dealing with inconsistency

- inconsistency arises with non-deductive, plausible reasoning
  - defaults
  - incomplete info
  - inexact, time-varying info
  - inferences based on heuristic assumptions (later retracted)
- what to do?
  - logic doesn't help:

    suppose P_1 &…& P_n imply O but ¬O. Logic only tells us that **one** of the P_i must go.
  - probability theory doesn't (immediately) help either:

    $p(h \mid e)$ where e is inconsistent ==> dividing by 0...
  - ex falso quodlibet

    so we need a method to draw reasonable inferences from inconsistent premisses
- default logic: prerequisite(x) & ◆ F(x) ∴ F(x)
  - ◆ p ≈ it is **consistent** to assume p, but p need not be known….

reason maintenance systems (RMS)

- deduction is 'additive'
  - Deduction theorem:

    $\{P_1, \ldots, P_n\}$ imply Q $\rightarrow$ $\{R\} \cup \{P_1, \ldots, P_n\}$ imply Q
- nonmonotonic
  - basic nonmonotonic inference:

    if ¬S cannot be inferred from KB - within given resource limits - then infer S
  - new evidence may cause deletions, not just additions:
    plausible guesses may turn out wrong, and may have to be removed later…
- RMS: mechanism to restore consistency
  - does book keeping
  - cannot decide which of several jointly inconsistent assumptions to drop
  - linear on average, $O(n^3)$ for nonmonotonic loops

RMS

- there is a node for each belief; there is a reason for each step
- RMS believes a node iff there is an argument for it and RMS believes all nodes in the argument
- set of current beliefs is recursively updated
  - if a new justification leads to new belief in node n, then further nodes may come to be (dis)believed via previously incomplete arguments
- reason for belief $B = \langle \{x_i\}, \{y_j\} \rangle \ni$ B is IN iff each x_i is IN, each y_j OUT
- assume B: $\langle \{\ \}, \{\neg B\} \rangle$

assume N1 (she loves me): <{be optimistic}, {N2 (she doesn't love me)}> assumption

N2: currently unjustified;

N3: <{rule2, N1}, { }> normal inference

later: N2: <{rule3, N4}, { }>;

suppose rule3, N4 are believed; then N2 believed; then N1 disbelieved; then N3 disbelieved. If N2 later goes OUT again, N1, N3 go back IN...

RMS makes the abstract relation of justification explicit

- maintains justifications, explains deductions
- **incrementally** updates beliefs when premises are added or removed
- 'dependency-directed backtracking'

- propositional deduction by **constraint propagation:**
- **deduce a value iff it follows from previous value and a single constraint**
- each cell/node can take on values {T,F,U}, constrained by constraints. Propagate until constraints are relaxed
- DB consists of beliefs: (assertion.T-value)
- term t = (n.F) is T iff n is F; is F iff n is T

logical constraints

- Logical constraints: disjunctive clauses
  - e.g. $p_1 \& \ldots \& p_n \rightarrow q \approx (v(p_1.F)(p_2.F)\ldots(p_n.F)(q.T))$
    so, when all p_i are T, q **must** be T!
- clauses can be interpreted in many ways, no preferred direction (MP, MT etc):
  - a,b,c contradictory $\approx \neg(a\&b\&c) \approx (v(a.F)(b.F)(c.F)) \approx$
    $a\&b \rightarrow \neg c$; $c\&b \rightarrow \neg a$; $a \rightarrow (c \rightarrow \neg b)$; $a \rightarrow (b \rightarrow \neg c)$; $c \rightarrow (a \rightarrow \neg b)$ …

- finding constraints for 'v', ' $\rightarrow$ ' etc.:
  - (v (p $\rightarrow$ q.F) (p.F) (q.T))
    i.e. if conditional T, antecedent T $\rightarrow$ consequent T as well
    if antecedent T, consequent F $\rightarrow$ conditional F
  - (v (p.T) (p $\rightarrow$ q.T)) i.e. if antecedent F $\rightarrow$ conditional T
  - (v (q.F) (p $\rightarrow$ q.T)) i.e. if consequent T $\rightarrow$ conditional T…

logical constraint propagation

p	q	p v q	p → q
T	T	T	T
T	F	T	F
F	T	T	T
F	F	F	T

v: exclude TTF, TFF, FTF, FFT:
¬(p&q&¬(pvq))&¬(p&¬q&¬(pvq))
&¬(¬p&q&¬(pvq))&¬(¬p&¬q&(pvq))
iff

(¬pv¬qv(pvq))
(¬pvqv(pvq))
(pv¬qv(pvq))
(pvqv¬(pvq))

(¬pv(pvq))
(¬qv(pvq))
(pvqv¬(pvq))

(v((or pq).F)(p.T)(q.T))
(v((or pq).T)(p.F))
(v((or pq).T)(q.F))

→ : exclude TTF, TFF, FTF, FFF:

(¬pvqv¬(p → q))
(pv¬qv(p → q))
(pvqv(p → q))
(¬pv¬qv(p → q))

(¬pvqv¬(p → q))
(pv(p → q))
(¬qv(p → q))

(v((→ pq).F)(p.F)(q.T))
(v((→ pq).T)(p.T))
(v((→ pq).T)(q.F))

logical constraint propagation, cont.

¬: (v((not p).T)(p.T))
(v((not p).F)(p.F))

e.g. assertion of (r v s), ¬r yields s:

(**or** r s) made T
constraints:
(v((**or** r s).F)(r.T)(s.T))

hence (s.T)

(**not** r) made T
constraints:
(v((**not** r).F)(r.F))
hence (r.F)

justification points to the single clause used to deduce a value!

unfortunately, this simple logic is incomplete!

q → p, ¬q → p implies p but (v(q.F)(p.T)) & (v(q.T)(p.T)) does not imply p (because q is unknown)…

logical constraint propagation, cont.

prem: (⟶ a b), (⟶ b c); prove (⟶ a c)

assume: ¬(⟶ a c) ⇔ ((⟶ a c).F); so

(v((⟶ a c).T)(a.T) hence (a.T)

(v((⟶ a b).F)(a.F)(b.T)) hence (b.T)

(v((⟶ b c).F)(b.F)(c.T)) hence **(c.T)**

(v((⟶ a c).T)(c.F)) hence **(c.F)**

contradiction: a **contradiction** is any **unsatisfiable** clause with all terms F.
e.g. if p.F, q.T, then (v(p.T)(q.F)) is a contradiction

• pick one underlying premise for retraction; to prevent reoccurrence of contradiction, deduce negation of premise via…

• **add: negation of all premises underlying the contradiction (log true!)**; from this one can deduce the negation of any premise if all others are believed.

add: (v((→ a b).F)((→ b c).F)((→ a c).T)))

• **remove assumption**: ((→ a c).F) becomes ((→ a c).U);

now, when one premise is retracted, its negation is deduced: hence ((→ a c).T) !!!

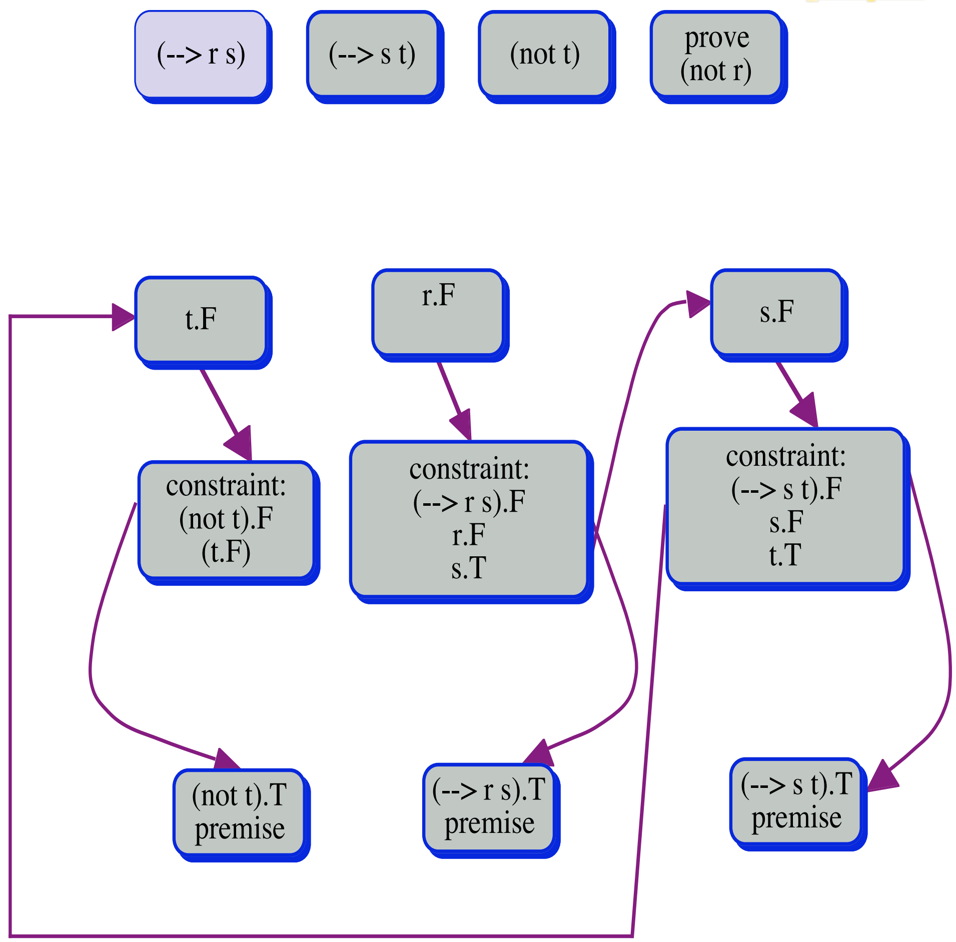
logical constraint propagation, cont.

prem: (⟶ a c), (⟶ b c), (or a b); RMS cannot deduce c!
assume: ¬c as additional premise
contradiction:
¬a, ¬b, a v b, i.e. (v((or a b).F)(a.T)(b.T))
underlying premise:
(⟶ a c)&(⟶ b c)&(or a b)&(not c)
negate this logical falsehood, add it as further premise:
* (v((⟶ a c).F)((⟶ b c).F)((or a b).F)((not c).F))
pick the assumption for retraction:
((not c).U)
* and premises imply ((not c).F) i.e. (c.T) !!

indirect proofs enhance the deductive power of 1-step propagation: to prove c, assume ¬c. If contradiction, add neg of all premises and deduce c; if no contradiction, c is not deducible, remove assumption of ¬c.

when premise p is retracted: remove all deduced T-values which depend on p (make U); check all clauses with an assertion whose T-value was removed to see if clause now invalidly supports another value; if so, retract that value recursively; check all assertions with removed T-values to see if a T-value can be deduced in other ways, i.e. make all deductions which would have been made, if RMS had started with the new premise set.

logical constraint propagation, cont.



here, the justification is a pointer **to** the clause used to deduce this value.

(why 'r)
r False from 1) (--> r s).T and 2) s.F
(why 2)
s False from 1) (--> s t).T and 2) t.F

most birds fly, few birds swim ...

$\forall$x (Bird x $\wedge$ ¬ Penguin x $\wedge$ ¬ Ostrich x $\wedge$ ¬ TwoWattledCassowary x
$\wedge$ ¬ BrokenWing x $\wedge$ ¬ LegsStuckInCement x $\wedge$ ……… $\Rightarrow$ Fly x)
premise: Bird birdie; prove: Fly birdie, i.e.
prove subgoals: ¬ Penguin birdie etc etc. Impossible! Instead

Bird x $\wedge$ poss Fly x then Fly x

where 'poss' means 'it's compatible with all we know that …'.
E.g. we know that $\forall$x (Penguin x $\Rightarrow$ ¬Fly x) etc.
"few birds swim": Bird x $\wedge$ poss ¬Swim x then ¬Swim x

Using RMS to make such non-monotonic inferences: **premise categories**:

(assert p)	p is true
(assume p)	p is less certain than solid fact
(likely p)	p is likely true…
(very_likely p)	….

in case of contradiction, **retract least likely premise** (ask user if premises are equally likely)

using RMS for nonmonotonic inferences

1) assert: (⇒ (Bird x)(poss Fly x))

(i.e. (assert '(⇒ (Bird x)(likely (Fly x))))

2) assert or deduce: Bird x

prove: Fly x [T-value likely] i.e. (likely (Fly x))

for each (likely p), generate automatically:

(⇒ (likely p) p), assumed LIKELY

if p leads to a contradiction, retract (⇒ (likely p) p)!!!

3) (⇒ (likely (Fly x))(Fly x)) ; this is an assumption of degree LIKELY

now we easily derive Fly x which depends on 3).

Suppose we get a contradiction: (v (1.F)(2.F)(3.F))

retract 3): (3.U), thus (3.F);

this, together with constraint for (3 ⇒ Fly x):

(v ((⇒ (likely (Fly x))(Fly x)).T)((Fly x).F)))

implies ((Fly x).F), i.e. ¬Fly x. OK!

using RMS for nonmonotonic inferences, instead of Default Logic

- non-monotonicity expresses **uncertainty** in terms of **typicality** and **defeasibility**
  - and in terms of assumptions and possible counterexamples

- 'A and poss C implies C' means
  'infer C from A if C is typical or no defeating reasons such as an incrementally built list of exceptions are in the data base';
  - this may be more plausible than 'if A then (0.25) C' which might only mean that many exceptions are known to 'if A then C'.
- RMS can also be used for counterfactuals….

- there are many other kinds of uncertainties….

using RMS to handle counterfactuals?

- counterfactuals: a tough nut for fol; RMS to the rescue
  - "if I were rich, I would give each of you $10000". True or false?
- p ➨ q: if p were the case, q would be the case
  - in ordinary fol, p ➨ q T for **arbitrary** q because of false antecedent!

- p ➨ q T in world w iff q T in p-containing w' s.t. w' is **most similar** to w
  - assume p T
  - propagate; … contradiction
  - find underlying maximal inconsistent set, using RMS
  - either: remove weakest member ≠ p, propagate;
    or: remove each member in maximal inconsistent set in turn to see which causes **minimal upheaval** in remaining beliefs; remove this, propagate;
  - if q T in resulting belief set, p ➨ q T !
  - now make p F again; propagate.