Module: Knowledge, Reasoning, and Planning
Part 2

Logical Agents
R&N: Chapter 7
Illustrative example: Wumpus World

Performance measure
- gold +1000,
- death -1000
(falling into a pit or being eaten by the wumpus)
- -1 per step, -10 for using the arrow

Environment
- Rooms / squares connected by doors.
- Squares adjacent to wumpus are smelly
- Squares adjacent to pit are breezy
- Glitter iff gold is in the same square
- Shooting kills wumpus if you are facing it
- Shooting uses up the only arrow
- Grabbing picks up gold if in same square
- Releasing drops the gold in same square
- Randomly generated at start of game. Wumpus only senses current room.

Sensors: Stench, Breeze, Glitter, Bump, Scream  [perceptual inputs]
Actuators: Left turn, Right turn, Forward, Grab, Release, Shoot

(Somewhat whimsical!)
# Wumpus world characterization

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully Observable</td>
<td>No – only local perception</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Yes – outcomes exactly specified</td>
</tr>
<tr>
<td>Static</td>
<td>Yes – Wumpus and Pits do not move</td>
</tr>
<tr>
<td>Discrete</td>
<td>Yes</td>
</tr>
<tr>
<td>Single-agent?</td>
<td>Yes – Wumpus is essentially a “natural feature.”</td>
</tr>
</tbody>
</table>
Exploring a wumpus world

The knowledge base of the agent consists of the rules of the Wumpus world plus the percept “nothing” in [1,1]

Boolean percept feature values: \(<0, 0, 0, 0, 0, 0>\)

Stench, Breeze, Glitter, Bump, Scream
T=0 The KB of the agent consists of the rules of the Wumpus world plus the percept “nothing” in [1,1]. By inference, the agent’s knowledge base also has the information that [1,2] and [2,1] are okay. Added as propositions.
Further exploration

@ T = 1 What follows?

Pit(2,2) or Pit(3,1)

Where next?

None, breeze, none, none, none

A – agent
V – visited
B - breeze

Stench, Breeze, Glitter, Bump, Scream
Where is Wumpus?

Wumpus cannot be in (1,1) or in (2,2) (Why?) \(\Rightarrow\) Wumpus in (1,3)

Not breeze in (1,2) \(\Rightarrow\) no pit in (2,2); but we know there is
pit in (2,2) or (3,1) \(\Rightarrow\) pit in (3,1)
We reasoned about the possible states the Wumpus world can be in, given our percepts and our knowledge of the rules of the Wumpus world. 
I.e., the content of KB at T=3.

What follows is what holds true in all those worlds that satisfy what is known at that time T=3 about the particular Wumpus world we are in.

Example property: P_in_(3,1)

\[ \text{Models(KB)} \subseteq \text{Models(P_in_\(3,1\))} \]

Essence of logical reasoning:
Given all we know, Pit_in_(3,1) holds. ("The world cannot be different.")
Formally: Entailment

Knowledge Base (KB) in the Wumpus World →
Rules of the wumpus world + new percepts

Situation after detecting nothing in [1,1], moving right, breeze in [2,1]. I.e. T=1.

Consider possible models for KB with respect to the cells (1,2), (2,2) and (3,1), with respect to the existence or non existence of pits

3 Boolean choices ⇒
8 possible interpretations
(enumerate all the models or “possible worlds” wrt Pit location)
Is KB consistent with all 8 possible worlds?

Worlds that violate KB (are inconsistent with what we know)

\[ KB = \text{Wumpus-world rules + observations (T=1)} \]

Q: Why does world violate KB?
So, KB defines all worlds that we hold possible.

Queries: we want to know the properties of those worlds. That’s how the semantics of logical entailment is defined.

Models of the KB and $\alpha_1$

$KB = \text{Wumpus-world rules} + \text{observations}$

$\alpha_1 = "[1,2] \text{ has no pit}"$, $KB \models \alpha_1$

- In every model in which KB is true, $\alpha_1$ is True (proved by “model checking”)

Note: $\alpha_1$ holds in more models than KB. That’s OK, but we don’t care about those worlds.
KB = wumpus-world rules + observations

$\alpha_2 = "[2,2] \text{ has no pit}"$, this is only True in some of the models for which KB is True, therefore $KB \not\models \alpha_2$

A model of KB where $\alpha_2$ does NOT hold!
Entailment via “Model Checking”

Inference by Model checking –
We enumerate all the KB models and check if $\alpha_1$ and $\alpha_2$ are True in all the models (which implies that we can only use it when we have a finite number of models).

I.e. using semantics directly.

$$\text{Models}(KB) \subseteq \text{Models}(\alpha)$$

$$KB \models \alpha$$
Example redux: More formal

How do we actually encode background knowledge and percepts in formal language?

None, none, none, none, none
Stench, Breeze, Glitter, Bump, Scream

None, breeze, none, none, none
A – agent
V – visited
B – breeze
Define propositions:

Let $P_{i,j}$ be true if there is a pit in $[i, j]$.
Let $B_{i,j}$ be true if there is a breeze in $[i, j]$.

Sentence 1 (R1): $\neg P_{1,1}$ [Given.]
Sentence 2 (R2): $\neg B_{1,1}$ [Observation $T = 0$.]
Sentence 3 (R3): $B_{2,1}$ [Observation $T = 1$.]

"Pits cause breezes in adjacent squares"
Sentence 4 (R4): $B_{1,1} \iff (P_{1,2} \lor P_{2,1})$
Sentence 5 (R5): $B_{2,1} \iff (P_{1,1} \lor P_{2,2} \lor P_{3,1})$

etc.

Notes: (1) one such statement about Breeze for each square.
(2) similar statements about Wumpuss, and stench and Gold and glitter. (Need more propositional letters.)
What about Time? What about Actions?

Is Time represented?
No!

Can include time in propositions:
- Explicit time \( P_{i,j,t} \), \( B_{i,j,t} \), \( L_{i,j,t} \) etc.
- Many more props: \( O(TN^2) \) (\( L_{i,j,t} \) for agent at (i,j) at time t)

Now, we can also model actions, use props: Move(i, j, k, l, t)
- E.g. Move(1, 1, 2, 1, 0)

What knowledge axiom(s) capture(s) the effect of an Agent move?

\[
\text{Move}(i, j, k, l, t) \Rightarrow (\neg L(i, j, t+1) \land L(k, l, t+1))
\]

Is this it?
What about i, j, k, and l?
What about Agent location at time t?
Improved: *Move implies a change in the world state; a change in the world state, implies a move occurred!*

\[ \text{Move}(i, j, k, l, t) \iff (L(i, j, t) \land \neg L(i, j, t+1) \land L(k, l, t+1)) \]

For all tuples \((i, j, k, l)\) that represent legitimate possible moves.

E.g. \((1, 1, 2, 1)\) or \((1, 1, 1, 2)\)

Still, some remaining subtleties when representing time and actions. What happens to propositions at time \(t+1\) compared to at time \(t\), that are *not* involved in any action?

E.g. \(P(1, 3, 3)\) is derived at some point.

What about \(P(1, 3, 4)\), True or False?

R&N suggests having \(P\) as an “atemporal var” since it cannot change over time. Nevertheless, we have many other vars that can change over time, called “fluents”.

Values of propositions not involved in any action should not change! “The Frame Problem” / Frame Axioms R&N 7.7.1
Successor-State Axioms

Axiom schema:
F is a fluent (prop. that can change over time)

For example:

\[ L^{t+1}_{1,1} = (L^t_{1,1} \land (\neg Forward^t \lor Bump^{t+1})) \]
\[ \lor (L^t_{1,2} \land (South^t \land Forward^t)) \]
\[ \lor (L^t_{2,1} \land (West^t \land Forward^t)) \]

i.e. \( L_{1,1} \) was “as before” with [no movement action or bump into wall]
or resulted from some action (movement into \( L_{1,1} \)).
Some example inferences
Section 7.7.1 R&N

Actions and inputs up to time 6
Note: includes turns!

\neg Stench^0 \wedge \neg Breeze^0 \wedge \neg Glitter^0 \wedge \neg Bump^0 \wedge \neg Scream^0 \quad ; \quad Forward^0

\neg Stench^1 \wedge Breeze^1 \wedge \neg Glitter^1 \wedge \neg Bump^1 \wedge \neg Scream^1 \quad ; \quad TurnRight^1

\neg Stench^2 \wedge Breeze^2 \wedge \neg Glitter^2 \wedge \neg Bump^2 \wedge \neg Scream^2 \quad ; \quad TurnRight^2

\neg Stench^3 \wedge Breeze^3 \wedge \neg Glitter^3 \wedge \neg Bump^3 \wedge \neg Scream^3 \quad ; \quad Forward^3

\neg Stench^4 \wedge \neg Breeze^4 \wedge \neg Glitter^4 \wedge \neg Bump^4 \wedge \neg Scream^4 \quad ; \quad TurnRight^4

\neg Stench^5 \wedge \neg Breeze^5 \wedge \neg Glitter^5 \wedge \neg Bump^5 \wedge \neg Scream^5 \quad ; \quad Forward^5

\neg Stench^6 \wedge \neg Breeze^6 \wedge \neg Glitter^6 \wedge \neg Bump^6 \wedge \neg Scream^6

\text{Ask}(KB, P_{3,1}) = \text{true}

\text{Ask}(KB, W_{1,3}) = \text{true}

Define “OK”:

\text{OK}^t_{x,y} \iff \neg P_{x,y} \wedge \neg(W_{x,y} \wedge WumpusAlive^t)

\text{Ask}(KB, \text{OK}^6_{2,2}) = \text{true},

so the square [2, 2] is OK

In milliseconds, with modern SAT solver.
No explicit time. Actions are what changes the world from “situation” to “situation”. More elegant, but still need frame axioms to capture what stays the same. Inherent with many representation formalisms: “physical” persistence does not come for free! (and probably shouldn’t)
The goal of logical inference is to decide whether $\mathit{KB} \models \alpha$, for some $\alpha$.

For example, given the rules of the Wumpus World, is $P_{22}$ entailed?

Relevant propositional symbols:

\[
\begin{align*}
R1: & \neg P_{1,1} \\
R2: & \neg B_{1,1} \\
R3: & B_{2,1}
\end{align*}
\]

"Pits cause breezes in adjacent squares"

\[
\begin{align*}
R4: & B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \\
R5: & B_{2,1} \iff (P_{1,1} \lor P_{2,2} \lor P_{3,1})
\end{align*}
\]

Inference by enumeration. We have 7 relevant symbols. Therefore $2^7 = 128$ interpretations.

Need to check if $P_{22}$ is true in all of the $\mathit{KB}$ models (interpretations that satisfy $\mathit{KB}$ sentences).

Q.: $\mathit{KB}$ has many more symbols. Why can we restrict ourselves to these symbols here?

But, be careful, typically we can’t!!
1) \( KB \models \alpha \)  entailment

All equivalent Prop. / FO Logic
Proof techniques

\[ M(KB) \subseteq M(\alpha) \] 

by defn. / semantic proofs / truth tables

“model checking”

(style I, R&N 7.4.4) Done.

\[ KB \vdash \alpha \]

soundness and completeness

logical deduction / symbol pushing

proof by inference rules (style II)

e.g. modus ponens (R&N 7.5.1)

\[ (KB \land \neg \alpha) \text{ is inconsistent} \]

Proof by contradiction

use CNF / clausal form

Resolution (style III, R&N 7.5)

SAT solvers (style IV, R&N 7.6)

most effective
Standard syntax and semantics for propositional logic. (CS-2800; see 7.4.1 and 7.4.2.)

Syntax:

\[
\begin{align*}
\text{Sentence} & \rightarrow \text{AtomicSentence} \mid \text{ComplexSentence} \\
\text{AtomicSentence} & \rightarrow \text{True} \mid \text{False} \mid P \mid Q \mid R \mid \ldots \\
\text{ComplexSentence} & \rightarrow (\text{Sentence}) \mid [\text{Sentence}] \\
& \mid \neg \text{Sentence} \\
& \mid \text{Sentence} \land \text{Sentence} \\
& \mid \text{Sentence} \lor \text{Sentence} \\
& \mid \text{Sentence} \Rightarrow \text{Sentence} \\
& \mid \text{Sentence} \Leftrightarrow \text{Sentence}
\end{align*}
\]

Operator Precedence: $\neg, \land, \lor, \Rightarrow, \Leftrightarrow$
Semantics

Note: Truth value of a sentence is built from its parts “compositional semantics”

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$\neg P$</th>
<th>$P \land Q$</th>
<th>$P \lor Q$</th>
<th>$P \Rightarrow Q$</th>
<th>$P \Leftrightarrow Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>
Logical equivalences

\[
\begin{align*}
(\alpha \land \beta) & \equiv (\beta \land \alpha) \quad \text{commutativity of } \land \\
(\alpha \lor \beta) & \equiv (\beta \lor \alpha) \quad \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) & \equiv (\alpha \land (\beta \land \gamma)) \quad \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) & \equiv (\alpha \lor (\beta \lor \gamma)) \quad \text{associativity of } \lor \\
\neg(\neg \alpha) & \equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \implies \beta) & \equiv (\neg \beta \implies \neg \alpha) \quad \text{contraposition} \\
(\alpha \implies \beta) & \equiv (\neg \alpha \lor \beta) \quad \text{implication elimination} \\
(\alpha \iff \beta) & \equiv ((\alpha \implies \beta) \land (\beta \implies \alpha)) \quad \text{biconditional elimination} \\
\neg(\alpha \land \beta) & \equiv (\neg \alpha \lor \neg \beta) \quad \text{de Morgan} \\
\neg(\alpha \lor \beta) & \equiv (\neg \alpha \land \neg \beta) \quad \text{de Morgan} \\
(\alpha \land (\beta \lor \gamma)) & \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \quad \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) & \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \quad \text{distributivity of } \lor \text{ over } \land
\end{align*}
\]

(*) key to go to clausal (Conjunctive Normal Form)
Implication for “humans”; clauses for machines.
de Morgan laws also very useful in going to clausal form.
Style II: Proof by inference rules

Modus Ponens (MP)

KB at $T = 1$:
- R1: $\neg P_{1,1}$
- R2: $\neg B_{1,1}$
- R3: $B_{2,1}$

R4: $B_{1,1} \iff (P_{1,2} \lor P_{2,1})$
R5: $B_{2,1} \iff (P_{1,1} \lor P_{2,2} \lor P_{3,1})$

How can we show that $K \models \neg P_{1,2}$?

Wumpus world at $T = 1$

Note: In formal proof, every step needs to be justified.

So, we used R2 and R4.
Length of Proofs

Why bother with inference rules? We could always use a truth table to check the validity of a conclusion from a set of premises.

But, resulting proof can be much shorter than truth table method.

Consider KB:
\[ p_1, \quad p_1 \rightarrow p_2, \quad p_2 \rightarrow p_3, \quad \ldots, \quad p_{(n-1)} \rightarrow p_n \]

To prove conclusion: \( p_n \)

Inference rules: \( n-1 \) MP steps
Truth table: \( 2^n \)

Key open question: Is there always a short proof for any valid conclusion? Probably not. The NP vs. co-NP question.
(The closely related: P vs. NP question carries a $1M prize.)
First, we need a conversion to Conjunctive Normal Form (CNF) or Clausal Form.

Let’s consider converting R4 in clausal form:

R4: \( B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \)

We have:

\( B_{1,1} \land (P_{1,2} \land P_{2,1}) \)

which gives (implication elimination):

\( (\iff B_{1,1} \land P_{1,2} \land P_{2,1}) \)

Also

\( (P_{1,2} \lor P_{2,1}) \land B_{1,1} \)

which gives:

\( (\iff (P_{1,2} \land P_{2,1}) \land B_{1,1}) \)

Thus,

\( (\iff P_{1,2} \iff P_{2,1}) \land B_{1,1} \)

leaving,

\( (\iff P_{1,2} \iff B_{1,1}) \)

\( (\iff P_{2,1} \iff B_{1,1}) \)

(note: clauses in red)
First, we need a conversion to Conjunctive Normal Form (CNF) or Clausal Form.

Let's consider converting R4 in clausal form:

\[ R4: \quad B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \]

We have:

\[ B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1}) \]

which gives (implication elimination):

\[ (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \]

Also

\[ (P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1} \]

which gives:

\[ (\neg (P_{1,2} \lor P_{2,1}) \lor B_{1,1}) \]

Thus,

\[ (\neg P_{1,2} \land \neg P_{2,1}) \lor B_{1,1} \]

leaving,

\[ (\neg P_{1,2} \lor B_{1,1}) \]

\[ (\neg P_{2,1} \lor B_{1,1}) \]

(note: clauses in red)

### Style III: Resolution

**Wumpus world at T = 1**

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A/B</td>
<td></td>
<td>P</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>O K</td>
</tr>
</tbody>
</table>

P?
KB at $T = 1$:

R1: $\neg P_{1,1}$
R2: $\neg B_{1,1}$
R3: $B_{2,1}$

R4: $B_{1,1} \iff (P_{1,2} \lor P_{2,1})$
R5: $B_{2,1} \iff (P_{1,1} \lor P_{2,2} \lor P_{3,1})$

KB at $T = 1$ in clausal form:

R1: $\neg P_{1,1}$
R2: $\neg B_{1,1}$
R3: $B_{2,1}$

R4a: $\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}$
R4b: $\neg P_{1,2} \lor B_{1,1}$
R4c: $\neg P_{2,1} \lor B_{1,1}$

R5a: $\neg B_{2,1} \lor P_{1,1} \lor P_{2,2} \lor P_{3,1}$
R5b: $\neg P_{1,1} \lor B_{2,1}$
R5c: $\neg P_{2,2} \lor B_{2,1}$
R5d: $\neg P_{3,1} \lor B_{2,1}$

Wumpus world at $T = 1$
How can we show that $KR \models \neg P_{1,2}$?

Proof by contradiction:
Need to show that $(KB \land P_{1,2})$ is inconsistent (unsatisfiable).

Resolution rule:

$$(\alpha \lor p) \text{ and } (\beta \lor \neg p)$$

gives resolvent (logically valid conclusion):

$$(\alpha \lor \beta)$$

If we can reach the empty clause, then $KB$ is inconsistent. (And, vice versa.)
KB at T=1 in clausal form:

R1: \( \neg P_{1,1} \)
R2: \( \neg B_{1,1} \)
R3: \( B_{2,1} \)

R4a: \( \neg B_{1,1} \lor P_{1,2} \lor P_{2,1} \)
R4b: \( \neg P_{1,2} \lor B_{1,1} \)
R4c: \( \neg P_{2,1} \lor B_{1,1} \)

R5a: \( \neg B_{2,1} \lor P_{1,1} \lor P_{2,2} \lor P_{3,1} \)
R5b: \( \neg P_{1,1} \lor B_{2,1} \)
R5c: \( \neg P_{2,2} \lor B_{2,1} \)
R5d: \( \neg P_{3,1} \lor B_{2,1} \)

Show that \( (KB \land P_{1,2}) \) is inconsistent. (unsatisfiable)

R4b with \( P_{1,2} \) resolves to \( B_{1,1} \),
which with R2, resolves to the empty clause, \( \Box \).
So, we can conclude \( KB \models \neg P_{1,2} \).
(make sure you use “what you want to prove.”)
KB at T=1 in clausal form:

R1: \( \neg P_{1,1} \)
R2: \( \neg B_{1,1} \)
R3: \( B_{2,1} \)
R4a: \( \neg B_{1,1} \lor P_{1,2} \lor P_{2,1} \)
R4b: \( \neg P_{1,2} \lor B_{1,1} \)
R4c: \( \neg P_{2,1} \lor B_{1,1} \)
R5a: \( \neg B_{2,1} \lor P_{1,1} \lor P_{2,2} \lor P_{3,1} \)
R5b: \( \neg P_{1,1} \lor B_{2,1} \)
R5c: \( \neg P_{2,2} \lor B_{2,1} \)
R5d: \( \neg P_{3,1} \lor B_{2,1} \)

Another example resolution proof

Note that R5a resolved with R1, and then resolved with R3, gives \((P_{2,2} \lor P_{3,1})\).

Almost there… to show KB \( \vdash (P_{2,2} \lor P_{3,1}) \), we need to show KB \( \land (\neg (P_{2,2} \lor P_{3,1})) \) is inconsistent. (Why? Semantically?) So, show KB \( \land \neg P_{2,2} \land \neg P_{3,1} \) is inconsistent. This follows from \((P_{2,2} \lor P_{3,1})\); because in two more resolution steps, we get the empty clause (a contradiction).
Consider KB:
\[ p_1, \ p_1 \rightarrow p_2, \ p_2 \rightarrow p_3, \ldots, \ p_{(n-1)} \rightarrow p_n \]

To prove conclusion: \( p_n \)

Resolution. Assert \( \neg p_n \)
with \( \neg p_{(n-1)} \lor p_n \) gives \( \neg p_{(n-1)} \)
with \( \neg p_{(n-2)} \lor p_{(n-1)} \) gives \( \neg p_{(n-2)} \)
...  
with \( \neg p_1 \lor p_2 \) gives \( \neg p_1 \)
with \( p_1 \) gives empty clause (contradiction).
QED
Note how resolution mimics Modus Ponens steps.

Inference rules: \( n \) resolution steps \quad \text{Truth table: } 2^n

So, efficient on these proofs!
What is hard for resolution?

Consider:

Given a fixed pos. int. N

\[(P(i,1) \lor P(i,2) \lor \ldots \lor P(i,N+1)) \land (\neg P(i,j) \lor \neg P(i,j) \lor \ldots \lor \neg P(i,j))\]

What does this encode?

Think of: P(i,j) for “object i in location j”

Pigeon hole problem...

Provable requires exponential number of resolution steps to reach empty clause (Haken 1985). Method “can’t count.”
Instead of using resolution to show that

\[ KB \land \neg \alpha \] is inconsistent,

modern Satisfiability (SAT) solvers operating on the clausal form are *much* more efficient.

The SAT solvers treat constraints (disjunctions) on Boolean formulas. This is a special form of resolution! Current solvers are very efficient on million+ variables and several millions of clauses.

Systematic: Davis Putnam (DPLL) + *series of improvements*

Stochastic local search: WalkSAT (issue?)

See R&N 7.6. “Ironically,” we are back to semantic model checking, but way more clever than basic truth assignment enumeration (exponentially faster)!
DPLL improvements

Backtracking + …

1) Component analysis (disjoint sets of constraints? Problem decomposition?)
2) Clever variable and value ordering (e.g. degree heuristics)
3) Intelligent backtracking and clause learning (conflict learning)
4) Random restarts (heavy tails in search spaces…)
5) Clever data structures

1+ Million Boolean vars & 10+ Million clause/constraints are feasible nowadays. (e.g. Minisat solver)

Has changed the world of verification (hardware/software) over the last decade (incl. Turing award for Clarke). Widely used in industry, Intel, Microsoft, IBM etc.
1) \( \text{KB} \models \alpha \)