Article

Supplementary: Cognitive Turing Machines: A Novel Framework for AI Complexity Theory

Partha Pratim Ray †0000-0003-2306-2792

[1] Sikkim university, India; parthapratimray1986@gmail.com
* Correspondence: parthapratimray1986@gmail.com

Contents

1. Cognitive Turing Machine Use Cases **2**
1.1. Generic Scenario 2
1.2. Data Classification 3
1.3. Data Clustering 4
1.4. Data Regression 4
1.5. Object Detection 5
1.6. Image Segmentation 6
1.7. Automatic Speech Recognition 6
1.8. Text Recognition and Prediction 7
1.9. Text-to-Text Generation 8
1.10. Text-to-Image Generation 9
1.11. Image-to-Text Generation Scenario 10
1.12. Image-to-Image Generation 11
1.13. Text-to-Video Generation Scenario 12
1.14. Video-to-Text Generation 13
1.15. Video-to-Video Generation 14
1.16. Video Classification 15
1.17. Text-to-Audio Generation CTM 16
1.18. Audio-to-Audio Generation CTM 17
1.19. Audio Classification 18
1.20. Multimodal Generative AI 19
1.21. Autonomous Robot Driving Scenario 21
1.22. Autonomous Agents 21
1.23. Federated Learning 22
1.24. Reinforcement Learning 23
1.25. Neuromorphic Computing 24
1.26. Transfer Learning 25
1.27. Time Series Forecasting 27
1.28. Artificial General Intelligence 28

2. AI Complexity Classes **30**
2.1. **AI-Complete Problem** 30
2.2. **AI-Hard Problem** 30
2.3. **AI-P Complete Problem** 30
2.4. **AI-NP Complete Problem** 31
2.5. **AI-PSPACE Complete Problem** 31
2.6. **AI-APX Problem** 32
2.7. **AI-EXP Problem** 32

Citation: Ray, P.P. Cognitive Turing Machines: A Novel Framework for AI Complexity Theory. *Journal Not Specified* **2024**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

3. Quantum AI Complexity Theory **33**
3.1. Quantum Cognitive Turing Machines 33
3.2. Quantum AI Complexity Classes 34
3.3. Quantum Cognitive Advantage 34
3.4. Quantum Cognitive Entanglement 35

1. Cognitive Turing Machine Use Cases

We present various use cases based on the newly proposed theoretical concepts of Cogntive Turning Machines (CTMs) in this section. The uses cases are given as demonstration purposes and as evidence of the the effectiveness of CTMs to be used in multitude of AI relates problem solving.

1.1. Generic Scenario

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that recognizes the language $L = \{w\#w \mid w \in \{a, b\}^*\}$. This CTM uses two cognitive operations: pattern recognition (ψ_{PR}) and working memory manipulation (ψ_{WM}).

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$
- $\Sigma = \{a, b, \#\}$
- $\Gamma = \{a, b, \#, \sqcup\}$
- q_0 is the initial state
- $F = \{q_5\}$
- $T = \{T_W, T_L\}$ where T_W is the working memory tape and T_L is the long-term memory tape
- $\Psi = \{\psi_{PR}, \psi_{WM}\}$

The transition function δ is defined as follows (see Figure 1):

$$
\begin{aligned}
\delta(q_0, a, \sqcup) &= (q_1, a, \sqcup, R, R, [\psi_{WM}]) \\
\delta(q_0, b, \sqcup) &= (q_1, b, \sqcup, R, R, [\psi_{WM}]) \\
\delta(q_1, a, \sqcup) &= (q_1, a, \sqcup, R, R, [\psi_{WM}]) \\
\delta(q_1, b, \sqcup) &= (q_1, b, \sqcup, R, R, [\psi_{WM}]) \\
\delta(q_1, \#, \sqcup) &= (q_2, \#, \sqcup, R, L, [\psi_{PR}]) \\
\delta(q_2, x, x) &= (q_2, x, x, R, L, [\psi_{PR}]) \text{ for } x \in \{a, b\} \\
\delta(q_2, \sqcup, \sqcup) &= (q_5, \sqcup, \sqcup, R, R, []) \\
\delta(q_2, x, y) &= (q_3, x, y, R, R, []) \text{ for } x \neq y, x, y \in \{a, b, \sqcup\}
\end{aligned}
$$

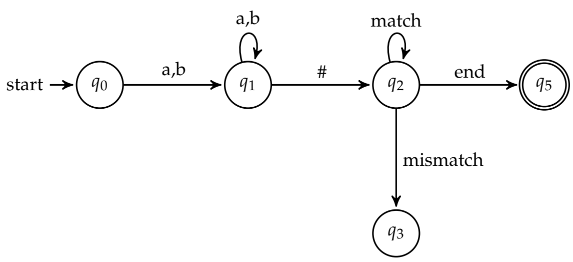

Figure 1. State diagram of the Cognitive Turing Machine.

This CTM works as follows:

1. It reads the first part of the input (before #) and stores it in the working memory tape using ψ_{WM}.
2. When it encounters #, it starts comparing the second part with the stored pattern using ψ_{PR}.

3. If all characters match and both parts end simultaneously, it accepts; otherwise, it rejects.

The CRB for this CTM on an input of length n is:

$$CRB_M(n) = \max\{O(n), O(n), O(1), O(n)\}$$

Where the components represent:

- $C_M(n) = O(n)$ (number of distinct configurations)
- $O_M(n) = O(n)$ (number of cognitive operations)
- $I_M(n) = O(1)$ (interactions between tapes)
- $E_M(n) = O(n)$ (cognitive energy expenditure)

Therefore, $CRB_M(n) = O(n)$, indicating that this language is in AI-C_2 (using two cognitive dimensions: pattern recognition and working memory).

1.2. Data Classification

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that performs multi-class classification on a set of data points.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$
- $\Sigma = \{\text{feature vectors}, \text{class labels}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$\}$ where \$ is the end-of-data marker
- q_0 is the initial state
- $F = \{q_5\}$
- $T = \{T_D, T_M, T_P\}$ where T_D is data memory, T_M is model memory, and T_P is prediction memory
- $\Psi = \{\psi_{FE}, \psi_{FM}, \psi_{PR}, \psi_{EV}\}$ where:
  - ψ_{FE}: Feature Extraction
  - ψ_{FM}: Feature Mapping
  - ψ_{PR}: Prediction
  - ψ_{EV}: Evaluation

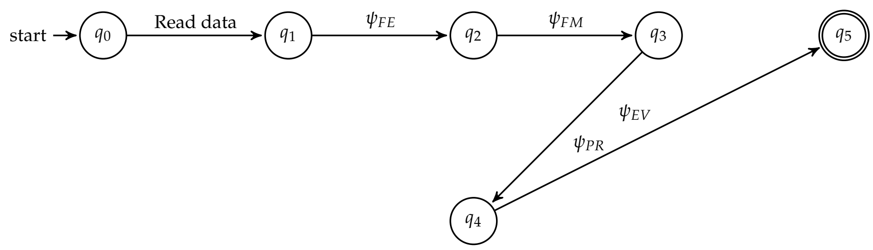

Figure 2. State diagram of the Classification CTM.

The Cognitive Resource Bound for this CTM classifying n data points with d features into k classes is:

$$CRB_M(n, d, k) = \max O(nd), O(ndk), O(k), O(nk)$$

Therefore, $CRB_M(n, d, k) = O(ndk)$, indicating that this classification task is in AI-C_4 (see Figure 2). This CTM works as follows:

1. It reads the input data points and their features into the data memory tape T_D.
2. The feature extraction operation ψ_{FE} processes the raw input features.
3. The feature mapping operation ψ_{FM} transforms the extracted features into a suitable representation for classification.
4. The prediction operation ψ_{PR} applies the classification model (stored in T_M) to the transformed features and generates class predictions.
5. Finally, the evaluation operation ψ_{EV} assesses the performance of the classifier by comparing predictions to true labels.

The components of the Cognitive Resource Bound represent:

- $C_M(n,d,k) = O(nd)$ (number of distinct configurations, based on input size)
- $O_M(n,d,k) = O(ndk)$ (number of cognitive operations, considering feature processing and classification)
- $I_M(n,d,k) = O(k)$ (interactions between tapes, primarily for model application)
- $E_M(n,d,k) = O(nk)$ (cognitive energy expenditure, mainly from prediction and evaluation)

This classification CTM demonstrates how complex machine learning tasks can be modeled within the framework of Cognitive Turing Machines, providing a theoretical foundation for analyzing the computational resources required for AI-based classification problems.

1.3. Data Clustering

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that performs k-means clustering on a set of data points.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$
- $\Sigma = \{\text{numerical data points}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$\}$ where \$ is the end-of-data marker
- q_0 is the initial state
- $F = \{q_5\}$
- $T = \{T_D, T_C, T_A\}$ where T_D is data memory, T_C is centroid memory, and T_A is assignment memory
- $\Psi = \{\psi_{PR}, \psi_{DM}, \psi_{CM}, \psi_{CO}\}$ where:
  - ψ_{PR}: Pattern Recognition
  - ψ_{DM}: Distance Measurement
  - ψ_{CM}: Centroid Movement
  - ψ_{CO}: Convergence Operation

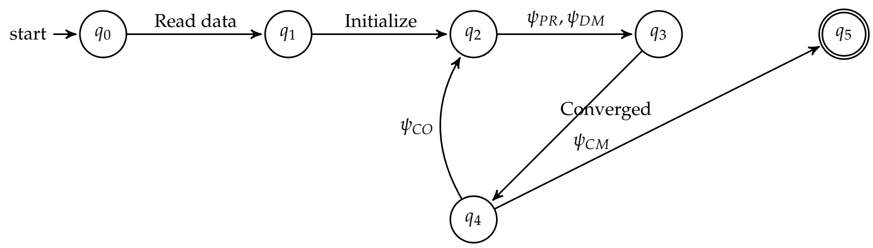

Figure 3. State diagram of the Clustering by using CTM.

The Cognitive Resource Bound for this CTM clustering n data points into k clusters with i iterations:

$$CRB_M(n,k,i) = \max\{O(nk), O(nki), O(k), O(n^2k)\}$$

Therefore, $CRB_M(n,k,i) = O(n^2k)$, indicating that this clustering task is in AI-C_4. Figure 3 shows the state diagram of clustering scenario.

1.4. Data Regression

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that performs linear regression on a set of data points.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$
- $\Sigma = \{\text{numerical data points (x,y)}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$\}$ where \$ is the end-of-data marker
- q_0 is the initial state
- $F = \{q_5\}$

- $T = \{T_D, T_P, T_E\}$ where T_D is data memory, T_P is parameter memory, and T_E is error memory
- $\Psi = \{\psi_{FP}, \psi_{BP}, \psi_{PE}, \psi_{CO}\}$ where:
  - ψ_{FP}: Forward Propagation
  - ψ_{BP}: Backward Propagation
  - ψ_{PE}: Parameter Estimation
  - ψ_{CO}: Convergence Operation

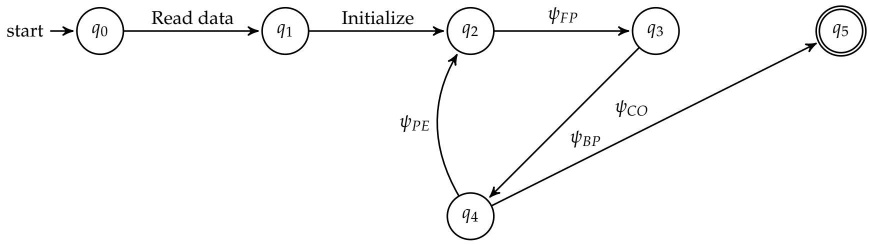

Figure 4. State diagram of the Regression by using CTM.

The Cognitive Resource Bound for this CTM performing regression on n data points with i iterations is:

$$CRB_M(n,i) = \max\{O(n), O(ni), O(i), O(ni)\}$$

Therefore, $CRB_M(n,i) = O(ni)$, indicating that this regression task is in AI-C_4 (see Figure 4).

1.5. Object Detection

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that performs object detection in images.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$
- $\Sigma = \{\text{pixel values}\} \cup \{\text{object classes}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{BBox}]\}$ where \$ is the end-of-data marker and [BBox] represents bounding box coordinates
- q_0 is the initial state
- $F = \{q_5\}$
- $T = \{T_I, T_F, T_A, T_D\}$ where:
  - T_I: Image memory
  - T_F: Feature map memory
  - T_A: Anchor box memory
  - T_D: Detection result memory
- $\Psi = \{\psi_{CV}, \psi_{FE}, \psi_{AP}, \psi_{BB}, \psi_{NMS}\}$ where:
  - ψ_{CV}: Computer Vision (convolutional processing)
  - ψ_{FE}: Feature Extraction
  - ψ_{AP}: Anchor Proposal
  - ψ_{BB}: Bounding Box Regression
  - ψ_{NMS}: Non-Maximum Suppression

The Cognitive Resource Bound for this CTM detecting objects (as shown Figure 5) in an image of size $n \times n$ with m potential objects is:

$$CRB_M(n,m) = \max\{O(n^2), O(n^2 \log n), O(m^2), O(n^2 m)\}$$

Therefore, $CRB_M(n,m) = O(n^2 m)$, indicating that this object detection task is in AI-C_5.

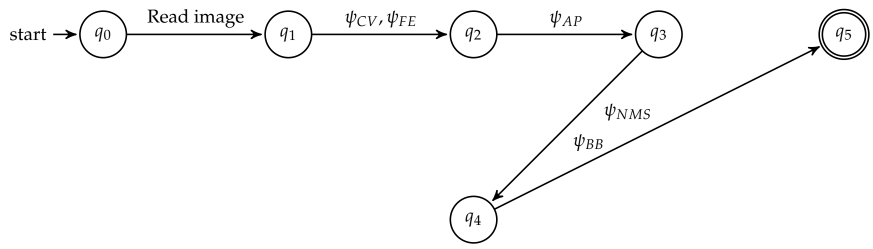

Figure 5. State diagram of the Object Detection by using CTM.

1.6. Image Segmentation

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that performs semantic image segmentation.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$
- $\Sigma = \{\text{pixel values}\} \cup \{\text{semantic labels}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{MASK}]\}$
- q_0 is the initial state
- $F = \{q_5\}$
- $T = \{T_I, T_F, T_E, T_M, T_S\}$ where:
  - T_I: Image memory
  - T_F: Feature map memory
  - T_E: Encoder memory
  - T_M: Mask memory
  - T_S: Segmentation result memory
- $\Psi = \{\psi_{CV}, \psi_{FE}, \psi_{EN}, \psi_{DE}, \psi_{CRF}\}$ where:
  - ψ_{CV}: Computer Vision (convolutional processing)
  - ψ_{FE}: Feature Extraction
  - ψ_{EN}: Encoder
  - ψ_{DE}: Decoder
  - ψ_{CRF}: Conditional Random Field

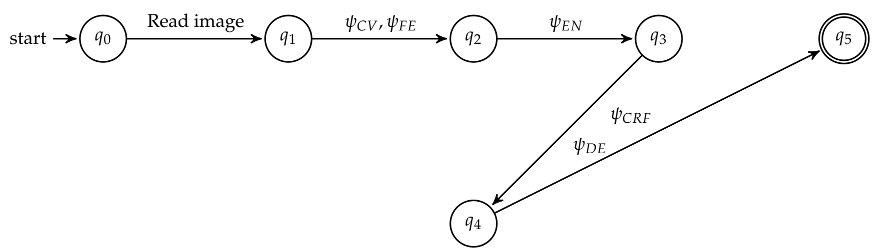

Figure 6. State diagram of the Image Segmentation CTM

The Cognitive Resource Bound (CRB) for this CTM segmenting an image of size $n \times n$ with k semantic classes is:

$$CRB_M(n, k) = \max\{O(n^2), O(n^2 \log n), O(n^2 k), O(n^4)\}$$

Therefore, $CRB_M(n, k) = O(n^4)$, indicating that this image segmentation task is in AI-C_5.

1.7. Automatic Speech Recognition

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that performs automatic speech recognition.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$
- $\Sigma = \{\text{audio waveform samples}\} \cup \{\text{phonemes}\} \cup \{\text{words}\}$

- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{SIL}]\}$ where \$ is the end-of-data marker and [SIL] represents silence
- q_0 is the initial state
- $F = \{q_6\}$
- $T = \{T_A, T_F, T_P, T_L, T_T\}$ where:
  - T_A: Audio memory
  - T_F: Feature memory
  - T_P: Phoneme memory
  - T_L: Language model memory
  - T_T: Transcription memory
- $\Psi = \{\psi_{SP}, \psi_{FE}, \psi_{AM}, \psi_{PD}, \psi_{LM}, \psi_{CTC}\}$ where:
  - ψ_{SP}: Signal Processing
  - ψ_{FE}: Feature Extraction (e.g., MFCC)
  - ψ_{AM}: Acoustic Modeling
  - ψ_{PD}: Phoneme Decoding
  - ψ_{LM}: Language Modeling
  - ψ_{CTC}: Connectionist Temporal Classification

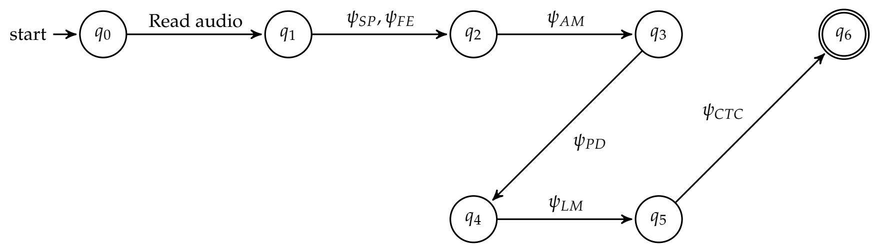

Figure 7. State diagram of the Automatic Speech Recognition CTM

The Cognitive Resource Bound (CRB) for this CTM processing an audio signal of length n with vocabulary size v is:

$$CRB_M(n, v) = \max\{O(n \log n), O(n^2), O(nv), O(n \log v)\}$$

Therefore, $CRB_M(n, v) = O(n^2)$, indicating that this automatic speech recognition task is in AI-C_6.

1.8. Text Recognition and Prediction

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that recognizes words from a sentence and predicts the next word.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$
- $\Sigma = \{\text{common English words}\}$
- $\Gamma = \Sigma \cup \{\sqcup\}$
- q_0 is the initial state
- $F = \{q_5\}$
- $T = \{T_W, T_L, T_P\}$ where T_W is working memory, T_L is long-term memory, and T_P is prediction buffer
- $\Psi = \{\psi_{NLU}, \psi_{PR}, \psi_L\}$ where:
  - ψ_{NLU}: Natural Language Understanding
  - ψ_{PR}: Pattern Recognition
  - ψ_L: Learning

This CTM works as follows:

1. It reads words from the input sentence (state q_0 to q_1).
2. For each word, it applies natural language understanding (ψ_{NLU}) to process its meaning (state q_1 to q_2).
3. It then uses pattern recognition (ψ_{PR}) to identify contextual patterns (state q_2 to q_3).

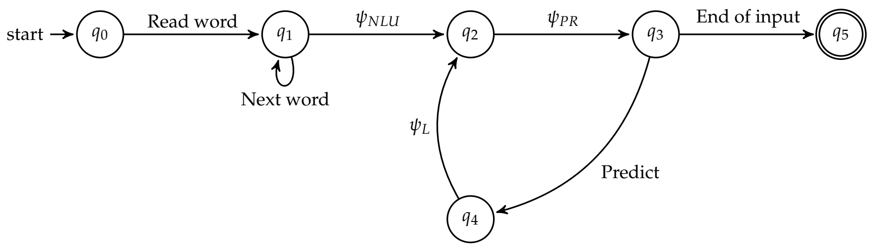

Figure 8. State diagram of the Text Recognition and Prediction CTM

4. Based on the patterns, it predicts the next word (state q_3 to q_4).
5. It learns from the prediction by comparing with the actual next word (ψ_L) and updates its model (state q_4 back to q_2).
6. The process repeats until the end of the input is reached (transition to accepting state q_5).

The Cognitive Resource Bound (CRB) for this CTM on an input of n words is:

$$CRB_M(n) = \max\{O(n), O(3n), O(n), O(n \log n)\}$$

Where the components represent:

- $C_M(n) = O(n)$ (number of distinct configurations)
- $O_M(n) = O(3n)$ (number of cognitive operations, 3 per word)
- $I_M(n) = O(n)$ (interactions between tapes)
- $E_M(n) = O(n \log n)$ (cognitive energy expenditure, assuming pattern matching complexity)

Therefore, $CRB_M(n) = O(n \log n)$, indicating that this language recognition and prediction task is in AI-C_3 (using three cognitive dimensions: natural language understanding, pattern recognition, and learning).

1.9. Text-to-Text Generation

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that generates text based on an initial prompt.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$
- $\Sigma = \{\text{common English words and punctuation}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$\}$ where \$ is the end-of-text marker
- q_0 is the initial state
- $F = \{q_5\}$
- $T = \{T_W, T_L, T_G\}$ where T_W is working memory, T_L is long-term memory, and T_G is generation buffer
- $\Psi = \{\psi_{NLU}, \psi_{NLG}, \psi_{CR}, \psi_{MC}\}$ where:
  - ψ_{NLU}: Natural Language Understanding
  - ψ_{NLG}: Natural Language Generation
  - ψ_{CR}: Contextual Reasoning
  - ψ_{MC}: Metacognition (for self-evaluation)

This CTM works as follows:

1. It reads the initial prompt (state q_0 to q_1).
2. It applies natural language understanding (ψ_{NLU}) to process the meaning of the prompt (state q_1 to q_2).
3. It uses contextual reasoning (ψ_{CR}) to determine the appropriate context for generation (state q_2 to q_3).
4. It then generates text using natural language generation (ψ_{NLG}) (state q_3 to q_4).
5. After generating each segment, it uses metacognition (ψ_{MC}) to evaluate the generated text (state q_4 back to q_2).

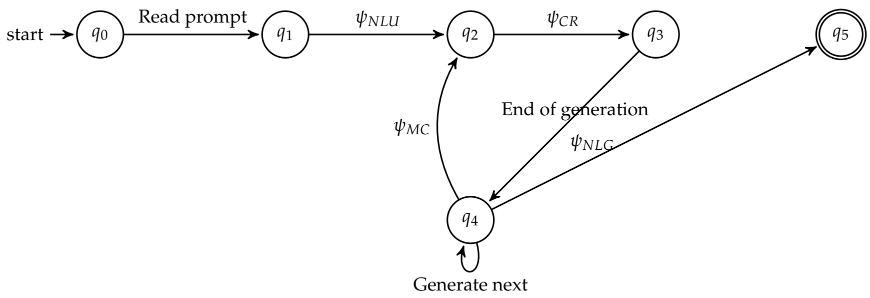

Figure 9. State diagram of the Text Generation CTM

6. The process repeats, generating more text, until a termination condition is met (transition to accepting state q_5).

The Cognitive Resource Bound (CRB) for this CTM generating n words is:

$$CRB_M(n) = \max\{O(n), O(4n), O(n), O(n^2)\}$$

Where the components represent:

- $C_M(n) = O(n)$ (number of distinct configurations)
- $O_M(n) = O(4n)$ (number of cognitive operations, 4 per word on average)
- $I_M(n) = O(n)$ (interactions between tapes)
- $E_M(n) = O(n^2)$ (cognitive energy expenditure, assuming quadratic complexity for contextual reasoning)

Therefore, $CRB_M(n) = O(n^2)$, indicating that this text generation task is in AI-C_4 (using four cognitive dimensions: natural language understanding, natural language generation, contextual reasoning, and metacognition).

1.10. Text-to-Image Generation

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that generates high-quality images from textual descriptions.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$
- $\Sigma = \{\text{text tokens}\} \cup \{\text{image tokens}\} \cup \{\text{style parameters}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, \text{[EMBED]}, \text{[LATENT]}, \text{[STYLE]}\}$
- q_0 is the initial state
- $F = \{q_8\}$
- $T = \{T_T, T_E, T_L, T_S, T_I, T_R, T_C\}$ where:
  - T_T: Text memory
  - T_E: Text embedding memory
  - T_L: Latent representation memory
  - T_S: Style memory
  - T_I: Image generation memory
  - T_R: Refinement memory
  - T_C: Consistency check memory
- $\Psi = \{\psi_{TE}, \psi_{LM}, \psi_{CA}, \psi_{SG}, \psi_{IG}, \psi_{UP}, \psi_{CC}\}$ where:
  - ψ_{TE}: Text Embedding
  - ψ_{LM}: Latent Mapping
  - ψ_{CA}: Cross-Attention
  - ψ_{SG}: Style Guidance
  - ψ_{IG}: Image Generation
  - ψ_{UP}: Upscaling and Refinement
  - ψ_{CC}: Consistency Check

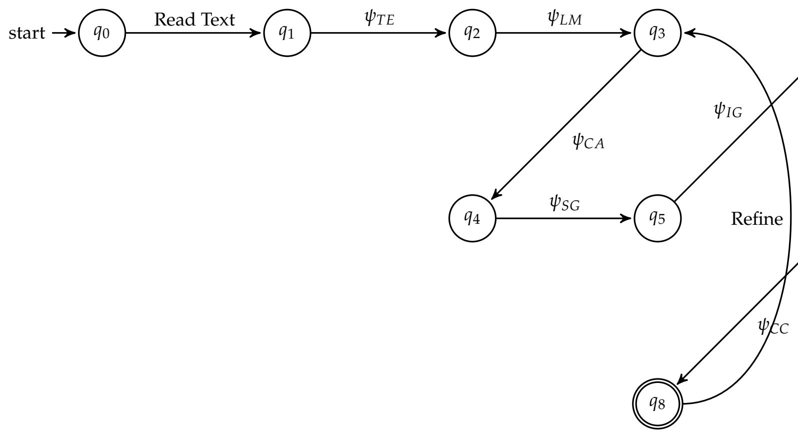

Figure 10. State diagram of the Text-to-Image Generation CTM

The Cognitive Resource Bound (CRB) for this CTM processing a text input of length n, generating an image of size $m \times m$, with latent space dimension d, and style parameter space size s is:

$$CRB_M(n, m, d, s) = \max\{O(n \log n), O(m^2 \log m), O(2^d), O(s!), O(nm^2 d), O(n^2 m^2)\}$$

Therefore, $CRB_M(n, m, d, s) = O(\max(2^d, s!, n^2 m^2))$, indicating that this text-to-image generation task is in AI-C$_7$.

1.11. Image-to-Text Generation Scenario

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that generates detailed textual descriptions from complex images using a transformer-like architecture.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$
- $\Sigma = \{\text{pixel values}\} \cup \{\text{common English words}\} \cup \{\text{special tokens}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{SEP}], [\text{CLS}]\}$ where \$ is the end-of-data marker
- q_0 is the initial state
- $F = \{q_6\}$
- $T = \{T_I, T_E, T_A, T_L, T_G\}$ where:
  - T_I: Image memory
  - T_E: Encoder memory
  - T_A: Attention memory
  - T_L: Language model memory
  - T_G: Generation buffer
- $\Psi = \{\psi_{CV}, \psi_{FE}, \psi_{SA}, \psi_{CA}, \psi_{FFN}, \psi_{NLG}, \psi_{MC}\}$ where:
  - ψ_{CV}: Computer Vision (convolutional processing)
  - ψ_{FE}: Feature Extraction
  - ψ_{SA}: Self-Attention
  - ψ_{CA}: Cross-Attention
  - ψ_{FFN}: Feed-Forward Network
  - ψ_{NLG}: Natural Language Generation
  - ψ_{MC}: Metacognition (for self-evaluation and refinement)

This CTM works as follows:

1. Read the input image into T_I (state q_0 to q_1).
2. Apply computer vision and feature extraction operations (ψ_{CV}, ψ_{FE}) to process the image (state q_1 to q_2).
3. Perform self-attention (ψ_{SA}) on the extracted features (state q_2 to q_3).

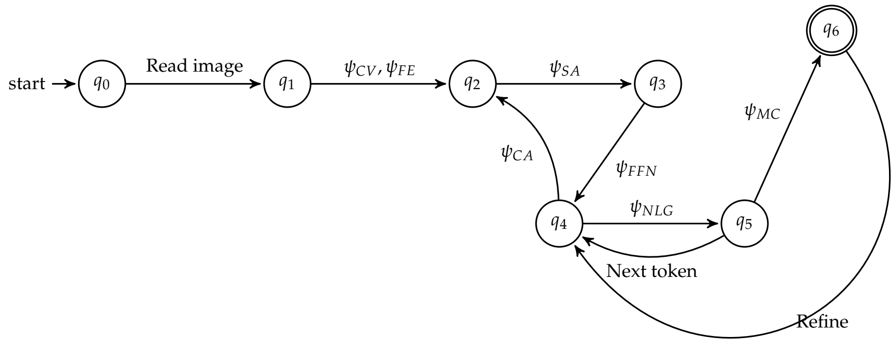

Figure 11. State diagram of the Advanced Image-to-Text Generation CTM

4. Apply a feed-forward network (ψ_{FFN}) to further process the attended features (state q_3 to q_4).
5. Use cross-attention (ψ_{CA}) to relate the processed features to the current text generation state (loop from q_4 to q_2).
6. Generate the next token of the description using ψ_{NLG} (state q_4 to q_5).
7. Repeat steps 5-6 until the description is complete.
8. Apply metacognition (ψ_{MC}) to evaluate and potentially refine the generated description (state q_5 to q_6, with possible loop back to q_4).

The Cognitive Resource Bound (CRB) for this CTM generating a description of m tokens from an image of size $n \times n$ with d attention heads and l encoder/decoder layers is:

$$CRB_M(n, m, d, l) = \max\{O(n^2), O(n^2dl), O(m^2dl), O(nmdl), O(nm \log m)\}$$

Where the components represent:

- $O(n^2)$: Image processing and feature extraction
- $O(n^2dl)$: Self-attention on image features
- $O(m^2dl)$: Self-attention on generated text
- $O(nmdl)$: Cross-attention between image and text
- $O(nm \log m)$: Metacognitive evaluation and refinement

Therefore, $CRB_M(n, m, d, l) = O(\max(n^2dl, m^2dl, nm \log m))$, indicating that this advanced image-to-text generation task is in AI-C_7, using seven distinct cognitive operations.

This CTM demonstrates several advanced concepts:

- Multi-head attention mechanisms, similar to transformer architectures
- Iterative refinement through metacognition
- Separation of encoding (image processing) and decoding (text generation) phases
- Complex interaction between visual and linguistic features

The high AI-C classification reflects the sophisticated nature of this task, which combines multiple AI domains including computer vision, natural language processing, and meta-learning.

1.12. Image-to-Image Generation

Consider a CTM $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that transforms input images into output images with different styles or characteristics.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$
- $\Sigma = \{\text{input image pixels}\} \cup \{\text{style parameters}\} \cup \{\text{output image pixels}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{FEATURE}], [\text{STYLE}], [\text{GEN}]\}$

- q_0 is the initial state
- $F = \{q_7\}$
- $T = \{T_I, T_F, T_S, T_L, T_G, T_R\}$ where:
  - T_I: Input image memory
  - T_F: Feature extraction memory
  - T_S: Style encoding memory
  - T_L: Latent representation memory
  - T_G: Generation memory
  - T_R: Refinement memory
- $\Psi = \{\psi_{FE}, \psi_{SE}, \psi_{FM}, \psi_{ST}, \psi_{IG}, \psi_{UP}, \psi_{QC}\}$ where:
  - ψ_{FE}: Feature Extraction
  - ψ_{SE}: Style Encoding
  - ψ_{FM}: Feature Manipulation
  - ψ_{ST}: Style Transfer
  - ψ_{IG}: Image Generation
  - ψ_{UP}: Upscaling
  - ψ_{QC}: Quality Check

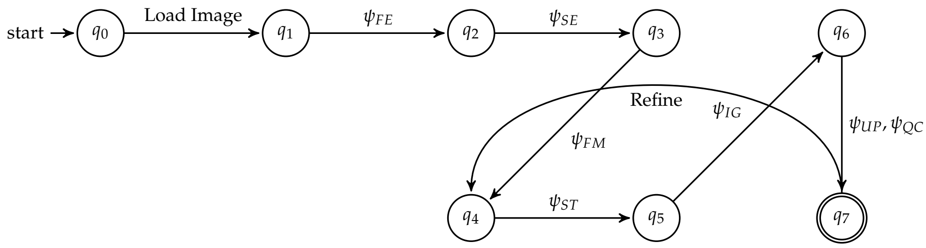

Figure 12. State diagram of the Image-to-Image Generation CTM

The Cognitive Resource Bound (CRB) for this CTM processing an input image of size $n \times n$, style parameter space of size s, and latent space dimension d is:

$$CRB_M(n, s, d) = \max\{O(n^2 \log n), O(s \log s), O(2^d), O(n^2 d), O(n^4)\}$$

Therefore, $CRB_M(n, s, d) = O(\max(2^d, n^4))$, indicating that this image-to-image generation task is in AI-C_7.

1.13. Text-to-Video Generation Scenario

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that generates video from textual descriptions.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$
- $\Sigma = \{\text{words}\} \cup \{\text{pixel values}\} \cup \{\text{motion vectors}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{FRAME}], [\text{SCENE}]\}$
- q_0 is the initial state
- $F = \{q_7\}$
- $T = \{T_T, T_S, T_F, T_M, T_V\}$ where:
  - T_T: Text memory
  - T_S: Scene description memory
  - T_F: Frame generation memory
  - T_M: Motion planning memory
  - T_V: Video composition memory
- $\Psi = \{\psi_{NLU}, \psi_{SD}, \psi_{IG}, \psi_{MP}, \psi_{TI}, \psi_{VC}, \psi_{GAN}\}$ where:
  - ψ_{NLU}: Natural Language Understanding
  - ψ_{SD}: Scene Decomposition
  - ψ_{IG}: Image Generation

– ψ_{MP}: Motion Planning
– ψ_{TI}: Temporal Interpolation
– ψ_{VC}: Video Composition
– ψ_{GAN}: Generative Adversarial Network

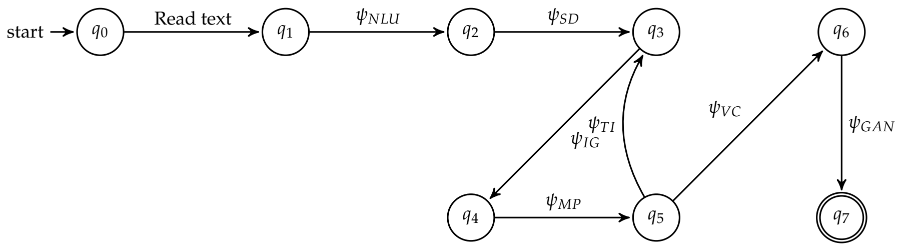

Figure 13. State diagram of the Text-to-Video Generation CTM

The CRB for this CTM generating a video of f frames from a text of length m is:

$$CRB_M(m, f) = \max\{O(m^2), O(f^2), O(mf^2), O(f^3)\}$$

Therefore, $CRB_M(m, f) = O(f^3)$, indicating that this text-to-video generation task is in AI-C_7.

1.14. Video-to-Text Generation

Now, let's consider a CTM $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that generates textual descriptions from video input.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$
- $\Sigma = \{\text{video frames}\} \cup \{\text{audio samples}\} \cup \{\text{text tokens}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, \text{[FRAME]}, \text{[AUDIO]}, \text{[EVENT]}, \text{[CAPTION]}\}$
- q_0 is the initial state
- $F = \{q_9\}$
- $T = \{T_V, T_A, T_F, T_E, T_C, T_L, T_T, T_S\}$ where:
  – T_V: Video frame memory
  – T_A: Audio memory
  – T_F: Feature extraction memory
  – T_E: Event detection memory
  – T_C: Context memory
  – T_L: Language model memory
  – T_T: Text generation memory
  – T_S: Summary memory
- $\Psi = \{\psi_{VF}, \psi_{AF}, \psi_{MM}, \psi_{ED}, \psi_{CA}, \psi_{LM}, \psi_{TG}, \psi_{SC}, \psi_{SU}\}$ where:
  – ψ_{VF}: Visual Feature Extraction
  – ψ_{AF}: Audio Feature Extraction
  – ψ_{MM}: Multimodal Fusion
  – ψ_{ED}: Event Detection
  – ψ_{CA}: Context Aggregation
  – ψ_{LM}: Language Modeling
  – ψ_{TG}: Text Generation
  – ψ_{SC}: Semantic Consistency Check
  – ψ_{SU}: Summary Generation

The Cognitive Resource Bound (CRB) for this CTM processing a video of f frames, each of size $n \times n$, with audio length a, generating a text of length t, and event space size e is:

$$CRB_M(f, n, a, t, e) = \max\{O(fn^2 \log n), O(a \log a), O(f \log f), O(e!), O(t \log t), O(fn^2at), O(2^e)\}$$

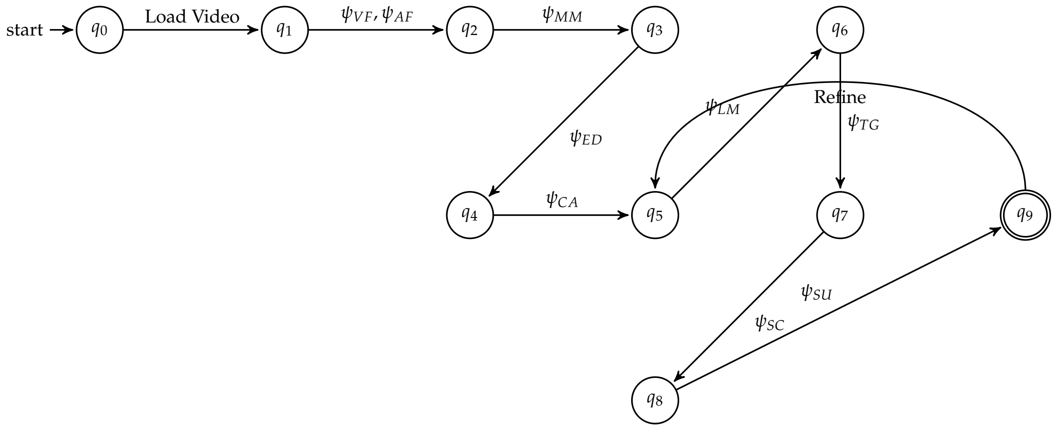

Figure 14. State diagram of the Video-to-Text Generation CTM

Therefore, $CRB_M(f,n,a,t,e) = O(\max(e!, 2^e, fn^2at))$, indicating that this video-to-text generation task is in AI-C$_9$.

These CTMs for text-to-image and video-to-text generation demonstrate the complexity of multimodal AI tasks. The high AI-C classifications reflect the computational demands of processing and generating content across different modalities.

In the text-to-image CTM, the exponential term $O(2^d)$ represents the complexity of the latent space, while $O(s!)$ captures the potential combinations of style parameters. The $O(n^2m^2)$ term reflects the interaction between text and image elements.

For the video-to-text CTM, $O(e!)$ and $O(2^e)$ represent the complexity of event detection and interpretation, while $O(fn^2at)$ captures the interactions between video frames, audio, and generated text.

These models provide a theoretical framework for understanding the computational requirements of advanced generative AI systems, highlighting the challenges in processing and synthesizing information across multiple modalities.

1.15. Video-to-Video Generation

Consider a CTM $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that transforms input videos into output videos with different styles or characteristics.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$
- $\Sigma = \{\text{input video frames}\} \cup \{\text{style parameters}\} \cup \{\text{output video frames}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{FRAME}], [\text{MOTION}], [\text{STYLE}], [\text{GEN}]\}$
- q_0 is the initial state
- $F = \{q_8\}$
- $T = \{T_V, T_F, T_M, T_S, T_L, T_G, T_T\}$ where:
  - T_V: Video frame memory
  - T_F: Feature extraction memory
  - T_M: Motion estimation memory
  - T_S: Style encoding memory
  - T_L: Latent representation memory
  - T_G: Generation memory
  - T_T: Temporal consistency memory
- $\Psi = \{\psi_{FE}, \psi_{ME}, \psi_{SE}, \psi_{FM}, \psi_{ST}, \psi_{VG}, \psi_{TC}, \psi_{QC}\}$ where:
  - ψ_{FE}: Feature Extraction
  - ψ_{ME}: Motion Estimation
  - ψ_{SE}: Style Encoding
  - ψ_{FM}: Feature Manipulation
  - ψ_{ST}: Style Transfer
  - ψ_{VG}: Video Generation

- ψ_{TC}: Temporal Consistency
- ψ_{QC}: Quality Check

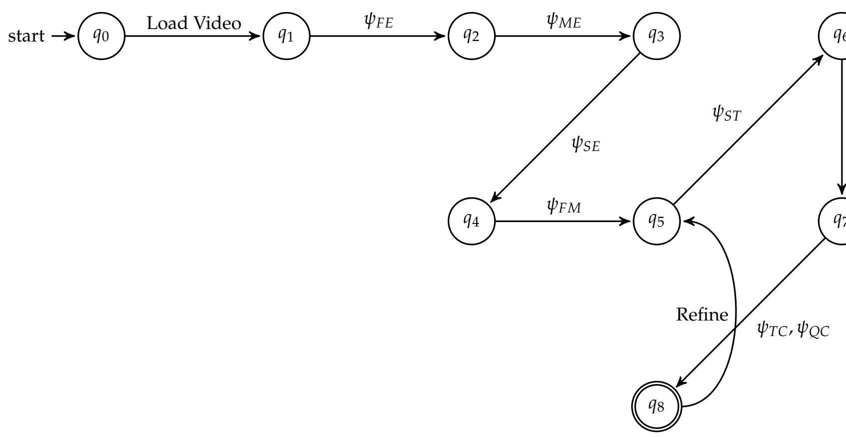

Figure 15. State diagram of the Video-to-Video Generation CTM

The Cognitive Resource Bound (CRB) for this CTM processing a video with f frames, each of size $n \times n$, style parameter space of size s, and latent space dimension d is:

$$CRB_M(f, n, s, d) = \max\{O(fn^2 \log n), O(f^2), O(s \log s), O(2^d), O(fn^2 d), O(f^2 n^4)\}$$

Therefore, $CRB_M(f, n, s, d) = O(\max(2^d, f^2 n^4))$, indicating that this video-to-video generation task is in AI-C_8.

1.16. Video Classification

Now, let's consider a CTM $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that classifies video inputs into predefined categories.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$
- $\Sigma = \{\text{video frames}\} \cup \{\text{audio samples}\} \cup \{\text{spatial features}\} \cup \{\text{temporal features}\} \cup \{\text{class labels}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, \text{[FRAME]}, \text{[AUDIO]}, \text{[SPAT]}, \text{[TEMP]}, \text{[FUSION]}\}$
- q_0 is the initial state
- $F = \{q_8\}$
- $T = \{T_V, T_A, T_S, T_T, T_F, T_M, T_C, T_P\}$ where:
  - T_V: Video frame memory
  - T_A: Audio memory
  - T_S: Spatial feature memory
  - T_T: Temporal feature memory
  - T_F: Fusion memory
  - T_M: Model memory
  - T_C: Classification memory
  - T_P: Probability distribution memory
- $\Psi = \{\psi_{PP}, \psi_{SF}, \psi_{TF}, \psi_{AF}, \psi_{MF}, \psi_{ML}, \psi_{CL}, \psi_{PR}\}$ where:
  - ψ_{PP}: Preprocessing
  - ψ_{SF}: Spatial Feature Extraction
  - ψ_{TF}: Temporal Feature Extraction
  - ψ_{AF}: Audio Feature Extraction
  - ψ_{MF}: Multimodal Fusion
  - ψ_{ML}: Model Learning
  - ψ_{CL}: Classification
  - ψ_{PR}: Probability Estimation

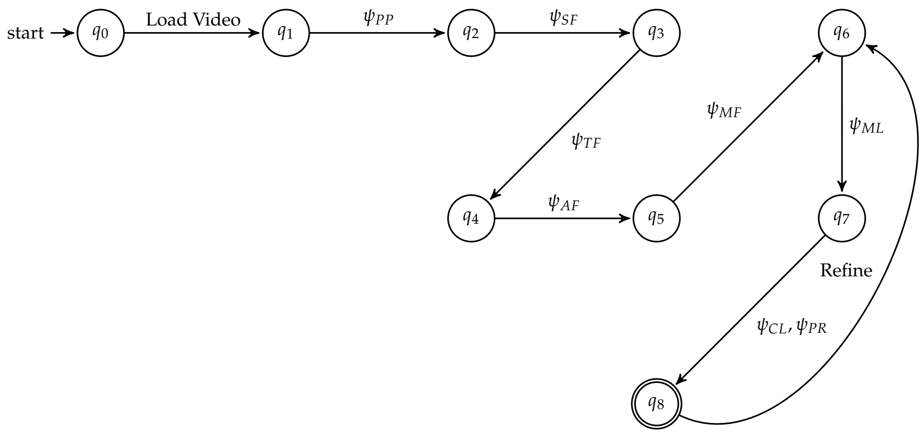

Figure 16. State diagram of the Video Classification CTM

The Cognitive Resource Bound (CRB) for this CTM processing a video with f frames, each of size $n \times n$, audio length a, spatial feature dimension d_s, temporal feature dimension d_t, number of classes k, and model complexity m is:

$$CRB_M(f, n, a, d_s, d_t, k, m) = \max\{O(fn^2 \log n), O(a \log a), O(fd_s), O(f^2 d_t), O(k \log k), O(m \log m),$$

Therefore, $CRB_M(f, n, a, d_s, d_t, k, m) = O(\max(2^m, fn^2 a d_s d_t k))$, indicating that this video classification task is in AI-C_8.

These CTM models for audio classification and video classification demonstrate the complexity of these multimodal AI tasks. The high AI-C classifications reflect the computational demands of processing and analyzing complex, multi-dimensional data.

In both models, the exponential term $O(2^m)$ represents the potential complexity of the classification model. This term dominates when the model is highly complex, which is often the case for deep learning models used in audio and video classification.

For the audio classification CTM, the term $O(ndk)$ captures the interaction between the audio length, feature dimension, and number of classes. This represents the core computational challenge in processing and classifying audio data.

In the video classification CTM, the term $O(fn^2 a d_s d_t k)$ reflects the multifaceted nature of video data, incorporating spatial (frame content), temporal (across frames), and audio features. This term highlights the significant computational requirements for processing and fusing multiple data modalities in video classification tasks.

1.17. Text-to-Audio Generation CTM

Consider a CTM $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that generates audio (speech or music) from textual input.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$
- $\Sigma = \{\text{text tokens}\} \cup \{\text{phonemes}\} \cup \{\text{prosody markers}\} \cup \{\text{audio samples}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, \text{[PHON]}, \text{[PROS]}, \text{[MEL]}, \text{[WAV]}\}$
- q_0 is the initial state
- $F = \{q_8\}$
- $T = \{T_T, T_P, T_R, T_M, T_V, T_A, T_Q\}$ where:
  - T_T: Text memory
  - T_P: Phoneme memory
  - T_R: Prosody memory
  - T_M: Mel-spectrogram memory
  - T_V: Voice characteristics memory
  - T_A: Audio generation memory

- T_Q: Quality control memory
- $\Psi = \{\psi_{TP}, \psi_{PA}, \psi_{PM}, \psi_{MS}, \psi_{VC}, \psi_{AG}, \psi_{PP}, \psi_{QC}\}$ where:
  - ψ_{TP}: Text-to-Phoneme Conversion
  - ψ_{PA}: Prosody Analysis
  - ψ_{PM}: Prosody Modeling
  - ψ_{MS}: Mel-Spectrogram Generation
  - ψ_{VC}: Voice Characteristic Modeling
  - ψ_{AG}: Audio Generation
  - ψ_{PP}: Post-processing
  - ψ_{QC}: Quality Control

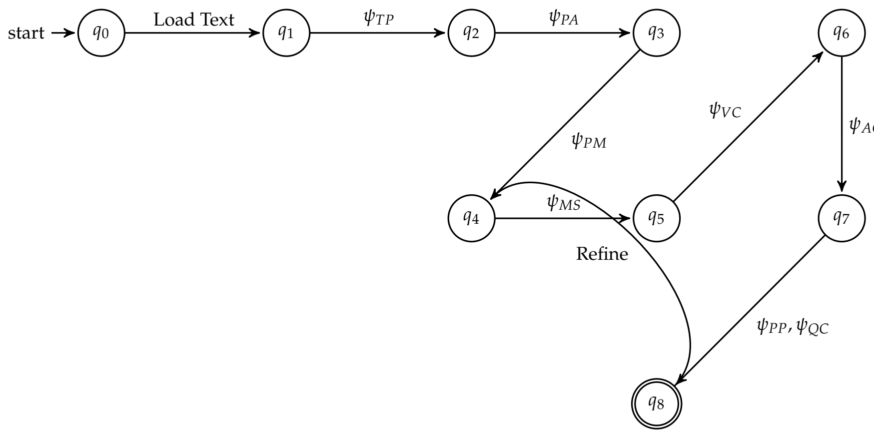

Figure 17. State diagram of the Text-to-Audio Generation CTM

The Cognitive Resource Bound (CRB) for this CTM processing a text of length n, with phoneme vocabulary size p, prosody feature dimension r, mel-spectrogram dimension m, and voice characteristic dimension v is:

$$CRB_M(n, p, r, m, v) = \max\{O(n \log n), O(np), O(nr), O(nm), O(2^v), O(n^2), O(npmrv)\}$$

Therefore, $CRB_M(n, p, r, m, v) = O(\max(2^v, n^2, npmrv))$, indicating that this text-to-audio generation task is in AI-C$_7$.

1.18. Audio-to-Audio Generation CTM

Now, let's consider a CTM $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that transforms input audio into output audio with different characteristics (e.g., voice conversion, style transfer).

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$
- $\Sigma = \{\text{input audio samples}\} \cup \{\text{spectral features}\} \cup \{\text{style parameters}\} \cup \{\text{output audio samp}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{SPEC}], [\text{FEAT}], [\text{STYLE}], [\text{GEN}]\}$
- q_0 is the initial state
- $F = \{q_9\}$
- $T = \{T_I, T_S, T_F, T_E, T_L, T_G, T_O, T_Q\}$ where:
  - T_I: Input audio memory
  - T_S: Spectrogram memory
  - T_F: Feature extraction memory
  - T_E: Style encoding memory
  - T_L: Latent representation memory
  - T_G: Generation memory
  - T_O: Output audio memory
  - T_Q: Quality control memory
- $\Psi = \{\psi_{PP}, \psi_{ST}, \psi_{FE}, \psi_{SE}, \psi_{LM}, \psi_{AG}, \psi_{IS}, \psi_{PO}, \psi_{QC}\}$ where:

- ψ_{PP}: Preprocessing
- ψ_{ST}: Spectrogram Transformation
- ψ_{FE}: Feature Extraction
- ψ_{SE}: Style Encoding
- ψ_{LM}: Latent Manipulation
- ψ_{AG}: Audio Generation
- ψ_{IS}: Inverse Spectrogram
- ψ_{PO}: Post-processing
- ψ_{QC}: Quality Control

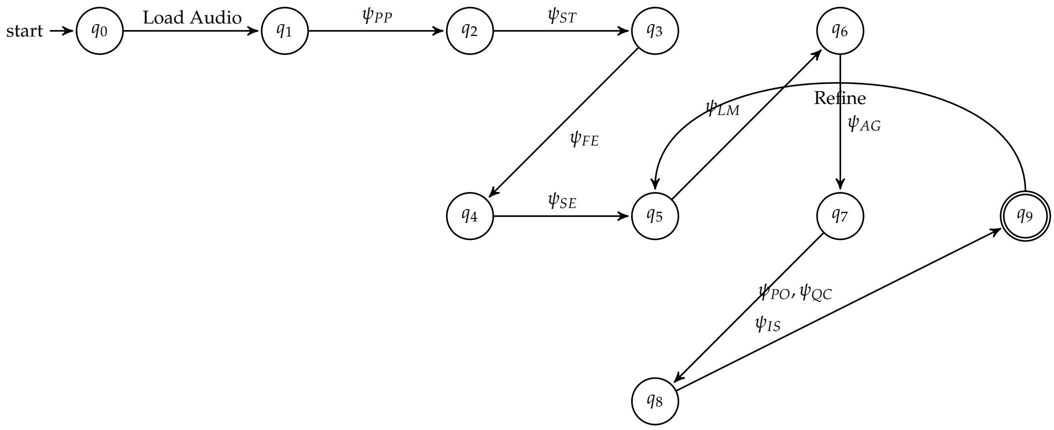

Figure 18. State diagram of the Audio-to-Audio Generation CTM

The Cognitive Resource Bound (CRB) for this CTM processing an input audio of length n, with spectral feature dimension d, style parameter space size s, latent space dimension l, and generation complexity g is:

$$CRB_M(n,d,s,l,g) = \max\{O(n\log n), O(nd), O(s\log s), O(2^l), O(g\log g), O(n^2), O(ndslg)\}$$

Therefore, $CRB_M(n,d,s,l,g) = O(\max(2^l, n^2, ndslg))$, indicating that this audio-to-audio generation task is in AI-C_8.

These CTM models for text-to-audio and audio-to-audio generation demonstrate the complexity of these advanced AI tasks in audio processing and generation. The high AI-C classifications reflect the computational demands of processing and manipulating complex audio data.

In the text-to-audio generation CTM:

- The term $O(2^v)$ represents the potential complexity of modeling voice characteristics.
- $O(n^2)$ captures potential quadratic complexity in sequence modeling for longer texts.
- $O(npmrv)$ reflects the interaction between text length, phoneme vocabulary, prosody features, mel-spectrogram dimensions, and voice characteristics.

For the audio-to-audio generation CTM:

- $O(2^l)$ represents the complexity of the latent space, which is crucial for style transfer and voice conversion tasks.
- $O(n^2)$ captures potential quadratic complexity in processing longer audio sequences.
- $O(ndslg)$ reflects the interaction between audio length, spectral features, style parameters, latent space, and generation complexity.

1.19. Audio Classification

Consider a CTM $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that classifies audio inputs into predefined categories.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$

- $\Sigma = \{\text{audio samples}\} \cup \{\text{frequency features}\} \cup \{\text{class labels}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{SPEC}], [\text{MFCC}], [\text{FEAT}]\}$
- q_0 is the initial state
- $F = \{q_7\}$
- $T = \{T_A, T_S, T_F, T_M, T_C, T_P\}$ where:
  - T_A: Audio sample memory
  - T_S: Spectrogram memory
  - T_F: Feature memory
  - T_M: Model memory
  - T_C: Classification memory
  - T_P: Probability distribution memory
- $\Psi = \{\psi_{PP}, \psi_{ST}, \psi_{FE}, \psi_{TF}, \psi_{ML}, \psi_{CL}, \psi_{PR}\}$ where:
  - ψ_{PP}: Preprocessing
  - ψ_{ST}: Spectrogram Transformation
  - ψ_{FE}: Feature Extraction
  - ψ_{TF}: Temporal Fusion
  - ψ_{ML}: Model Learning
  - ψ_{CL}: Classification
  - ψ_{PR}: Probability Estimation

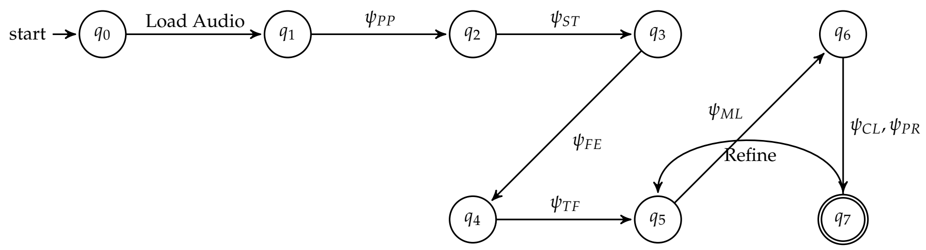

Figure 19. State diagram of the Audio Classification CTM

The Cognitive Resource Bound (CRB) for this CTM processing an audio signal of length n, with feature dimension d, number of classes k, and model complexity m is:

$$CRB_M(n, d, k, m) = \max\{O(n \log n), O(n \log d), O(d^2), O(m \log m), O(k \log k), O(ndk), O(2^m)\}$$

Therefore, $CRB_M(n, d, k, m) = O(\max(2^m, ndk))$, indicating that this audio classification task is in AI-C$_7$.

1.20. Multimodal Generative AI

Consider an enhanced Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that generates coherent content across multiple modalities (text, image, audio, video, tactile) based on a given multimodal prompt.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}\}$
- $\Sigma = \{\text{text, image data, audio data, video data, tactile data, semantic concepts}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{MODAL}], [\text{GEN}], [\text{ALIGN}], [\text{FUSION}], [\text{REFINE}]\}$
- q_0 is the initial state
- $F = \{q_{10}\}$
- $T = \{T_P, T_E, T_C, T_G, T_I, T_A, T_V, T_T, T_S, T_F, T_R\}$ where:
  - T_P: Multimodal prompt memory
  - T_E: Unified embedding memory
  - T_C: Cross-modal context memory
  - T_G: Text generation memory
  - T_I: Image generation memory

  - T_A: Audio generation memory
  - T_V: Video generation memory
  - T_T: Tactile signal generation memory
  - T_S: Semantic consistency memory
  - T_F: Multimodal fusion memory
  - T_R: Refinement and coherence memory
- $\Psi = \{\psi_{NLU}, \psi_{UE}, \psi_{CA}, \psi_{TG}, \psi_{IG}, \psi_{AG}, \psi_{VG}, \psi_{TSG}, \psi_{SC}, \psi_{MF}, \psi_{CO}, \psi_{MA}, \psi_{GR}\}$ where:
  - ψ_{NLU}: Natural Language Understanding
  - ψ_{UE}: Unified Multimodal Embedding
  - ψ_{CA}: Cross-modal Attention
  - ψ_{TG}: Text Generation
  - ψ_{IG}: Image Generation
  - ψ_{AG}: Audio Generation
  - ψ_{VG}: Video Generation
  - ψ_{TSG}: Tactile Signal Generation
  - ψ_{SC}: Semantic Consistency Check
  - ψ_{MF}: Multimodal Fusion
  - ψ_{CO}: Content Orchestration
  - ψ_{MA}: Multimodal Alignment
  - ψ_{GR}: Global Refinement

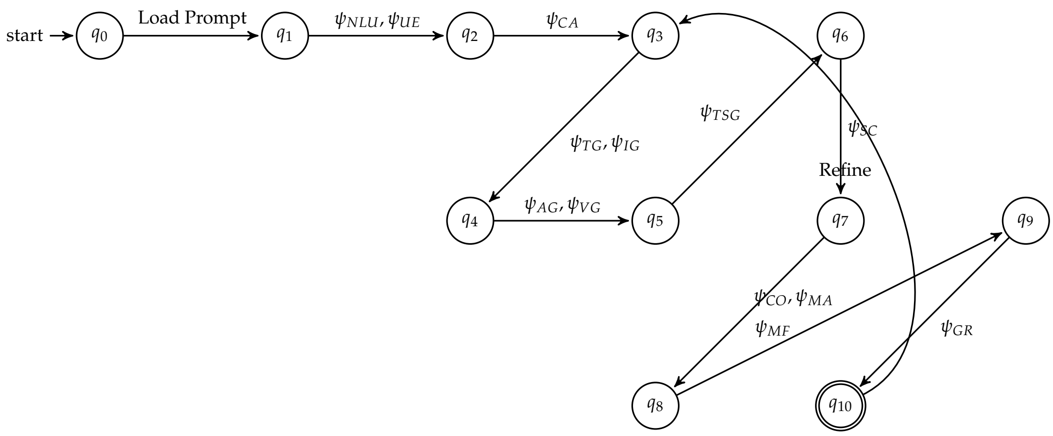

Figure 20. State diagram of the Advanced Multimodal Generative AI CTM

The enhanced Cognitive Resource Bound (CRB) for this CTM processing a multimodal prompt of complexity p, generating content with complexity n across m modalities, with semantic concept space s, fusion complexity f, and refinement iterations r is:

$$CRB_M(p,n,m,s,f,r) = \max\{O(p^2m), O(n^2m^2), O(2^m), O(s\log s), O(f^m), O(r(nm)^2), O(pnm^2sfr)\}$$

Therefore, $CRB_M(p,n,m,s,f,r) = O(\max(2^m, f^m, r(nm)^2, pnm^2sfr))$, indicating that this advanced multimodal generative AI task is in AI-C_{10}.

This enhanced model incorporates several advanced aspects of multimodal AI:

1. Unified Multimodal Embedding (ψ_{UE}): Creates a shared representation space for all modalities, allowing for better cross-modal understanding and generation.

2. Semantic Consistency Check (ψ_{SC}): Ensures that generated content across different modalities is semantically consistent.

3. Multimodal Fusion (ψ_{MF}): Combines information from different modalities to create a coherent multimodal output.

4. Multimodal Alignment (ψ_{MA}): Aligns generated content across modalities to ensure temporal and spatial coherence.

5. Global Refinement (ψ_{GR}): Performs iterative refinement to improve overall coherence and quality of the multimodal output.

6. Tactile Signal Generation (ψ_{TSG}): Extends the model to include tactile feedback, broadening the scope of multimodal interaction.

The CRB now includes additional terms: - $O(p^2m)$ represents the complexity of processing the multimodal prompt - $O(s \log s)$ captures the complexity of managing the semantic concept space - $O(f^m)$ represents the complexity of multimodal fusion across m modalities - $O(r(nm)^2)$ accounts for the iterative refinement process - $O(pnm^2sfr)$ captures the overall interaction between all components of the system

The higher AI-C classification (AI-C_{10}) reflects the increased complexity and sophistication of this advanced multimodal generative AI system. This model provides a more comprehensive framework for understanding and analyzing the computational requirements of cutting-edge multimodal AI systems, highlighting the challenges in creating coherent, semantically consistent content across multiple modalities.

1.21. Autonomous Robot Driving Scenario

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that controls an autonomous driving robot.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$
- $\Sigma = \{\text{sensor data}\} \cup \{\text{map data}\} \cup \{\text{traffic rules}\} \cup \{\text{control commands}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, \text{[OBSTACLE]}, \text{[ROUTE]}\}$
- q_0 is the initial state
- $F = \{q_7\}$
- $T = \{T_S, T_M, T_P, T_D, T_C\}$ where:
  - T_S: Sensor data memory
  - T_M: Map and navigation memory
  - T_P: Perception memory
  - T_D: Decision-making memory
  - T_C: Control command memory
- $\Psi = \{\psi_{SP}, \psi_{OD}, \psi_{LO}, \psi_{PM}, \psi_{RP}, \psi_{DM}, \psi_{MPC}\}$ where:
  - ψ_{SP}: Sensor Processing
  - ψ_{OD}: Object Detection
  - ψ_{LO}: Localization
  - ψ_{PM}: Path Mapping
  - ψ_{RP}: Route Planning
  - ψ_{DM}: Decision Making
  - ψ_{MPC}: Model Predictive Control

The CRB for this CTM processing sensor data of size n and map data of size m for a route of length l is:

$$CRB_M(n, m, l) = \max\{O(n^2), O(m \log m), O(l^2), O(nml)\}$$

Therefore, $CRB_M(n, m, l) = O(nml)$, indicating that this autonomous driving task is in AI-C_7.

1.22. Autonomous Agents

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that models an autonomous agent in a complex environment.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$
- $\Sigma = \{\text{environmental states}\} \cup \{\text{agent actions}\} \cup \{\text{reward signals}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, \text{[GOAL]}, \text{[PLAN]}\}$

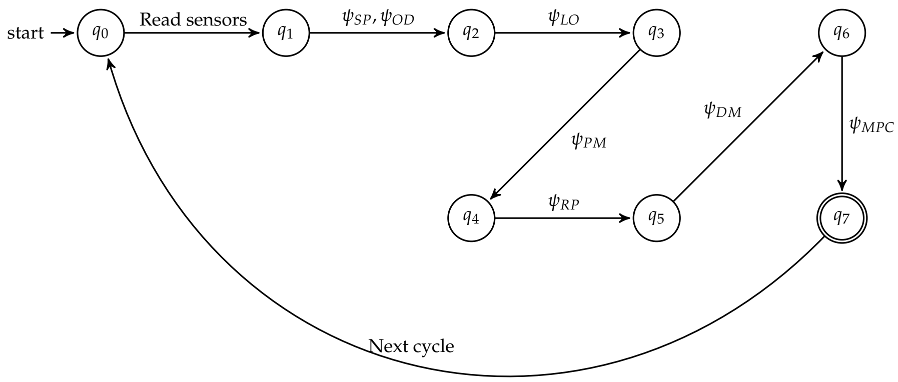

Figure 21. State diagram of the Autonomous Robot Driving by using CTM.

- q_0 is the initial state
- $F = \{q_8\}$
- $T = \{T_E, T_K, T_G, T_P, T_A, T_M\}$ where:
  - T_E: Environment perception memory
  - T_K: Knowledge base memory
  - T_G: Goal memory
  - T_P: Planning memory
  - T_A: Action memory
  - T_M: Meta-cognitive memory
- $\Psi = \{\psi_{EP}, \psi_{KR}, \psi_{GF}, \psi_{PP}, \psi_{DM}, \psi_{RL}, \psi_{MC}, \psi_{CO}\}$ where:
  - ψ_{EP}: Environment Perception
  - ψ_{KR}: Knowledge Representation
  - ψ_{GF}: Goal Formulation
  - ψ_{PP}: Path Planning
  - ψ_{DM}: Decision Making
  - ψ_{RL}: Reinforcement Learning
  - ψ_{MC}: Metacognition
  - ψ_{CO}: Communication and Coordination

The Cognitive Resource Bound (CRB) for this CTM processing environmental data of size n, with knowledge base size k, and planning horizon h is:

$$CRB_M(n,k,h) = \max\{O(n^2), O(k \log k), O(h^2), O(nkh), O(2^h)\}$$

Therefore, $CRB_M(n,k,h) = O(\max(nkh, 2^h))$, indicating that this autonomous agent task is in AI-C_8.

1.23. Federated Learning

Consider a CTM that implements federated learning across multiple decentralized edge devices.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$
- $\Sigma = \{\text{local data, model parameters, encrypted updates}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{LOCAL}], [\text{GLOBAL}]\}$
- q_0 is the initial state
- $F = \{q_6\}$
- $T = \{T_L, T_G, T_E, T_A, T_U\}$ where:
  - T_L: Local data memory

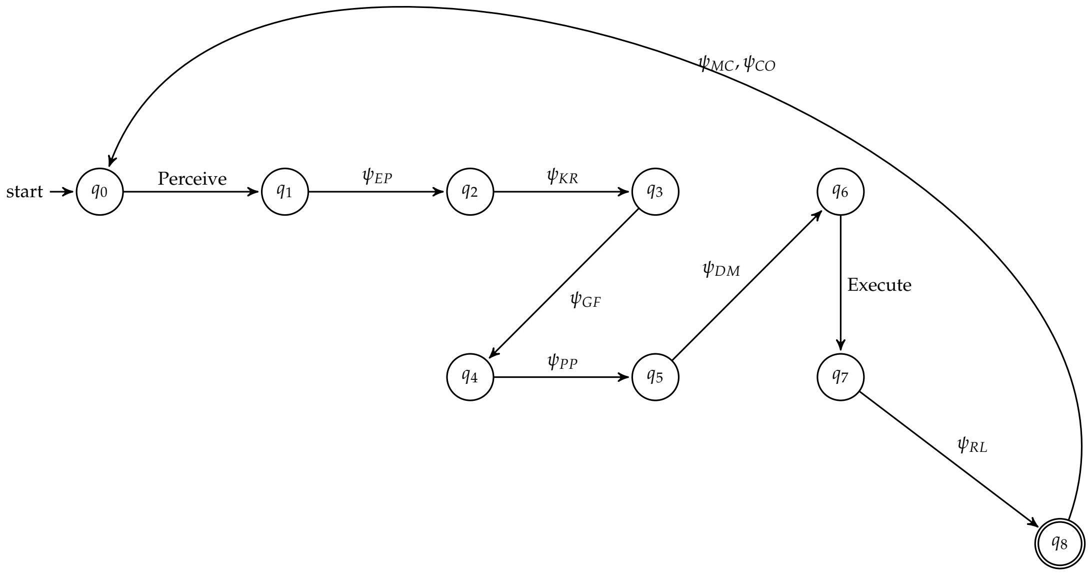

Figure 22. State diagram of the Autonomous Agents CTM

- T_G: Global model memory
- T_E: Encrypted update memory
- T_A: Aggregation memory
- T_U: Update memory

- $\Psi = \{\psi_{LT}, \psi_{EP}, \psi_{SE}, \psi_{AG}, \psi_{MU}, \psi_{DP}\}$ where:
  - ψ_{LT}: Local Training
  - ψ_{EP}: Encryption
  - ψ_{SE}: Secure Aggregation
  - ψ_{AG}: Aggregation
  - ψ_{MU}: Model Update
  - ψ_{DP}: Differential Privacy

The CRB for this CTM with d devices, each having n data points, model size m, and r communication rounds is:

$$CRB_M(d, n, m, r) = \max\{O(dnm), O(dm \log d), O(rm), O(d^2m)\}$$

Therefore, $CRB_M(d, n, m, r) = O(d^2m)$, indicating that this federated learning task is in AI-C_6.

1.24. Reinforcement Learning

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that models an advanced reinforcement learning system capable of meta-learning and multi-agent cooperation.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$
- $\Sigma = \{\text{states}\} \cup \{\text{actions}\} \cup \{\text{rewards}\} \cup \{\text{agent identifiers}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, \text{[POLICY]}, \text{[VALUE]}, \text{[META]}\}$
- q_0 is the initial state
- $F = \{q_8\}$

- $T = \{T_S, T_A, T_R, T_P, T_V, T_M, T_C\}$ where:
  - T_S: State memory
  - T_A: Action memory
  - T_R: Reward memory
  - T_P: Policy memory
  - T_V: Value function memory
  - T_M: Meta-learning memory
  - T_C: Cooperation memory
- $\Psi = \{\psi_{PE}, \psi_{PO}, \psi_{VE}, \psi_{TD}, \psi_{ML}, \psi_{IA}, \psi_{CA}\}$ where:
  - ψ_{PE}: Policy Evaluation
  - ψ_{PO}: Policy Optimization
  - ψ_{VE}: Value Estimation
  - ψ_{TD}: Temporal Difference Learning
  - ψ_{ML}: Meta-Learning
  - ψ_{IA}: Intrinsic Motivation Assessment
  - ψ_{CA}: Cooperative Action Selection

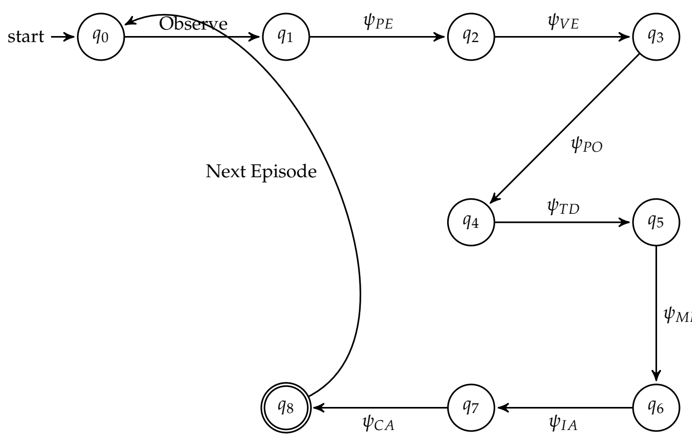

Figure 23. State diagram of the Advanced Reinforcement Learning CTM

The Cognitive Resource Bound (CRB) for this CTM with state space size s, action space size a, number of agents n, meta-learning depth m, and episode length t is:

$$CRB_M(s, a, n, m, t) = \max\{O(s^2 a), O(sa\log(sa)), O((sa)^n), O(m!), O(t\log t), O(sanmt)\}$$

Therefore, $CRB_M(s, a, n, m, t) = O(\max((sa)^n, m!, sanmt))$, indicating that this advanced reinforcement learning task is in AI-C$_7$.

1.25. Neuromorphic Computing

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that models a neuromorphic computing system inspired by the structure and function of biological neural networks.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$
- $\Sigma = \{\text{spike trains}\} \cup \{\text{synaptic weights}\} \cup \{\text{neuromodulators}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{NEURON}], [\text{SYNAPSE}], [\text{CIRCUIT}]\}$
- q_0 is the initial state
- $F = \{q_7\}$
- $T = \{T_I, T_N, T_S, T_C, T_P, T_L\}$ where:
  - T_I: Input spike train memory

- T_N: Neuron state memory
- T_S: Synaptic weight memory
- T_C: Circuit configuration memory
- T_P: Plasticity memory
- T_L: Learning rule memory

- $\Psi = \{\psi_{SP}, \psi_{NI}, \psi_{SU}, \psi_{CP}, \psi_{HP}, \psi_{NM}\}$ where:
  - ψ_{SP}: Spike Processing
  - ψ_{NI}: Neuronal Integration
  - ψ_{SU}: Synaptic Update
  - ψ_{CP}: Circuit Plasticity
  - ψ_{HP}: Homeostatic Plasticity
  - ψ_{NM}: Neuromodulation

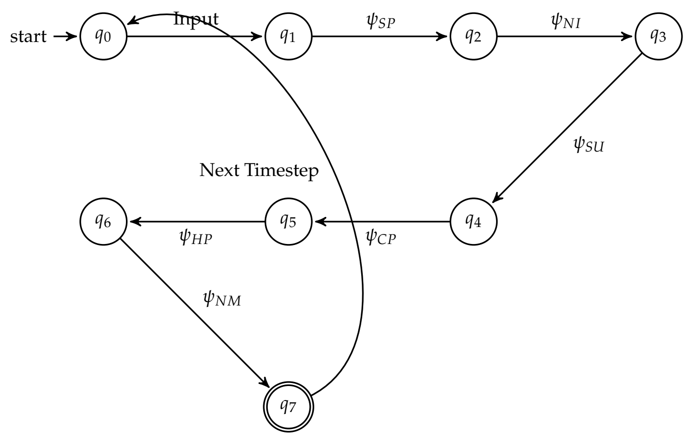

Figure 24. State diagram of the Neuromorphic Computing CTM

The Cognitive Resource Bound (CRB) for this CTM with n neurons, s synapses, c circuits, m neuromodulators, and time horizon t is:

$$CRB_M(n, s, c, m, t) = \max\{O(n \log n), O(s \log s), O(c2^n), O(m!), O(t \log t), O(nscmt)\}$$

Therefore, $CRB_M(n, s, c, m, t) = O(\max(c2^n, m!, nscmt))$, indicating that this neuromorphic computing task is in AI-C_6.

1.26. Transfer Learning

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that models a transfer learning system capable of adapting knowledge from a source domain to a target domain.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$
- $\Sigma = \{\text{source data}\} \cup \{\text{target data}\} \cup \{\text{model parameters}\} \cup \{\text{task descriptors}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{FEATURE}], [\text{MODEL}], [\text{TASK}]\}$
- q_0 is the initial state
- $F = \{q_9\}$
- $T = \{T_S, T_T, T_F, T_M, T_A, T_K, T_D\}$ where:
  - T_S: Source domain memory
  - T_T: Target domain memory
  - T_F: Feature representation memory
  - T_M: Model parameter memory
  - T_A: Adaptation strategy memory

  - T_K: Knowledge distillation memory
  - T_D: Domain discrepancy memory
- $\Psi = \{\psi_{FE}, \psi_{DA}, \psi_{KD}, \psi_{FM}, \psi_{TA}, \psi_{DD}, \psi_{MP}, \psi_{EV}\}$ where:
  - ψ_{FE}: Feature Extraction
  - ψ_{DA}: Domain Adaptation
  - ψ_{KD}: Knowledge Distillation
  - ψ_{FM}: Feature Mapping
  - ψ_{TA}: Task Adaptation
  - ψ_{DD}: Domain Discrepancy Minimization
  - ψ_{MP}: Model Parameter Transfer
  - ψ_{EV}: Evaluation

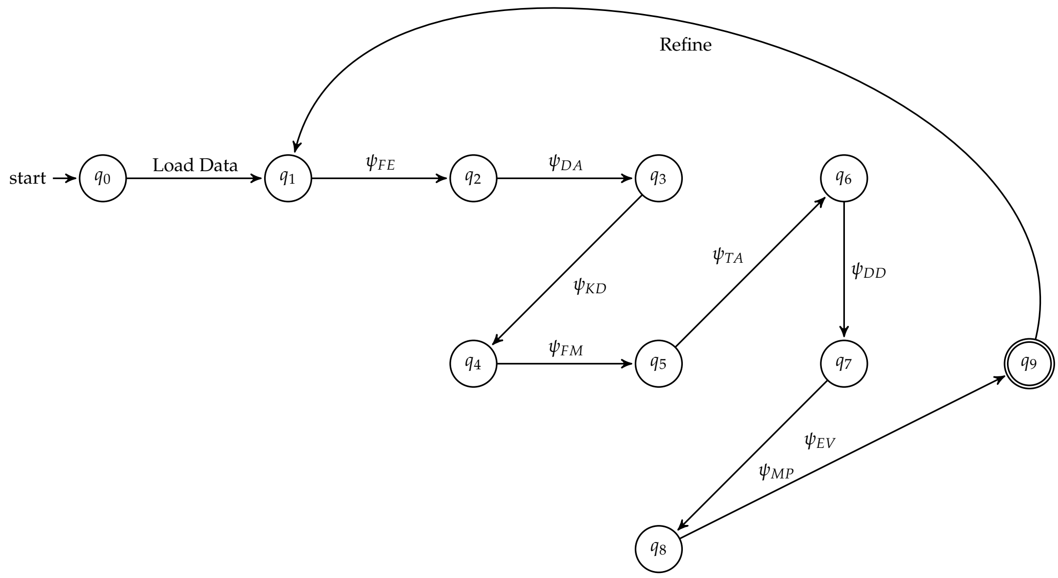

Figure 25. State diagram of the Transfer Learning

The Cognitive Resource Bound (CRB) for this CTM processing source domain data of size n_s, target domain data of size n_t, with feature space dimension d, model complexity m, and adaptation iterations i is:

$$CRB_M(n_s, n_t, d, m, i) = \max\{O(n_s \log n_s), O(n_t \log n_t), O(d^2), O(m \log m), O(2^d), O(i(n_s + n_t)), O(n$$

Therefore, $CRB_M(n_s, n_t, d, m, i) = O(\max(2^d, n_s n_t d m i))$, indicating that this transfer learning task is in AI-C$_8$.

This Transfer Learning CTM operates as follows:

1. It begins by loading data from both source and target domains (q_0 to q_1).
2. Feature extraction (ψ_{FE}) is performed on both domains (q_1 to q_2).
3. Domain adaptation (ψ_{DA}) aligns the feature spaces of source and target domains (q_2 to q_3).
4. Knowledge distillation (ψ_{KD}) transfers learned representations from source to target (q_3 to q_4).
5. Feature mapping (ψ_{FM}) creates a shared representation space (q_4 to q_5).
6. Task adaptation (ψ_{TA}) fine-tunes the model for the target task (q_5 to q_6).

7. Domain discrepancy minimization (ψ_{DD}) reduces differences between domains (q_6 to q_7).
8. Model parameter transfer (ψ_{MP}) adjusts the model for the target domain (q_7 to q_8).
9. Evaluation (ψ_{EV}) assesses the transferred model's performance (q_8 to q_9).
10. The process may iterate to refine the transfer (q_9 back to q_1).

The CRB components represent:

- $O(n_s \log n_s)$ and $O(n_t \log n_t)$: Data processing for source and target domains
- $O(d^2)$: Feature space transformations
- $O(m \log m)$: Model parameter adjustments
- $O(2^d)$: Potential complexity of feature interactions
- $O(i(n_s + n_t))$: Iterative adaptation process
- $O(n_s n_t dmi)$: Overall interaction of all components

This Transfer Learning CTM model captures the essence of modern transfer learning techniques, including domain adaptation, knowledge distillation, and feature alignment. The high AI-C classification (AI-C_8) reflects the complexity of transferring knowledge between domains, especially when dealing with high-dimensional feature spaces or significant domain shifts.

The exponential term $O(2^d)$ in the CRB highlights the potential challenge of the "curse of dimensionality" in high-dimensional feature spaces, a common issue in transfer learning. The polynomial term $O(n_s n_t dmi)$ represents the intricate interactions between source and target data, feature dimensions, model complexity, and adaptation iterations.

1.27. Time Series Forecasting

Consider a CTM $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that performs time series forecasting.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$
- $\Sigma = \{\text{time series data points}\} \cup \{\text{timestamps}\} \cup \{\text{forecast values}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{TREND}], [\text{SEASONAL}], [\text{RESIDUAL}]\}$
- q_0 is the initial state
- $F = \{q_6\}$
- $T = \{T_D, T_P, T_S, T_R, T_M, T_F\}$ where:
  - T_D: Data memory
  - T_P: Preprocessing memory
  - T_S: Seasonality detection memory
  - T_R: Trend and residual memory
  - T_M: Model memory
  - T_F: Forecast memory
- $\Psi = \{\psi_{PP}, \psi_{SD}, \psi_{TD}, \psi_{RD}, \psi_{MF}, \psi_{FC}, \psi_{EV}\}$ where:
  - ψ_{PP}: Preprocessing
  - ψ_{SD}: Seasonality Detection
  - ψ_{TD}: Trend Decomposition
  - ψ_{RD}: Residual Decomposition
  - ψ_{MF}: Model Fitting
  - ψ_{FC}: Forecasting
  - ψ_{EV}: Evaluation

The Cognitive Resource Bound (CRB) for this CTM processing a time series of length n, with seasonality period s, forecasting horizon h, and model complexity m is:

$$CRB_M(n, s, h, m) = \max\{O(n \log n), O(n \log s), O(m^2), O(nh), O(n^2)\}$$

Therefore, $CRB_M(n, s, h, m) = O(\max(m^2, n^2))$, indicating that this time series forecasting task is in AI-C_6.

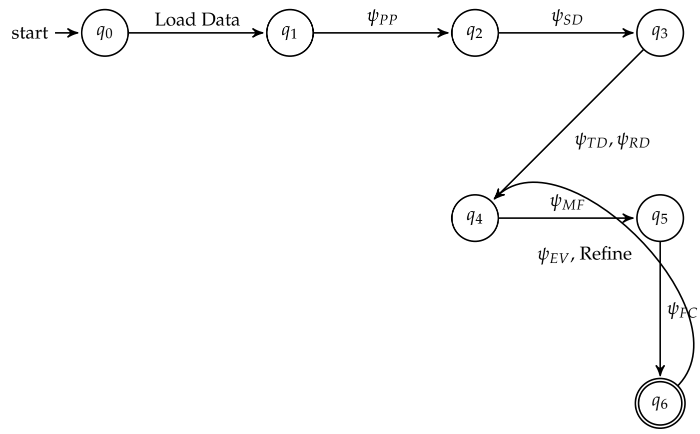

Figure 26. State diagram of the Time Series Forecasting CTM

These CTM models for image-to-image generation, video-to-video generation, and time series forecasting demonstrate the complexity of these AI tasks. The high AI-C classifications reflect the computational demands of processing and generating complex data structures.

In the image-to-image and video-to-video CTMs, the exponential term $O(2^d)$ represents the complexity of the latent space. The $O(n^4)$ and $O(f^2n^4)$ terms reflect the intricate pixel-level manipulations and temporal consistency requirements.

For the time series forecasting CTM, $O(m^2)$ captures the model complexity, while $O(n^2)$ represents the potential for complex long-range dependencies in the time series.

1.28. Artificial General Intelligence

Consider a Cognitive Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$ that models an advanced AGI system capable of complex reasoning, learning, and adaptation across multiple domains.

- $Q = \{q_0, q_1, ..., q_{12}\}$
- $\Sigma = \{\text{sensory inputs}\} \cup \{\text{abstract concepts}\} \cup \{\text{logical propositions}\} \cup \{\text{actions}\} \cup \{\text{emotional states}\}$
- $\Gamma = \Sigma \cup \{\sqcup, \$, [\text{CONCEPT}], [\text{RULE}], [\text{GOAL}], [\text{PLAN}], [\text{EMOTION}], [\text{BELIEF}]\}$
- q_0 is the initial state
- $F = \{q_{12}\}$
- $T = \{T_S, T_K, T_R, T_G, T_P, T_M, T_C, T_L, T_E, T_B, T_I, T_A\}$ where:
  - T_S: Sensory memory
  - T_K: Knowledge base memory
  - T_R: Reasoning memory
  - T_G: Goal memory
  - T_P: Planning memory
  - T_M: Meta-cognitive memory
  - T_C: Creativity memory
  - T_L: Learning memory
  - T_E: Emotional memory
  - T_B: Belief system memory
  - T_I: Introspection memory
  - T_A: Adaptive strategy memory
- $\Psi = \{\psi_{MP}, \psi_{AB}, \psi_{AN}, \psi_{CR}, \psi_{NLP}, \psi_{SL}, \psi_{RL}, \psi_{MC}, \psi_{CI}, \psi_{TO}, \psi_{EM}, \psi_{BU}, \psi_{IN}, \psi_{AD}\}$ where:
  - ψ_{MP}: Multi-modal Perception
  - ψ_{AB}: Abstraction
  - ψ_{AN}: Analogical Reasoning
  - ψ_{CR}: Causal Reasoning

- ψ_{NLP}: Natural Language Processing
- ψ_{SL}: Symbolic Learning
- ψ_{RL}: Reinforcement Learning
- ψ_{MC}: Metacognition
- ψ_{CI}: Creative Ideation
- ψ_{TO}: Theory of Mind
- ψ_{EM}: Emotional Modeling
- ψ_{BU}: Belief Updating
- ψ_{IN}: Introspection
- ψ_{AD}: Adaptive Strategy Formation

The transition function δ is defined as a complex mapping:

$$\delta : Q \times \Gamma^{12} \to Q \times \Gamma^{12} \times \{L, R, S\}^{12} \times \Psi^*$$

This allows for intricate interactions between all memory tapes and cognitive operations.

The Cognitive Resource Bound (CRB) for this advanced AGI CTM is more nuanced:

$$CRB_M(n, k, d, c, e, b, i, a) = \max\{O(n^2), O(k \log k), O(2^d), O(c!), O(e \log e), O(b^2), O(i \log i), O(a^2), O$$

Where:

- n: size of sensory input
- k: size of knowledge base
- d: reasoning depth
- c: creativity factor
- e: emotional complexity
- b: belief system complexity
- i: introspection depth
- a: adaptive strategy space

Therefore, $CRB_M(n, k, d, c, e, b, i, a) = O(\max(2^d, c!, nkdcebia))$, indicating that this advanced AGI complex reasoning task is in AI-C_{14}.

This enhanced AGI CTM incorporates several additional advanced capabilities:

- Emotional modeling (ψ_{EM}) allows the system to understand and simulate emotional states, crucial for human-like decision making and social interaction.
- Belief updating (ψ_{BU}) enables the system to dynamically adjust its belief system based on new information and experiences.
- Introspection (ψ_{IN}) allows for deep self-analysis and understanding of the system's own cognitive processes.
- Adaptive strategy formation (ψ_{AD}) enables the system to create and modify strategies for problem-solving based on past experiences and current context.

The complexity of this AGI system is reflected in its high AI-C classification (AI-C_{14}). This indicates the extraordinary computational and cognitive resources required for a system approaching human-level general intelligence.

The CRB now includes additional terms:

- $O(e \log e)$ represents the complexity of emotional processing
- $O(b^2)$ captures the intricacy of belief system management
- $O(i \log i)$ reflects the depth of introspection
- $O(a^2)$ represents the complexity of adaptive strategy formation

The term $O(nkdcebia)$ captures the intricate interactions between all aspects of the AGI system, highlighting the interconnected nature of general intelligence.

2. AI Complexity Classes

We now propose and prove the completeness of problems for our main AI complexity classes. These problems serve as canonical representatives of their respective classes, allowing us to understand the fundamental nature of cognitive computation at different complexity levels. Figure 27 presents the diagrammatic approach of AI complexity classes.

2.1. *AI-Complete Problem*

Following Yampolskiy's definitions [?][?], we can formally define AI-completeness in terms of these cognitive operations:

Definition 1 (AI-Complete Problem)**.** *A problem C is AI-complete if:*

1. $C \in$ *AI-Problems, where AI-Problems is the set of problems solvable by a human oracle.*
2. $\forall P \in$ *AI-Problems*$, \exists f : P \leq_p C$*, where f is a polynomial-time reduction and $\leq_p$ denotes polynomial-time reducibility.*

2.2. *AI-Hard Problem*

We can then define AI-hardness:

Definition 2 (AI-Hard Problem)**.** *A problem H is AI-hard if and only if there exists an AI-complete problem C such that $C \leq_p H$.*

The cognitive operations in Ψ are designed to capture the essence of AI-complete and AI-hard problems. For example, the natural language understanding operation ψ_{NLU} is believed to be AI-complete, as it requires human-like intelligence to perform effectively across all domains.

It's important to note that while these operations are defined formally, their exact implementation in a physical system remains an open question. As Groppe and Jain suggest [?], current AI systems can only solve restricted versions of AI-complete problems.

Furthermore, as Šekrst points out [?], even if we find polynomial-time solutions to problems involving these cognitive operations, it may not necessarily solve the broader challenge of strong AI.

2.3. *AI-P Complete Problem*

Definition 3 (Bounded Cognitive Reasoning (BCR))**.** *Instance: A tuple (F, R, Q, k) where*

- $F = \{f_1, ..., f_m\}$ *is a set of facts in propositional logic*
- $R = \{r_1, ..., r_n\}$ *is a set of inference rules of the form $p_1 \wedge ... \wedge p_j \rightarrow q$*
- *Q is a query in propositional logic*
- *k is a positive integer*

Question: Does there exist a sequence of at most k applications of rules from R to facts in F and previously derived facts that results in Q?

Theorem 1. *BCR is AI-P complete.*

Proof. First, we show BCR $\in$ AI-P: Construct a CTM M that:

1. Initializes a "fact tape" with F
2. For each step i from 1 to k:
   - For each rule $r \in R$:
     - Check if the antecedents of r match facts on the fact tape
     - If so, add the consequent to the fact tape
   - If Q is on the fact tape, accept
3. If k steps completed without accepting, reject

This algorithm runs in time $O(k|R|(|F| + k|R|))$, which is polynomial in the input size.

Now, we prove BCR is AI-P hard: Let $L \in$ AI-P be decided by CTM M in time $p(n)$ for some polynomial p. We reduce L to BCR as follows:

1. For input x, construct an instance (F, R, Q, k) of BCR where:
   - F encodes the initial configuration of M on x
   - R contains rules simulating M's transition function
   - Q represents M's accepting state
   - $k = p(|x|)$
2. This reduction is computable in polynomial time and $x \in L$ if and only if the constructed BCR instance is a yes-instance.

We conclude that BCR is AI-P complete. □

2.4. *AI-NP Complete Problem*

Definition 4 (Cognitive Satisfiability (CSAT))**.** *Instance: A boolean formula ϕ over variables $V = \{v_1, ..., v_n\}$ representing cognitive states. Question: Does there exist an assignment $\alpha : V \to \{0, 1\}$ such that $\phi(\alpha(v_1), ..., \alpha(v_n)) = 1$?*

Theorem 2. *CSAT is AI-NP complete.*

Proof. First, we show CSAT $\in$ AI-NP: Construct a non-deterministic CTM M that:

1. Non-deterministically generates an assignment α
2. Evaluates ϕ under α
3. Accepts if $\phi(\alpha) = 1$, rejects otherwise

This process takes polynomial time, so CSAT $\in$ AI-NP.

Now, we prove CSAT is AI-NP hard: Let $L \in$ AI-NP be decided by non-deterministic CTM M in time $p(n)$ for some polynomial p. We reduce L to CSAT as follows:

1. For input x, construct a boolean formula ϕ_x that:
   - Has variables representing each cell of each tape of M at each time step up to $p(|x|)$
   - Has clauses ensuring:
     - The initial configuration correctly represents x
     - Each step follows M's transition function
     - The final configuration is accepting
2. This reduction is computable in polynomial time and $x \in L$ if and only if ϕ_x is satisfiable.

Thus CSAT is AI-NP complete. □

2.5. *AI-PSPACE Complete Problem*

Definition 5 (Quantified Cognitive Boolean Formula (QCBF))**.** *Instance: A fully quantified boolean formula $\Phi = Q_1 v_1 Q_2 v_2 ... Q_n v_n \phi(v_1, ..., v_n)$ where each Q_i is either $\exists$ or $\forall$, and ϕ is a boolean formula over variables $v_1, ..., v_n$ representing cognitive states. Question: Is Φ true?*

Theorem 3. *QCBF is AI-PSPACE complete.*

Proof. First, we show QCBF $\in$ AI-PSPACE: Construct a CTM M that:

1. Recursively evaluates Φ using the following algorithm EVAL(Ψ):
   - If Ψ has no quantifiers, return the value of the boolean formula
   - If $\Psi = \exists v \Psi'$, return EVAL($\Psi'[v = 0]$) OR EVAL($\Psi'[v = 1]$)
   - If $\Psi = \forall v \Psi'$, return EVAL($\Psi'[v = 0]$) AND EVAL($\Psi'[v = 1]$)
2. This algorithm uses space polynomial in the size of Φ

Now, we prove QCBF is AI-PSPACE hard: Let $L \in$ AI-PSPACE be decided by CTM M using space $p(n)$ for some polynomial p. We reduce L to QCBF as follows:

1. For input x, construct a QCBF instance Φ_x that:
   - Has variables representing each cell of each tape of M
   - Uses existential quantifiers to guess a sequence of configurations
   - Uses universal quantifiers to verify all possible steps
   - Has a boolean formula verifying that the guessed sequence is a valid accepting computation of M on x
2. This reduction is computable in polynomial time and $x \in L$ if and only if Φ_x is true.

Hence, QCBF is AI-PSPACE complete. □

2.6. ***AI-APX Problem***

Definition 6 (AI-APX)**.** *A maximization problem* Π *is in AI-APX (AI Approximable) if there exists a polynomial-time cognitive Turing machine* M *and a constant* $\alpha > 1$ *such that for every instance* I *of* Π*:*

$$\frac{OPT(I)}{M(I)} \leq \alpha$$

where $OPT(I)$ *is the optimal solution value for instance* I*, and* $M(I)$ *is the solution value produced by* M*.*

Theorem 4 (AI-APX Completeness)**.** *The Cognitive Max-SAT problem is AI-APX-complete.*

Definition 7 (Cognitive Max-SAT)**.** *Given a cognitive boolean formula* ϕ *in CNF, find an assignment that satisfies the maximum number of clauses.*

Proof. 1. Cognitive Max-SAT $\in$ AI-APX: There exists a polynomial-time cognitive algorithm that achieves a 2-approximation for Max-SAT.

2. For any problem $\Pi \in$ AI-APX, we can construct a polynomial-time reduction to Cognitive Max-SAT that preserves the approximation ratio.

The detailed proof follows the structure of the classical Max-SAT APX-completeness proof, adapted to the cognitive setting. □

Proposition 1 (AI-APX Hardness)**.** *There exists a problem in AI-APX that is not in AI-P unless AI-P = AI-NP.*

2.7. ***AI-EXP Problem***

Definition 8 (AI-EXP)**.** *A language* L *is in AI-EXP (AI Exponential Time) if there exists a cognitive Turing machine* M *and a constant* $k > 0$ *such that:*

- *For all* $x \in L$*,* M *accepts* x
- *For all* $x \notin L$*,* M *rejects* x
- M *halts on all inputs* x *in time* $O(2^{n^k})$*, where* $n = |x|$

Theorem 5 (AI-EXP Hierarchy)**.** *AI-P* $\subsetneq$ *AI-EXP*

Proof. 1. AI-P $\subseteq$ AI-EXP: Any polynomial-time cognitive Turing machine runs in exponential time.

2. AI-P $\neq$ AI-EXP: We use diagonalization to construct a language in AI-EXP that is not in AI-P.

- Let $M_1, M_2, ...$ be an enumeration of all polynomial-time cognitive Turing machines.
- Define $L = \{1^n : M_n \text{ does not accept } 1^n \text{ in } n^{\log n} \text{ steps}\}$
- $L \in$ AI-EXP: We can simulate M_n on 1^n for $n^{\log n}$ steps in exponential time.
- $L \notin$ AI-P: Assume $L \in$ AI-P. Then there exists k such that M_k decides L in polynomial time. Consider the input 1^k:
  - If M_k accepts 1^k, then $1^k \notin L$, a contradiction.

- If M_k rejects 1^k, then $1^k \in L$, a contradiction.

We conclude that AI-P $\subsetneq$ AI-EXP. □

Proposition 2 (AI-EXP Complexity)**.** *AI-NP $\subsetneq$ AI-EXP*

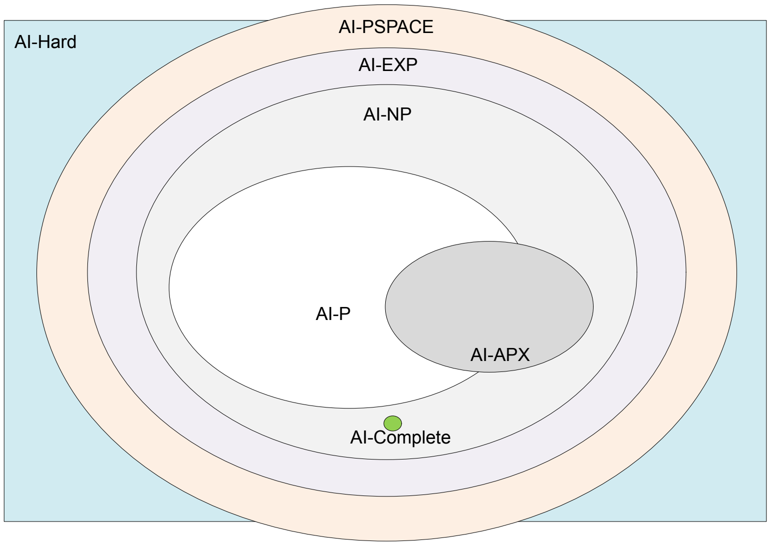

Figure 27. AI Complexity classes.

3. Quantum AI Complexity Theory

We now extend our theory to the quantum domain, introducing quantum CTMs (QCTMs) and related complexity classes. This extension allows us to explore the potential advantages of quantum computation in the context of AI and cognitive processing.

3.1. *Quantum Cognitive Turing Machines*

Definition 9 (Quantum Cognitive Turing Machine (QCTM))**.** *A QCTM is a tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, F, T, \Psi)$, where:*

- *Q is a finite set of quantum states*
- *Σ is the input alphabet*
- *Γ is the tape alphabet*
- *$q_0 \in Q$ is the initial state*
- *$F \subseteq Q$ is the set of accepting states*
- *$T = \{T_1, ..., T_k\}$ is a finite set of quantum tapes representing cognitive functions*
- *Ψ is a finite set of quantum cognitive operations*
- *$\delta : Q \times \Gamma^{k+1} \to \mathbb{C}^{(Q \times \Gamma^{k+1} \times \{L,R\}^{k+1} \times \Psi^*)}$ is the quantum transition function*

such that for any configuration c, $\sum_{c'} |\delta(c, c')|^2 = 1$, where the sum is over all possible next configurations c'.

The quantum transition function δ allows for superposition of computational paths, enabling quantum parallelism in cognitive processing. The unitarity condition ensures that the evolution of the QCTM is reversible and preserves probability.

Definition 10 (Quantum Cognitive Operation)**.** *A quantum cognitive operation $\psi \in \Psi$ is a unitary transformation on the quantum state space of the cognitive tapes. Formally, $\psi : \mathcal{H}_T \to \mathcal{H}_T$, where $\mathcal{H}_T$ is the Hilbert space representing the state of the cognitive tapes.*

3.2. Quantum AI Complexity Classes

Definition 11 (QAI-P)**.** *QAI-P is the class of languages L for which there exists a QCTM M and a polynomial p such that:*

1. *For all inputs x, M halts in at most $p(|x|)$ steps*
2. *If $x \in L$, M accepts with probability at least 2/3*
3. *If $x \notin L$, M rejects with probability at least 2/3*

Definition 12 (QAI-BQP)**.** *QAI-BQP is the class of languages L for which there exists a QCTM M and polynomials p and q such that:*

1. *For all inputs x, M halts in at most $p(|x|)$ steps*
2. *M uses at most $q(|x|)$ qubits*
3. *If $x \in L$, M accepts with probability at least 2/3*
4. *If $x \notin L$, M rejects with probability at least 2/3*

Theorem 6 (QAI-P and QAI-BQP Relationship)**.** *QAI-BQP $\subseteq$ QAI-P*

Proof. Any QCTM using polynomially many qubits can be simulated by a QCTM without this restriction, so the inclusion follows directly from the definitions. □

Theorem 7 (Classical Simulation)**.** *For all $k \geq 1$, AI-$C_k \subseteq$ QAI-C_k*

Proof. Let $L \in$ AI-C_k. Then there exists a CTM M with k cognitive tapes that decides L. We can construct a QCTM M' that simulates M as follows:

1. M' uses k quantum tapes to represent the k cognitive tapes of M
2. For each state q of M, M' has a corresponding basis state $|q\rangle$
3. The transition function δ' of M' is defined to mimic δ of M: $\delta'(|q\rangle|\gamma_1\rangle...|\gamma_k\rangle) = |q'\rangle|\gamma'_1\rangle...|\gamma'_k\rangle|d_1\rangle...|d_k\rangle|\psi\rangle$ where $\delta(q, \gamma_1, ..., \gamma_k) = (q', \gamma'_1, ..., \gamma'_k, d_1, ..., d_k, \psi)$
4. Quantum cognitive operations in Ψ' are defined to act as their classical counterparts when applied to basis states

This construction ensures that M' evolves exactly as M, without using any quantum superposition or interference. Thus, $L \in$ QAI-C_k. □

3.3. Quantum Cognitive Advantage

We now explore potential advantages of quantum cognitive computation.

Theorem 8 (Quantum Speedup for Cognitive Search)**.** *There exists a cognitive search problem that can be solved by a QCTM in $O(\sqrt{N})$ steps, where N is the size of the search space, while any classical CTM requires $\Omega(N)$ steps.*

Proof. We adapt Grover's algorithm to the cognitive setting. Grover's algorithm is known to provide a quadratic speedup for unstructured search problems in the quantum computing paradigm. Here, we outline how this algorithm can be applied within the context of a Quantum Cognitive Turing Machine (QCTM):

1. Superposition Initialization: Encode the search space into a superposition state across the quantum cognitive tapes. This step involves preparing the initial state as a uniform superposition of all possible states. Formally, this can be written as:

$$\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

where N is the size of the search space and $|x\rangle$ are the basis states representing possible solutions.

2. Oracle Operation: Define a quantum cognitive operation (oracle) that marks the target state. This operation flips the phase of the state that corresponds to the solution. If $|w\rangle$ is the target state, the oracle O acts as follows:

$$O|x\rangle = \begin{cases} -|x\rangle & \text{if } x = w \\ |x\rangle & \text{if } x \neq w \end{cases}$$

3. Grover Diffusion Operator: Apply the quantum cognitive version of Grover's diffusion operator. The diffusion operator amplifies the probability amplitude of the target state while reducing the amplitude of other states. This can be described by the unitary operation D:

$$D = 2|\psi\rangle\langle\psi| - I$$

where $|\psi\rangle$ is the initial superposition state, and I is the identity operator.

4. Iteration: Perform the iteration of the oracle and diffusion operators $O(\sqrt{N})$ times. Each iteration increases the probability of measuring the target state.

5. Measurement: Measure the quantum state after $O(\sqrt{N})$ iterations to obtain the solution with high probability.

The efficiency of Grover's algorithm ensures that the QCTM solves the cognitive search problem in $O(\sqrt{N})$ steps. In contrast, any classical CTM performing an unstructured search would require examining each element in the search space, resulting in $\Omega(N)$ steps. This difference establishes the quadratic speedup achieved by the QCTM.

Thus, we have shown that there exists a cognitive search problem for which a QCTM provides a significant computational advantage over a classical CTM. □

3.4. ***Quantum Cognitive Entanglement***

We introduce the concept of quantum cognitive entanglement to model complex interdependencies between cognitive processes.

Definition 13 (Cognitive Entanglement Measure)**.** *For a QCTM M with quantum cognitive tapes $T_1, ..., T_k$, the Cognitive Entanglement Measure E_M is defined as:*

$$E_M = \frac{1}{k(k-1)} \sum_{i \neq j} S(\rho_{T_i T_j})$$

where $S(\rho)$ is the von Neumann entropy and $\rho_{T_i T_j}$ is the reduced density matrix of tapes T_i and T_j.

Proposition 3 (Entanglement and Cognitive Power)**.** *For any language L, if $L \in$ QAI-P and $L \notin$ AI-P, then any QCTM deciding L in polynomial time must have non-zero Cognitive Entanglement Measure for infinitely many input lengths.*