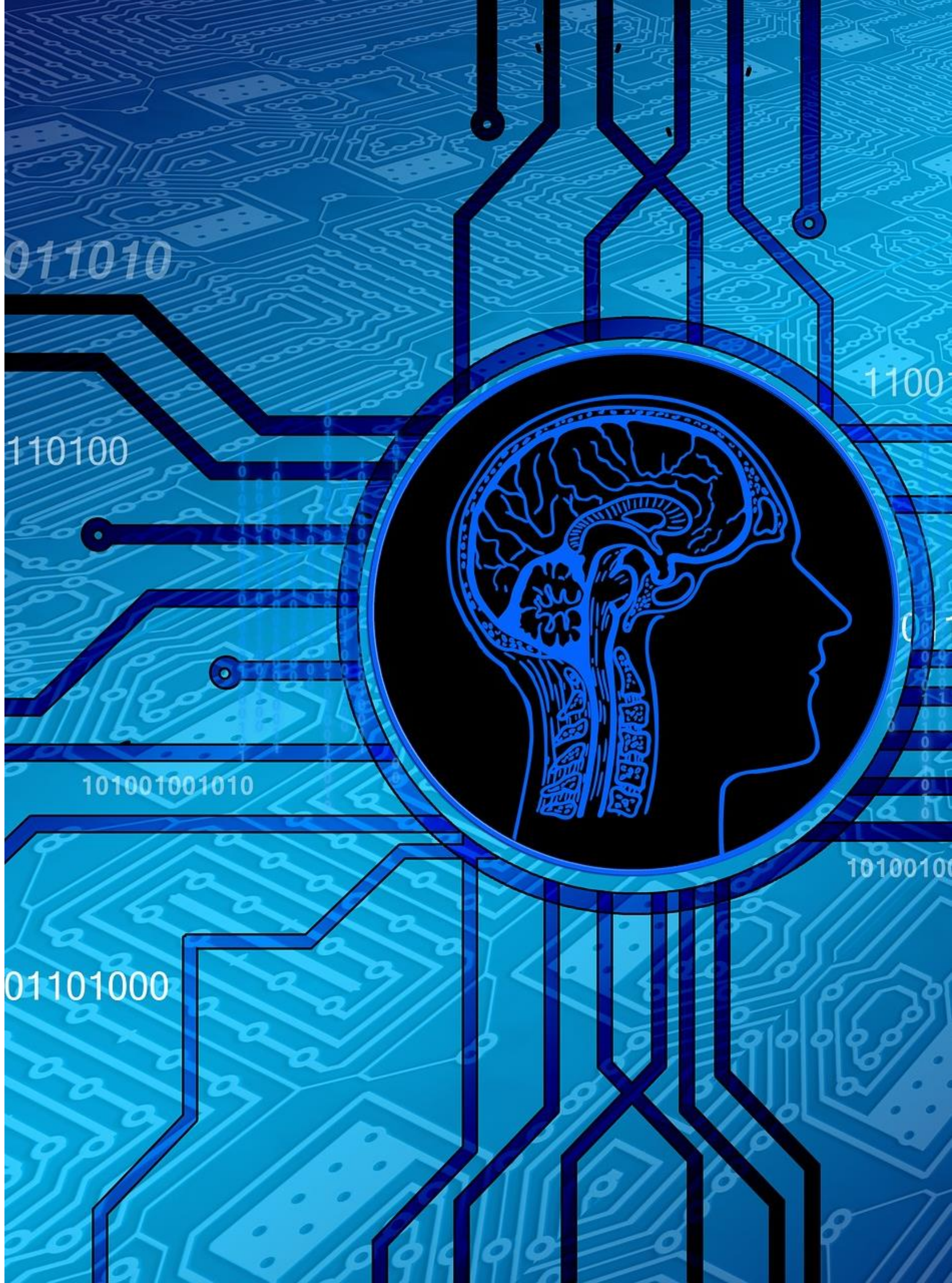
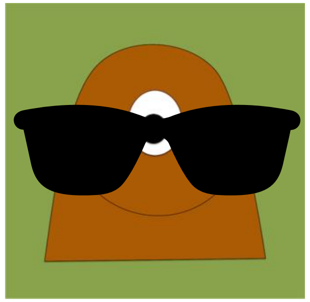
Revision (Logic)

Informatics 2D: Reasoning and Agents



Logical Agents: KBs



- Logical agents apply inference to a knowledge base to derive new information and make decisions.
- Basic concepts of logic:
  - syntax: formal structure of sentences
  - semantics: truth of sentences with respect to models
  - entailment: necessary truth of one sentence given another
  - inference: deriving sentences from other sentences
  - soundness: derivations produce only entailed sentences
  - completeness: derivations can produce all entailed sentences
- Wumpus world requires the ability to represent partial and negated information
- Propositional logic solves many problems but lacks expressive power.

Logical equivalence

$(\alpha \land \beta) \equiv (\beta \land \alpha)$	commutativity of $\land$
$(\alpha \lor \beta) \equiv (\beta \lor \alpha)$	commutativity of $\lor$
$((\alpha \land \beta) \land \gamma) \equiv (\alpha \land (\beta \land \gamma))$	associativity of $\land$
$((\alpha \lor \beta) \lor \gamma) \equiv (\alpha \lor (\beta \lor \gamma))$	associativity of $\lor$
$\neg(\neg\alpha) \equiv \alpha$	double-negation elimination
$(\alpha \rightarrow \beta) \equiv (\neg\beta \rightarrow \neg\alpha)$	contraposition
$(\alpha \rightarrow \beta) \equiv (\neg\alpha \lor \beta)$	implication elimination
$(\alpha \leftrightarrow \beta) \equiv ((\alpha \rightarrow \beta) \land (\beta \rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \land \beta) \equiv (\neg\alpha \lor \neg\beta)$	de Morgan
$\neg(\alpha \lor \beta) \equiv (\neg\alpha \land \neg\beta)$	de Morgan
$(\alpha \land (\beta \lor \gamma)) \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma))$	distributivity of $\land$ over $\lor$
$(\alpha \lor (\beta \land \gamma)) \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma))$	distributivity of $\lor$ over $\land$

Two sentences are logically equivalent iff true in the same models:

$$\alpha \equiv \beta \text{ iff } \alpha \models \beta \text{ and } \beta \models \alpha$$

Validity and Satisfiability

A sentence is valid if it is true in *all models*

- *true*, $A \lor \neg A$, $A \Rightarrow A$, $(A \land (A \Rightarrow B)) \Rightarrow B$

Validity is connected to inference via the Deduction Theorem

- $KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

A sentence is satisfiable if it is true in *some model*

- e.g., $A \lor B$, C

A sentence is unsatisfiable if it is true in *no models*

- e.g., $A \land \neg A$

Satisfiability is connected to inference via the following:

- $KB \models \alpha$ if and only if $(KB \land \neg\alpha)$ is unsatisfiable
- prove α by *reductio ad absurdum*

Effective Propositional Inference

- Two algorithms: DPLL & WalkSAT
  - DPLL is a decision procedure, i.e., it will return true (yes) or false (no) for a set of propositional clauses (cf. complete algorithm)
  - They work with Conjunctive Normal Form (CNF)

The DPLL algorithm

Determine if an input propositional logic sentence (in CNF) is satisfiable.

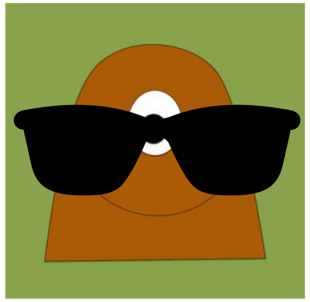
Improvements over truth table enumeration:

1. Early termination
2. Pure symbol heuristic
3. Unit clause heuristic

The WalkSAT algorithm

- Incomplete, local search algorithm
- Evaluation function:
  - The min-conflict heuristic of minimizing the number of unsatisfied clauses
- Algorithm checks for satisfiability by randomly flipping the values of variables
- Balance between greediness and randomness

First-Order Logic



- objects and relations are semantic primitives
- syntax: constants, functions, predicates, equality, quantifiers.
- Predicates (applied to terms) have a truth value (i.e., true or false)
  - e.g., < (less than) is a predicate so, $x < 3 + 5$ is either true or false
- Functions just construct new terms out of other terms
- Increased expressive power: sufficient to define Wumpus world.
  - Quantifiers, Equality …

Unification and Generalised Modus Ponens

- Rules for quantifiers — UI, EI etc.
- Reducing FOL to PL
- Unification
- Generalised modus ponens (GMP)

Most General Unifier (MGU)

Unifying *Knows(John, x)* and *Knows(y, z)*

$$\theta = \{y/John, x/z\} \quad \text{or} \quad \theta = \{y/John, x/John, z/John\}$$

The first unifier is more general than the second.

FOL: There is a **single** most general unifier (MGU) that is unique up to renaming of variables.

$$\text{MGU} = \{y/John, x/z\}$$

Can be viewed as an equation solving problem.

- *i.e. solve* $Knows(John, x) \stackrel{?}{=} Knows(y, z)$

MGU Exercises

- P(x,A) =?= P(f(y),y)
- P(x,g(x)) =?= P(f(y),y)

- Decomposition
- Conflict
- Eliminate
- Delete
- Switch
- Coalesce
- Occurs Check

Inference

- Forward chaining
  - Informally: Unify all the assumptions in an implication rule with facts in KB to discharge them. If successful, add instantiated conclusion to KB.
    This is like a discovery process.
- Backward chaining
  - Informally: Unify conclusion of an implication rule with some fact in KB. If successful, add instantiated assumptions to KB as new goals.
    This is like a decomposition into sub-problems.

Resolution

- Formal statements for various types of resolution.
- Refutation method works by looking for a contradiction i.e., tries to derive falsity (empty clause).
- Need to negate the goal before converting it to clausal form.

Example

9. Which of the following clauses is the result of resolving clause $P(G(y)) \vee \neg Q(G(y))$ with clause $\neg P(F(x)) \vee Q(x)$, assuming they can be resolved?

   (a) $P(G(y)) \vee \neg P(F(x))$.

   (b) $Q(y) \vee \neg Q(F(y))$.

   (c) Resolution fails.

   (d) $P(G(y)) \vee P(F(G(y))) \vee \neg Q(G(y))$.

   (e) $P(G(y)) \vee \neg P(F(G(y)))$.

Example

9. Which of the following clauses is the result of resolving clause $P(G(y)) \vee \neg Q(G(y))$ with clause $\neg P(F(x)) \vee Q(x)$, assuming they can be resolved?

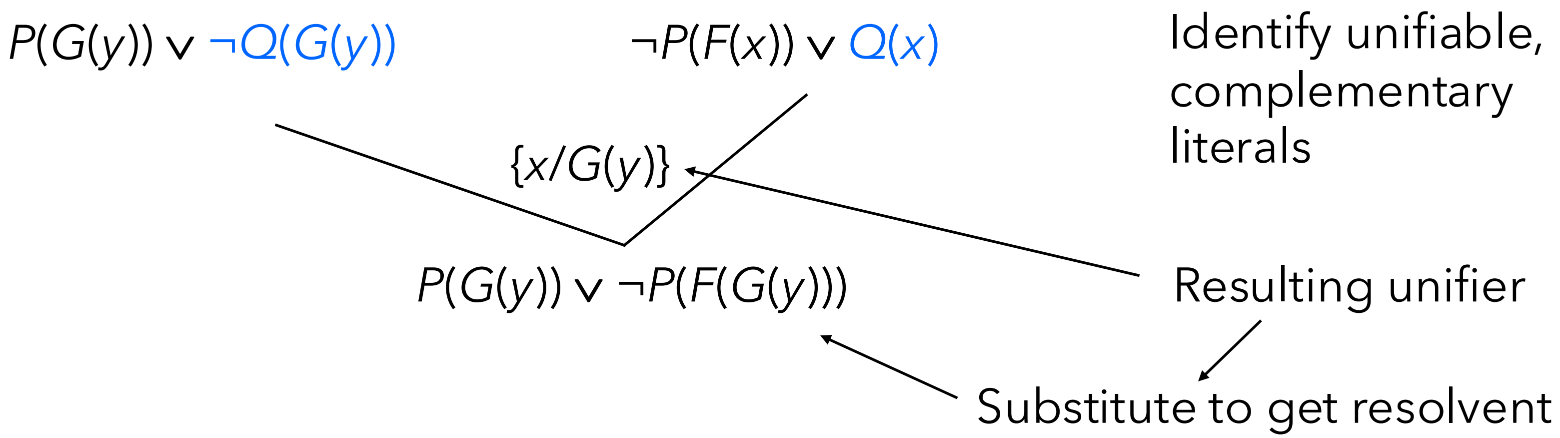
(a) $P(G(y)) \vee \neg P(F(x))$.

(b) $Q(y) \vee \neg Q(F(y))$.

(c) Resolution fails.

(d) $P(G(y)) \vee P(F(G(y))) \vee \neg Q(G(y))$.

(e) $P(G(y)) \vee \neg P(F(G(y)))$.



Thank you!