Theorem Proving

Christopher Lynch
Clarkson University

Color Code for Talk

- Blue = where we are in the talk
- Green = Literal selected for inference
- Red = Emphasizes new points
  - Also used later for marked positions

Contents

- TP in Propositional Logic
- TP in First Order Logic
- TP in FOL with Equality
- Completeness
- Constrained Deduction
- TP modulo an Equational Theory

Propositional Logic

- Axioms:

$p \vee q$

$p \Rightarrow q$

$q \Rightarrow p$

- Conclusion:

$p \wedge q$

Convert to Clauses

- Negate Conclusion (goal)
- Convert to cnf
  - Remove implications, move negations inside (Demorgan, double negation), distribute
- Each conjunct becomes a clause
- Need to show unsatisfiability of set of clauses

Definitions

- An atom is a propositional symbol
- A literal is an atom or negated atom
- View each clause C as a multiset of literals
- neg(C) = set of all negated literals in C
  - neg($p \vee \neg q$) ={$\neg q$}

Example of Conversion

$p \vee q$

$\neg p \vee q$ (was $p \Rightarrow q$)

$p \vee \neg q$ (was $q \Rightarrow p$)

$\neg p \vee \neg q$ (was $p \wedge q$)

Resolution Inference Rules

$$\frac{\Gamma \vee A \qquad \neg A \vee \Delta}{\Gamma \vee \Delta}$$ Resolution

$$\frac{\Gamma \vee A \vee A}{\Gamma \vee A}$$ Positive Factoring

Soundness and Completeness

- Completeness: If a set S of clauses is unsatisfiable then the empty clause can be derived from S
- Soundness: If a set S of clauses is satisfiable then the empty clause cannot be derived from S
  - since conclusion of rule follows from premises
- Formally $\perp \in S_\omega$ iff S is unsatisfiable

Restrictions

- Search space is way too large
- How do we make search space smaller
- Eliminate Redundant Clauses
  - Subsumption
  - Tautology Deletion
- Only allow certain inferences
  - Ordered Resolution
  - Selected Resolution

Resolution Example

1(given). $p \vee q$
2(given) $\neg p \vee q$
3(given) $p \vee \neg q$
4(given) $\neg p \vee \neg q$
5(1,2,Res) $q \vee q$
6(5,Fac) q
7(3,4,Res) $\neg q \vee \neg q$
8(6,7,Res) $\neg q$
9(6,8,Res) $\perp$

Subsumption

- If $C \subseteq D$ then C subsumes D
  - $p \vee \neg r$ subsumes $p \vee \neg q \vee \neg r$
- Subsumed clauses can be removed without losing completeness

Tautology Deletion

- A tautology is a clause that is always true
  - i.e., same atom appears positively and negatively
  - $p \vee \neg q \vee \neg p$
- Tautologies can be removed without losing completeness

Restrictions

- Search space is way too large
- How do we make search space smaller
- Eliminate Redundant Clauses
  - Subsumption
  - Tautology Deletion
- Only allow certain inferences
  - Ordered Resolution
  - Selected Resolution

Redundancy (an abstract notion)

- C is redundant in S if there exist $C_1,\cdots, C_n$ in S such that each C_i is smaller than C and $C_1,\cdots,C_n \models C$
  - Clauses are compared using multiset ordering
- Redundant clauses can be removed without losing completeness
  - A subsumed clause is redundant
  - A tautology is redundant (n = 0)

Ordered Resolution

- We can give a total precedence on atoms
- Extend to negated atoms so that $\neg p > p$
- Let $\max(C) = \{L \in C \mid \neg \exists M \in C, M > L\}$
  - i.e., nothing smaller in clause
  - max (C) is a singleton set
- Inferences only need to be applied to a max literal in each clause
  - Inference system still complete

Ordered Resolution

$\Gamma \vee A \quad \neg A \vee \Delta$

---------- where $A \in \max(\Gamma \vee A)$

$\Gamma \vee \Delta$ and $\neg A \in \max(\neg A \vee \Delta)$

$\Gamma \vee A \vee A$

---------- where $A \in \max(\Gamma \vee A \vee A)$

$\Gamma \vee A$

Selection Rule

- $sel(C) \subseteq C$
- sel(C) is valid if sel(C) contains either
  - all members of max(C) OR
  - at least one member of neg(C)
- Example, for $\neg p \vee \neg q \vee r$ (with $p < q < r$), can select $\neg p$ or $\neg q$ or r
- Only need inferences involving selected literal in each clause

Why Ordered Resolution

- $S = \{p, q, \neg p \vee \neg q\}$
  - Without Ordered Resolution, 2 ways to refute
- $S = \{p_1, \cdots, p_n, \neg p_1 \vee \cdots \vee \neg p_n\}$
  - Without Ordered Resolution, 2^n ways to refute

Selected Resolution

$\Gamma \vee A \quad \neg A \vee \Delta$

---------- where $A \in sel(\Gamma \vee A)$

$\Gamma \vee \Delta$ and $\neg A \in sel(\neg A \vee \Delta)$

$\Gamma \vee A \vee A$

---------- where $A \in sel(\Gamma \vee A \vee A)$

$\Gamma \vee A$

Notes about Selection Rule

- You must select something in each clause
- Selecting a negative is forward chaining
- The selection rule can be different for each clause
- My formulation different than standard one (but identical)
- Selected Resolution is complete with valid selection rule (assume valid from now on)

Summarizing so far

- Propositional Resolution
- Two abstract notions covering all practical notions used by theorem provers
  - Redundancy allows us to delete clauses
  - Selected Resolution allows us to avoid inferences
- $\bot \in S_\omega$ iff S is unsatisfiable
  - S_ω is called saturated set

Invalid selection is not complete

$p \vee q$

$\neg p \vee q$

$p \vee \neg q$

$\neg p \vee \neg q$

Every inference creates a tautology and we cannot get $\bot$ (even if we keep tautologies)

Contents

- TP in Propositional Logic
- TP in First Order Logic
- TP in FOL with Equality
- Completeness
- Constrained Deduction
- TP modulo an Equational Theory

First Order Logic

- Axioms:

$\exists$ x P(x)

$\forall$ x $\exists$ y P(x) $\Rightarrow$ Q(x,y)

- Conclusion

$\exists$ x $\exists$ y P(x) $\wedge$ Q(x,y)

Skolemization

- Each existential variable is viewed as a function of all previous universal variables
  - Constant if no previous universal variables
- Result is equisatisfiable
- After Skolemization, all variables are universal, so quantifiers are removed

Convert to Clauses

- Negate Conclusion (goal)
- Convert to cnf
  - Remove implications, move negations inside, distribute, remove existential variables by Skolemization
- Each conjunct becomes a clause
- Need to show unsatisfiability of set of clauses

Example of Conversion

P(a) (was $\exists$ x P(x))

$\neg$ P(x) $\vee$ Q(x,f(x))

(was $\forall$ x $\exists$ y P(x) $\Rightarrow$ Q(x,y))

$\neg$ P(x) $\vee$ $\neg$ Q(x,y)

(was $\exists$ x $\exists$ y P(x) $\wedge$ Q(x,y))

Note: Variables in different clauses are considered as different variables

Definitions

- A term is a variable or n-ary function symbol applied to n terms ($n \geq 0$)
- An atom is an n-ary predicate symbol applied to n terms ($n \geq 0$)
- A literal is an atom or negated atom
- View each clause C as a multiset of literals

Soundness and Completeness

- $\perp \in S_\omega$ iff S is unsatisfiable
- Same as Propositional case

Resolution Inference Rules

$\Gamma \vee A \qquad \neg B \vee \Delta$

---------- Resolution

$(\Gamma \vee \Delta)\sigma$ where $\sigma = mgu(A,B)$

$\Gamma \vee A \vee B$

---------- Positive Factoring

$(\Gamma \vee A)\sigma$ where $\sigma = mgu(A,B)$

Resolution Example

1(given). P(a)

2(given) $\neg$ P(x) $\vee$ Q(x,f(x))

3(given) $\neg$ P(x) $\vee \neg$ Q(x,y)

4(1,2,Res) Q(a,f(a))

5(1,3,Res) $\neg$ Q(a,y)

6(4,5,Res) $\perp$

Restrictions

- Search space is possibly infinite
- How do we make search space smaller
- Eliminate Redundant Clauses
  - Subsumption
  - Tautology Deletion
- Only allow certain inferences
  - Ordered Resolution
  - Selected Resolution

Tautology Deletion

- A tautology is a clause that is always true
  - i.e., same atom appears positively and negatively
  - $P(x) \vee \neg Q(x) \vee \neg P(x)$
- Tautologies can be removed without losing completeness

Subsumption

- If $\exists \sigma$ $C\sigma \subseteq D$ then C subsumes D
  - $P(x) \vee \neg R(fx)$ subsumes
    $P(a) \vee \neg Q(b) \vee \neg R(f(a))$
- Subsumed clauses can be removed without losing completeness

Redundancy

- C is redundant in S if $\forall \theta$ $\exists C_1,\cdots, C_n$ in S and σ such that each $C_i\sigma$ is smaller than $C\theta$ and $C_1\sigma,\cdots,C_n\sigma \models C\theta$
- Redundant clauses can be removed without losing completeness
- Special case when there exists C' and σ such that $C'\sigma = C$ and σ not id

Restrictions

- Search space is possibly infinite
- How do we make search space smaller
- Eliminate Redundant Clauses
  - Subsumption
  - Tautology Deletion
- Only allow certain inferences
  - Ordered Resolution
  - Selected Resolution

Ordered Resolution

- Extend to negated atoms so that $\neg A > A$
- Let $\max(C) = \{L \in C \mid \neg \exists M \in C, M > L\}$
  - i.e., nothing smaller in clause
  - max (C) might not be a singleton set
- Inferences only need to be applied to a max literal in each clause
  - Inference system still complete

Properties of Ordering

- We need an Ordering on atoms that is
  - well-founded (no infinite descending chain)
  - Total on ground atoms
  - Stable under substitution ($A \leq B \Rightarrow A\sigma \leq B\sigma$)
- LPO and RPO work
- Note: Cannot be total on all atoms because how do we compare P(x) and P(y)

Ordered Resolution

$$\frac{\Gamma \vee A \qquad \neg B \vee \Delta}{(\Gamma \vee \Delta)\sigma} \quad \text{where } A\sigma \in \max(\Gamma \vee A)\sigma \text{ and } \neg B\sigma \in \max(\neg B \vee \Delta)\sigma \text{ and } \sigma = \mathrm{mgu}(A,B)$$

$$\frac{\Gamma \vee A \vee B}{(\Gamma \vee A)\,\sigma} \quad \text{where } A \in \max(\Gamma \vee A \vee B) \text{ and } \sigma = \mathrm{mgu}(A,B)$$

Without Ordered Resolution

1(given) P(a)
2(given) ¬ P(x) ∨ P(fx)
3(1,2,Res) P(fa)
4(3,2,Res) P(ffa)
5(4,2,Res) P(fffa)
…

But how do we prevent this?

1(given) Leq(x,x)
2(given) ¬ Leq(sx,y) ∨ Leq(x,y)
3(1,2,Res) Leq(x,sx)
4(3,2,Res) Leq(x,ssx)
…
Answer: I don't know

With Ordered Resolution

1(given) P(a)
2(given) ¬ P(x) ∨ P(fx)

No inferences at all

Selection Rule

- Same as before
- All maximal literals or one negative literal must be selected
- Only need inferences involving selected literal in each clause
- Selected Resolution is Complete

Selected Resolution

$\Gamma \vee A \quad \neg B \vee \Delta$

$(\Gamma \vee \Delta)\sigma$ where $A\sigma \in sel(\Gamma \vee A)\sigma$ and $\neg B\sigma \in sel(\neg B \vee \Delta)\sigma$ and $\sigma = mgu(A,B)$

$\Gamma \vee A \vee B$

$(\Gamma \vee A)\sigma$ where $A \in sel(\Gamma \vee A \vee B)\sigma$ and $\sigma = mgu(A,B)$

Contents

- TP in Propositional Logic
- TP in First Order Logic
- TP in FOL with Equality
- Completeness
- Constrained Deduction
- TP modulo an Equational Theory

Notes about Selection Rule

- The selection rule can be different for each clause
- However, different instances of the same clause must be selected consistently when used in an inference
- Since this is confusing, usually the selection rule just applied to uninstantiated clause

FOL with Equality

- Axioms

$\exists x\ Z(x)$

$\forall x \forall y\ Z(x) \Rightarrow f(x,y) = y$

$\forall x \forall y\ g(f(x,y)) = f(g(x),g(y))$

- Conclusion

$\exists x \exists y\ f(g(x),y) = y$

- Interpretation:f(x,y)=x+y,g(x)=2*x,Z=Zero

Example of Conversion

$Z(a)$ (was $\exists x\, Z(x)$)

$\neg Z(x) \vee f(x,y) = y$

(was $\forall x \forall y\, Z(x) \Rightarrow f(x,y) = y$)

$g(f(x,y)) = f(g(x),g(y))$

(was $\forall x \forall y\, g(f(x,y)) = f(g(x),g(y))$)

$f(g(x),y) \neq y$ (was $\exists x \exists y\, f(g(x),y) = y$)

Axiomatizing Equality

$x = x$

$x \neq y \vee y = x$

$x \neq y \vee y \neq z \vee x = z$

$x_1 \neq y_1 \vee \cdots \vee x_n \neq y_n \vee f(x_1,\cdots,x_n) = f(y_1,\cdots,y_n)$

$x_1 \neq y_1 \vee \cdots x_n \neq y_n \vee \neg P(x_1,\cdots,x_n) \vee P(y_1,\cdots,y_n)$

Definitions

- A term is a variable, or function symbol applied to terms
- An atom is a predicate symbol applied to terms, or s=t where s and t are terms
- A literal is an atom or negated atom
- View each clause C as a multiset of literals

Why not to axiomatize

- Viewing equality like other predicates and applying resolution to axioms blows up
  - Look at transitivity
- Instead we ignore axioms and add a new inference rule (Paramodulation)
  - Generalizing substitution of equals for equals

Ground Paramodulation

$\Gamma \vee s=t \qquad L[s] \vee \Delta$

$\Gamma \vee L[t] \vee \Delta$ Paramodulation

$\Gamma \vee t \neq t$

Γ Equation Resolution

No inferences into variables

- The restriction that s' is not a variable is crucial
- Because a variable could unify with everything
- And the search space would be huge

Paramodulation

$\Gamma \vee s=t \qquad L[s'] \vee \Delta$

$(\Gamma \vee L[t] \vee \Delta)\sigma$ Paramodulation where σ=mgu(s,s') and s' is not a variable

$\Gamma \vee s \neq t$

$\Gamma\sigma$ Equation Resolution where σ=mgu(s,t)

Completeness

- Paramodulation + Equation Resolution + Resolution + Factoring is Complete for FOL with equality

Paramodulation Example

1(given) Z(a)
2(given) $\neg$ Z(x) $\vee$ f(x,y) = y
3(given) g(f(x,y)) = f(g(x),g(y))
4(given) f(g(x),y) $\neq$ y
5(1,2,Res) f(a,y) = y
6(5,3,Para) f(g(a),g(y)) = g(y)
7(6,4,Para) g(y) $\neq$ g(y)
8(7,Eqres) $\perp$

Tautology Deletion

- A tautology is a clause of the form:
  - A $\vee$ $\neg$ A $\vee$ Γ or
  - $\Gamma \vee t = t$
- Tautologies can be removed without losing completeness

Restrictions

- Search space is possibly infinite
- How do we make search space smaller
- Eliminate Redundant Clauses
  - Subsumption
  - Tautology Deletion
- Only allow certain inferences
  - Ordered Resolution
  - Selected Resolution

Demodulation

- If we have $\Gamma \vee L[s']$
- and if we have s = t
- and if $\exists \sigma$ such that $s\sigma = s'$
- then we can add $\Gamma \vee L[t\sigma]$
- and remove $\Gamma \vee L[s']$
- Note: as long as $s\sigma = t\sigma \leq \Gamma \vee L[s']$

Demodulation Example

- We have $P(x) \vee Q(fhx)$
- and we have $fy = gy$
- then add $P(x) \vee Q(ghx)$
- and delete $P(x) \vee Q(fhx)$

Restrictions

- Search space is possibly infinite
- How do we make search space smaller
- Eliminate Redundant Clauses
  - Subsumption
  - Tautology Deletion
- Only allow certain inferences
  - Ordered Resolution
  - Selected Resolution

Redundancy

- Subsumption is exactly the same as before
- Redundancy is defined just like before

Properties of Ordering

- We need an ordering on terms that is
  - well-founded (no infinite descending chain)
  - Total on ground terms
  - Stable under substitution ($s \leq t \Rightarrow s\sigma \leq t\sigma$)
  - Stable under context ($s \leq t \Rightarrow A[s] \leq A[t]$)
- LPO and RPO work

Extending Ordering to Atoms

- An equation s=t is viewed as a multiset of two elements {s,t} (note: s=t same as t=s)
- Equations compared as multisets
- Terms in disequations slightly bigger than terms in equations
- Example: If a > b > c then
  - $a \neq c \;>\; a = b \;>\; a = c \;>\; b \neq c$

Ground Superposition

$$\frac{\Gamma \vee s=t \qquad L[s] \vee \Delta \qquad s=t \in \max(\Gamma \vee s=t) \qquad L[s] \in \max(L[s] \vee \Delta)}{\Gamma \vee L[t] \vee \Delta \qquad s \in \max(s=t) \qquad s \in \max(L[s])}$$

$$\frac{\Gamma \vee t \neq t \qquad t \neq t \in \max(\Gamma \vee t \neq t)}{\Gamma}$$

Superposition (not Ordered Paramodulation)

- Let $\max(C) = \{L \in C \mid \neg \exists M \in C, M > L\}$
- Definition works for equations too
  - $\max(f(x)=g(y)) = \{f(x),g(y)\}$
- Inferences only need to be applied to a max literal in each clause and a max term in each equation

Superposition

$$\frac{\Gamma \vee s=t \qquad L[s'] \vee \Delta}{(\Gamma \vee L[t] \vee \Delta)\sigma}$$

Restrictions on next slide

$$\frac{\Gamma \vee s \neq t}{\Gamma\sigma}$$

Superposition Restrictions

σ=mgu(s,s')
s' is not a variable
$s\sigma=t\sigma \in \max((\Gamma \vee s=t)\sigma)$
$L[s]\sigma \in \max((L[s] \vee \Delta)\sigma)$
$s\sigma \in \max(s\sigma=t\sigma)$
$s'\sigma \in \max(L[s']\sigma)$

σ=mgu(s,t)
$s\sigma \neq t\sigma \in \max((\Gamma \vee s \neq t)\sigma)$

Counterexample (a > b > c)

1. b=c
2. a=b $\vee$ a=c
3. a $\neq$ b $\vee$ a $\neq$ c

4(2,3,Sup) b $\neq$ b $\vee$ a=c $\vee$ a $\neq$ c

This is a tautology, and we don't want to save tautologies. We must loosen restrictions

Completeness

- Unit Clause (one literal in each clause)
- Horn Clause (at most one positive literal in each clause)
- Superposition + Ordered Equation Resolution + Ordered Resolution + Ordered Factoring is Complete for Unit Clauses and Horn Clauses

Ground Merging Paramodulation

$\Gamma \vee s=t \vee s=u \quad t=v \vee \Delta$ $\quad s=t \in \max(\Gamma \vee s=t)$
---------- $\quad t=v \in \max(t=v \vee \Delta)$
$\Gamma \vee s=v \vee s=u \vee \Delta$ $\quad s \in \max(s=t)$
$t \in \max(\{t,u\})$
$t \in \max(t=v)$

Ground Merging Paramodulation

1. b=c
2. a=b ∨ a=c
3. a ≠ b ∨ a ≠ c

4(1,2,MP) a=c ∨ a=c

Then the proof can be finished

Completeness

- Superposition + Ordered Equation Resolution + Merging Paramodulation + Ordered Resolution + Ordered Factoring is Complete for FOL with equality
- Merging Paramodulation does not apply to Horn Clauses or Unit Clauses

Merging Paramodulation

$$\frac{\Gamma \vee s=t \vee s'=u \qquad t'=v \vee \Delta \qquad (s=t)\sigma \in \max(\Gamma \vee s=t)\sigma}{(t'=v)\sigma \in \max(t=v \vee \Delta)\sigma}$$

$$(\Gamma \vee s=v \vee s=u \vee \Delta)\sigma$$

$s\sigma \in \max(s=t)\sigma$

$t\sigma \in \max(\{t,u\})\sigma$

$t\sigma \in \max(t=v)\sigma$

$\sigma = mgu(t=t', s=s')$

t is not a variable

Ground Equational Factoring

$$\frac{\Gamma \vee s=t \vee s=u \qquad s=t \in \max(\Gamma \vee s=t)}{\Gamma \vee t \neq u \vee s=u}$$

$s \in \max(s=t)$

$t \in \max(\{t,u\})$

Ground Equational Factoring

1. b=c
2. a=b ∨ a=c
3. a ≠ b ∨ a ≠ c

4(2,EqFac) b ≠ c ∨ a=c

Then the proof can be finished

Completeness

- Superposition + Ordered Equation Resolution + Equational Factoring + Ordered Resolution + Ordered Factoring is Complete for FOL with equality
- Equational Factoring does not apply to Horn Clauses or Unit Clauses

Equational Factoring

$$\frac{\Gamma \vee s=t \vee s'=u \qquad (s=t)\sigma \in \max(\Gamma \vee s=t)\sigma}{(\Gamma \vee t \neq u \vee s=u)\sigma \qquad s\sigma \in \max(s=t)\sigma,\ t\sigma \in \max(\{t,u\})\sigma,\ \sigma = \mathrm{mgu}(s,s')}$$

Word Problem as Superposition

- Word Problem: $s_1=t_1,\cdots,s_n=t_n \models u=v$?
- Superposition: Is $s_1=t_1,\cdots,s_n=t_n,u \neq v$ unsat?
- Superposition for unit equalities is called Ordered Completion
- Ordered Completion never fails

Example of Ordered Completion

1. f(x,y) = f(y,x)
2. f(a,b) = c
3. f(b,a) ≠ c

4(1,2,Sup) f(b,a) = c

5(3,4,Sup) c ≠ c

6(5,EqRes) ⊥

Selected Paramodulation

$\Gamma \vee s=t \qquad L[s'] \vee \Delta \qquad s\sigma \in sel((\Gamma \vee s=t)\sigma)$

$s'\sigma \in sel((L[s'] \vee \Delta)\sigma)$

$(\Gamma \vee L[t] \vee \Delta)\sigma$ where σ=mgu(s=s',s=s'')

and s' is not a variable

$\Gamma \vee s \neq t$

Equation Resolution

$\Gamma\sigma$ σ=mgu(s,t) $(s \neq t)\sigma \in sel((\Gamma \vee s \neq t)\sigma)$

Selection Rule

- Select all max literals or one negative literal
- Select all max terms in each selected (dis)equation
- If s=t and s'=u appear in clause and s and s' are unifiable, and s is selected, then t is selected
- Only do inferences involving selected terms
- Selected Resolution is Complete

Notes on Selected Paramodulation

- The Selected Paramodulation rule covers both Superposition and Merging Paramodulation
- It could alternatively be added to Equational Factoring to cover that

Fast Theorem Provers

- The world's fastest theorem provers for FOL with equality: OTTER, SPASS, E, VAMPIRE and WALDMEISTER (which is only for unit clauses) use exactly these techniques
- They also use efficient data structures

What is a propositional model?

- Model M = set of atoms
- Semantics (C is clause, S is set of clauses)
  - $M \models p$ iff $p \in M$
  - $M \models \neg p$ iff $p \notin M$
  - $M \models C$ iff $\exists L \in C, M \models L$
  - $M \models S$ iff $\forall C \in S, M \models C$

Contents

- TP in Propositional Logic
- TP in First Order Logic
- TP in FOL with Equality
- Completeness
- Constrained Deduction
- TP modulo an Equational Theory

Ordering on Clauses

- We have ordering < on literals
- Extend to ordering on clauses by comparing with multiset ordering
  - i.e., C > D if D is the result of removing some literals from C and replacing them with smaller
  - Ex (p > q > r): $\neg p \vee \neg q \vee r > \neg p \vee q \vee \neg r$
    remove $\{\neg q, r\}$ replace with $\{q, \neg r\}$

Creating a model by induction

- Suppose S saturated by Ordered Resolution and does not contain the empty clause
- Suppose $C_1 < \cdots < C_n$ are the clauses in S
- We create a model M for S inductively by using the algorithm on the next slide

Model Construction Theorem

- If S is saturated by Ordered Resolution
- and S does not contain the empty clause
- Then M (created by algorithm) is a model of S

Algorithm for Creating Model

M = ∅
for (i = 1 to n)
  if (not ($M \models C_i$))
    let L = max(C_i)
    if (L is positive and L not duplicate)
      M = M ∪ {L}
return M

MC Example (p<q<r<s)

	M
p ∨ q	q
¬ p ∨ r	
p ∨ ¬ q ∨ r	r
¬ q ∨ ¬ r ∨ ¬ s	

M = {q,r} is a model

Unsaturated Set of Clauses

- When S is not saturated, the algorithm might fail to construct a model
- When it fails to construct a model, the least counterexample indicates a resolution not performed
- We could use model construction to guide which inferences to do

Resolution gives new model

	M
$p \vee q$	q
$p \vee r$	r
$q \vee \neg r$	

M = {q,r} is a model

MC on Unsaturated Set

	M
$p \vee r$	r
$q \vee \neg r$	

M = {r} is not a model

Second clause is smallest counterexample

Resolve on failed literal

Algorithm (reminder)

M = ∅

for (i = 1 to n)

  if (not ($M \models C_i$))

    let L = max(C_i)

    if (L is positive and L not duplicate)

      M = M ∪ {L}

return M

Why would MC fail?

- If it fails, there must exist a clause C st
  - Either max (C) is positive and not added to model because it is a duplicate
  - Or max(C) is negative

Completeness Theorem

- If S is an unsatisfiable set of clauses saturated by Ordered Resolution, then S contains the empty clause
- Contrapositive: If S is saturated and doesn't contain empty clause then S is satisfiable
- Proof: Algorithm always constructs a model for such an S

Proof that M is a model

- If M is not a model, consider smallest counterexample C with L = max(C)
- If L is of form ¬A, then clause that added A to model can be resolved with C, yielding a smaller counterexample (contr.)
- If L is of form A, then A must be duplicated in clause, and factoring gives smaller c.e.

Redundancy

- Recall that a clause it redundant if it is implied by smaller ones
- Therefore a redundant clause could never be the smallest c.e. or produce an atom
- So redundant clauses are not need in completeness proof
- So can be removed

Lifting to FOL

- We construct model from all ground instances of S
- Recall that ordering is well-founded, total on ground atoms, and stable under substitution
- This implies that all inferences can be lifted (i.e., each ground inference is an instance of a nonground inference)

Completeness in FOL

- Lifting argument means model construction works for all ground instances
- So completeness argument is identical
- Only difference is that model cannot be constructed in practice
  - There are infinitely many ground instances
  - It is only theoretical

$S = \{P(a,b) \ , \ P(x,y) \vee Q(x,y) \}$

	M
P(a,b)	P(a,b)
P(b b) ∨ Q(b,b)	Q(b,b)
P(b,a) ∨ Q(b,a)	Q(b,a)
P(a,b) ∨ Q(a,b)	
P(a,a) ∨ Q(a,a)	Q(a,a)

Completeness proof for Completion

- Let E be a set of ground equations saturated by completion. Want to show E is confluent
- Let Red(E) = set of all equations in E whose lhs is irreducible by smaller equations in E
- Then Red(E) is confluent (no overlaps)
- By induction, Red(E) ⊨ E, since any equation in E but not in Red(E) is reducible by E and so implied by smaller members of E (by saturation)
- Therefore E is confluent

Representing Equational Clauses

- View every atom as an equation
  - Add a new (smallest) constant $\top$
  - Then nonequational atom A is viewed as $A=\top$
- Now equality is the only predicate
- Resolution = Paramodulation + Eq. Res.
- Factoring = Equational Factoring + Eq. Res.

Factoring

$$\frac{\Gamma \vee A=\top \vee A=\top}{\Gamma \vee \top\neq\top \vee A=\top}\ \text{EqFac}$$

$$\frac{\Gamma \vee \top\neq\top \vee A=\top}{\Gamma \vee A=\top}\ \text{EqRes}$$

Resolution

$$\frac{\Gamma \vee A=\top \quad A \neq \top \vee \Delta}{\Gamma \vee \top\neq \top \vee \Delta}\ \text{Para}$$

$$\frac{\Gamma \vee \top\neq \top \vee \Delta}{\Gamma \vee \Delta}\ \text{EqRes}$$

Equational Model

- M = confluent set of ground equations
- View equations in M as rewrite rules
- So $M \models s=t$ iff s and t rewrite to the same term
- We will guarantee M is confluent by not allowing any left hand side to be a subterm of another left hand side

MC Algorithm

M = ∅

for (i = 1 to n)

  if (not (M ⊨ C_i))

    let L = max(C_i)

    if (L is positive lhs(L) is not reducible by M and lhs(L) does not appear twice in C_i)

      M = M ∪ {L}

return M

MC continued

	M
a=b	a=b
fb=c	fb=c
fb=a	

M = {a=b,fb=c} is not a model

Third clause is smallest counterexample

Superpose (demodulate) into fb using fb=c

MC on Unsaturated Set(f>a>b>c)

	M
a=b	a=b
fb=a	fb=a
fa=c	

M = {a=b,fb=a} is not a model

Third clause is smallest counterexample

Superpose (demodulate) into fa using a=b

MC continued

	M
a=c	a=c
a=b	
fb=c	fb=c

M = {a=c,fb=c} is not a model

Second clause is smallest counterexample

Superpose (demodulate) into a=b using a=c

MC continued

	M
b=c	b=c
a=c	a=c
fb=c	

M = {b=c,a=c} is not a model

Third clause is smallest counterexample

Superpose (demodulate) into fb=c using b=c

Why would MC fail?

- If it fails, there must exist a clause C containing literal L such that
  - Either L is positive and not added to model because lhs(L) appears twice
  - Or L is positive and not added to model because lhs(L) is reducible by L
  - Or max(C) is negative

MC continued

	M
b=c	b=c
a=c	a=c
fc=c	fc=c

M = {b=c,a=c,fc=c} is a model

Proof that M is a model

- Suppose M is not a model and consider smallest c.e. C as before with L = max(C)
- If L is a positive equation u=v
  - If lhs(L) appears twice in C then EqFac gives smaller c.e.
  - If lhs(L) is reducible by some s=t in M then clause that produced s=t can be superposed into u=v giving smaller c.e. (see next slide)

Proof M is a model (continued)

- If L is disequation $u \neq v$ then $M \models u=v$ so u and v are reducible to same term. Let s=t be the first equation in reduction of u. Then clause that produced s=t can be superposed into u=v giving smaller c.e.

Problem lifting to FOL

- Some ground inferences cannot be lifted to FOL
- The problem is that we do not allow superpositions into variables
- An inference on ground level might be a superposition into a variable when lifted

Completeness and Redundancy

- Completeness argument same as before
- Redundant clauses are not needed in proof as before, so they can be removed

Example where we cannot lift

$R \vee a=b \qquad P(a) \vee Q(a)$

$R \vee P(b) \vee Q(a)$

would need to be instance of

$R \vee a=b \qquad P(x) \vee Q(x)$

$R \vee P(b) \vee Q(x)$

To Solve Lifting Problem

- We only allow a ground instance from S if the substitution is reduced by M
- Then we can use lifting to show that M is a model of reduced instances of S
- That implies M is a model of all instances of S
- So completeness is recovered

Why Constraints

- Constraints can be used to restrict the ground instances of a clause
- This is another way to limit the search space
- We will consider equality and ordering constraints

Contents

- TP in Propositional Logic
- TP in First Order Logic
- TP in FOL with Equality
- Completeness
- Constrained Deduction
- TP modulo an Equational Theory

Constrained Clause

- Equality constraints
  - $P(x,y) \vee Q(x,y) \mid x = a \wedge y = b$
- Ordering constraints
  - $P(x,y) \vee Q(x,y) \mid x > y$
- Constraint is a conjunction of equality and ordering constraints

  $D\sigma$ is an instance of $D \mid c$ if $c\sigma$ is true

Basic Paramodulation

$\Gamma \vee s=t \mid c_1 \qquad L[s'] \vee \Delta \mid c_2$

Paramodulation

$\Gamma \vee L[t] \vee \Delta \mid s=s' \wedge c_1 \wedge c_2$

s' is not a variable

$\Gamma \vee s \neq t \mid c$

Equation Resolution

$\Gamma \mid s=t \wedge c$

Without Basic Paramodulation

$P(x) \vee Q(fx) \qquad fa=b$

$P(a) \vee Q(b) \qquad a=c$

$P(c) \vee Q(b)$

Completeness of BP

- Basic Paramodulation is complete if initial clauses have constraint $\top$
- Clause with unsat. constraint is redundant
- The restriction of no inferences into variable positions becomes stronger
- All the ordering conditions are the same
- Redundancy becomes weaker (you will see)

With Basic Paramodulation

$P(x) \vee Q(fx) \qquad fa=b$

$P(x) \vee Q(b) \mid x=a$

Now inference with a=c is not allowed, because it is at a variable position

Note:$P(c) \vee Q(b)$ can be gotten another way

BP with marked terms

- Can also represent equality constraints with marked terms, then no inferences into marked terms
  - i.e., mgu is applied but term is marked
- This is a simple way to implement it in a standard theorem prover

Redundancy Counterexample

1. $\neg P(x,y) \vee P(x,b)$
2. $\neg P(a,b)$
3. $a=c$
4. $P(c,b)$

5(1,2,Res) $\neg P(a,y)$

5 subsumes 2, but then we can't get $\perp$

BP with marked terms

$P(x) \vee Q(fx)$ $\quad fa=b$

$P(a) \vee Q(b)$

Constrained position is marked

No inference into marked position

Reduced Relative to

- $D_1|c_1$ is reduced to $D_2|c_2$ if $c_2\sigma$ is reduced whenever $c_1\sigma$ is $\forall$ reduced σ
- $P(a,y)$ is not reduced relative to $P(a,b)$ because $P(d,b)$ is reduced but $P(a,b)$ is not

New Redundancy notion

- D|c is reduced in S if all instances of D|c are implied by smaller instances of S that are reduced relative to D|c
- For completeness only have to prove for reduced instances. This implies all instances.

Simplification Example

- P(a,b) subsumes P(a,b)
- P(a,b) subsumes P(a,b)
- P(a,b) subsumes P(a,b)
- P(y,b) subsumes P(a,b)
- P(a,b) does not subsume P(a,b)

Practical Redundancy notion

- D|c is redundant in S if D|c is redundant (in the usual way) by clauses in S which are only marked where D|c is marked
  - Applies to all redundancy methods, for example subsumption and simplification

Redundancy decision

- When redundancy is not OK, you may choose to remove marks to make it OK
- This just removes some basicness from the procedure

BP and Selection Rule

Selected positions in "from" clause may be marked in conclusion, because we could have reduced them first

¬ R(y) ∨ fx=x ∨ gy=hy P(ga) ∨ Q(b)

¬ R(a) ∨ fx=x ∨ P(ha) ∨ Q(b)

BP with rhs and lhs selected

- Given fx=gx, fa=b, gc=d
- Saturation adds fc=d, ga=b
- Trying to refute h(fx,gx) ≠ a
- One step gives h(gx,gx) ≠ a, h(b,ga) ≠ a, h(d,gc) ≠ a, h(fc,d) ≠ a, h(fa,b) ≠ a
- Two steps give h(gc,d) ≠ a, h(d,gc) ≠ a, h(b,b) ≠ a, h(d,d) ≠ a
- One fewer unmarked position each step(NP)

Word Problem vs E-unification

- Let E be a set of equations
- Word Problem:
  - Given s and t, does E ⊨ s=t
- E-unification problem:
  - Given s and t, is there θ st E ⊨ sθ=tθ
- If E saturated by Paramodulation with lhs selected then Word Problem Decidable
- If E saturated by Paramodulation with lhs and rhs selected, E-unification problem decidable in NP

Priority on Selection Rules

We can prioritize selected positions

Positions with higher priority marked in conclusion

e.g. if leftmost positions have higher priority

P ∨ a=b f(c,a)=d

P ∨ f(c,b)=d

Ordering Constraints

$P(x,y,z) \vee f(x,y)=f(y,x) \quad Q(f(a,z))$

$P(x,y,z) \vee Q(f(y,x)) \mid f(x,y)=f(a,z) \wedge$
$f(x,y)>f(y,x) \wedge (f(x,y)=f(y,x))>P(x,y,z)$

Same problem with redundancy

Can always remove ordering constraints

Contents

- TP in Propositional Logic
- TP in First Order Logic
- TP in FOL with Equality
- Completeness
- Constrained Deduction
- TP modulo an Equational Theory

E-unification

- Let E be a set of equations
- E-unification problem:
  - Given s and t, is there θ st $E \models s\theta=t\theta$

AC-unification

- AC = { (x+y)+z=x+(y+z) , x+y=y+x }
- Some E-unifiers of s=t
  - $\theta_1 = [x_1 \mapsto y_1, x_2 \mapsto y_2]$
  - $\theta_2 = [x_1 \mapsto y_2, y_1 \mapsto x_2]$
  - $\theta_3 = [x_1 \mapsto y_1+z, y_2 \mapsto x_2+z]$
  - $\theta_4 = [x_1 \mapsto z_1+z_2, x_2 \mapsto z_3+z_4, y_1 \mapsto z_1+z_3, y_2 \mapsto z_2+z_4]$
- Note: there is no mgu for AC theory

Complete Set of Unifiers

- U is a $CSU_E(s,t)$ if
  - All members of U unify s and t, and
  - $\forall$ unifiers θ of s and t, $\exists\ \sigma \in U$ st $\sigma \leq_E \theta$

CSU A-example

- $U = \{\theta_1, \theta_2, \theta_3, \cdots\} = CSU_A(x+a,a+x)$
- $\theta_1 = [x \mapsto a]$
- $\theta_2 = [x \mapsto a+a]$
- $\theta_3 = [x \mapsto (a+a)+a]$
- …
- There is no finite set of A-unifiers
- A is called infinitary

CSU AC-example

- $U = \{\theta_1, \theta_2, \theta_3, \theta_4\} = CSU_{AC}(x_1+x_2,y+a)$

- $\theta_1 = [x_1 \mapsto y, x_2 \mapsto a]$
- $\theta_2 = [x_1 \mapsto a, x_2 \mapsto y]$
- $\theta_3 = [x_1 \mapsto z+a, y \mapsto x_2+z]$
- $\theta_4 = [x_2 \mapsto z+a, y \mapsto x_1+z]$

E-compatible Ordering

- E is compatible with > if $s>t$, $s =_E s'$, $t =_E t'$ $\Rightarrow s'>t'$

- Some theories have no E-compatible order
- E.g., UI = {x+0=x, x+x=0}
- $x =_{UI} x+0 > 0 =_{UI} x+x > x$

Conditions on Equational Theory

- In order for TP modulo E, we need
  - It must be decidable to find CSU
  - E-unif is finitary
  - We have an E-compatible ordering

TP modulo AC

$$\frac{\Gamma \vee s=t \quad L[s'] \vee \Delta}{(\Gamma \vee L[t] \vee \Delta)\sigma} \quad \text{where } \sigma \in CSU_{AC}(s,s')$$

$$\frac{\Gamma \vee s=t}{\Gamma\sigma} \quad \text{where } \sigma \in CSU_{AC}(s,t)$$

TP modulo E

- If E meets conditions on previous slide
- Then TP with E-unification is complete
  - Get CSU instead of mgu, so you have one inference for each member of CSU
- Provided you allow inferences with extensions of clauses

Extension Rules

$$\frac{\Gamma \vee s=t \quad L[s'+t'] \vee \Delta}{(\Gamma \vee L[t+t'] \vee \Delta)\sigma} \quad \text{where } \sigma \in CSU_{AC}(x+s,s'+t')$$

$$\frac{\Gamma \vee s=t \quad s'=t' \vee \Delta}{(\Gamma \vee x+t=x'+t')\sigma} \quad \text{where } \sigma \in CSU_{AC}(x+s,x'+s')$$

AC Example

$a+e=b \quad a+c+e \neq d$

$b+c \neq d$

$a+b=c \quad a+d=e$

$c+d=b+e$

Basic Paramodulation modulo

$\Gamma \vee s=t \mid c_1 \quad L[s'] \vee \Delta \mid c_2$

$\Gamma \vee L[t] \vee \Delta \mid s=s' \wedge c_1 \wedge c_2$

Now constraints are solved modulo E

Advantages of BP modulo

- We don't need to compute CSU, just decide satisfiability
- We can be lazy and not check constraints until empty clause
- Since we don't compute CSU, just one clause created per inference
- Allows infinitary theories

Disadvantages of BP modulo

- Constraints can get huge
- Redundancy is a problem

A-example

$f(a+x,x+a)=h(x,a+a)$ $\quad f(z,z) \neq h(a,z)$

$h(x,a+a) \neq h(a,z) \mid a+x=z \wedge x+a=z$

$\bot \mid a+x=z \wedge x+a=z \wedge x=a \wedge a+a=z$

Satisfiable with $[x \mapsto a,\ z \mapsto a+a]$