Part 1: Understanding Our Philippine-Specific Machine (0:30 - 1:15)

[Zoom into the main display showing current state: IDLE]

"At the heart of our system is this vending machine with seven distinct states: **IDLE**, **COIN_INSERTED**, **ITEM_SELECTED**, **DISPENSING**, **INSUFFICIENT_FUNDS**, **OUT_OF_STOCK**, and **RETURNING_CHANGE**."

[Point to the state display showing 'IDLE']

"Right now, we're in the **IDLE** state - waiting for customer interaction. This is our initial state q_0."

[Hover over coin buttons showing ₱5, ₱10, ₱20]

"Our alphabet Σ has six possible inputs, but note our coin denominations: only ₱5, ₱10, and ₱20 coins are accepted. This reflects real-world Philippine vending machine constraints."

Part 2: The Basic Transaction with Philippine Pricing (1:15 - 2:15)

[Click on ₱20 coin button]

"Let's start a transaction with Philippine currency. When I send an INSERT_COIN input with value ₱20..."

[Watch transition: IDLE → COIN_INSERTED]

"...we transition from **IDLE** to **COIN_INSERTED**. Our balance is now ₱20."

[Point to the Water product (₱20)]

"Notice our products have Philippine-specific pricing. Water costs exactly ₱20. Let me select it..."

[Click on Water product]

"SELECT_ITEM input with exact balance match..."

[Watch transition: COIN_INSERTED → ITEM_SELECTED]

"...takes us to **ITEM_SELECTED** state. Perfect match - no change needed!"

Part 3: Complex Philippine Transaction (2:15 - 3:00)

[Click Cancel button to return to IDLE, then start new transaction]

"Let's try a more complex Philippine purchase. I'll buy Chocolate for ₱35."

[Click ₱20 coin button, then ₱10 coin button, then ₱5 coin button]

"INSERT_COIN(₱20) → INSERT_COIN(₱10) → INSERT_COIN(₱5) = ₱35 total."

[Point to balance display showing ₱35]

"We now have exactly ₱35 - the price of Chocolate."

[Click on Chocolate product (₱35)]

"SELECT_ITEM input with exact amount..."

[Watch transition: COIN_INSERTED → ITEM_SELECTED]

"...successfully selects the Chocolate."

Part 4: Insufficient Funds - Philippine Pricing Challenge (3:00 - 3:45)

[Click Cancel, start new transaction with ₱20 + ₱10 = ₱30]

"Now let's see what happens with insufficient funds. I have ₱30 but want Cola for ₱25..."

[Actually select Cola product]

"SELECT_ITEM input with ₱30 for ₱25 item..."

[Watch transition: COIN_INSERTED → ITEM_SELECTED]

"...actually transitions to **ITEM_SELECTED** because we have MORE than enough! The issue is having EXACT CHANGE with Philippine coins."

[Click Cancel, start new with only ₱20]

"Now with only ₱20 trying to buy Cola for ₱25..."

[Click Cola product]

"SELECT_ITEM input with insufficient funds..."

[Watch transition: COIN_INSERTED → INSUFFICIENT_FUNDS]

"...correctly goes to **INSUFFICIENT_FUNDS**. We need ₱5 more."

Part 5: The Formal DFA Model with Philippine Constraints (3:45 - 4:30)

[Zoom into the state diagram area]

"Let me show you the formal mathematical model with Philippine constraints:"

[Point to each component as you mention them]

1. **States (Q):** Same 7 states, but transitions now handle ₱5/₱10/₱20 increments
2. **Alphabet (Σ):** {INSERT_COIN(₱5), INSERT_COIN(₱10), INSERT_COIN(₱20), SELECT_ITEM, CANCEL, CONFIRM_PURCHASE, RESET, COLLECT_CHANGE}
3. **Transition Function (δ):** Now includes logic for Philippine coin combinations
4. **Initial State (q_0):** IDLE
5. **Accepting States (F):** {IDLE, DISPENSING}

[Highlight specific transition rules in the log]

"Key Philippine-specific rules:

- δ(COIN_INSERTED, SELECT_ITEM) = ITEM_SELECTED if balance ≥ price (exact change not required)
- But change can only be in ₱5, ₱10, ₱20 denominations
- Example: Buying ₱22 Lemonade with ₱30 gives ₱8 change... which is IMPOSSIBLE with our coins!"

Part 6: Change Calculation Challenge (4:30 - 5:15)

[Start transaction: ₱20 + ₱10 = ₱30, select Lemonade ₱22]

"Watch this interesting edge case. I have ₱30, Lemonade is ₱22..."

[Click Confirm Purchase]

"CONFIRM_PURCHASE input..."

[Watch: ITEM_SELECTED → DISPENSING → RETURNING_CHANGE]

"...dispenses, but calculates change: ₱30 - ₱22 = ₱8"

[Point to change display showing ₱8]

"₱8 change! But we only have ₱5, ₱10, ₱20 coins. The machine cannot give ₱3 or ₱8 change!"

[Point to log entry about change calculation]

"This reveals a real-world constraint: vending machines often have minimum pricing or specific price points to avoid impossible change situations."

Part 7: Philippine Product Pricing Analysis (5:15 - 5:45)

[Point to product grid]

"Look at our Philippine product pricing:

- Water: ₱20 (divisible by 5)
- Cola: ₱25 (divisible by 5)
- Cookies: ₱28 (NOT divisible by 5!)
- Energy Drink: ₱40 (divisible by 5, 10, 20)

Products priced at ₱22, ₱24, ₱27, ₱28, ₱35 create change calculation challenges with ₱5/₱10/₱20 coins."

[Demonstrate buying Cookies ₱28 with ₱30]

"Cookies: ₱28 with ₱30 = ₱2 change (impossible!)
Lemonade: ₱22 with ₱30 = ₱8 change (impossible!)
This shows why real vending machines often have prices ending in 0 or 5!"

Part 8: Real-World Connection & Conclusion (5:45 - 6:15)

[Show the full simulation with highlighted impossible change scenarios]

"This simulation demonstrates something profound: even with deterministic automata theory, real-world constraints (like Philippine coin denominations) create design challenges."

"The DFA model is perfect, but the real-world implementation must account for:

1. Coin denomination limitations
2. Change calculation constraints
3. Product pricing strategies
4. Customer experience with exact change"

[Reset to initial state]

"This project shows:

1. How DFAs model currency-based transaction systems
2. The importance of real-world constraints in automata design
3. Formal language recognition with financial transactions
4. Practical challenges in implementing theoretical models"

Final Screen (6:15 - 6:30)

[Show interface with log showing change calculation challenges]

"Thank you for watching. This Philippine vending machine DFA not only demonstrates Automata Theory but also highlights the fascinating intersection between perfect mathematical models and imperfect real-world constraints."