Artificial Intelligence

Knowledge, Reasoning and Knowledge Representation Uncertain Knowledge and Reasoning

1111AI04
MBA, IM, NTPU (M6132) (Fall 2022)
Wed 2, 3, 4 (9:10-12:00) (B8F40)

https://meet.google.com/miy-fbif-max

Min-Yuh Day, Ph.D, Associate Professor

Institute of Information Management, National Taipei University

https://web.ntpu.edu.tw/~myday

2022-10-05

Syllabus

Week Date Subject/Topics

1 2022/09/14 Introduction to Artificial Intelligence

2 2022/09/21 Artificial Intelligence and Intelligent Agents

3 2022/09/28 Problem Solving

4 2022/10/05 Knowledge, Reasoning and Knowledge Representation; Uncertain Knowledge and Reasoning

5 2022/10/12 Case Study on Artificial Intelligence I

6 2022/10/19 Machine Learning: Supervised and Unsupervised Learning

Syllabus

Week Date Subject/Topics

7 2022/10/26 The Theory of Learning and Ensemble Learning

8 2022/11/02 Midterm Project Report

9 2022/11/09 Deep Learning and Reinforcement Learning

10 2022/11/16 Deep Learning for Natural Language Processing

11 2022/11/23 Invited Talk: AI for Information Retrieval

12 2022/11/30 Case Study on Artificial Intelligence II

Syllabus

Week	Date	Subject/Topics
13	2022/12/07	Computer Vision and Robotics
14	2022/12/14	Philosophy and Ethics of AI and the Future of AI
15	2022/12/21	Final Project Report I
16	2022/12/28	Final Project Report II
17	2023/01/04	Self-learning
18	2023/01/11	Self-learning

Knowledge, Reasoning and Knowledge Representation

Uncertain Knowledge and Reasoning

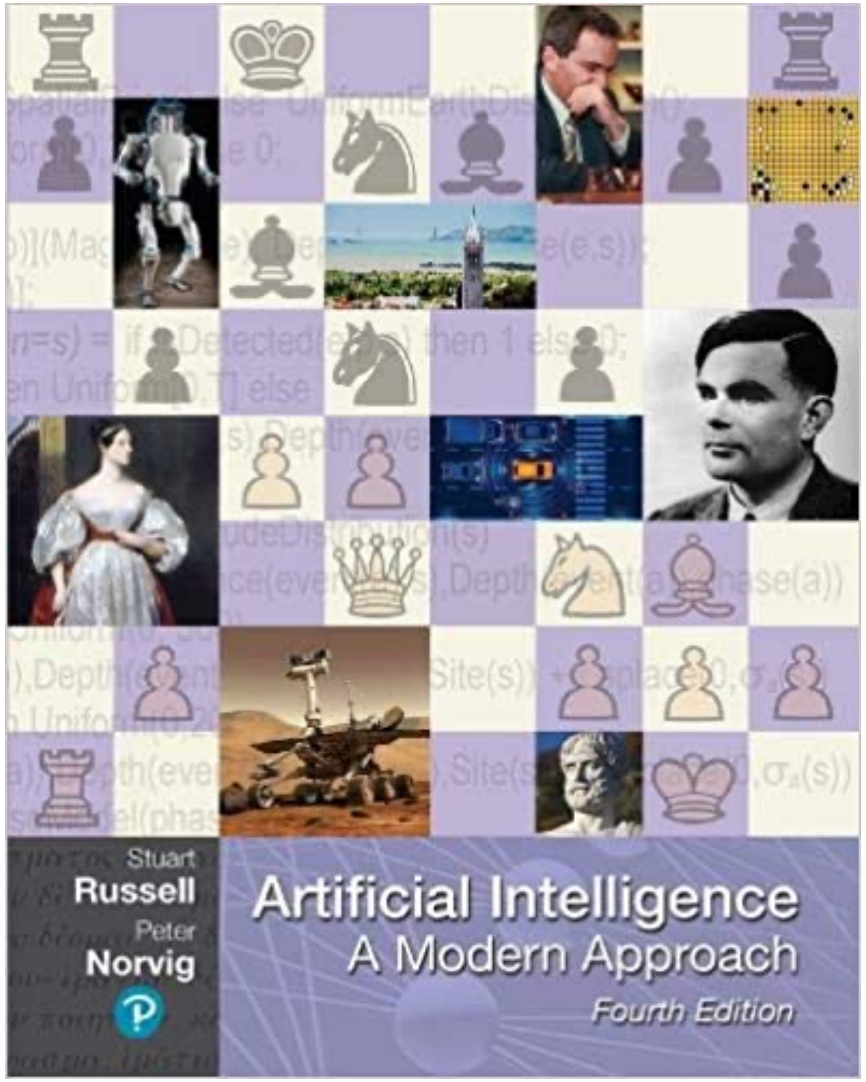
Outline

- Knowledge and Reasoning
  - Logical Agents
  - First-Order Logic
  - Inference in First-Order Logic
  - Knowledge Representation
  - Knowledge Graph (KG)
- Uncertain Knowledge and Reasoning
  - Quantifying Uncertainty
  - Probabilistic Reasoning
  - Making Complex Decisions

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

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

Artificial Intelligence:
3. Knowledge and Reasoning

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

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

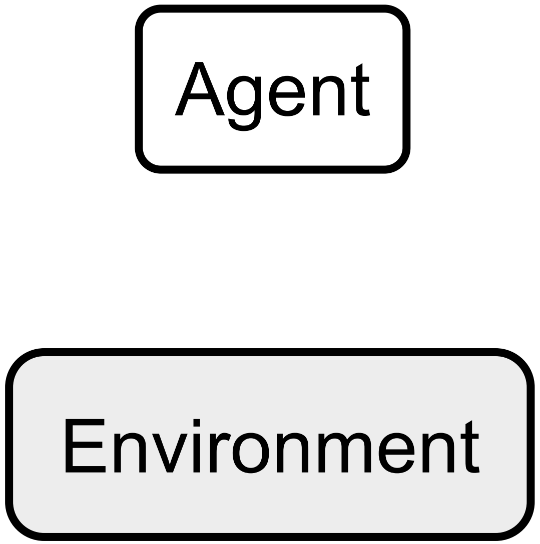
Intelligent Agents

4 Approaches of AI

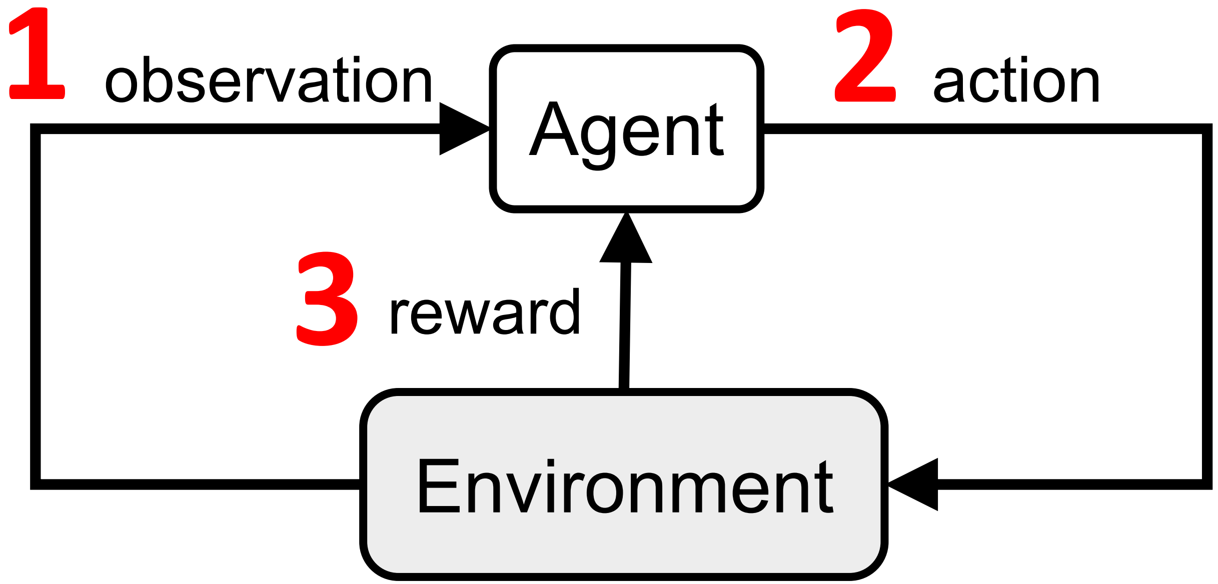
2. Thinking Humanly: The Cognitive Modeling Approach	3. Thinking Rationally: The "Laws of Thought" Approach
1. Acting Humanly: The Turing Test Approach (1950)	4. Acting Rationally: The Rational Agent Approach

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

Reinforcement Learning (DL)

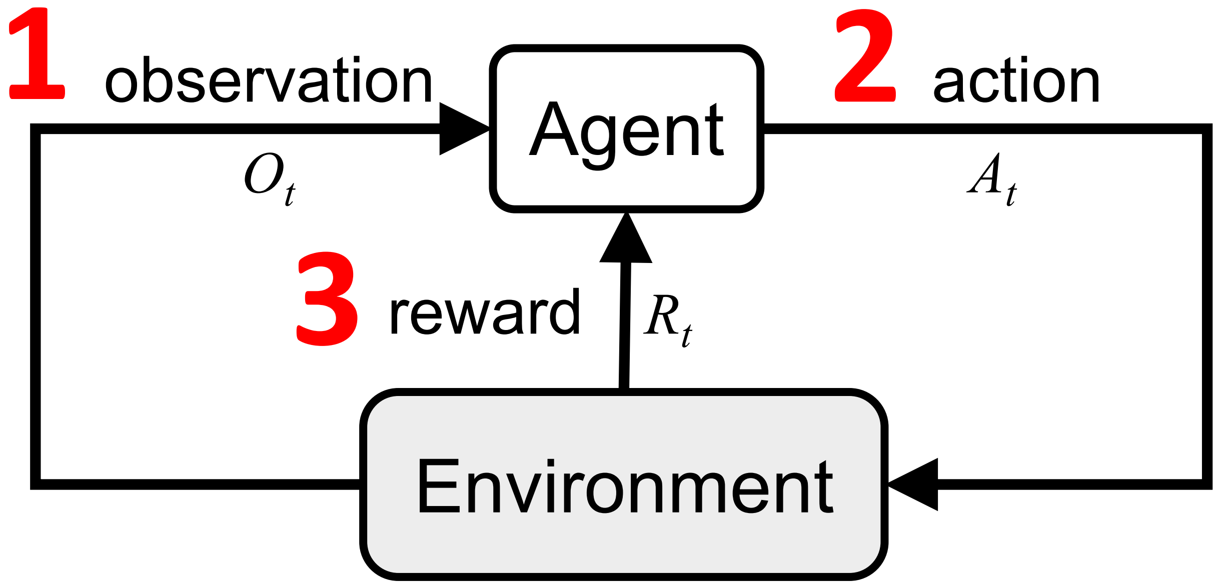


Reinforcement Learning (DL)



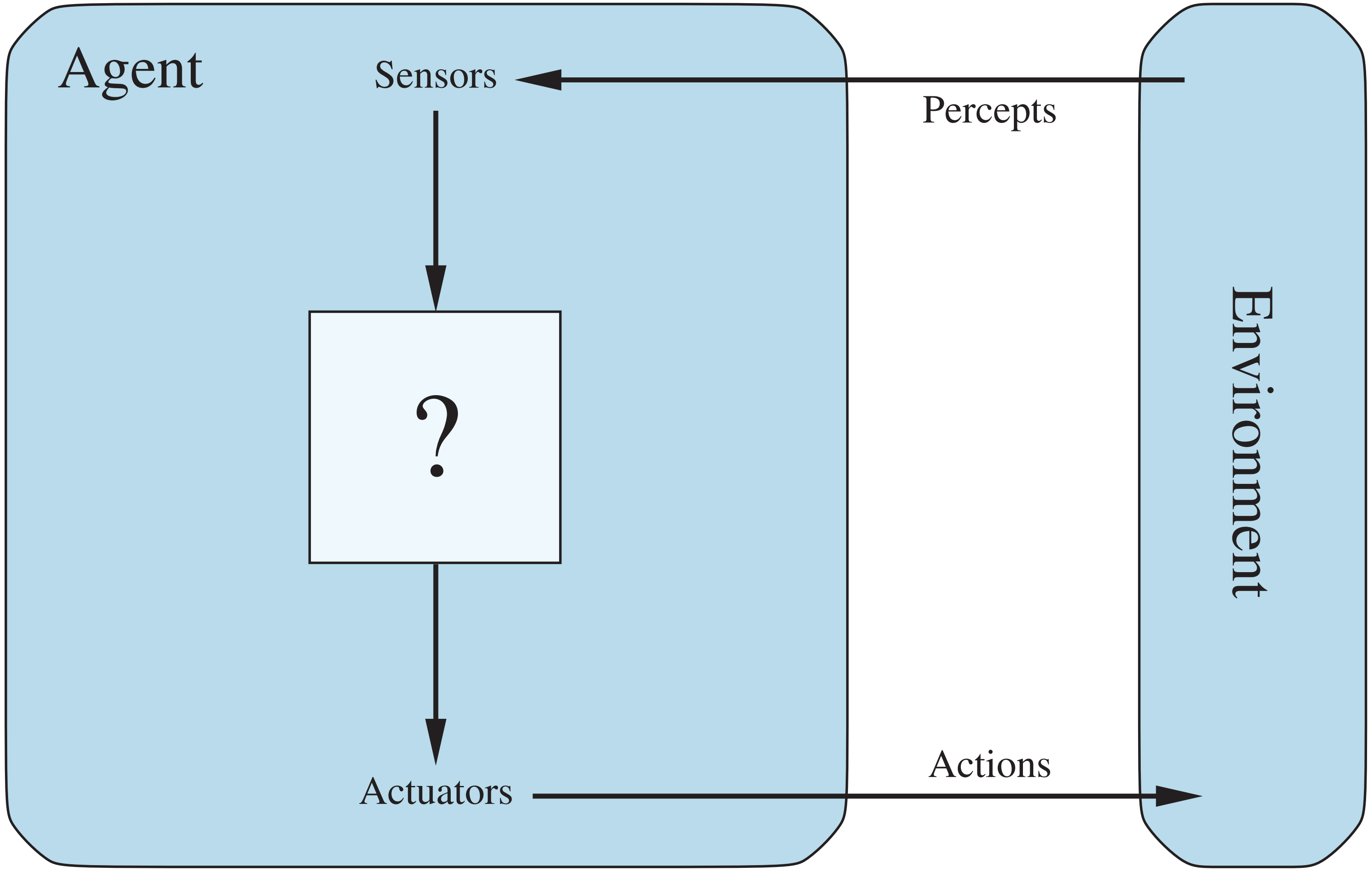
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning (DL)



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

Agents interact with environments through sensors and actuators



Logical Agents

Logical Agents

Knowledge-based Agents

KB Agents

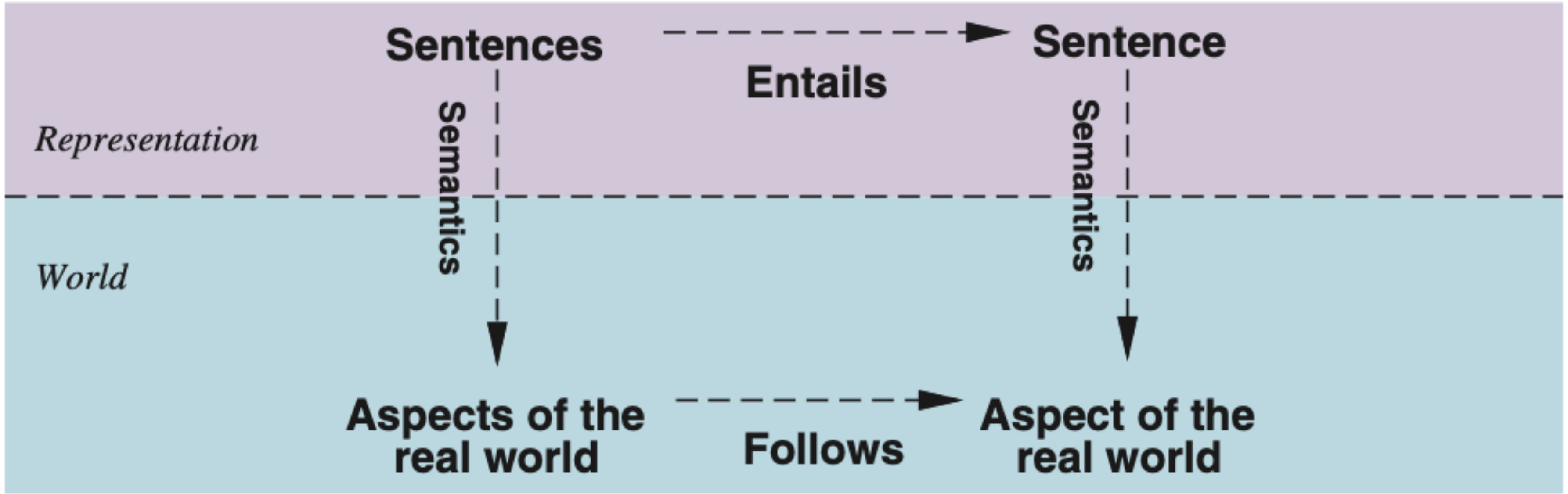
Knowledge-based Agent (KB Agent)

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

$$
\begin{array}{rcl}
Sentence & \rightarrow & AtomicSentence \mid ComplexSentence \\
AtomicSentence & \rightarrow & True \mid False \mid P \mid Q \mid R \mid \ldots \\
ComplexSentence & \rightarrow & (\ Sentence\ ) \\
& \mid & \neg\ Sentence \\
& \mid & Sentence \land Sentence \\
& \mid & Sentence \lor Sentence \\
& \mid & Sentence \ \Rightarrow \ Sentence \\
& \mid & Sentence \ \Leftrightarrow \ Sentence \\
\\
\textsc{Operator Precedence} & : & \neg, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

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	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
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*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
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.

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

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

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

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 $KB \wedge \neg\alpha$
  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

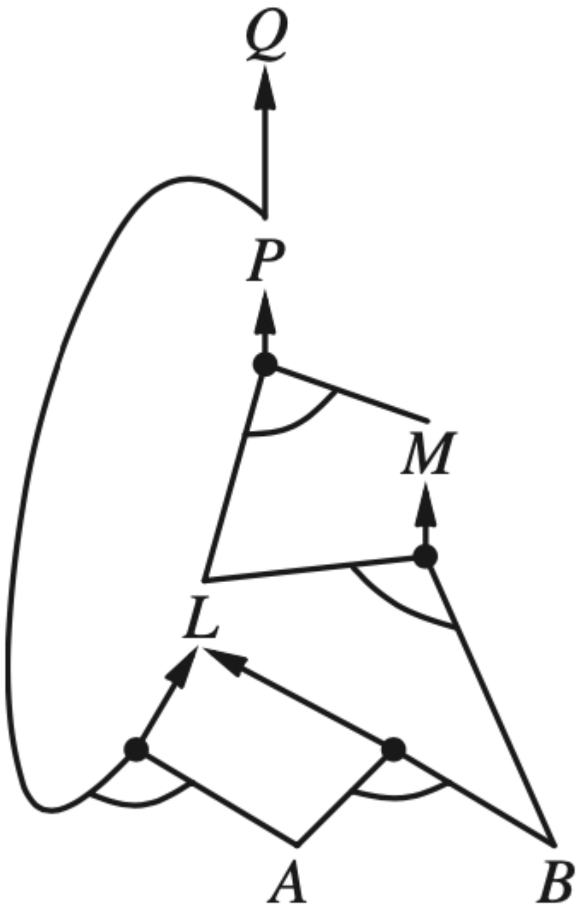
```
function PL-FC-ENTAILS?(KB, q) returns true or false
  inputs: KB, the knowledge base, a set of propositional definite clauses
          q, the query, a proposition symbol
  count ← a table, where count[c] is initially the number of symbols in clause c's premise
  inferred ← a table, where inferred[s]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

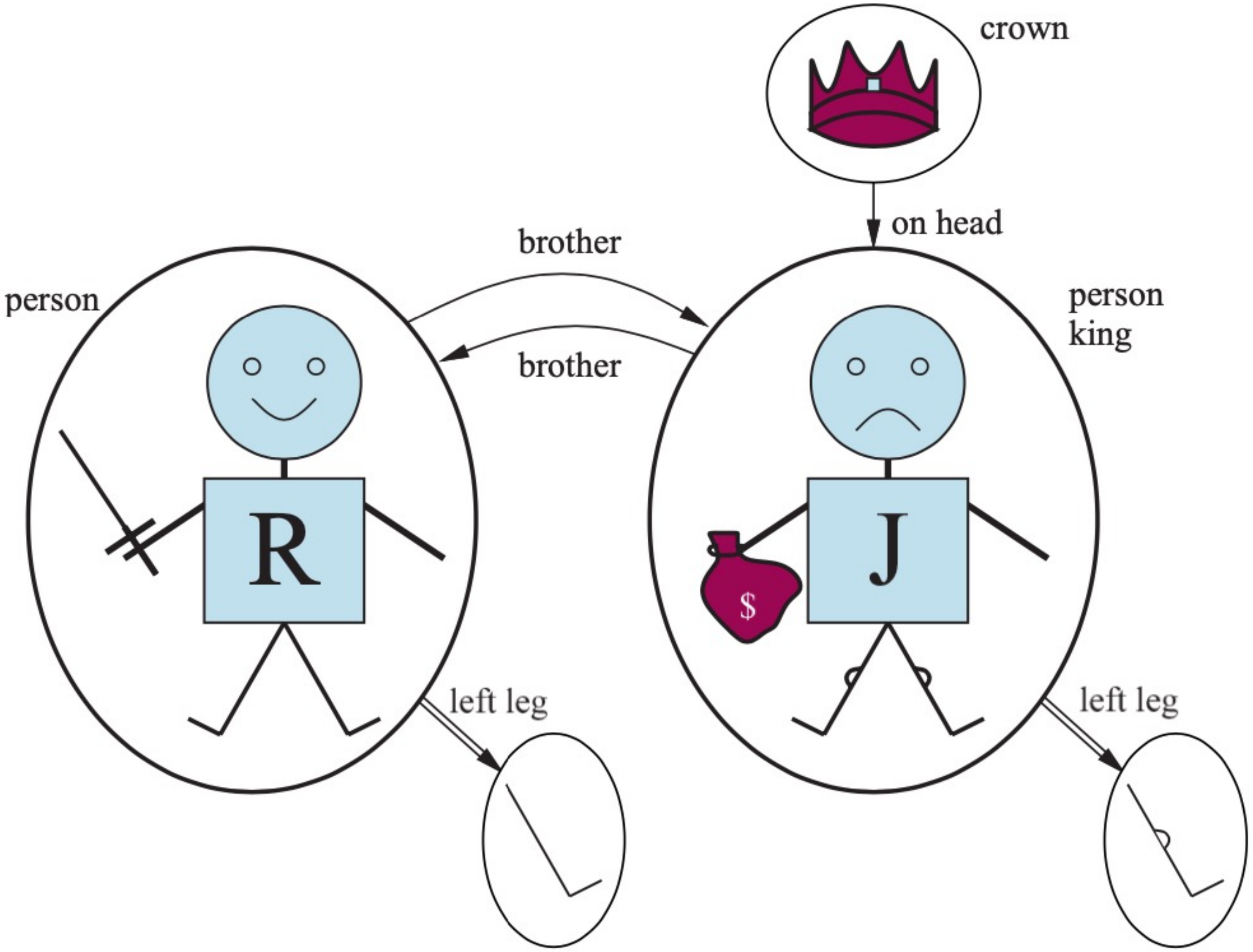
Formal languages and their ontological and epistemological commitments

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).



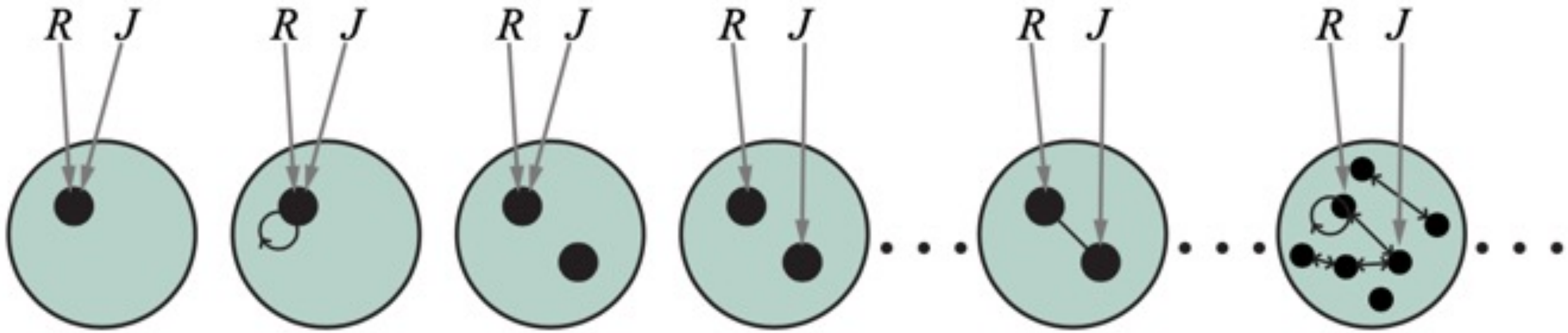
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

The syntax of first-order logic with equality

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

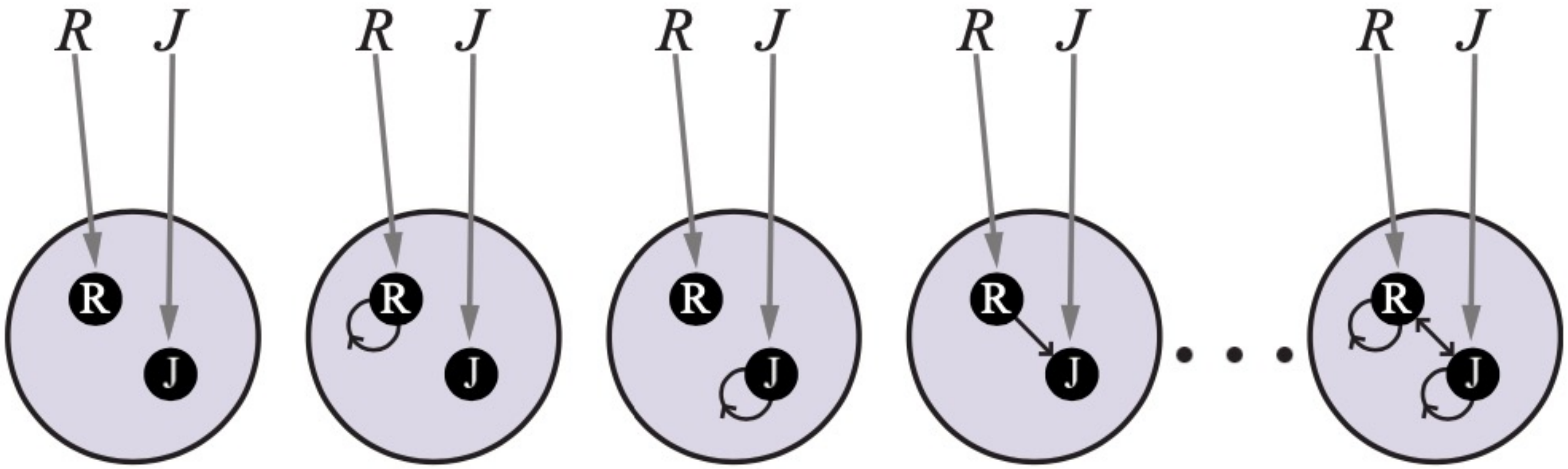
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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



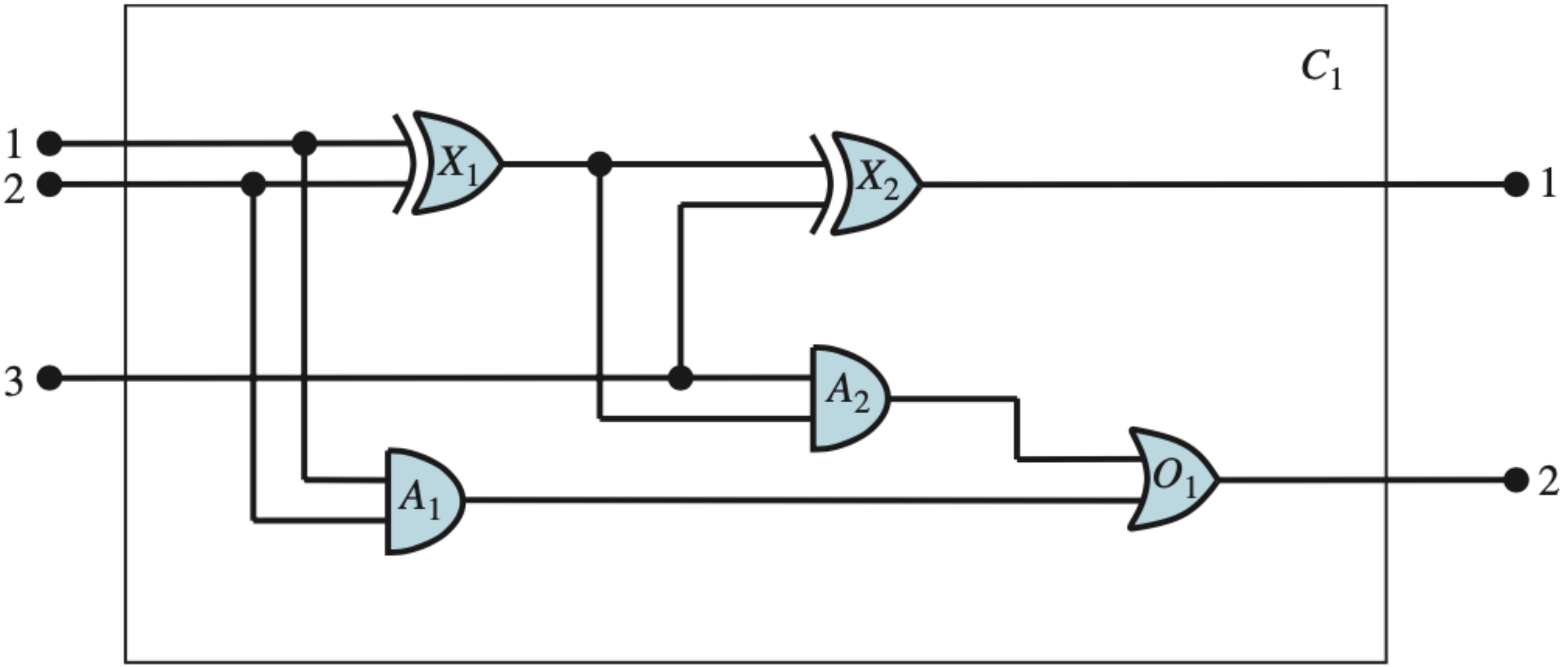
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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

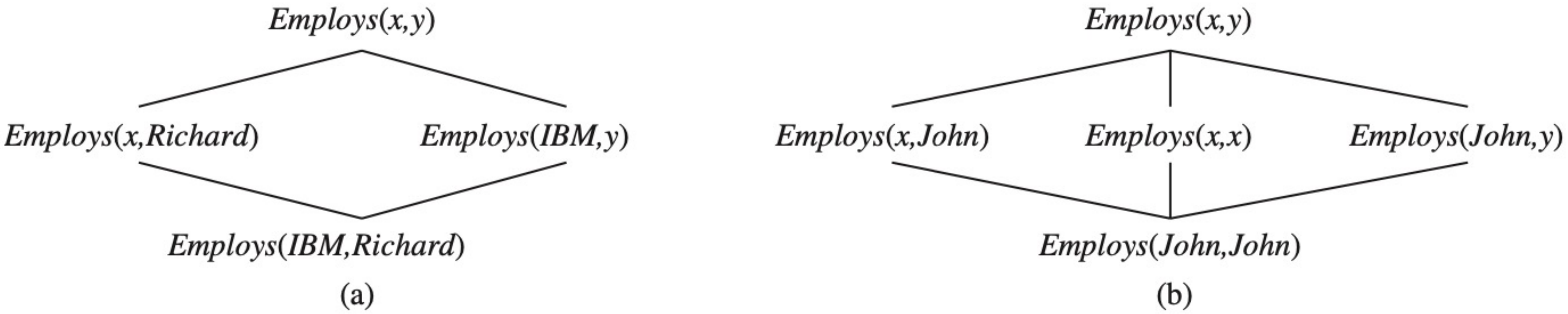
function UNIFY(x, y, θ=*empty*) **returns** a substitution to make x and y identical, or *failure*
  if θ = *failure* **then return** *failure*
  else if $x = y$ **then return** θ
  else if VARIABLE?(x) **then return** UNIFY-VAR(x, y, θ)
  else if VARIABLE?(y) **then return** UNIFY-VAR(y, x, θ)
  else 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 $\{var/val\} \in \theta$ for some val **then return** UNIFY(val, x, θ)
  else if $\{x/val\} \in \theta$ 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

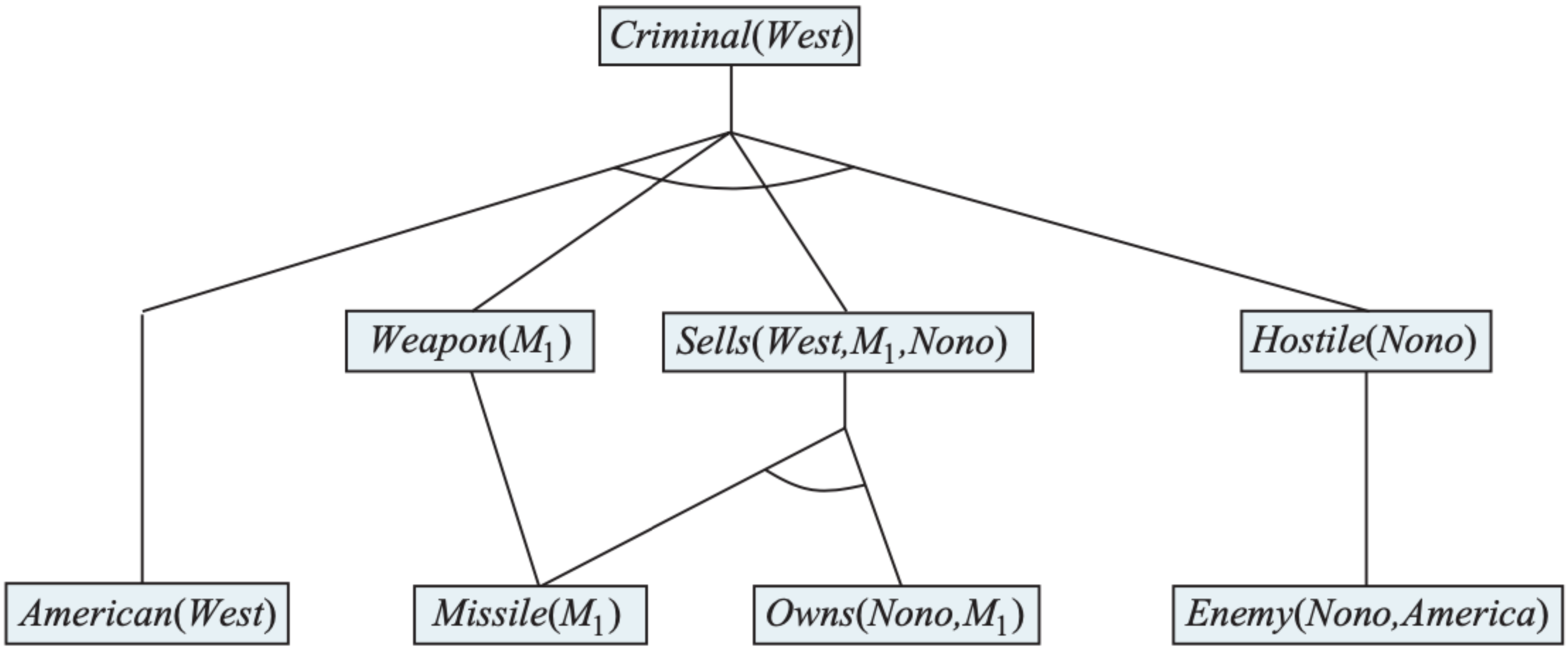
A conceptually straightforward, but inefficient, forward-chaining algorithm

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

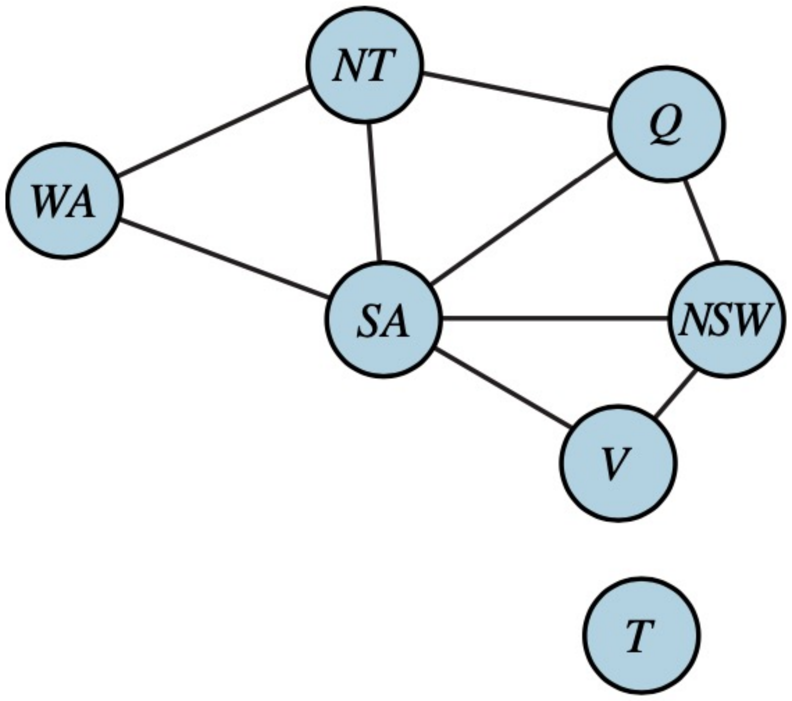
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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)

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

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

(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

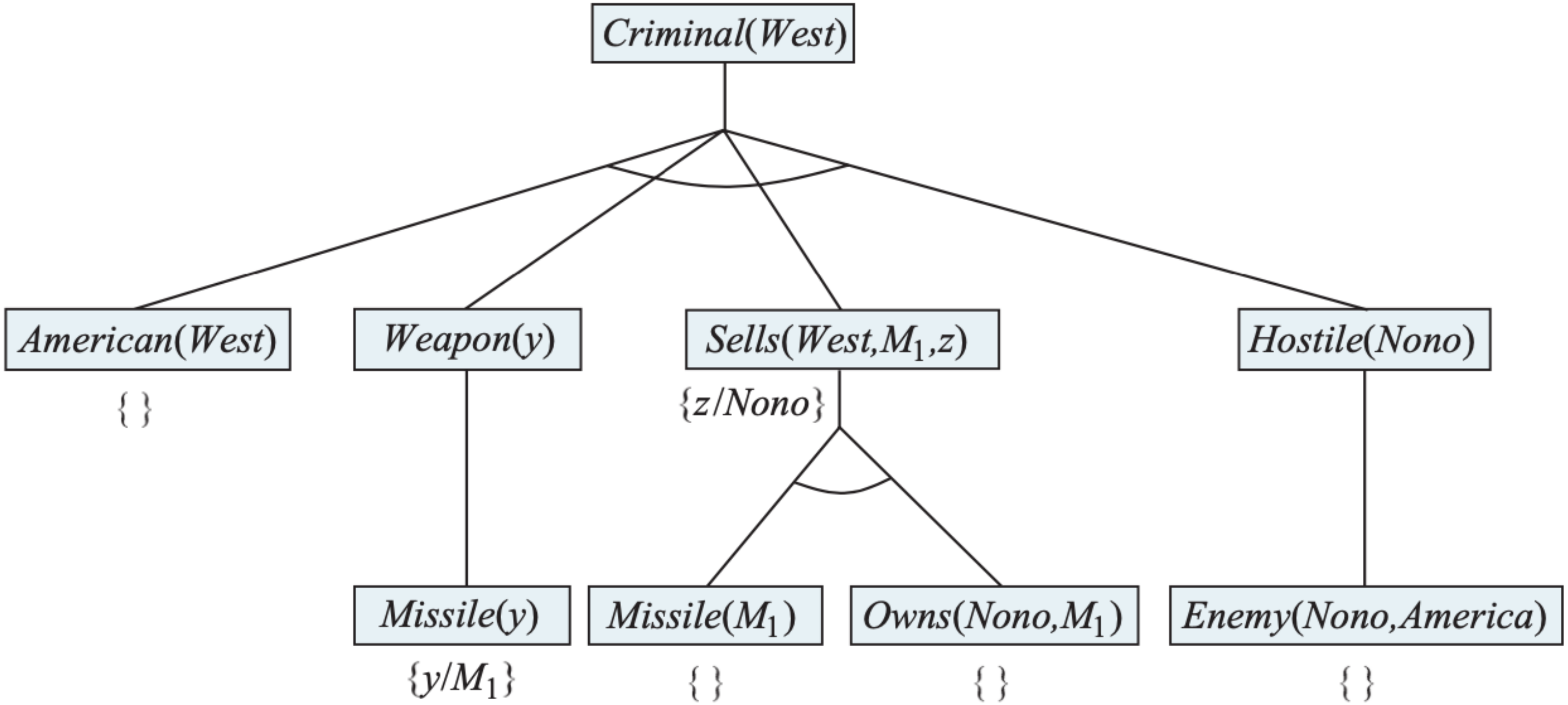
function FOL-BC-ASK(KB, $query$) **returns** a generator of substitutions
  return FOL-BC-OR(KB, $query$, { })

function FOL-BC-OR(KB, $goal$, θ) **returns** a substitution
  for each $rule$ in FETCH-RULES-FOR-GOAL(KB, $goal$) **do**
    ($lhs \Rightarrow rhs$) ← 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 $\theta = failure$ **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

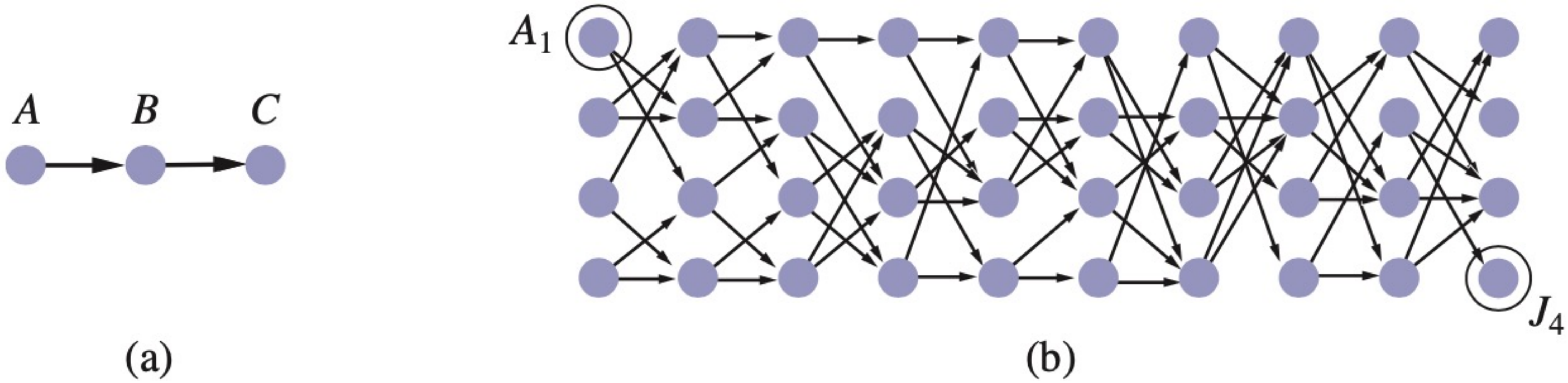
Pseudocode representing the result of compiling the Append predicate

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, $continuation$)

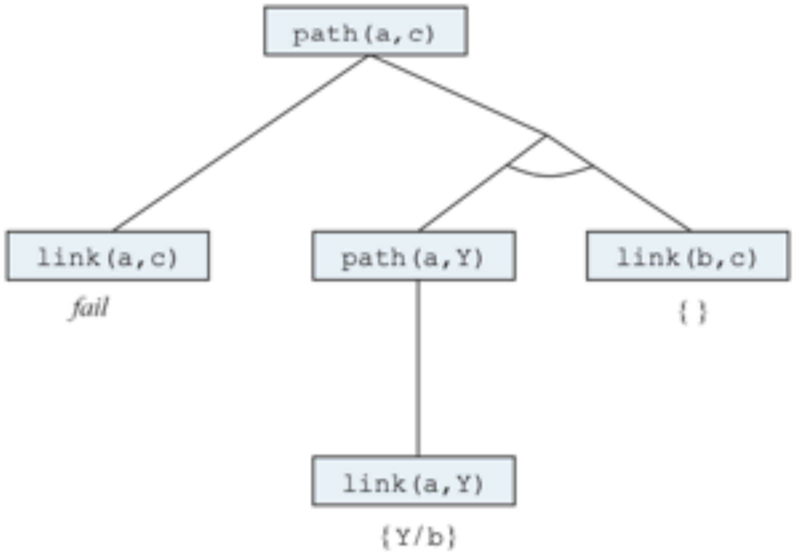
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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

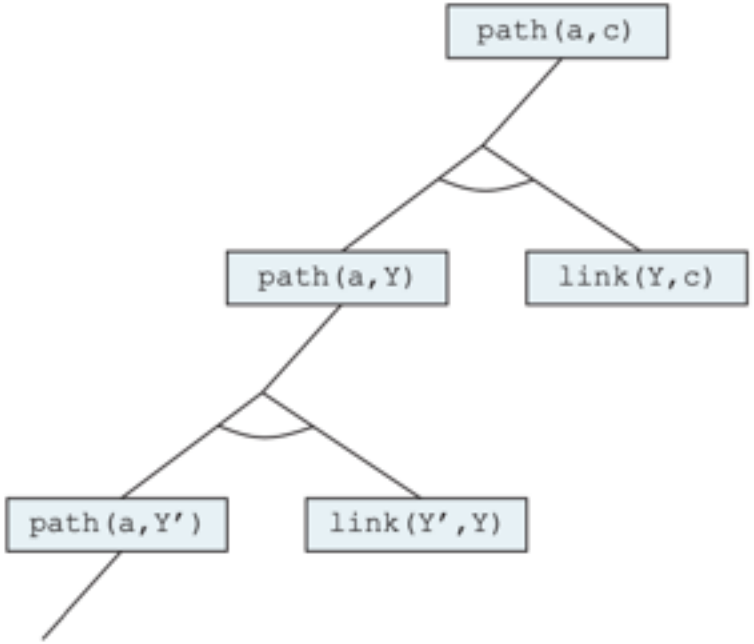


Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Proof that a path exists from A to C.



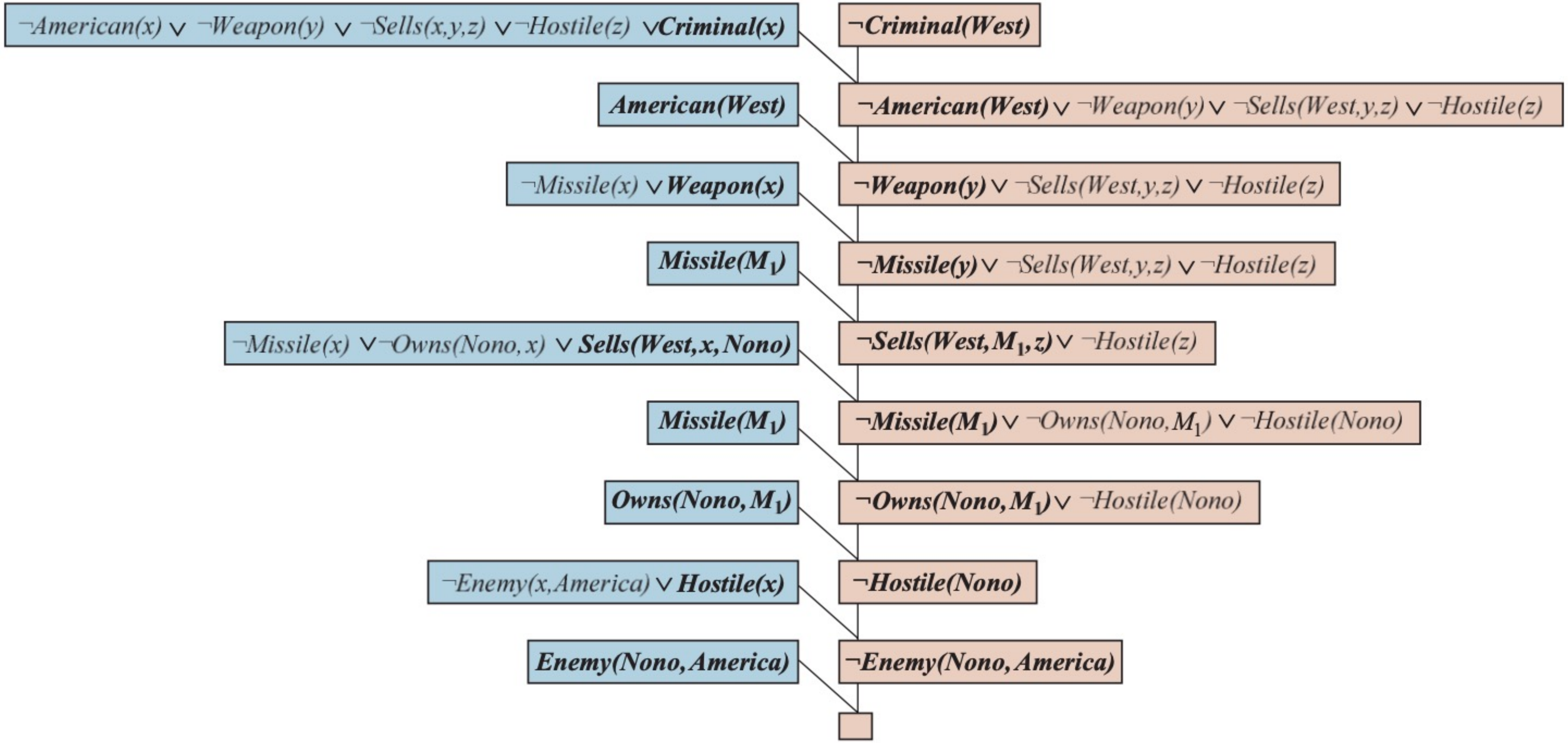
(a)



(b)

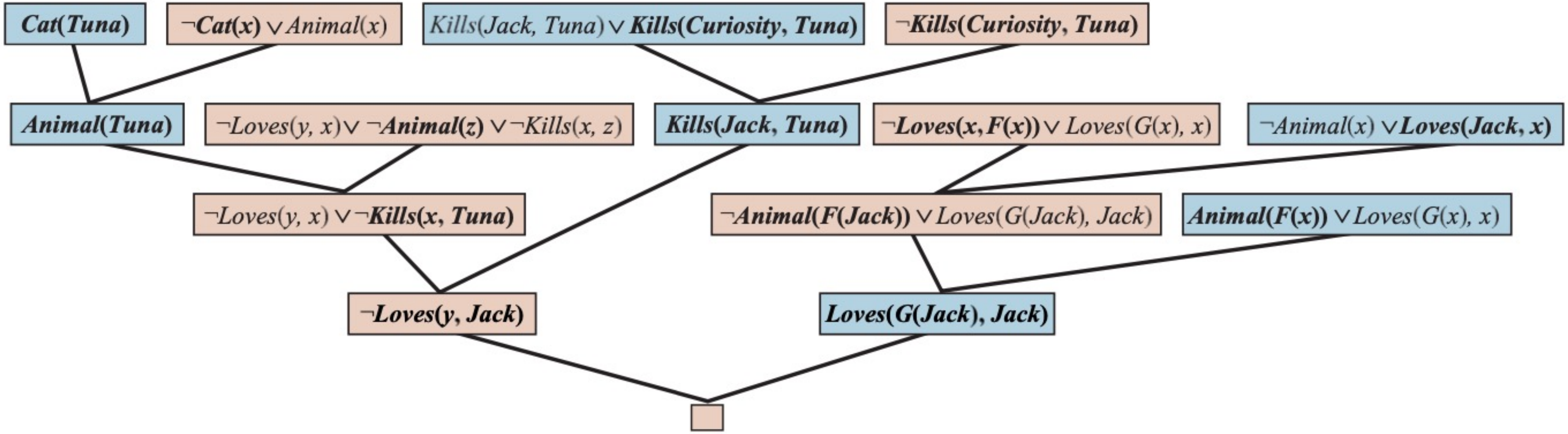
Infinite proof tree generated
when the clauses are
in the “wrong” order

A resolution proof that West is a criminal



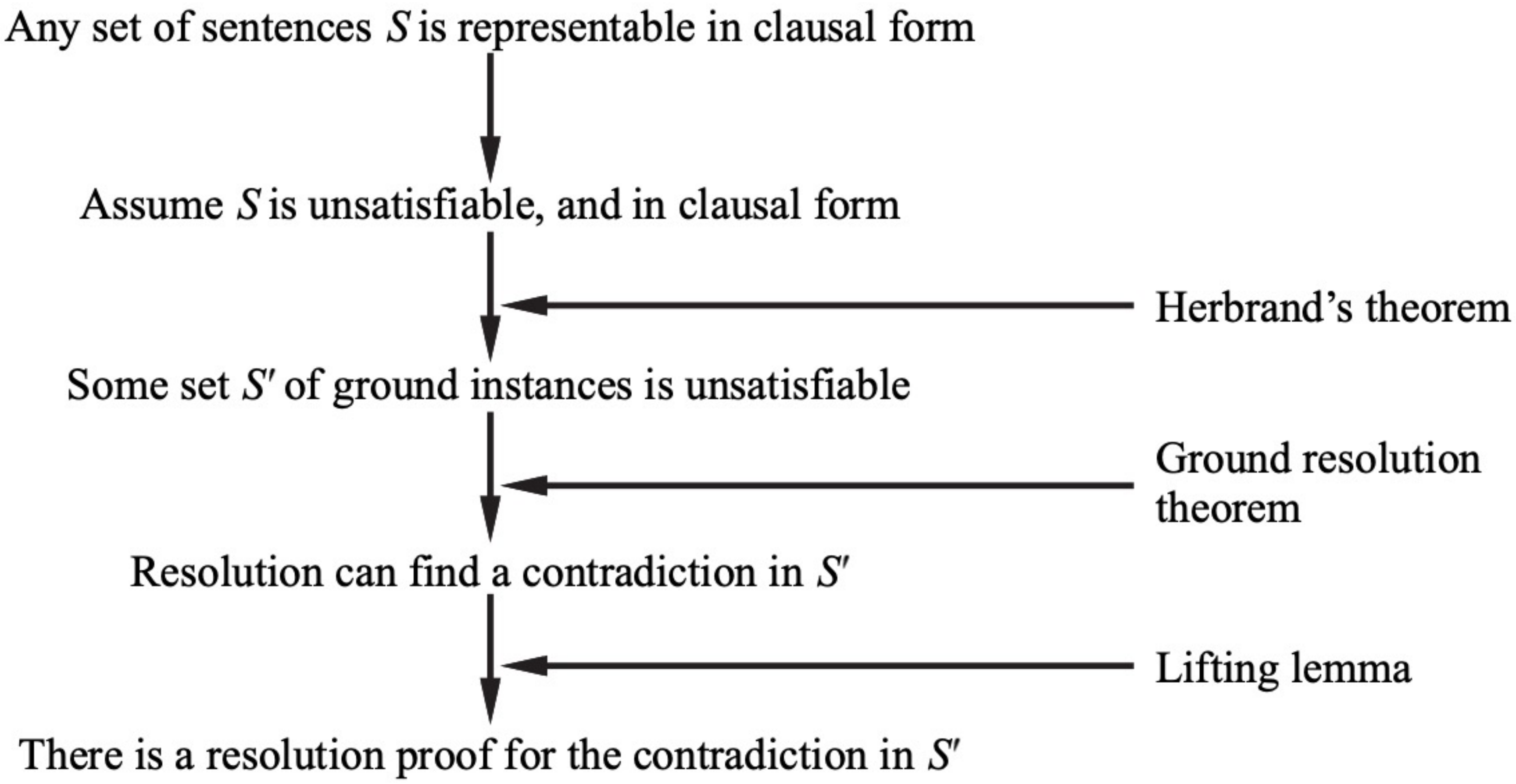
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A resolution proof that Curiosity killed the cat



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

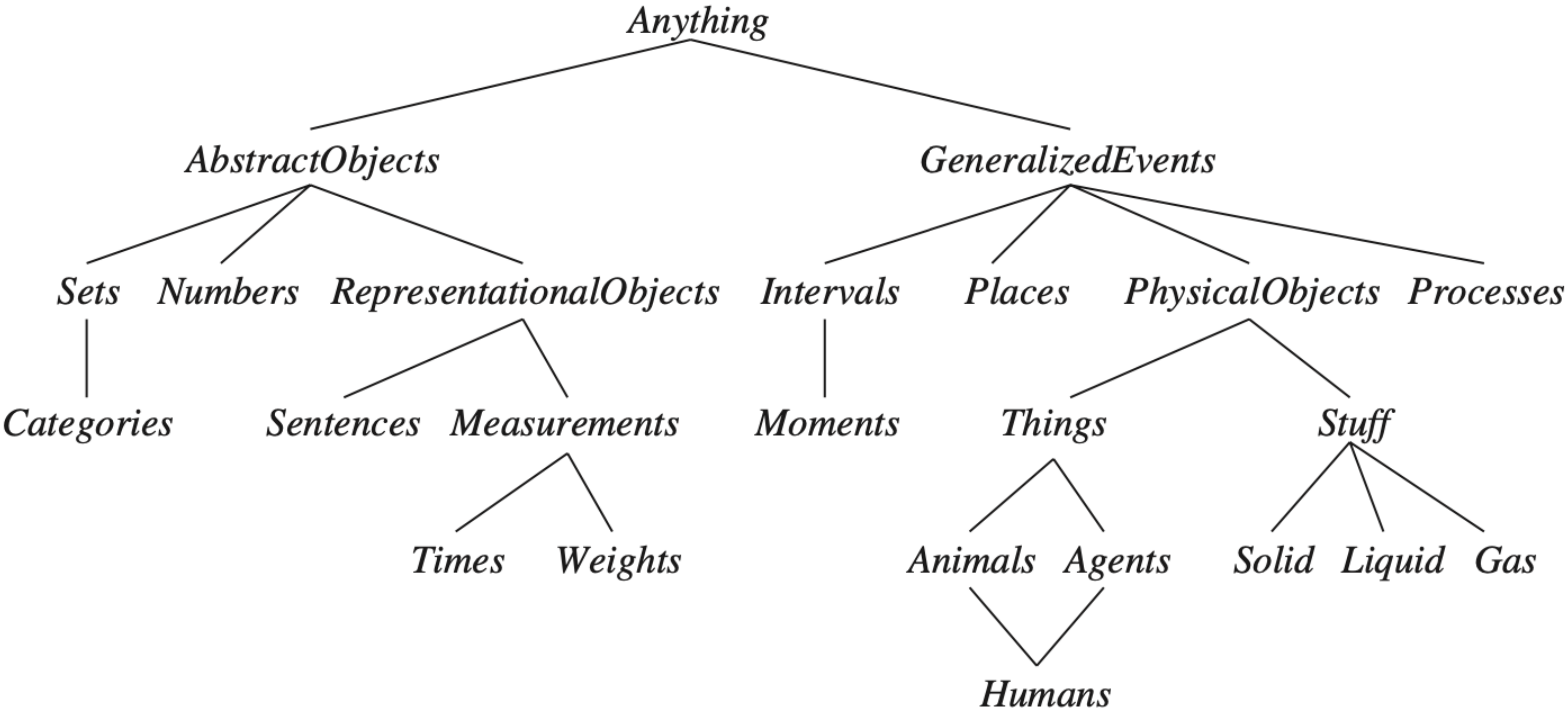
Structure of a completeness proof for resolution



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

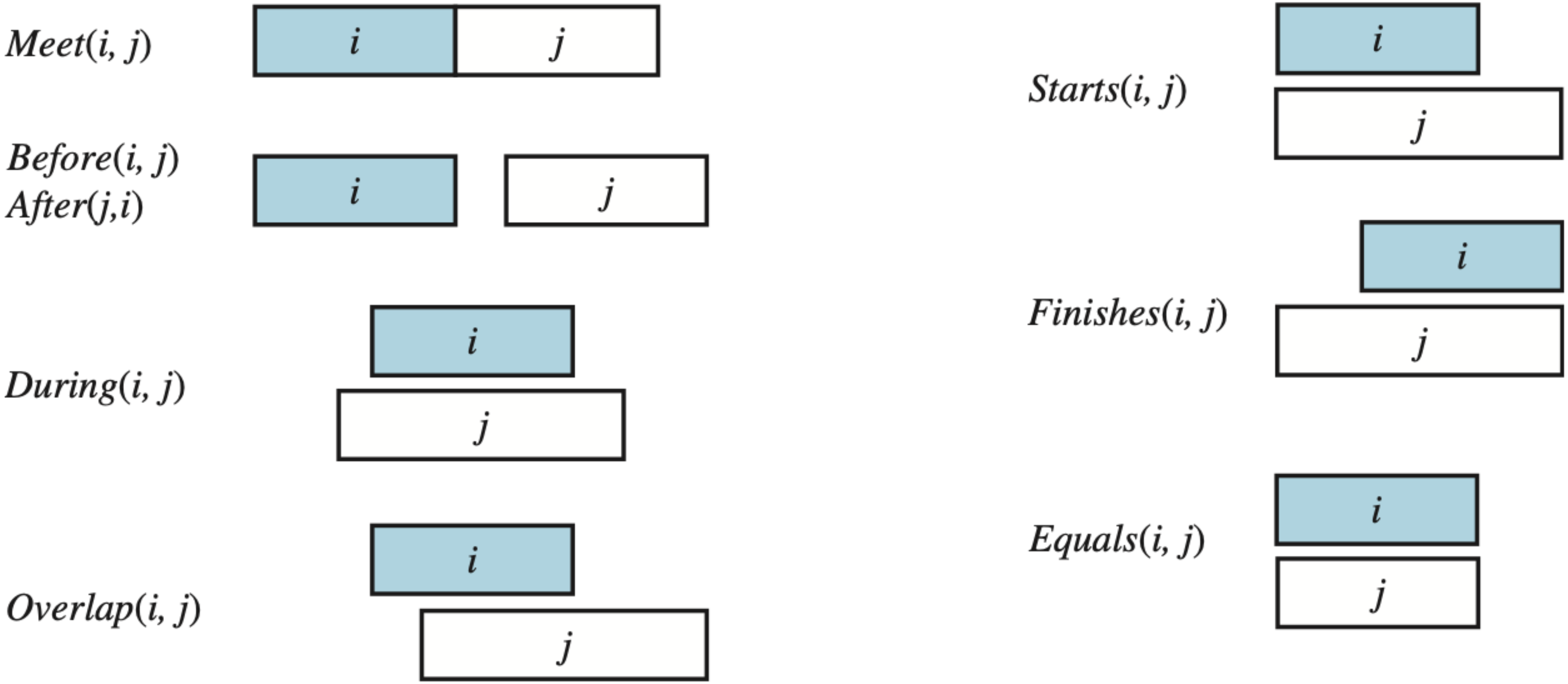
Knowledge Representation

The Upper Ontology of the World



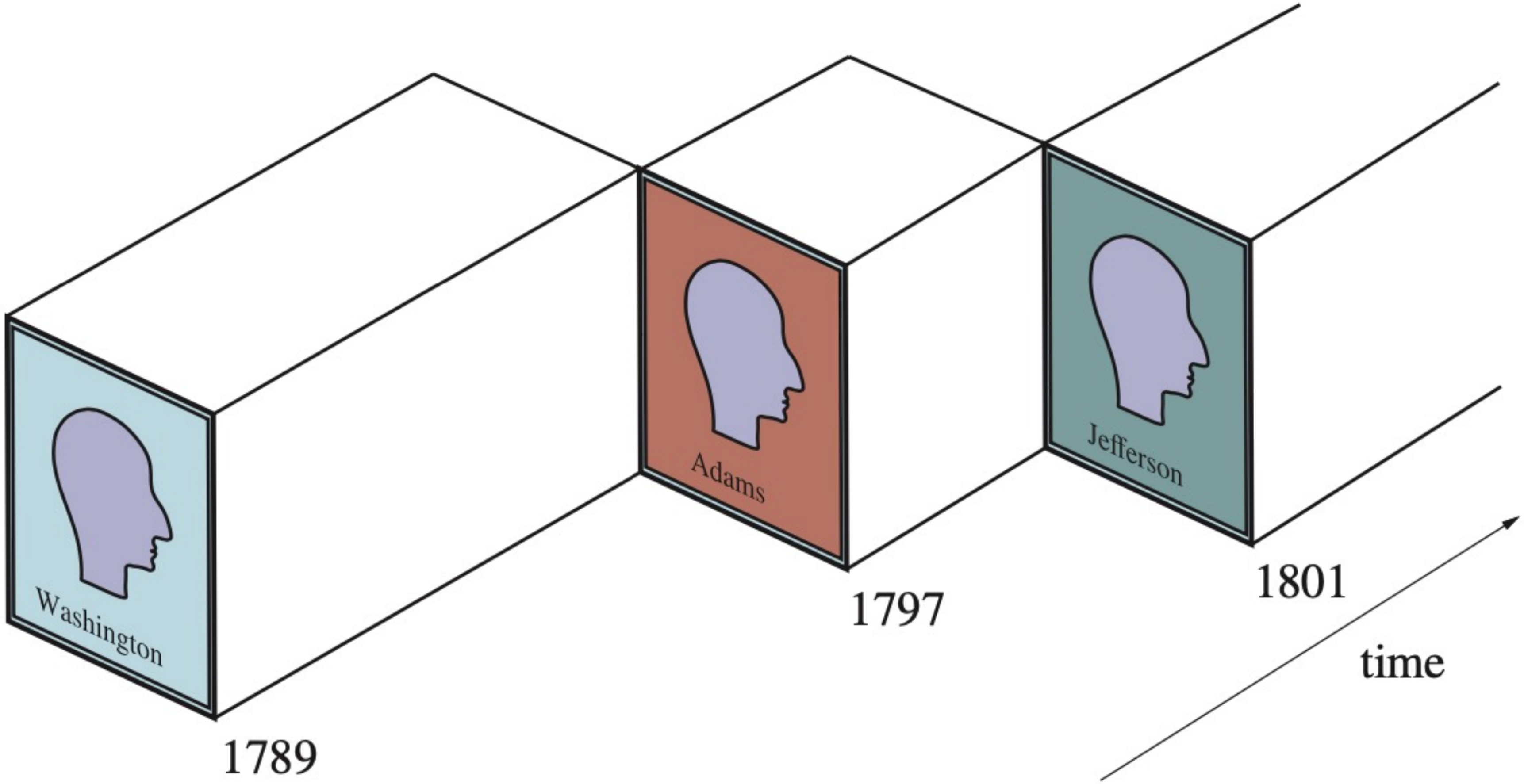
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Predicates on time intervals



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

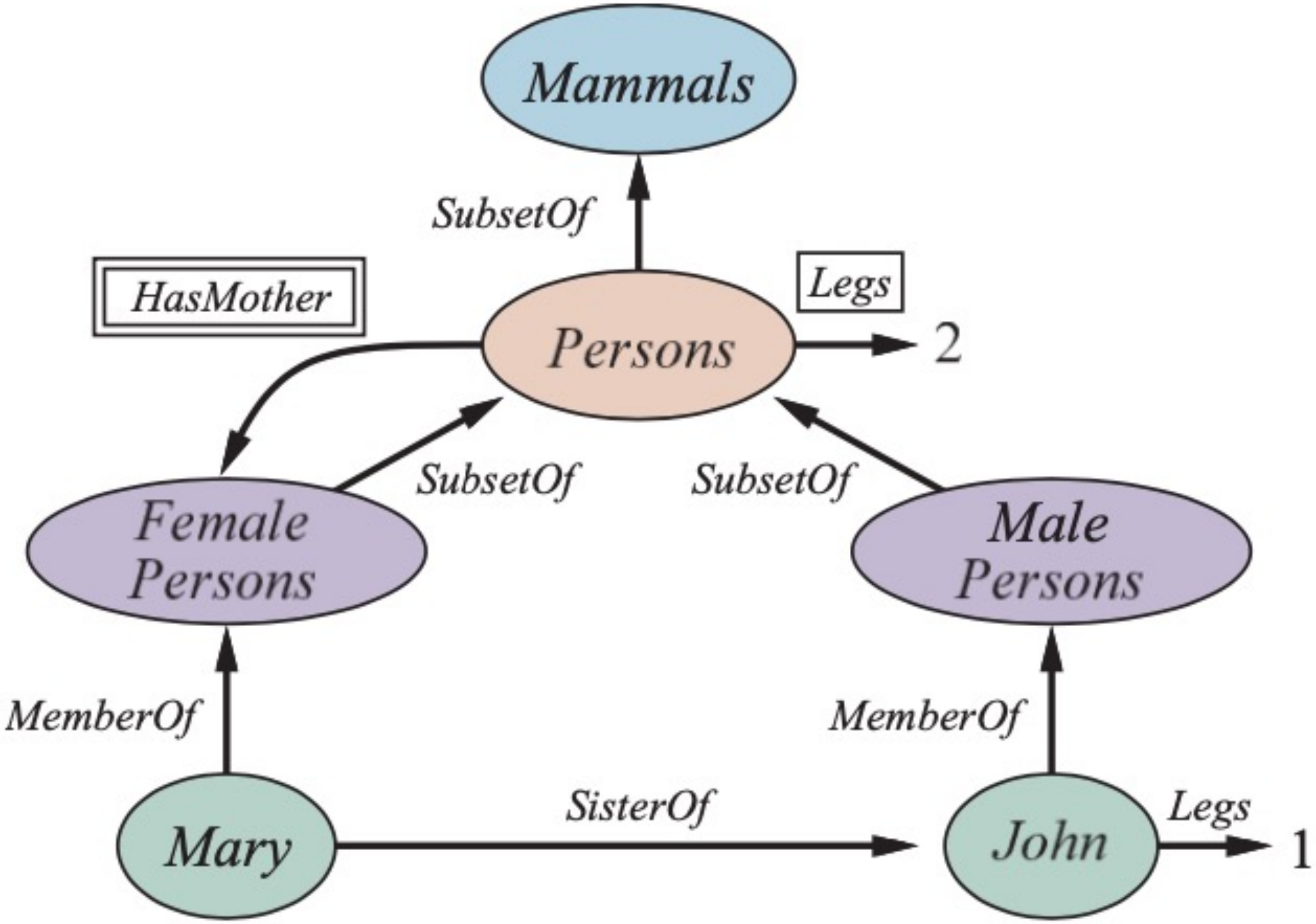
A schematic view of the object President (USA) for the early years



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

A semantic network

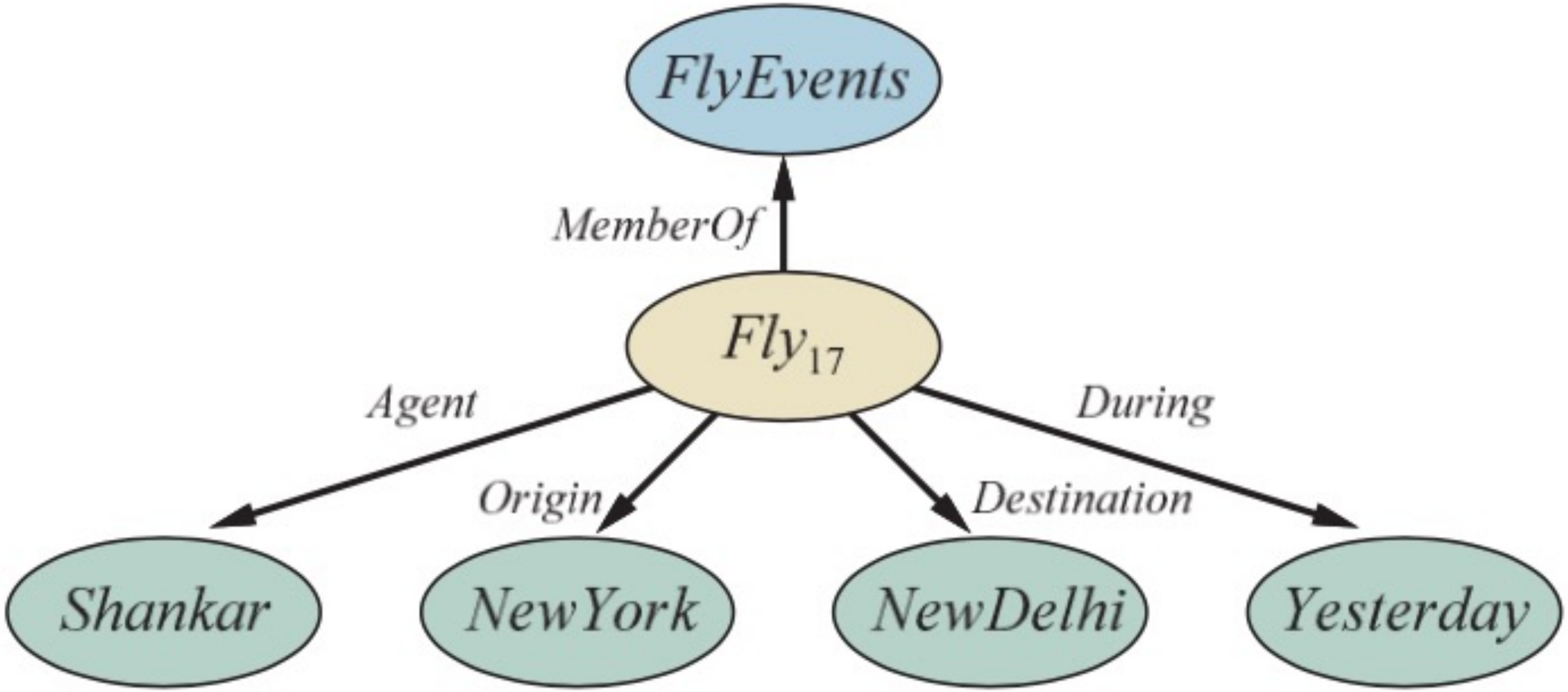
with four objects (John, Mary, 1, and 2) and four categories Relations are denoted by labeled links



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.

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

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

Knowledge Graph (KG)

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

Knowledge Graph (KG)

- **Knowledge Graph (KG)**
  - **A knowledge graph is a multi-relational graph composed of entities and relations, which are regarded as nodes and different types of edges, respectively (Ji et al., 2021).**
  - **Represents knowledge as concepts (entities) and their relationships (Facts)**
  - **Triple of facts**
    - *SPO: (subject, predicate, object)*
    - *HRT: (head, relation, tail)*
- **Common Knowledge Graph: DBpedia, YAGO, Wikidata**

Knowledge Graph, Facts, Triple, Embedding

- G
  - Knowledge graph
- F
  - Set of facts
- (h, r, t)
  - Triple of head, relation, and tail
- $(\mathbf{h}, \mathbf{r}, \mathbf{t})$
  - Embedding of head, relation, and tail

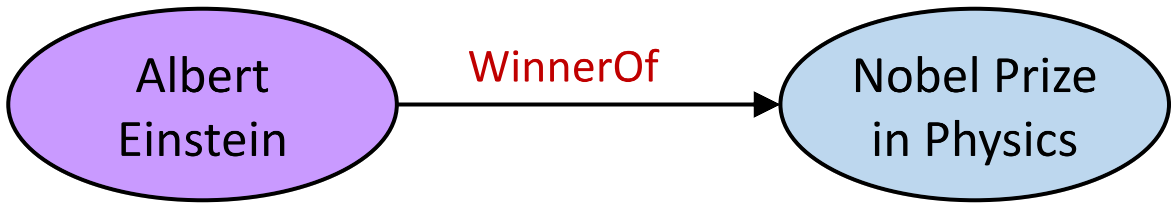
Source: Ji, S., Pan, S., Cambria, E., Marttinen, P., & Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems.

Knowledge Representation
Factual Triple and Knowledge Graph

- Albert Einstein, winner of the 1921 Nobel prize in physics
- The Nobel Prize in Physics 1921 was awarded to Albert Einstein "for his services to Theoretical Physics, and especially for his discovery of the law of the photoelectric effect."

Triple (Albert Einstein, **WinnerOf**, Nobel Prize in Physics)

Knowledge Graph



Factual Triples in Knowledge Base

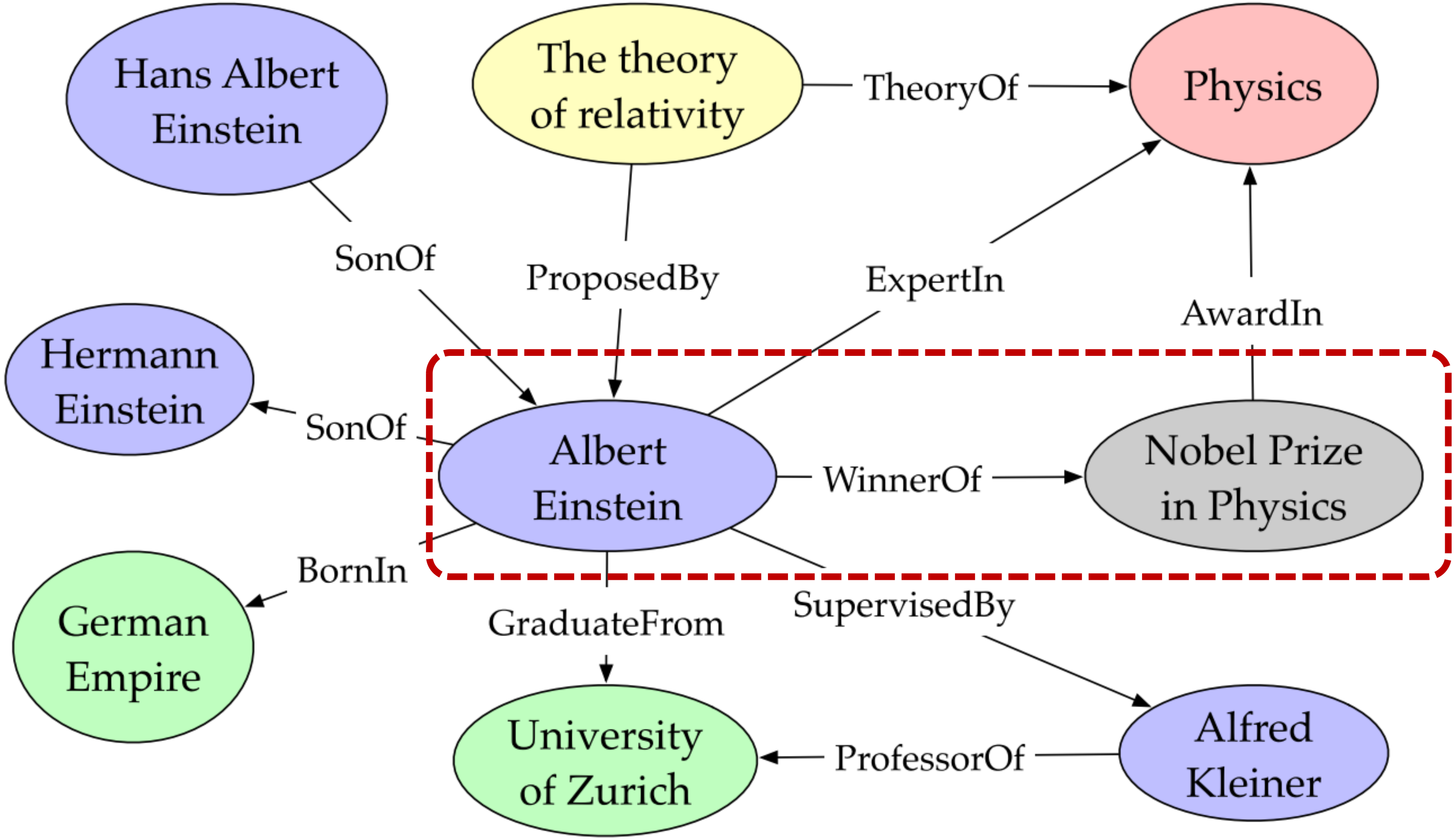
$$(h, r, t)$$

(Albert Einstein, **BornIn**, German Empire)
(Albert Einstein, **SonOf**, Hermann Einstein)
(Albert Einstein, **GraduateFrom**, University of Zurich)
(Albert Einstein, **WinnerOf**, Nobel Prize in Physics)
(Albert Einstein, **ExpertIn**, Physics)
(Nobel Prize in Physics, **AwardIn**, Physics)
(The theory of relativity, **TheoryOf**, Physics)
(Albert Einstein, **SupervisedBy**, Alfred Kleiner)
(Alfred Kleiner, **ProfessorOf**, University of Zurich)
(The theory of relativity, **ProposedBy**, Albert Einstein)
(Hans Albert Einstein, **SonOf**, Albert Einstein)

Source: Ji, S., Pan, S., Cambria, E., Marttinen, P., & Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems.

Entities and Relations in Knowledge Graph

(Albert Einstein, **WinnerOf**, Nobel Prize in Physics)



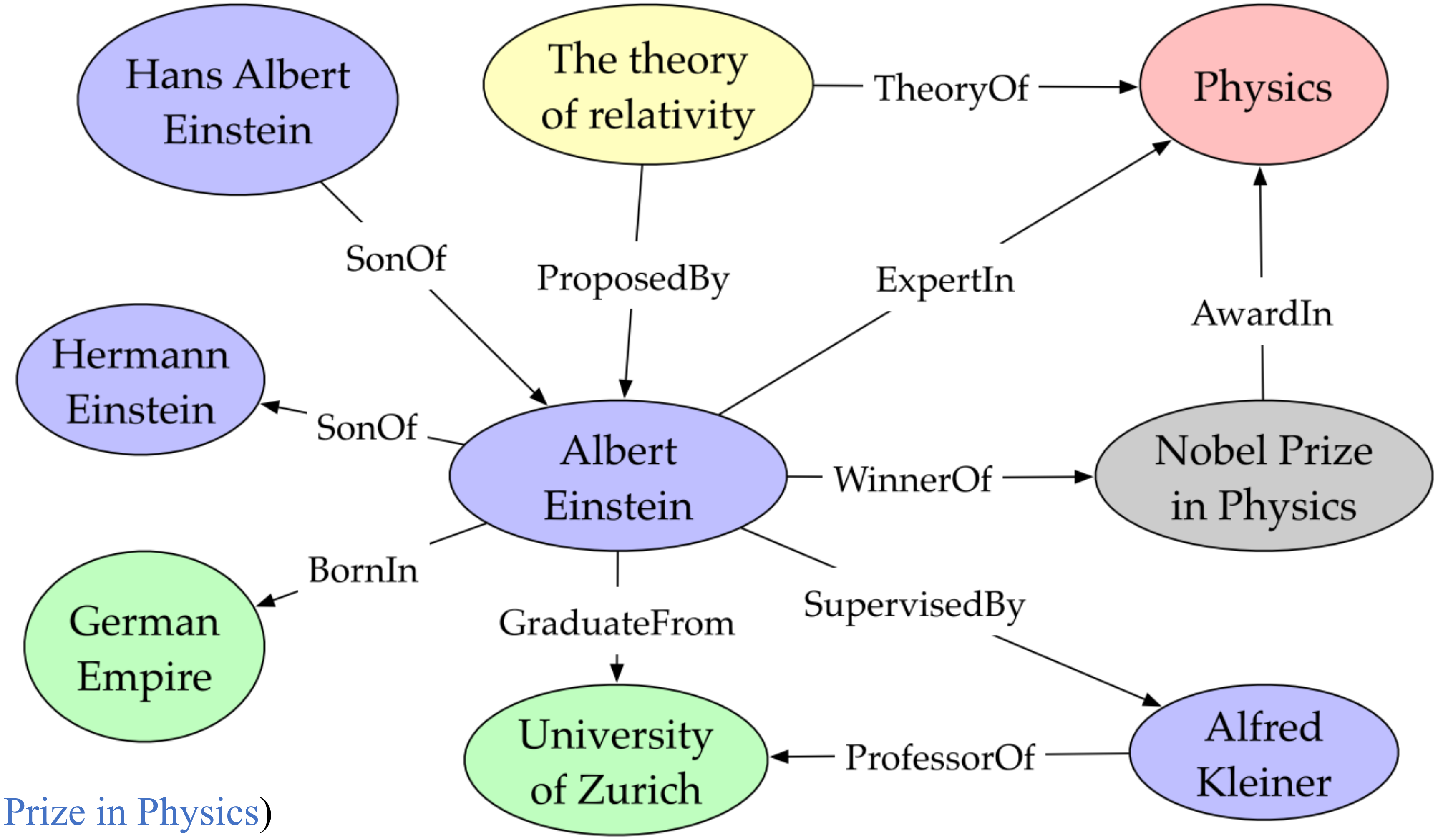
Source: Ji, S., Pan, S., Cambria, E., Marttinen, P., & Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems.

knowledge base and knowledge graph

Factual triples in knowledge base

(Albert Einstein, **BornIn**, German Empire)
(Albert Einstein, **SonOf**, Hermann Einstein)
(Albert Einstein, **GraduateFrom**, University of Zurich)
(Albert Einstein, **WinnerOf**, Nobel Prize in Physics)
(Albert Einstein, **ExpertIn**, Physics)
(Nobel Prize in Physics, **AwardIn**, Physics)
(The theory of relativity, **TheoryOf**, Physics)
(Albert Einstein, **SupervisedBy**, Alfred Kleiner)
(Alfred Kleiner, **ProfessorOf**, University of Zurich)
(The theory of relativity, **ProposedBy**, Albert Einstein)
(Hans Albert Einstein, **SonOf**, Albert Einstein)

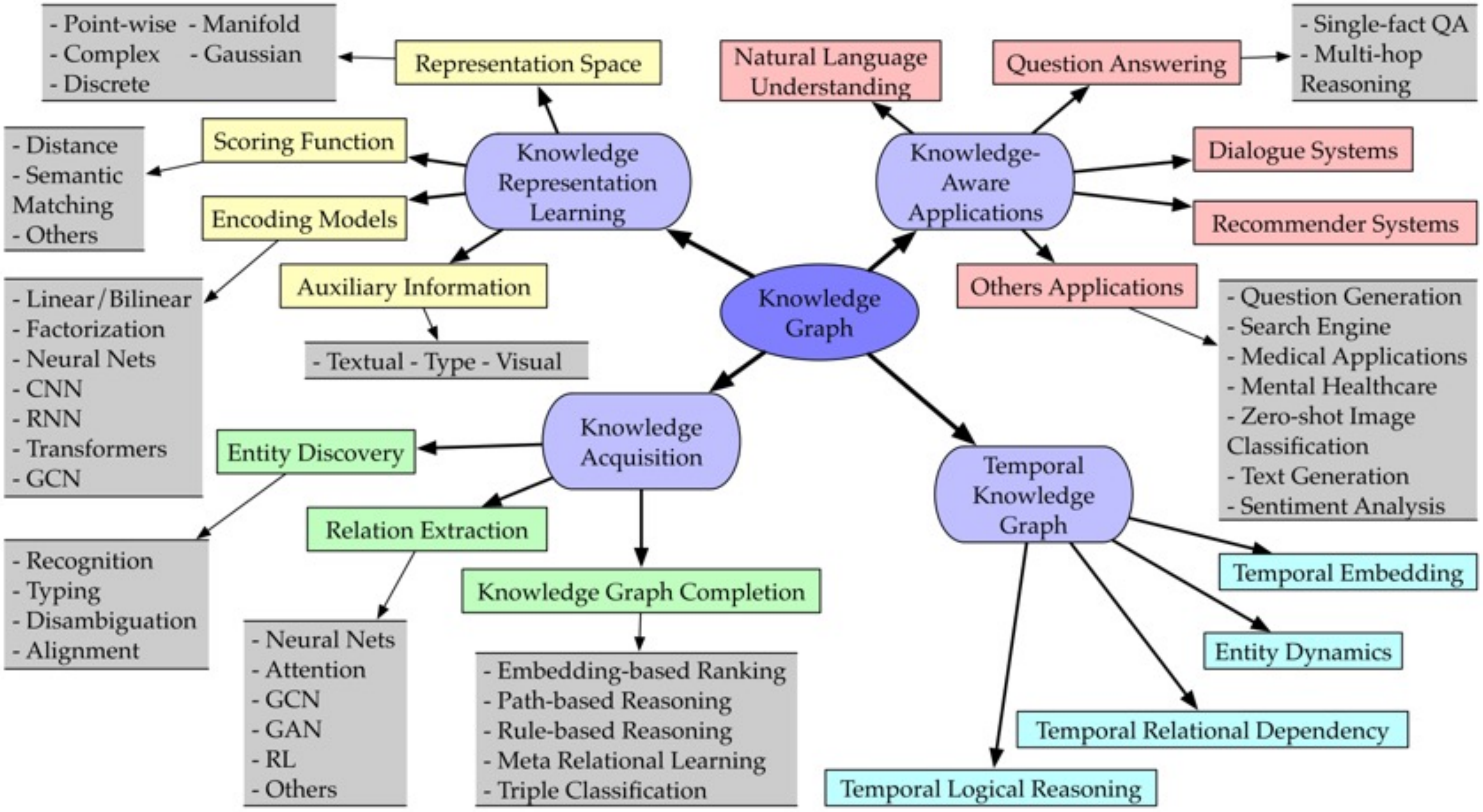
Entities and relations in knowledge graph



(Albert Einstein, **WinnerOf**, Nobel Prize in Physics)

Source: Ji, S., Pan, S., Cambria, E., Marttinen, P., & Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems.

Categorization of Research on Knowledge Graphs



Source: Ji, S., Pan, S., Cambria, E., Marttinen, P., & Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems.

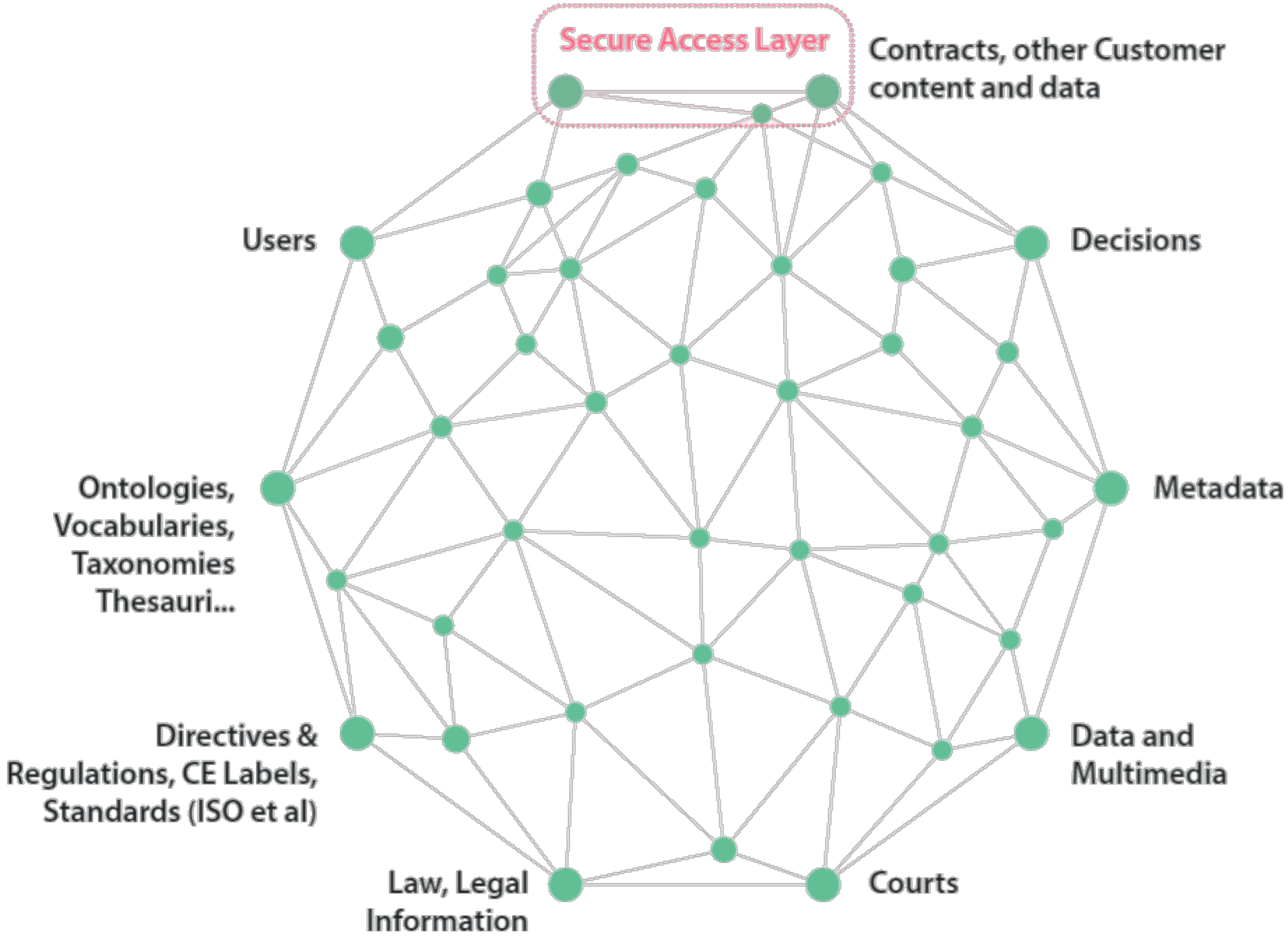
Knowledge Graph Completion (KGC) Datasets

Knowledge Graph Completion (KGC) Dataset	#Entity	#Relation	#Train	#Valid	#Test	Reference
WN18RR	40,943	11	86,835	3,034	3,134	Toutanova & Chen (2015); Zhang et al. (2020)
FB15k-237	14,541	237	272,115	17,535	20,466	Dettmers et al. (2018); Zhang et al. (2020)
YAGO3-10	123,182	37	1,079,040	5,000	5,000	Mahdisoltani et al. (2015); Zhang et al. (2020)

Domain-Specific Knowledge Graph

- **Domain-Specific Knowledge Graph**
  - **PubMed Knowledge Graph (PKG)**
    - **Extracting biological entities from 29 million PubMed abstracts**
  - **Lynx: Legal Knowledge Graph for Multilingual Compliance Services**
    - Legal Knowledge Graph (LKG) integrates and links heterogeneous compliance data sources including legislation, case law, standards and other private contracts.

Lynx: Legal Knowledge Graph for Multilingual Compliance Services



Source: Lynx Legal Knowledge Graph (LKG), https://lynx-project.eu/

Automated Planning

A PDDL description of an air cargo transportation planning problem

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

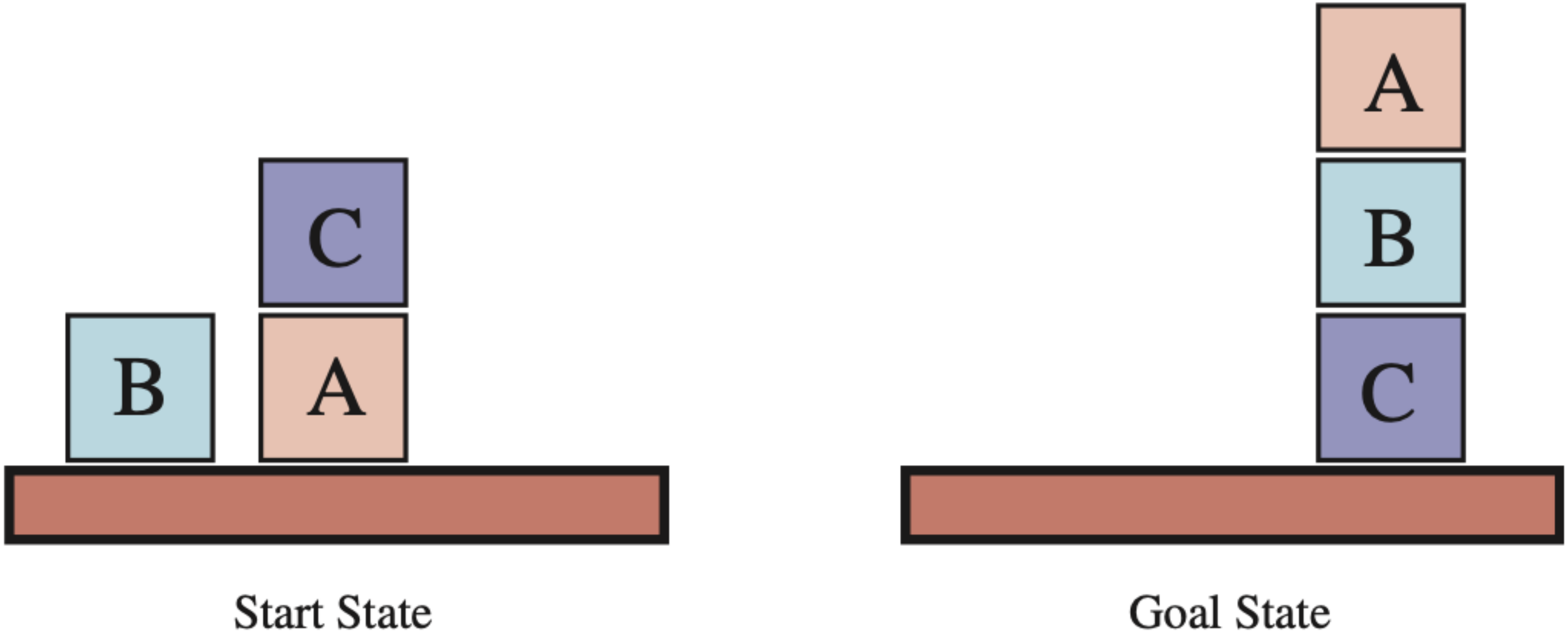
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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

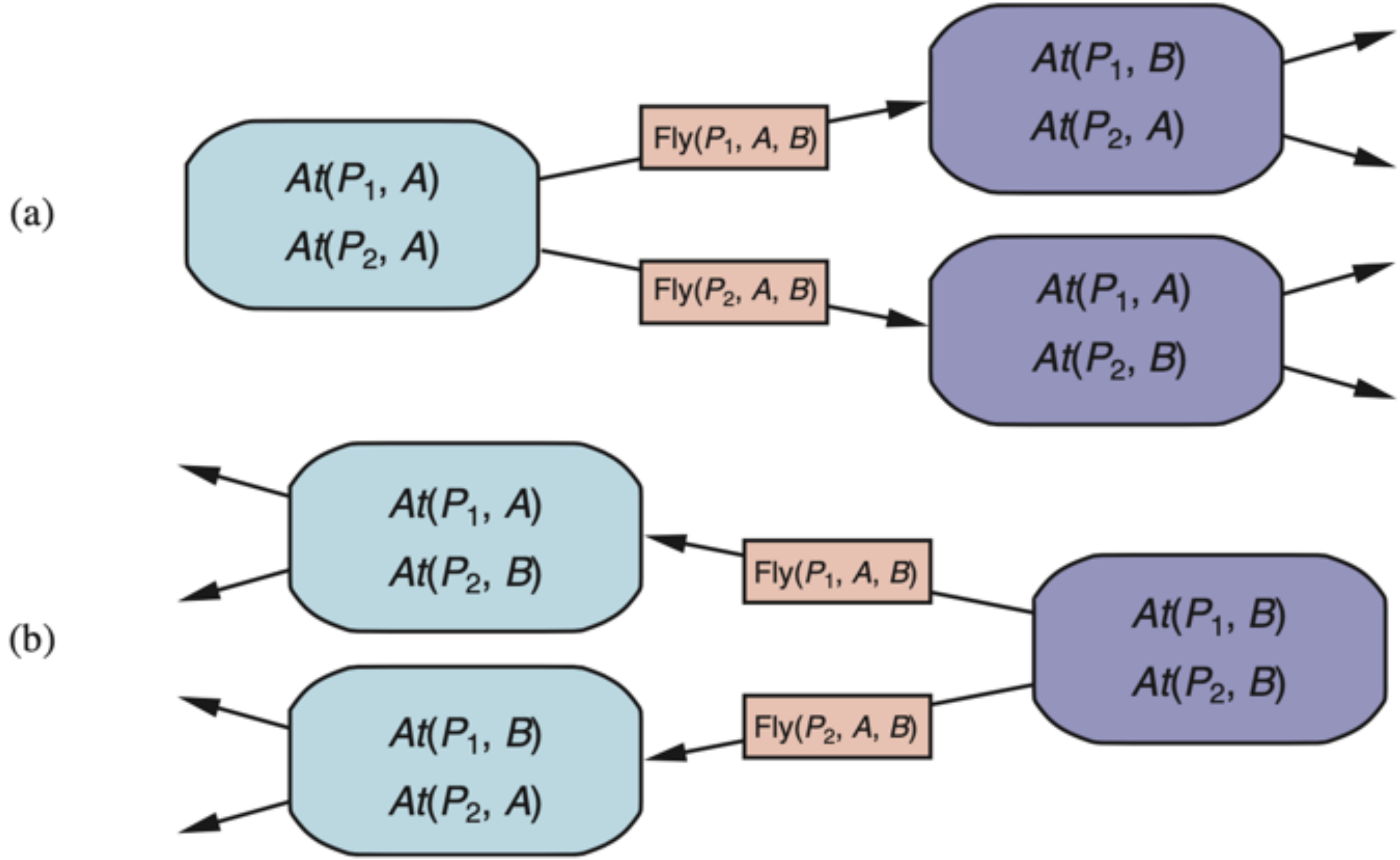
A planning problem in the blocks world: building a three-block tower

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

Two approaches to searching for a plan (a)

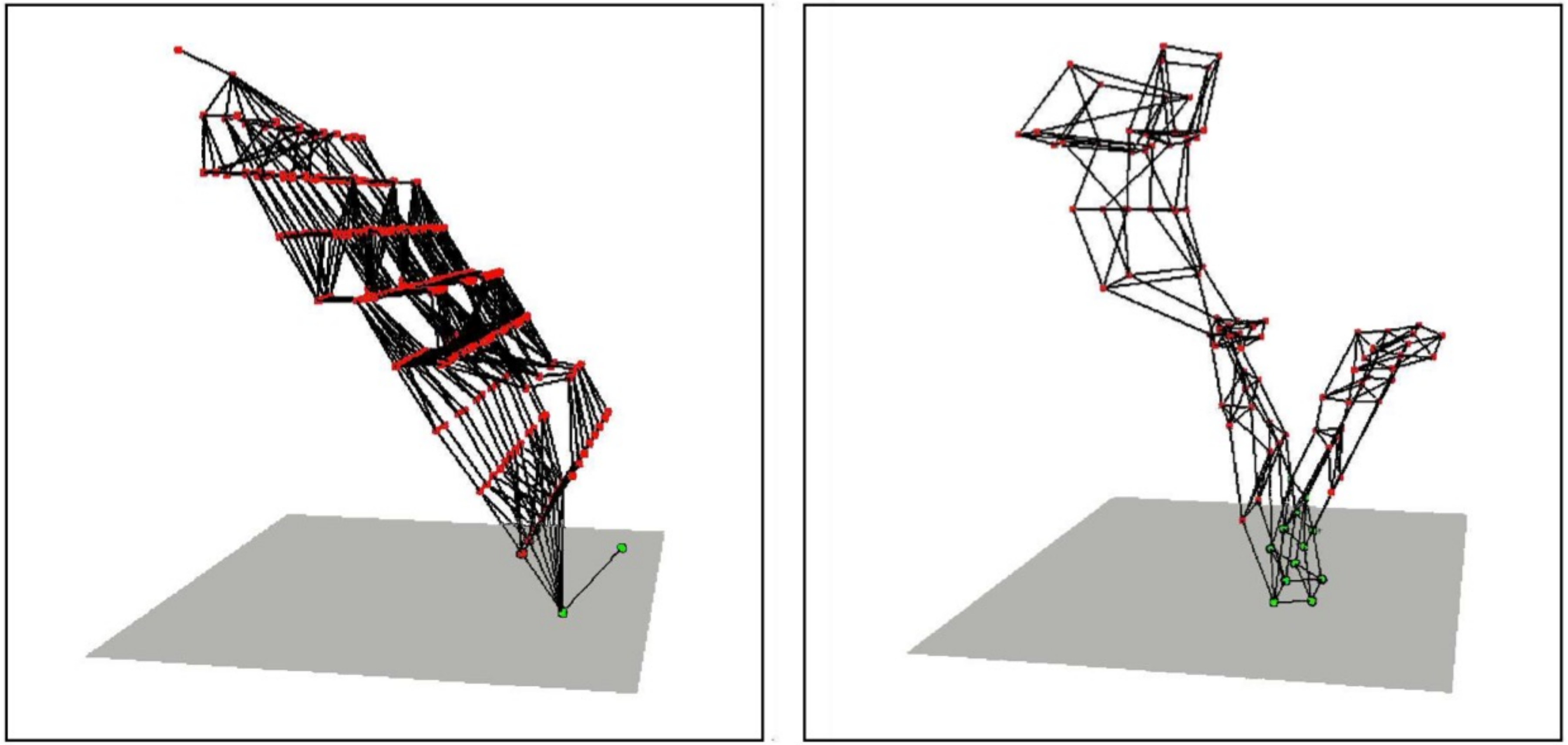
Forward (progression) search

(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



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

Definitions of possible refinements for two high-level actions

$$
\begin{aligned}
& \mathit{Refinement}(\mathit{Go}(\mathit{Home}, \mathit{SFO}), \\
& \quad \text{STEPS: } [\mathit{Drive}(\mathit{Home}, \mathit{SFOLongTermParking}), \\
& \qquad\qquad \mathit{Shuttle}(\mathit{SFOLongTermParking}, \mathit{SFO})]\ ) \\
& \mathit{Refinement}(\mathit{Go}(\mathit{Home}, \mathit{SFO}), \\
& \quad \text{STEPS: } [\mathit{Taxi}(\mathit{Home}, \mathit{SFO})]\ )
\end{aligned}
$$

$$
\begin{aligned}
& \mathit{Refinement}(\mathit{Navigate}([a,b],[x,y]), \\
& \quad \text{PRECOND: } a = x \ \wedge \ b = y \\
& \quad \text{STEPS: } [\,]\ ) \\
& \mathit{Refinement}(\mathit{Navigate}([a,b],[x,y]), \\
& \quad \text{PRECOND: } \mathit{Connected}([a,b],[a-1,b]) \\
& \quad \text{STEPS: } [\mathit{Left}, \mathit{Navigate}([a-1,b],[x,y])]\ ) \\
& \mathit{Refinement}(\mathit{Navigate}([a,b],[x,y]), \\
& \quad \text{PRECOND: } \mathit{Connected}([a,b],[a+1,b]) \\
& \quad \text{STEPS: } [\mathit{Right}, \mathit{Navigate}([a+1,b],[x,y])]\ )
\end{aligned}
$$

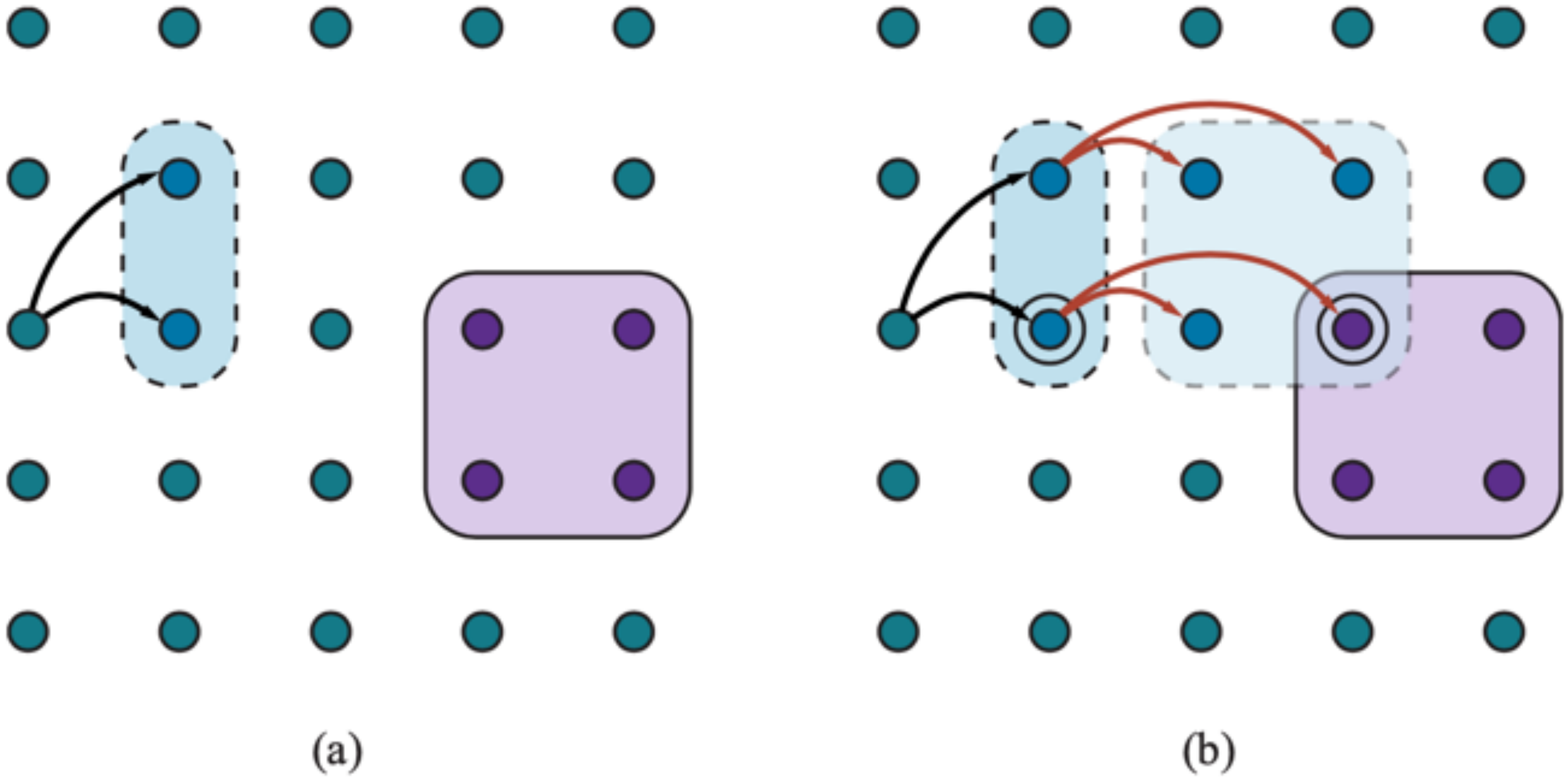
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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
```

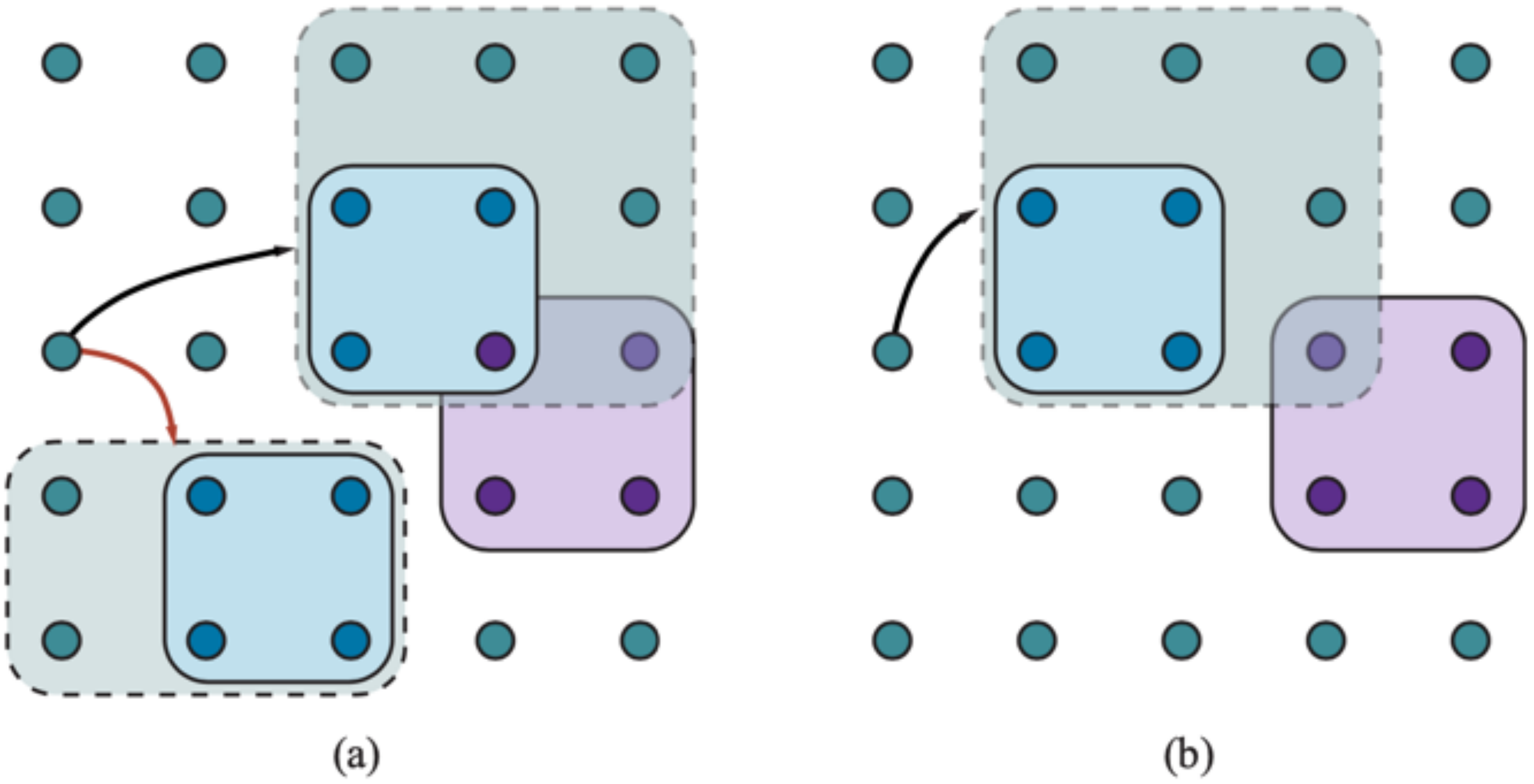
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Schematic examples of reachable sets



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

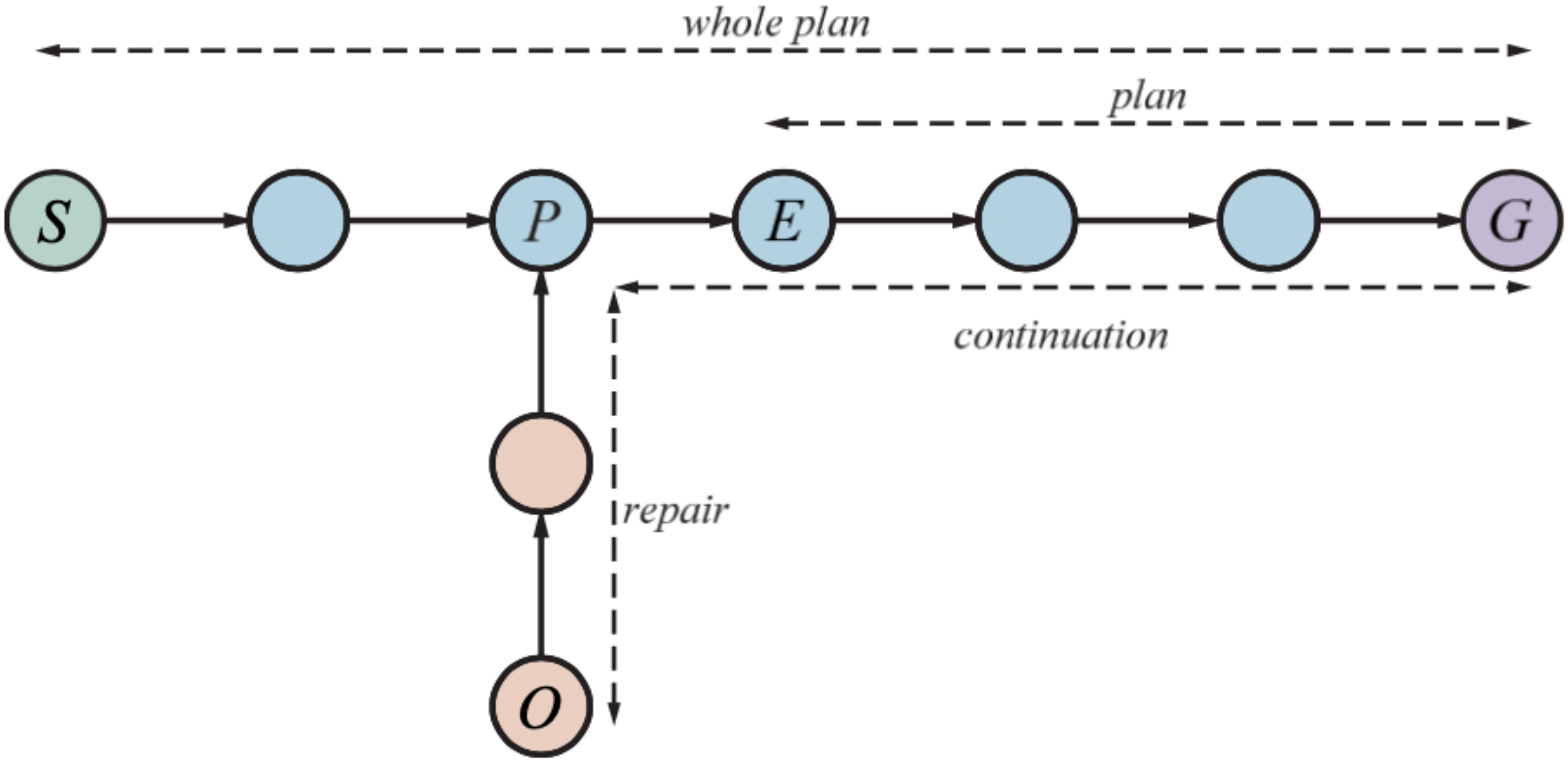
A hierarchical planning algorithm
Decompose solution

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



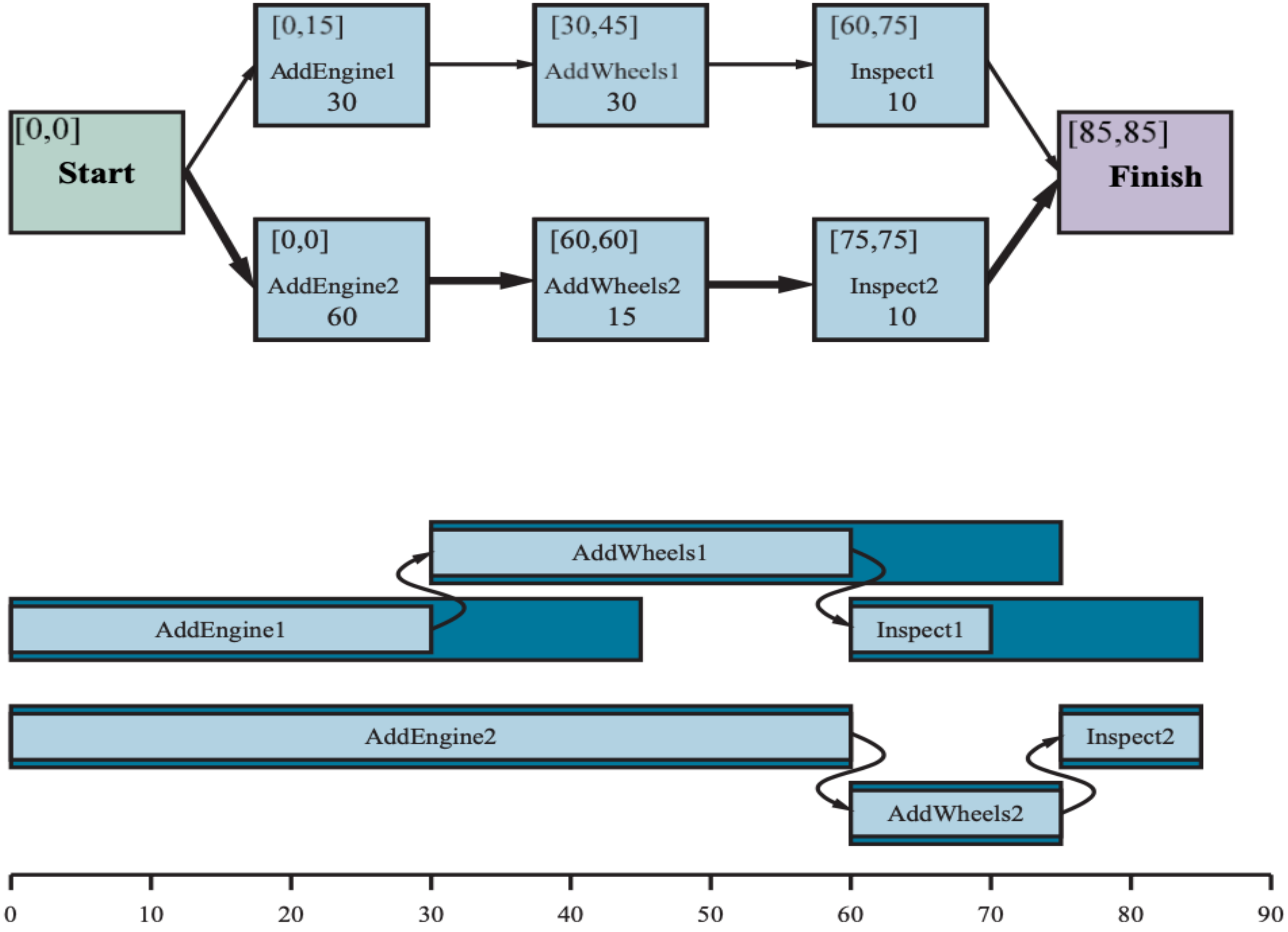
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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

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

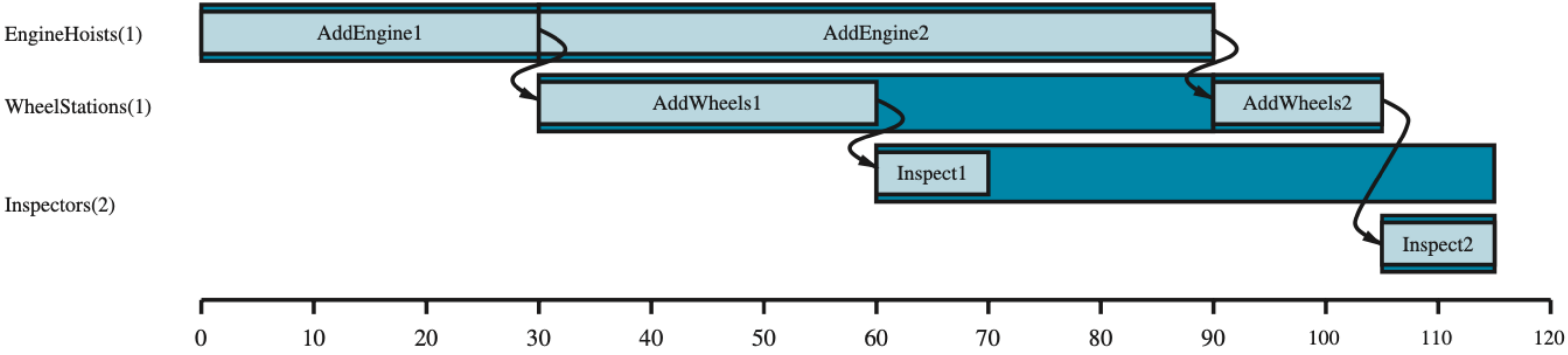
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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

Artificial Intelligence: Uncertain Knowledge and Reasoning

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

Artificial Intelligence:
4. Uncertain Knowledge and Reasoning

- Quantifying Uncertainty
- Probabilistic Reasoning
- Probabilistic Reasoning over Time
- Probabilistic Programming
- Making Simple Decisions
- Making Complex Decisions
- Multiagent Decision Making

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

Quantifying Uncertainty

DT-Agent
A Decision-Theoretic Agent that Selects Rational Actions

function DT-AGENT(*percept*) **returns** an *action*
  persistent: *belief_state*, probabilistic beliefs about the current state of the world
    action, the agent's action

  update *belief_state* based on *action* and *percept*
  calculate outcome probabilities for actions,
    given action descriptions and current *belief_state*
  select *action* with highest expected utility
    given probabilities of outcomes and utility information
  return *action*

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

Agent 1 has inconsistent beliefs

Proposition	Agent 1's belief	Agent 2 bets	Agent 1 bets	Agent 1 payoffs for each outcome a, b	$a, \neg b$	$\neg a, b$	$\neg a, \neg b$
a	0.4	\$4 on a	\$6 on $\neg a$	–\$6	–\$6	\$4	\$4
b	0.3	\$3 on b	\$7 on $\neg b$	–\$7	\$3	–\$7	\$3
$a \vee b$	0.8	\$2 on $\neg(a \vee b)$	\$8 on $a \vee b$	\$2	\$2	\$2	–\$8
				–\$11	–\$1	–\$1	–\$1

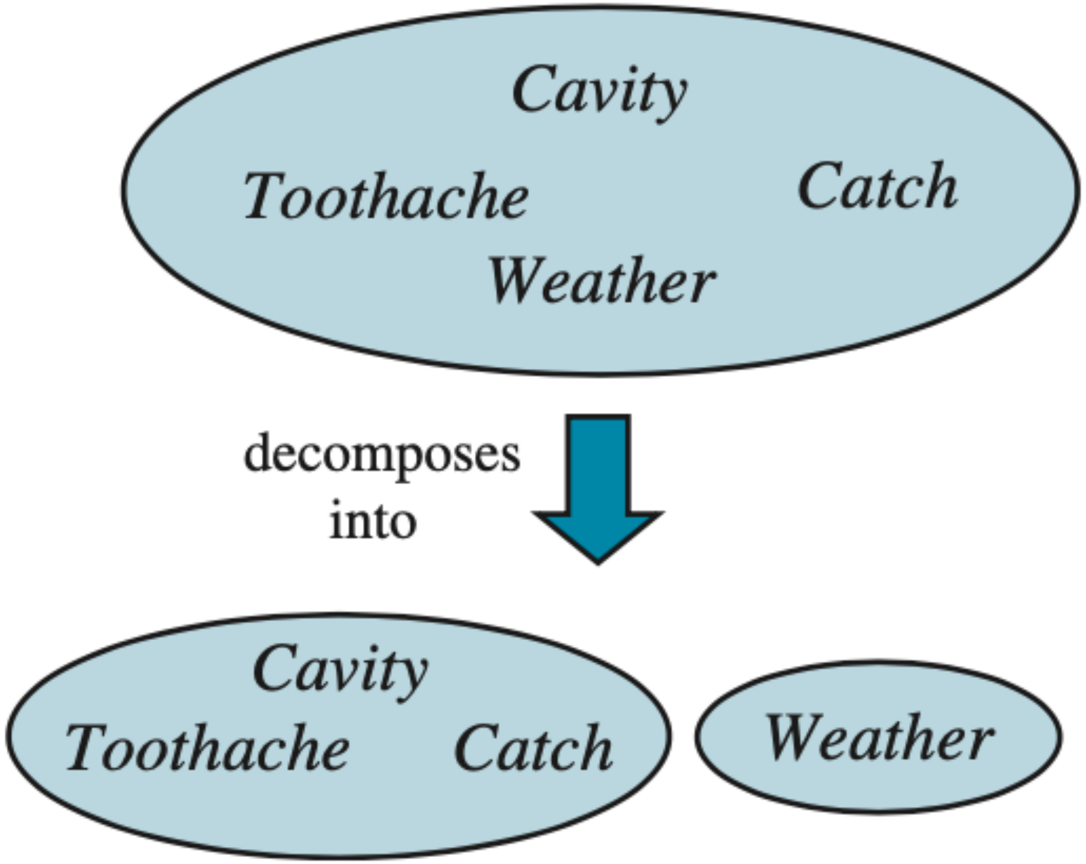
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A full joint distribution for the Toothache, Cavity, Catch world

	toothache		*¬toothache*	
	catch	*¬catch*	*catch*	*¬catch*
cavity	0.108	0.012	0.072	0.008
¬cavity	0.016	0.064	0.144	0.576

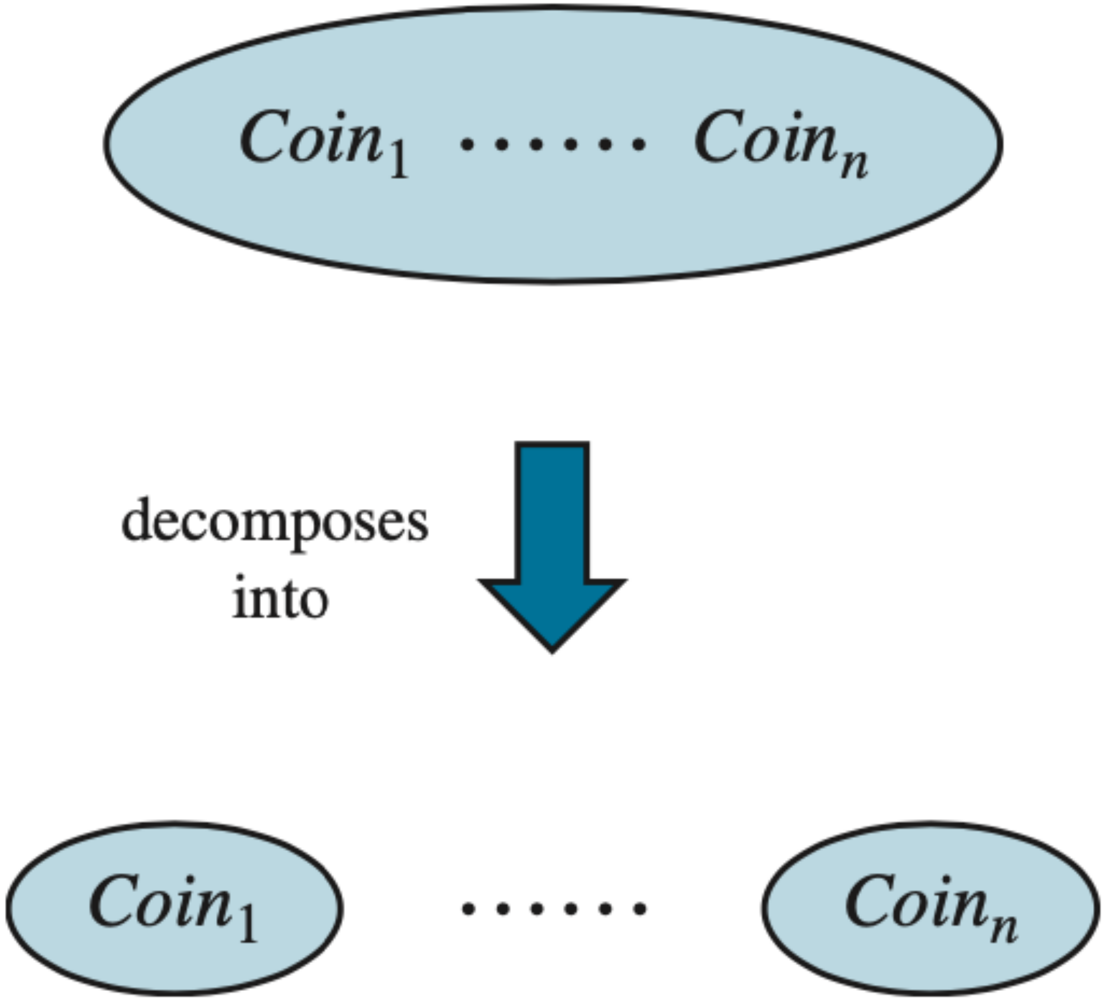
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Weather and Dental problems are independent



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

Coin flips are independent

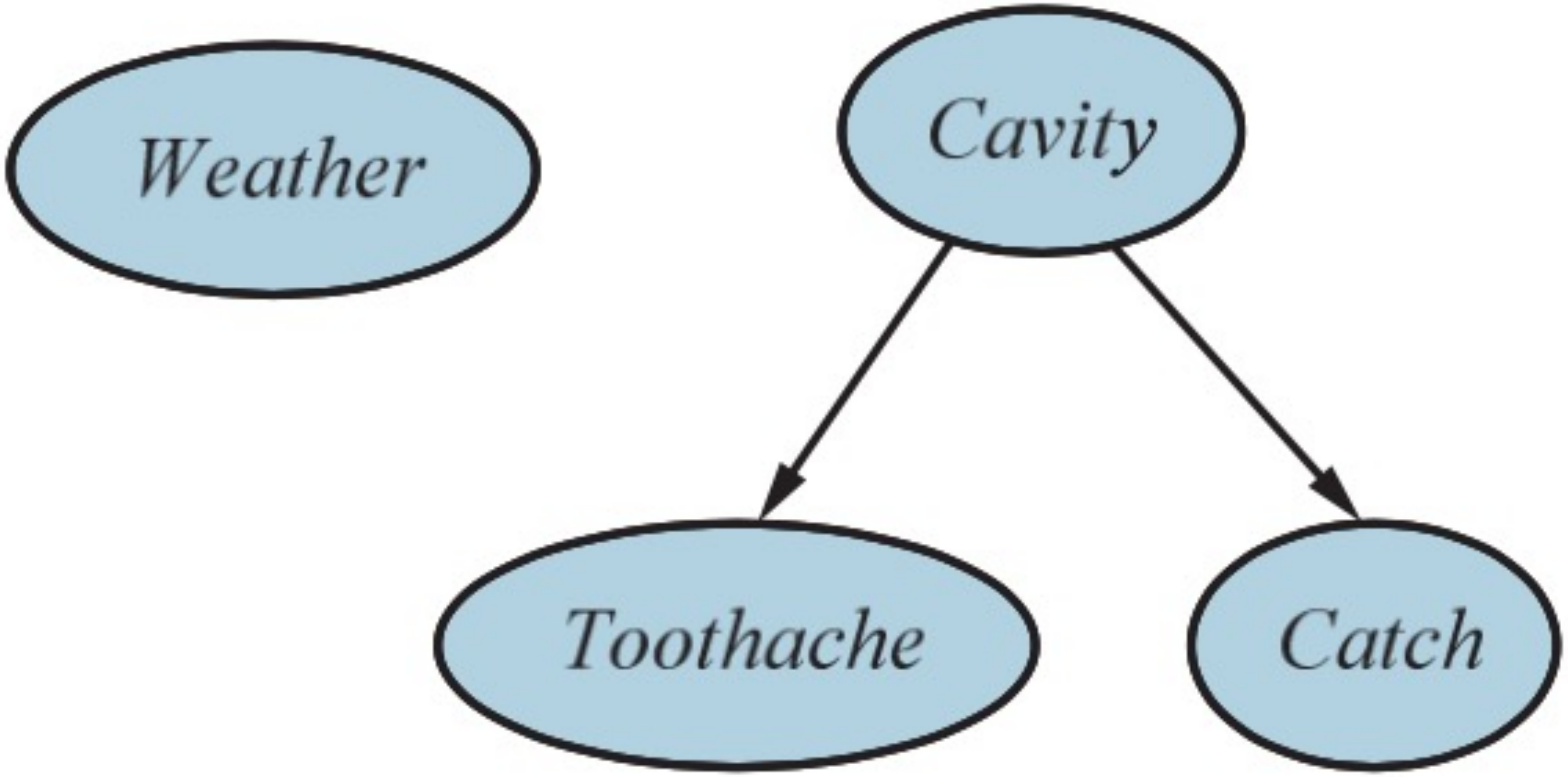


Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Probabilistic Reasoning

A Simple Bayesian Network

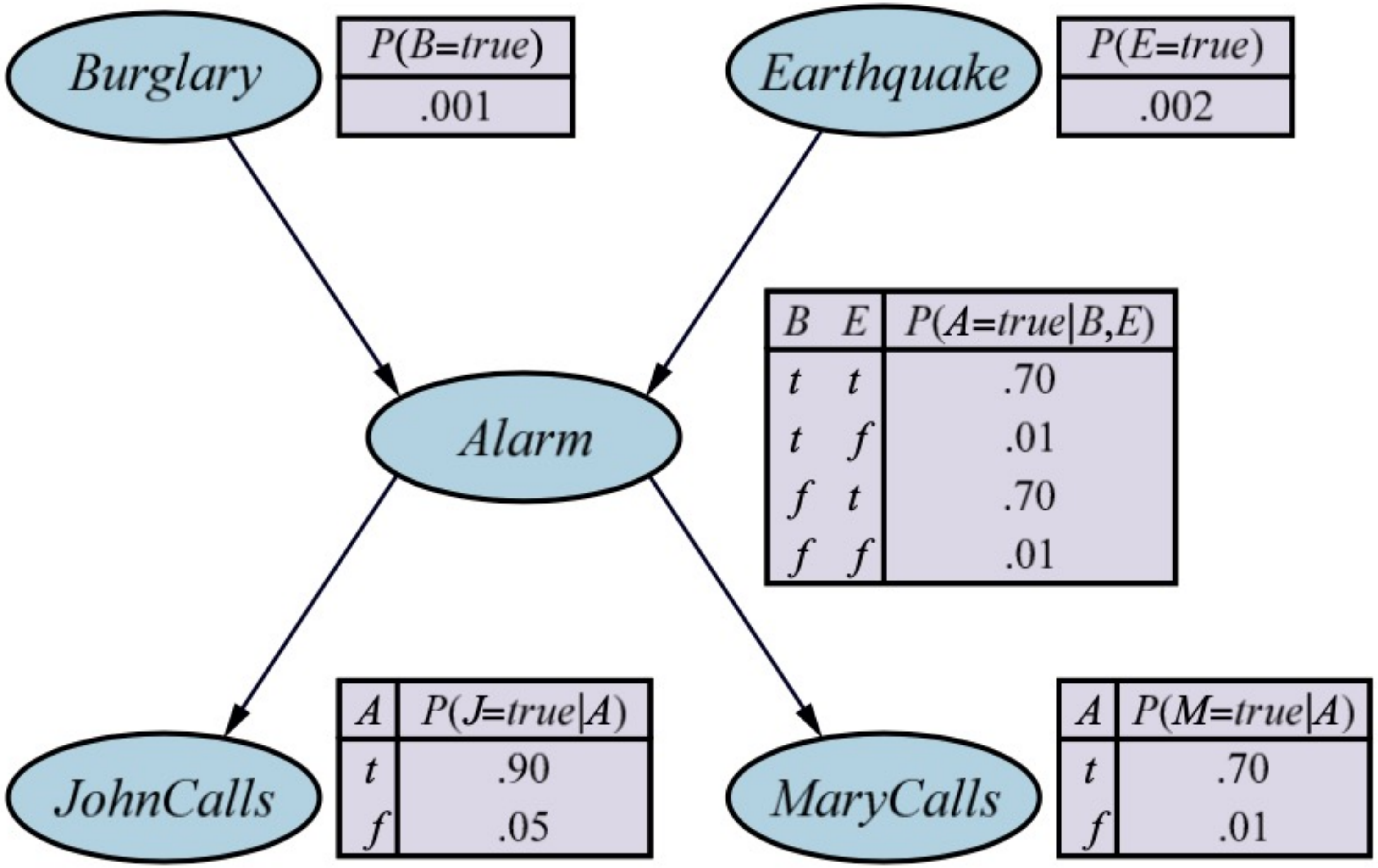
Weather is independent to the other three variables.
Toothache and Catch are conditionally independent, given Cavity.



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

A Typical Bayesian Network

Topology and the Conditional Probability Tables (CPTs)



Conditional Probability Table

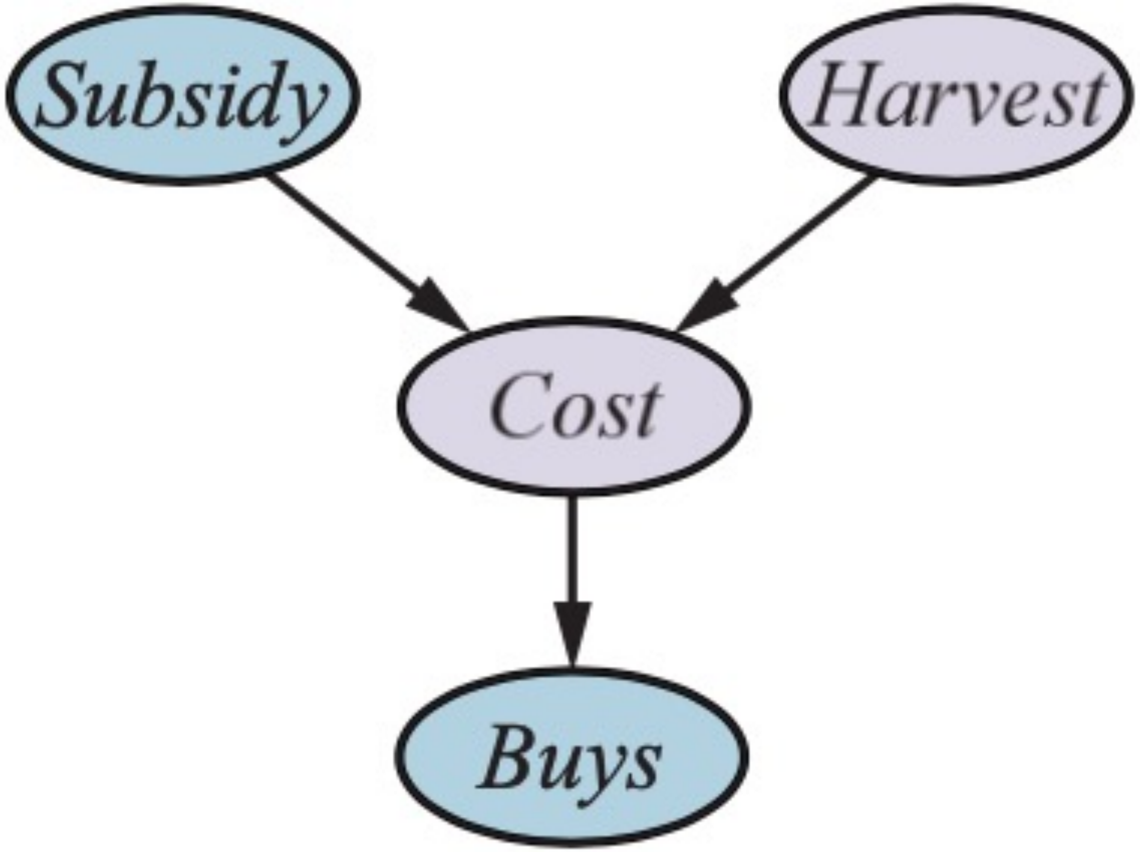
for P(Fever |Cold, Flu, Malaria)

$Cold$	Flu	$Malaria$	$P(fever \mid \cdot)$	$P(\neg fever \mid \cdot)$
f	f	f	0.0	1.0
f	f	t	0.9	**0.1**
f	t	f	0.8	**0.2**
f	t	t	0.98	$0.02 = 0.2 \times 0.1$
t	f	f	0.4	**0.6**
t	f	t	0.94	$0.06 = 0.6 \times 0.1$
t	t	f	0.88	$0.12 = 0.6 \times 0.2$
t	t	t	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

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

A Simple Network

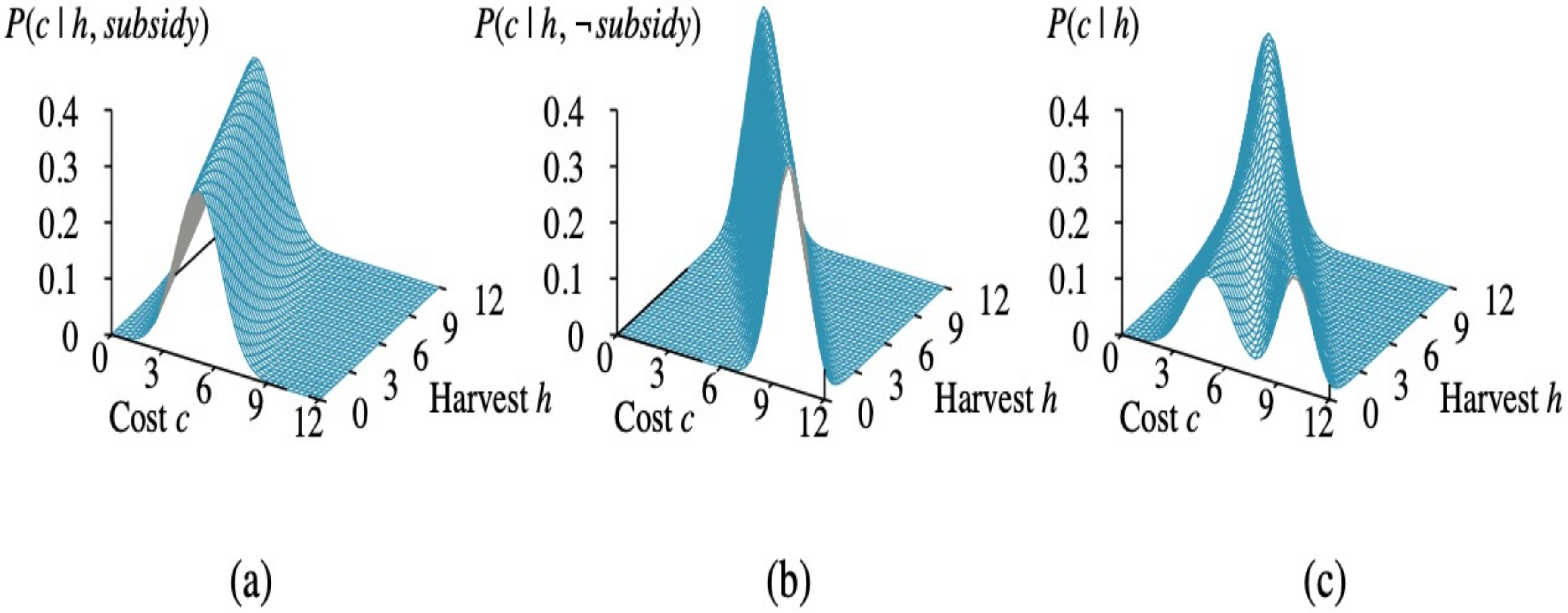
with discrete variables (Subsidy and Buys)
and continuous variables (Harvest and Cost)



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

Probability distribution

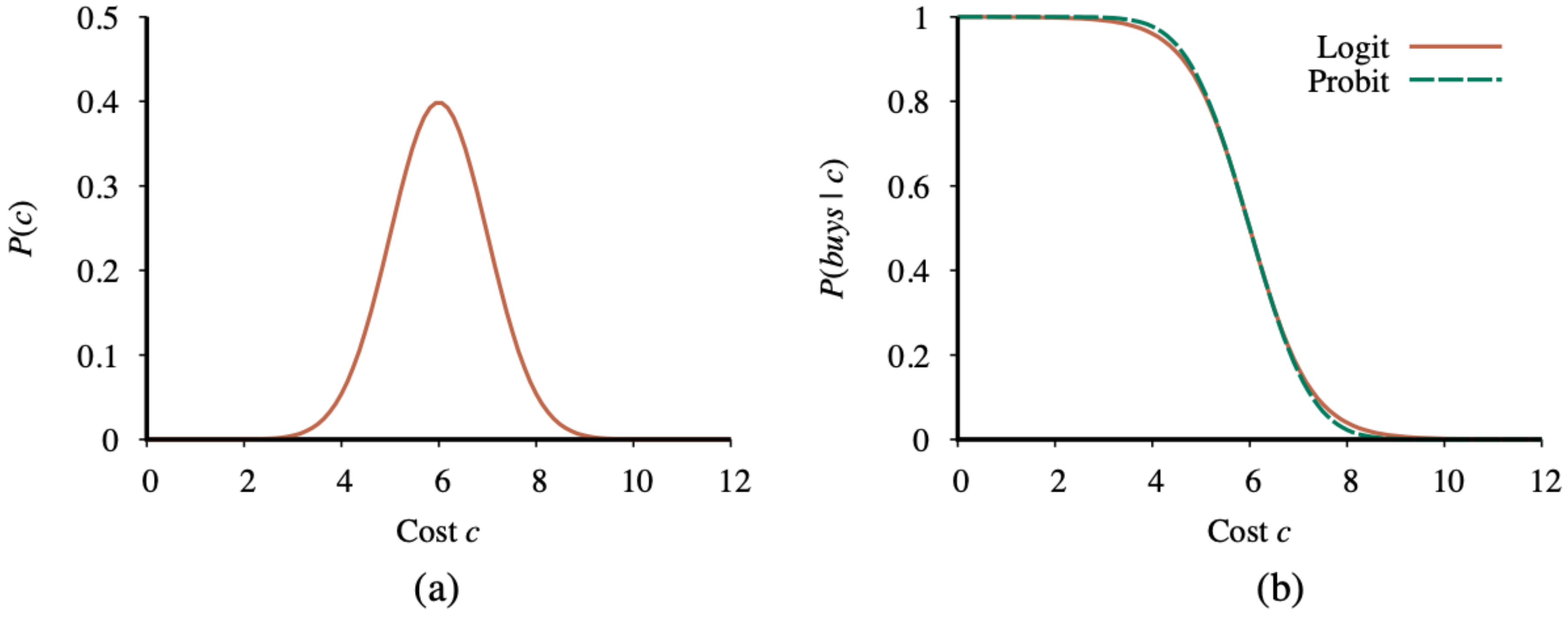
over Cost as a function of Harvest size



distribution P (Cost | Harvest), obtained by summing over the two subsidy cases.

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

A normal (Gaussian) distribution for the cost threshold

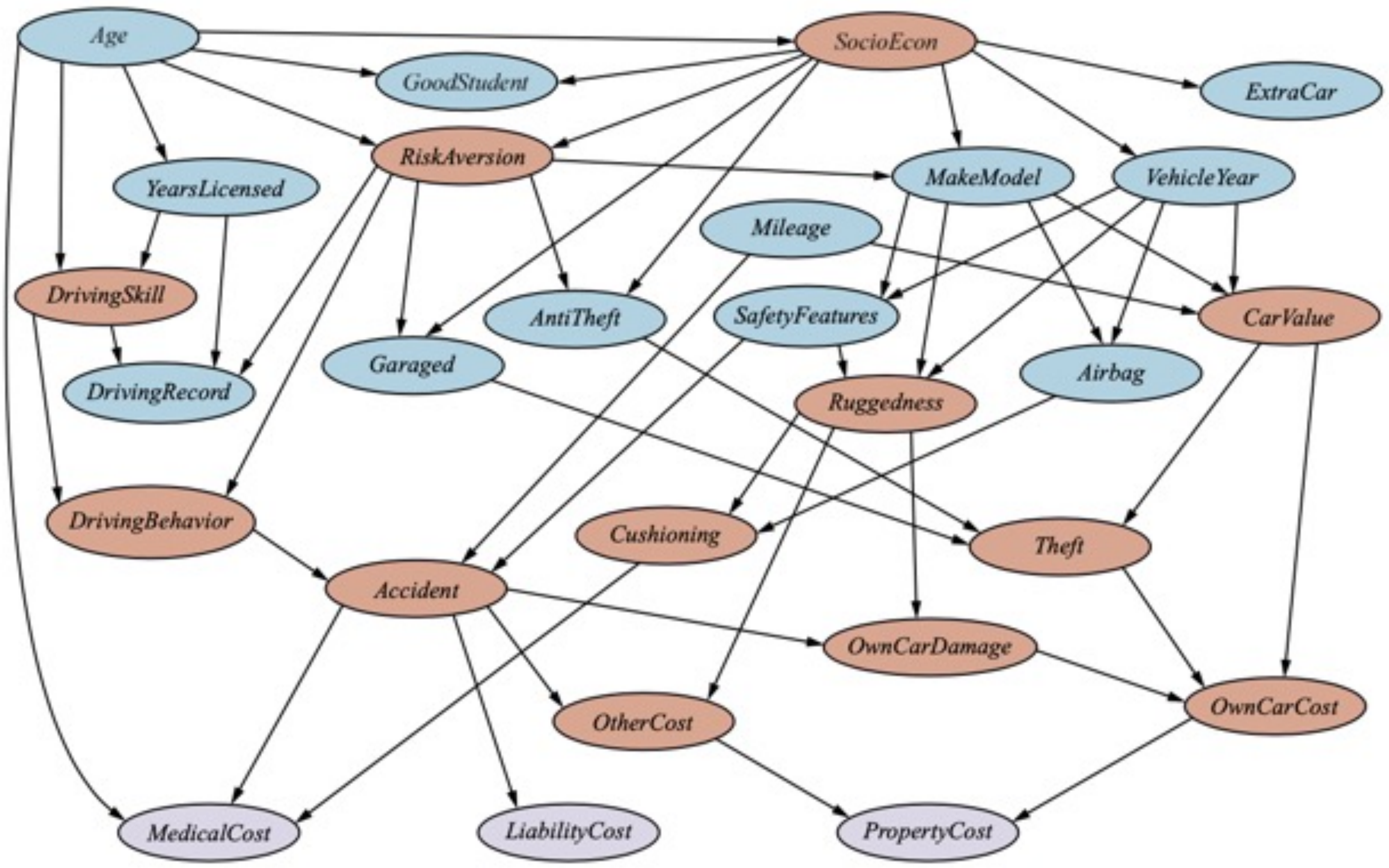


Expit and Probit models for the probability of buys given cost

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

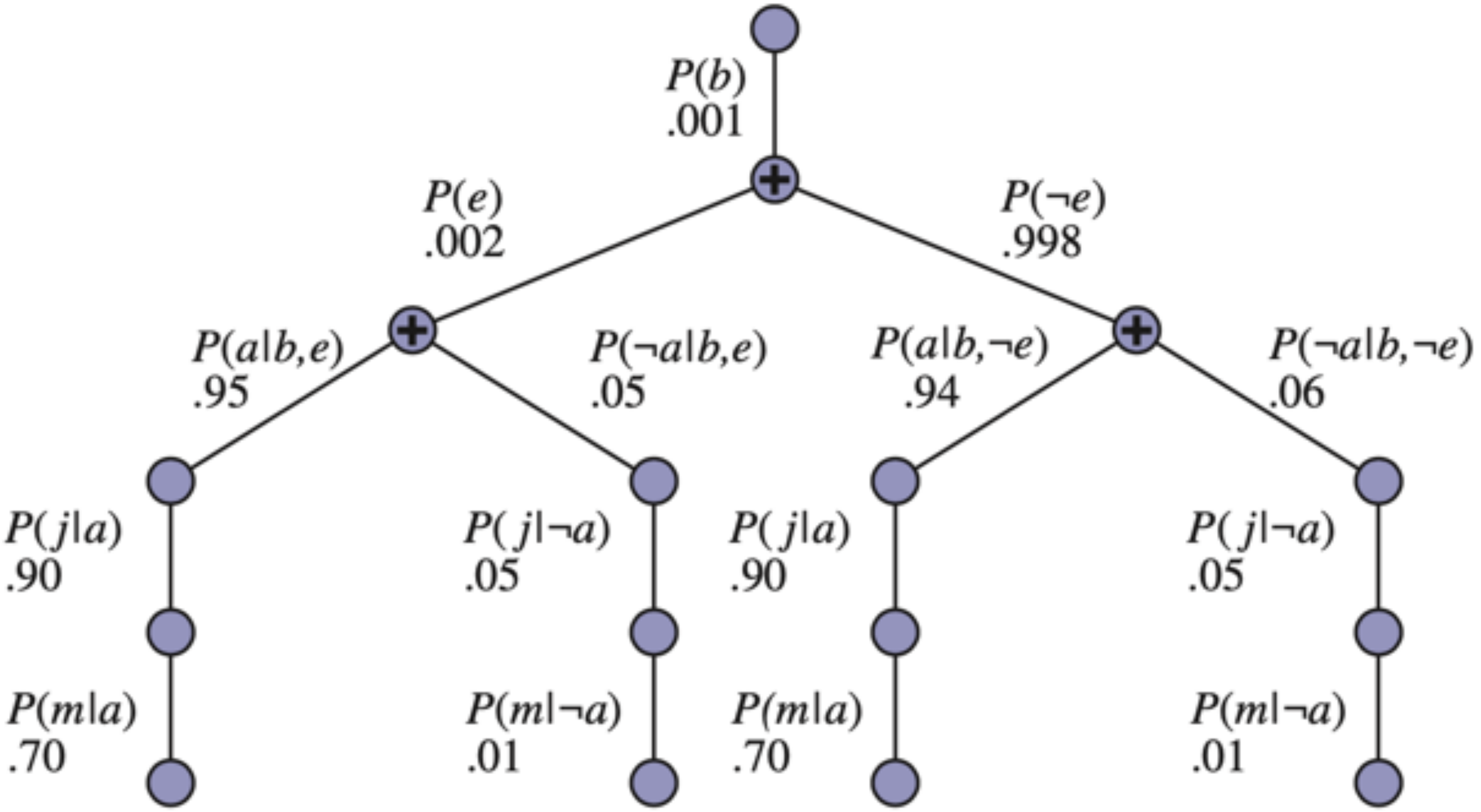
A Bayesian Network

for evaluating car insurance applications



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

The structure of the expression



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

The Enumeration Algorithm for Exact Inference in Bayes Nets

function ENUMERATION-ASK(X, $\mathbf{e}$, bn) **returns** a distribution over X
  inputs: X, the query variable
    $\mathbf{e}$, observed values for variables $\mathbf{E}$
    bn, a Bayes net with variables $vars$

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL($vars$, $\mathbf{e}_{x_i}$)
      where $\mathbf{e}_{x_i}$ is $\mathbf{e}$ extended with $X = x_i$
  return NORMALIZE($\mathbf{Q}(X)$)

function ENUMERATE-ALL($vars$, $\mathbf{e}$) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $V \leftarrow$ FIRST($vars$)
  if V is an evidence variable with value v in $\mathbf{e}$
    then return $P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}$)
    else return $\sum_v P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_v$)
      where $\mathbf{e}_v$ is $\mathbf{e}$ extended with $V = v$

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

Pointwise Multiplication

$\mathbf{f}(X,Y) \times \mathbf{g}(Y,Z) = \mathbf{h}(X,Y,Z)$

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	.3	t	t	.2	t	t	t	$.3 \times .2 = .06$
t	f	.7	t	f	.8	t	t	f	$.3 \times .8 = .24$
f	t	.9	f	t	.6	t	f	t	$.7 \times .6 = .42$
f	f	.1	f	f	.4	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

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

The Variable Elimination Algorithm for Exact Inference in Bayes Nets

function ELIMINATION-ASK(X, $\mathbf{e}$, bn) **returns** a distribution over X
  inputs: X, the query variable
    $\mathbf{e}$, observed values for variables $\mathbf{E}$
    bn, a Bayesian network with variables $vars$

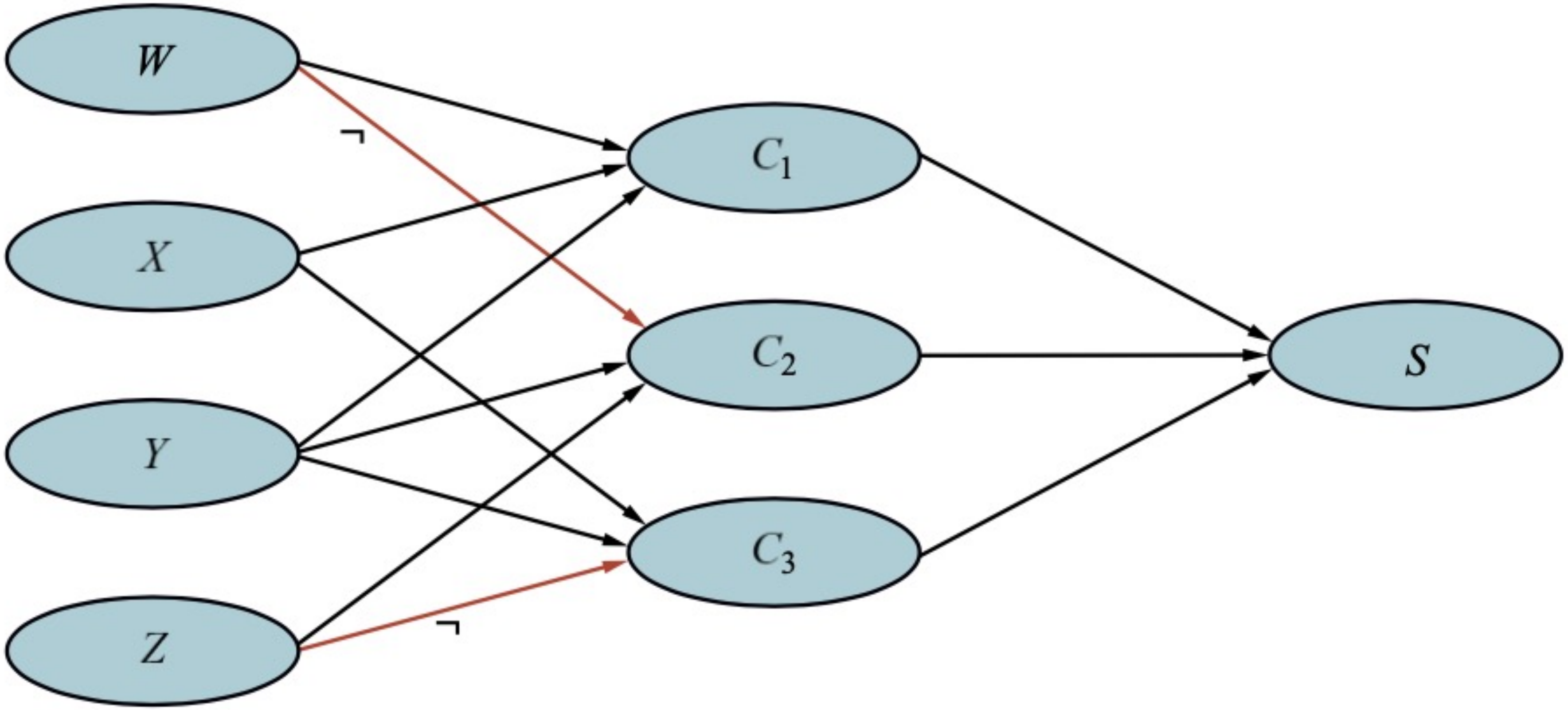
  $factors \leftarrow [\,]$
  for each V **in** ORDER($vars$) **do**
    $factors \leftarrow$ [MAKE-FACTOR(V, $\mathbf{e}$)] $+$ $factors$
    if V is a hidden variable **then** $factors \leftarrow$ SUM-OUT(V, $factors$)
  return NORMALIZE(POINTWISE-PRODUCT($factors$))

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

Bayes Net Encoding

of the 3-CNF (Conjunctive Normal Form) Sentence

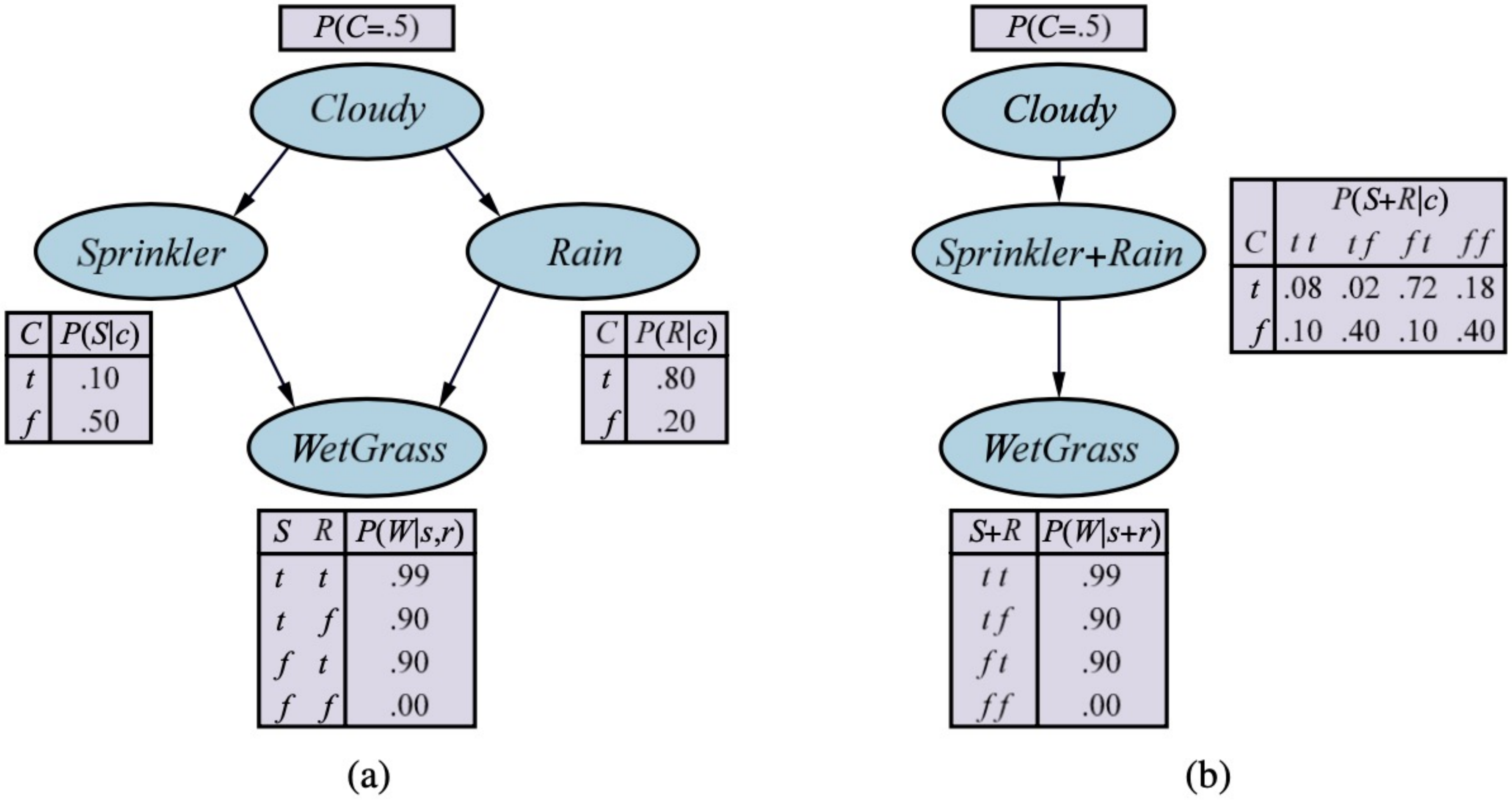
$(W \lor X \lor Y) \land (\neg W \lor Y \lor Z) \land (X \lor Y \lor \neg Z)$



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

Multiply Connected Network

(b) A clustered equivalent



A Sampling Algorithm

that generates events from a Bayesian network

function PRIOR-SAMPLE(bn) **returns** an event sampled from the prior specified by bn
  inputs: bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  $\mathbf{x} \leftarrow$ an event with n elements
  for each variable X_i **in** $X_1, \ldots, X_n$ **do**
    $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return $\mathbf{x}$

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

The Rejection-Sampling Algorithm

for answering queries given evidence in a Bayesian network

function REJECTION-SAMPLING(X, $\mathbf{e}$, bn, N) **returns** an estimate of $\mathbf{P}(X \mid \mathbf{e})$
  inputs: X, the query variable
    $\mathbf{e}$, observed values for variables $\mathbf{E}$
    bn, a Bayesian network
    N, the total number of samples to be generated
  local variables: $\mathbf{C}$, a vector of counts for each value of X, initially zero

  for j = 1 **to** N **do**
    $\mathbf{x} \leftarrow$ PRIOR-SAMPLE(bn)
    if $\mathbf{x}$ is consistent with $\mathbf{e}$ **then**
      $\mathbf{C}[j] \leftarrow \mathbf{C}[j]+1$ where x_j is the value of X in $\mathbf{x}$
  return NORMALIZE($\mathbf{C}$)

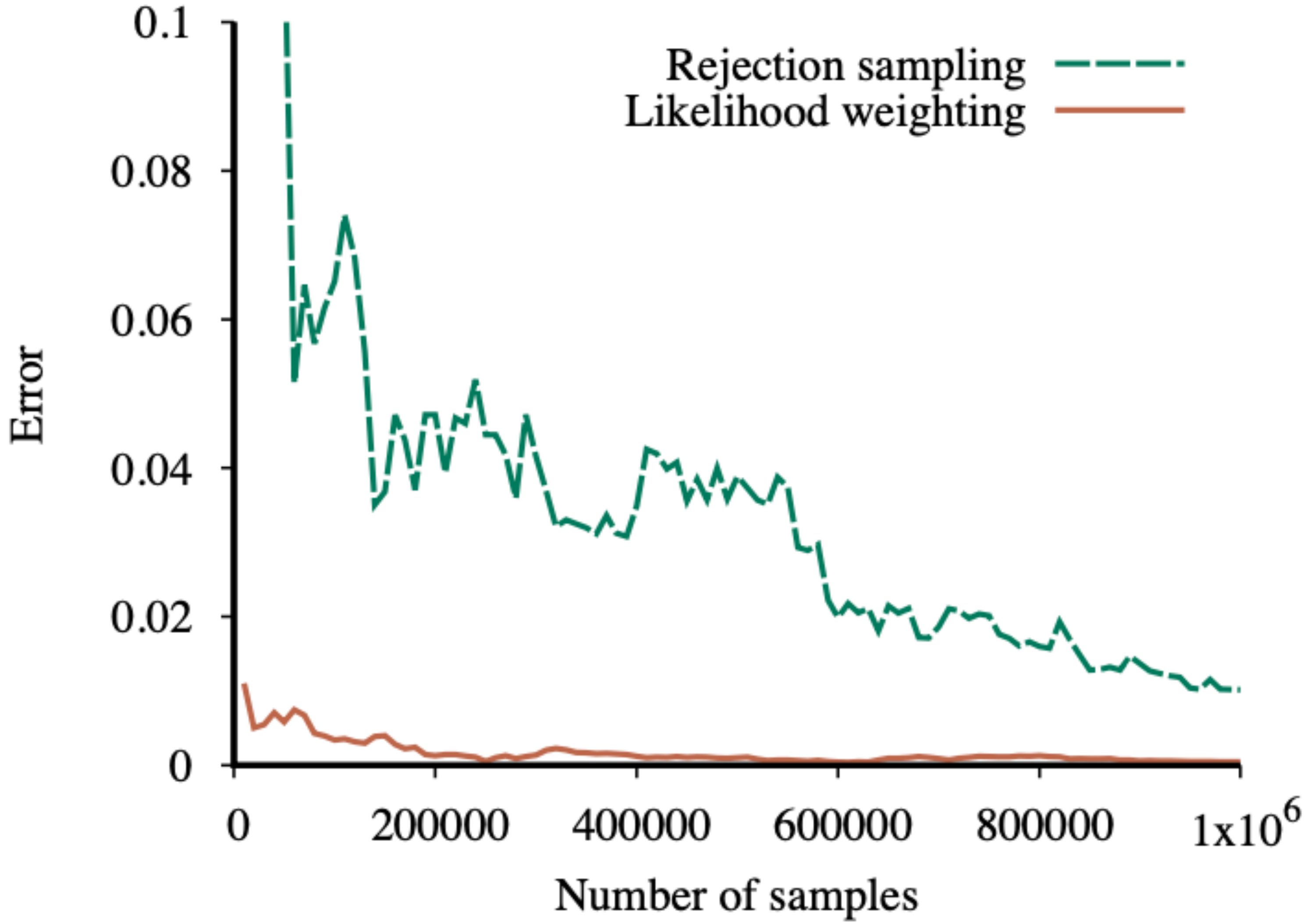
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

The Likelihood-Weighting Algorithm

for inference in Bayesian networks

```
function LIKELIHOOD-WEIGHTING(X, e, bn, N) returns an estimate of P(X | e)
  inputs: X, the query variable
          e, observed values for variables E
          bn, a Bayesian network specifying joint distribution P(X_1, ..., X_n)
          N, the total number of samples to be generated
  local variables: W, a vector of weighted counts for each value of X, initially zero

  for j = 1 to N do
     x, w ← WEIGHTED-SAMPLE(bn, e)
     W[j] ← W[j] + w where x_j is the value of X in x
  return NORMALIZE(W)

function WEIGHTED-SAMPLE(bn, e) returns an event and a weight

  w ← 1; x ← an event with n elements, with values fixed from e
  for i = 1 to n do
     if X_i is an evidence variable with value x_ij in e
        then w ← w × P(X_i = x_ij | parents(X_i))
        else x[i] ← a random sample from P(X_i | parents(X_i))
  return x, w
```

Performance of rejection sampling and likelihood weighting on the insurance network



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

The Gibbs Sampling Algorithm

for approximate inference in Bayes nets

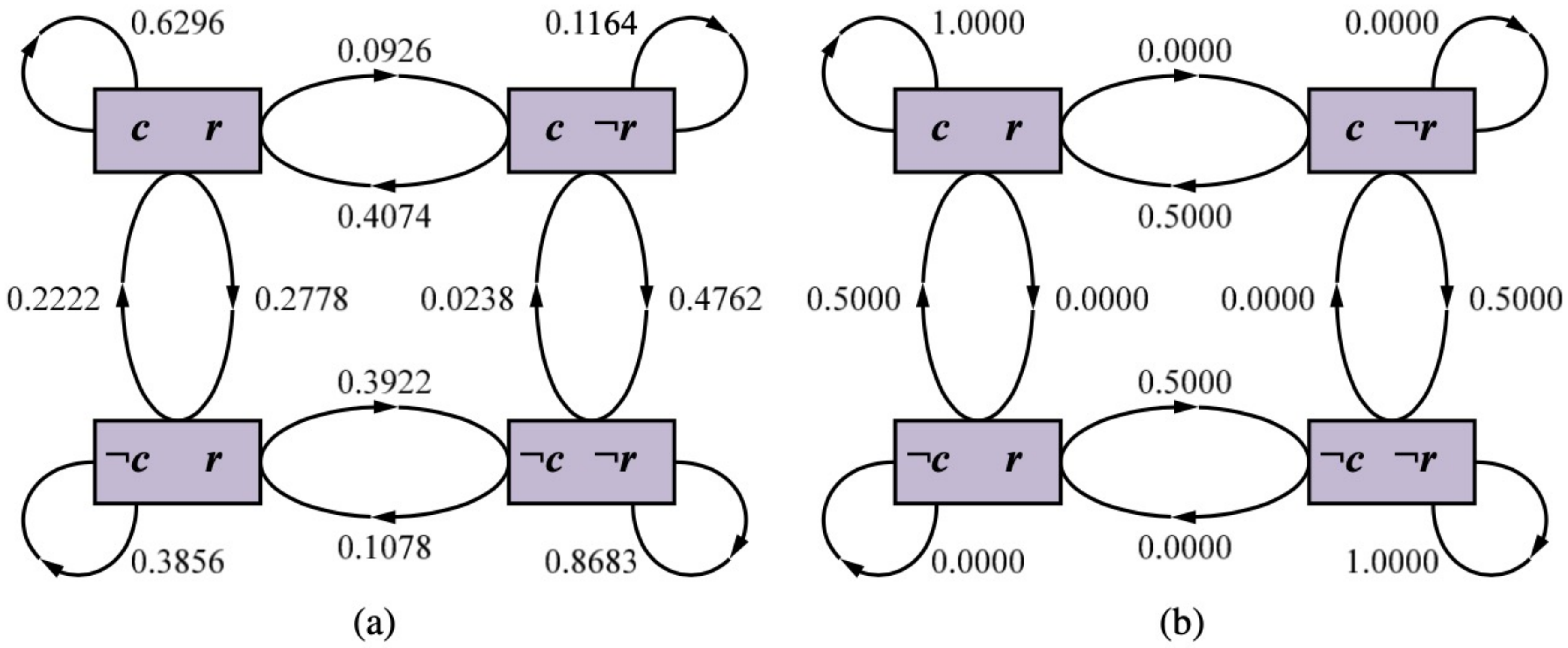
function GIBBS-ASK(X, $\mathbf{e}$, bn, N) **returns** an estimate of $\mathbf{P}(X \mid \mathbf{e})$
    local variables: $\mathbf{C}$, a vector of counts for each value of X, initially zero
        $\mathbf{Z}$, the nonevidence variables in bn
        $\mathbf{x}$, the current state of the network, initialized from $\mathbf{e}$

    initialize $\mathbf{x}$ with random values for the variables in $\mathbf{Z}$
    for $k = 1$ **to** N **do**
        choose any variable Z_i from $\mathbf{Z}$ according to any distribution $\rho(i)$
        set the value of Z_i in $\mathbf{x}$ by sampling from $\mathbf{P}(Z_i \mid mb(Z_i))$
        $\mathbf{C}[j] \leftarrow \mathbf{C}[j] + 1$ where x_j is the value of X in $\mathbf{x}$
    return NORMALIZE($\mathbf{C}$)

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

The States and Transition Probabilities of the Markov Chain

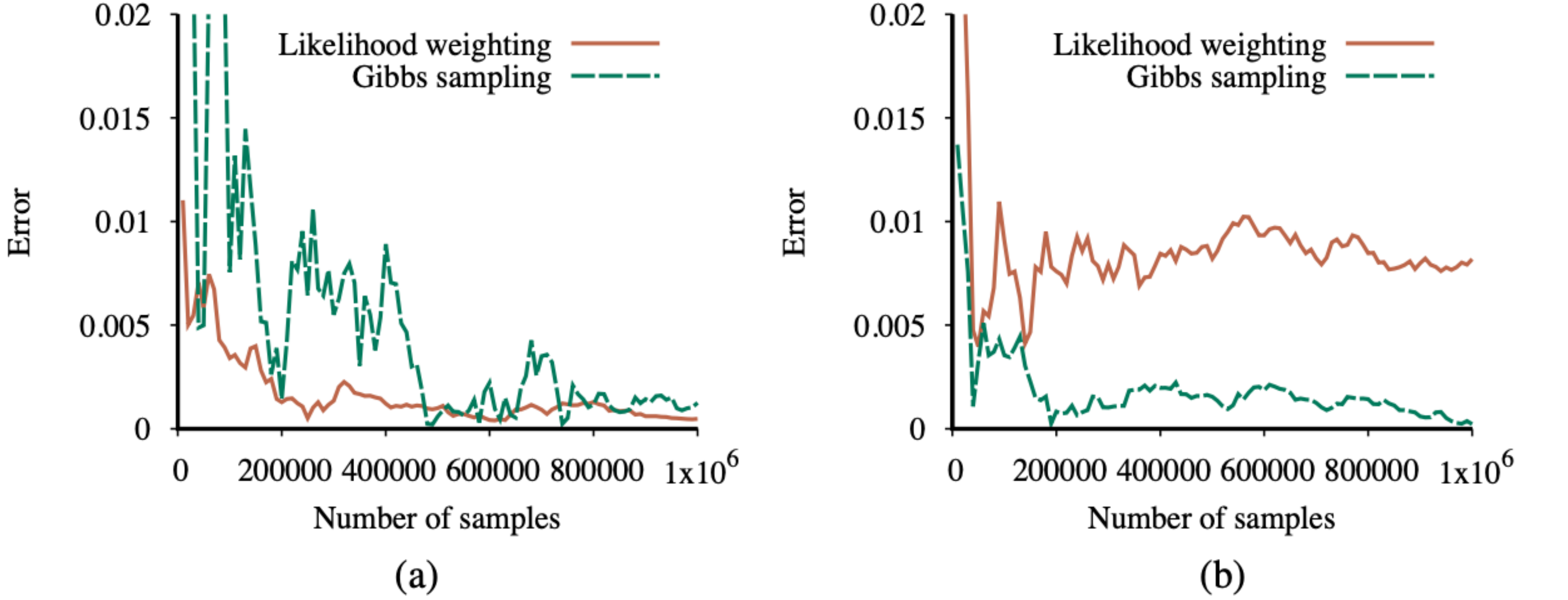
for the query $\mathbf{P}(Rain \mid Sprinkler = true, WetGrass = true)$



Transition Probabilities when the CPT for Rain constrains it to have the same value as Cloudy

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

Performance of Gibbs sampling compared to likelihood weighting on the car insurance network



(a)

for the standard query on PropertyCost

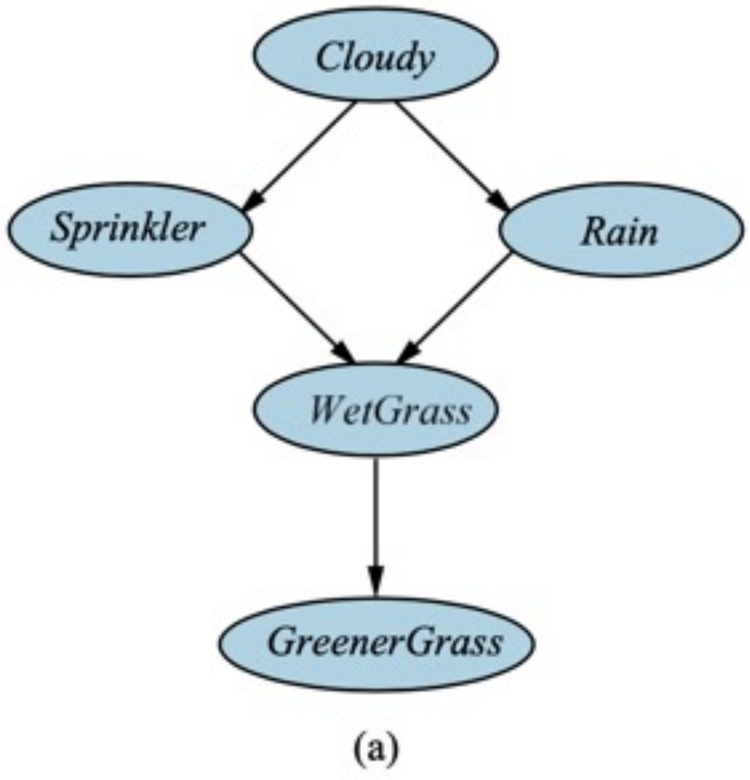
(b)

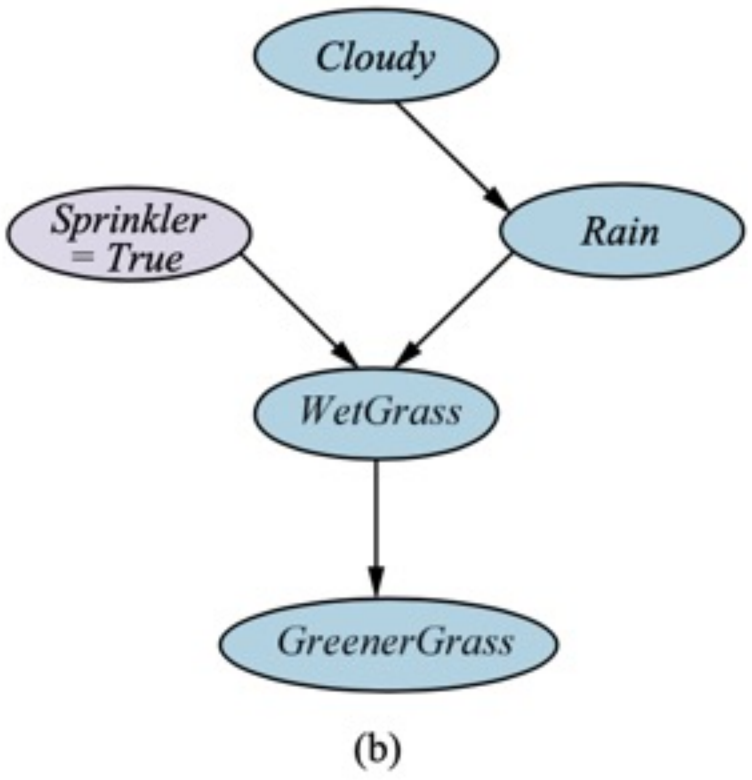
for the case where the output variables are observed and Age is the query variable

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

A Causal Bayesian Network

representing cause-effect relations among five variables





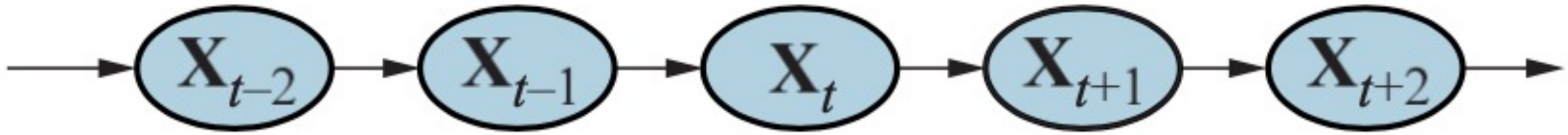
The network after performing the action "turn Sprinkler on."

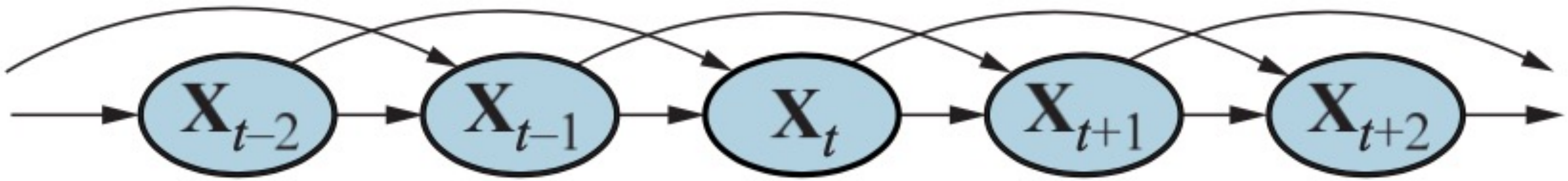
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Probabilistic Reasoning over Time

Bayesian network structure

corresponding to a First-order Markov Process with state defined by the variables X_t.

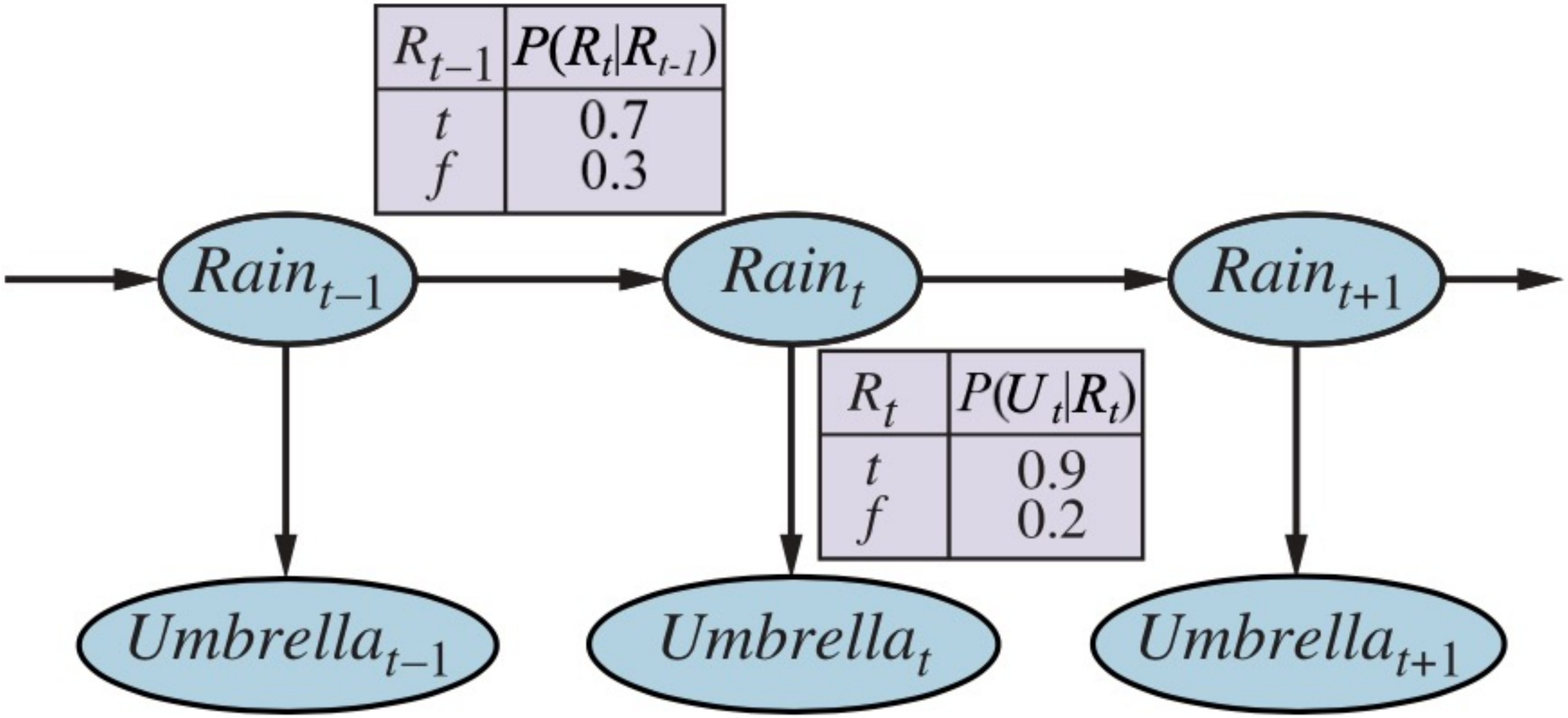




A Second-order Markov Process

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

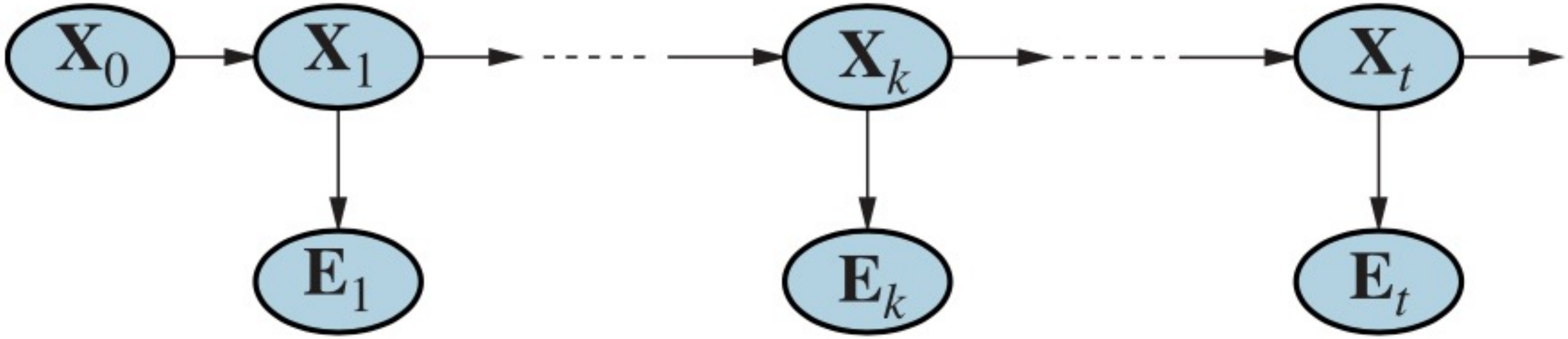
Bayesian Network Structure and Conditional Distributions describing the umbrella world



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

Smoothing computes $P(X_k \mid e_{1:t})$

the posterior distribution of the state at some past time k given a complete sequence of observations from 1 to t.



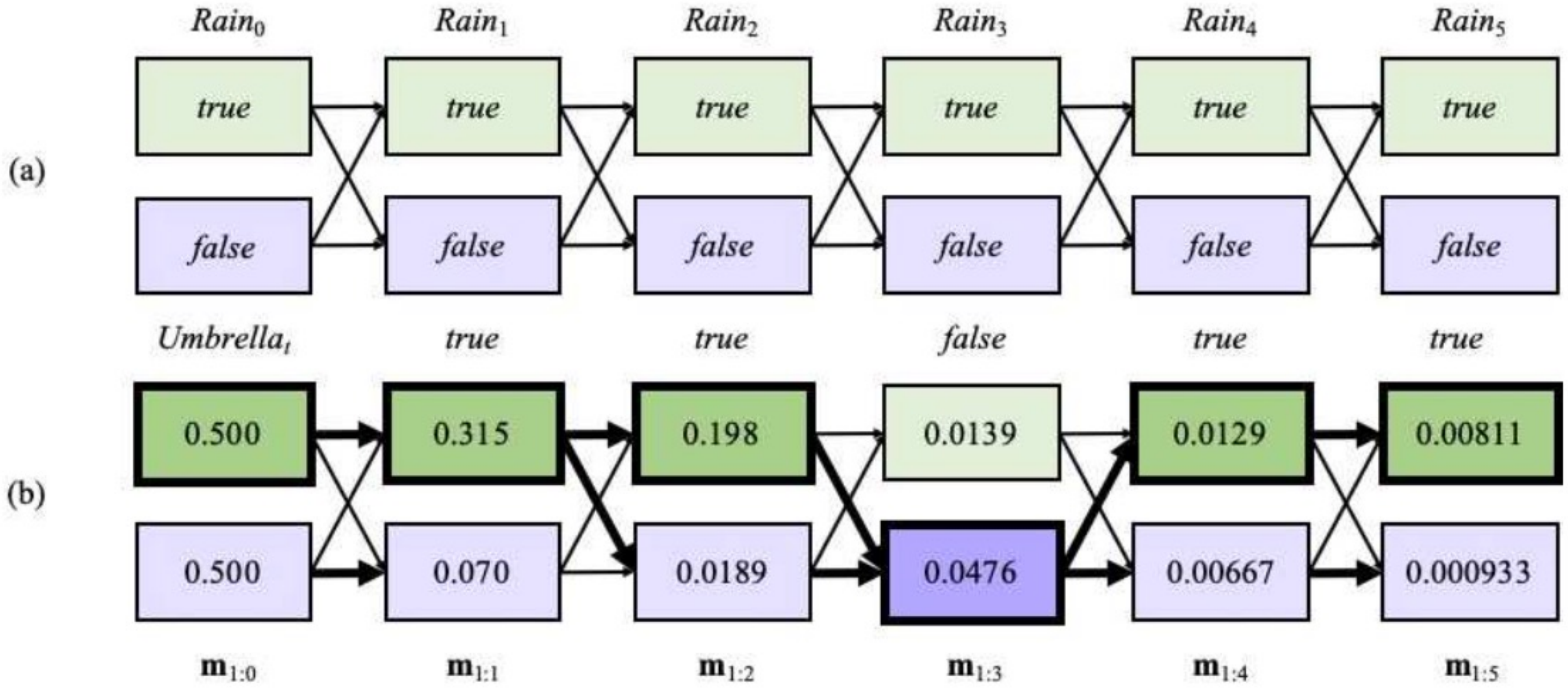
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

The Forward–Backward Algorithm for Smoothing

function FORWARD-BACKWARD(**ev**, *prior*) **returns** a vector of probability distributions
  inputs: **ev**, a vector of evidence values for steps $1, \ldots, t$
    prior, the prior distribution on the initial state, $\mathbf{P}(\mathbf{X}_0)$
  local variables: **fv**, a vector of forward messages for steps $0, \ldots, t$
    b, a representation of the backward message, initially all 1s
    sv, a vector of smoothed estimates for steps $1, \ldots, t$

  fv$[0] \leftarrow$ *prior*
  for $i = 1$ **to** t **do**
    fv$[i] \leftarrow$ FORWARD(**fv**$[i-1]$, **ev**$[i]$)
  for $i = t$ **down to** 1 **do**
    sv$[i] \leftarrow$ NORMALIZE(**fv**$[i] \times$ **b**)
    b $\leftarrow$ BACKWARD(**b**, **ev**$[i]$)
  return sv

Possible state sequences for $Rain_t$ can be viewed as paths through a graph of the possible states at each time step



Operation of the Viterbi algorithm
for the umbrella observation sequence *[true, true, false, true, true]*

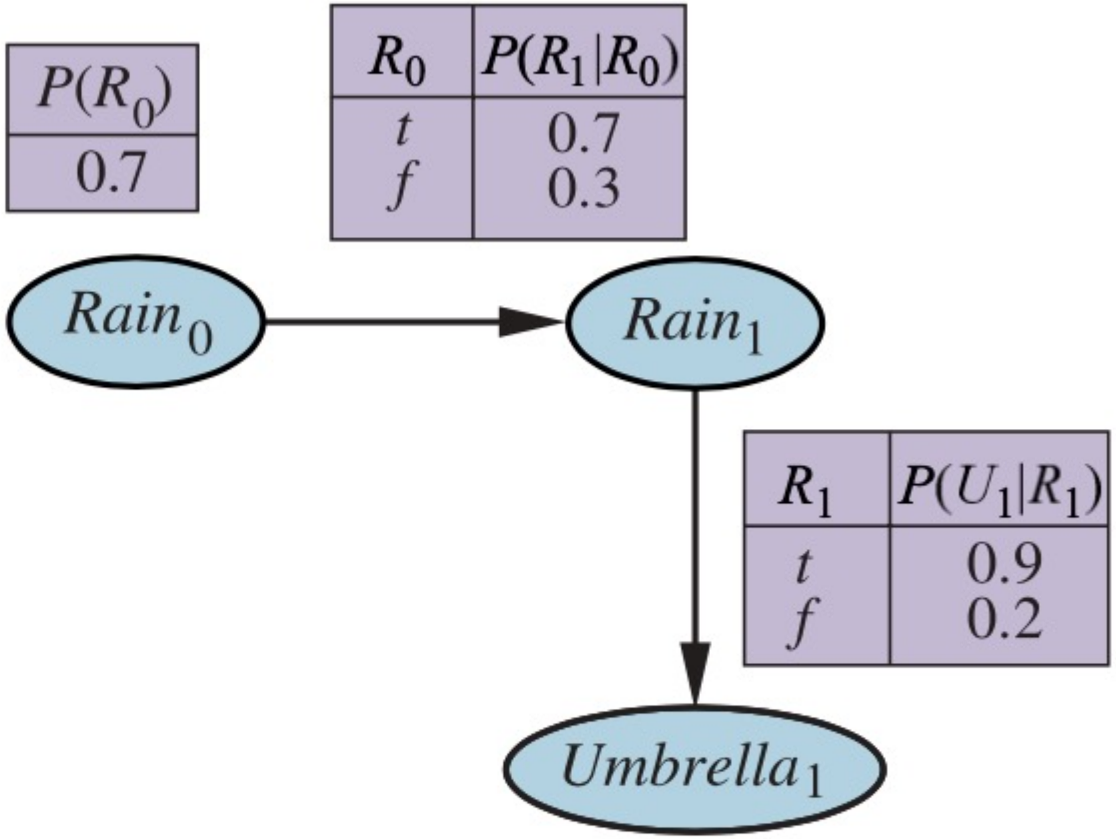
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

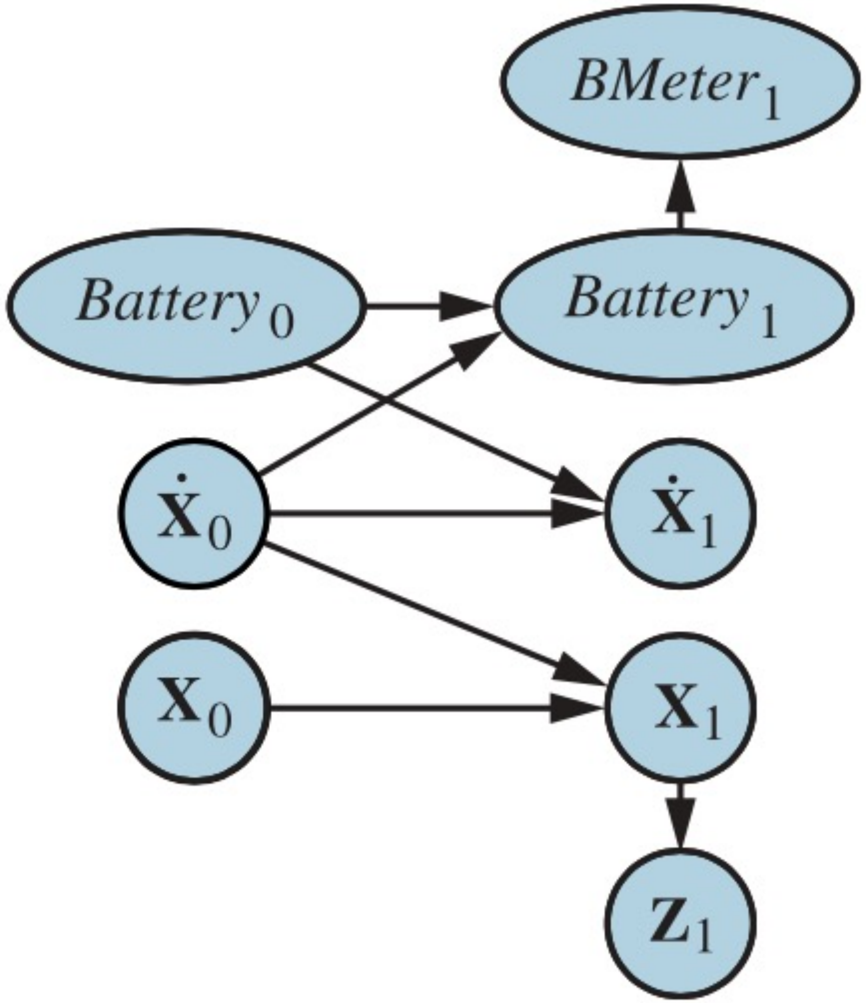
Algorithm for Smoothing with a Fixed Time Lag of d Step

```
function FIXED-LAG-SMOOTHING(e_t, hmm, d) returns a distribution over X_{t-d}
  inputs: e_t, the current evidence for time step t
          hmm, a hidden Markov model with S × S transition matrix T
          d, the length of the lag for smoothing
  persistent: t, the current time, initially 1
              f, the forward message P(X_t | e_{1:t}), initially hmm.PRIOR
              B, the d-step backward transformation matrix, initially the identity matrix
              e_{t-d:t}, double-ended list of evidence from t − d to t, initially empty
  local variables: O_{t-d}, O_t, diagonal matrices containing the sensor model information

  add e_t to the end of e_{t-d:t}
  O_t ← diagonal matrix containing P(e_t | X_t)
  if t > d then
     f ← FORWARD(f, e_{t-d})
     remove e_{t-d-1} from the beginning of e_{t-d:t}
     O_{t-d} ← diagonal matrix containing P(e_{t-d} | X_{t-d})
     B ← O_{t-d}^{-1} T^{-1} B T O_t
  else B ← B T O_t
  t ← t + 1
  if t > d + 1 then return NORMALIZE(f × B1) else return null
```

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

Specification of the prior, transition model, and sensor model for the umbrella DBN

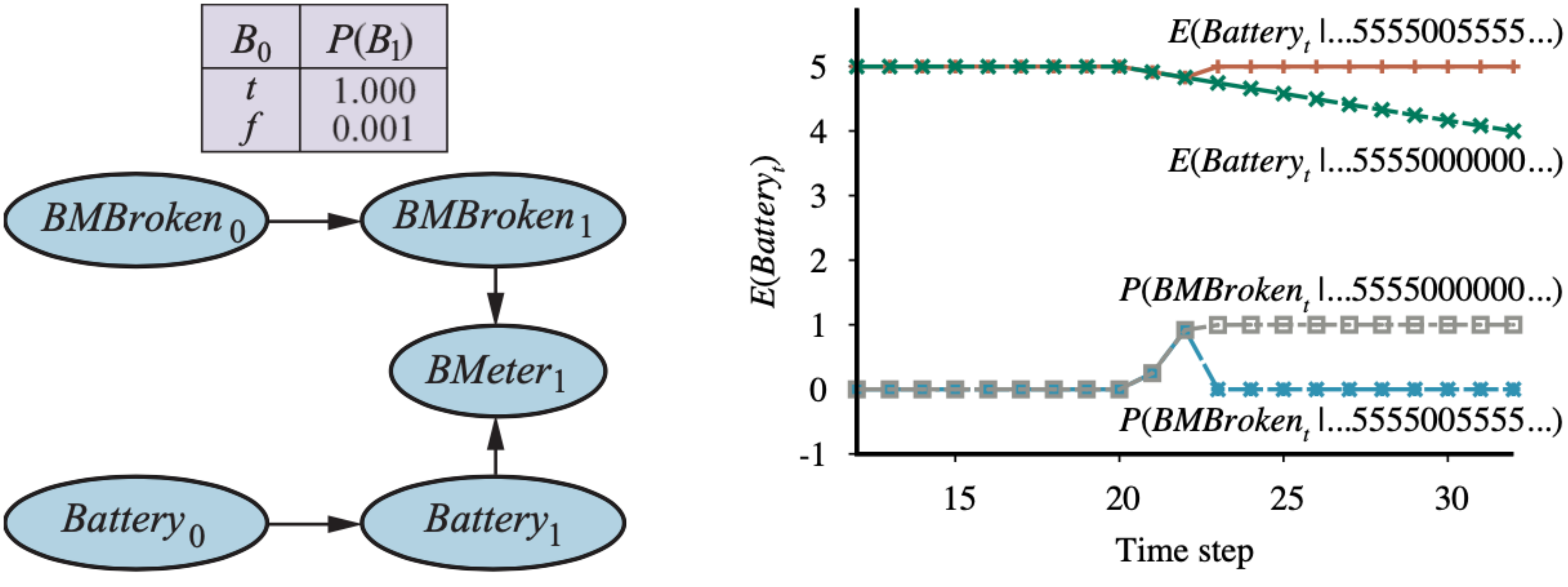




Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

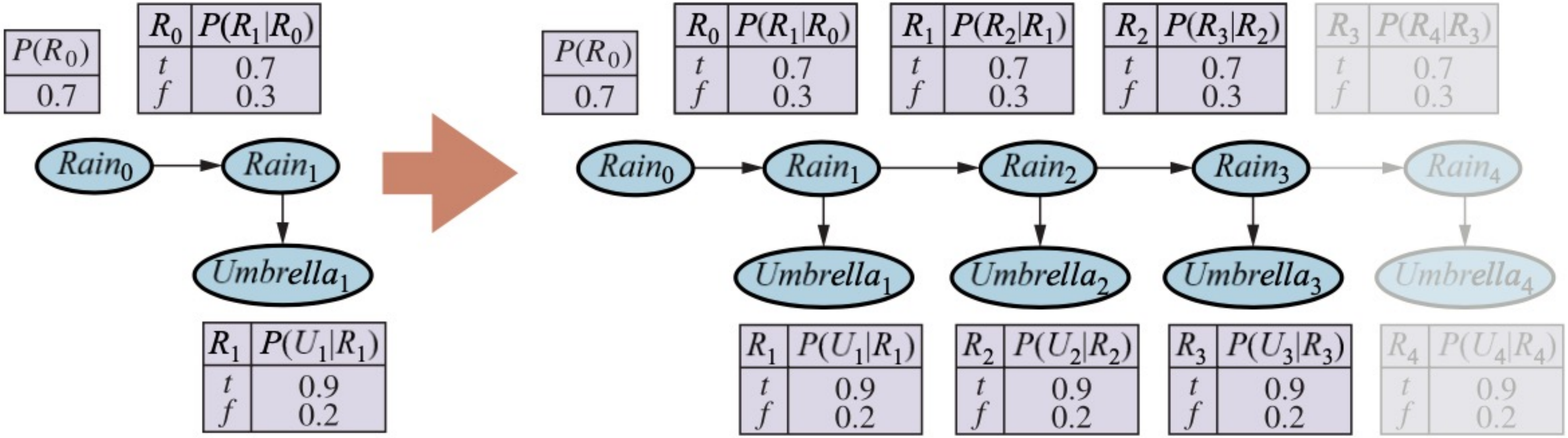
A DBN fragment

the sensor status variable required for modeling persistent failure of the battery sensor



(a) (b)

Unrolling a Dynamic Bayesian Network



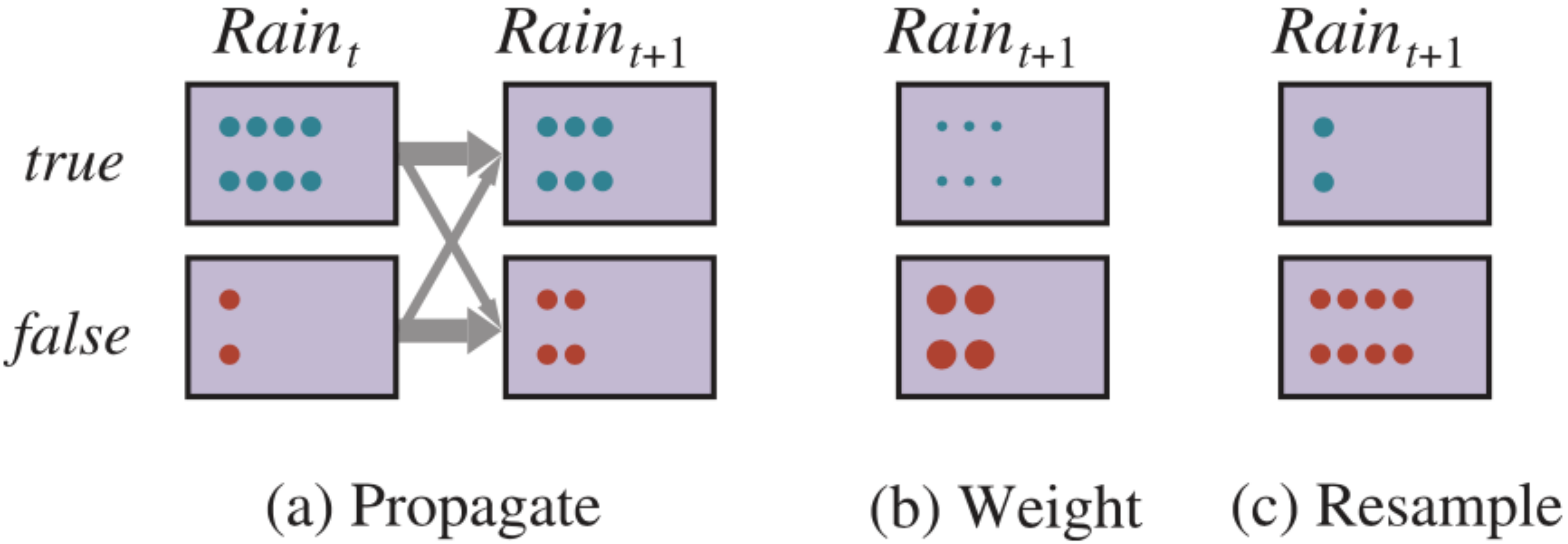
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

The Particle Filtering Algorithm

function PARTICLE-FILTERING($\mathbf{e}$, N, dbn) **returns** a set of samples for the next time step
  inputs: $\mathbf{e}$, the new incoming evidence
    N, the number of samples to be maintained
    dbn, a DBN defined by $\mathbf{P}(\mathbf{X}_0)$, $\mathbf{P}(\mathbf{X}_1 \mid \mathbf{X}_0)$, and $\mathbf{P}(\mathbf{E}_1 \mid \mathbf{X}_1)$
  persistent: S, a vector of samples of size N, initially generated from $\mathbf{P}(\mathbf{X}_0)$
  local variables: W, a vector of weights of size N

  for i = 1 to N **do**
    $S[i] \leftarrow$ sample from $\mathbf{P}(\mathbf{X}_1 \mid \mathbf{X}_0 = S[i])$ *// step 1*
    $W[i] \leftarrow \mathbf{P}(\mathbf{e} \mid \mathbf{X}_1 = S[i])$ *// step 2*
  $S \leftarrow$ WEIGHTED-SAMPLE-WITH-REPLACEMENT(N, S, W) *// step 3*
  return S

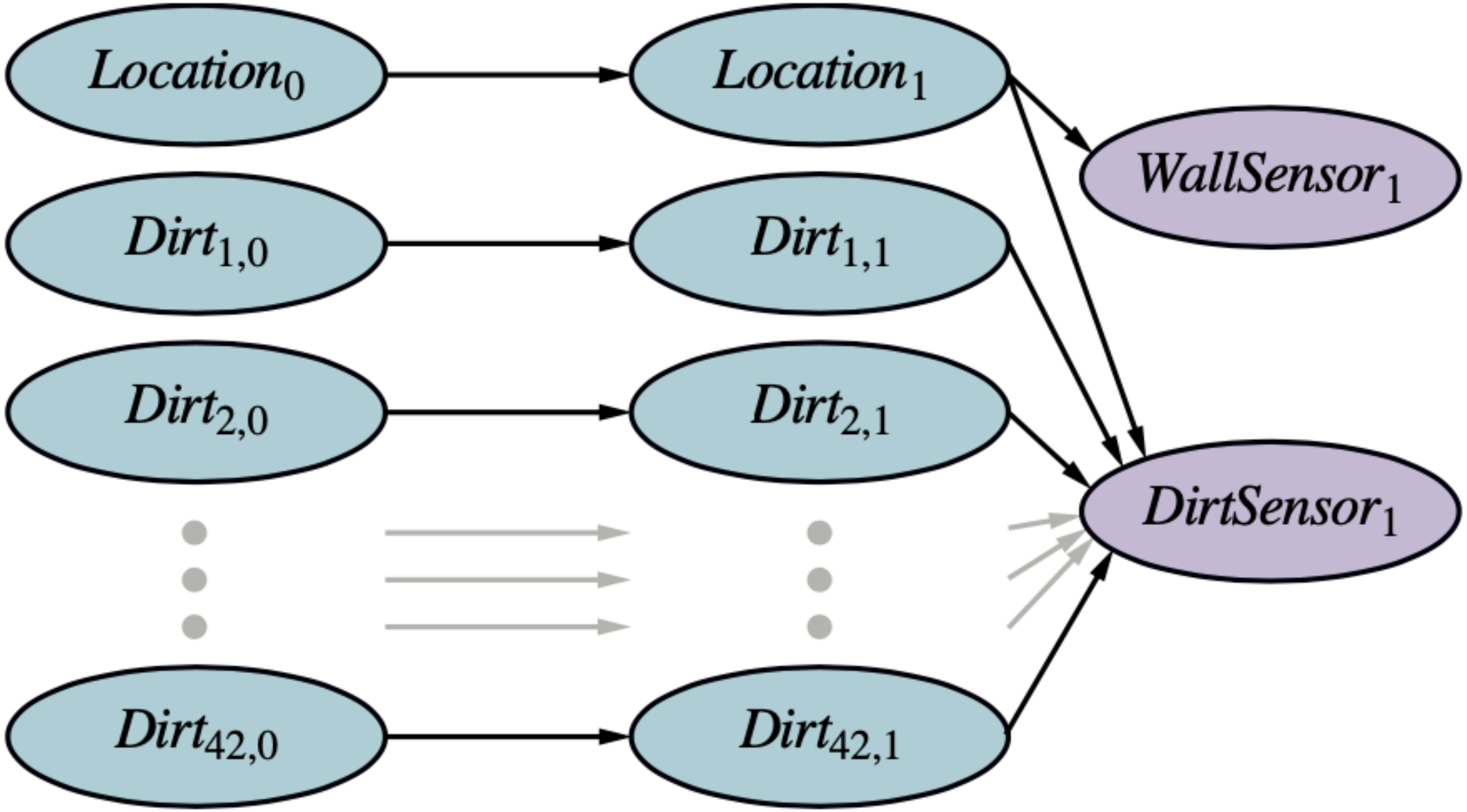
The Particle Filtering Update Cycle for the Umbrella DBN



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

A Dynamic Bayes Net

for simultaneous localization and mapping in the stochastic-dirt vacuum world

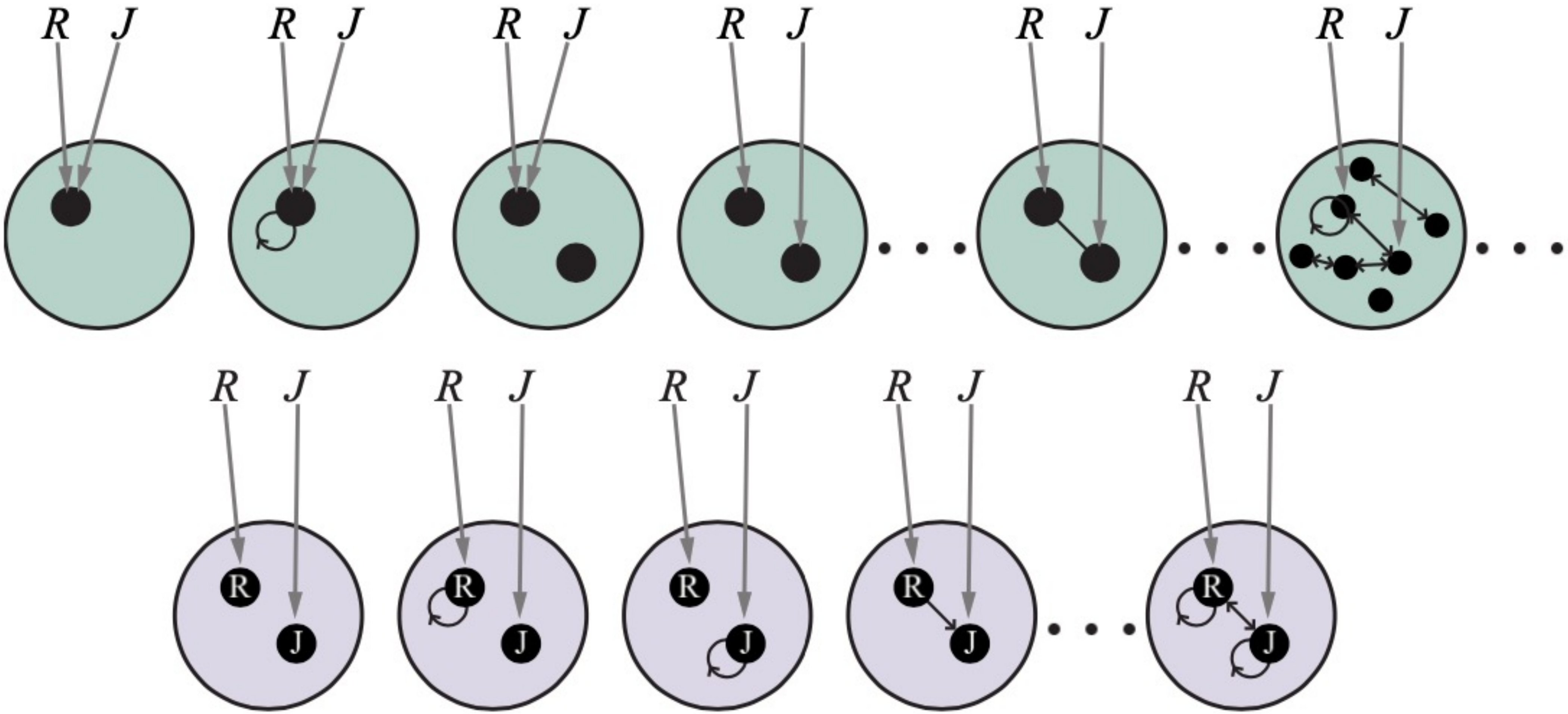


Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Probabilistic Programming

Possible Worlds

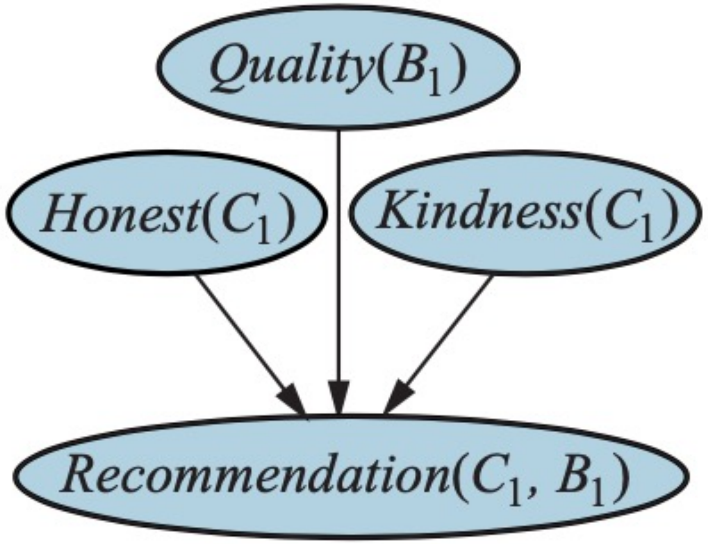
for a language with two constant symbols, R and J



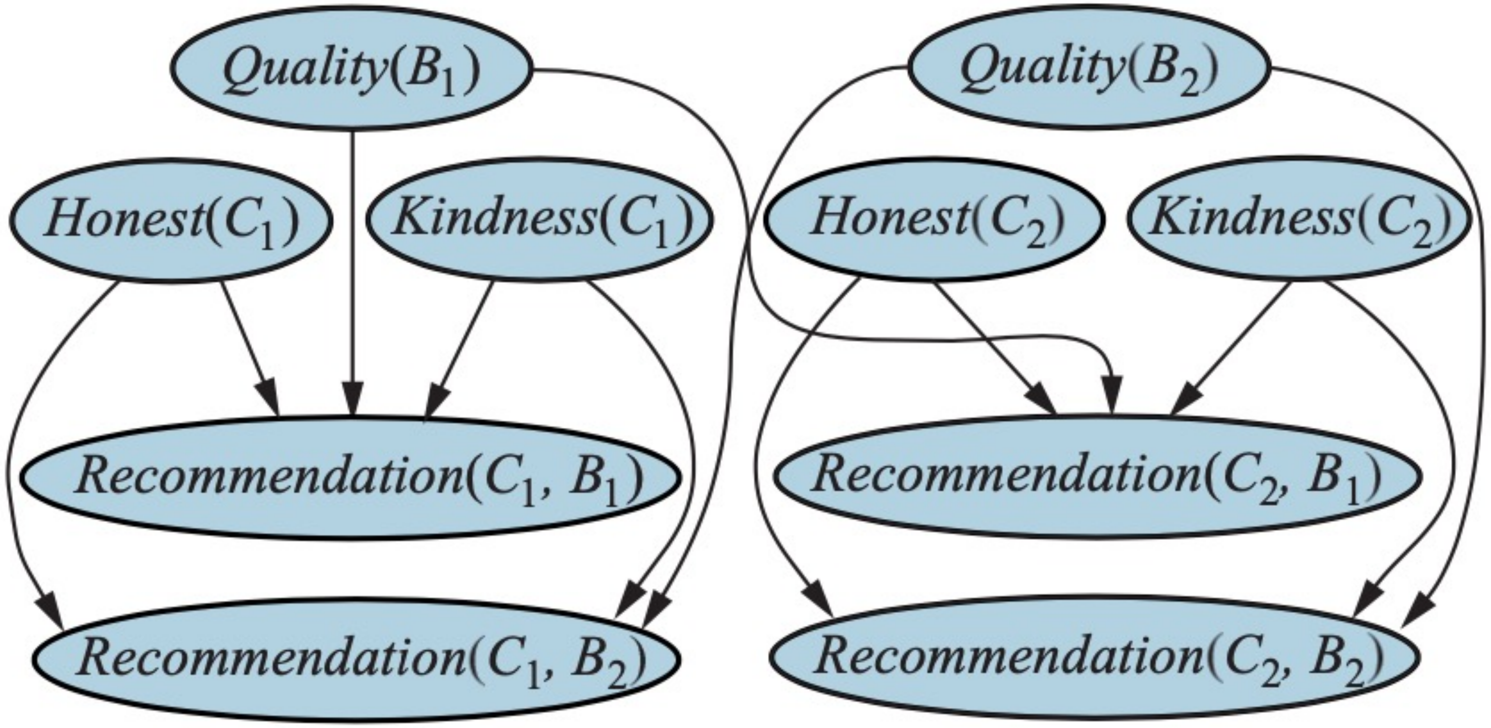
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Bayes Net for a Single customer C1

recommending a single book B1. Honest(C1) is Boolean



(a)

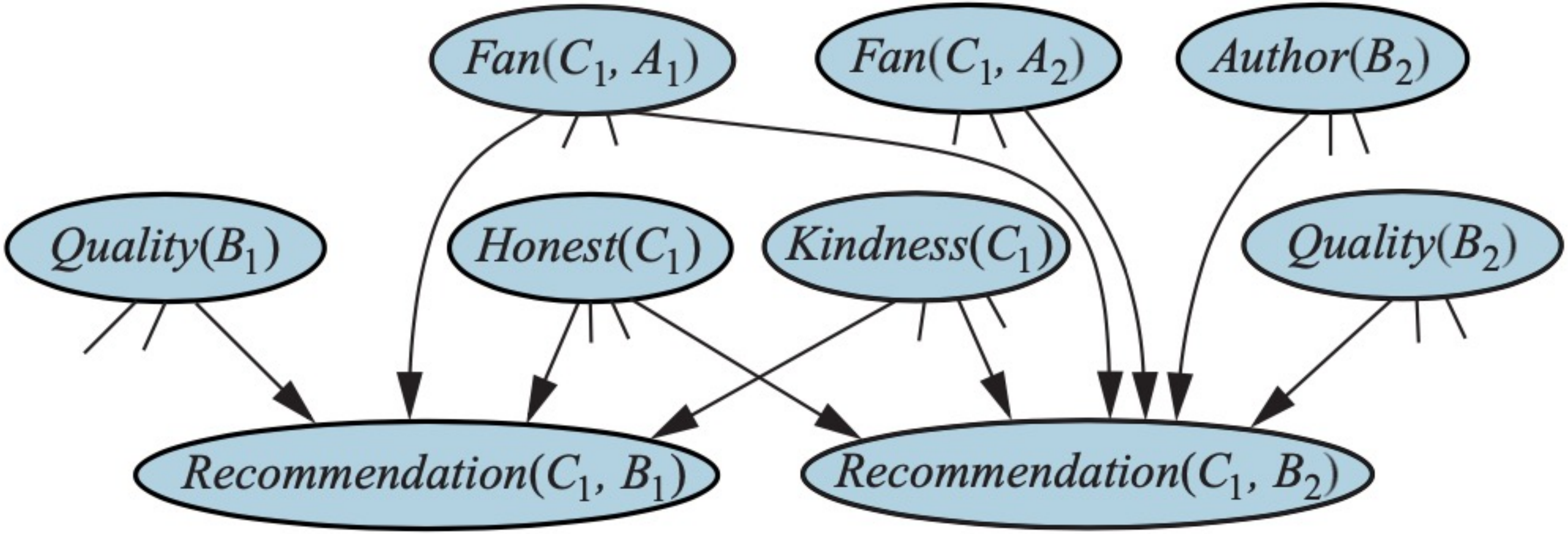


(b)

Bayes net with two customers and two books

Bayes Net

for the book recommendation when Author(B2) is unknown



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

One particular world for the book recommendation OUPM

Variable	Value	Probability
$\#Customer$	2	0.3333
$\#Book$	3	0.3333
$Honest_{\langle Customer,,1\rangle}$	*true*	0.99
$Honest_{\langle Customer,,2\rangle}$	*false*	0.01
$Kindness_{\langle Customer,,1\rangle}$	4	0.3
$Kindness_{\langle Customer,,2\rangle}$	1	0.1
$Quality_{\langle Book,,1\rangle}$	1	0.05
$Quality_{\langle Book,,2\rangle}$	3	0.4
$Quality_{\langle Book,,3\rangle}$	5	0.15
$\#LoginID_{\langle Owner,\langle Customer,,1\rangle\rangle}$	1	1.0
$\#LoginID_{\langle Owner,\langle Customer,,2\rangle\rangle}$	2	0.25
$Recommendation_{\langle LoginID,\langle Owner,\langle Customer,,1\rangle\rangle,1\rangle,\langle Book,,1\rangle}$	2	0.5
$Recommendation_{\langle LoginID,\langle Owner,\langle Customer,,1\rangle\rangle,1\rangle,\langle Book,,2\rangle}$	4	0.5
$Recommendation_{\langle LoginID,\langle Owner,\langle Customer,,1\rangle\rangle,1\rangle,\langle Book,,3\rangle}$	5	0.5
$Recommendation_{\langle LoginID,\langle Owner,\langle Customer,,2\rangle\rangle,1\rangle,\langle Book,,1\rangle}$	5	0.4
$Recommendation_{\langle LoginID,\langle Owner,\langle Customer,,2\rangle\rangle,1\rangle,\langle Book,,2\rangle}$	5	0.4
$Recommendation_{\langle LoginID,\langle Owner,\langle Customer,,2\rangle\rangle,1\rangle,\langle Book,,3\rangle}$	1	0.4
$Recommendation_{\langle LoginID,\langle Owner,\langle Customer,,2\rangle\rangle,2\rangle,\langle Book,,1\rangle}$	5	0.4
$Recommendation_{\langle LoginID,\langle Owner,\langle Customer,,2\rangle\rangle,2\rangle,\langle Book,,2\rangle}$	5	0.4
$Recommendation_{\langle LoginID,\langle Owner,\langle Customer,,2\rangle\rangle,2\rangle,\langle Book,,3\rangle}$	1	0.4

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

An OUPM for Citation Information Extraction

type Researcher, Paper, Citation
random String Name(Researcher)
random String Title(Paper)
random Paper PubCited(Citation)
random String Text(Citation)
random Boolean Professor(Researcher)
origin Researcher Author(Paper)

$$\#Researcher \sim OM(3,1)$$
$$Name(r) \sim NamePrior()$$
$$Professor(r) \sim Boolean(0.2)$$
$$\#Paper(Author = r) \sim \textbf{if } Professor(r) \textbf{ then } OM(1.5, 0.5) \textbf{ else } OM(1, 0.5)$$
$$Title(p) \sim PaperTitlePrior()$$
$$CitedPaper(c) \sim UniformChoice(\{Paper\ p\})$$
$$Text(c) \sim HMMGrammar(Name(Author(CitedPaper(c))), Title(CitedPaper(c)))$$

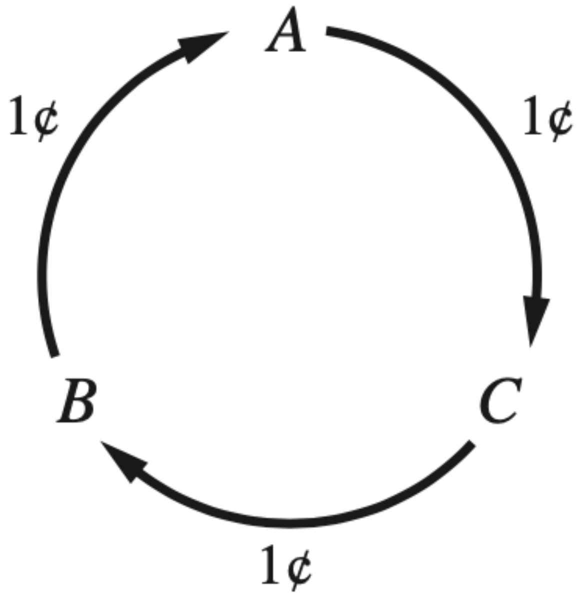
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Making Simple Decisions

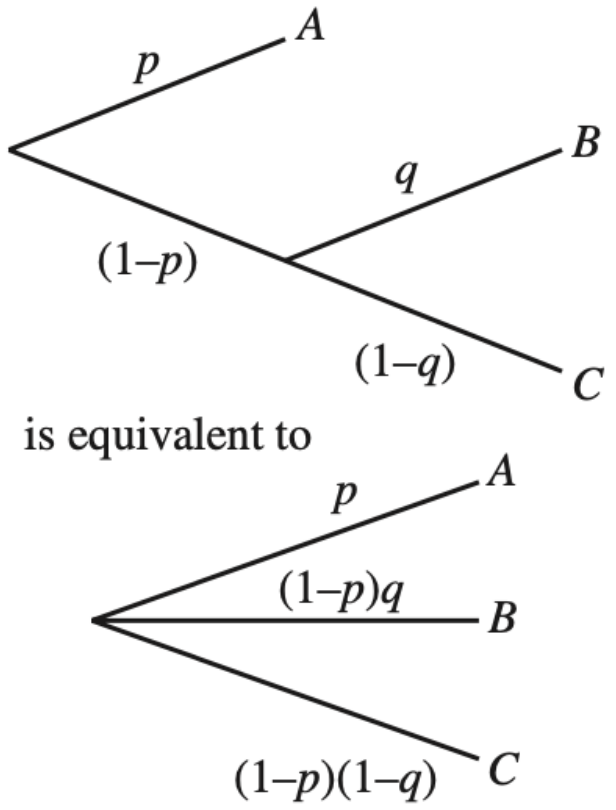
Nontransitive preferences $A \succ B \succ C \succ A$

can result in irrational behavior:

a cycle of exchanges each costing one cent



(a)

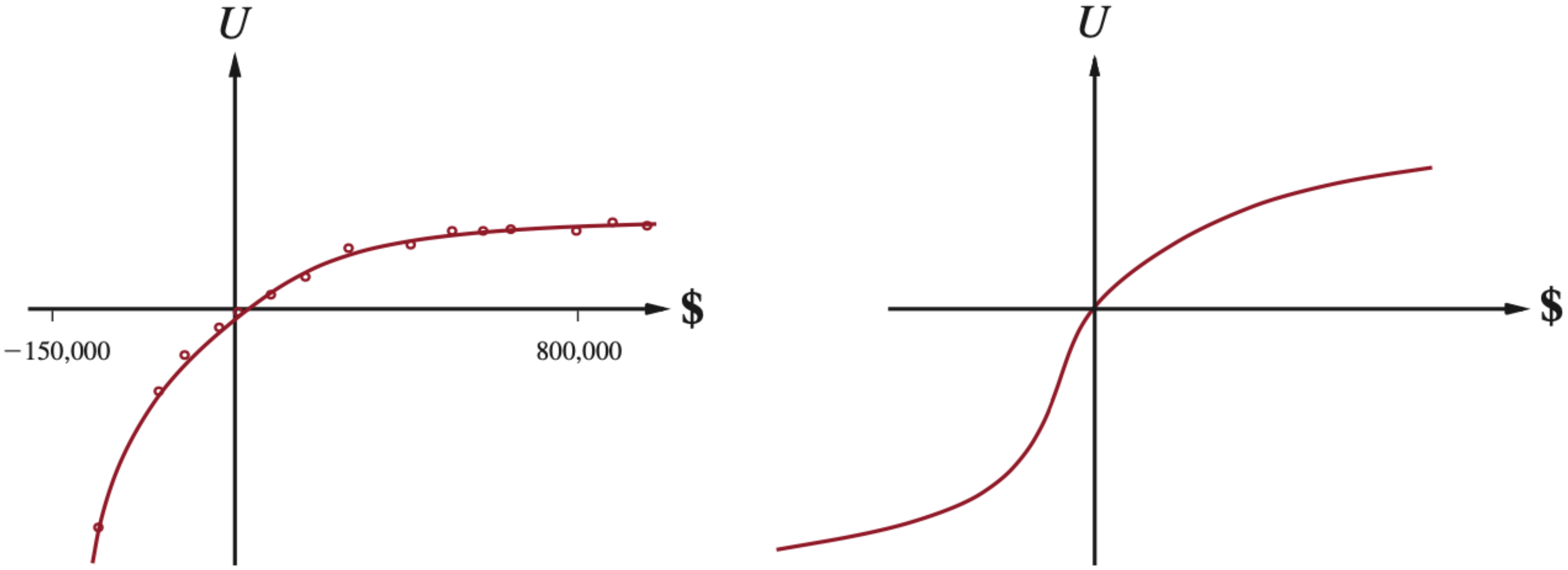


(b)

The decomposability axiom

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

The Utility of Money

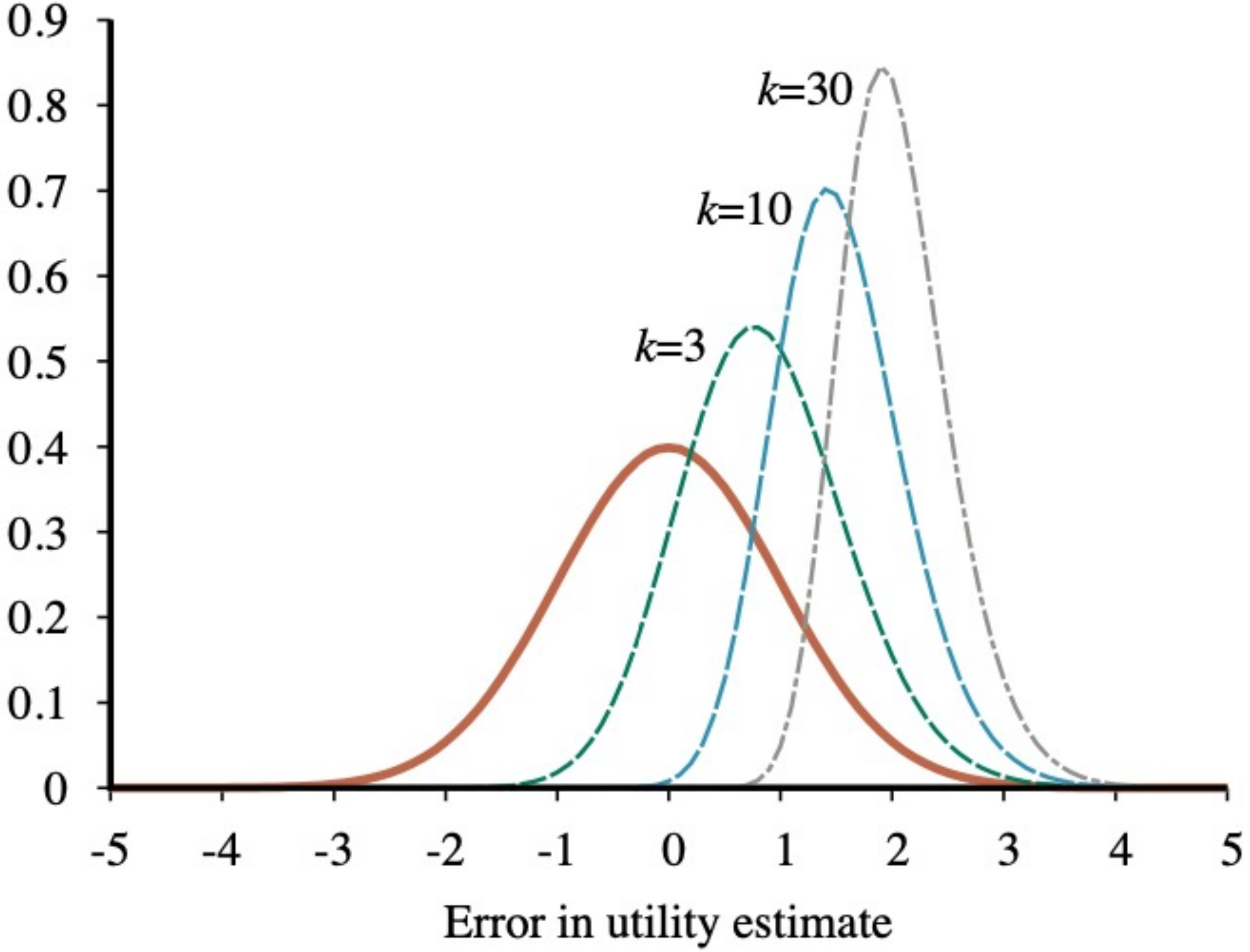


(a) (b)

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

Unjustified optimism

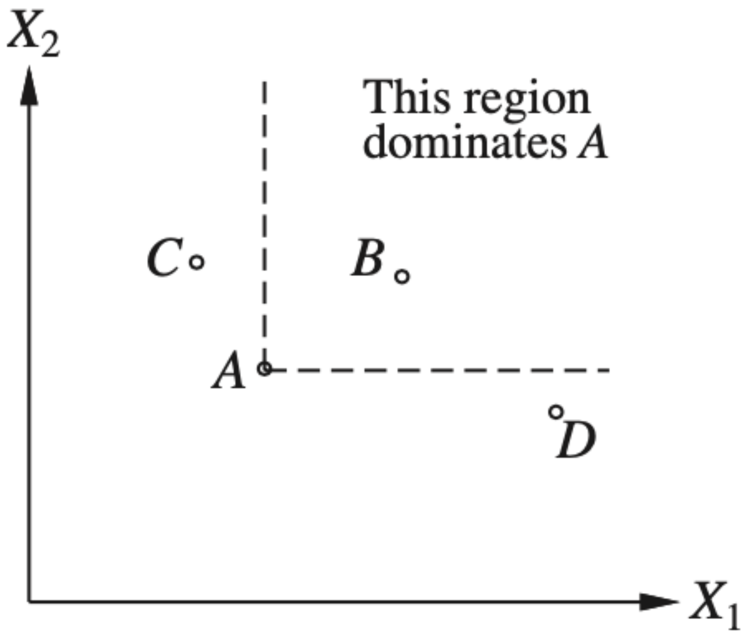
caused by choosing the best of k options



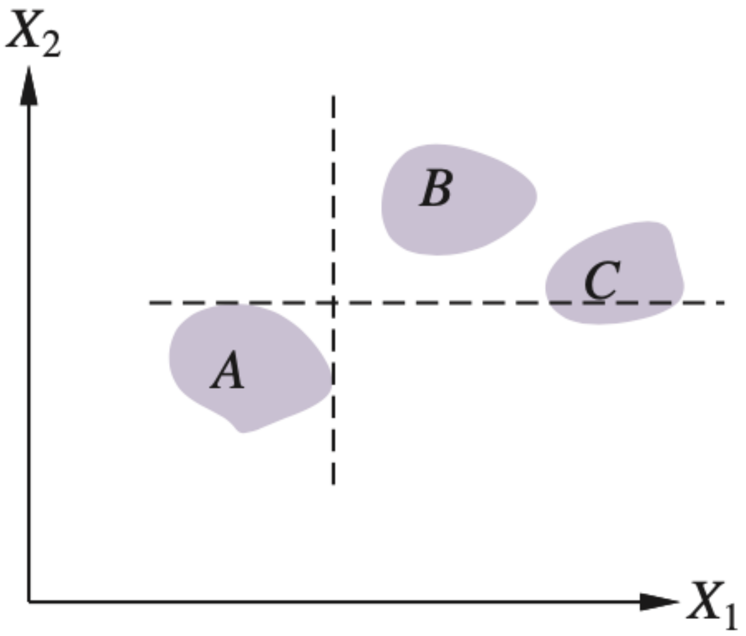
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Strict dominance

(a) Deterministic (b) Uncertain

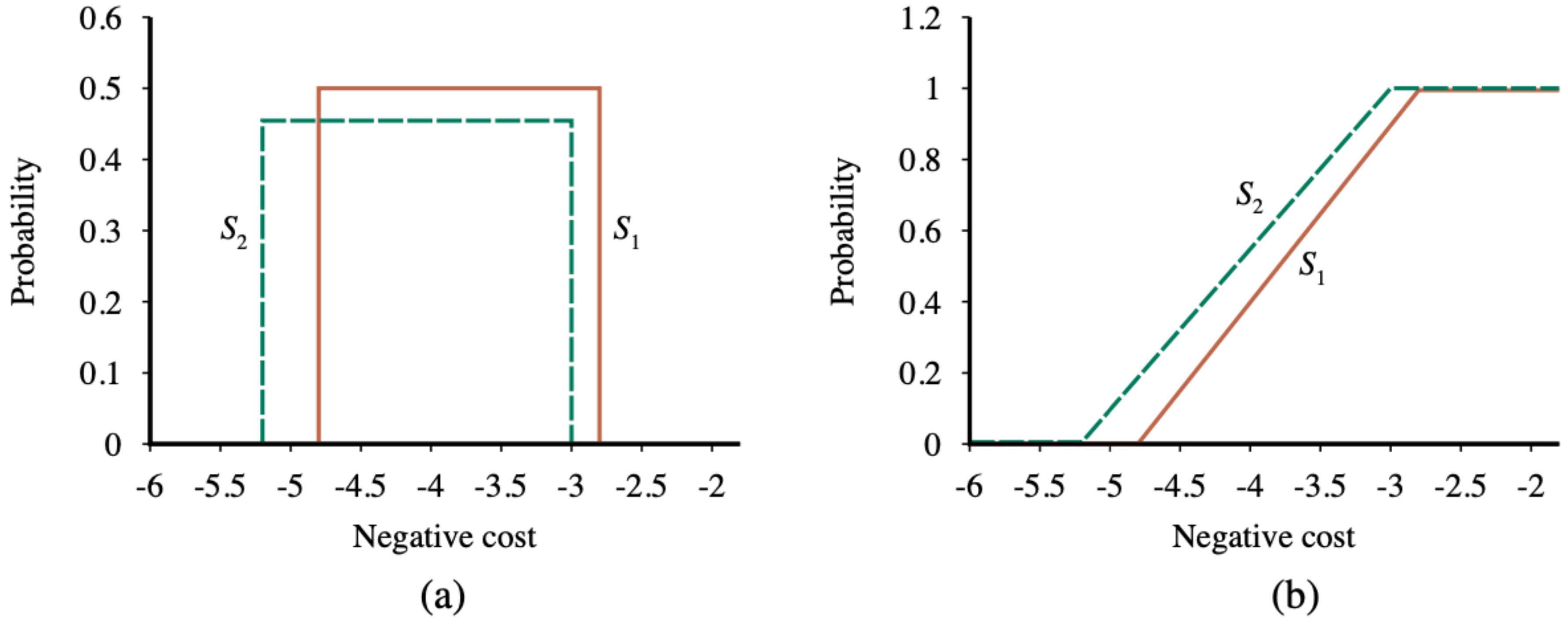


(a)



(b)

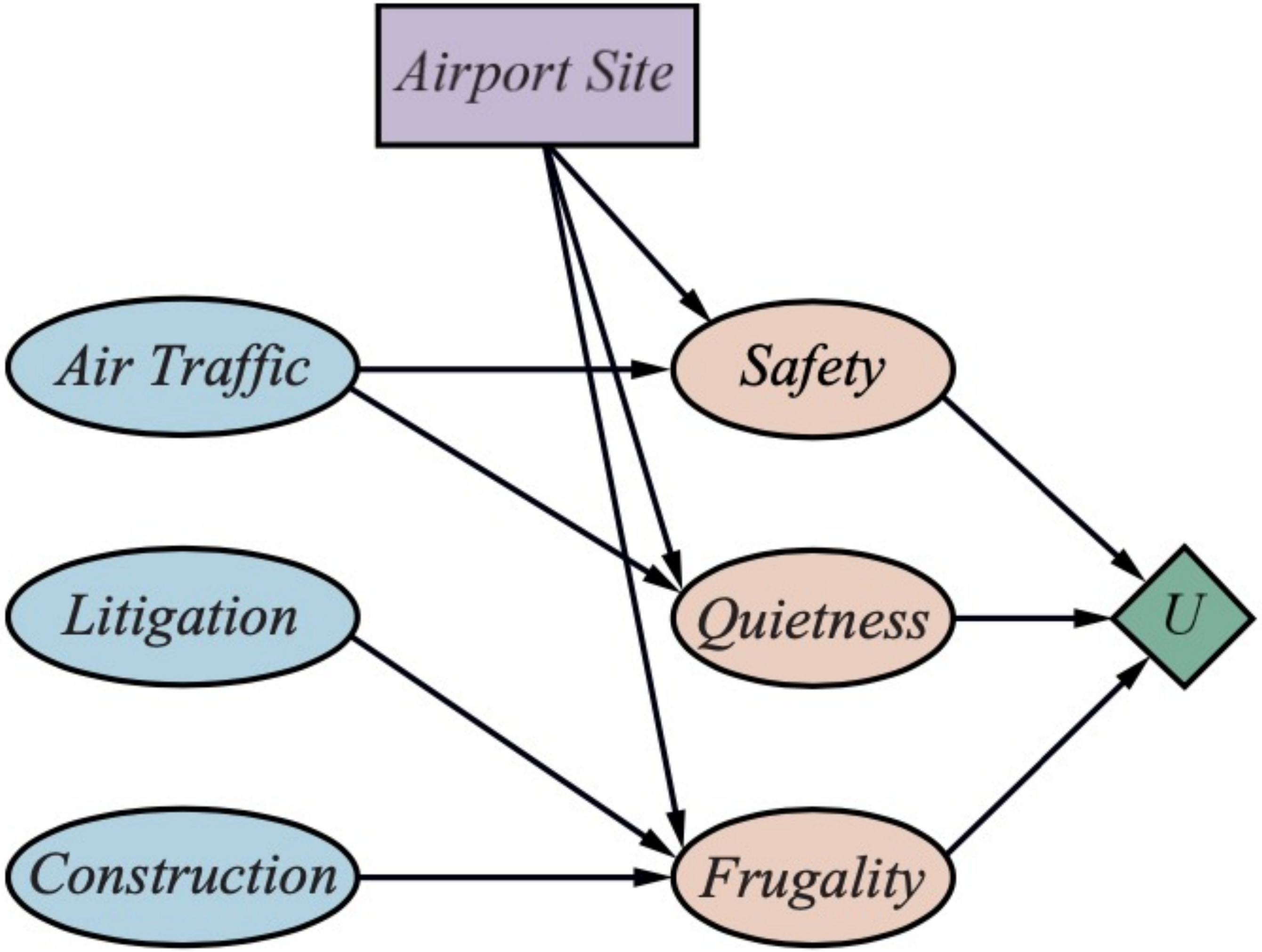
Stochastic dominance



Cumulative distributions for the frugality of S1 and S2.

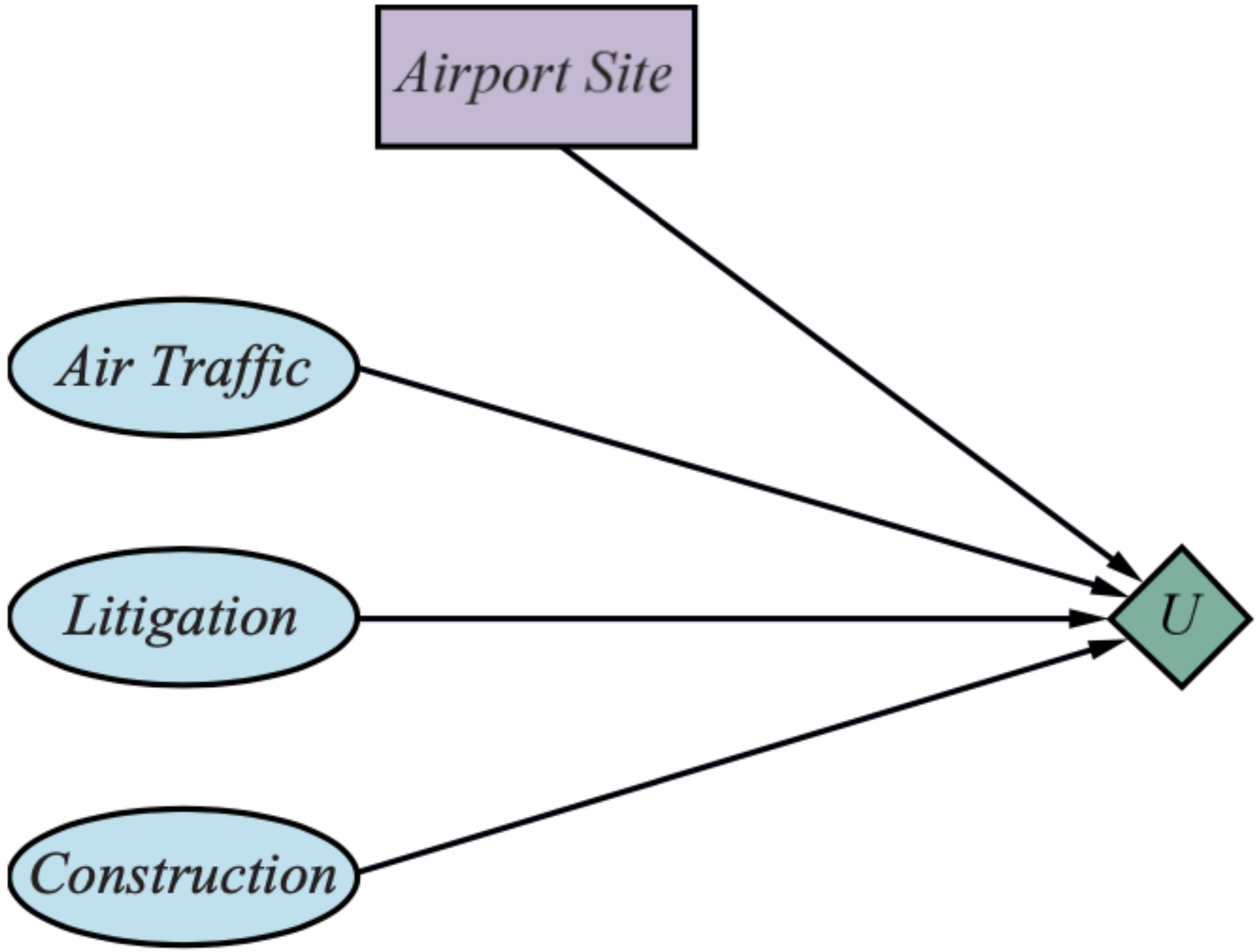
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

A decision network for the airport-siting problem



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

A simplified representation of the airport-siting problem

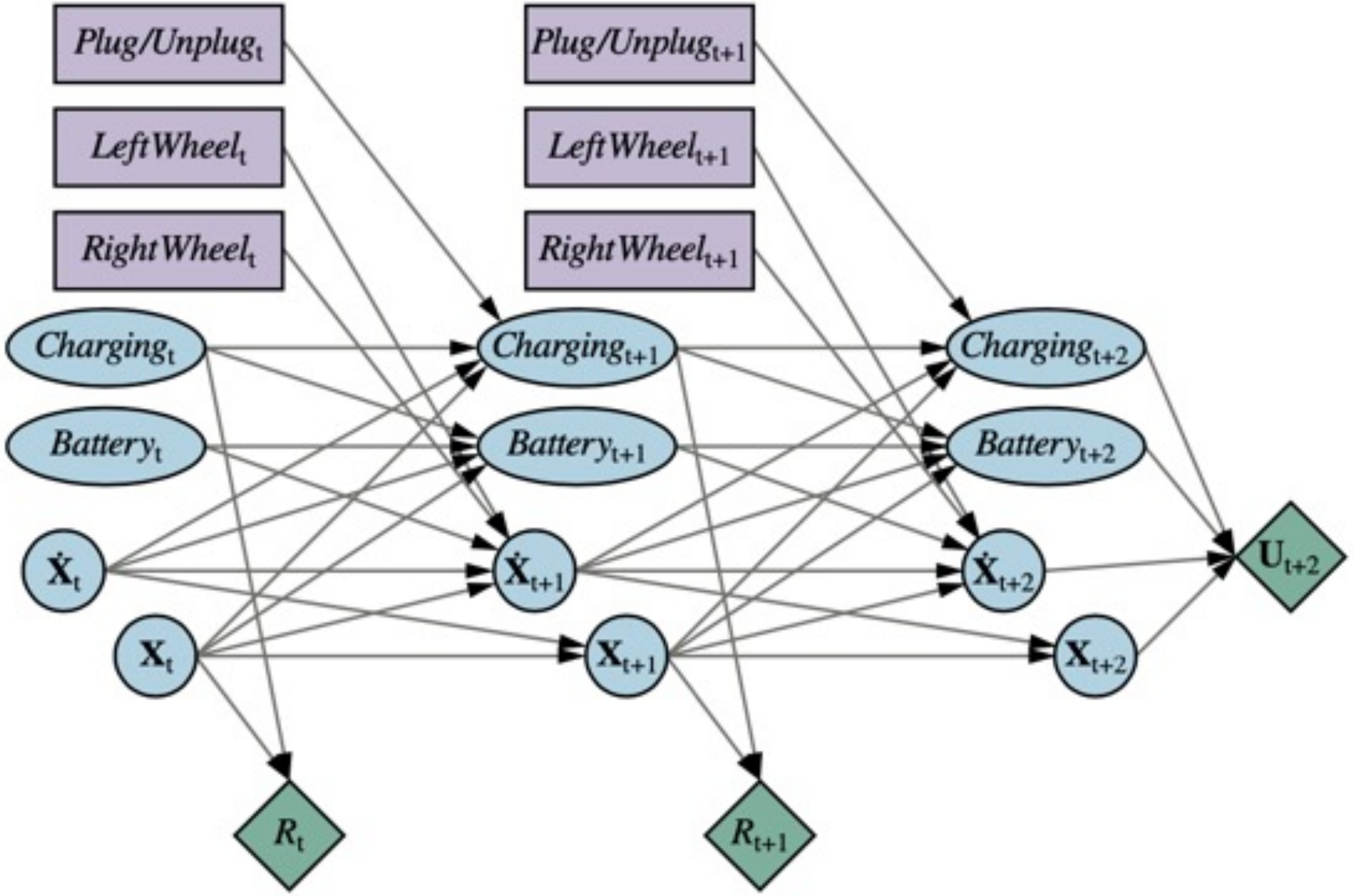


Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Making Complex Decisions

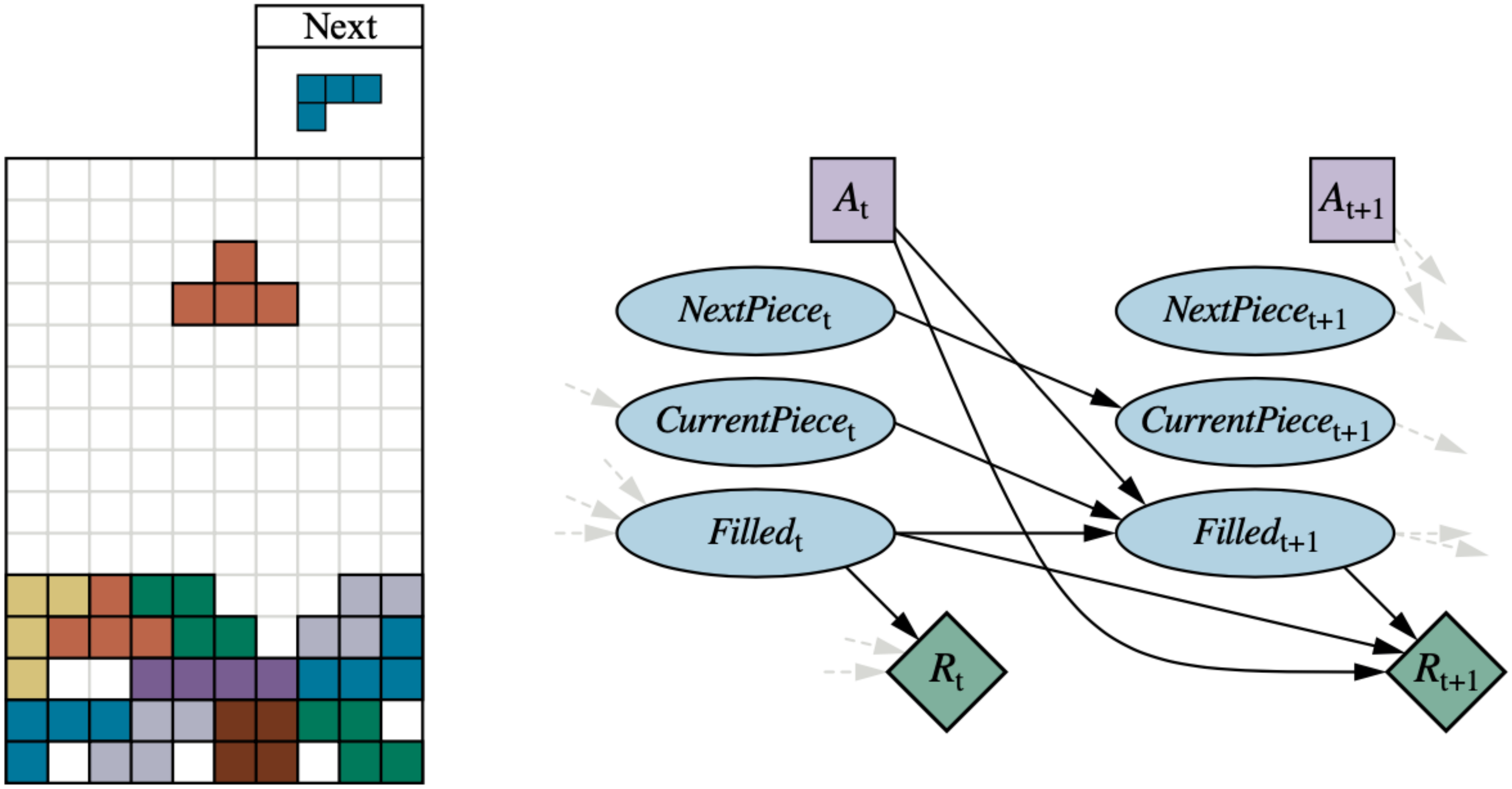
A dynamic decision network

for a mobile robot with state variables for battery level, charging status, location, and velocity, and action variables for the left and right wheel motors and for charging.



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

The game of Tetris
The DDN for the Tetris MDP



(a) (b)

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

The Value Iteration Algorithm for calculating utilities of states

function VALUE-ITERATION(mdp, ϵ) **returns** a utility function
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$,
    rewards $R(s, a, s')$, discount γ
    ϵ, the maximum error allowed in the utility of any state
  local variables: U, U', vectors of utilities for states in S, initially zero
    δ, the maximum relative change in the utility of any state

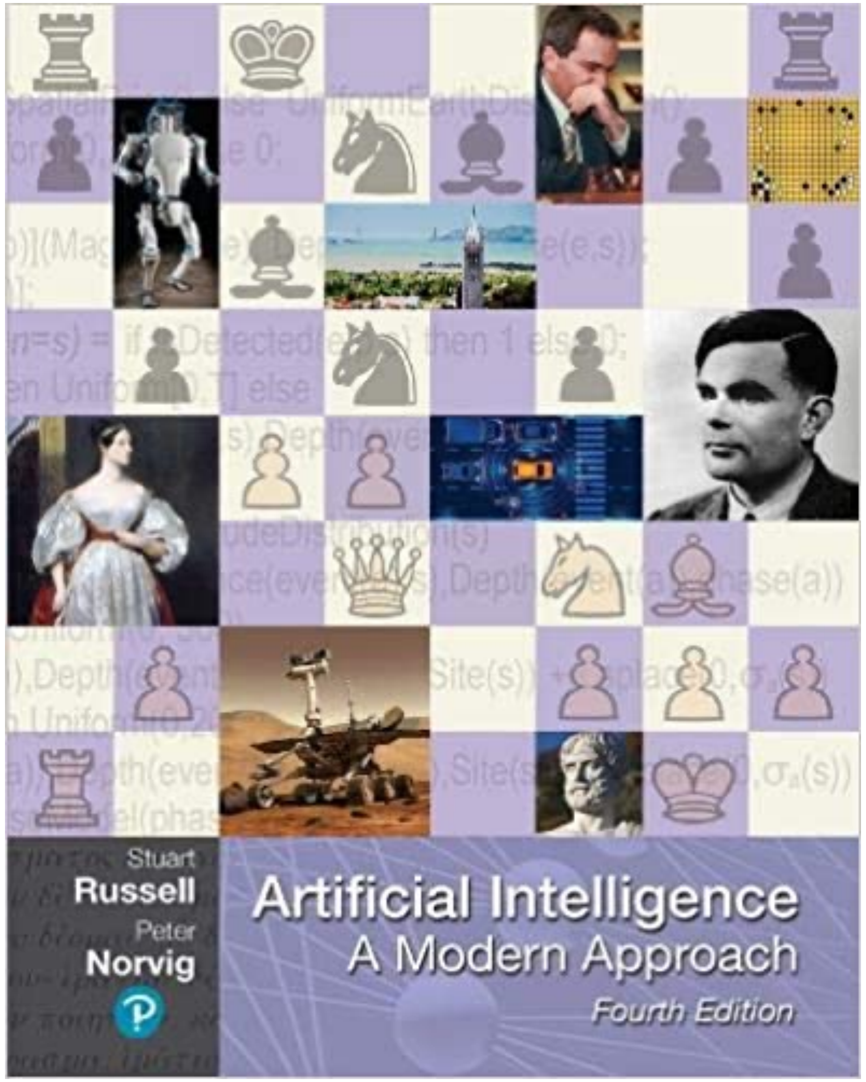
  repeat
    $U \leftarrow U'$; $\delta \leftarrow 0$
    for each state s **in** S **do**
      $U'[s] \leftarrow \max_{a \in A(s)}$ Q-VALUE(mdp, s, a, U)
      if $|U'[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U'[s] - U[s]|$
  until $\delta \leq \epsilon(1-\gamma)/\gamma$
  return U

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

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 (AIMA)

- **Artificial Intelligence: A Modern Approach (AIMA)**
  - http://aima.cs.berkeley.edu/
- **AIMA Python**
  - http://aima.cs.berkeley.edu/python/readme.html
  - https://github.com/aimacode/aima-python
- **Logic, KB Agent**
  - http://aima.cs.berkeley.edu/python/logic.html
- **Probability Models (DTAgent)**
  - http://aima.cs.berkeley.edu/python/probability.html
- **Markov Decision Processes (MDP)**
  - http://aima.cs.berkeley.edu/python/mdp.html

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

Artificial Intelligence: A Modern Approach (AIMA)

- ⌂ US Edition
- ⌂ Global Edition
- Acknowledgements
- Code
- Courses
- Editions
- Errata
- Exercises
- Figures
- Instructors Page
- Pseudocode
- Reviews

Artificial Intelligence: A Modern Approach, 4th US ed.

by Stuart Russell and Peter Norvig

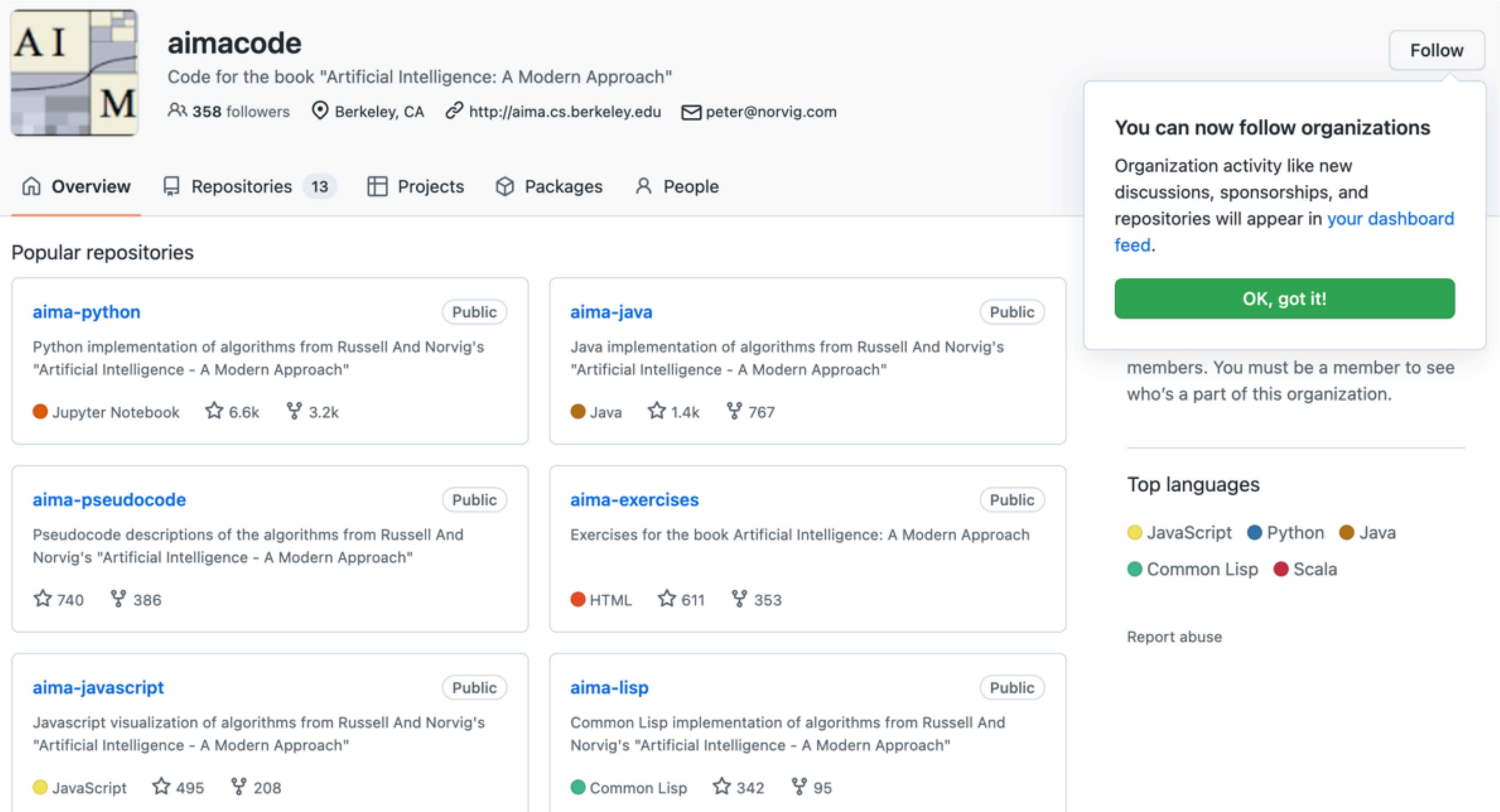
The authoritative, most-used AI textbook, adopted by over 1500 schools.

Table of Contents for the US Edition (or see the Global Edition)

Exercises (website)
Figures (pdf)
Code (website); Pseudocode (pdf)
Covers: US, Global

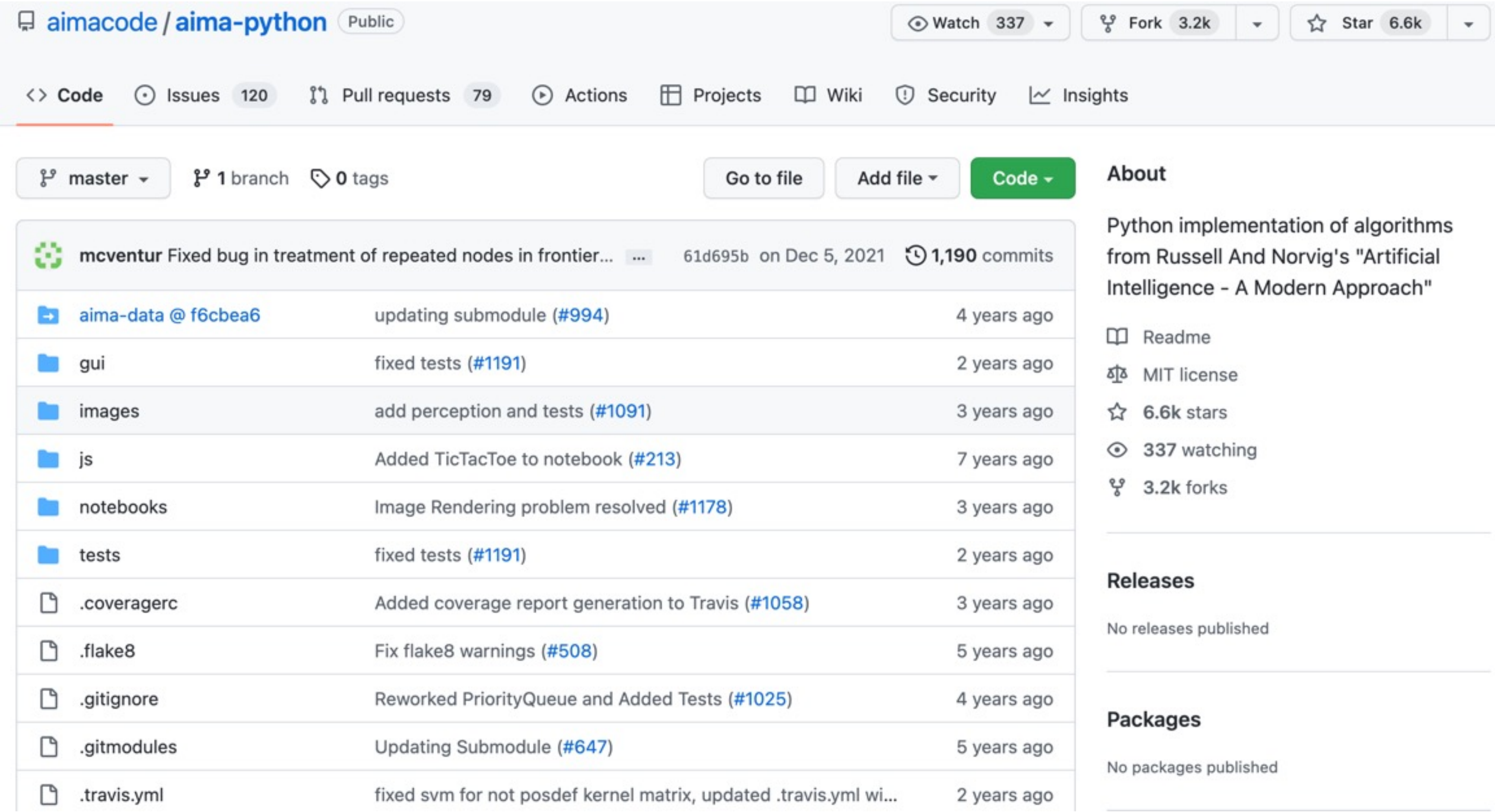
http://aima.cs.berkeley.edu/

AIMA Code



https://github.com/aimacode

AIMA Python



https://github.com/aimacode/aima-python

Papers with Code
State-of-the-Art (SOTA)

Search for papers, code and tasks | Browse State-of-the-Art | Follow | Discuss | Trends | About | Log In/Register

Browse State-of-the-Art

1509 leaderboards • 1327 tasks • 1347 datasets • 17810 papers with code

Follow on Twitter for updates

Computer Vision

Task	Leaderboards	Papers with code
Semantic Segmentation	33 leaderboards	667 papers with code
Image Classification	52 leaderboards	564 papers with code
Object Detection	54 leaderboards	467 papers with code
Image Generation	51 leaderboards	231 papers with code
Pose Estimation	40 leaderboards	231 papers with code

▸ See all 707 tasks

Natural Language Processing

Machine Translation | Language Modelling | Question Answering | Sentiment Analysis | Text Generation

https://paperswithcode.com/sota

Python in Google Colab (Python101)

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

python101.ipynb - Colaboratory

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT?authuser=2#scrollTo=wsh36fLxDKC3

python101.ipynb

File Edit View Insert Runtime Tools Help

COMMENT SHARE

CODE TEXT CELL CELL CONNECTED EDITING

```python
# Future Value
pv = 100
r = 0.1
n = 7
fv = pv * ((1 + (r)) ** n)
print(round(fv, 2))
```

```
194.87
```

[11]

```python
amount = 100
interest = 10 #10% = 0.01 * 10
years = 7

future_value = amount * ((1 + (0.01 * interest)) ** years)
print(round(future_value, 2))
```

```
194.87
```

[12]

```python
# Python Function def
def getfv(pv, r, n):
    fv = pv * ((1 + (r)) ** n)
    return fv
fv = getfv(100, 0.1, 7)
print(round(fv, 2))
```

```
194.87
```

[13]

```python
# Python if else
score = 80
if score >=60 :
    print("Pass")
else:
    print("Fail")
```

```
Pass
```

https://tinyurl.com/aintpupython101

Summary

- Knowledge and Reasoning
  - Logical Agents
  - First-Order Logic
  - Inference in First-Order Logic
  - Knowledge Representation
  - Knowledge Graph (KG)
- Uncertain Knowledge and Reasoning
  - Quantifying Uncertainty
  - Probabilistic Reasoning
  - Making Complex Decisions

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.
- Steven D'Ascoli (2022), Artificial Intelligence and Deep Learning with Python: Every Line of Code Explained For Readers New to AI and New to Python, Independently published.
- Nithin Buduma, Nikhil Buduma, Joe Papa (2022), Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms, 2nd Edition, O'Reilly Media.