REVISION, SYMBOLIC AI

SET THEORETIC NOTATION

$A = \{a, b\}$

OPERATORS $\cap \; \cup \; \in \; \subseteq$ (intersection, union, member, subset of)

PROPOSITIONAL ATOM = $p, q, r, \ldots$

LITERALS = $p, \neg p, \top, \bot$

PROPOSITIONAL FORMULA

– anything made only from atoms, $\top, \bot, \neg, \wedge, \vee, \to, \leftrightarrow$

PRECEDENCE

1. NOT $\neg$, THERE EXISTS $\exists$, FOR ALL $\forall$
2. AND $\wedge$ (CONJUNCTION), OR $\vee$ (DISJUNCTION)
3. IF-THEN $\to$, IFF $\leftrightarrow$
4. LOGICALLY IMPLIES (SEMANTICALLY IMPLIES) $\models$, DOES NOT IMPLY $\not\models$; LOGICALLY EQUIVALENT $\equiv$

$h_v(p)$ = **PROPOSITIONAL EVALUATION**

VALID FORMULA $h_v(A) = T$ FOR ANY VALUATION OF ATOMS, v

SATISFIABLE FORMULA $h_v(A) = T$ FOR SOME VALUATION OF ATOMS

EQUIVALENCE $A \equiv B$ IF $h_v(A) = h_v(B)$ FOR ANY VALUATION OF v

CLAUSAL-FORM REPRESENTATION

CNF = $(a \vee b) \wedge (c \vee d)$ = $\{\{a,b\},\{c,d\}\}$ — both: p, $\neg p$, $a \wedge b \wedge c$, $a \vee b \vee c$

DNF = $(a \wedge b) \vee (c \wedge d)$

RESOLVENT ⇒ $(p \vee a) \wedge (\neg p \vee b) \equiv (a \vee b)$ — $\{p,a\},\{\neg p,b\} \equiv \{a,b\}$ (resolvent)

DAVIS PUTNAM = DP

1. **UNIT CLAUSE** = clause with only 1 s ⇒ remove $\{s\}$, s is true; $\{s\}\{\neg s,a\}\ldots \Rightarrow \{a\}\ldots$ — remove $\neg s$ from clauses
2. **PURE LITS** = s appears, $\neg s$ doesn't ⇒ s is true, remove clauses with s; $\{s,a\}\{b\}\{s,c\}\ldots \Rightarrow \{b\}\ldots$
3. **RESOLUTION** — s and $\neg s$ appear once & combine disjunctions; $\{s,a\},\{\neg s,b\} = \{s \vee a\} \wedge \{\neg s \vee b\} \Rightarrow \{a \vee b\} = \{a,b\}$ (can get messy)
4. **DLL = DPLL** ⇒ try one atom is false to see if satisfiable; if not, try that atom is true and test satisfiable; if not ⇒ unsatisfiable

$q \to p \equiv \neg q \vee p$

$\neg \forall x A \equiv \exists x (\neg A)$

$\neg \exists x A \equiv \forall x (\neg A)$

$\exists x (A \to B) \equiv (\forall x A) \to B$

$\forall x (A \to B) \equiv (\exists x A) \to B$

$C_1 = \{p, r\}$

$C_2 = \{\neg p, s\}$

RESOLVENT $(C_1, C_2, p) = \{r, s\}$ = RESOLVENT $(C_2, C_1, \neg p)$

The third term is as it appears in the first

$\mathcal{L}$-TERM = constants, variables, function of constant/variable

$\mathcal{L}$-FORMULA = TERMS CONNECTED WITH $\neg \vee \wedge \to \leftrightarrow = \forall \exists$

FOL SIGNATURE, $\mathcal{L}$, CONTAINS $C = \{\text{constants}\}$ eg identity(x); P_i = set of i-input predicates

PREDICATES RETURNING BOOLEAN, P_i = set of i-[illegible] …

[illegible] ⇒ **FUNCTIONS** MAP OBJECTS TO OTHER OBJECTS, eg DOUBLE(x) = [illegible]

$\mathcal{L}$-STRUCTURE M HAS A DIAGRAM

$\varphi(x)$ is the set of all things that $\varphi(x)$ = TRUE

GROUND TERM = NO VARIABLES

BOUND VARIABLE = VARIABLE UNDER A QUANTIFIER, $\exists$ or $\forall$

EXAMPLE $\forall x \, (x \to y)$ x is bound, y is **FREE**

SENTENCE = NO FREE VARIABLES

LOGIC PROGRAMMING NOTATION

CLAUSE = $\mathcal{L}$-formula (disjunctive, bound variables) $A \leftarrow B, C$ (head, body) $\equiv A \leftarrow B \wedge C \ldots$ equivalent

DEFINITE CLAUSE HAS EXACTLY ONE POSITIVE LITERAL, eg $(A \vee \neg B \vee \neg C \ldots)$

PROGRAM = SET OF DEFINITE CLAUSES

FACT = 1 positive literal, no negatives, eg $\{A\}$. RULE has negatives

QUERY = ALL NEGATIVE ATOMS, eg $\{\neg A \vee \neg B \vee \neg C \ldots\}$

SUBSTITUTION θ = (X/a) ∴ REPLACE X by a

IDEMPOTENT SUBSTITUTION $(F\theta)\theta = F\theta$

UNIFIERS = SUBSTITUTIONS THAT MAKE FORMULAS MATCH, eg $F(y,b)$ and $F(a,y)$

MGU = **MOST GENERAL UNIFIER** OF TWO FORMULAS, F_1, F_2 = most general substitutions

$\varphi(x)$ = set of all things that are X or make X true, $\varphi[\ldots]$ = [illegible]; $\varphi[\ldots]$ = always true

HERBRAND INTERPRETATION = all ground terms refer to themselves

B_L = **HERBRAND BASE** = set of all ground atoms/set of all functions/predicates applied to all objects

M = **HERBRAND MODEL** = set of atoms making statements F true

[illegible] = subset of B_L that are true in M; SUPPORTED = implied true

$T_P \uparrow^{\omega}$ = **MINIMAL SUPPORTED MODEL** = smallest set of things that are true from it

Eventually equivalent when it stops growing is $T_P \uparrow^{\omega}$

Start with empty set, add all things implied to be true

$T_P \uparrow^0 = \{\}$

IMMEDIATE CONSEQUENCE OPERATOR $= T_P(T_P \uparrow^{n-1})$

$T_P \uparrow^1$ = {first round of implications}

$T_P \uparrow^n = T_P$[nth round] = {when it has stopped changing}

DEFINITE CLAUSE $\forall x \forall y \, (\neg p(x,y) \vee q(x,y)) \equiv q(x,y) \leftarrow p(x,y)$

QUERY $\forall y \forall y \, (\neg p(y,y) \vee \neg q(y,y)) \equiv \; \leftarrow p(y,y), q(y,y)$ (positive head; negation of negatives)

SLD TREE RESOLUTION SEARCH TREE QUERY $\leftarrow p(x) \equiv \neg p(x)$

= DO ALL THESE STATEMENTS $C_1, C_2, C_3 \ldots$ IMPLY $\neg p(x)$ $[\,? \models \exists x \, p(x)\,]$

Example clause C_1 [$a = f(x, y)$], $\theta_1 = \{y, x\}$ … [illegible]

All clauses with p in head ⇒ C_1, C_2, $C_3 = q(y), r(y), s(y)$ for example

Root: $p(x)$; C_1 → $t(y)$ → ▪; C_2 → infinite loop; C_3 → $r(y), s(y)$ → $r(y)$ → □ ← empty set = successful condition

FINITE FAILED TREE = all branches end in finite failure

Finite failed refutation = can't [illegible]

Make substitutions to make it true

SUCCESSFUL TREE = at least one successful branch

SEARCHES

BREADTH-FIRST FIFO

DEPTH-FIRST LIFO

DEPTH-LIMITED SEARCH

ITERATIVE DEEPENING SEARCH

UNIFORM COST SEARCH = UCS = EXPAND NODE W/ LOWEST COST TO FRONTIER

INFORMED A^* = EXPAND NODE W/ LOWEST COST INVESTED + COST EST. SUM = $f = g + h$

BI-DIRECTIONAL = SEARCH FROM START + SEARCH BACKWARDS FROM GOAL

GRAPH-SEARCH = KEEP TRACK OF EXPLORED STATES
TREE-SEARCH = WITHOUT SET OF EXPLORED STATES

b = BRANCHING, d = MIN DEPTH, m = MAX DEPTH OF SEARCH

BRUTE FORCE $O(b^m)$

ALPHA-BETA → $O(b^{m/2})$ α = MIN VALUE SECURED BY MAX PLAYER, β = MAX VALUE SECURED BY MIN PLAYER

START WITH $\alpha = -\infty$, $\beta = +\infty$

MAX PLAYER, IF VAL > β RETURN β, IF VAL > α, α = MAX(α, VAL) CALL MIN WITH α, β

MIN PLAYER, IF VAL < α RETURN α, IF VAL < β, β = MIN(β, VAL), CALL MAX WITH α, β

HEURISTIC = ESTIMATE OF COST TO REACH GOAL

ADMISSABLE HEURISTIC = NEVER OVER-ESTIMATE THE COST
– CAN BE FOUND AS THE COST OF THE SOLUTION TO A RELAXED PROBLEM

CONSISTENT HEURISTIC = ESTIMATION DOESN'T GO DOWN AS WE PROGRESS

COST ESTIMATE N→GOAL = $h(N)$ ≤ COST($N \to N'$) + $h(N')$ = COST EST OF N' → GOAL

BELIEF STATE

UNINFORMED GIVEN WHAT AN AGENT KNOWS, THERE WILL BE A SET OF POSSIBLE STATES IT COULD BE IN – THESE STATES ARE INDISTINGUISHABLE W/ WHAT IT KNOWS.

BELIEF STATE SPACE = SET OF ALL BELIEF STATES EG $b = \{(B,1,1), (B,1,0)\}$

SOLUTION = ALL STATES IN BELIEF STATE ACHIEVE THE GOAL

PARTIALLY OBSERVABLE PERCEPT(s) = INFO RECEIVED BY AGENT IN STATE s

PARTIALLY OBSERVABLE AND-OR SEARCH – OR NODES BRANCH ES OF DIFFERENT AGENT ACTIONS
– AND NODES BRANCH FOR DIFFERENT FEEDBACK FROM ENV

SOLUTION OF AND-OR INCLUDES ALL POSSIBLE FEEDBACK (AND BRANCHES) AND IT CONTAINS A GOAL STATE IN EACH LEAF

DEFINITE LOGIC PROGRAM

C = {CONSTANTS} EG {a, b}
P_1 = {ORDINARY PREDICATES} EG {p, q} P_2 = {TWO-PLACE PREDICATES} EG {r, s, t}

p(X) ← q(X) ⇒ LOGIC PROGRAMMING SYNTAX
r(X,Y) ← S(X,Y), T(Y,Y) COMMA = AND
p(a)
q(b)
S(a,b)

PLANNING DOMAIN DEFINITION LANGUAGE = PDDL

FLUENT = GROUND, FUNCTION-LESS ATOM, EG At(Fred, Sydney)

STATE = CONJUNCTION OF FLUENTS
– ALL FLUENTS NOT MENTIONED ARE CONSIDERED TO BE FALSE

~~ACTIONS~~

ACTION = NAME, LIST OF PARAMETER VARIABLE, PRE-CONDITIONS, POST-EFFECTS

APPLICABLE ACTION = ACTION WHOSE PRECONDITIONS ARE MET IN CURRENT STATE

DELETE LIST IS LIST OF NEG LITERALS IN POST, ADD LIST IS LIST OF POSITIVE INPUT

APPLYING ACTION = REMOVE DELETE LIST FROM STATE, ADD THE ADD LIST

EXAMPLE:

INIT (IsCargo(C_1) ∧ IsPlane(P_1) ∧ IsAirport(SFO) ∧ IsAirport(JFK) ...)
GOAL (At(C_1, JFK))
ACTION (Fly(p, from, to),
PRE At(p, from), IsAirport(from), IsAirport(to), from ≠ to))

C_1, P_1, SFO, JFK... = OBJECTS IN PLANNING PROBLEM P

Fly(P_1, SFO, JFK) = GROUND ACTION IN PLANNING DOMAIN D

SEQUENTIAL PLAN = LIST OF GROUND ACTIONS

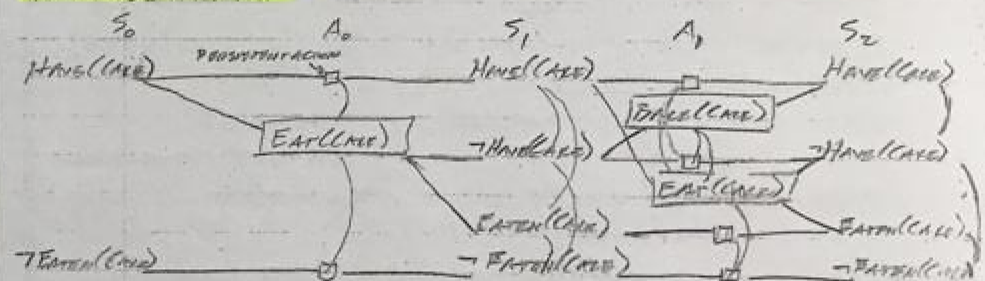
INDUCED STATE SEQUENCE = LIST OF STATES FROM EXECUTABLE PLAN

EXECUTABLE PLAN = EACH STEP IS APPLICABLE FROM S_0, ∃ INDUCED STATE SEQUENCE

VALID PLAN = EXECUTABLE PLAN THAT ACHIEVES GOAL

RELEVANT ACTIONS = COULD BE LAST STEP IN ACHIEVING GOAL

PLANNING GRAPH



MUTEX ACTIONS: INCONSISTENT EFFECTS = ONE NEGATES AN EFFECT OF THE OTHER

INTERFERENCE = EFFECTS OF ONE NEGATE PRECONDITION OF OTHER

COMPETING NEEDS = PRECONDITIONS ARE MUT EX

MUTEX LITERALS: LITERALS ARE NEGATION OF EACH OTHER

INCONSISTENT SUPPORT = PATHS THROUGH MUTEX ACTIONS, DIMINISHES W/ STEPS

LEVEL COST OF G = LEVEL AT WHICH IT FIRST APPEARS

MAX LEVEL HEURISTIC = MAXIMUM LEVEL COST OF ANY ONE GOAL (ADMISSIBLE)

LEVEL SUM HEURISTIC = SUM OF ALL LEVEL COSTS

SET-LEVEL HEURISTIC = LEVEL WHERE ALL GOALS APPEAR, NONE MUTEX

NO GOOD = GOAL THAT ISN'T MET, DIMINISHES

TERMINATE WHEN THE GRAPH (ACTIONS/LITERALS) STOPS CHANGE + MUTEX/NO-GOOD