A simple story....

Mary heard the ice-cream man coming.
She remembered her pocket money.
She rushed into the house.

Questions
- What is an “ice-cream man” and why does he make noise?
- Why does she think of money?
- Why does she go into the house?
- Why does she rush?
- How old is she?

Can constraint satisfaction support structured thought?

Useful properties
- content addressibility (by name and by partial content)
- default assignments
- graceful degradation (noisy cues, removal of weights)
- spontaneous generalization

Can relationships and constraints among different types of information really be reduced to pairwise interactions among simple features/descriptors?

Schemas: data structures for representing generic concepts in memory
- objects, situations, events, actions, sequences of events/actions...

Schemas: Essential properties

Schemas have variables
- Slots have restrictions (e.g., AGENT must be animate)
- Default values (values in absence of more specific information)
  - But must be context-sensitive (agent in breaking window vs. bubble)

Schemas can embed
- BREAK contains DO and CAUSE
- Not always simpler (e.g., room with picture of room)

Schemas range across levels of abstraction
- Original focus on lexical level (like GIVE, BREAK)
- Also intended to span larger "events" (e.g., restaurant "script")

Schemas represent knowledge rather than definitions
- Not "definitional" but what is "normal"
Challenges for traditional theories of schemas

- How to select relevant schemas (best-match problem)
- How to integrate multiple schemas (birthday party in restaurant)
- How to create new schemas
  - Specialize/generalize existing ones? Hybrids?
  - Transition from single instance to “general” knowledge
  - Proliferation makes selection problem more difficult

Schemas in constraint satisfaction networks

- Situations composed of primitive “features”
- A schema consists of knowledge about what features go with other features (i.e. constraints between features)
- Certain subpatterns tend to act in concert
  - Support each other and inhibit same sets of other units (“stable coalitions”)
- Good interpretations are goodness maxima / energy minima
- No structure corresponds to a schema
  - more like a description of structured/systematic behavior of system
  - No selection: use all knowledge all the time

Schema model (Rumelhart et al., 1986)

- Two subjects each imagined 8 different versions of 5 room types
  - kitchen, office, bathroom, bedroom, living room
- For each imagined room, subject decided which of 40 descriptors applied to it
  - Network has 40 units (one per descriptor); fully connected
- Weights set based on the likelihoods, across rooms, that the two descriptors agreed (both on or both off)
- Biases set based on likelihoods that each single descriptor was included in a room
- Five room types are only implicit in pattern of weights and biases (nothing explicit)

$$w_{ij} = -\ln \frac{p(x_i = 0 \& x_j = 1)p(x_i = 1 \& x_j = 0)}{p(x_i = 1 \& x_j = 1)p(x_i = 0 \& x_j = 0)}$$

$$bias_i = -\ln \frac{p(x_i = 0)}{p(x_i = 1)}$$

Each large square is a unit whose incoming weights (white = positive; black = negative) are displayed inside it
- Internal position of each incoming weight corresponds to the sending unit’s position in large display
- Clamp on “oven” unit, run network ("Imagine a room with an oven; what else is in it?"

- Final pattern called “kitchen” prototype (although nothing explicit about a kitchen per se was involved)
Living room

Goodness surface: Bathroom, Office, Bedroom

State space (bottom of cube) is interpolation within the unique plane determined by three points (kitchen, office, and bedroom prototypes).

Goodness surface is plotted over this subset of state space.
Schemas: Essential properties

Schemas have variables
- Slots have restrictions (e.g., AGENT must be animate)
- Default values (values in absence of more specific information)
  - But must be context-sensitive (agent in breaking window vs. bubble)

Schemas can embed
- BREAK contains DO and CAUSE
- Not always simpler (e.g., room with picture of room)

Schemas range across levels of abstraction
- Original focus on lexical level (like GIVE, BREAK)
- Also intended to span larger "events" (e.g., restaurant "script")

Schemas represent knowledge rather than definitions
- Not "definitional" but what is "normal"
Challenges for traditional theories of schemas

- How to select relevant schemas (best-match problem)
- How to integrate multiple schemas (birthday party in restaurant)
- How to create new schemas
  - Specialize/generalize existing ones? Hybrids?
  - Transition from single instance to “general” knowledge
  - Proliferation makes selection problem more difficult