农夫山泉

TEST BANK

Software and Symbolic AI

---

48 Multiple Choice
48 True False
12 Essay

Nongfu

Symbolic AI

Multiple Choice

Directions : Read each question carefully, choose the correct answer from A, B, C, D!

1. **What distinguishes Symbolic AI from statistical or neural approaches?**
   A. It learns through trial and error using layers of neurons.
   B. It represents knowledge using symbols, logic, and explicit reasoning rules.
   C. It ignores reasoning and focuses only on perception tasks.
   D. It uses probability distributions instead of symbolic structures.

2. **In Symbolic AI, what is the primary role of the 'Knowledge Base'?**
   A. To visualize the system's reasoning in real time.
   B. To store and organize structured facts, concepts, and rules about a specific domain.
   C. To perform low-level hardware control and data transfer.
   D. To replace the inference engine's logic.

3. **In a Symbolic AI system, what is the main function of the 'Inference Engine'?**
   A. To apply logical rules to known facts in order to infer new conclusions.
   B. To collect raw sensory data from the environment.
   C. To convert natural language into sound waves.
   D. To visualize user preferences.

4. **Why did early symbolic expert systems like MYCIN eventually lose popularity?**
   A. They lacked internet connectivity.
   B. They were too dependent on statistical learning.
   C. Their rule bases were difficult to maintain and scale as domains grew.
   D. They had no inference mechanisms.

5. **What does 'Forward Chaining' mean in the context of rule-based reasoning?**
   A. Ignoring existing facts and generating new ones from scratch.
   B. Starting from a hypothesis and working backward to find facts.
   C. Randomly choosing rules until a solution appears.
   D. Starting from known facts and applying rules to reach new conclusions.

6. **How does Symbolic AI handle knowledge representation differently from statistical AI approaches?**
   A. It stores relationships and facts in logical structures that machines can interpret explicitly.
   B. It uses trial-and-error pattern discovery through massive datasets.

C. It depends entirely on probability distributions to define meaning.
D. It focuses on continuous learning without predefined logic.

7. **Which of the following statements correctly describes backward chaining?**
   A. It uses sensory data to generate symbolic patterns.
   B. It begins with a goal and searches backward to find supporting rules or facts.
   C. It automatically learns rules from unstructured data.
   D. It skips inference and directly outputs predictions.

8. **How does Symbolic AI contribute to the development of *explainable AI (XAI)*?**
   A. By automating explanations through neural weight mapping.
   B. By using random exploration to generate hidden connections.
   C. By concealing decision-making steps for faster inference.
   D. By providing reasoning processes that can be traced and understood by humans.

9. **Which limitation most often restricts Symbolic AI's performance in complex environments?**
   A. Its excessive dependence on unsupervised neural layers.
   B. Its overreliance on sensory perception from robotics systems.
   C. Its inability to efficiently process uncertain or unstructured real-world data.
   D. Its lack of compatibility with logic-based programming languages.

10. **Why do many modern AI researchers combine Symbolic AI with machine learning?**
    A. To integrate logical reasoning with adaptive, data-driven learning.
    B. To replace symbolic logic entirely with gradient-based learning.
    C. To remove human-designed rules from inference engines.
    D. To prioritize visual processing over reasoning.

11. **Which of the following scenarios best illustrates Symbolic AI in real-world use?**
    A. An expert system that provides legal advice using structured rules and defined logic.
    B. A translation model that learns from millions of sentence pairs without logic rules.
    C. A social media algorithm that predicts trends using only data correlations.
    D. A camera app that identifies faces using pattern recognition.

12. **What defines a production rule within Symbolic AI reasoning systems?**
    A. A probabilistic model used for approximating rule outcomes.
    B. A fixed neural connection between symbolic and numeric data.
    C. A sequence of random patterns stored for later learning.
    D. A conditional statement that follows an "if–then" structure to derive logical conclusions.

13. **Which of the following best describes a *knowledge representation* in Symbolic AI?**
    A. A collection of unrelated numeric data.

B. A structured model of facts, relationships, and logic.
C. A set of randomly generated weights.
D. A list of neural activations.

14. What kind of reasoning allows Symbolic AI to draw conclusions from incomplete or uncertain information?
A. Deductive reasoning
B. Inductive reasoning
C. Abductive reasoning
D. Associative reasoning

15. In symbolic systems, what are *frames* used for?
A. Storing pre-trained neural layers.
B. Representing stereotyped situations with slots and values.
C. Managing file systems in AI hardware.
D. Measuring inference speed.

16. Which programming language is historically associated with Symbolic AI development?
A. Python
B. Lisp
C. C++
D. Swift

17. What is a *semantic network* in the context of Symbolic AI?
A. A type of deep learning architecture.
B. A graph structure linking concepts through relationships.
C. A computer network optimized for AI processing.
D. A reinforcement learning loop.

18. What makes symbolic reasoning "transparent"?
A. It runs faster than neural networks.
B. Its logical steps can be explicitly inspected and explained.
C. It hides reasoning behind probabilistic outputs.
D. It requires no human-defined rules.

19. Which of the following problems is most difficult for Symbolic AI to handle?
A. Rule-based decision-making
B. Structured logical inference
C. Unstructured visual pattern recognition
D. Knowledge base querying

20. In Symbolic AI, rules of inference such as *Modus Ponens* are used to:

A. Train neural weights
B. Deduce new knowledge from known premises
C. Store procedural memory
D. Represent sensory data

21. What is one way Symbolic AI supports Natural Language Processing (NLP)?

A. Through logic-based parsing and semantic interpretation
B. By random token prediction
C. By compressing sentences using autoencoders
D. By ignoring grammatical structure

22. Which of the following best defines *knowledge engineering*?

A. Automating rule extraction from massive data
B. Designing and maintaining symbolic knowledge bases
C. Compiling high-level programming code
D. Measuring AI hardware performance

23. Which AI approach is most likely to combine Symbolic reasoning and statistical learning?

A. Reactive AI
B. Hybrid AI
C. Evolutionary AI
D. Unsupervised clustering

24. Why did Symbolic AI face challenges in perception and vision tasks?

A. Because logic-based systems struggled with ambiguous, noisy data.
B. Because rule-based systems required too many GPUs.
C. Because symbolic systems couldn't encode human language.
D. Because they depended only on deep neural representations.

True False

Directions : Read each question carefully, choose the correct answer from true or false!

1. Symbolic AI is often described as GOFAI (Good old fashioned AI) T
2. Symbolic AI is also called rule-based AI T
3. Symbolic AI is more suited for tasks requiring large amounts of data to learn. F
4. Symbolic AI does not use logic for reasoning. F

5. In Symbolic AI, reasoning is transparent and explainable. T
6. Chess programs from the early AI era is one of the main examples of Symbolic AI T
7. Symbolic AI cannot easily adapt to new or unexpected information T
8. In Symbolic AI, symbols must always correspond to physical objects. F
9. OWL (Web Ontology Language) is used to represent structured knowledge in symbolic AI. T
10. Symbolic AI can be integrated with neural networks to create hybrid AI systems. T
11. Symbolic AI is especially good at recognizing images or sounds. F
12. Symbolic AI systems require less human knowledge input compared to machine learning models. F
13. Symbolic AI represents knowledge using symbols, logic, and structured relationships between concepts. T
14. The inference engine in a symbolic AI system applies logical rules to existing facts to derive new conclusions. T
15. Symbolic AI relies primarily on neural network weights and statistical correlations. F
16. Forward chaining in symbolic reasoning starts from known facts and applies rules to infer new information. T
17. Backward chaining begins with a hypothesis or goal and works backward to find supporting evidence. T
18. Symbolic AI systems are typically easy to scale and maintain as knowledge bases grow larger. F
19. Expert systems such as MYCIN and DENDRAL are early examples of Symbolic AI applications. T
20. One advantage of Symbolic AI is that its reasoning process can be easily understood and explained by humans. T
21. Symbolic AI performs best in uncertain, unstructured, and noisy real-world environments. F
22. Knowledge representation in Symbolic AI can include semantic networks, ontologies, and frames. T
23. Symbolic AI uses probabilistic models to approximate reasoning instead of formal logic. F
24. Combining Symbolic AI with data-driven machine learning can lead to more powerful and interpretable hybrid AI systems. T

Software

I. Multiple Choice

Directions : Read each question carefully, choose the correct answer from A, B, C, D!

1. **Which statement best explains the role of software in computational systems?**
   A. Software provides physical infrastructure for logic gates.
   B. Software transforms theoretical algorithms into executable operations through hardware interaction.
   C. Software only manages storage devices.
   D. Software is limited to user interface design.

2. **Which of the following statements best differentiates system software and application software?**
   A. System software manages hardware and supports other programs, while application software allows users to perform specific tasks.
   B. System software is optional, but application software is mandatory for hardware operation.
   C. Application software controls devices directly, while system software handles user interfaces.
   D. Both system and application software operate independently of the operating system.

3. **Which software type is designed to help programmers write, test, and debug other programs?**
   A. Firmware
   B. Utility software
   C. Application software
   D. Development software

4. **What was the main lesson learned from the software crisis of the late 20th century?**
   A. Hardware advancements would eventually eliminate the need for software maintenance.
   B. Complex software projects required systematic engineering methods, not ad-hoc coding.
   C. Software development could be completed without documentation.
   D. Programmers should rely solely on machine code to avoid bugs.

5. **What is the kernel of an operating system responsible for?**

A. Running application interfaces directly
B. Handling software licensing
C. Managing CPU scheduling, memory, and device communication
D. Translating programming languages into binary

6. **Why was the UNIX operating system developed in 1969 considered significant?**
A. It focused on graphical design for end users
B. It introduced a multi-user, multitasking environment and influenced modern OS design
C. It removed the need for hardware drivers
D. It was the first commercial spreadsheet system

7. **In the software hierarchy, what did middleware serve as?**
A. A communication bridge connecting applications and databases
B. The outer layer that manages user interaction
C. A backup layer for operating systems
D. The portion that handles physical input/output devices

8. **During the software maintenance phase of the Software Development Life Cycle (SDLC), which activity is most critical to ensure long-term software reliability?**
A. Writing initial user requirements
B. Designing user interfaces
C. Translating business goals into system architecture
D. Fixing bugs, improving performance, and updating functionalities

9. **Why is version control software critical in modern software engineering?**
A. It encrypts user passwords automatically
B. It compiles source code for multiple operating systems
C. It tracks, manages, and merges code changes among multiple developers
D. It automatically improves CPU performance

10. **How does the Agile development model differ from the Waterfall model?**
A. It eliminates documentation requirements entirely
B. It encourages iterative development and continuous feedback
C. It relies on a rigid, sequential process
D. It focuses only on hardware testing

11. **Which statement best characterizes open-source software?**
A. It is licensed only to government agencies
B. It can only be installed with proprietary hardware
C. It is developed collaboratively and made freely accessible for use and improvement
D. It prohibits redistribution or modification

12. **What defines hybrid software systems in modern computing?**
A. They use analog hardware for all processing tasks

B. They rely exclusively on declarative programming languages
C. They replace human decision-making completely
D. They integrate symbolic, rule-based logic with data-driven adaptability

13. **Which of the following best describes *system software*?**
    A. Software used exclusively by end-users.
    B. Programs that control and manage hardware components.
    C. Software that generates marketing data.
    D. A mobile app installed by users.

14. **What is *firmware* typically responsible for?**
    A. Executing user applications
    B. Managing hardware at the lowest operational level
    C. Creating web interfaces
    D. Installing operating systems

15. **What is the main purpose of *utility software*?**
    A. Enhancing hardware graphics
    B. Performing maintenance and optimization tasks
    C. Storing user passwords
    D. Designing user interfaces

16. **In the Software Development Life Cycle (SDLC), which phase focuses on defining what the system should do?**
    A. Testing
    B. Analysis
    C. Deployment
    D. Maintenance

17. **Which of these is an example of *open-source* operating system software?**
    A. macOS
    B. Windows
    C. Linux
    D. iOS

18. **Which type of software allows multiple virtual computers to run on a single physical machine?**
    A. Compiler
    B. Emulator
    C. Hypervisor
    D. Loader

19. **Which of the following is NOT an example of application software?**
    A. Microsoft Word

B. Photoshop
C. Device Driver
D. PowerPoint

20. **The process of finding and fixing logical or runtime errors in software is called:**
A. Debugging
B. Compiling
C. Parsing
D. Refactoring

21. **What is a *patch* in software maintenance?**
A. A complete rewrite of the software
B. A small update to fix bugs or security issues
C. A tool for uninstalling applications
D. A method for data compression

22. **Which software development model emphasizes continuous integration and frequent releases?**
A. Spiral Model
B. V-Model
C. Agile Model
D. Waterfall Model

23. **What distinguishes *proprietary software* from *open-source software*?**
A. Proprietary software allows modification of its code.
B. Proprietary software is closed and owned by a company.
C. Proprietary software is always free.
D. Proprietary software is illegal to distribute.

24. **Which of the following describes *middleware* most accurately?**
A. Software that directly manages input/output devices.
B. A bridge enabling communication between different applications or services.
C. A backup copy of application data.
D. A graphical user interface engine.

II. True False

Directions : Read each question carefully, choose the correct answer from true or false!

1. Software is a physical component in a computer F
2. Software cannot function without a hardware T
3. Microsoft Excel and Google Chrome is an example of an application software T
4. Application software is used to perform user-specific tasks. T
5. An operating system is considered application software. F
6. Software tells the computer what to do. T
7. Without software, a computer cannot work. T
8. System software acts as a bridge between hardware and application software.T
9. Open-source software means not anyone can view, change, and share its code. F
10. All software is stored permanently in the computer's ROM. F
11. Middleware is software that connects two separate applications or systems. System software acts as a bridge between hardware and application software. T
12. Spreadsheet is an application software for calculations and data visualization. T
13. Firmware is a type of software permanently stored in hardware devices to control their functions. T
14. Utility software helps manage, maintain, and protect computer resources. T
15. System software and application software perform the exact same purpose. F
16. The process of updating software to fix bugs or add new features is called software maintenance. T
17. The kernel is the outermost layer of an operating system that handles user interface design. F
18. Software piracy refers to the unauthorized copying, distribution, or use of software. T
19. Closed-source software allows anyone to access and modify its source code freely. F
20. Device drivers are examples of system software that enable hardware components to communicate with the operating system. T
21. In cloud computing, software and data are stored and accessed over the internet instead of on a local device. T
22. Software bugs are intentional errors added to programs to test user reactions. F
23. Programming languages such as Python and Java are used to create software applications. T
24. The term "software lifecycle" refers only to the design and coding stages of a program. F

III. Essay

Directions : Read each question carefully, write your answer!

Symbolic AI

1. **How does Symbolic AI represent knowledge differently from neural network–based AI, and what are the implications of this difference?**

Symbolic AI represents knowledge using explicit symbols, rules, and logic — similar to how humans use language and reasoning. In contrast, neural networks store knowledge implicitly through weights and patterns in data. This means Symbolic AI is easier to interpret and explain but struggles with uncertainty or incomplete information, while neural networks excel at recognizing patterns but are often “black boxes.” The implication is that Symbolic AI offers transparency, whereas neural networks offer adaptability

2. **What challenges arise when trying to scale Symbolic AI systems to handle the complexity of real-world situations?**

Symbolic AI struggles with scalability because manually encoding every rule and relationship becomes impractical as systems grow more complex. Real-world situations often involve ambiguity, exceptions, and incomplete data — things that rule-based systems find hard to manage. Updating and maintaining massive rule databases also require constant human input, making Symbolic AI less efficient for dynamic or unpredictable environments.

3. **Can Symbolic AI effectively capture human commonsense reasoning, or are its rule-based systems too rigid?**

Symbolic AI can represent human reasoning through explicit rules and logical structures, but it struggles to fully capture the flexibility and context sensitivity of human commonsense thinking. Humans often rely on intuition, experience, and incomplete information, while Symbolic AI depends on predefined symbols and logical rules.

Because of this, symbolic systems can become too rigid — they work well in structured environments (like mathematics or expert systems) but fail when facing ambiguous or uncertain real-world situations.

4. **In what ways does Symbolic AI differ from machine learning–based AI?**

Symbolic AI and machine learning–based AI differ mainly in how they represent and process knowledge. Symbolic AI relies on explicit rules, logic, and symbols defined by humans to represent knowledge and reasoning. It focuses on clear, rule-based decision-making, where every step can be explained and traced. In contrast, machine learning–based AI learns patterns directly from data using algorithms and statistical models. It does not require hand-coded rules but instead improves its performance automatically as it processes more information.

While Symbolic AI offers strong explainability and logical reasoning, it struggles with flexibility and learning from raw data. Machine learning, on the other hand, excels at handling complex, uncertain, or unstructured data—such as images and speech—but often lacks transparency in its decision-making. Together, they represent two complementary approaches to achieving intelligent behavior in machines.

Software AI

1. **How does the quality of data used in software-based AI systems influence the outcomes and decisions those systems make?**

The quality of data directly determines how accurately a software-based AI can learn patterns and make predictions. If the data is biased, incomplete, or inaccurate, the AI will produce flawed results — even if its algorithms are well-designed. High-quality data ensures that AI systems can make fair, reliable, and consistent decisions. Essentially, “garbage in, garbage out” applies: the performance of AI software is only as good as the data it’s trained on.

2. **In what ways can software AI be designed to ensure ethical decision-making and reduce algorithmic bias?**

Software AI can be made more ethical by implementing transparent algorithms, using diverse and representative datasets, and involving human oversight in decision-making. Developers can include fairness constraints, accountability mechanisms, and explainability tools to ensure users understand how the AI reaches its conclusions. Regular audits and ethical frameworks are also essential to minimize unintended bias or discrimination in AI behavior.

3. **Why is software important in Artificial Intelligence systems?**

Software is important because it turns AI theories and algorithms into working programs that machines can run. It connects the thinking part (algorithms) with the physical part (hardware). Without software, AI could not learn, reason, or solve problems effectively.

4. **How does AI software differ from traditional software programs?**

AI software differs from traditional software programs in its ability to learn, adapt, and make intelligent decisions. While traditional software follows fixed rules and instructions set by programmers, AI software is designed to learn from data and improve its performance over time. Traditional programs produce predictable outputs based on specific inputs, but AI systems can analyze patterns, make predictions, and adjust their behavior without explicit human intervention. This adaptability allows AI to handle complex tasks such as image recognition, natural language processing, and decision-making that go beyond rule-based automation. In essence, traditional software executes predefined logic, whereas AI software evolves intelligently through continuous learning and data processing.

Software and Symbolic AI

1. **How can combining Symbolic AI with machine learning help bridge the gap between logical reasoning and pattern recognition in AI systems?**

Combining the two approaches — a method often called *neuro-symbolic AI* — allows systems to use both human-like reasoning (from Symbolic AI) and data-driven learning (from machine learning). Symbolic AI provides structure, logic, and explainability, while machine learning adds flexibility and adaptability. Together, they can reason more effectively, learn from new data, and explain their decisions — a powerful combination for complex problem-solving.

2. **What might the future of AI look like if both Software AI and Symbolic AI were integrated into a single unified system**

A unified AI system that blends Software AI and Symbolic AI could achieve human-like understanding and reasoning. It would not only learn from data but also explain its thought process clearly. Such systems could revolutionize fields like medicine, law, and education — where decisions require both logic and empathy. However, challenges in integration, computational cost, and ethics must be addressed before this becomes a widespread reality.

3. **In what ways does the software design of a Symbolic AI system affect its reasoning speed and accuracy?**

Efficient data structures and rule organization can speed up reasoning by reducing the time needed to search through rules and facts. A well-optimized inference engine improves logical processing and avoids redundant steps.

4. **How does software enable Symbolic AI systems to represent knowledge using symbols and rules?**

Software enables Symbolic AI systems to represent knowledge by providing tools to create and process symbols and logical rules. It uses programming languages like LISP or Prolog to define relationships, store symbolic data, and apply “if–then” logic for reasoning. Through components such as knowledge bases and inference engines, software allows AI to manipulate symbols, draw conclusions, and simulate human-like thinking. In this way, software serves as the foundation that turns symbolic representations into intelligent, rule-based decision-making.