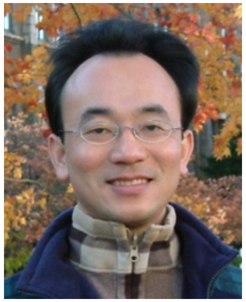
人工智慧
(Artificial Intelligence)
知識推理和知識表達
(Knowledge, Reasoning and Knowledge Representation)

1092AI04
MBA, IM, NTPU (M5010) (Spring 2021)
Wed 2, 3, 4 (9:10-12:00) (B8F40)



Min-Yuh Day
戴敏育
Associate Professor
副教授
Institute of Information Management, National Taipei University
國立臺北大學 資訊管理研究所
https://web.ntpu.edu.tw/~myday

2021-03-17

課程大綱 (Syllabus)

國立臺北大學
National Taipei University

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
1	2021/02/24	人工智慧概論 (Introduction to Artificial Intelligence)
2	2021/03/03	人工智慧和智慧代理人 (Artificial Intelligence and Intelligent Agents)
3	2021/03/10	問題解決 (Problem Solving)
4	2021/03/17	知識推理和知識表達 (Knowledge, Reasoning and Knowledge Representation)
5	2021/03/24	不確定知識和推理 (Uncertain Knowledge and Reasoning)
6	2021/03/31	人工智慧個案研究 I (Case Study on Artificial Intelligence I)

課程大綱 (Syllabus)

國立臺北大學 National Taipei University

週次 (Week) 日期 (Date) 內容 (Subject/Topics)

7 2021/04/07 放假一天 (Day off)

8 2021/04/14 機器學習與監督式學習
(Machine Learning and Supervised Learning)

9 2021/04/21 期中報告
(Midterm Project Report)

10 2021/04/28 學習理論與綜合學習
(The Theory of Learning and Ensemble Learning)

11 2021/05/05 深度學習
(Deep Learning)

12 2021/05/12 人工智慧個案研究 II
(Case Study on Artificial Intelligence II)

課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)

13 2021/05/19 強化學習 (Reinforcement Learning)

14 2021/05/26 深度學習自然語言處理 (Deep Learning for Natural Language Processing)

15 2021/06/02 機器人技術 (Robotics)

16 2021/06/09 人工智慧哲學與倫理，人工智慧的未來 (Philosophy and Ethics of AI, The Future of AI)

17 2021/06/16 期末報告 I (Final Project Report I)

18 2021/06/23 期末報告 II (Final Project Report II)

Knowledge, Reasoning and Knowledge Representation

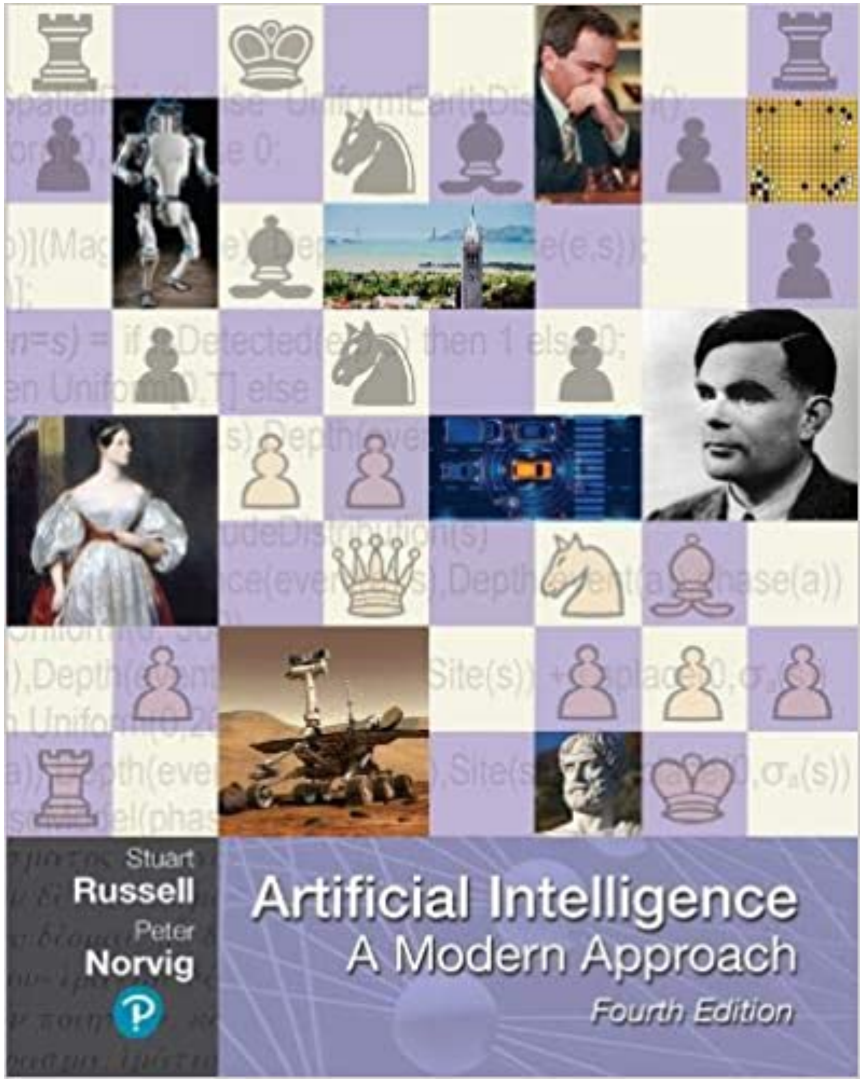
Outline

- Logical Agents
- First-Order Logic
- Inference in First-Order Logic
- Knowledge Representation
- Automated Planning

Stuart Russell and Peter Norvig (2020),

Artificial Intelligence: A Modern Approach,

4th Edition, Pearson



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

https://www.amazon.com/Artificial-Intelligence-A-Modern-Approach/dp/0134610997/

Artificial Intelligence: A Modern Approach

1. Artificial Intelligence
2. Problem Solving
3. Knowledge and Reasoning
4. Uncertain Knowledge and Reasoning
5. Machine Learning
6. Communicating, Perceiving, and Acting
7. Philosophy and Ethics of AI

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Artificial Intelligence:
Knowledge and Reasoning

Artificial Intelligence:
3. Knowledge and Reasoning

- Logical Agents
- First-Order Logic
- Inference in First-Order Logic
- Knowledge Representation
- Automated Planning

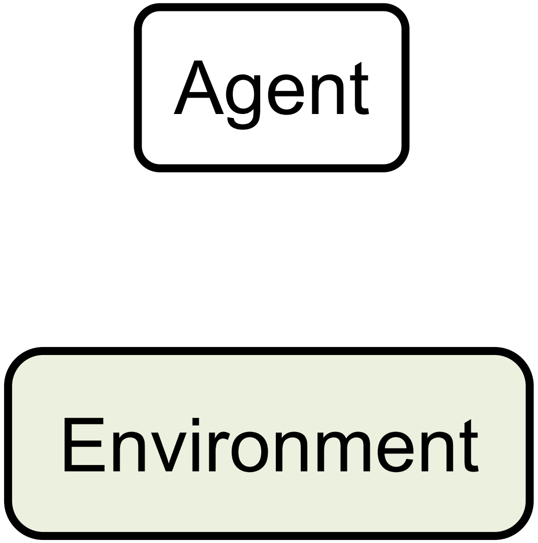
Intelligent Agents

4 Approaches of AI

2. Thinking Humanly: The Cognitive Modeling Approach	3. Thinking Rationally: The “Laws of Thought” Approach
1. Acting Humanly: The Turing Test Approach (1950)	4. Acting Rationally: The Rational Agent Approach

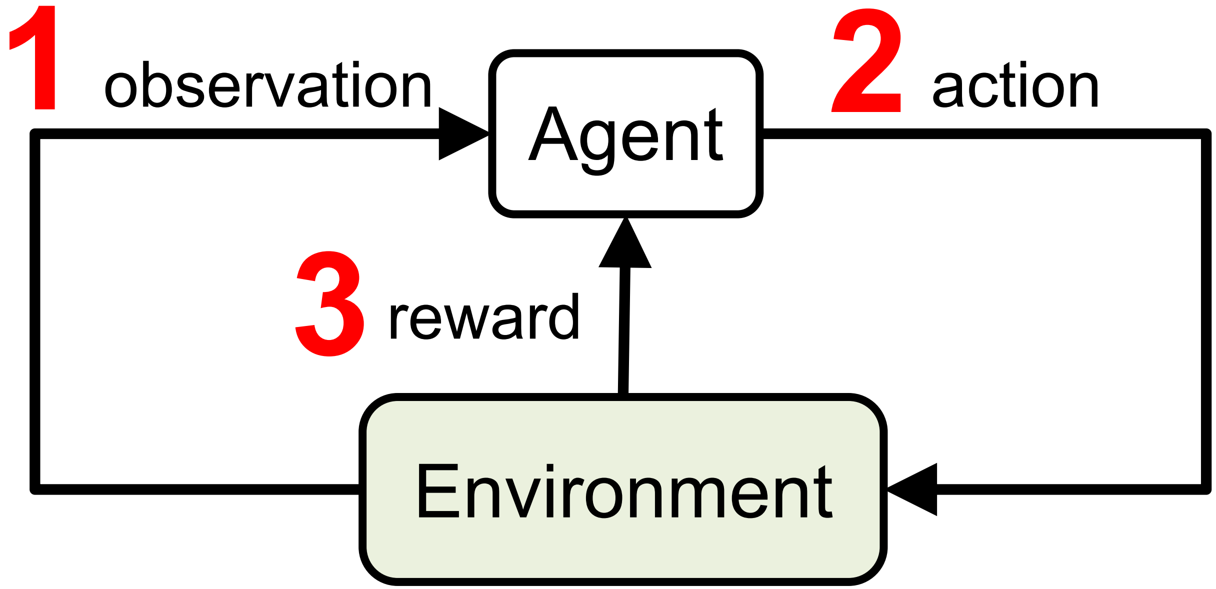
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Reinforcement Learning (DL)



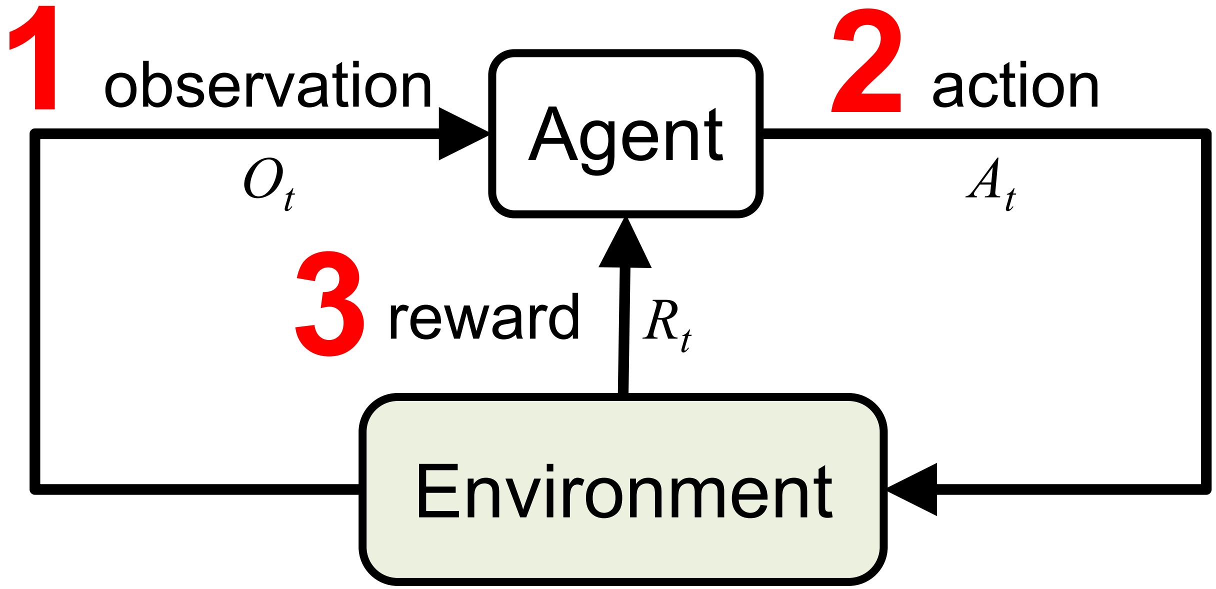
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford

Reinforcement Learning (DL)



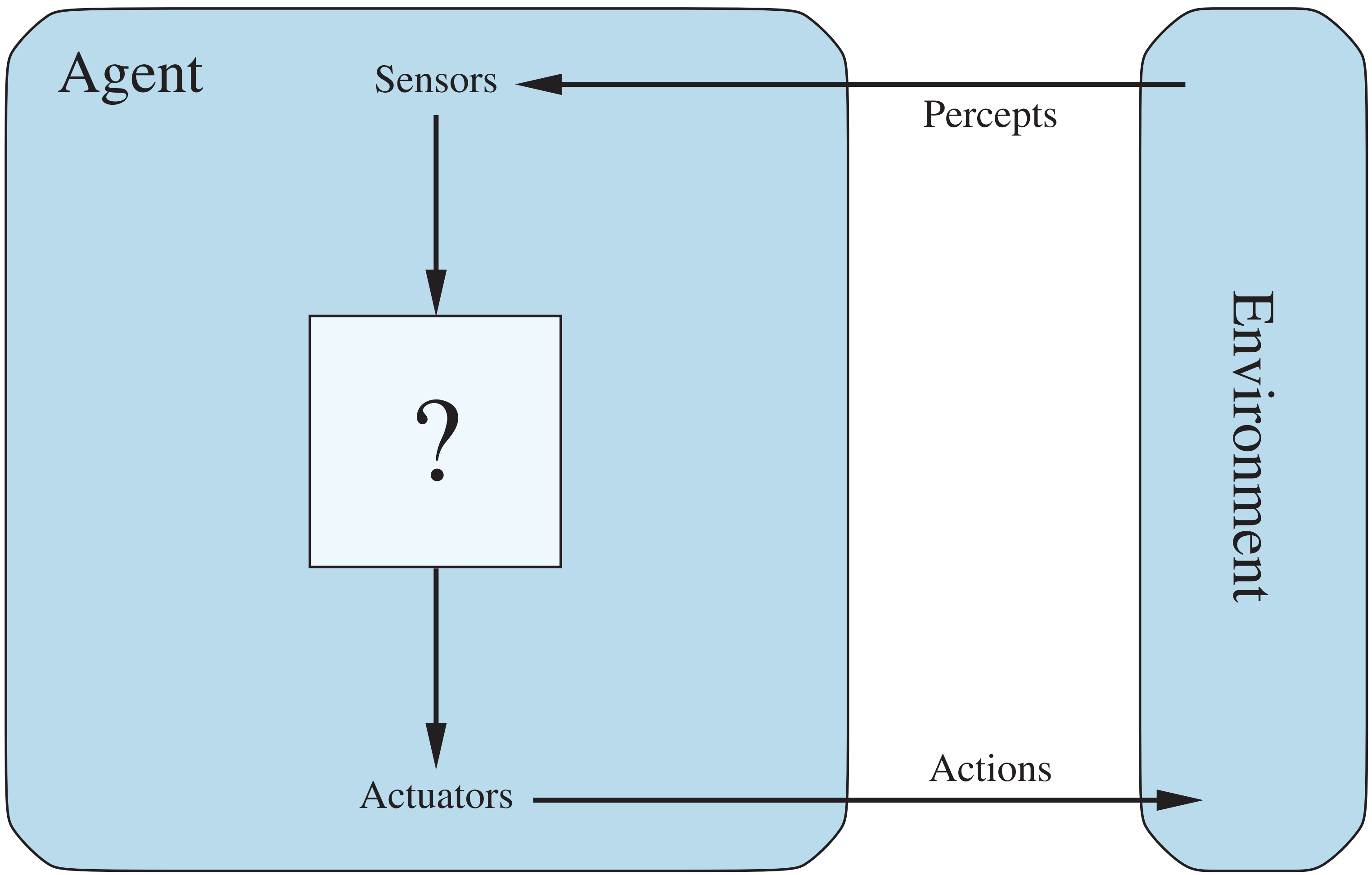
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford

Reinforcement Learning (DL)



Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford

Agents interact with environments through sensors and actuators



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Logical Agents

Logical Agents

Knowledge-based Agents

KB Agents

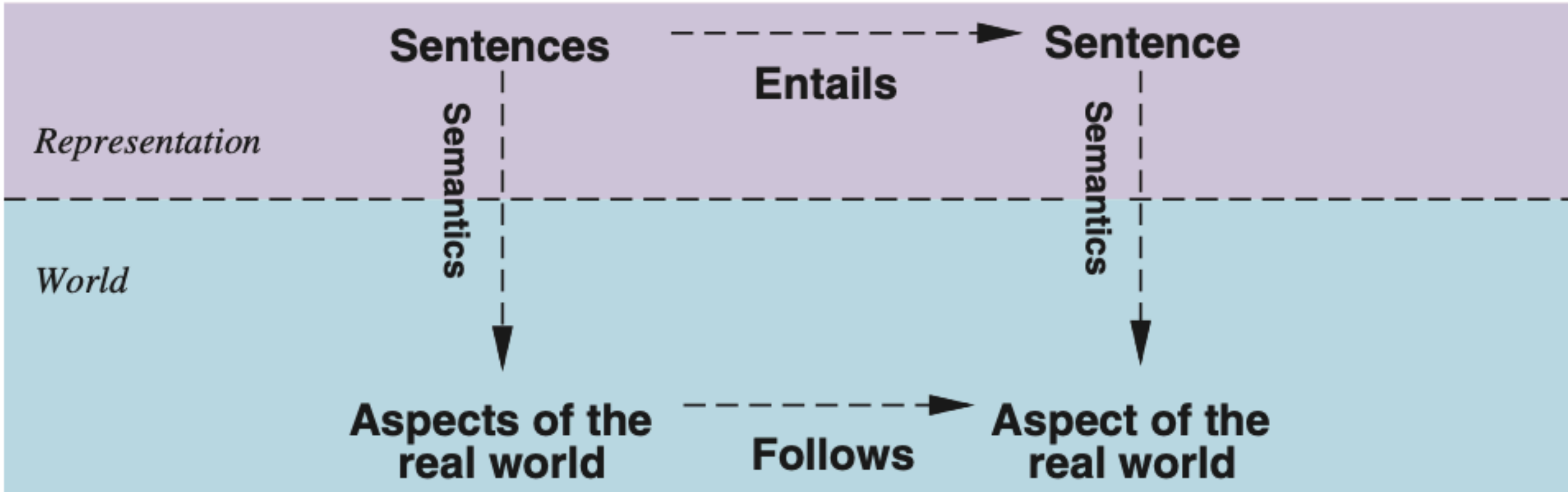
Knowledge-based Agent (KB Agent)

function KB-AGENT($percept$) **returns** an $action$
  persistent: KB, a knowledge base
    t, a counter, initially 0, indicating time

  TELL(KB, MAKE-PERCEPT-SENTENCE($percept$, t))
  $action \leftarrow$ ASK(KB, MAKE-ACTION-QUERY(t))
  TELL(KB, MAKE-ACTION-SENTENCE($action$, t))
  $t \leftarrow t + 1$
  return $action$

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Sentences are physical configurations of the agent



Reasoning is a process of constructing new physical configurations from old ones

Logical reasoning should ensure that the new configurations represent aspects of the world that actually follow from the aspects that the old configurations represent.

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A BNF (Backus–Naur Form) grammar of sentences in propositional logic

$$
\begin{array}{rcl}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} & \rightarrow & \mathit{True} \mid \mathit{False} \mid P \mid Q \mid R \mid \ldots \\
\mathit{ComplexSentence} & \rightarrow & (\ \mathit{Sentence}\ ) \\
& \mid & \neg\ \mathit{Sentence} \\
& \mid & \mathit{Sentence} \land \mathit{Sentence} \\
& \mid & \mathit{Sentence} \lor \mathit{Sentence} \\
& \mid & \mathit{Sentence} \;\Rightarrow\; \mathit{Sentence} \\
& \mid & \mathit{Sentence} \;\Leftrightarrow\; \mathit{Sentence}
\end{array}
$$

$$
\textsc{Operator Precedence} \quad : \quad \neg, \land, \lor, \Rightarrow, \Leftrightarrow
$$

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Truth Tables (TT) for the Five Logical Connectives

P	Q	$\neg P$	$P \land Q$	$P \lor Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A Truth Table constructed for the knowledge base given in the text

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*false*	*true*	*false*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A Truth-Table (TT) enumeration algorithm for deciding propositional entailment

```
function TT-ENTAILS?(KB, α) returns true or false
  inputs: KB, the knowledge base, a sentence in propositional logic
          α, the query, a sentence in propositional logic

  symbols ← a list of the proposition symbols in KB and α
  return TT-CHECK-ALL(KB, α, symbols, { })

function TT-CHECK-ALL(KB, α, symbols, model) returns true or false
  if EMPTY?(symbols) then
      if PL-TRUE?(KB, model) then return PL-TRUE?(α, model)
      else return true        //  when KB is false, always return true
  else
      P ← FIRST(symbols)
      rest ← REST(symbols)
      return (TT-CHECK-ALL(KB, α, rest, model ∪ {P = true})
              and
              TT-CHECK-ALL(KB, α, rest, model ∪ {P = false }))
```

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Standard Logical Equivalences

The symbols α, β, and γ stand for arbitrary sentences of propositional logic.

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) && \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) && \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) && \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) && \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) && \text{De Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) && \text{De Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) && \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) && \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A grammar for Conjunctive Normal Form (CNF), Horn clauses, and definite clauses

$$
\begin{aligned}
CNFSentence &\rightarrow Clause_1 \wedge \cdots \wedge Clause_n \\
Clause &\rightarrow Literal_1 \vee \cdots \vee Literal_m \\
Fact &\rightarrow Symbol \\
Literal &\rightarrow Symbol \mid \neg Symbol \\
Symbol &\rightarrow P \mid Q \mid R \mid \ldots \\
HornClauseForm &\rightarrow DefiniteClauseForm \mid GoalClauseForm \\
DefiniteClauseForm &\rightarrow Fact \mid (Symbol_1 \wedge \cdots \wedge Symbol_l) \Rightarrow Symbol \\
GoalClauseForm &\rightarrow (Symbol_1 \wedge \cdots \wedge Symbol_l) \Rightarrow False
\end{aligned}
$$

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A simple resolution algorithm for propositional logic

function PL-RESOLUTION(KB, α) **returns** *true* or *false*
  inputs: KB, the knowledge base, a sentence in propositional logic
    α, the query, a sentence in propositional logic

  clauses ← the set of clauses in the CNF representation of $KB \land \neg\alpha$
  new ← { }
  while *true* **do**
    for each pair of clauses C_i, C_j **in** *clauses* **do**
      resolvents ← PL-RESOLVE(C_i, C_j)
      if *resolvents* contains the empty clause **then return** *true*
      new ← *new* ∪ *resolvents*
    if *new* ⊆ *clauses* **then return** *false*
    clauses ← *clauses* ∪ *new*

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

The forward-chaining algorithm for propositional logic

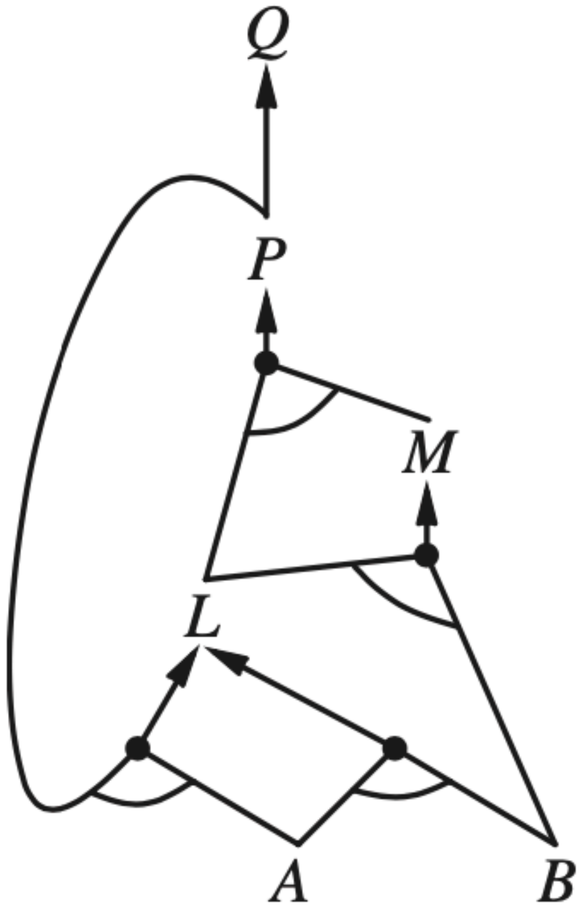
```
function PL-FC-ENTAILS?(KB, q) returns true or false
  inputs: KB, the knowledge base, a set of propositional definite clauses
          q, the query, a proposition symbol
  count ← a table, where count[c] is initially the number of symbols in clause c's premise
  inferred ← a table, where inferred[s] is initially false for all symbols
  queue ← a queue of symbols, initially symbols known to be true in KB

  while queue is not empty do
     p ← POP(queue)
     if p = q then return true
     if inferred[p] = false then
        inferred[p] ← true
        for each clause c in KB where p is in c.PREMISE do
           decrement count[c]
           if count[c] = 0 then add c.CONCLUSION to queue
  return false
```

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A set of Horn clauses

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$

(a)



(b)

The corresponding AND–OR graph

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

First-Order Logic

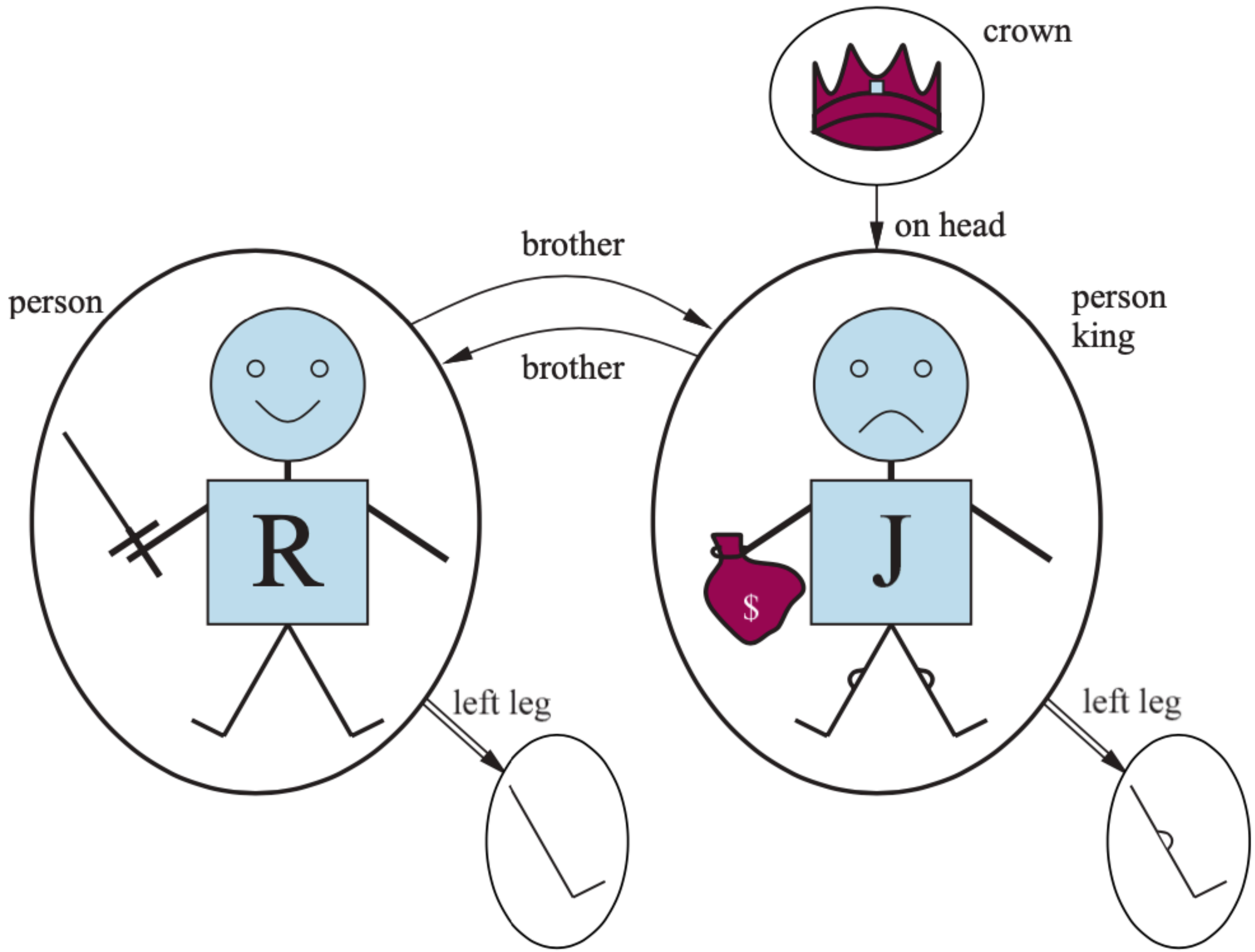
Formal languages and their ontological and epistemological commitments

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $\in [0, 1]$
Fuzzy logic	facts with degree of truth $\in [0, 1]$	known interval value

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A model containing five objects

two binary relations (brother and on-head), three unary relations (person, king, and crown), and one unary function (left-leg).



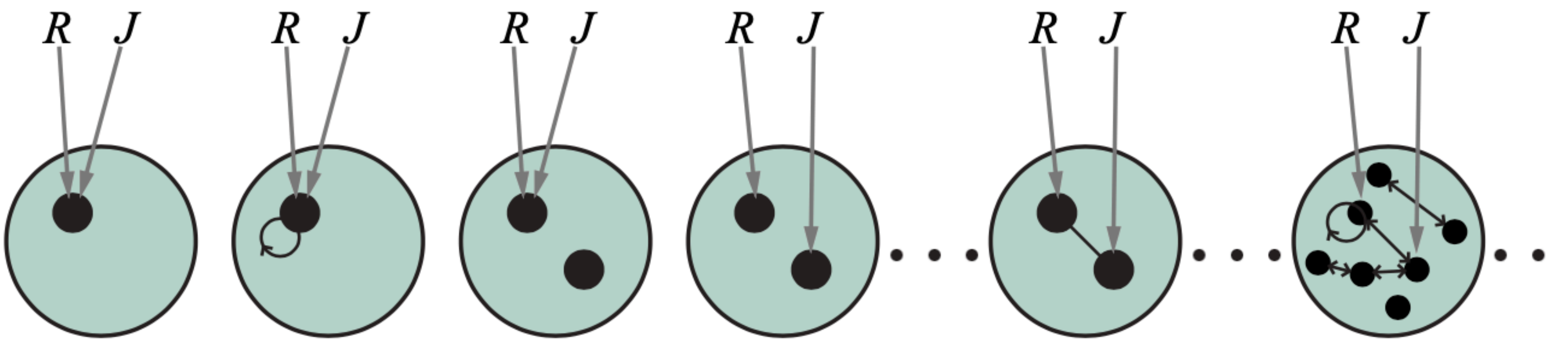
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

The syntax of first-order logic with equality

$$
\begin{array}{rcl}
Sentence & \rightarrow & AtomicSentence \mid ComplexSentence \\
AtomicSentence & \rightarrow & Predicate \mid Predicate(Term, \ldots) \mid Term = Term \\
ComplexSentence & \rightarrow & (\ Sentence\ ) \\
& \mid & \neg\ Sentence \\
& \mid & Sentence \land Sentence \\
& \mid & Sentence \lor Sentence \\
& \mid & Sentence \ \Rightarrow \ Sentence \\
& \mid & Sentence \ \Leftrightarrow \ Sentence \\
& \mid & Quantifier\ Variable, \ldots\ Sentence \\
\\
Term & \rightarrow & Function(Term, \ldots) \\
& \mid & Constant \\
& \mid & Variable \\
\\
Quantifier & \rightarrow & \forall \mid \exists \\
Constant & \rightarrow & A \mid X_1 \mid John \mid \cdots \\
Variable & \rightarrow & a \mid x \mid s \mid \cdots \\
Predicate & \rightarrow & True \mid False \mid After \mid Loves \mid Raining \mid \cdots \\
Function & \rightarrow & Mother \mid LeftLeg \mid \cdots \\
\\
\text{OPERATOR PRECEDENCE} & : & \neg, =, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

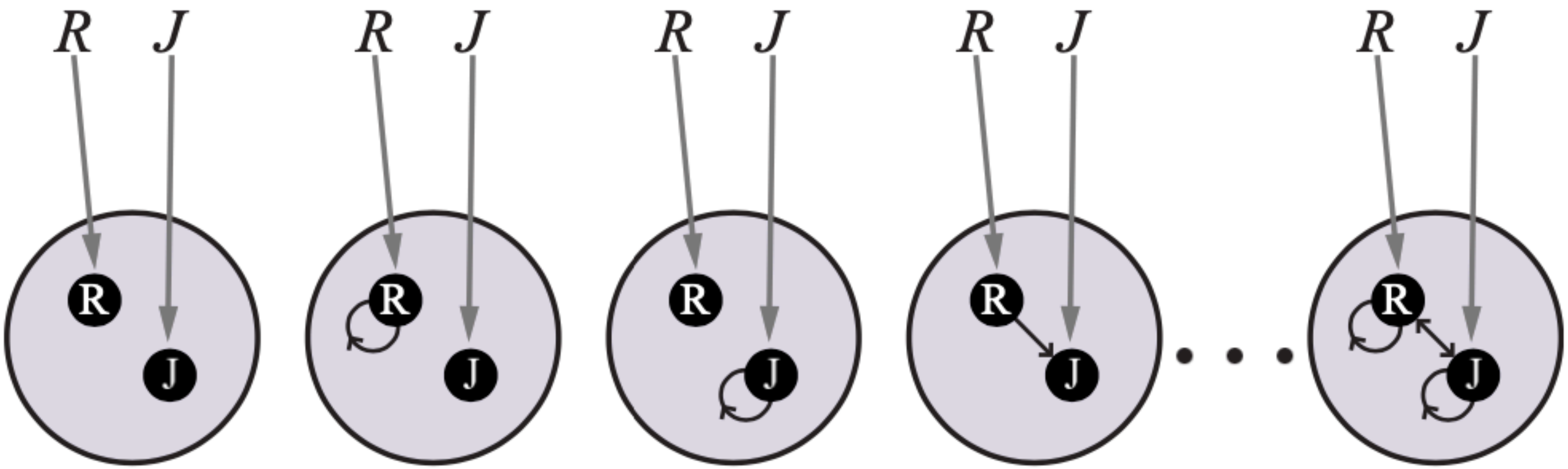
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Some members of the set of all models for a language with two constant symbols, R and J, and one binary relation symbol



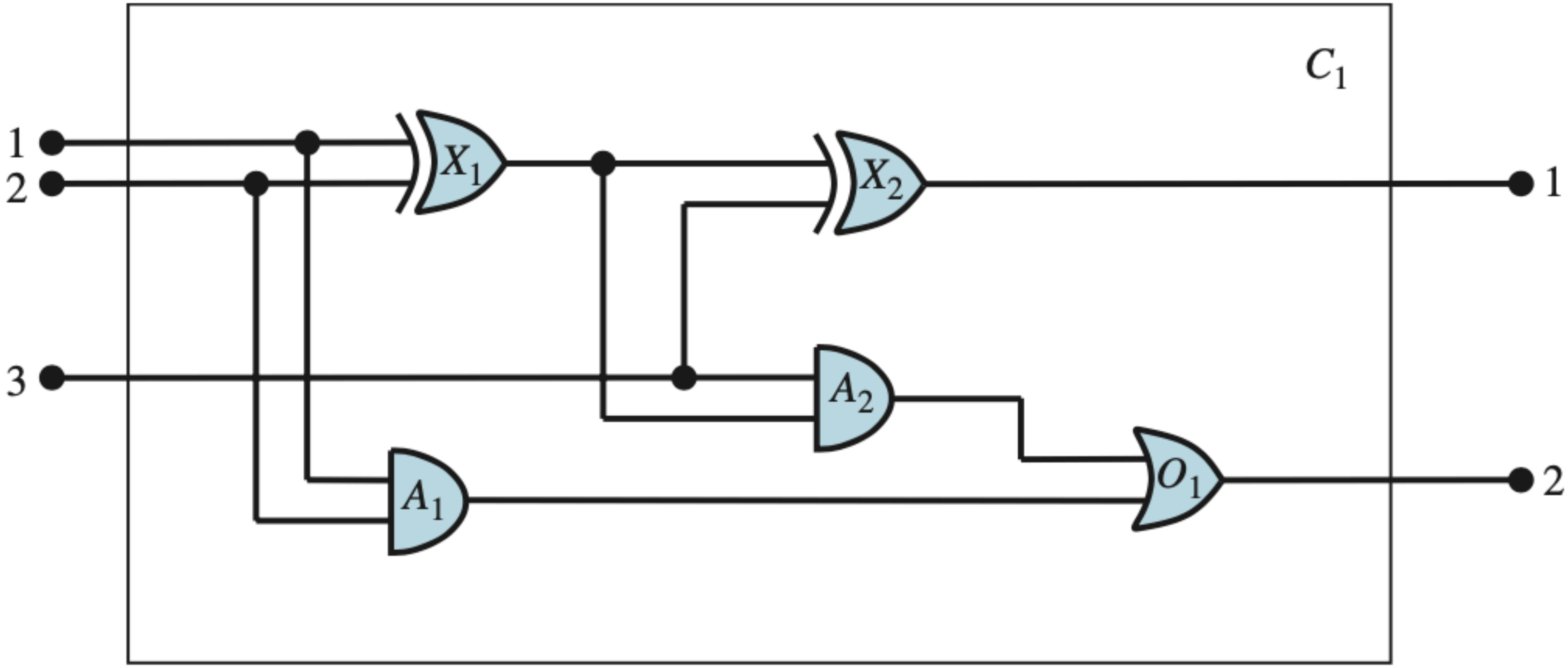
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Some members of the set of all models for a language with two constant symbols, R and J, and one binary relation symbol, under database semantics



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A digital circuit C1, purporting to be a one-bit full adder.



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Inference in First-Order Logic

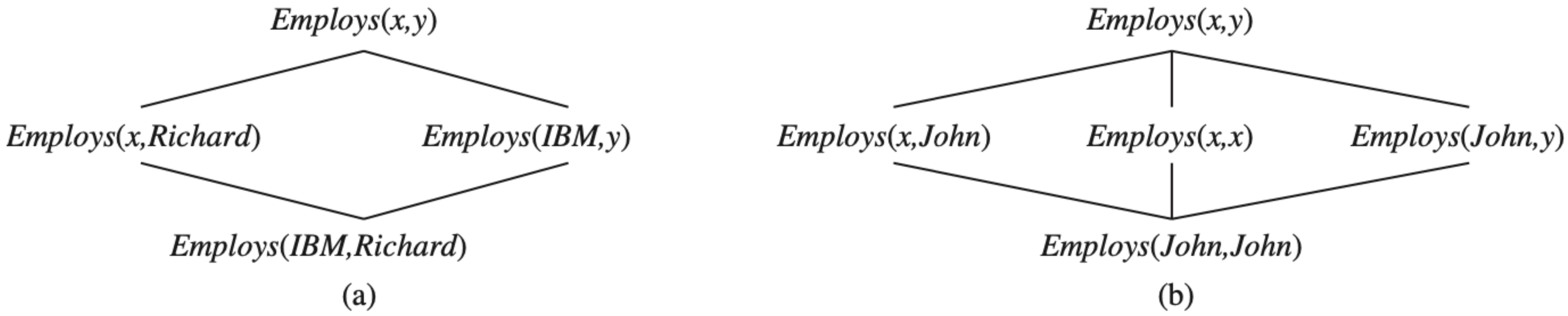
The unification algorithm

```
function UNIFY(x, y, θ=empty) returns a substitution to make x and y identical, or failure
   if θ = failure then return failure
   else if x = y then return θ
   else if VARIABLE?(x) then return UNIFY-VAR(x, y, θ)
   else if VARIABLE?(y) then return UNIFY-VAR(y, x, θ)
   else if COMPOUND?(x) and COMPOUND?(y) then
       return UNIFY(ARGS(x), ARGS(y), UNIFY(OP(x), OP(y), θ))
   else if LIST?(x) and LIST?(y) then
       return UNIFY(REST(x), REST(y), UNIFY(FIRST(x), FIRST(y), θ))
   else return failure

function UNIFY-VAR(var, x, θ) returns a substitution
   if {var/val} ∈ θ for some val then return UNIFY(val, x, θ)
   else if {x/val} ∈ θ for some val then return UNIFY(var, val, θ)
   else if OCCUR-CHECK?(var, x) then return failure
   else return add {var/x} to θ
```

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

The subsumption lattice whose lowest node is Employs (IBM , Richard)

The subsumption lattice for the sentence Employs (John, John)



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

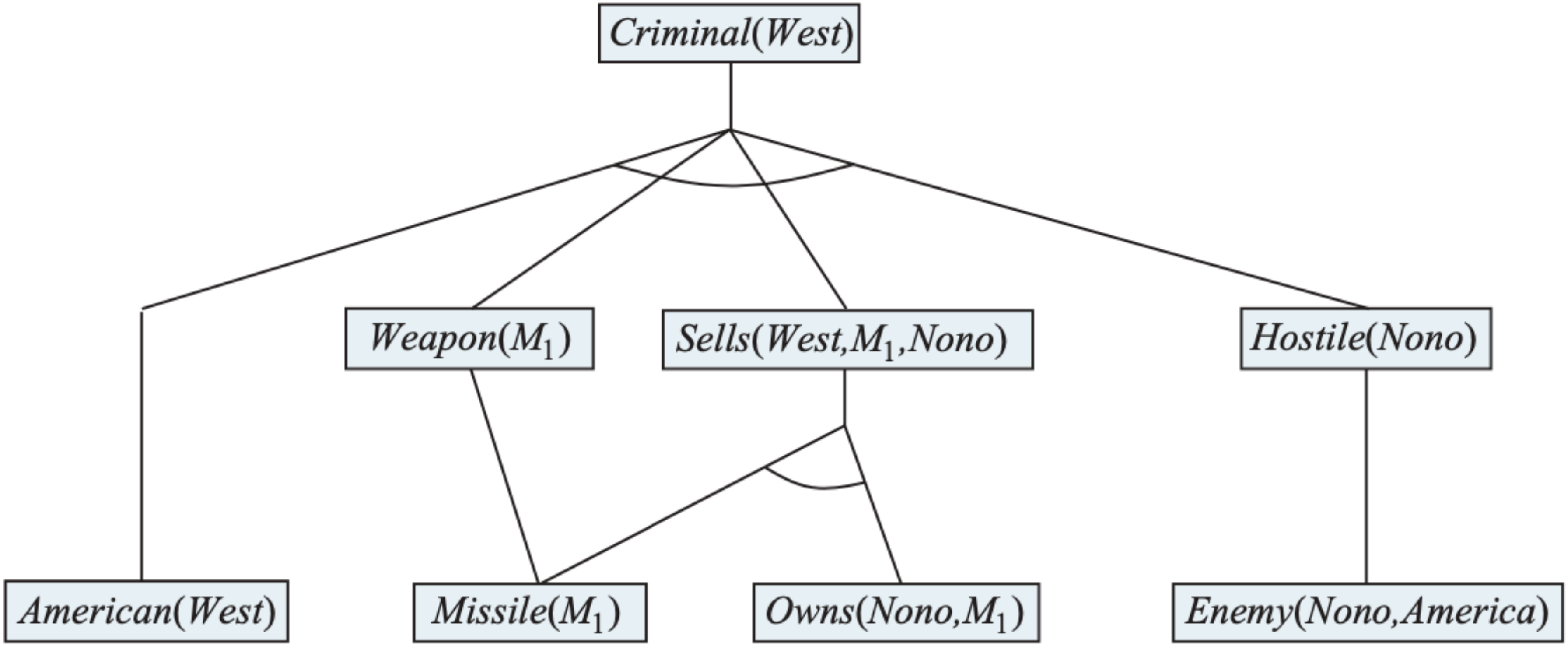
A conceptually straightforward, but inefficient, forward-chaining algorithm

function FOL-FC-ASK(KB, α) **returns** a substitution or *false*
  inputs: KB, the knowledge base, a set of first-order definite clauses
    α, the query, an atomic sentence

  while *true* **do**
    $new \leftarrow \{\}$ // *The set of new sentences inferred on each iteration*
    for each *rule* **in** KB **do**
      $(p_1 \wedge \ldots \wedge p_n \Rightarrow q) \leftarrow$ STANDARDIZE-VARIABLES(*rule*)
      for each θ such that SUBST($\theta, p_1 \wedge \ldots \wedge p_n$) = SUBST($\theta, p'_1 \wedge \ldots \wedge p'_n$)
          for some $p'_1, \ldots, p'_n$ in KB
        $q' \leftarrow$ SUBST(θ, q)
        if q' does not unify with some sentence already in KB or *new* **then**
          add q' to *new*
          $\phi \leftarrow$ UNIFY(q', α)
          if ϕ is not *failure* **then return** ϕ
    if $new = \{\}$ **then return** *false*
    add *new* to KB

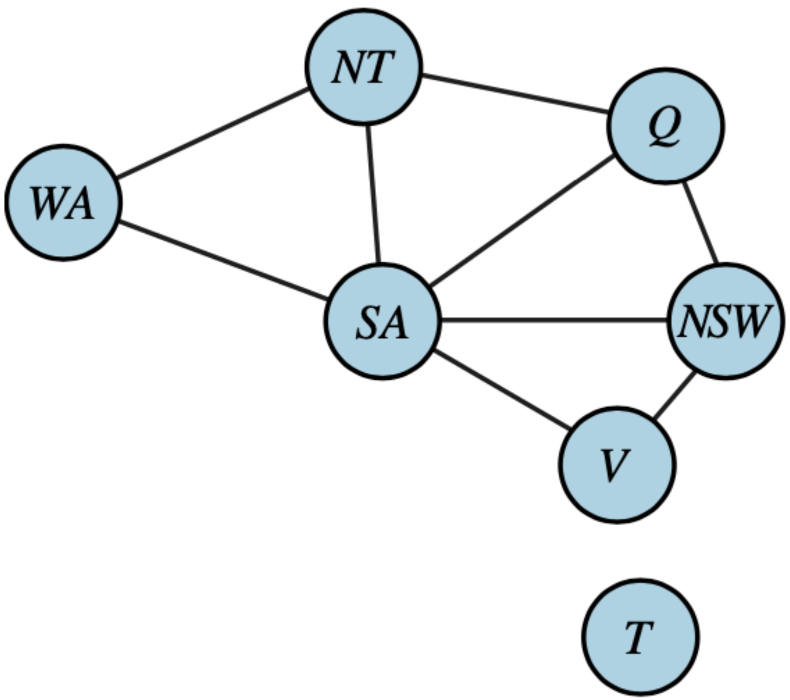
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

The proof tree generated by forward chaining on the crime example



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Constraint graph for coloring the map of Australia



(a)

$$
\begin{aligned}
& \mathit{Diff}(wa, nt) \wedge \mathit{Diff}(wa, sa) \wedge \\
& \quad \mathit{Diff}(nt, q) \wedge \mathit{Diff}(nt, sa) \wedge \\
& \quad \mathit{Diff}(q, nsw) \wedge \mathit{Diff}(q, sa) \wedge \\
& \quad \mathit{Diff}(nsw, v) \wedge \mathit{Diff}(nsw, sa) \wedge \\
& \quad \mathit{Diff}(v, sa) \Rightarrow \mathit{Colorable}()
\end{aligned}
$$

$$
\begin{array}{ll}
\mathit{Diff}(Red, Blue) & \mathit{Diff}(Red, Green) \\
\mathit{Diff}(Green, Red) & \mathit{Diff}(Green, Blue) \\
\mathit{Diff}(Blue, Red) & \mathit{Diff}(Blue, Green)
\end{array}
$$

(b)

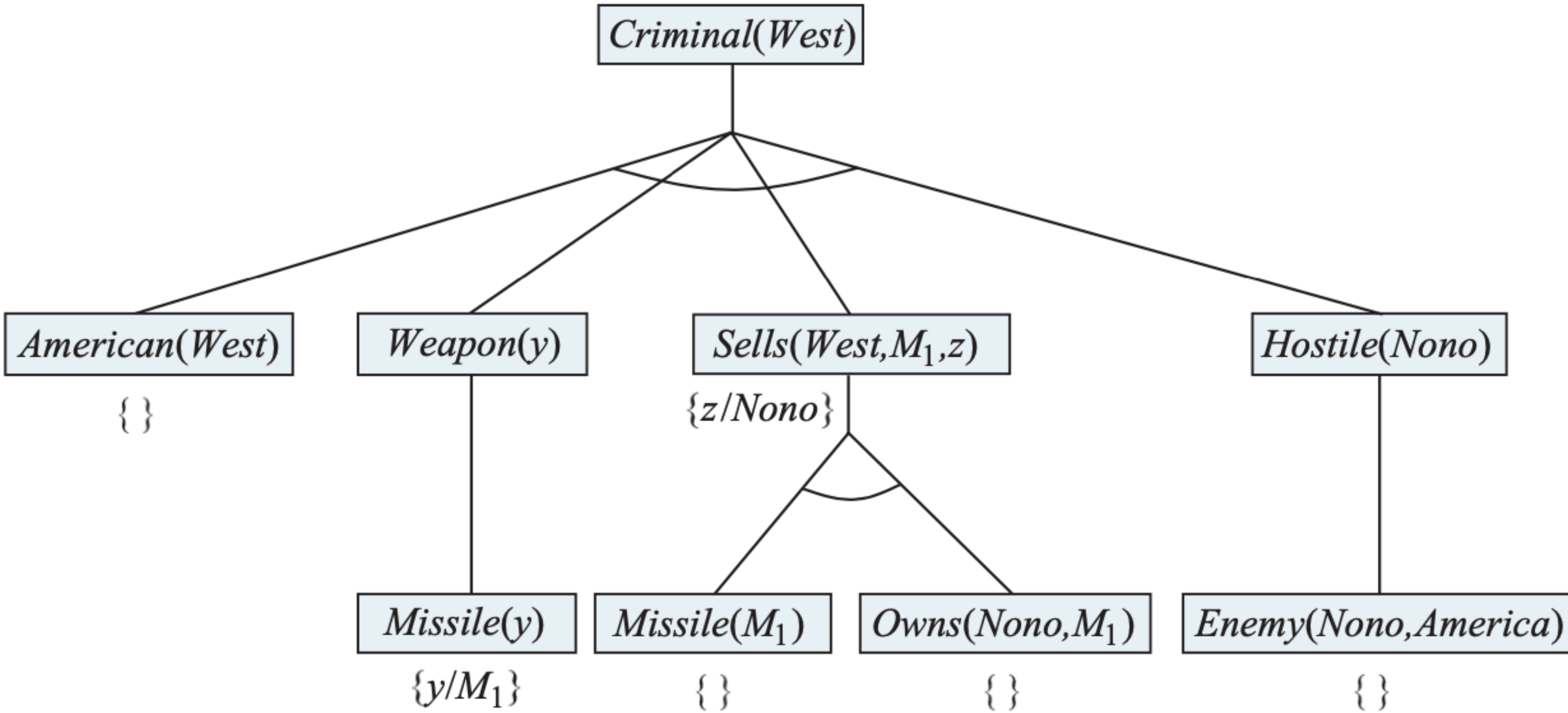
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A simple backward-chaining algorithm for first-order knowledge bases

```
function FOL-BC-ASK(KB, query) returns a generator of substitutions
  return FOL-BC-OR(KB, query, { })

function FOL-BC-OR(KB, goal, θ) returns a substitution
  for each rule in FETCH-RULES-FOR-GOAL(KB, goal) do
    (lhs ⇒ rhs) ← STANDARDIZE-VARIABLES(rule)
    for each θ′ in FOL-BC-AND(KB, lhs, UNIFY(rhs, goal, θ)) do
      yield θ′

function FOL-BC-AND(KB, goals, θ) returns a substitution
  if θ = failure then return
  else if LENGTH(goals) = 0 then yield θ
  else
    first,rest ← FIRST(goals), REST(goals)
    for each θ′ in FOL-BC-OR(KB, SUBST(θ, first), θ) do
      for each θ″ in FOL-BC-AND(KB, rest, θ′) do
        yield θ″
```

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Proof tree constructed by backward chaining t o prove that West is a criminal



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

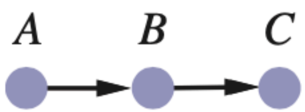
Pseudocode representing the result of compiling the Append predicate

procedure APPEND(ax, y, az, $continuation$)

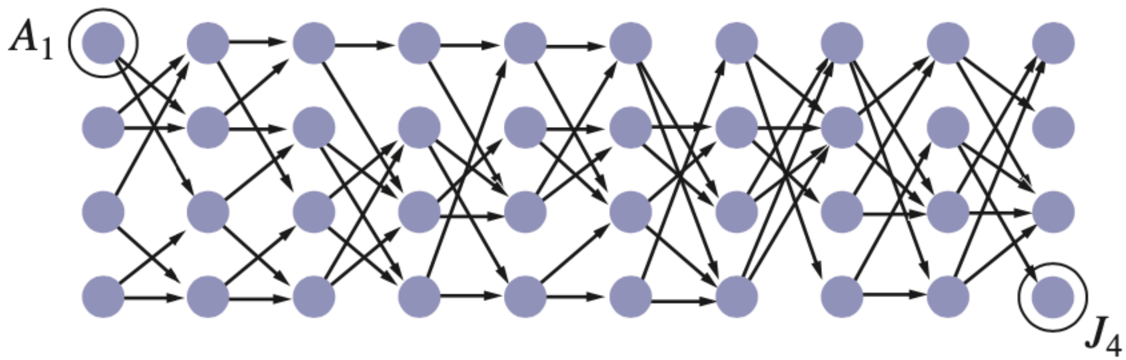
$trail \leftarrow$ GLOBAL-TRAIL-POINTER()
if $ax = [\,]$ and UNIFY(y, az) **then** CALL($continuation$)
RESET-TRAIL($trail$)
$a, x, z \leftarrow$ NEW-VARIABLE(), NEW-VARIABLE(), NEW-VARIABLE()
if UNIFY(ax, $[a] + x$) and UNIFY(az, $[a \mid z]$) **then** APPEND(x, y, z, $continuation$)

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Finding a path from A to C can lead Prolog into an infinite loop.



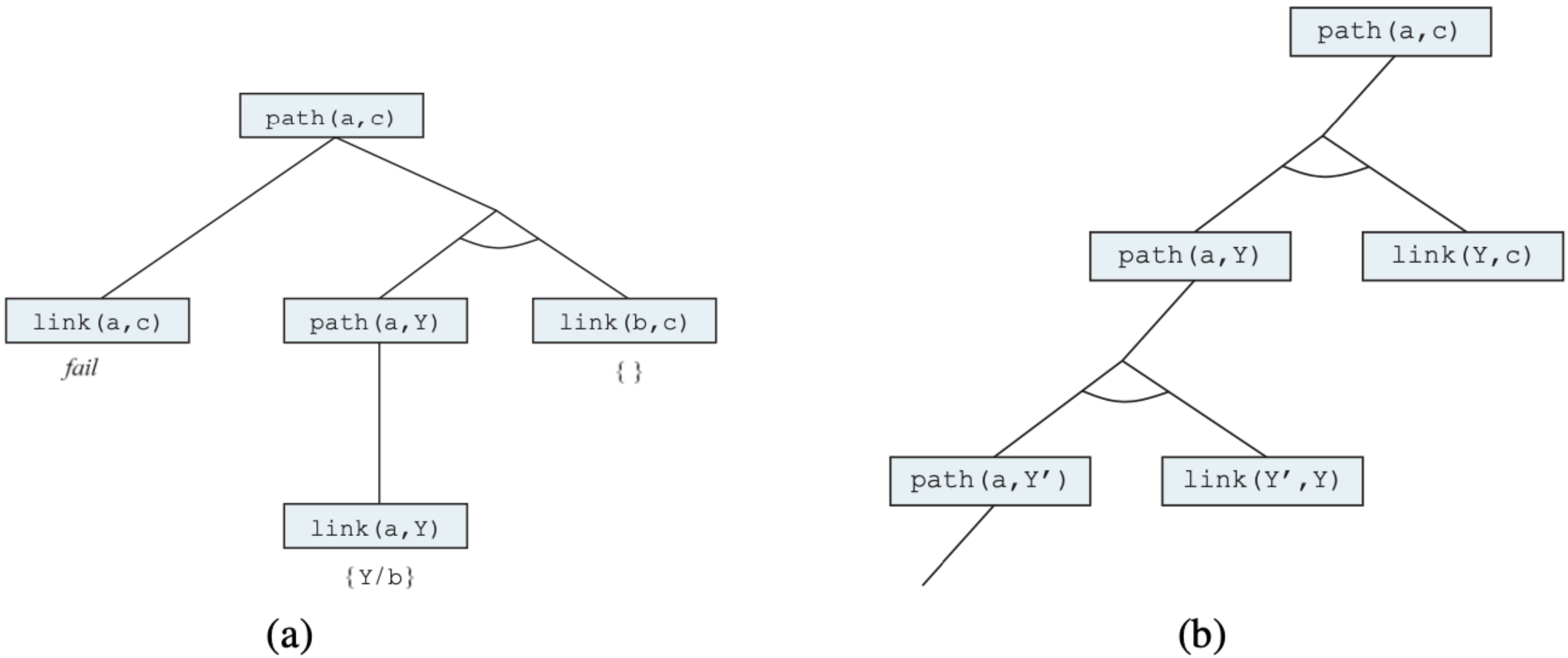
(a)



(b)

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

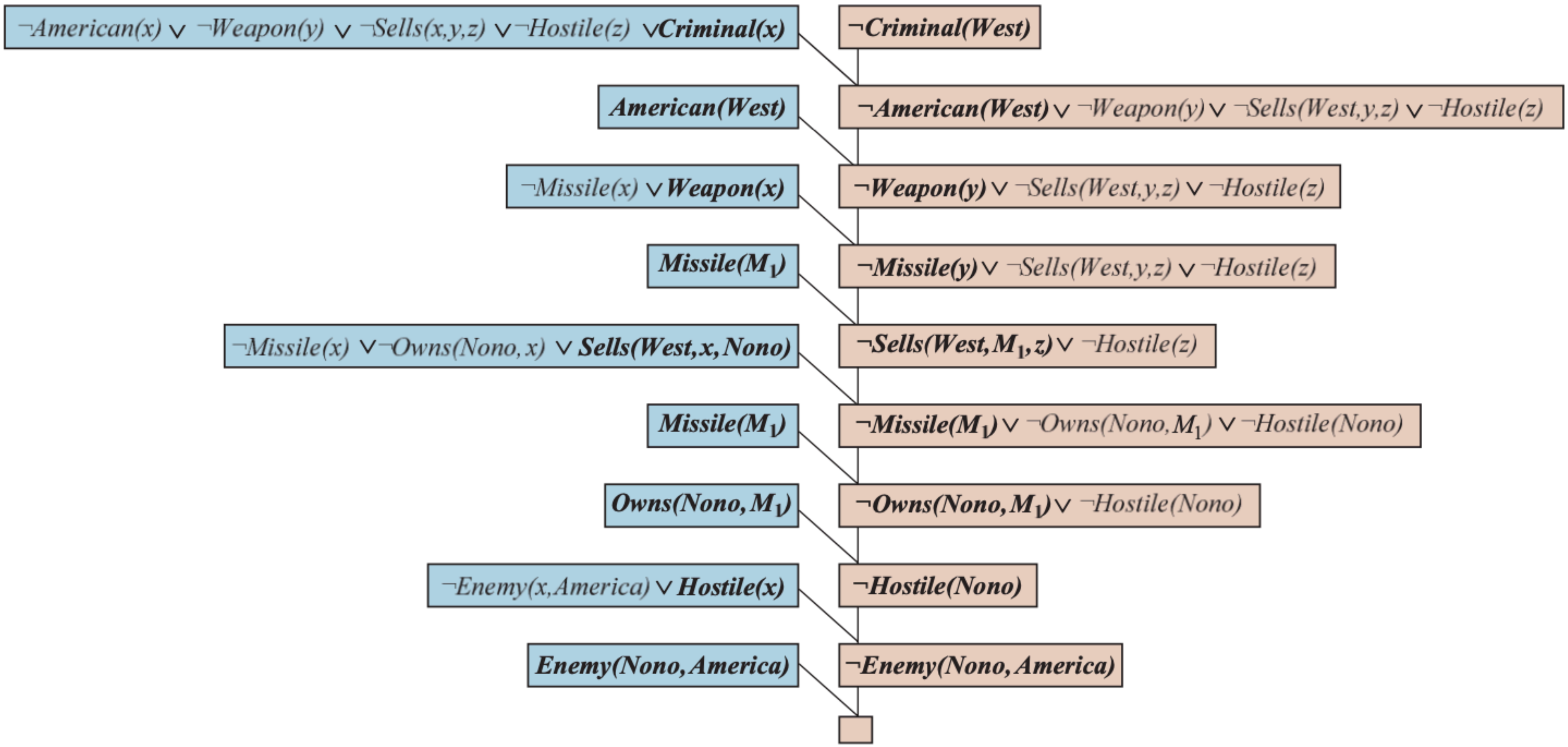
Proof that a path exists from A to C.



Infinite proof tree generated when the clauses are in the “wrong” order

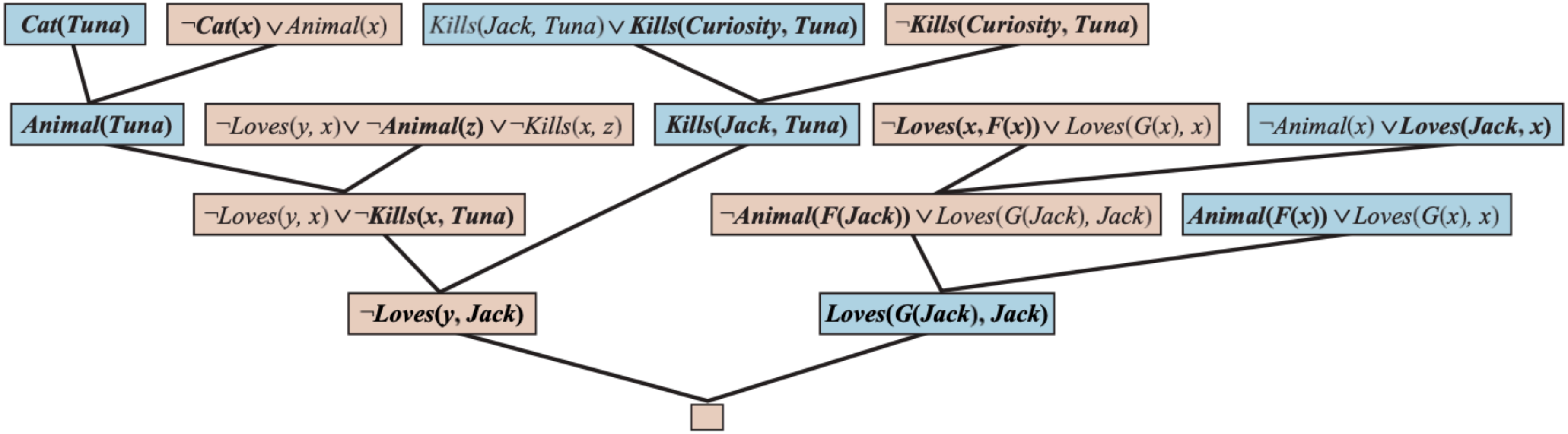
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A resolution proof that West is a criminal



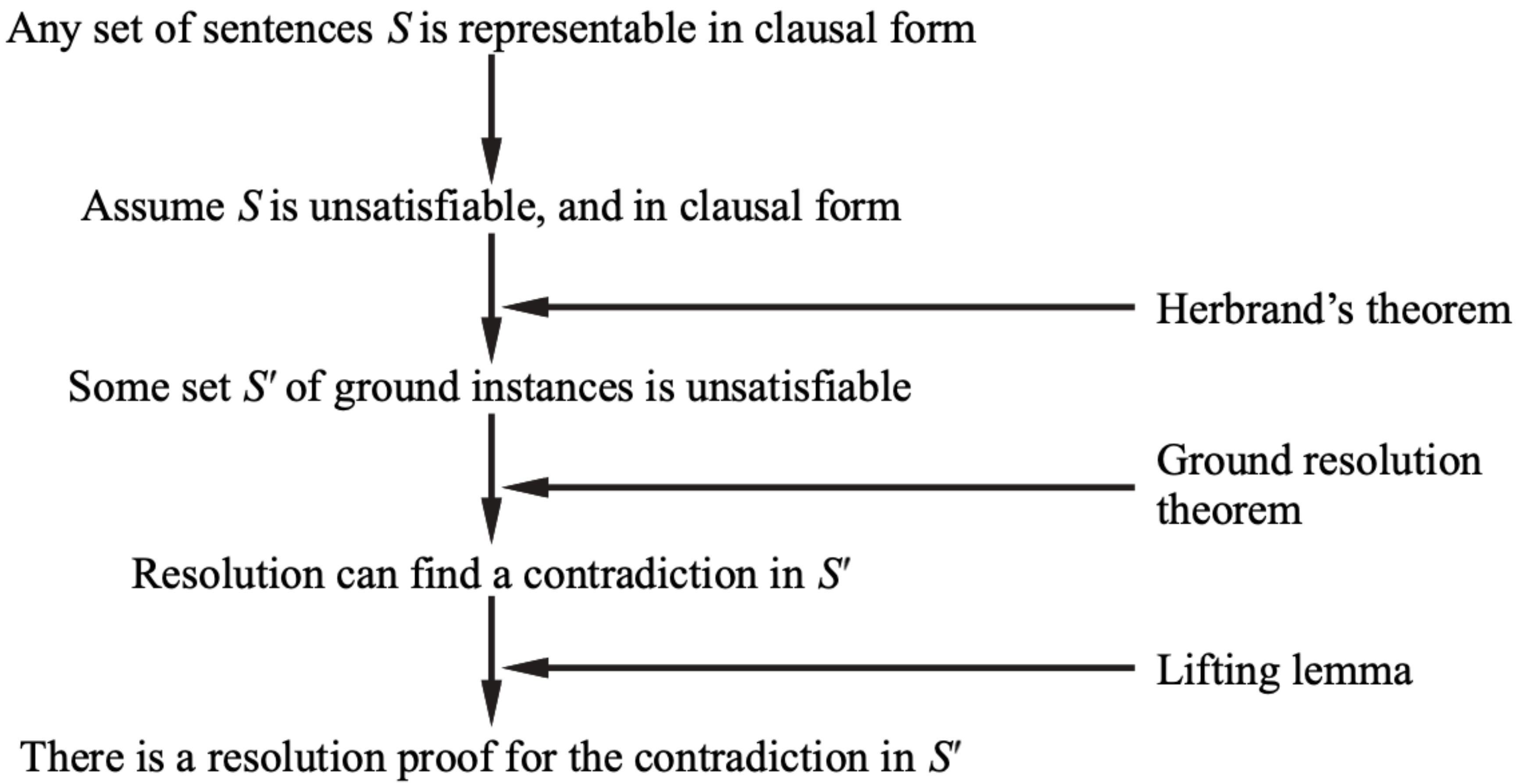
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A resolution proof that Curiosity killed the cat



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

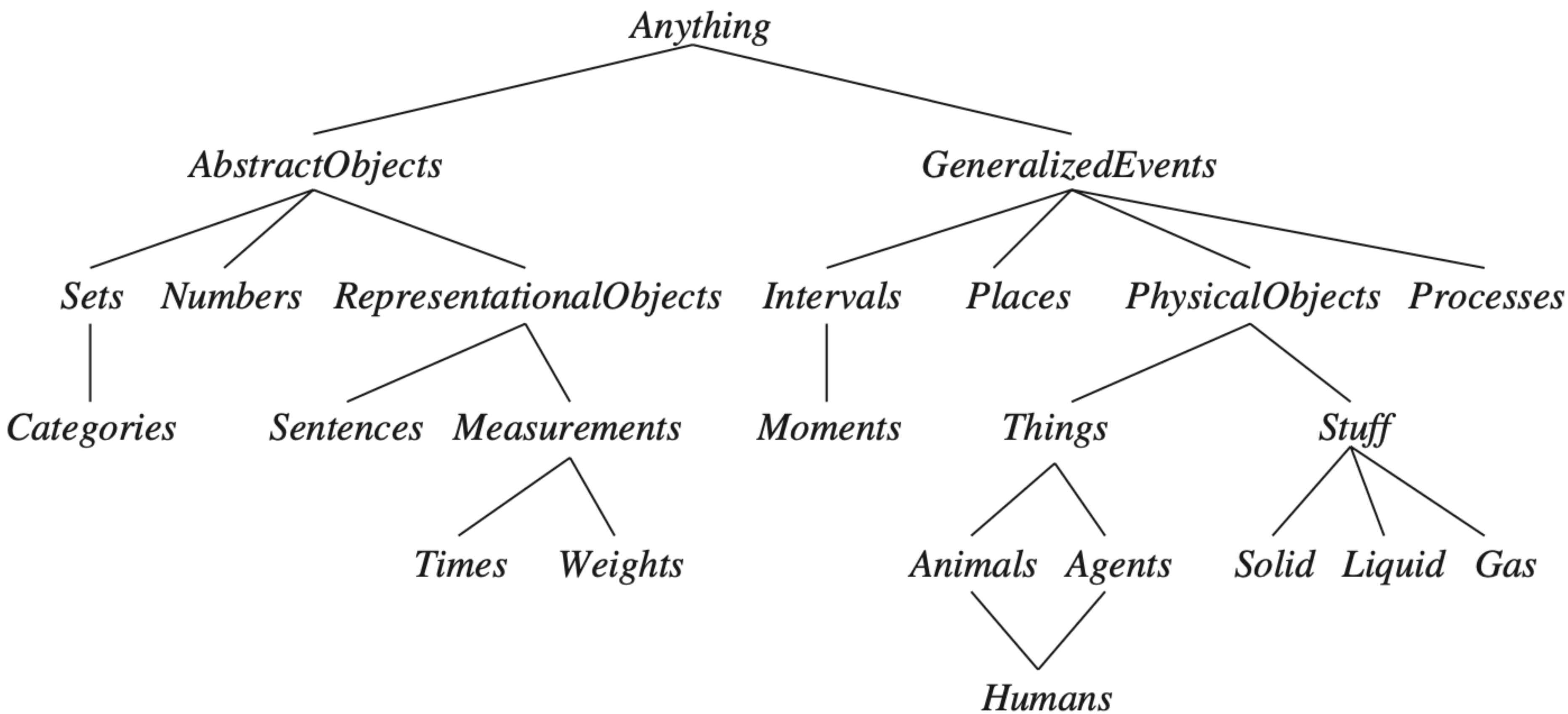
Structure of a completeness proof for resolution



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

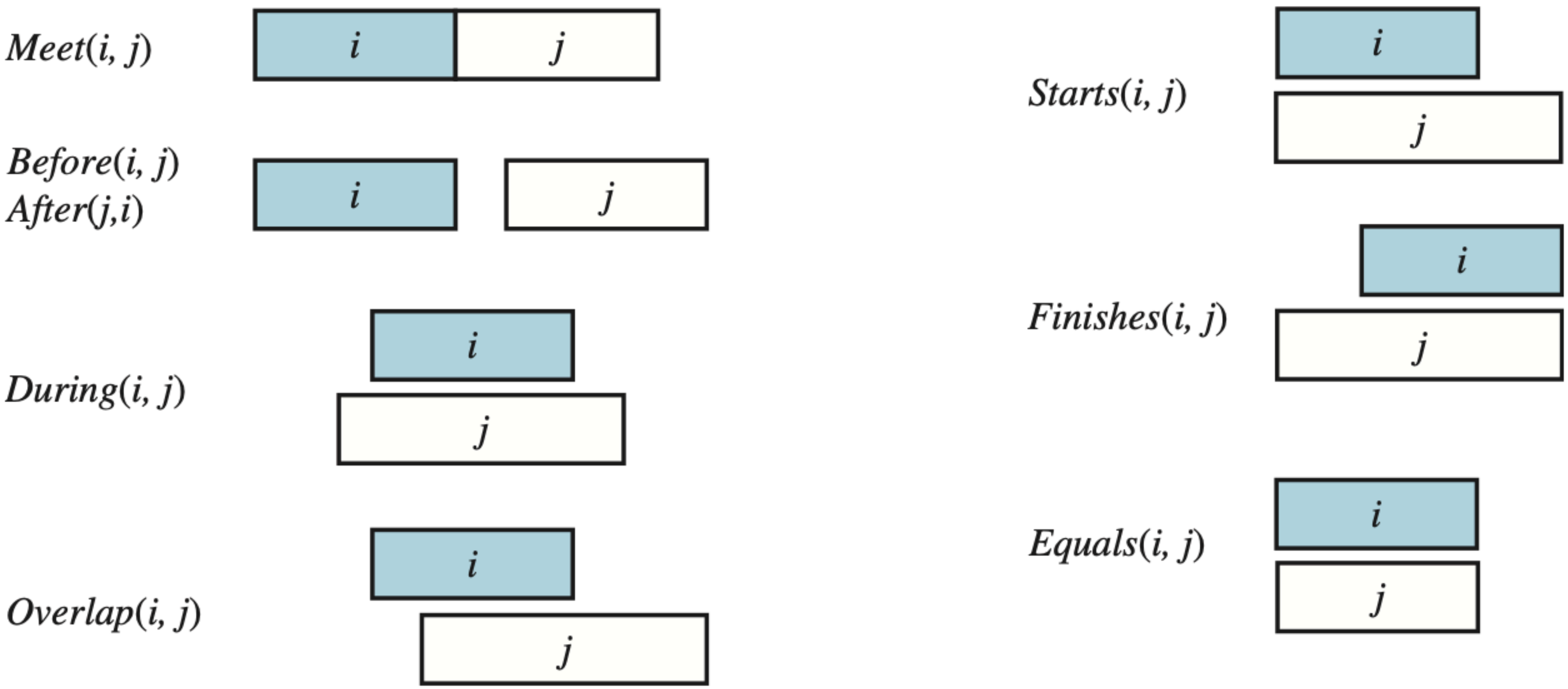
Knowledge Representation

The Upper Ontology of the World



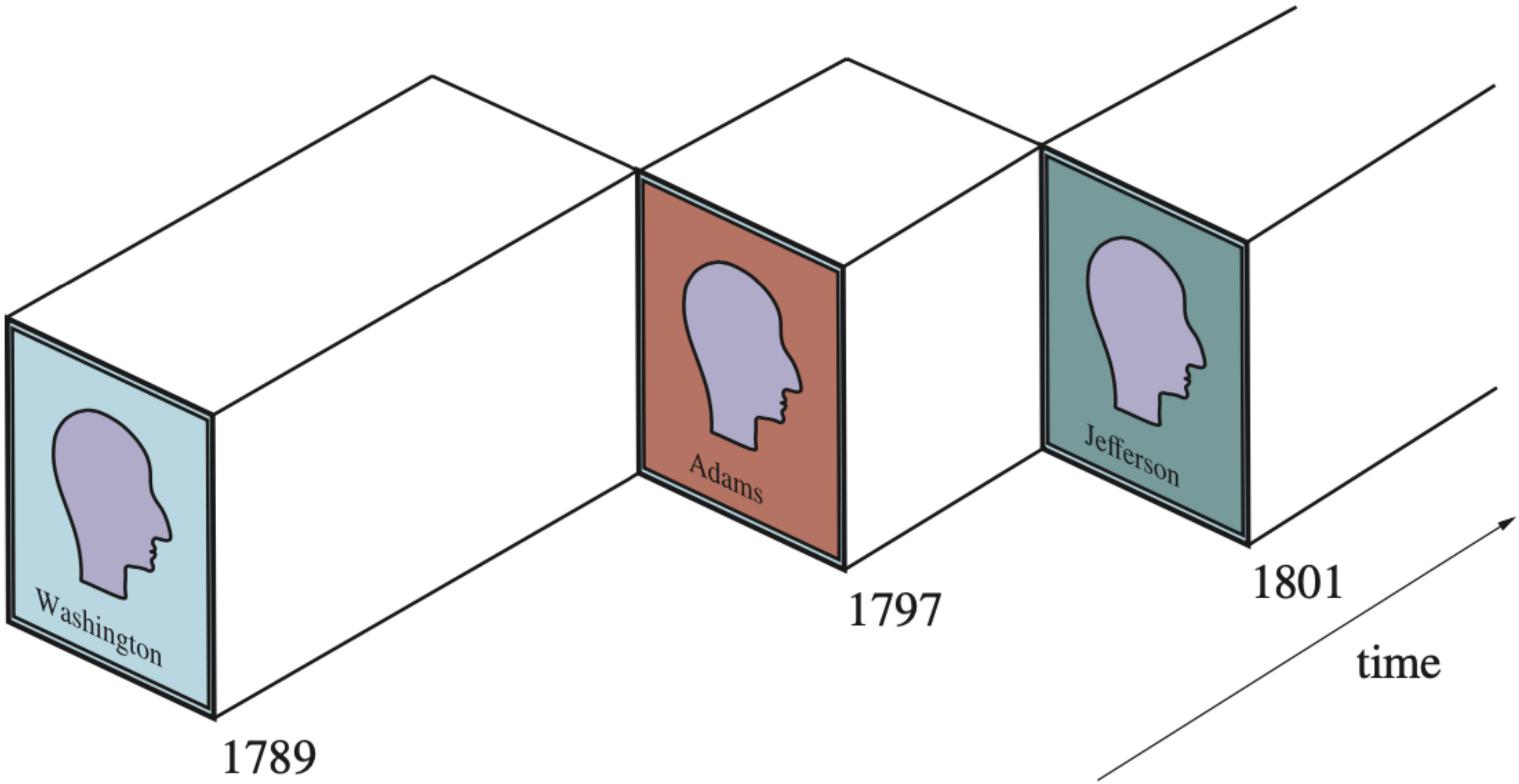
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Predicates on time intervals



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A schematic view of the object President (USA) for the early years

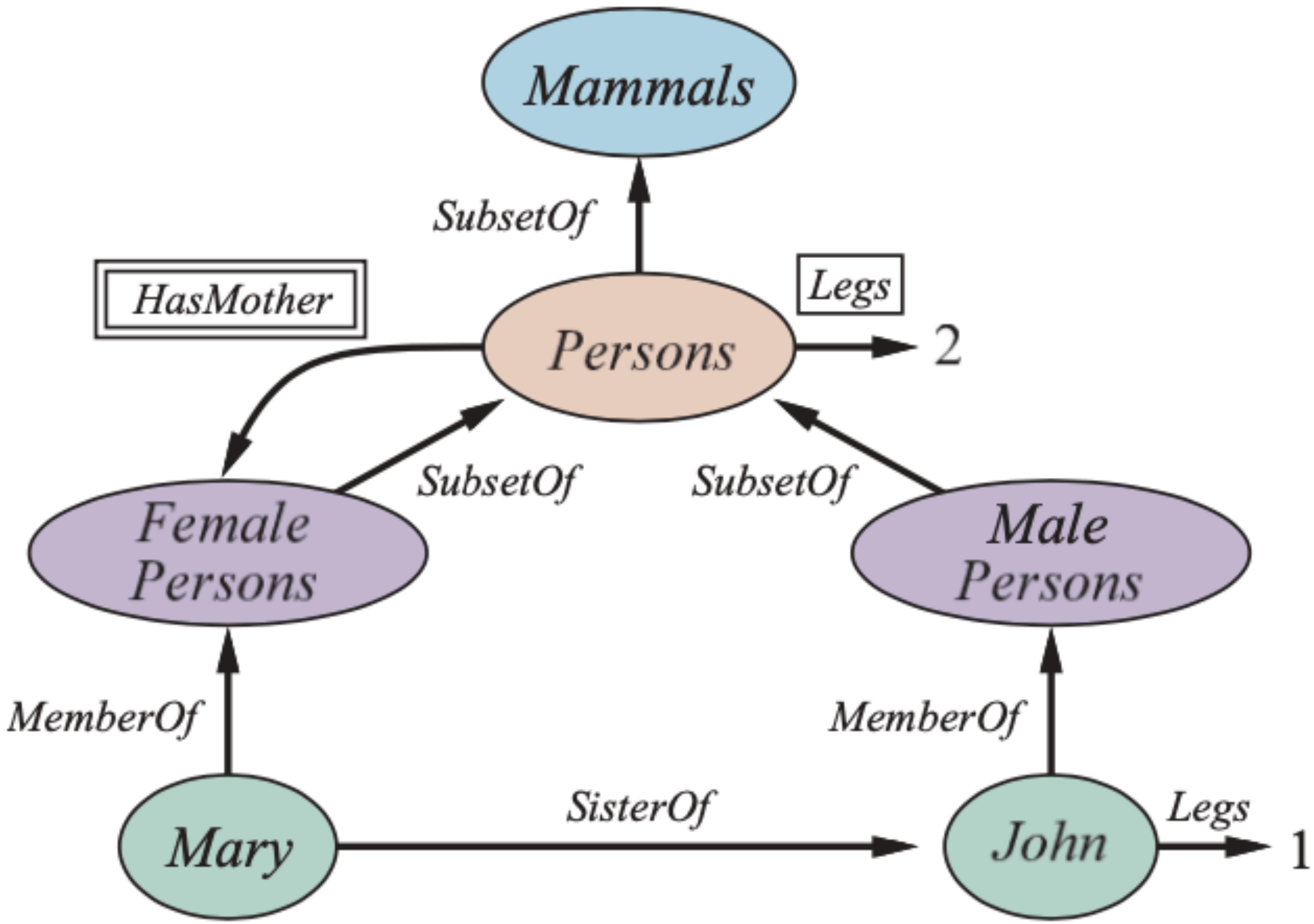


Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A semantic network

with four objects (John, Mary, 1, and 2) and four categories

Relations are denoted by labeled links

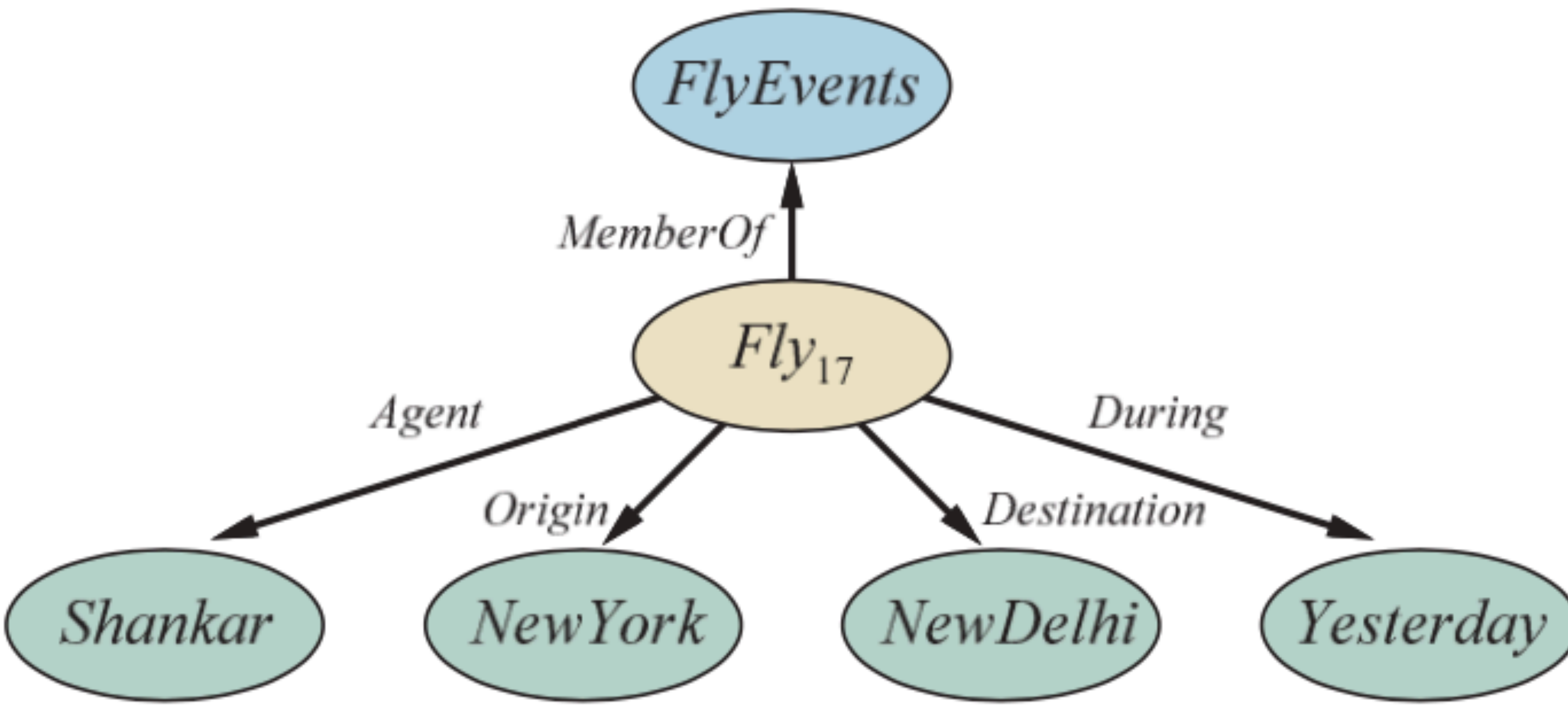


Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Semantic network

Representation of the logical assertion

Fly (Shankar, NewYork, NewDelhi, Yesterday)



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

The syntax of descriptions in a subset of the CLASSIC language.

$$
\begin{aligned}
\mathit{Concept} &\rightarrow \textbf{Thing} \mid \mathit{ConceptName} \\
&\mid \textbf{And}(\mathit{Concept}, \ldots) \\
&\mid \textbf{All}(\mathit{RoleName}, \mathit{Concept}) \\
&\mid \textbf{AtLeast}(\mathit{Integer}, \mathit{RoleName}) \\
&\mid \textbf{AtMost}(\mathit{Integer}, \mathit{RoleName}) \\
&\mid \textbf{Fills}(\mathit{RoleName}, \mathit{IndividualName}, \ldots) \\
&\mid \textbf{SameAs}(\mathit{Path}, \mathit{Path}) \\
&\mid \textbf{OneOf}(\mathit{IndividualName}, \ldots) \\
\mathit{Path} &\rightarrow [\mathit{RoleName}, \ldots] \\
\mathit{ConceptName} &\rightarrow \mathit{Adult} \mid \mathit{Female} \mid \mathit{Male} \mid \ldots \\
\mathit{RoleName} &\rightarrow \mathit{Spouse} \mid \mathit{Daughter} \mid \mathit{Son} \mid \ldots
\end{aligned}
$$

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Automated Planning

A PDDL description of an air cargo transportation planning problem

$$
\begin{aligned}
&Init(At(C_1, SFO) \land At(C_2, JFK) \land At(P_1, SFO) \land At(P_2, JFK) \\
&\quad \land Cargo(C_1) \land Cargo(C_2) \land Plane(P_1) \land Plane(P_2) \\
&\quad \land Airport(JFK) \land Airport(SFO)) \\
&Goal(At(C_1, JFK) \land At(C_2, SFO)) \\
&Action(Load(c, p, a), \\
&\quad \text{PRECOND: } At(c, a) \land At(p, a) \land Cargo(c) \land Plane(p) \land Airport(a) \\
&\quad \text{EFFECT: } \neg At(c, a) \land In(c, p)) \\
&Action(Unload(c, p, a), \\
&\quad \text{PRECOND: } In(c, p) \land At(p, a) \land Cargo(c) \land Plane(p) \land Airport(a) \\
&\quad \text{EFFECT: } At(c, a) \land \neg In(c, p)) \\
&Action(Fly(p, from, to), \\
&\quad \text{PRECOND: } At(p, from) \land Plane(p) \land Airport(from) \land Airport(to) \\
&\quad \text{EFFECT: } \neg At(p, from) \land At(p, to))
\end{aligned}
$$

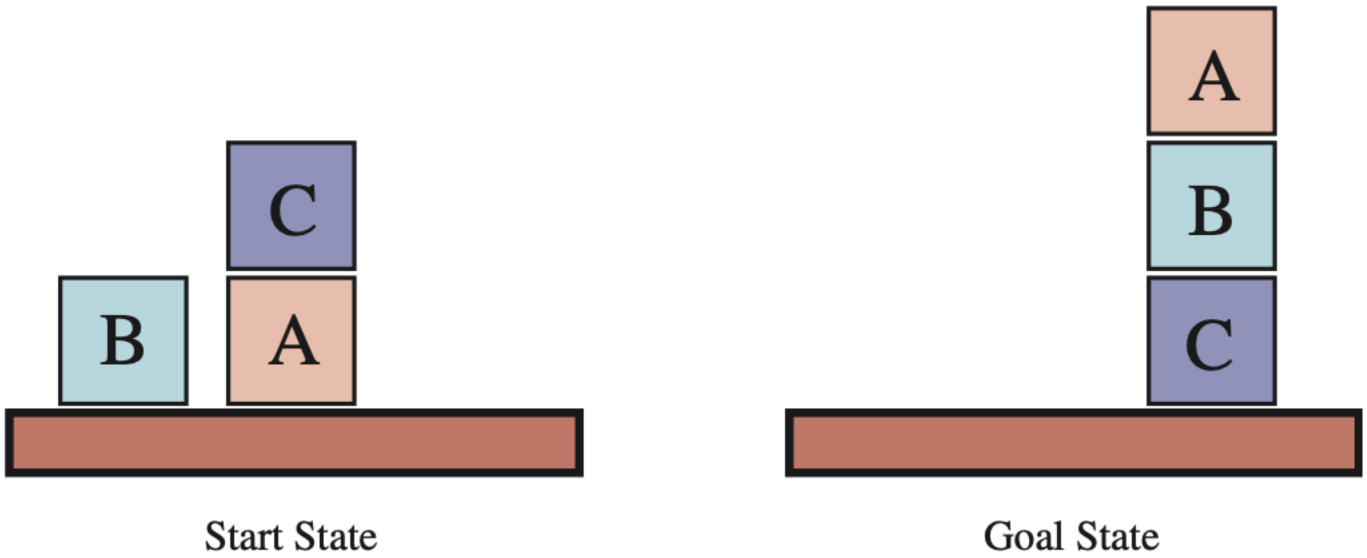
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

The simple spare tire problem

$$
\begin{aligned}
&Init(Tire(Flat) \land Tire(Spare) \land At(Flat, Axle) \land At(Spare, Trunk))\\
&Goal(At(Spare, Axle))\\
&Action(Remove(obj, loc),\\
&\quad \text{PRECOND: } At(obj, loc)\\
&\quad \text{EFFECT: } \neg At(obj, loc) \land At(obj, Ground))\\
&Action(PutOn(t,\ Axle),\\
&\quad \text{PRECOND: } Tire(t) \land At(t, Ground) \land \neg At(Flat, Axle) \land \neg At(Spare, Axle)\\
&\quad \text{EFFECT: } \neg At(t, Ground) \land At(t, Axle))\\
&Action(LeaveOvernight,\\
&\quad \text{PRECOND: }\\
&\quad \text{EFFECT: } \neg At(Spare, Ground) \land \neg At(Spare, Axle) \land \neg At(Spare, Trunk)\\
&\qquad\qquad \land \neg At(Flat, Ground) \land \neg At(Flat, Axle) \land \neg At(Flat,\ Trunk))
\end{aligned}
$$

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Diagram of the blocks-world problem



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

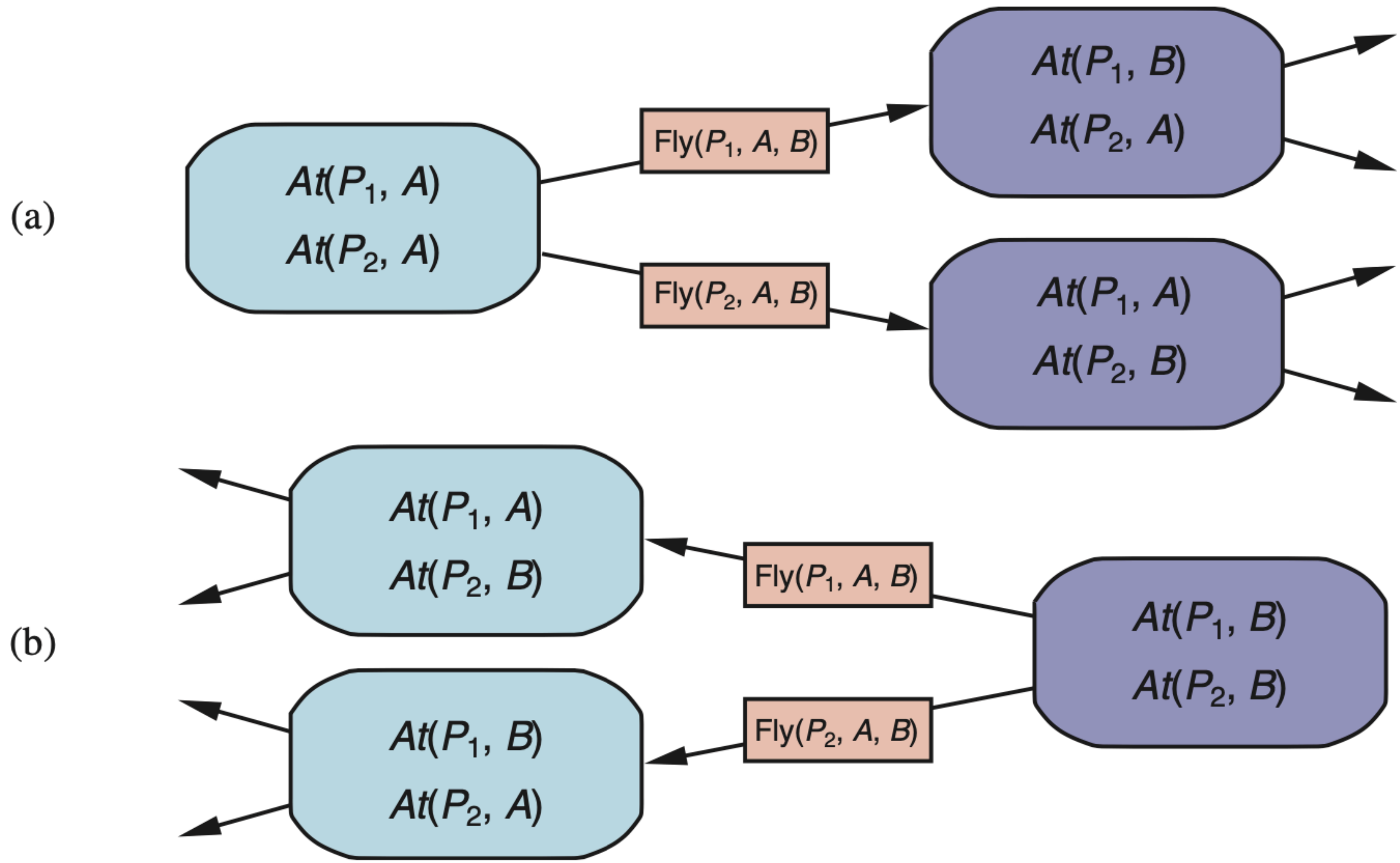
A planning problem in the blocks world: building a three-block tower

$$\begin{aligned}
&Init(On(A, Table) \land On(B, Table) \land On(C, A) \\
&\quad \land Block(A) \land Block(B) \land Block(C) \land Clear(B) \land Clear(C) \land Clear(Table)) \\
&Goal(On(A, B) \land On(B, C)) \\
&Action(Move(b, x, y), \\
&\quad \text{PRECOND: } On(b, x) \land Clear(b) \land Clear(y) \land Block(b) \land Block(y) \land \\
&\qquad\qquad (b \neq x) \land (b \neq y) \land (x \neq y), \\
&\quad \text{EFFECT: } On(b, y) \land Clear(x) \land \neg On(b, x) \land \neg Clear(y)) \\
&Action(MoveToTable(b, x), \\
&\quad \text{PRECOND: } On(b, x) \land Clear(b) \land Block(b) \land Block(x), \\
&\quad \text{EFFECT: } On(b, Table) \land Clear(x) \land \neg On(b, x))
\end{aligned}$$

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

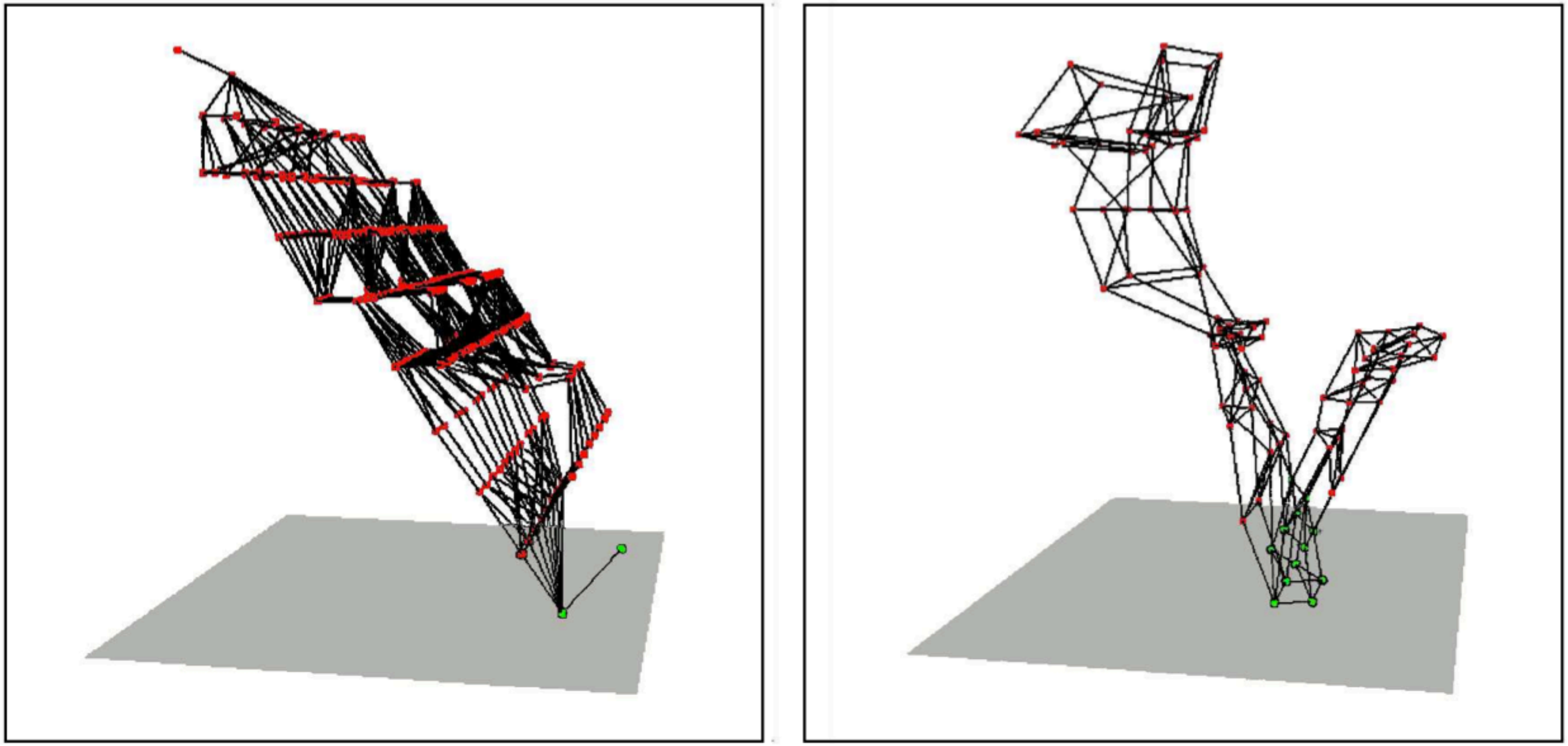
Two approaches to searching for a plan

(a) Forward (progression) search
(b) Backward (regression) search



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Two state spaces from planning problems with the ignore-delete-lists heuristic



Definitions of possible refinements for two high-level actions

$Refinement(Go(Home, SFO),$
  STEPS: $[Drive(Home, SFOLongTermParking),$
    $Shuttle(SFOLongTermParking, SFO)]\ )$
$Refinement(Go(Home, SFO),$
  STEPS: $[Taxi(Home, SFO)]\ )$

$Refinement(Navigate([a, b], [x, y]),$
  PRECOND: $a = x \ \wedge\ b = y$
  STEPS: $[\,]\ )$
$Refinement(Navigate([a, b], [x, y]),$
  PRECOND: $Connected([a, b], [a - 1, b])$
  STEPS: $[Left, Navigate([a - 1, b], [x, y])]\ )$
$Refinement(Navigate([a, b], [x, y]),$
  PRECOND: $Connected([a, b], [a + 1, b])$
  STEPS: $[Right, Navigate([a + 1, b], [x, y])]\ )$

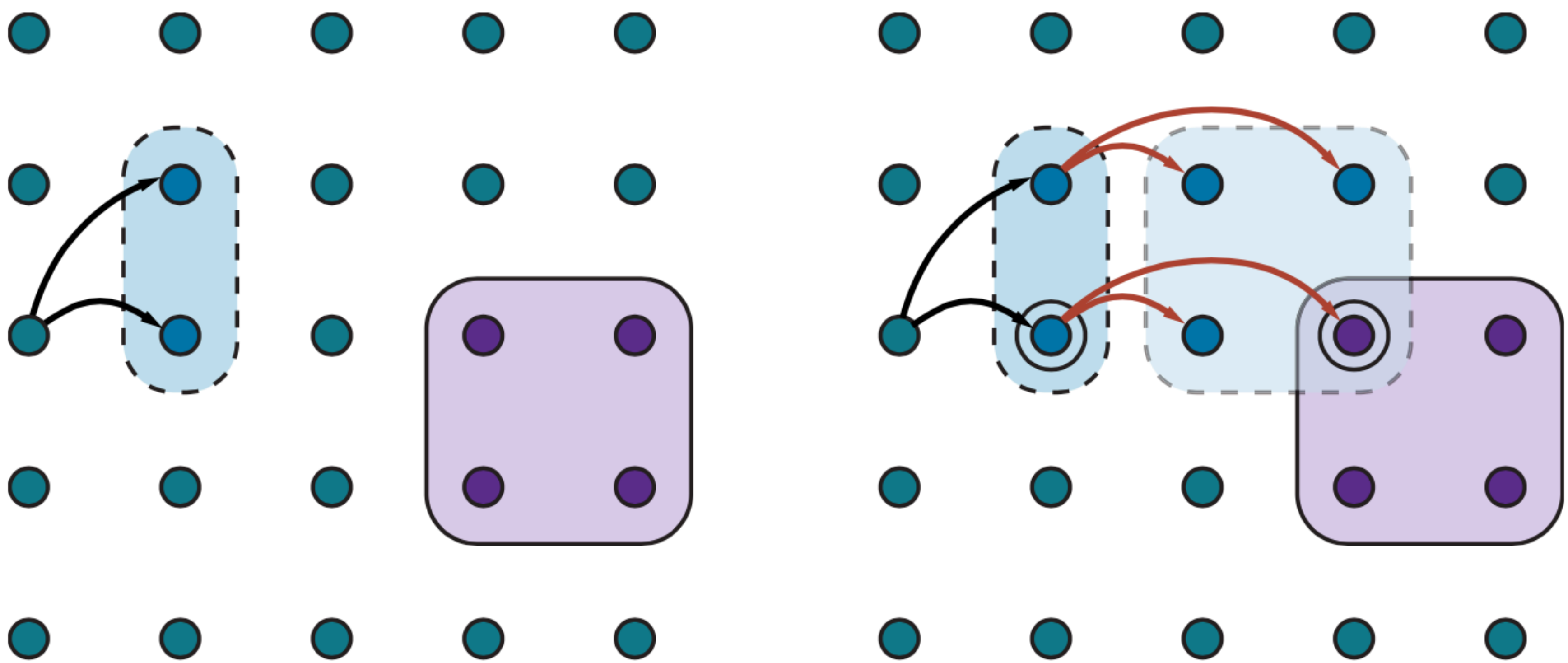
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A breadth-first implementation of hierarchical forward planning search

```
function HIERARCHICAL-SEARCH(problem, hierarchy) returns a solution or failure
  frontier ← a FIFO queue with [Act] as the only element
  while true do
    if IS-EMPTY(frontier) then return failure
    plan ← POP(frontier)          // chooses the shallowest plan in frontier
    hla ← the first HLA in plan, or null if none
    prefix,suffix ← the action subsequences before and after hla in plan
    outcome ← RESULT(problem.INITIAL, prefix)
    if hla is null then          // so plan is primitive and outcome is its result
      if problem.IS-GOAL(outcome) then return plan
    else for each sequence in REFINEMENTS(hla, outcome, hierarchy) do
      add APPEND(prefix, sequence, suffix) to frontier
```

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

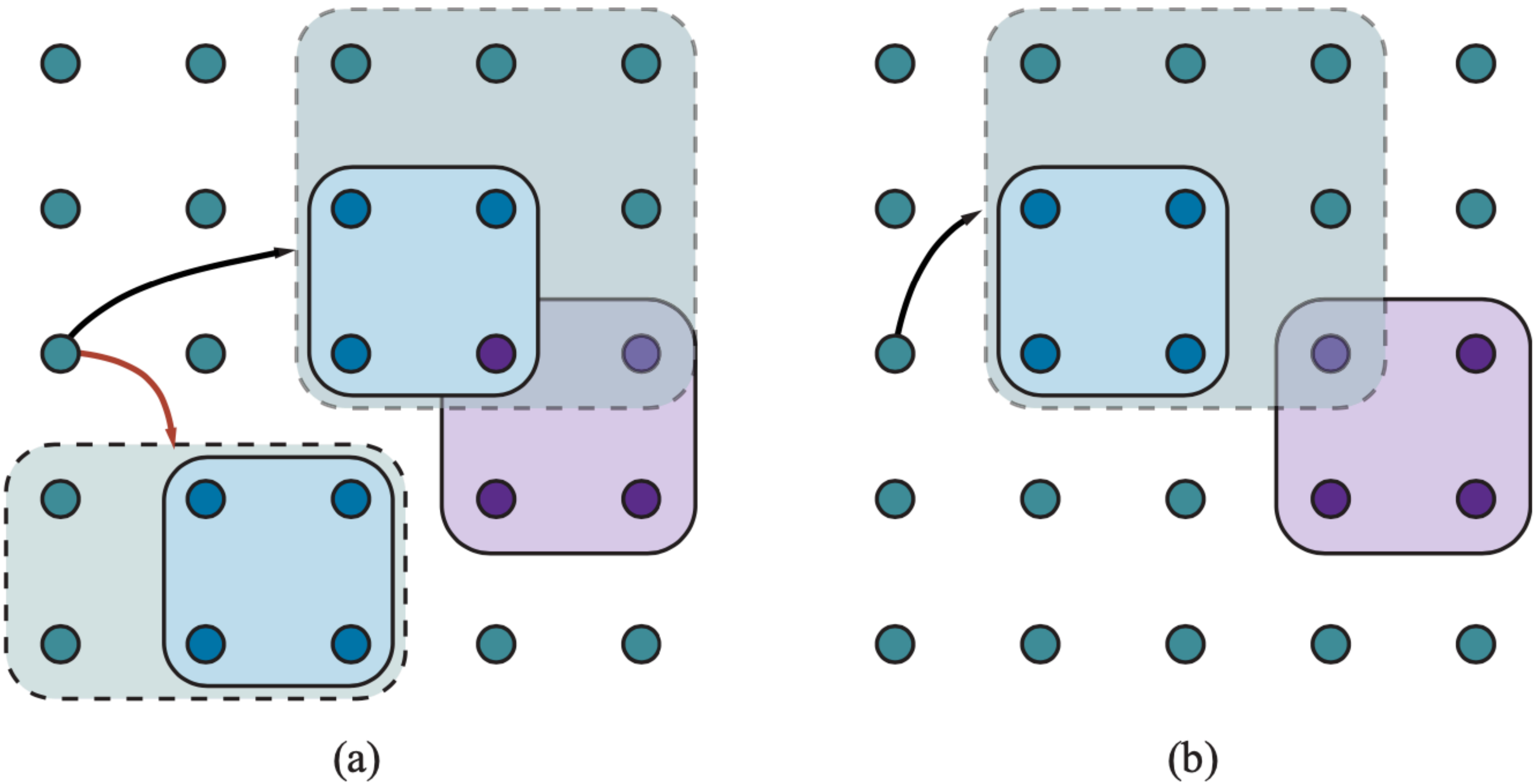
Schematic examples of reachable sets



(a) (b)

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Goal achievement for high-level plans with approximate descriptions



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A hierarchical planning algorithm

```
function ANGELIC-SEARCH(problem, hierarchy, initialPlan) returns solution or fail
  frontier ← a FIFO queue with initialPlan as the only element
  while true do
     if EMPTY?(frontier) then return fail
     plan ← POP(frontier)        // chooses the shallowest node in frontier
     if REACH⁺(problem.INITIAL, plan) intersects problem.GOAL then
        if plan is primitive then return plan        // REACH⁺ is exact for primitive plans
        guaranteed ← REACH⁻(problem.INITIAL, plan) ∩ problem.GOAL
        if guaranteed≠{ } and MAKING-PROGRESS(plan, initialPlan) then
           finalState ← any element of guaranteed
           return DECOMPOSE(hierarchy, problem.INITIAL, plan, finalState)
        hla ← some HLA in plan
        prefix,suffix ← the action subsequences before and after hla in plan
        outcome ← RESULT(problem.INITIAL, prefix)
        for each sequence in REFINEMENTS(hla, outcome, hierarchy) do
           frontier ← Insert(APPEND(prefix, sequence, suffix), frontier)
```

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

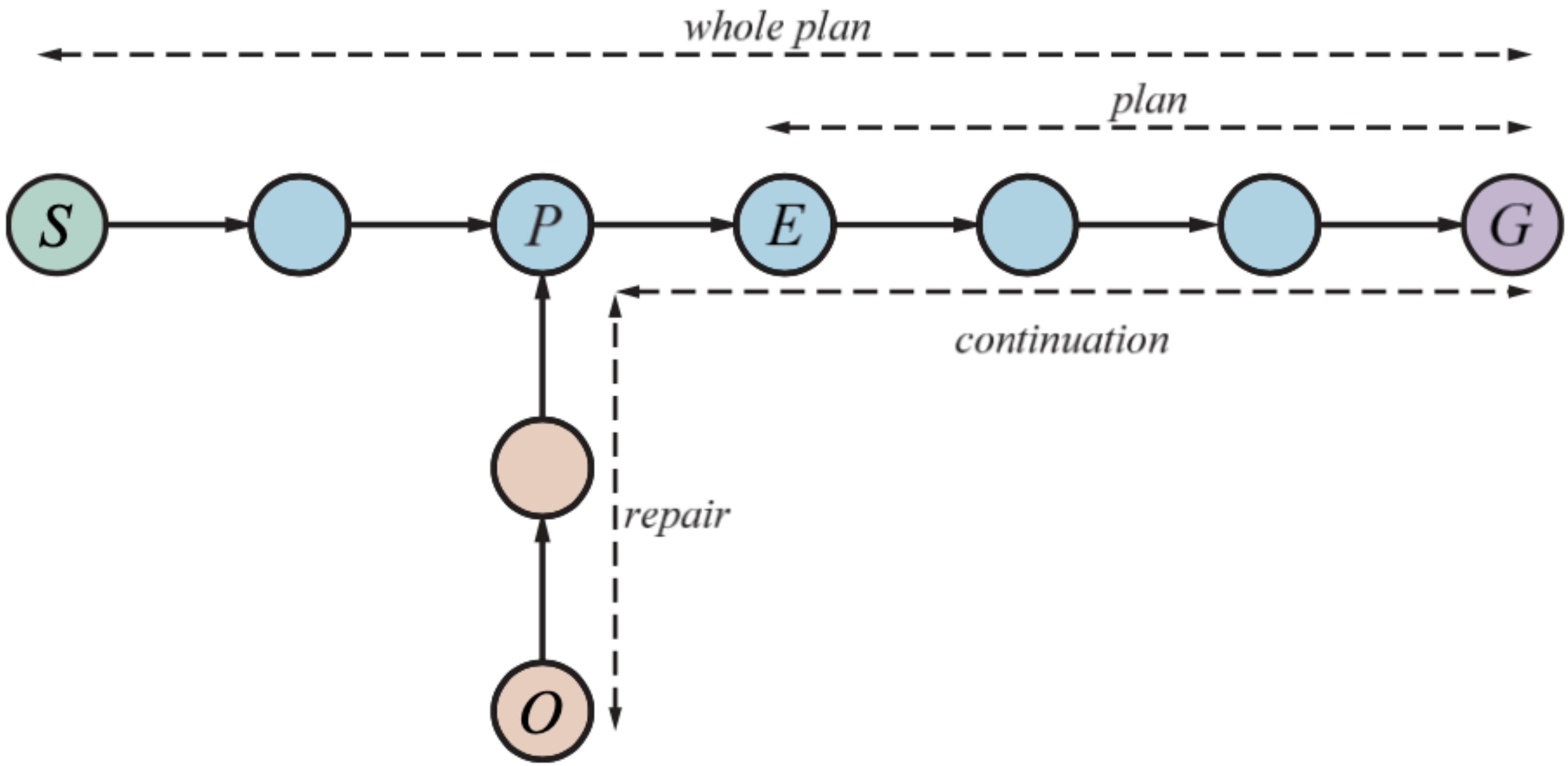
A hierarchical planning algorithm Decompose solution

function DECOMPOSE(*hierarchy*, s_0, *plan*, s_f) **returns** a solution

    solution ← an empty plan
    while *plan* is not empty **do**
        action ← REMOVE-LAST(*plan*)
        s_i ← a state in REACH^{-}(s_0, *plan*) such that $s_f \in$ REACH^{-}(s_i, *action*)
        problem ← a problem with INITIAL = s_i and GOAL = s_f
        solution ← APPEND(ANGELIC-SEARCH(*problem*, *hierarchy*, *action*), *solution*)
        $s_f \leftarrow s_i$
    return *solution*

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

At first, the sequence “whole plan” is expected to get the agent from S to G



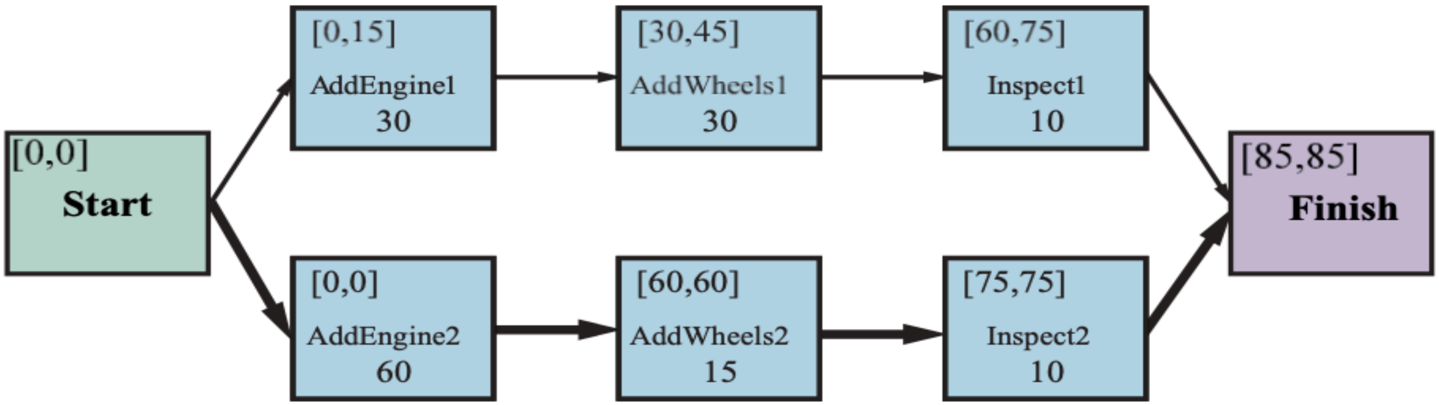
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

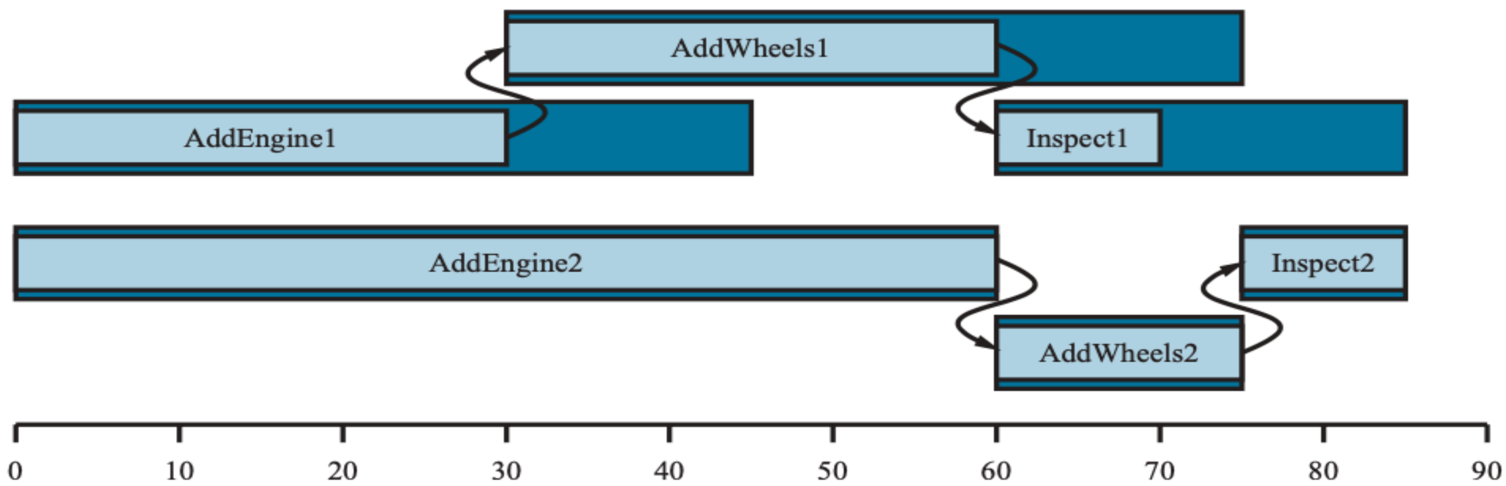
A job-shop scheduling problem for assembling two cars, with resource constraints

$$
\begin{aligned}
&Jobs(\{AddEngine1 \prec AddWheels1 \prec Inspect1\},\\
&\quad\quad \{AddEngine2 \prec AddWheels2 \prec Inspect2\})\\[1em]
&Resources(EngineHoists(1),\ WheelStations(1),\ Inspectors(e2),\ LugNuts(500))\\[1em]
&Action(AddEngine1,\ \textsc{Duration}{:}30,\\
&\quad\quad \textsc{Use}{:}EngineHoists(1))\\
&Action(AddEngine2,\ \textsc{Duration}{:}60,\\
&\quad\quad \textsc{Use}{:}EngineHoists(1))\\
&Action(AddWheels1,\ \textsc{Duration}{:}30,\\
&\quad\quad \textsc{Consume}{:}LugNuts(20),\ \textsc{Use}{:}WheelStations(1))\\
&Action(AddWheels2,\ \textsc{Duration}{:}15,\\
&\quad\quad \textsc{Consume}{:}LugNuts(20),\ \textsc{Use}{:}WheelStations(1))\\
&Action(Inspect_i,\ \textsc{Duration}{:}10,\\
&\quad\quad \textsc{Use}{:}Inspectors(1))
\end{aligned}
$$

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

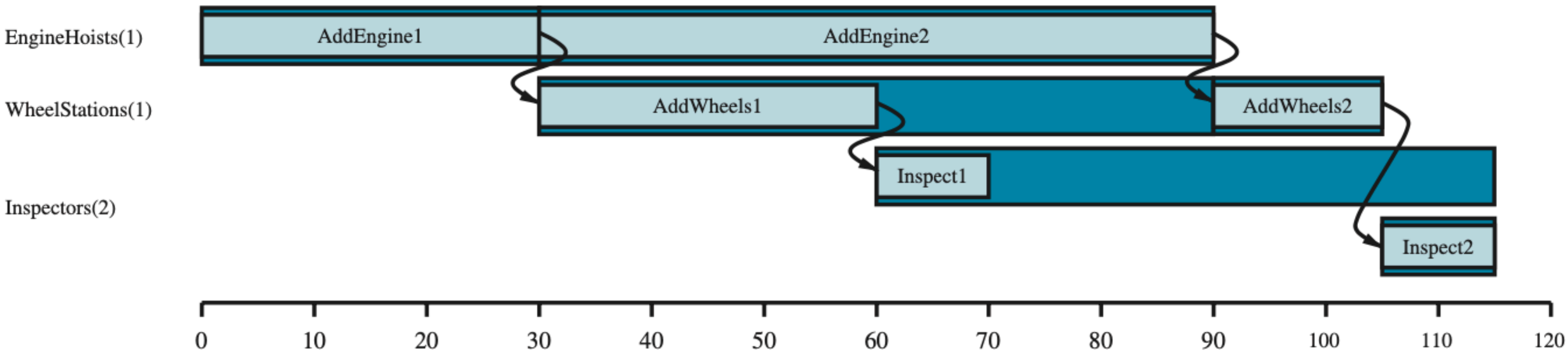
A representation of the temporal constraints for the job-shop scheduling problem





Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A solution to the job-shop scheduling problem



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

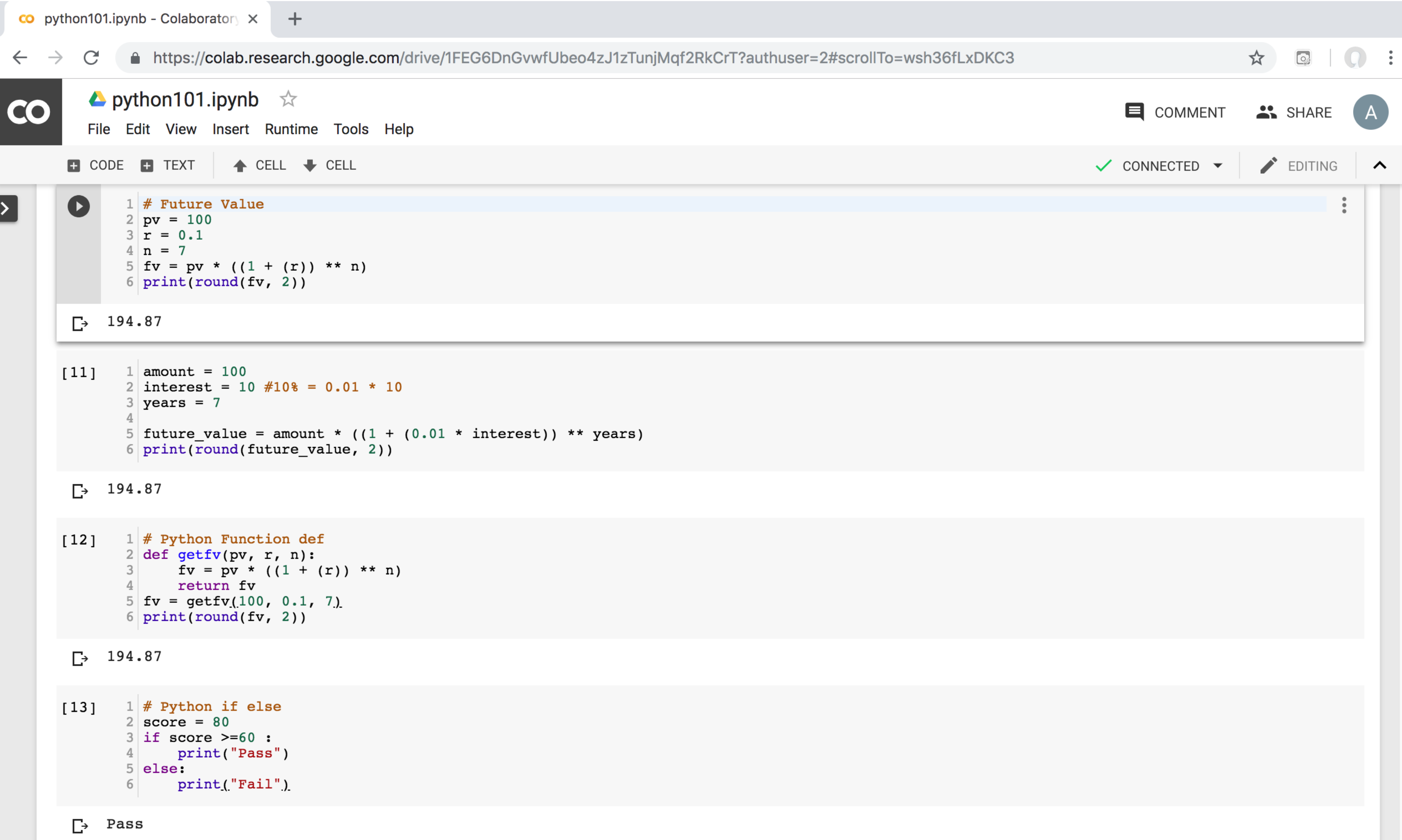
AIMA Python

- Artificial Intelligence: A Modern Approach (AIMA)
  - http://aima.cs.berkeley.edu/
- AIMA Python
  - http://aima.cs.berkeley.edu/python/readme.html
- Logic, KB Agent
  - http://aima.cs.berkeley.edu/python/logic.html

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Summary

- Logical Agents
- First-Order Logic
- Inference in First-Order Logic
- Knowledge Representation
- Automated Planning

References

- Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson.
- Aurélien Géron (2019), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd Edition, O'Reilly Media.