

Artificial Intelligence An Introduction¹

Instructor: Dr. B. John Oommen

Chancellor's Professor

Fellow: IEEE; Fellow: IAPR

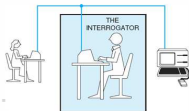
School of Computer Science, Carleton University, Canada.

¹The primary source of these notes are the slides of Professor Hwee Tou Ng from Singapore. I sincerely thank him for this.

History of AI

- Leibniz, Babbage, Boole, Frege, Russell, Tarski ...
- Turing (1930's)
 - Turing Machine (TM)
 - **Turing Test** “Operationalizing” Intelligence
 - Machine’s ability to demonstrate intelligence
 - Human Judge “converses” with human and machine
 - BOTH try to appear human
 - All participants are placed in isolated locations
 - If Judge cannot reliably tell the machine from the human, the machine **Passes** the test

- The Test itself



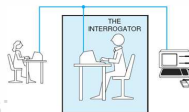
- Turing-Church Thesis:

If a problem is not solvable by a TM, it is not solvable by people either

History of AI

- Leibniz, Babbage, Boole, Frege, Russell, Tarski ...
- Turing (1930's)
 - Turing Machine (TM)
 - **Turing Test** “Operationalizing” Intelligence
 - Machine’s ability to demonstrate intelligence
 - Human Judge “converses” with human and machine
 - BOTH try to appear human
 - All participants are placed in isolated locations
 - If Judge cannot reliably tell the machine from the human, the machine **Passes** the test

- The Test itself



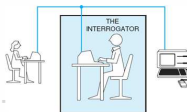
- Turing-Church Thesis:

If a problem is not solvable by a TM, it is not solvable by people either

History of AI

- Leibniz, Babbage, Boole, Frege, Russell, Tarski ...
- Turing (1930's)
 - Turing Machine (TM)
 - **Turing Test** “Operationalizing” Intelligence
 - Machine’s ability to demonstrate intelligence
 - Human Judge “converses” with human and machine
 - BOTH try to appear human
 - All participants are placed in isolated locations
 - If Judge cannot reliably tell the machine from the human, the machine **Passes** the test

- **The Test itself**



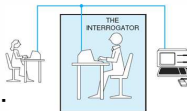
- Turing-Church Thesis:

If a problem is not solvable by a TM, it is not solvable by people either

History of AI

- Leibniz, Babbage, Boole, Frege, Russell, Tarski ...
- Turing (1930's)
 - Turing Machine (TM)
 - **Turing Test** “Operationalizing” Intelligence
 - Machine’s ability to demonstrate intelligence
 - Human Judge “converses” with human and machine
 - BOTH try to appear human
 - All participants are placed in isolated locations
 - If Judge cannot reliably tell the machine from the human, the machine **Passes** the test

- The Test itself



- **Turing-Church Thesis:**

If a problem is not solvable by a TM, it is not solvable by people either

History of AI: 1940's

- 1940s: McCulloch-Pitts, Wiener, Ashby
 - Neuron models
 - Cybernetics - Feedback
 - Teleological behavior
 - Study of design and purpose
 - All things to be designed for or directed toward a final result
 - There is an inherent purpose or final cause for all that exists
 - Homeostat
 - Device built by Ashby in 1948
 - Adaptive ultrastable system from four bomb control units
 - Had inputs and feedback
 - Used magnetically-driven water-filled potentiometers
 - Stabilizes effects of disturbances introduced into the system
 - *Time*: "Closest thing to a synthetic brain... designed by man"

History of AI: 1940's

- 1940s: McCulloch-Pitts, Wiener, Ashby
 - Neuron models
 - Cybernetics - Feedback
 - Teleological behavior
 - Study of design and purpose
 - All things to be designed for or directed toward a final result
 - There is an inherent purpose or final cause for all that exists
 - Homeostat
 - Device built by Ashby in 1948
 - Adaptive ultrastable system from four bomb control units
 - Had inputs and feedback
 - Used magnetically-driven water-filled potentiometers
 - Stabilizes effects of disturbances introduced into the system
 - *Time*: "Closest thing to a synthetic brain... designed by man"

History of AI: 1940's

- 1940s: McCulloch-Pitts, Wiener, Ashby
 - Neuron models
 - Cybernetics - Feedback
 - Teleological behavior
 - Study of design and purpose
 - All things to be designed for or directed toward a final result
 - There is an inherent purpose or final cause for all that exists
 - Homeostat
 - Device built by Ashby in 1948
 - Adaptive ultrastable system from four bomb control units
 - Had inputs and feedback
 - Used magnetically-driven water-filled potentiometers
 - Stabilizes effects of disturbances introduced into the system
 - *Time*: "Closest thing to a synthetic brain... designed by man"

History of AI: 1940's

- 1940s: Walter, von Neumann
 - Machina Speculatrix (Elmer 1948, and Elsie 1949)
 - First electronic autonomous robots
 - Rich connections between a small number of brain cells -
Very complex behaviors
 - Described as tortoises due to their shape and slow motion
 - “Taught us” about the secrets of organization and life
 - Three-wheeled tortoise robots
 - Could find their way to a recharging station
 - Self-reproducing automata
 - Self-replication: Process by which a thing copies of itself
 - Self-reproductive systems: Produce copies of themselves
 - Primitives: From metal bar and wire
 - Self-assembling systems
 - Assemble copies of themselves from finished parts
 - Self-reproducing “computer programs”

History of AI: 1940's

- 1940s: Walter, von Neumann
 - Machina Speculatrix (Elmer 1948, and Elsie 1949)
 - First electronic autonomous robots
 - Rich connections between a small number of brain cells - Very complex behaviors
 - Described as tortoises due to their shape and slow motion
 - “Taught us” about the secrets of organization and life
 - Three-wheeled tortoise robots
 - Could find their way to a recharging station
 - Self-reproducing automata
 - Self-replication: Process by which a thing copies of itself
 - Self-reproductive systems: Produce copies of themselves
 - Primitives: From metal bar and wire
 - Self-assembling systems
 - Assemble copies of themselves from finished parts
 - Self-reproducing “computer programs”

History of AI: 1950's

- 1950s: Simon, Newell, McCarthy, Minsky: “AI” (1956)
 - Fundamentals of Classification
 - Neural networks
 - Perceptron

History of AI: Since 1960's

- 1960s: Lisp, Adaline, Fuzzy sets (Zadeh 65)
- 1960s: General Problem Solver (GPS), Logic Theory
- 1970s: Backpropagation, Fuzzy Controllers
- 1970s: Knowledge Engineering, Genetic Algorithms (GA)
- 1970s: Production systems, Expert systems
- 1970s: Natural Language Processing (NLP)
- SHRDLU
 - SHRDLU was an early NLP developed by Winograd at MIT
 - Micro Planner and Lisp programming language on a PDP-6
 - SHRDLU was derived from ETAOIN SHRDLU
 - Arrangement of the alpha keys on a Linotype machine in descending frequency order

History of AI: Since 1960's

- 1960s: Lisp, Adaline, Fuzzy sets (Zadeh 65)
- 1960s: General Problem Solver (GPS), Logic Theory
- 1970s: Backpropagation, Fuzzy Controllers
- 1970s: Knowledge Engineering, Genetic Algorithms (GA)
- 1970s: Production systems, Expert systems
- 1970s: Natural Language Processing (NLP)
- SHRDLU
 - SHRDLU was an early NLP developed by Winograd at MIT
 - Micro Planner and Lisp programming language on a PDP-6
 - SHRDLU was derived from ETAOIN SHRDLU
 - Arrangement of the alpha keys on a Linotype machine in descending frequency order

History of AI: 1970's & 1980's

- 1970s: Theorem proving, Planning
- 1980s: NN / Connectionist boom, Boltzmann Machine
- 1980s: Knowledge Representation (KR)
- 1980s: More semantics in NLP (Conceptual Dependency)
- 1980s: Symbolic Machine Learning (ML)

History of AI: Since 1990's

- More NN
- Subsumption Architecture (Brooks)
 - Decompose complicated intelligent behaviour
 - Many "simple" behaviour modules organized into layers
 - Each layer implements a particular goal
 - Higher layers are increasingly abstract
 - A robot's layers:
 - "What layer could be "light or object"?
 - "On top of the robot is the level "under ground"?
 - "What wheels in turn they under "control the world"?
 - Uses a bottom-up design

History of AI: Since 1990's

- More NN
- Subsumption Architecture (Brooks)
 - Decompose complicated intelligent behaviour
 - Many “simple” behaviour modules organized into layers
 - Each layer implements a particular goal
 - Higher layers are increasingly abstract
 - A robot's layers:
 - Lowest layer could be “avoid an object”
 - On top of it would be the layer “wander around”
 - Which in turn lies under “explore the world”
 - Uses a bottom-up design

History of AI: Since 1990's

- Reinforcement Learning
- Bayesian Belief Nets
- Data Mining
- More NN, More GA, GP, Artificial-Life
- More GAs, Genetic Programming (GP), Artificial-Life
- “Bottom-up or behavior-based AI” vs “Top-down AI”
- “Emergent Computing”, Swarm Intelligence
- Self-Organization...

History of AI: Since 1990's

- Reinforcement Learning
- Bayesian Belief Nets
- Data Mining
- More NN, More GA, GP, Artificial-Life
- More GAs, Genetic Programming (GP), Artificial-Life
- “Bottom-up or behavior-based AI” vs “Top-down AI”
- “Emergent Computing”, Swarm Intelligence
- Self-Organization...

What is Intelligence

- **Intelligence is:**
 - Intellectual (?) behavior that we admire
 - But don't understand
 - Intelligence is manifested in **behavior**
 - **Closely related to surviving in a complex world**
 - Or ...

“2” kinds of AI (or 3 or 4)

- Engineering vs “Cognitive Science”
 - Making usefully smart machines, somehow:
 - Expert systems; Deep Blue; some Data Mining
 - Understanding *how* minds work
 - AI to express and test psychological/linguistic etc. theories

Kinds of AI

- Classical/Top-down / Symbolic vs Behavior-based / Bottom-up / Subsymbolic Mind vs Brain
 - “Physical symbol system hypothesis”
 - Hi-level approach is *brittle*
 - Bottom-up approach often unimpressive
- *Scruffies* vs *Neats*

Kinds of AI

- Classical/Top-down / Symbolic vs Behavior-based / Bottom-up / Subsymbolic Mind vs Brain
 - “Physical symbol system hypothesis”
 - Hi-level approach is *brittle*
 - Bottom-up approach often unimpressive
- *Scruffies* vs *Neats*

Kinds of AI

- Weak AI vs Strong AI
 - Chinese Room argument (John Searle)
 - No such things as AI...
 - An experiment: Someone who knows only English
 - Sits alone in a room following English instructions for manipulating strings of Chinese characters
 - To those outside the room it appears as if someone in the room understands Chinese.
 - Shows that while computers may *appear* to converse in natural language, they cannot – even in principle.
 - Searle argues that computers merely use syntactic rules to manipulate symbol strings
 - Have no understanding of meaning or semantics.

AI: All about Tradeoffs

Theoretical insights in AI: Concern tradeoffs

- **Tradeoffs: Efficiency and Generality**
- Tradeoffs: Robustness and Power
- Tradeoffs: Design complexity - Ability to degrade gracefully
- Tradeoffs: Prior cooking and Achievement
- Tradeoffs: Memory and Inference
- **Above All Tradeoffs: Memory and Time**

AI: All about Tradeoffs

Theoretical insights in AI: Concern tradeoffs

- Tradeoffs: Efficiency and Generality
- **Tradeoffs: Robustness and Power**
- Tradeoffs: Design complexity - Ability to degrade gracefully
- Tradeoffs: Prior cooking and Achievement
- Tradeoffs: Memory and Inference
- **Above All Tradeoffs: Memory and Time**

AI: All about Tradeoffs

Theoretical insights in AI: Concern tradeoffs

- Tradeoffs: Efficiency and Generality
- Tradeoffs: Robustness and Power
- **Tradeoffs: Design complexity - Ability to degrade gracefully**
- Tradeoffs: Prior cooking and Achievement
- Tradeoffs: Memory and Inference
- **Above All Tradeoffs: Memory and Time**

AI: All about Tradeoffs

Theoretical insights in AI: Concern tradeoffs

- Tradeoffs: Efficiency and Generality
- Tradeoffs: Robustness and Power
- Tradeoffs: Design complexity - Ability to degrade gracefully
- **Tradeoffs: Prior cooking and Achievement**
- Tradeoffs: Memory and Inference
- **Above All Tradeoffs: Memory and Time**

AI: All about Tradeoffs

Theoretical insights in AI: Concern tradeoffs

- Tradeoffs: Efficiency and Generality
- Tradeoffs: Robustness and Power
- Tradeoffs: Design complexity - Ability to degrade gracefully
- Tradeoffs: Prior cooking and Achievement
- **Tradeoffs: Memory and Inference**
- **Above All Tradeoffs: Memory and Time**

AI: All about Tradeoffs

Theoretical insights in AI: Concern tradeoffs

- Tradeoffs: Efficiency and Generality
- Tradeoffs: Robustness and Power
- Tradeoffs: Design complexity - Ability to degrade gracefully
- Tradeoffs: Prior cooking and Achievement
- Tradeoffs: Memory and Inference
- **Above All Tradeoffs: Memory and Time**

AI must be Scruffy...

- Neatness is impossible in complex domains
- Complex domains: Structure that requires solutions
- Found by exploring branching paths in a search space
 - No. of branches is exponential function of path depth
- Any intelligent agent needs to find tricks and shortcuts
- Even in formally specified domains!
- Unless: Infinitely large and fast computers
- Good shortcuts cannot be worked out in advance
- They are not perfect - even in mathematics
- Shortcuts & laziness: Go hand in hand...
- Key to intelligence (Gauss 1..100)

AI must be *Scruffy*...

- Neatness is impossible in complex domains
- Complex domains: Structure that requires solutions
- Found by exploring branching paths in a search space
 - No. of branches is exponential function of path depth
- **Any intelligent agent needs to find tricks and shortcuts**
- **Even in formally specified domains!**
- **Unless: Infinitely large and fast computers**
- Good shortcuts cannot be worked out in advance
- They are not perfect - even in mathematics
- Shortcuts & laziness: Go hand in hand...
- **Key to intelligence (Gauss 1..100)**

AI must be Scruffy...

- Neatness is impossible in complex domains
- Complex domains: Structure that requires solutions
- Found by exploring branching paths in a search space
 - No. of branches is exponential function of path depth
- Any intelligent agent needs to find tricks and shortcuts
- Even in formally specified domains!
- Unless: Infinitely large and fast computers
- Good shortcuts cannot be worked out in advance
- They are not perfect - even in mathematics
- Shortcuts & laziness: Go hand in hand...
- Key to intelligence (Gauss 1..100)

The *Real World* is even Harder

- Lack of complete initial information
- Range of things to do is large (branching factor!)
- Search spaces are huge
- Things happen fast
- There are deadlines
- Rapidly accessible and executable heuristics
- Must be learned by trial and error (for example)
- Such heuristic rules are bound to be fallible
 - Overgeneralization
 - Poor observations, weak sensors
 - Errors in measurement
 - Inadequate concepts
 - Noise, environmental variance etc...

The *Real World* is even Harder

- Lack of complete initial information
- Range of things to do is large (branching factor!)
- Search spaces are huge
- Things happen fast
- There are deadlines
- **Rapidly accessible and executable heuristics**
- **Must be learned by trial and error (for example)**
- Such heuristic rules are bound to be **fallible**
 - Overgeneralization
 - Poor observations, weak sensors
 - Errors in measurement
 - Inadequate concepts
 - Noise, environmental variance etc...

The Real World is even Harder

- Lack of complete initial information
- Range of things to do is large (branching factor!)
- Search spaces are huge
- Things happen fast
- There are deadlines
- Rapidly accessible and executable heuristics
- Must be learned by trial and error (for example)
- Such heuristic rules are bound to be **fallible**
 - Overgeneralization
 - Poor observations, weak sensors
 - Errors in measurement
 - Inadequate concepts
 - Noise, environmental variance etc...

Problems with Heuristics

- Rules and facts should be **consistent**
 - Consistency is **undecidable**
 - (Approximate) Consistency **checking** is explosive
 - **Maintaining** consistency also explosive
- To **revise a belief**, you need
 - **Fallible** heuristics
 - Allow for finding related beliefs
 - Identifying and retracting underlying assumptions etc.
- A huge reason maintenance system won't do

Problems with Heuristics

- Rules and facts should be consistent
 - Consistency is undecidable
 - (Approximate) Consistency checking is explosive
 - Maintaining consistency also explosive
- To revise a belief, you need
 - Fallible heuristics
 - Allow for finding related beliefs
 - Identifying and retracting underlying assumptions etc.
- A huge reason maintenance system won't do

Problems with Heuristics

- Rules and facts should be consistent
 - Consistency is undecidable
 - (Approximate) Consistency checking is explosive
 - Maintaining consistency also explosive
- To revise a belief, you need
 - Fallible heuristics
 - Allow for finding related beliefs
 - Identifying and retracting underlying assumptions etc.
- A huge reason maintenance system won't do

Semantics is *Scruffy* too

- Conceptual schemes: Open-ended
- Unlike formal languages
- There is no formal, recursive semantics for NL:
 - We don't know the extension-assigning functions!
- Concepts:
 - May be indeterminate, vague, or ambiguous
 - Prompt conceptual innovations
 - Empirical concepts: No crisp necess./suff. conditions
 - Many concepts are *theoretical*

Semantics is *Scruffy* too

- Conceptual schemes: Open-ended
- Unlike formal languages
- There is no formal, recursive semantics for NL:
 - We don't know the extension-assigning functions!
- Concepts:
 - May be indeterminate, vague, or ambiguous
 - Prompt conceptual innovations
 - Empirical concepts: No crisp necess./suff. conditions
 - Many concepts are **theoretical**

Scruffiness is *Inevitable*

- Scruffiness *Inevitable* for any resource-limited being!
- No practical strategy to reduce scruffiness works **always**
- AI must be scruffy, for neat reasons
- Thus: Study what the **history** has come up with
 - Of course: Theories about such inevitably scruffy systems
 - As **neat** as possible (maximally falsifiable etc.!!)

Scruffiness is *Inevitable*

- Scruffiness *Inevitable* for any resource-limited being!
- No practical strategy to reduce scruffiness works **always**
- **AI must be scruffy, for neat reasons**
- Thus: Study what the **history** has come up with
 - Of course: Theories about such inevitably scruffy systems
 - As **neat** as possible (maximally falsifiable etc.!!)

By the Way

- Nearly anything you want to compute you can't !!
 - Because there are countably many Turing machines
 - But *Uncountably* many functions
- The interesting things you can compute
 - Too expensive to compute
 - So, you can't compute them
 - Exponential worst case run-time functions
 - $T(n) = kC^n$ e.g. 1 input item takes 10^{-7} sec, $n=50$, complexity is 2^n : $20 * 10^{13}$ years
- Biological systems must use **approximate solutions**
 - Learning: On-line regularity detection for prediction
 - Experimentation and mental simulation
- "To be adaptable, an organism must be suboptimal"
(Gould)

By the Way

- Nearly anything you want to compute you can't !!
 - Because there are countably many Turing machines
 - But *Uncountably* many functions
- The interesting things you can compute
 - Too expensive to compute
 - So, you can't compute them
 - Exponential worst case run-time functions
 - $T(n) = kC^n$ e.g. 1 input item takes 10^{-7} sec, $n=50$, complexity is $2^n : 20 * 10^{13}$ years
- Biological systems must use **approximate solutions**
 - Learning: On-line regularity detection for prediction
 - Experimentation and mental simulation
- "To be adaptable, an organism must be suboptimal"
(Gould)

By the Way

- Nearly anything you want to compute you can't !!
 - Because there are countably many Turing machines
 - But *Uncountably* many functions
- The interesting things you can compute
 - Too expensive to compute
 - So, you can't compute them
 - Exponential worst case run-time functions
 - $T(n) = kC^n$ e.g. 1 input item takes 10^{-7} sec, $n=50$, complexity is 2^n : $20 * 10^{13}$ years
- Biological systems must use **approximate solutions**
 - **Learning: On-line regularity detection for prediction**
 - **Experimentation and mental simulation**
- "To be adaptable, an organism must be suboptimal"
(Gould)

By the Way

- Nearly anything you want to compute you can't !!
 - Because there are countably many Turing machines
 - But *Uncountably* many functions
- The interesting things you can compute
 - Too expensive to compute
 - So, you can't compute them
 - Exponential worst case run-time functions
 - $T(n) = kC^n$ e.g. 1 input item takes 10^{-7} sec, $n=50$, complexity is 2^n : $20 * 10^{13}$ years
- Biological systems must use **approximate solutions**
 - Learning: On-line regularity detection for prediction
 - Experimentation and mental simulation
- “To be adaptable, an organism must be suboptimal”
(Gould)

AI is **Highly Interdisciplinary**

Many fields have contributed to AI

- In the form of ideas, viewpoints and techniques
- **Philosophy**: Logic, reasoning, mind as a physical system
- **Mathematics**: Formal representation and proofs
- **Mathematics**: Computation, (un)decidability, (in)tractability
- **Mathematics**: Probability, fuzzy theory
- **Psychology**: Learning, perception, motor control
- **Economics**: Theory of rational decisions, game theory

AI is **Highly** Interdisciplinary

Other fields that have contributed to AI:

- **Linguistics**: Knowledge representation, grammar
- **Neuroscience**: Physical substrate for mental activities
- **Biology**: Adaptation, evolution of complex systems
- **Controls**: Homeostatic systems, stability, optimal agents
- **Complex Systems Theory** etc. etc. etc....

AI Systems

- Think like humans
 - Cognitive modelling (AI + Psychology)
- Act like humans
 - Turing test approach: needs NLP, KR, ML, ...
- Think rationally
 - First-Order-Logic based problem solving and planning
 - Closely related to automated theorem proving
- Act rationally
 - A rational agent acts so as to achieve its goals
 - Given its beliefs & **limited rationality**
- **Autonomous agents, robots, evolutionary computation**

Some Subareas of AI...

- Heuristic search
 - Problem solving, planning, game playing
- Theorem proving
- Knowledge-based (KB) systems
 - Knowledge Engineering (KE);
 - Knowledge Representation (KR); Expert systems
- Natural Language Processing (NLP)
 - Story understanding
 - Speech recognition
 - Question answering

Some Subareas of AI...

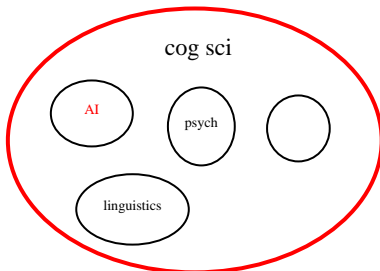
- Heuristic search
 - Problem solving, planning, game playing
- Theorem proving
- Knowledge-based (KB) systems
 - Knowledge Engineering (KE);
 - Knowledge Representation (KR); Expert systems
- Natural Language Processing (NLP)
 - Story understanding
 - Speech recognition
 - Question answering

Some Subareas of AI...

- Heuristic search
 - Problem solving, planning, game playing
- Theorem proving
- Knowledge-based (KB) systems
 - Knowledge Engineering (KE);
 - Knowledge Representation (KR); Expert systems
- **Natural Language Processing (NLP)**
 - Story understanding
 - Speech recognition
 - Question answering

Some Subareas of AI...

- Perception
- Vision
- Robotics
- Machine Learning
- Pattern Recognition



Intelligence is (?) Reasoning + Knowledge

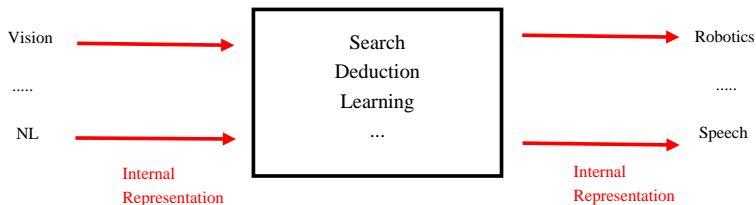
- Reasoning
 - Universal inference methods
 - “Weak” methods, e.g. hill climbing
 - Domain-independent search through symbolic state spaces
 - Problem-solving/planning **theorem proving - first principles**
- Knowledge
 - Universal methods → combinatorial explosion
 - “Strong” methods:
 - Heuristics
 - Domain-dependent knowledge
 - Shallow deductions
 - **Expert systems**

Intelligence is (?) Reasoning + Knowledge

- Reasoning
 - Universal inference methods
 - “Weak” methods, e.g. hill climbing
 - Domain-independent search through symbolic state spaces
 - Problem-solving/planning theorem proving - first principles
- Knowledge
 - Universal methods → combinatorial explosion
 - “Strong” methods:
 - Heuristics
 - Domain-dependent knowledge
 - Shallow deductions
 - Expert systems

Goal of AI

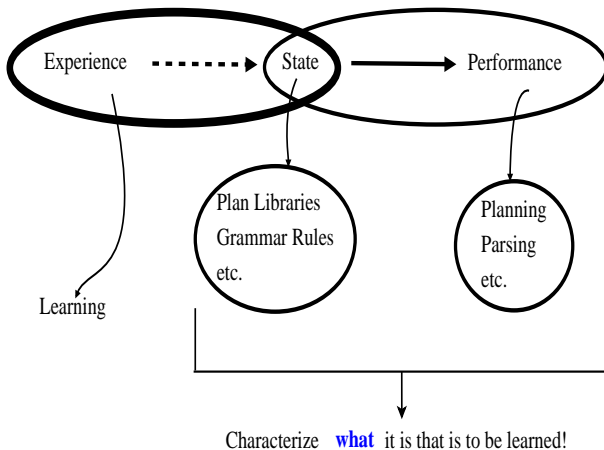
Build a person / animal



Internal representation:

- Not NL
- All representations inter-translatable
- Unambiguous, explicit referents, only gist remembered
- Support inferences

Why is AI not just “Learning”?



- To learn anything you should already “know” a lot
- Without strong clues of domain, nothing is learned
- There are many kinds of learning...