_1 - C_1) = \frac{\nu}{1-\nu} \nabla \Phi \text{ at } q_1.$$

This solves of inc. feas.

Suff. cond: $\begin{cases} D_k \Phi \geq 0, \text{ all } k \\ S_0 = S_1 = S \end{cases}$

2.11 Contracting before type known to A (and P).

Book: increasing level of generality.

Here: general case (others below)

Setup: vNM utility functions
u for A, v for P. (as book).

- Notation: In order not to confuse u vs U, v vs ν, write
  - p for ν
  - $y_i = t_i - C_i(q_i)$
    to be inserted for t_i
  - $x_i = S_i(q_i) - C_i(q_i) - y_i$
    not to be inserted until after differentiating.

- Ex ante participation constraint
  $0 \leq E[u(\cdot)] = p\,u(y_0) + (1-p)\,u(y_1)$,
  multiplier $= \mu$

- Ex post compatibility:
  $y_0 \geq y_1 + \Phi$, multiplier $= \lambda$.

#6 & #7 corrected

Lagrangian for the non-pooling case (if e.g. S-M, and this approach yields $q_0 < q_1$, then solve separately the case $q_0 = q_1 = q$):

$$
\begin{aligned}
& p\, v(S_0(q_0) - C_0(q_0) - y_0) \\
+ & (1-p)\, v(S_1(q_1) - C_1(q_1) - y_1) \\
& + \mu \cdot (p\, u(y_0) + (1-p)\, u(y_1)) \\
& + \lambda \cdot (y_0 - y_1 - \Phi(q_1))
\end{aligned}
$$

1^{st} o.c.

y_0: $\quad p\, v'(x_0) = \mu p\, u'(y_0) + \lambda$

y_1: $\quad (1-p)\, v'(x_1) = \mu (1-p)\, u'(y_1) - \lambda$

add to get $\mu = Ev' / Eu'$

Cramér's rule: $\lambda = \dfrac{p(1-p)}{Eu'} \begin{vmatrix} v'(x_0) & u'(y_0) \\ v'(x_1) & u'(y_1) \end{vmatrix}$

1^{st} o.c.:

at q_0: $\cancel{p\, v'(x_0)} \cdot (S_0' - C_0') = 0$

at q_1: $\underbrace{(1-p)\, v'(x_1)}_{>0} \cdot (S_1' - C_1') = \lambda\, \Phi'$

≥ 0, so downward shift in q_1 (except case $\lambda = 0$ which implements full information q_1)

#8 corrected

At q_1:

$$S_1' - C_1' = \overbrace{\frac{p}{1-p}\Phi'}^{\text{"usual" expr.}} \cdot \frac{1-p}{v'(x_1)\,Eu'} \begin{vmatrix} v'(x_0) & u'(y_0) \\ v'(x_1) & u'(y_1) \end{vmatrix}$$

Cases – with none of the players risk-seeking :

* Case P risk neutral.
  Determinant $= \left(u(y_1) - u(y_0)\right) v'$
  and at q_1:
  $$S_1' - C_1' = \underbrace{\frac{p}{1-p}\Phi'}\cdot \underbrace{(1-p)\frac{u'(y_1)-u'(y_0)}{Eu'}}_{\text{turns out: } <1}$$

* Case P risk averse, A risk neutral
  Determinant $= \left(v'(x_0) - v'(x_1)\right) u'$
  $\lambda \geq 0 \Rightarrow \text{det.} \geq 0 \Rightarrow x_0 \leq x_1$
  If $\underline{S_1 = S_0}$ this turns out $\Leftrightarrow$ to
  $$\cancel{S(q_0) - C_0(q_0) \leq S(q_1) C_0(q_1)}$$
  $$S(q_0) - C_0(q_0) \leq S(q_1) - C_0(q_1)$$
  but q_0 maximizes $S - C_0$. So $x_0 = x_1$, and $\lambda = 0$. We get $\underline{S' - C_1 = 0}$ at q_1.

* Other cases do not simplify

Brief on signals. By example.

P says: "I want to check your financial statement".

(– assumed <u>verifiable</u> (can be weakened!))

Why would P want this information?

Because it (possibly) <u>correlates with type.</u>

E.g.: assume each agent has either surplus K or 0 (break even) and that the efficient agent is more likely to have K than the inefficient.

→ 4 states of the world:

Signal \ Type	eff.	ineff
0	·	·
K	·	·

→ 2 compatibility constraints & 2 participation constraints — enough for 4 contracts (with =).

⇝ ~~to~~ precisely unless signal is uninformative.