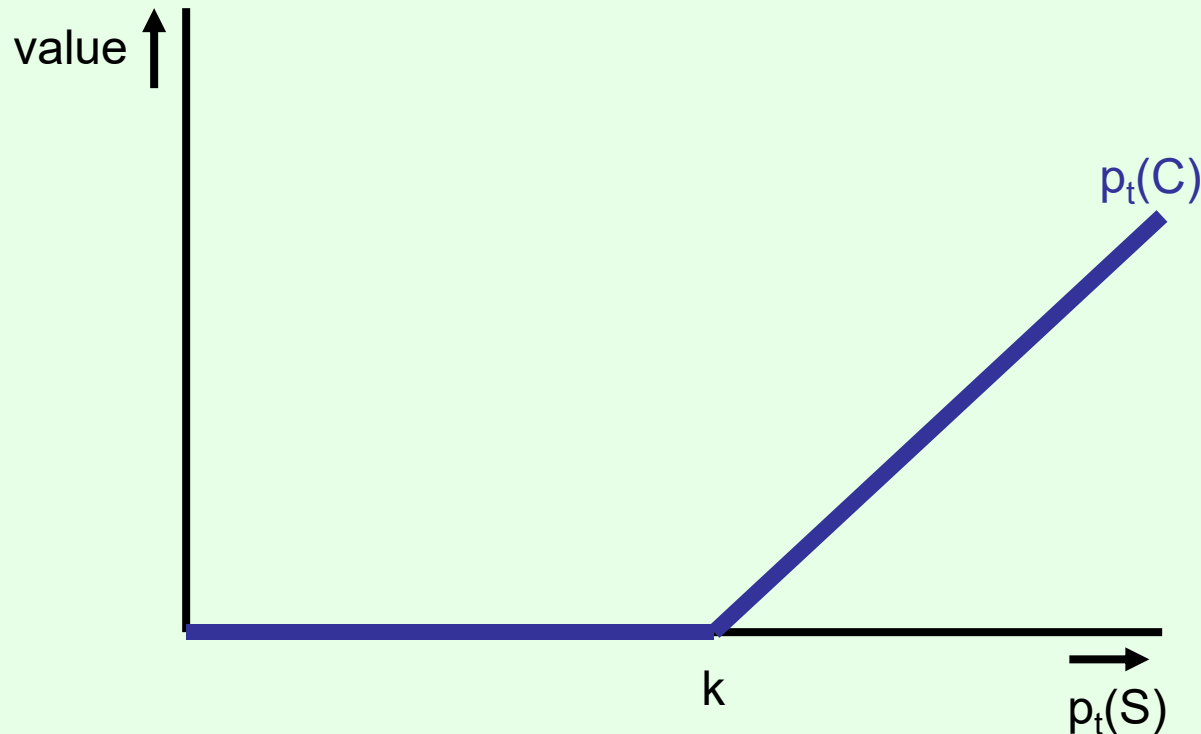


Securities & Expressive Securities Markets

conitzer@cs.duke.edu

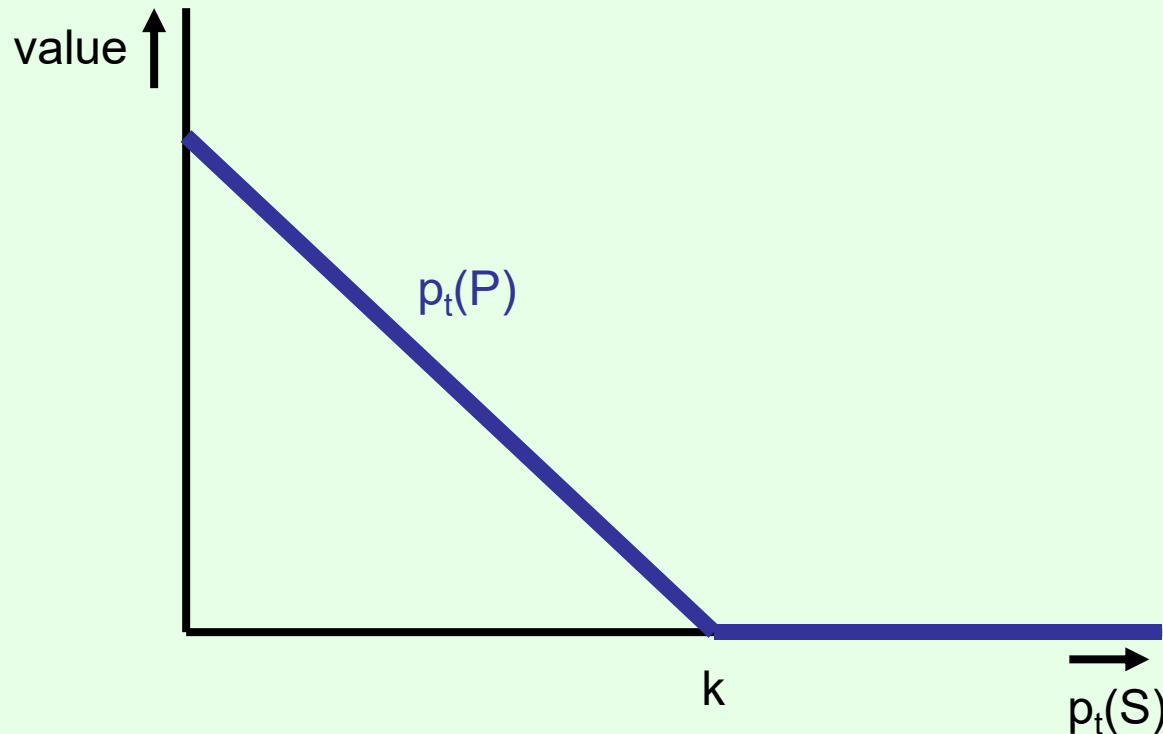
Call options

- A (European) call option $C(S, k, t)$ gives you the right to buy stock S at (strike) price k on (expiry) date t
 - American call option can be exercised early
 - European one easier to analyze
- How much is a call option worth at time t (as a function of the price of the stock)?



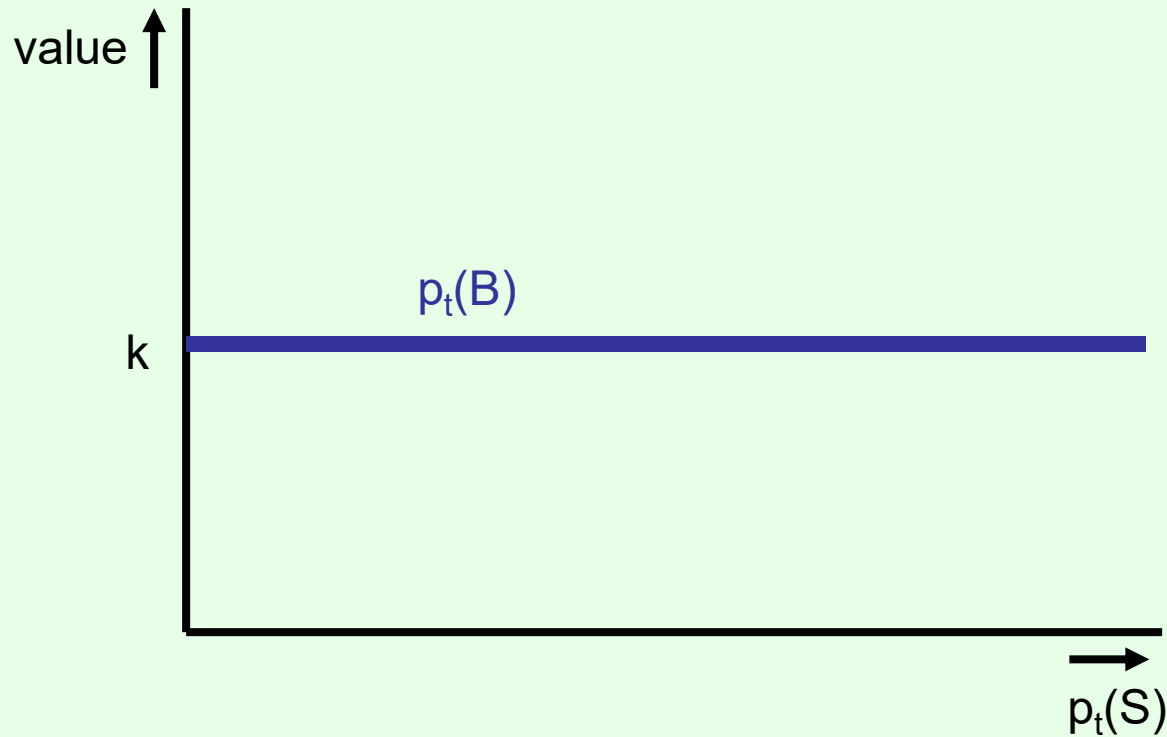
Put options

- A (European) put option $P(S, k, t)$ gives you the right to sell stock S at (strike) price k on (expiry) date t
- How much is a put option worth at time t (as a function of the price of the stock)?

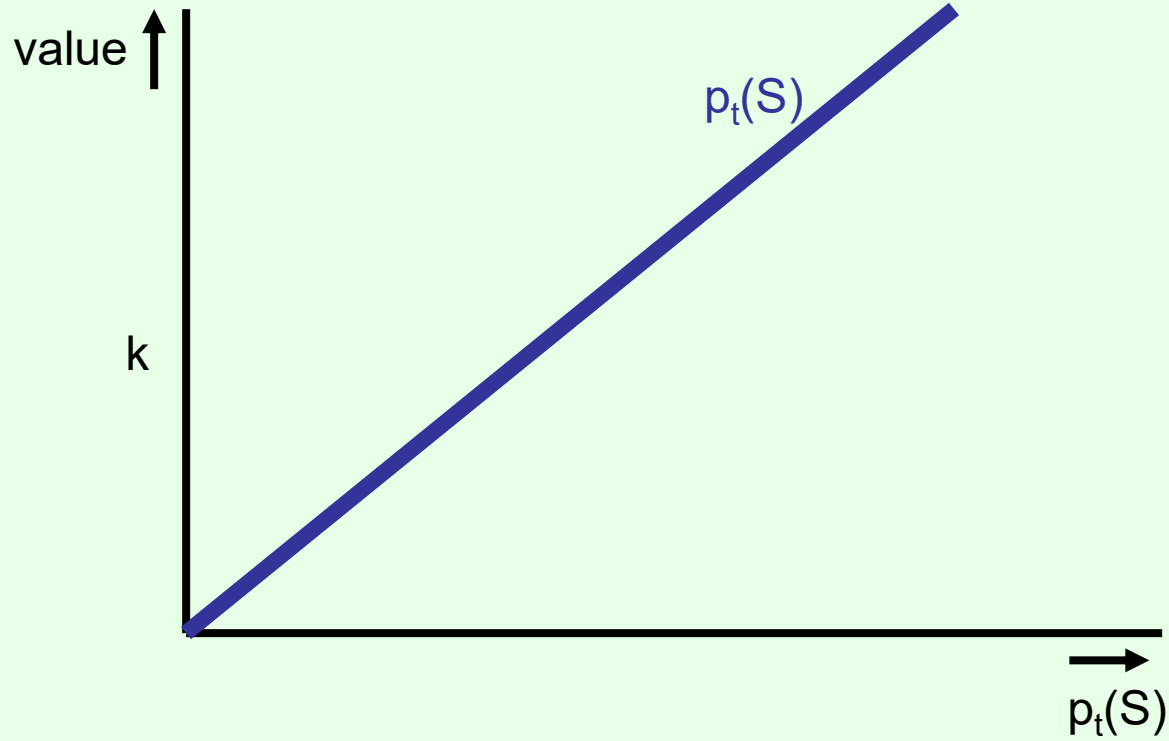


Bonds

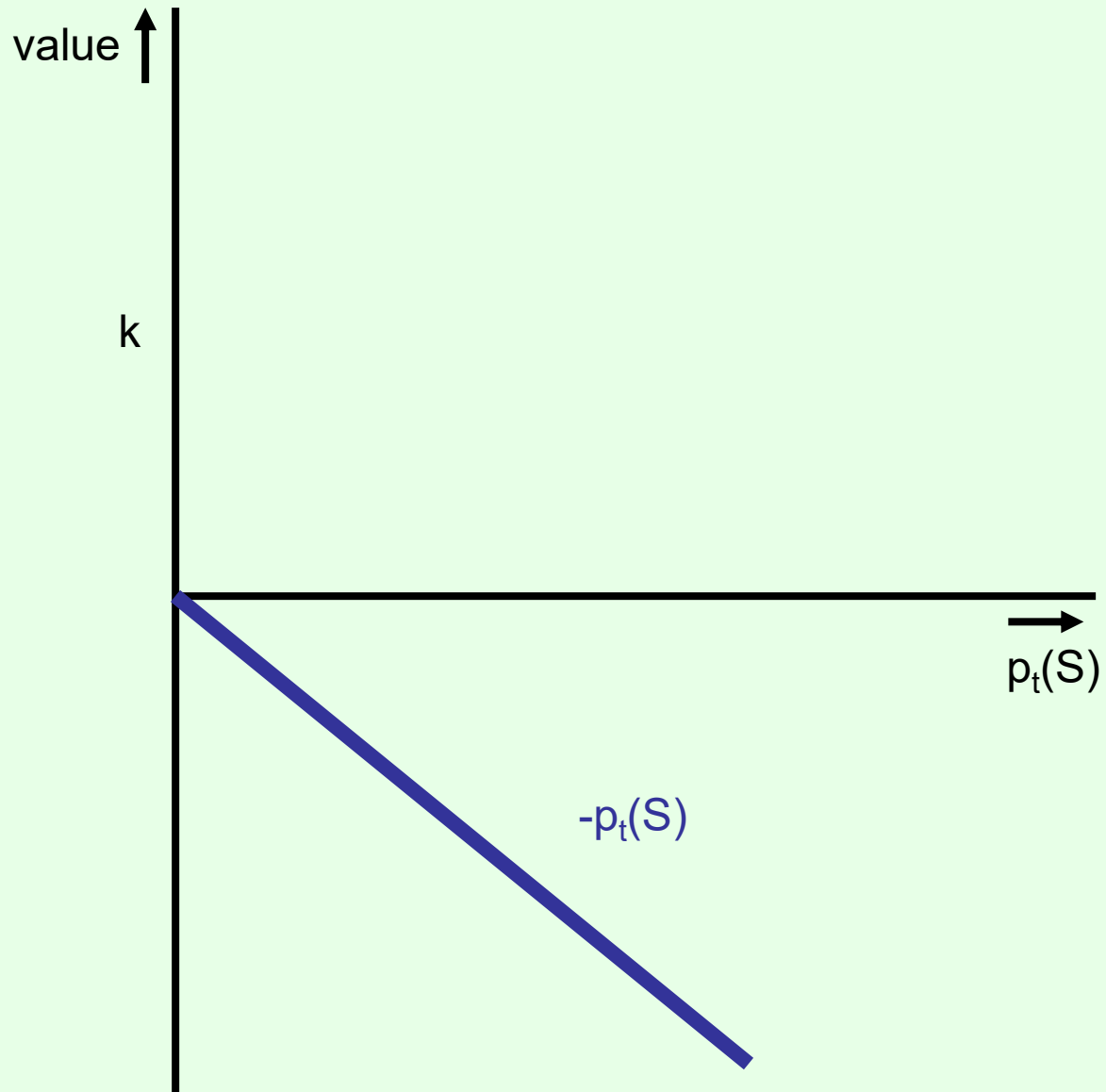
- A bond $B(k, t)$ pays off k at time t



Stocks

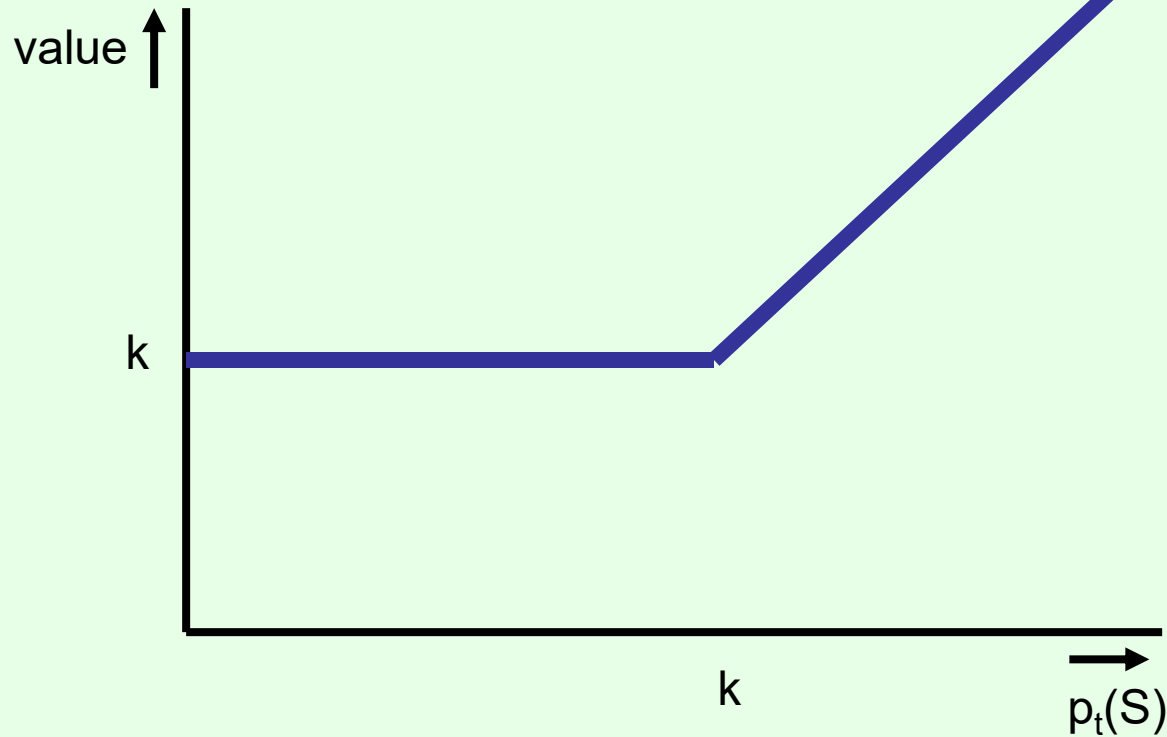


Selling a stock (short)



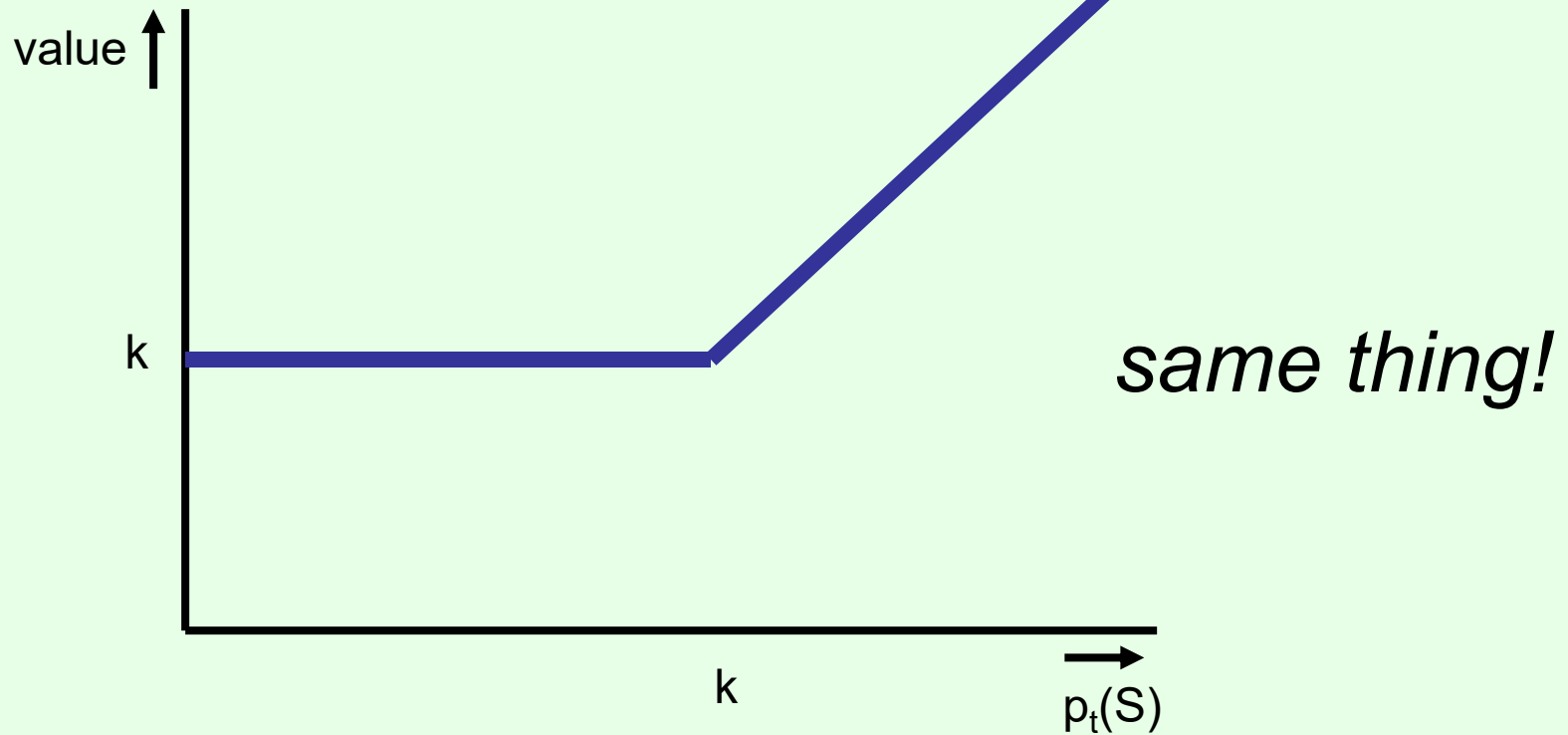
A portfolio

- One call option $C(S, k, t)$ + one bond $B(k, t)$



Another portfolio

- One put option $P(S, k, t)$ + one stock S



Put-call parity

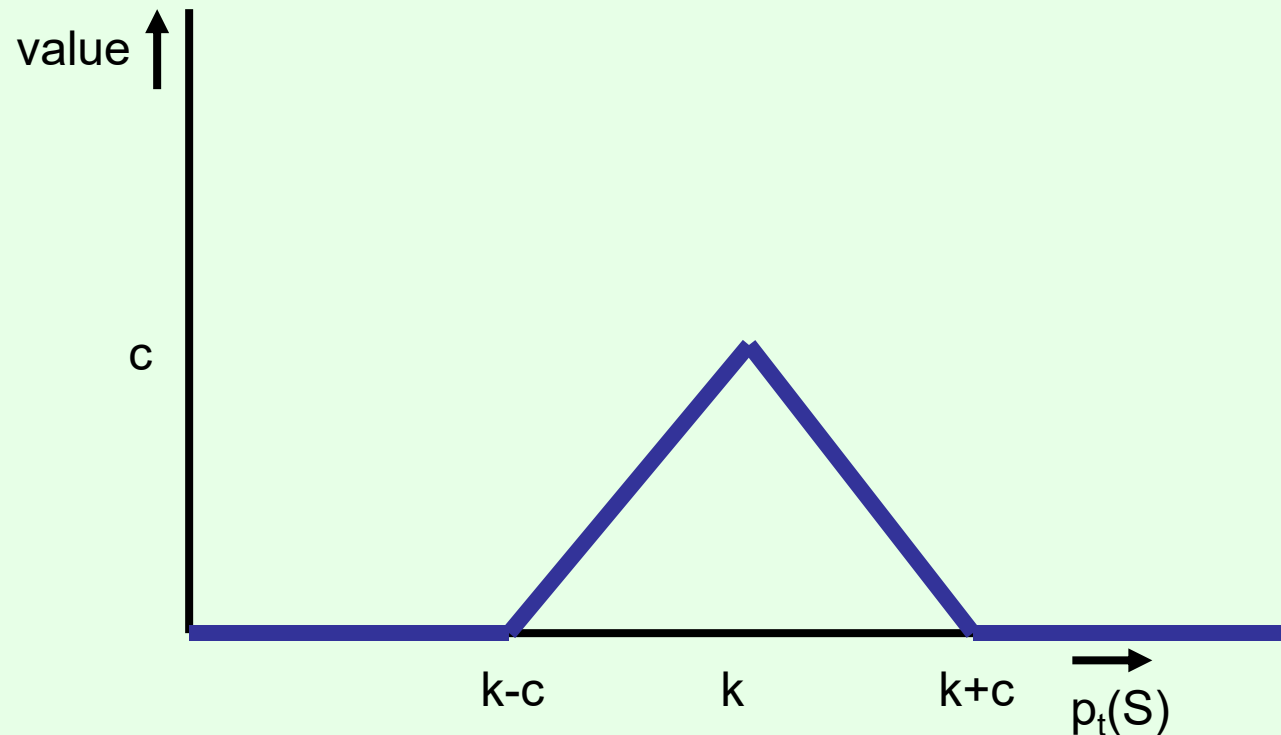
- $C(S, k, t) + B(k, t)$ will have the same value at time t as $P(S, k, t) + S$ (regardless of the value of S)
- **Assume** stocks pay no dividends
- Then, portfolio should have the same value at any time before t as well
- I.e., for any $t' < t$, it should be that $p_{t'}(C(S, k, t)) + p_{t'}(B(k, t)) = p_{t'}(P(S, k, t)) + p_{t'}(S)$
- **Arbitrage** argument: suppose (say) $p_{t'}(C(S, k, t)) + p_{t'}(B(k, t)) < p_{t'}(P(S, k, t)) + p_{t'}(S)$
- Then: buy $C(S, k, t) + B(k, t)$, sell (short) $P(S, k, t) + S$
- Value of portfolio at time t is 0
- Guaranteed profit!

Another perspective: auctioneer

- **Auctioneer** receives buy and sell offers, has to choose which to accept
- E.g.: offers received: buy(S, \$10); sell(S, \$9)
- Auctioneer can accept both offers, profit of \$1
- E.g. (put-call parity):
 - sell(C(S, k, t), \$3)
 - sell(B(k, t), \$4)
 - buy(P(S, k, t), \$5)
 - buy(S, \$4)
- Can accept all offers at no risk!

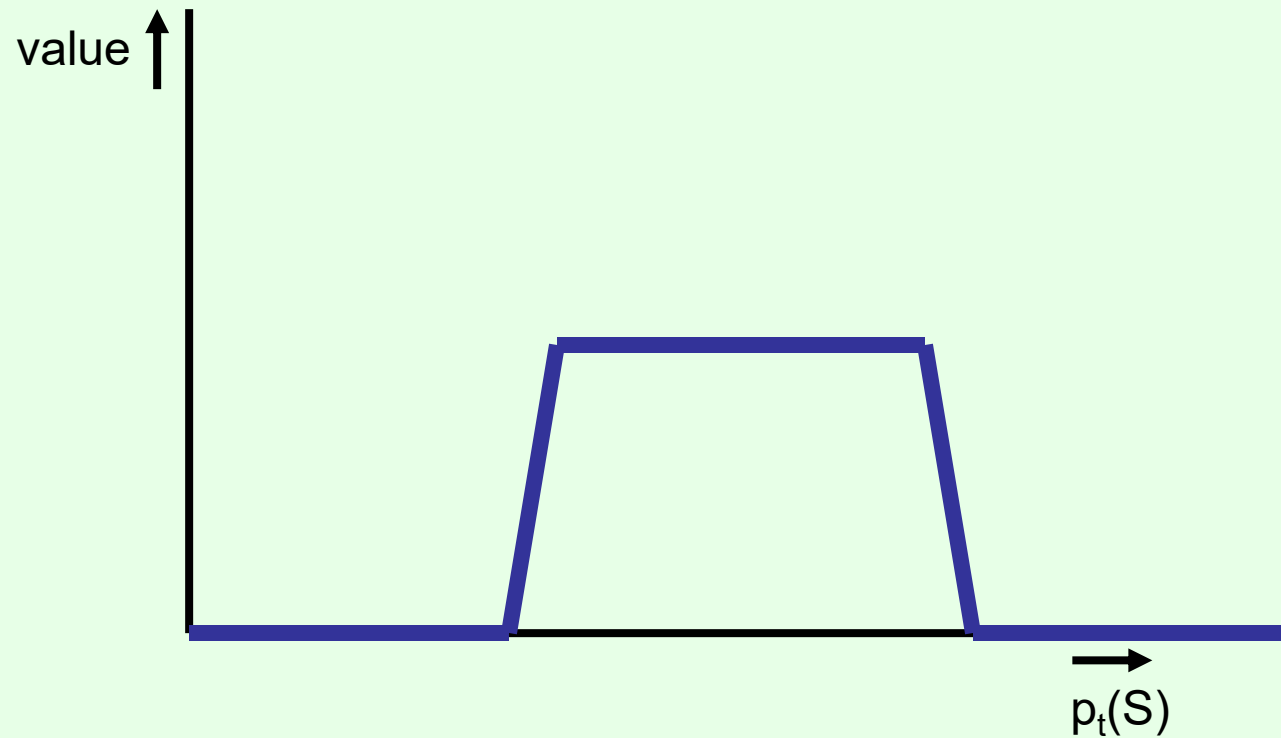
“Butterfly” portfolio

- 1 call at strike price $k-c$
- -2 calls at strike k
- 1 call at strike $k+c$



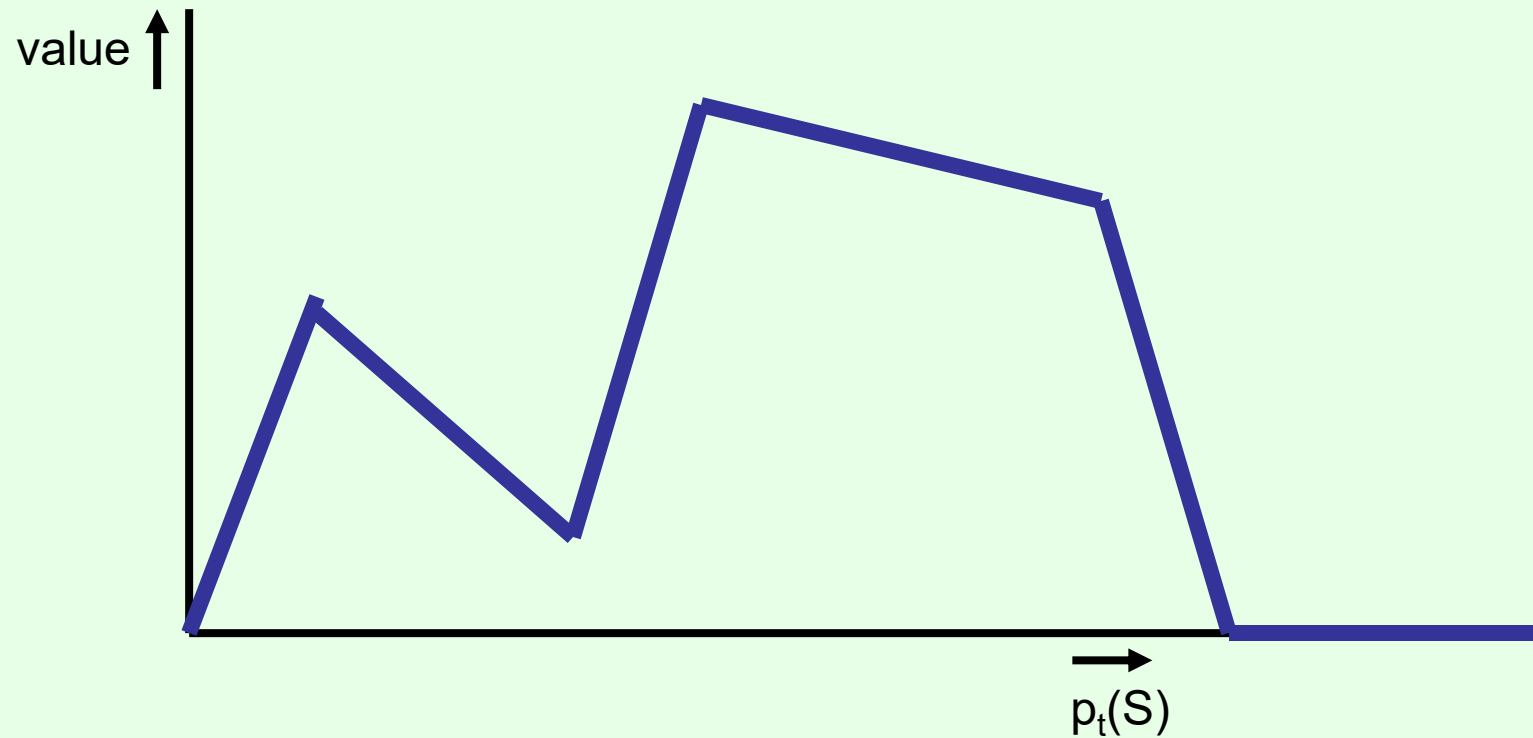
Another portfolio

- Can we create this portfolio?



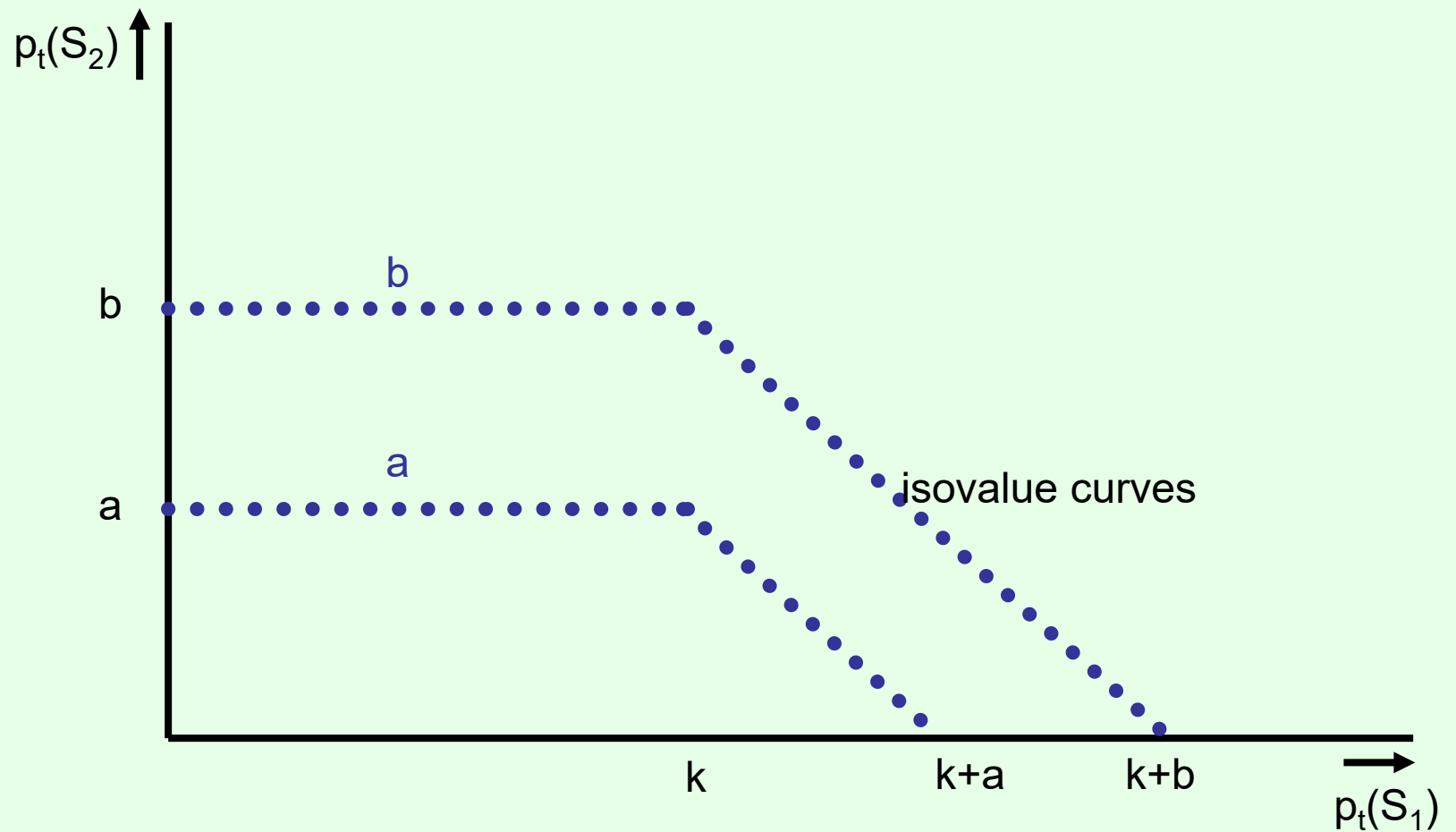
Yet another portfolio

- How about this one?



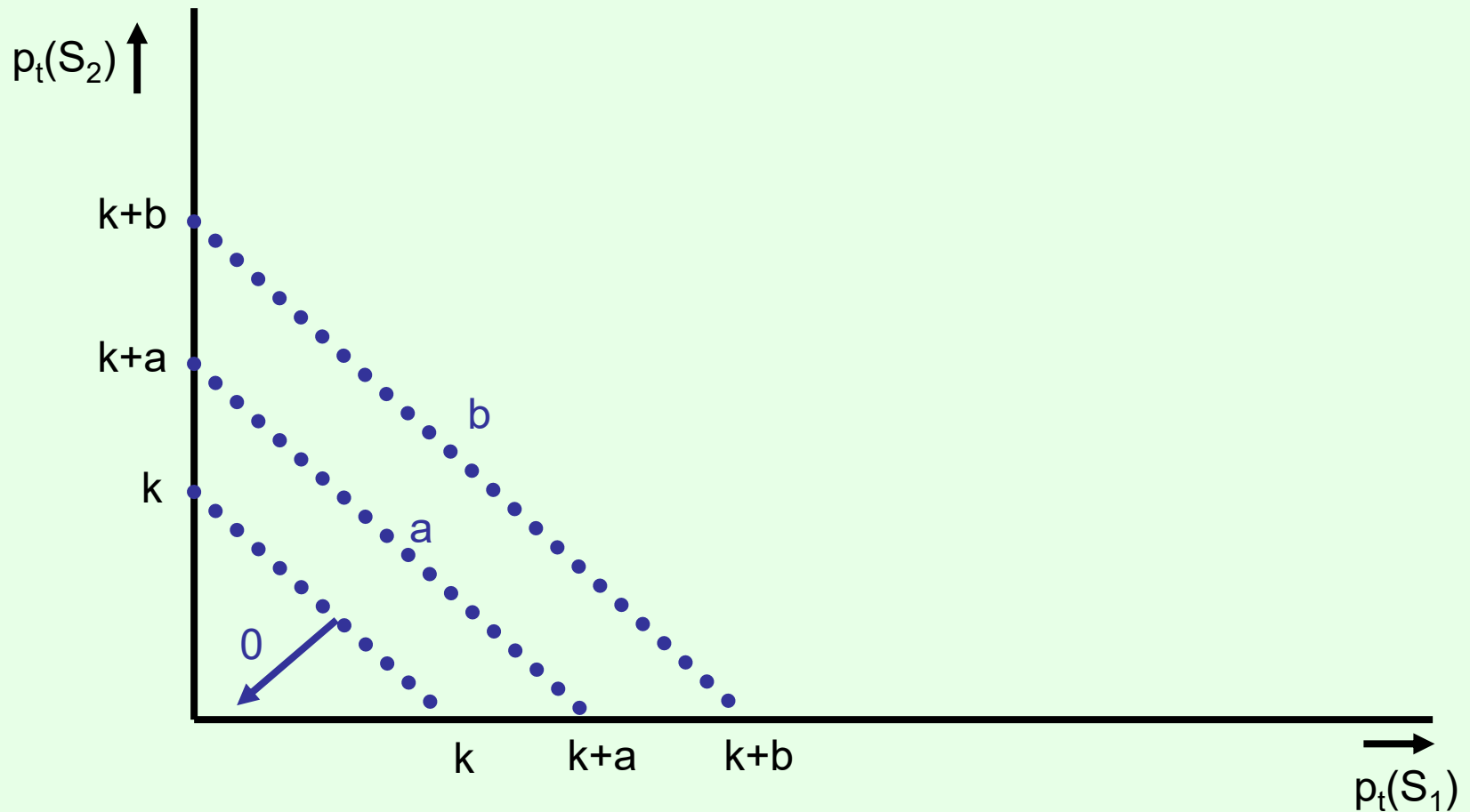
Two different stocks

- A portfolio with $C(S_1, k, t)$ and S_2



Another portfolio

- Can we create this portfolio?
(In effect, a call option on S_1+S_2)

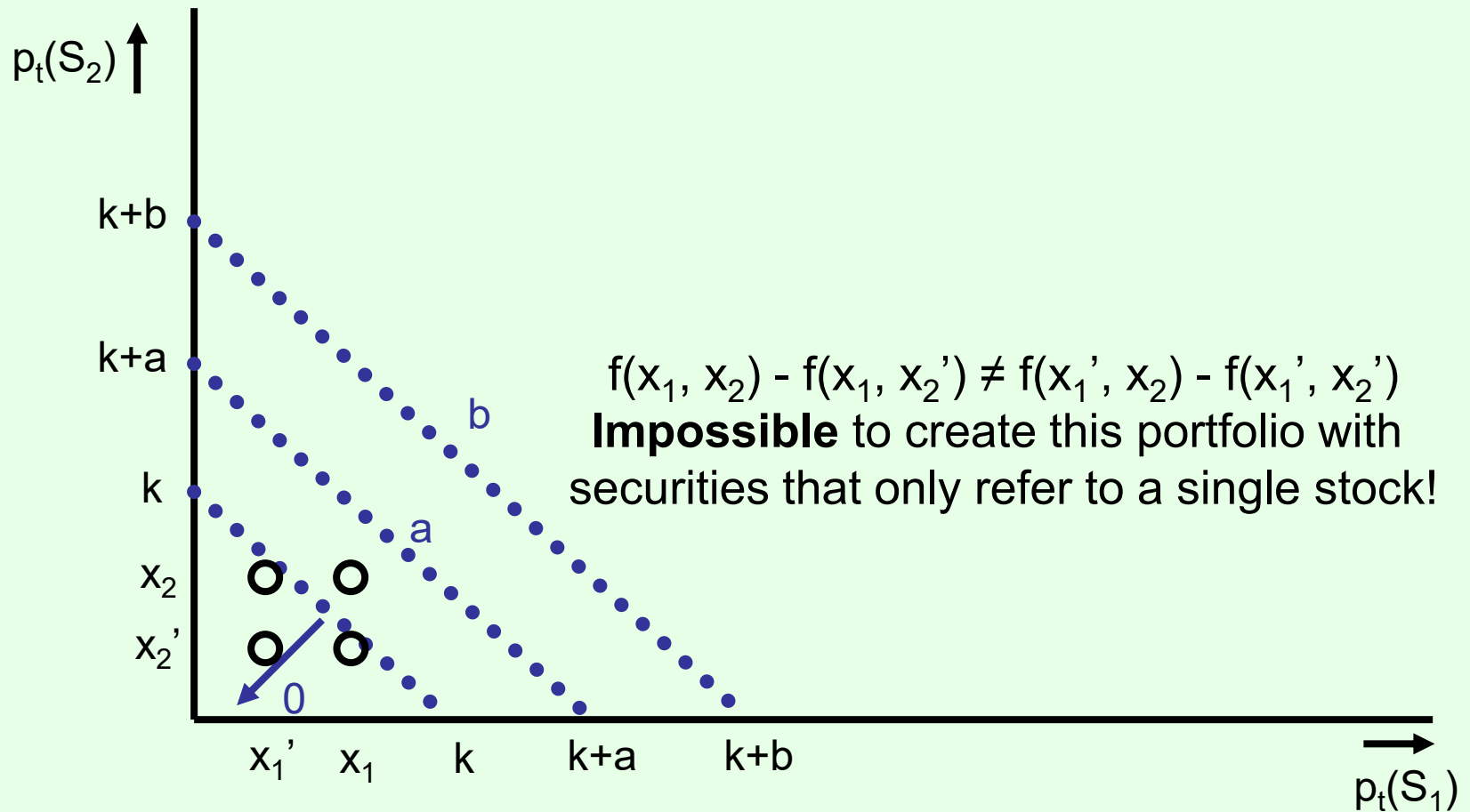


A useful property

- Suppose your portfolio pays off $f(p_t(S_1), p_t(S_2)) = f_1(p_t(S_1)) + f_2(p_t(S_2))$ (additive decomposition over stocks)
- This is all we know how to do
- Then: $f(x_1, x_2) - f(x_1, x_2') = f(x_1) + f(x_2) - f(x_1) - f(x_2') = f(x_2) - f(x_2') = f(x_1', x_2) - f(x_1', x_2')$

Portfolio revisited

- Can we create this portfolio?
(In effect, a call option on $S_1 + S_2$)



Securities conditioned on finite set of outcomes

- E.g., PredictIt: security that pays off 1 if Trump is the Republican nominee in 2020
- Can we construct a portfolio that pays off 1 if Sanders is the Democratic nominee AND Trump is the Republican nominee?

| | Trump not nom. | Trump nom. |
|------------------|----------------|------------|
| Sanders not nom. | \$0 | \$0 |
| Sanders nom. | \$0 | \$1 |









Arrow-Debreu securities

- Suppose S is the set of **all** states that the world can be in tomorrow
- For each s in S , there is a corresponding Arrow-Debreu security that pays off 1 if s happens, 0 otherwise
- E.g., s could be: Sanders is nominee and Trump is nominee and S_1 is at \$4 and S_2 at \$5 and butterfly 432123 flaps its wings in Peru and...
- Not practical, but conceptually useful
- Can think about Arrow-Debreu securities **within a domain** (e.g., states only involve stock trading prices)
- Practical for small number of states




With Arrow-Debreu securities you can do anything...

- Suppose you want to receive \$6 in state 1, \$8 in state 2, \$25 in state 3
- ... simply buy 6 AD securities for state 1, 8 for state 2, 25 for state 3
- Linear algebra: Arrow-Debreu securities are a basis for the space of all possible securities

The auctioneer problem

- Tomorrow there must be one of   
- Agent 1 offers \$5 for a security that pays off \$10 if  or 
- Agent 2 offers \$8 for a security that pays off \$10 if  or 
- Agent 3 offers \$6 for a security that pays off \$10 if 
- Can we accept some of these at offers **at no risk?**

Reducing auctioneer problem to ~combinatorial exchange winner determination problem

- Let (x, y, z) denote payout under  ,  ,  , respectively
- Previous problem's bids:
 - 5 for $(0, 10, 10)$
 - 8 for $(10, 0, 10)$
 - 6 for $(10, 0, 0)$
- Equivalently:
 - $(-5, 5, 5)$
 - $(2, -8, 2)$
 - $(4, -6, -6)$
- Sum of accepted bids should be $(\leq 0, \leq 0, \leq 0)$ to have no risk
- Sometimes possible to partially accept bids

A bigger instance (4 states)

- Objective: maximize our **worst-case** profit
- 3 for $(0, 0, 11, 0)$
- 4 for $(0, 2, 0, 8)$
- 5 for $(9, 9, 0, 0)$
- 3 for $(6, 0, 0, 6)$
- 1 for $(0, 0, 0, 10)$

- What if they are partially acceptable?

Settings with large state spaces

- Large = exponentially large
 - Too many to write down
- Examples:
- $S = S_1 \times S_2 \times \dots \times S_n$
 - E.g., $S_1 = \{\text{Sanders not nom.}, \text{Sanders nom.}\}$, $S_2 = \{\text{Trump not nom.}, \text{Trump nom.}\}$, $S = \{(-C, -T), (-C, +T), (+C, -T), (+C, +T)\}$
 - If all S_i have the same size k , there are k^n different states
- S is the set of all rankings of n candidates
 - E.g., outcomes of a horse race
 - $n!$ different states (assuming no ties)

Bidding languages

- How should **trader** (bidder) express preferences?
- Logical bidding languages [Fortnow et al. 2004]:
 - (1) “If Trump nominated OR (Pence nominated AND Sanders nominated), I want to receive \$10; I’m willing to pay \$6 for this.”
- If the state is a ranking [Chen et al. 2007] :
 - (2a) “If horse A ranks 2nd, 3rd, or 4th I want to receive \$10; I’m willing to pay \$6 for this.”
 - (2b) “If one of horses A, C, D rank 2nd, I want to receive \$10; I’m willing to pay \$6 for this.”
 - (2c) “If horse A ranks ahead of horse C, I want to receive \$10; I’m willing to pay \$6 for this.”
- Winner determination problem is NP-hard for all of these, except for (2a) and (2b) which are in P **if** bids can be partially accepted

A different computational problem

closely related to ([separation problem](#) for) winner determination

- Given that the auctioneer has accepted some bids, what is the worst-case outcome (state) for the auctioneer?
- For example:
 - Must pay 2 to trader A if horse X or Z is first
 - Must pay 3 to trader B if horse Y is first or second
 - Must pay 6 to trader C if horse Z is second or third
 - Must pay 5 to trader D if horse X or Y is third
 - Must pay 1 to trader E if horse X or Z is second

Reduction to weighted bipartite matching

- Must pay 2 to trader A if horse X or Z is first
- Must pay 3 to trader B if horse Y is first or second
- Must pay 6 to trader C if horse Z is second or third
- Must pay 5 to trader D if horse X or Y is third
- Must pay 1 to trader E if horse X or Z is second

