Informatics 1

Lecture 11 Reprise

Michael Fourman

Models and Satisfaction

- Four methods:
  - Enumeration (Truth Tables)
  - Naive search with simplification
  - Directed search with unit propagation (U-P)
  - Lazy search, with watched literals and U-P

X has a model iff

$X \cup \{A\}$ has a model or $X \cup \{\neg A\}$ has a model.

where X is a any set of formulae and A any propositional letter,

X has a model iff

Xu{A,B} has a model or Xu{¬A,B} has a model

or

Xu{A,¬B} has a model or Xu{¬A,¬B} has a model.

X has a model iff

$X \cup \{A,B,C\}$ has a model or $X \cup \{\neg A,B,C\}$ has a model

or

$X \cup \{A,\neg B,C\}$ has a model or $X \cup \{\neg A,\neg B,C\}$ has a model

or

$X \cup \{A,B,\neg C\}$ has a model or $X \cup \{\neg A,B,\neg C\}$ has a model

or

$X \cup \{A,\neg B,\neg C\}$ has a model or $X \cup \{\neg A,\neg B,\neg C\}$ has a model.

$X \cup V$ has a model iff

$X \cup V \cup \{A\}$ has a model or $X \cup V \cup \{\neg A\}$ has a model

2^N cases

The Davis-Putnam method is based on two simple facts about truth-table logic.

First, where X is a any set of formulae and A any propositional letter,

X has a model iff

$$X \cup \{A\} \text{ has a model or } X \cup \{\neg A\} \text{ has a model.}$$

Second, where A and B are formulae,

$$\mathbf{A \wedge (\neg A \vee B) \equiv A \wedge B}$$

and

$$\mathbf{A \wedge (A \vee B) \equiv A.}$$

It follows that the application of unit propagation to any set of propositional clauses results in an equivalent set.

Valuations, V (e.g V = {¬A,¬B,C})

a **consistent** set (conjunction) of literals

(at most one of L and ¬L is in V)

$V \models X$ "V **models** X" iff

$$\forall C \in X \,.\, V \text{ establishes } C$$

$V \models C$ "V **establishes** C" iff

$$V \cap C \neq \emptyset$$

$V \models \neg C$ "V **contradicts** C" iff

$$\forall L \in C \,.\, (\neg L) \in V$$

$V \models \neg X$ "V **contradicts** X" iff

$$\exists C \in X \,.\, V \models \neg C$$

C is a (disjunctive) clause
X is a CNF

X is satisfiable iff for some valuation V
$V \models X$

Write a function S(X, V) such that

S(X, V) is true iff there is a valuation $W \supseteq V$. $W \models X$

Then S(X, ∅) is true iff X is satisfiable.

For each literal L such that $L \notin V$ and $\neg L \notin V$

S(X, V) iff S(X, V^L) or S(X, V^¬L)

For any given V it is easy to test whether $V \models X$

So a simple, correct (and slow) implementation is

S(X, V) =

$V \models X$ || $\exists L$. $L \notin V$ and $(\neg L) \notin V$ such that

S(X, V^L) || S(X, V^¬L)

X is satisfiable iff for some valuation V
$V \models X$

if $V \models \neg X$ and $V \subseteq W$ then $W \models \neg X$

So we can abandon a line of enquiry once $V \models \neg X$

For any given V it is easy to test whether $V \models \neg X$

So a less simple, less slow, correct implementation is

$S(X, V) =$

$V \models X \;||\; (V \not\models \neg X \;\&\&$

$\exists L.\ L \notin V$ and $(\neg L) \notin V$ such that

$(S(X, V \wedge L) \;||\; S(X, V \wedge \neg L)))$

Unit Literals

$V, X \models L$ "V, X **entails** L" **iff**

$(\neg L) \notin V$ and $\exists C \in X \,.\, L \in C$ and $V \models \neg(C \backslash \{L\})$

$\forall L' \neq L \in C \,.\, (\neg L') \in V$

If $W, X \models L$ then $S(X, W)$ iff $S(X, W \wedge L)$

$$S(X, V) =$$

$$V \models X \;\|\; \exists L.\ V, X \models L \text{ and } S(X, V \wedge L) \;\|$$

$$(V \not\models \neg X \;\&\&$$

$$\exists L.\ L \notin V \text{ and } \neg L \notin V \text{ such that}$$

$$(\ S(X, V \wedge L) \;\|\; S(X, V \wedge \neg L)\ )\ )$$

Watched Literals

V ⊨ X || **∃L. V, X ⊨ L and S(X, V^L)** ||

(V ⊭ ¬X && *keep searching*

For each C ∈ X we watch two literals

either both are not contradicted by V

or at least one is in V

We react only when one

When we consider V^L′ …

C reacts if C is watching ¬L′

(since L′ contradicts ¬L′)

Why Watched Literals Work

V, L′ ⊨ X || ∃L. V, L′, X ⊨ L and S(X, V^L′^L)||

(V, L′ ⊭ ¬X && *keep searching*

For each C ∈ X we watch two literals either both are not contradicted by V
or at least one is in V

When we consider V^L′ and L∉ V

1. V, L′, X ⊨ L if some C is watching both L and ¬L′
   and these are the **only** literals in C\V
2. V, L′ ⊨ ¬X only if u-p produces a contradiction
3. V, L′ ⊨ X iff there is no contradiction and no
   no remaining watched literals

1. $V \not\models \neg C$ unless V contradicts every literal in C
2. $V \models C$ iff V establishes some literal in C
3. $V, C \models L$ iff V contradicts every literal but L in C

As V gets longer it establishes and contradicts more literals. We watch literals, as long as there are two or more uncontradicted literals we watch two of them.

If one of our watched literals is contradicted, we try to find another . If we fail, then 3 is true; we claim our prize, and try to establish L; other clauses may try to establish other literals, at which point we may discover a contradiction, or continue.

If the search backtracks, V gets shorter, both of our two watched literals are again uncontradicted.