CS-E4800 Artificial Intelligence

Jussi Rintanen

Department of Computer Science
Aalto University

January 26, 2017

The 1st Home Assignment

- Assignment available on the web page since Friday
- Return solution to Jussi.Rintanen@aalto.fi by **23:59 Thursday February 2**
- Produce an archive file that contains
  - all source code,
  - document describing experimentation with your code
- File name is your student ID, e.g. **35093A.tar**
- Feel free to ask questions and help, and do it well before the deadline.

Logic vs. Thinking and Intelligence

Intelligence and Thinking = Logic and Inference

(View held for centuries until it fell out of favor: Philosophy 1930ies..., AI 1980ies...)

Aristoteles (384-322 BC)

Collection of valid forms of inference, including...

$$\begin{array}{l}\text{every A is B}\\ \text{every B is C}\\ \hline \text{every A is C}\end{array} \qquad \begin{array}{l}\text{some A is B}\\ \text{every B is C}\\ \hline \text{some A is C}\end{array}$$

$$\begin{array}{l}\text{every A is B}\\ \text{every A is C}\\ \hline \text{some B is C}\end{array} \qquad \begin{array}{l}\text{no A is B}\\ \text{some A is C}\\ \hline \text{some C is not B}\end{array}$$

George Boole

The Mathematical Analysis of Logic, 1847
An Investigation of the Laws of Thought, 1854

notation	
Boole	modern
xy	$x \wedge y$
$x + y - xy$	$x \vee y$
$1 - x$	$\neg x$

Logic: Modern View

Thinking $\neq$ Logic, Inference

- Few "absolute facts" to base inference on
- The purpose and result of thinking is not true facts

The above is for general human-like intelligence.
Logic useful for AI in limited domains (as most applications are!)

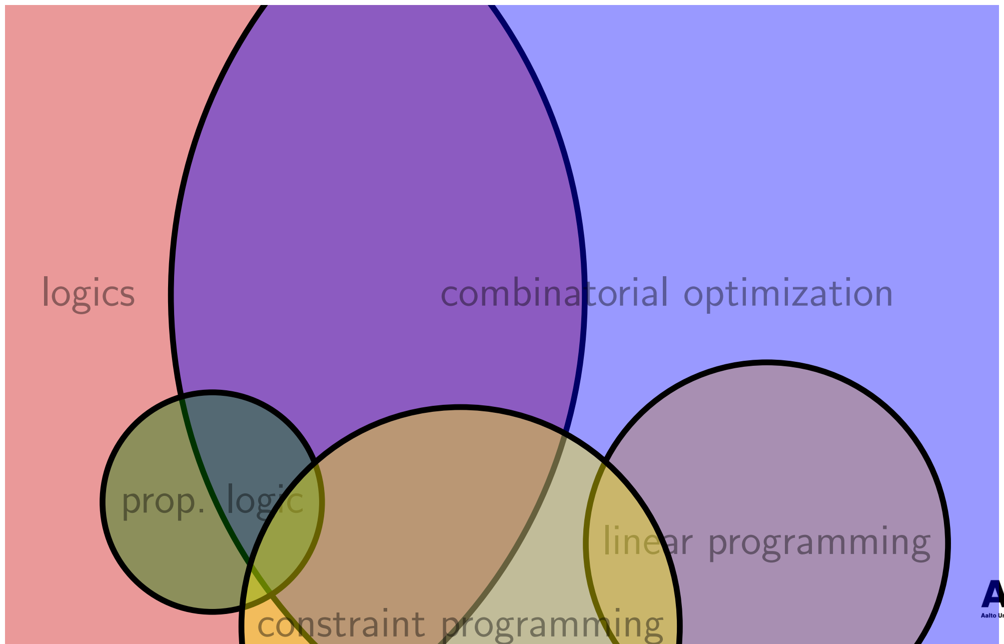
Logic and AI: Modern View

- Definition: "Chair is something the *purpose* of which is to be sit on (by one person)"
- What about a broken chair you cannot sit on?
- You could use a non-chair object as a chair? Is it a chair then?
- Chairs: objects similar to usual chairs or otherwise suitable for sitting ("family resemblance")

Conclusion:

- There is no precise definition of chair
- Why would you even care about a definition of chair?

Logic vs. Combinatorial Optimization



Logic, Constraint Programming, Combinatorial Optimization

- Constraint Programming and Combinatorial Optimization
  - logical constraints
  - arithmetic constraints (integer, real, rational)
  - relational constraints
- Propositional logic (satisfiability problem SAT)
- more powerful logics

Examples:

- Mixed Integer-Linear Programming: integer+real
- Boolean satisfiability (SAT): Boolean
- SMT (SAT modulo Theories): Boolean+integer+real+bitvectors+others

Propositional logic

- atomic propositions $p_1, p_2, \ldots$
- connectives $\vee$, $\wedge$, $\neg$, $\rightarrow$, $\ldots$
- applications: finite state systems
- advantage: *only* NP-complete

Predicate logic

- logic of relational structures
- atomic propositions $P(t_1, \ldots, t_n)$
- same connectives as in the propositional logic
- applications: infinite systems
- disadvantage: computationally highly intractable

Temporal logics

- logic of time
- operators:
  - $\Box$ always (future, past),
  - $\Diamond$ sometimes,
  - $\circ$ next,
  - $[t_1, t_2]$ in interval from t_1 to t_2
- different timelines: linear, branching
- applications: systems with progression of time (AI, CAV)

Temporal logic are modal logics: temporal operators not truth-functional

Dynamic logics

- logic of actions and programs
- operators $[\pi]\phi$ "ϕ holds after every execution of π" for actions/programs π of forms
  - a (atomic program)
  - $\pi_1; \pi_2$ (sequence of two programs)
  - $\pi_1 \cup \pi_2$ (nondeterministic choice between programs)
  - $\pi_1 *$ (iteration of program arbitrarily many times)
  - ϕ? (test whether ϕ holds)
- applications: software, intelligent agents (AI, CAV)

Epistemic logics

- logics of belief and knowledge
- operators
  - $\Box\phi$ (agent believes/knows that ϕ)
- applications: computer security

Conditional logics

- logics of counterfactuals
  "If Sanders had been the Democratic candidate, Trump would not be president"
- connective
  - $\phi > \psi$ (if ϕ were true, ψ would be true)
- applications: philosophy and academic work on AI
- disadvantage: inferences far too weak

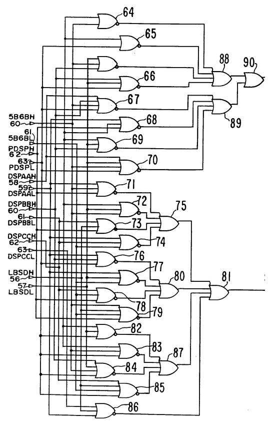
(Hundreds of further logics exist, addressing all kinds of (also marginal) features of natural language.)

Logics in Practice (IT, CS, AI)

- "Small" applications of logic everywhere: DB, IC, programming, ...
- Propositional logic most used when automated reasoning is needed:
  - low complexity
  - best algorithms often work OK with tens of thousands of atomic propositions (formula sizes 100 MB, 1 GB)
- Demand for more expressive logics, but they are too expensive
- Temporal logics used as parts of specification languages
- Other logics in limited applications, formulas of small sizes, or not with fully automated inference

Applications of Logic in Computer Science

Correctness of integrated circuits, protocols, software



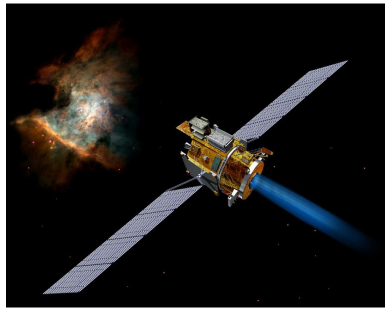
T. Larrabee: "Test pattern generation using Boolean satisfiability", 1992

Biere et al.: "Symbolic model checking without BDDs", 1999

(approach of Biere et al. based on Kautz & Selman: "Planning as Satisfiability", European Conference on Artificial Intelligence, 1992; Kautz & Selman, AAAI 1996)

Applications of Logic in Computer Science

Control of autonomous intelligent systems

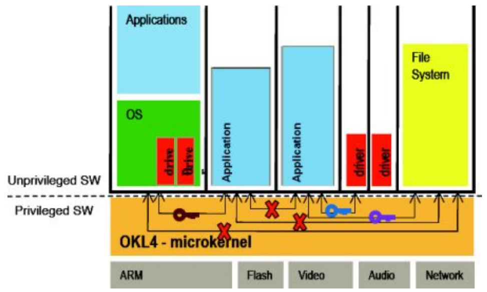


Williams and Nayak: "A model-based approach to reactive self-configuring systems", 1996

Muscettola et al.: "Remote agent: to boldly go where no AI system has gone before", 1998

Applications of Logic in Computer Science

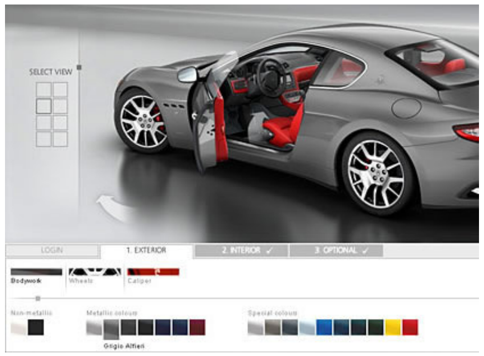
Software correctness proofs



seL4/OKL4-microkernel in over one billion cell phones

Klein et al.: "seL4: formal verification of an OS kernel", 2009

Applications of Logic in Computer Science



Product configuration; a combination of components is chosen based on requirements and information about dependencies and incompatibilities

T. Soininen and I. Niemelä: "Developing a Declarative Rule Language for Applications in Product Configuration", 1998.

Logic in Artificial Intelligence

- models of reasoning and inference (humans, intelligent systems)
- representation of incomplete data and knowledge, reasoning about it
- representation of large system models
- large-scale data structures

Inference and Reasoning

Bob is a professor
Professors work at a university

Bob works at a university

Jack is looking at Anne. Anne is looking at George.
Jack is married, but George is not.

Is a married person looking at an unmarried person?
A) Yes
B) No
C) Cannot be determined

Incomplete Data

Incomplete Data in conventional data structures

We could express unknown with a special value ?.

A	B	C	D	E	F
0	1	1	?	?	?

But how to represent that at least one of D and E is 1?

Incomplete Data in logic

Any form of incomplete data can be represented:
Example: $\neg A$, B, C, $D \vee E$

State-Space Search, Analysis of Transition Systems

A killer application of propositional logic in CS

- Possible behaviors (of components) known exactly, must infer actual behavior (of whole system)
- Planning: action sequence initial to goal state
- Verification: event sequence initial to fault state
- Diagnosis: event sequence satisfying observations
- Diagnosability: Can always detect failures?

(This will be covered next week!)

Cardinality Constraints ≥ 1, ≤ 1

At Least One of $x_1, x_2, \ldots, x_n$

The disjunction $x_1 \vee x_2 \vee \cdots \vee x_n$

At Most One of $x_1, x_2, \ldots, x_n$

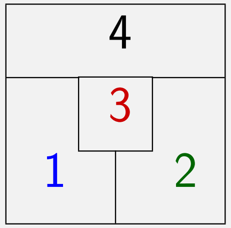
"At Most One Of" $=$ "No Two Of"

$\neg(x_1 \wedge x_2), \neg(x_1 \wedge x_3), \ldots, \neg(x_1 \wedge x_n)$
$\neg(x_2 \wedge x_3), \neg(x_2 \wedge x_4), \ldots, \neg(x_2 \wedge x_n)$
$\vdots$
$\neg(x_{n-2} \wedge x_{n-1}), \neg(x_{n-2} \wedge x_n)$
$\neg(x_{n-1} \wedge x_n)$

These together (conjunction) give "Exactly One Of".

Map Coloring

Color map with (at most) three colors



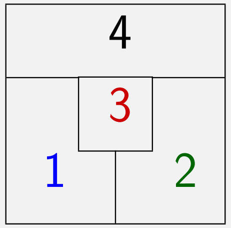
Every region i has exactly one color:
$R_i \vee G_i \vee B_i \quad \neg(R_i \wedge G_i) \quad \neg(G_i \wedge B_i)$
$\neg(B_i \wedge R_i)$

Neighboring regions i and j have different colors:
$\neg(R_i \wedge R_j) \quad \neg(G_i \wedge G_j) \quad \neg(B_i \wedge B_j)$

Unsatisfiable $\Longrightarrow$ **3-coloring doesn't exist!**

Map Coloring

Color map with (at most) four colors

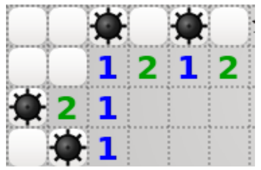


Every region i has exactly one color:
$R_i \vee G_i \vee B_i \vee W_i \quad \neg(R_i \wedge G_i)$
$\neg(G_i \wedge B_i) \quad \neg(B_i \wedge R_i)$
$\neg(W_i \wedge R_i) \quad \neg(W_i \wedge G_i) \quad \neg(W_i \wedge B_i)$

Neighboring regions i and j have different colors:
$\neg(R_i \wedge R_j)\ \neg(G_i \wedge G_j)\ \neg(B_i \wedge B_j)\ \neg(W_i \wedge W_j)$

Satisfiable $\Longrightarrow$ **4-coloring exists!**

Minesweeper



	$m_{0,1}$	$m_{0,2}$	$m_{0,3}$	$m_{0,4}$
$m_{1,1}$		1	2	

cell	at least, at most
1	$m_{1,1} \vee m_{0,1} \vee m_{0,2} \vee m_{0,3}$, $\neg(m_{1,1} \wedge m_{0,1}), \neg(m_{1,1} \wedge m_{0,2}), \neg(m_{1,1} \wedge m_{0,3})$, $\neg(m_{0,1} \wedge m_{0,2}), \neg(m_{0,1} \wedge m_{0,3}), \neg(m_{0,2} \wedge m_{0,3})$
2	$(m_{0,2} \wedge m_{0,3}) \vee (m_{0,2} \wedge m_{0,4}) \vee (m_{0,3} \wedge m_{0,4})$, $\neg(m_{0,2} \wedge m_{0,3} \wedge m_{0,4})$

Logical consequences: $\neg m_{1,1} \quad \neg m_{0,1} \quad m_{0,4} \quad m_{0,2} \vee m_{0,3}$

(Almost) Perfect Player of Minesweeper

1. Construct formula Φ for all cells with a number, as shown on the previous slide.
2. For each unknown cell (x, y), test if $\Phi \models \neg m_{x,y}$.
3. If yes, play (x, y).
4. If no such cell exists, pick a cell with number n and m unknown neighbors so that $\frac{n}{m}$ is the smallest possible, and play it.
   (Question: Is this the best that can be done?)
5. Repeat from step 1.

As long as step 4 can be avoided, the player never fails.

Other Logic Puzzles

- Many other logic puzzles have simple solutions based on satisfiability or logical consequence.
- Sudoku solvable with one satisfiability test, if solution unique.

The Zebra Puzzle

```
There are five houses.
The Englishman lives in the red house.
The Spaniard owns the dog.
Coffee is drunk in the green house.
The Ukrainian drinks tea.
The green house is immediately to the right of the ivory house.
The Old Gold smoker owns snails.
Kools are smoked in the yellow house.
Milk is drunk in the middle house.
The Norwegian lives in the first house.
The man who smokes Chesterfields lives in the house next to
 the man with the fox.
Kools are smoked in the house next to the house where the horse
 is kept.
The Lucky Strike smoker drinks orange juice.
The Japanese smokes Parliaments.
The Norwegian lives next to the blue house.
```

The Zebra Puzzle: Implicit Assumptions

Not all assumptions of the puzzle are explicit:

1. Each gentleman has exactly one drink, pet, house and cigarette.
2. Each drink, pet, house, and cigarette belongs to exactly one gentleman.
3. The houses are in a row (1st, 2nd, 3rd (middle), 4th, 5th)

All assumptions have to be made explicit!

The Zebra Puzzle

The atomic propositions needed in formalizing the puzzle (e.g. $(\mathit{LivesIn}(\mathit{Eng}, 2))$ are constructed with the help of the following object names.

$H = \{1, 2, 3, 4, 5\}$
$M = \{\text{Eng}, \text{Spa}, \text{Ukr}, \text{Nor}, \text{Jap}\}$
$C = \{\text{OldGold}, \text{Kools}, \text{Chesterfields}, \text{LuckyStrike}, \text{Parliaments}\}$
$P = \{\text{Red}, \text{Green}, \text{Ivory}, \text{Yellow}, \text{Blue}\}$
$A = \{\text{Dog}, \text{Snails}, \text{Fox}, \text{Horse}, \text{Zebra}\}$
$D = \{\text{Coffee}, \text{Tea}, \text{Milk}, \text{OrangeJuice}, \text{Water}\}$

We use the same kind of schematic representation we saw in connection with NDL.

The Zebra Puzzle

The Englishman lives in the red house:
$\bigvee_{i \in H}(\mathit{LivesIn}(\mathit{Eng}, i) \wedge \mathit{Color}(i, \mathit{Red}))$

The Spaniard owns the dog: $\mathit{Pet}(\mathit{Spa}, \mathit{Dog})$

Coffee is drunk in the green house:
$\bigvee_{i \in M} \bigvee_{j \in H}(\mathit{LivesIn}(i, j) \wedge \mathit{Drinks}(i, \mathit{Coffee}) \wedge \mathit{Color}(j, \mathit{Green}))$

The Ukrainian drinks tea: $\mathit{Drinks}(\mathit{Ukr}, \mathit{Tea})$

The green house is to the right of the ivory house:
$\bigvee_{i \in H} \bigvee_{j \in H}(\mathit{Color}(i, \mathit{Ivory}) \wedge \mathit{Color}(j, \mathit{Green}) \wedge (j = i+1))$

The Old Gold smoker owns snails:
$\bigvee_{i \in M}(\mathit{Smokes}(i, \mathit{OldGold}) \wedge \mathit{Pet}(i, \mathit{Snails}))$

The Zebra Puzzle

The Englishman lives in the red house:
$\bigvee_{i \in H}(\mathit{LivesIn}(\mathit{Eng}, i) \wedge \mathit{Color}(i, \mathit{Red}))$

Written in propositional logic:

$$\begin{aligned}
&(\mathit{LivesIn}(\mathit{Eng}, 1) \wedge \mathit{Color}(1, \mathit{Red}))\\
&\vee(\mathit{LivesIn}(\mathit{Eng}, 2) \wedge \mathit{Color}(2, \mathit{Red}))\\
&\vee(\mathit{LivesIn}(\mathit{Eng}, 3) \wedge \mathit{Color}(3, \mathit{Red}))\\
&\vee(\mathit{LivesIn}(\mathit{Eng}, 4) \wedge \mathit{Color}(4, \mathit{Red}))\\
&\vee(\mathit{LivesIn}(\mathit{Eng}, 5) \wedge \mathit{Color}(5, \mathit{Red}))
\end{aligned}$$

The Zebra Puzzle

The green house is to the right of the ivory house:
$\bigvee_{i \in H} \bigvee_{j \in H}(\mathit{Color}(i, \mathit{Ivory}) \wedge \mathit{Color}(j, \mathit{Green}) \wedge (j = i+1))$

Written in propositional logic:

$$\begin{aligned}
&(\mathit{Color}(1, \mathit{Ivory}) \wedge \mathit{Color}(2, \mathit{Green}))\\
&\vee(\mathit{Color}(2, \mathit{Ivory}) \wedge \mathit{Color}(3, \mathit{Green}))\\
&\vee(\mathit{Color}(3, \mathit{Ivory}) \wedge \mathit{Color}(4, \mathit{Green}))\\
&\vee(\mathit{Color}(4, \mathit{Ivory}) \wedge \mathit{Color}(5, \mathit{Green}))
\end{aligned}$$

The Zebra Puzzle

Kools are smoked in the yellow house:
$\bigvee_{i\in M}\bigvee_{j\in H}(\mathit{LivesIn}(i,j)\wedge \mathit{Color}(j,\mathit{Yellow})\wedge \mathit{Smokes}(i,\mathit{Kools}))$

Milk is drunk in the middle house:
$\bigvee_{i\in M}(\mathit{LivesIn}(i,3)\wedge \mathit{Drinks}(i,\mathit{Milk}))$

The Norwegian lives in the first house: $\mathit{LivesIn}(\mathit{Nor},1)$

The man who smokes Chesterfields lives in the house next to the man with the fox:
$\bigvee_{i\in M}\bigvee_{j\in M}\bigvee_{k\in H}\bigvee_{l\in H}(\mathit{Smokes}(i,\mathit{Chesterfields})\wedge \mathit{Pet}(j,\mathit{Fox})\wedge \mathit{LivesIn}(i,k)\wedge \mathit{LivesIn}(j,l)\wedge(|j-l|=1))$

The Zebra Puzzle

Kools are smoked in the house next to the house where the horse is kept:
$\bigvee_{i\in M}\bigvee_{j\in M}\bigvee_{k\in H}\bigvee_{l\in H}(\mathit{Smokes}(i,\mathit{Kools})\wedge \mathit{Pet}(j,\mathit{Horse})\wedge \mathit{LivesIn}(i,k)\wedge \mathit{LivesIn}(j,l)\wedge(|k-l|=1))$

The Lucky Strike smoker drinks orange juice:
$\bigvee_{i\in M}(\mathit{Smokes}(i,\mathit{LuckyStrike})\wedge \mathit{Drinks}(i,\mathit{OrangeJuice}))$

The Japanese smokes Parliaments:
$\mathit{Smokes}(\mathit{Jap},\mathit{Parliaments})$

The Norwegian lives next to the blue house:
$\bigvee_{i\in H}\bigvee_{j\in H}(\mathit{LivesIn}(\mathit{Nor},i)\wedge(|i-j|=1)\wedge \mathit{Color}(j,\mathit{Blue}))$

The Zebra Puzzle

Define ExactlyOneOf$(x_1,\ldots,x_n)=$
$(\bigvee_{i=1}^{n}x_i)\wedge\bigwedge_{1\le i<j\le n}\neg(x_i\wedge x_j)$

The background assumptions:
ExactlyOneOf$(\{\mathit{LivesIn}(m,h)|h\in H\})$ for each $m\in M$
ExactlyOneOf$(\{\mathit{LivesIn}(m,h)|m\in M\})$ for each $h\in H$
ExactlyOneOf$(\{\mathit{Smokes}(m,c)|c\in C\})$ for each $m\in M$
ExactlyOneOf$(\{\mathit{Smokes}(m,c)|m\in M\})$ for each $c\in C$
ExactlyOneOf$(\{\mathit{Pet}(m,a)|a\in A\})$ for each $m\in M$
ExactlyOneOf$(\{\mathit{Pet}(m,a)|m\in M\})$ for each $a\in A$
ExactlyOneOf$(\{\mathit{Color}(h,p)|p\in P\})$ for each $h\in H$
ExactlyOneOf$(\{\mathit{Color}(h,p)|h\in H\})$ for each $p\in P$
ExactlyOneOf$(\{\mathit{Drinks}(m,d)|m\in M\})$ for each $d\in D$
ExactlyOneOf$(\{\mathit{Drinks}(m,d)|d\in D\})$ for each $m\in M$

Quantifiers $\forall$ and $\exists$

Universal Quantification $\forall$ (For All)

$\bigwedge_{i\in H}\phi(i)$ is sometimes written as $\forall i\in H.\phi(i)$.
$\forall i\in H.\phi(i)$ can be reduced to conjunctions whenever H is finite.

Existential Quantification $\exists$ (For At Least One)

$\bigvee_{i\in H}\phi(i)$ is sometimes written as $\exists i\in H.\phi(i)$.
$\exists i\in H.\phi(i)$ can be reduced to disjunctions whenever H is finite.

The propositional logic does not include quantification. First-order predicate logic includes quantification, also over infinite domains. Not included in this course!

Combinatorial Explosion

- The combinatorial explosion shows up in state-space search when the states are described succinctly in terms of state variables
- State space size $\mathcal{O}(2^n)$ for n state variables
- Conventional graph algorithms are not applicable to very large graphs with $> 10^8, 10^9$ or 10^{10} states
- Logic representations one approach to avoid these limitations (sometimes)

Combinatorial Explosion

Two approaches to state-space search with logic:

1. Reachability by a sequence of actions as a satisfiability test
2. Formulas as data structure for sets and relations

Closely related, same formulas used in both cases.

Representation of Sets of Bitvectors

Valuations as bit-vectors

a valuation of n variables $\sim$ n-bit bit-vector

Example

Let $X = \{A, B, C, D\}$. A valuation that assigns $A = 1, B = 0, C = 1, D = 1$ corresponds to the bit-vector $\overset{ABCD}{1010}$, and the valuation that assigns 1 only to B corresponds to the bit-vector $\overset{ABCD}{0100}$.

Atomic Formulas as Sets

Propositional formulas as a set data structure

formula $\sim$ set of bit-vectors

Example

Formula B represents all bit-vectors of the form ?1??, and the formula A represents all bit-vectors 1???. Formula B therefore represents the set

$$\{0\mathbf{1}00, 0\mathbf{1}01, 0\mathbf{1}10, 0\mathbf{1}11, 1\mathbf{1}00, 1\mathbf{1}01, 1\mathbf{1}10, 1\mathbf{1}11\}$$

and formula A represents the set

$$\{\mathbf{1}000, \mathbf{1}001, \mathbf{1}010, \mathbf{1}011, \mathbf{1}100, \mathbf{1}101, \mathbf{1}110, \mathbf{1}111\}.$$

Negated Atomic Formulas as Sets

Similarly, $\neg B$ represents all bit-vectors of the form ?0??, which is the set

$$\{0\mathbf{0}00, 0\mathbf{0}01, 0\mathbf{0}10, 0\mathbf{0}11, 1\mathbf{0}00, 1\mathbf{0}01, 1\mathbf{0}10, 1\mathbf{0}11\}.$$

This is the complement of the set represented by B!

Union, Intersection, Complement

Connectives as set-operations

If ϕ_1 and ϕ_2 represent bit-vector sets S_1 and S_2, then

1. $\phi_1 \wedge \phi_2$ represents set $S_1 \cap S_2$,
2. $\phi_1 \vee \phi_2$ represents set $S_1 \cup S_2$, and
3. $\neg\phi_1$ represents set $\overline{S_1}$.

Example

$A \wedge B$ represents the set $\{1100, 1101, 1110, 1111\}$ and $A \vee B$ represents the set $\{0100,0101,0110,0111,1000,1001,1010,1011,1100,1101,1110,1111\}$.

Logical Notions as Set-Theoretical Notions

1. "Is ϕ satisfiable?" corresponds to "Is the set represented by ϕ non-empty?"
2. $\phi \models \alpha$ corresponds to "Is the set represented by ϕ a subset of the set represented by α?".
3. "Is ϕ valid?" corresponds to "Is the set represented by ϕ the universal set?"

Logic vs. Set Theory: Summary

state sets	**formulas over X**
those $\frac{2^{\lvert X\rvert}}{2}$ bitvectors where x is true	$x \in X$
$\overline{E}$ (complement)	$\neg E$
$E \cup F$	$E \vee F$
$E \cap F$	$E \wedge F$
$E \backslash F$ (set difference)	$E \wedge \neg F$
the empty set $\emptyset$	$\bot$ (constant *false*)
the universal set	$\top$ (constant *true*)
question about sets	**question about formulas**
$E \subseteq F$?	$E \models F$?
$E \subset F$?	$E \models F$ and $F \not\models E$?
$E = F$?	$E \models F$ and $F \models E$?