Arbitrage
จากแนวคิดสู่การปฏิบัติ

Part II-A: วิธีคิดแบบ Arbitrageur (บท 4-6)

หลักการคิด 7 ข้อ, Generalized Option Thinking,

Payoff Algebra, Epistemology — สอนวิธี "มอง" ไม่ใช่แค่ "สูตร"

บทที่ 4: หลักการคิดแบบ Arbitrageur

บทนี้ไม่ได้สอนสูตร — สอน**วิธีมองโลก**ที่ทำให้เห็น Arb ที่คนอื่นมองไม่เห็น

4.1 หลักการที่ 1: "ของเดียวกันต้องราคาเดียวกัน"

ทุกครั้งที่เห็นสินทรัพย์ ถามว่า: **"มีวิธีอื่นสร้าง payoff เดียวกันไหม? ราคาเท่ากันไหม?"**

> **ฝึก**
>
> เห็น SET50 Futures → ถามทันที: "Spot × e^((r-d)T) เท่าไร? ตรงกับ Futures price ไหม?"
>
> เห็น Call K=100 → ถามทันที: "P + S - PV(K) เท่าไร? ตรงกับ Call price ไหม?"

4.2 หลักการที่ 2: "ถ้าสร้างเองได้ = รู้ราคา"

ราคาที่ถูกต้อง = ต้นทุน replicate ถ้าราคาจริง ≠ ต้นทุน replicate → opportunity

4.3 หลักการที่ 3: "ได้อะไรมาต้องเสียอะไรไป"

ถ้าเจอ "กำไรแน่นอน 5% ต่อวัน" → **99% มี risk ซ่อนอยู่**

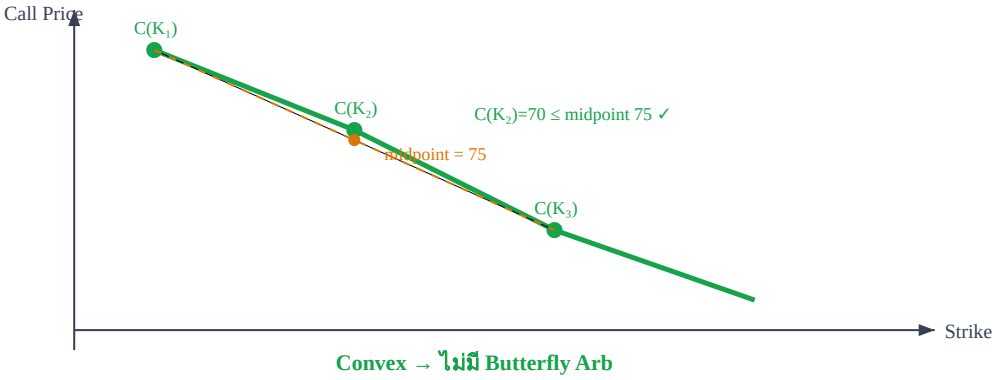
> **Devil's Advocate Checklist**
>
> เจอ "arb" แล้วถามตัวเอง 6 ข้อ:
>
> 1. ข้อมูลผิดหรือเปล่า? (stale quote, wrong symbol)
> 2. ค่าธรรมเนียมคิดครบไหม?
> 3. Liquidity จริงไหม? (bid-ask wide, size เล็ก)
> 4. Execute ได้ทั้ง 2 ขาพร้อมกันไหม?
> 5. มี structural reason ไหม? (ex-div, corporate action)
> 6. **ทำไมคนอื่นไม่ทำ?** ← ข้อสำคัญที่สุด

4.4 หลักการที่ 4: "ไม่รู้ exact แต่รู้ขอบเขต"

ราคานอกขอบเขต = Arb ที่พิสูจน์ได้ เช่น Call ≥ max(0, $S-Ke^{-rT}$) ← ถ้า Call < ค่านี้ → ซื้อทันที

4.5 หลักการที่ 5: "ลำดับต้องถูก"

```
Monotonicity: C(K₁) ≥ C(K₂) เมื่อ K₁ < K₂ ← Call ถูกลงเมื่อ Strike สูงขึ้น
Convexity: C(K₁) - 2C(K₂) + C(K₃) ≥ 0 ← Butterfly ต้อง ≥ 0
```



4.6 หลักการที่ 6: "Call บอก Put, Put บอก Call"

จาก PCP: รู้ Call → คำนวณ Put ได้ทันที ถ้า market price ≠ theoretical → opportunity

4.7 หลักการที่ 7: Generalized Option Thinking ★

> **แนวคิดจาก quant-arb-engine**
>
> **"ทุกอย่างที่มี conditional payoff คือ option"** — ไม่ใช่แค่ Call/Put บน exchange

สิ่งที่เห็น	Hidden Option ที่ซ่อนอยู่	วิธีคิด
Funding rate	Option on carry regime	จ่ายถ้า basis กว้าง รับถ้าแคบ
Stop-loss order	Synthetic put (มี execution risk)	"ถ้าราคาลงถึง X ขายทันที" = put ที่อาจ slip
Earn yield 11%	Option on platform solvency	ได้ 11% ถ้า platform ไม่ล้ม
PM Yes/No	Native digital option	จ่าย $1 ถ้าเหตุการณ์เกิด
Weekend price gap	Option on reference absence	ราคา crypto เคลื่อนไหวแต่ TradFi ปิด
Convert rate gap	Instant exercise option on pricing lag	ถ้าราคา convert ≠ spot → arb ทันที
Liquidation cascade	Path-dependent barrier event	"ถ้าราคาลงถึง X → forced selling เพิ่ม"

กระบวนการ: เห็น conditional payoff → จัดหมวด → ใช้ประโยชน์

ทุกครั้งที่เจอ conditional payoff ถามว่า:

1. **Underlying** คืออะไร? (ราคาหุ้น, solvency, เวลา?)
2. **Strike/threshold** อยู่ที่ไหน?
3. **Expiry** เมื่อไร?
4. **Shape**: linear, convex, binary?
5. **Long/Short**?
6. สามารถ **combine** กับ block อื่นได้ไหม?

บทที่ 5: Payoff Algebra — "เขียนสิ่งที่อยากได้เป็นสมการ แล้ว solve"

หนังสือส่วนใหญ่สอนย้อนกลับ: "นี่คือ Bull Call Spread มันมีหน้าตาแบบนี้"
บทนี้สอนไปข้างหน้า: **"คุณอยากได้ payoff แบบนี้ → ต้องใช้ instruments อะไร?"**

5.1 Building Blocks — ชิ้นส่วนที่ใช้ประกอบ

2 ชั้น ที่ต้องคิดพร้อมกัน:

Platform Blocks → กำหนดเศรษฐศาสตร์

Earn Yield, Convert Rate, Promo Rate, Margin Offset, Staking

"ต้นทุน, yield, margin ลดได้ไหม"

ผสมทั้ง 2 ชั้น!

Instrument Blocks → กำหนดรูปร่าง Payoff

Spot, Futures, Call, Put, Spread, PM Above, PM Range, Box, Conversion

"ถ้าราคาเป็นแบบนี้ → ได้/เสียเท่าไร"

5.2 Step 1: เขียน Desired Payoff เป็น Piecewise Function

ตัวอย่าง: "อยากได้กำไรถ้าราคาขึ้น แต่ไม่เกิน 2500 ไม่อยากเสียถ้าลง"

แปลเป็น piecewise:

```
P(S) = { 0            ถ้า S < 2000 (ราคาปัจจุบัน)
       { S - 2000     ถ้า 2000 ≤ S ≤ 2500
       { 500          ถ้า S > 2500
```

5.3 Step 2: Decompose เป็น Known Building Blocks

```
Building block payoffs:
  Long Call(K)     = max(S - K, 0)
  Short Call(K)    = -max(S - K, 0)
```

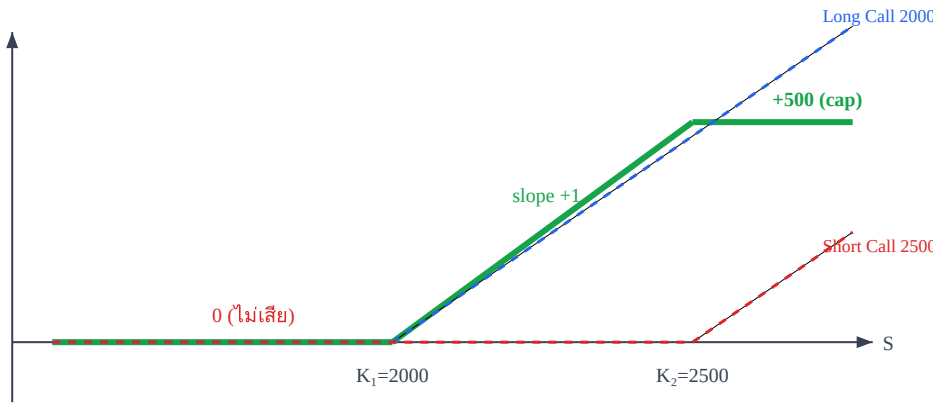
```
  Call Spread(K₁,K₂) = max(S-K₁, 0) - max(S-K₂, 0)
  PM Above Yes(K) = 1 ถ้า S > K, else 0
```

5.4 Step 3: Solve — หา instruments ที่ตรงกัน

```
P(S) = { 0 ถ้า S < 2000, S-2000 ถ้า 2000≤S≤2500, 500 ถ้า S > 2500 }

นี่คือ: Call Spread(2000, 2500)
= max(S-2000, 0) - max(S-2500, 0)
= { 0        ✓ | S-2000   ✓ | 500 ✓ }

คำตอบ: ซื้อ 1× Call Spread (2000/2500)
```



= Call Spread(2000, 2500) → ซื้อ 1 ชุดจบ!

5.5 ตัวอย่างซับซ้อน: Piecewise → PM + Options

```
"อยากได้ +$100 ถ้า S > 2500, +$50 ถ้า 2200 < S ≤ 2500, -$20 ถ้า S ≤ 2200"

Decompose by jumps:
  ที่ S=2200: payoff เปลี่ยนจาก -20 เป็น +50 (กระโดด +70)
  ที่ S=2500: payoff เปลี่ยนจาก +50 เป็น +100 (กระโดด +50)

P(S) = -20 + 70 × 1(S > 2200) + 50 × 1(S > 2500)

คำตอบ:
  -$20 (ต้นทุน entry)
```

```
  + 70 × PM Above Yes($2200) ← จ่าย $70 ถ้าเกิน 2200
  + 50 × PM Above Yes($2500) ← จ่าย $50 ถ้าเกิน 2500
```

พลังของ Payoff Algebra

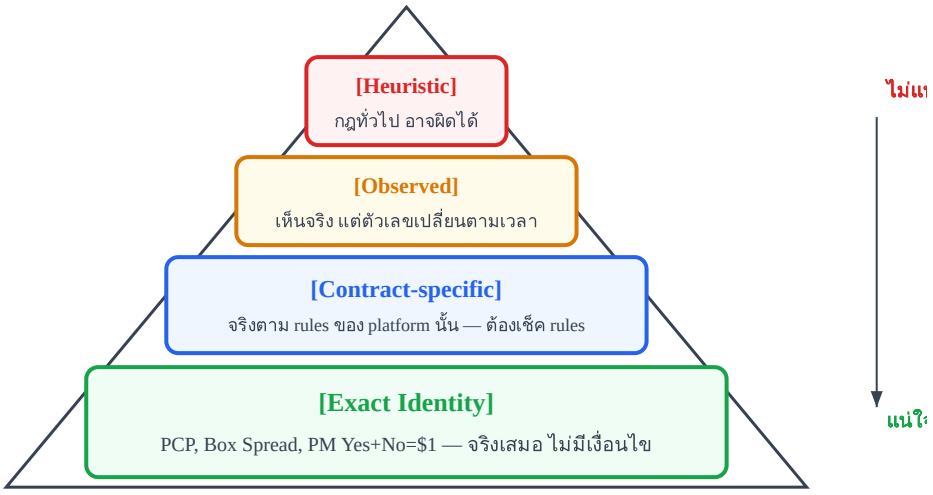
ไม่ต้องจำชื่อกลยุทธ์ — **เขียนสิ่งที่อยากได้เป็นสมการ → solve → ได้ instruments**

สร้าง payoff shape อะไรก็ได้ที่นึกออก → ถ้า instruments มีอยู่ → สร้างได้!

บทที่ 6: Epistemology — "รู้ว่ารู้อะไร แน่ใจแค่ไหน"

Arb ที่ดีต้องแยกได้ว่า: **"สิ่งที่รู้แน่ 100%" vs "สิ่งที่แค่เดา"** — ถ้าแยกไม่ได้ จะ treat heuristic เหมือน exact identity → อันตราย

6.1 4 ระดับความแน่ใจ



6.2 ตัวอย่าง: Label Claims ที่ถูกต้อง

Claim	Label ที่ถูก	ทำไม
$C + Ke^{-rT} = P + S$	[Exact Identity]	จริงเสมอ ไม่มีเงื่อนไข (European)
PM Yes + PM No = $1.00	[Exact Identity]	โดย contract design
Polymarket ใช้ Binance price เป็น resolver	[Contract-specific]	อาจเปลี่ยน platform ได้ ต้องเช็ค
BTC funding rate = -0.002% เมื่อวาน	[Observed]	จริงเมื่อวาน วันนี้อาจต่างแล้ว
Funding rate มักจะ mean-revert	[Heuristic]	มักจริง แต่ไม่เสมอ (ช่วง bull run funding สูงนาน)
IV > RV → ขาย options ได้กำไร	[Heuristic]	ส่วนใหญ่จริง แต่ tail event ทำให้เจ็บหนัก

Common Mistake: Treat Heuristic เหมือน Exact Identity

"Funding rate จะกลับมา" ← ถ้า treat เหมือน [Exact Identity] → size ใหญ่เกินไป → เจ็บหนักเมื่อ funding ไม่กลับ

ต้อง label ว่า [Heuristic] → size เล็กลง → รับได้ถ้าผิด

6.3 วิธีใช้: Label ทุก claim ก่อนตัดสินใจ

```
เจอ "arb" → ก่อน execute ถามว่า:

1. Edge มาจากอะไร?
   → ถ้า PCP violation → [Exact] → size ได้เต็มที่
   → ถ้า "IV สูงเกินไป" → [Heuristic] → size ต้องเล็ก

2. ข้อมูลที่ใช้เป็น level ไหน?
   → ถ้า live quote → [Observed] → ยังต้องเช็ค stale ไหม
   → ถ้า "เมื่อเดือนก่อน IV ≈ 30%" → [Observed - stale] → ใช้ไม่ได้!
```

สรุป Part II-A (บท 4-6)

ตอนนี้คุณมี:

✓ **7 หลักการคิด** ที่ช่วยมองเห็น Arb

✓ **Generalized Option Thinking** — ทุก conditional payoff คือ option

✓ **Payoff Algebra** — เขียนสิ่งที่อยากได้ → solve → ได้ instruments

✓ **Epistemology** — แยกได้ว่า claim แน่ใจระดับไหน → size ตาม

→ **Part II-B: Arb Taxonomy, Non-Negative Design, Discovery Pipeline, Belief Patterns**

จบ Part II-A