Logic and Mechanized Reasoning
SAT Solving Basics

Marijn J.H. Heule

Carnegie Mellon University

Tseitin Transformation

Unit Propagation and Resolution

Pure Literals and Autarkies

Tseitin Transformation

Unit Propagation and Resolution

Pure Literals and Autarkies

Tseitin: Introduction

Recall: converting a propositional formula A into CNF can result in an exponential blowup. How to avoid that?

Idea: focus on converting A into a satisfiability-equivalent CNF formula (instead of logical equivalence)

How: add definitions and replace parts of the formula (can be seen as the reverse of substitution)

Tseitin: Small Example

Consider the formula $\Gamma = p \vee (q \wedge r)$

We can add the definition $d \leftrightarrow (q \wedge r)$

Replacing $(q \wedge r)$ by d results in CNF $p \vee d$

The clauses representing the definition are:

$$(\neg d \vee q) \wedge (\neg d \vee r) \wedge (d \vee \neg q \vee \neg r)$$

An equisatisfiable formula of Γ in CNF is:

$$(p \vee d) \wedge (\neg d \vee q) \wedge (\neg d \vee r) \wedge (d \vee \neg q \vee \neg r)$$

Satisfying the resulting formula satisfies Γ on original variables

Tseitin: A Linear-Size Transformation

Why is the Tseitin transformation interesting?

- Each connective can be replaced by a new definition
- At most a linear number of definitions
- Definitions can be easily converted into clauses
- Easily obtain a satisfying assignment for original formula
- Resulting in an efficient transformation into CNF

Tseitin: Implementation and Optimizations

Implementation:

- Convert the formula first to NNF
- Generate the definitions from left to right

Optimizations:

- Reuse definitions when possible
- Avoid definitions by interpreting an NNF formula as a CNF formula: e.g. $p \vee (q \wedge \neg r) \vee \neg s$
- Mostly one direction of definition is required

Tseitin: Definitions into Clauses

It is easy to turn a definition $d \leftrightarrow \mathrm{DEF}(p_1, \ldots, p_n)$ into clauses

Example

def	Γ_d	$\Gamma_{\neg d}$
$\mathrm{AND}(p_1, \ldots, p_n)$	$(d \vee \neg p_1 \vee \cdots \vee \neg p_n)$	$(\neg d \vee p_1), \ldots, (\neg d \vee p_n)$
$\mathrm{OR}(p_1, \ldots, p_n)$	$(d \vee \neg p_1), \ldots, (d \vee \neg p_n)$	$(\neg d \vee p_1 \vee \cdots \vee p_n)$
$\mathrm{ITE}(c, t, f)$	$(d \vee \neg c \vee \neg t), (d \vee c \vee \neg f)$	$(\neg d \vee \neg c \vee t), (\neg d \vee c \vee f)$

Tseitin: Larger Example without Optimization

Consider the formula $\Gamma = \neg(p \wedge q \leftrightarrow r) \wedge (s \rightarrow (p \wedge t))$

Convert into NNF:

$$\big((p \wedge q \wedge \neg r) \vee (r \wedge (\neg p \vee \neg q))\big) \wedge (\neg s \vee (p \wedge t))$$

Which results in the following definitions:

- $d_0 \leftrightarrow p \wedge q$
- $d_1 \leftrightarrow d_0 \wedge \neg r$
- $d_2 \leftrightarrow \neg p \vee \neg q$
- $d_3 \leftrightarrow r \wedge d_2$
- $d_4 \leftrightarrow d_1 \vee d_3$
- $d_5 \leftrightarrow p \wedge t$
- $d_6 \leftrightarrow r \vee d_5$
- $d_7 \leftrightarrow d_4 \wedge d_6$

Tseitin: Larger Example with Optimization

Consider the formula $\Gamma = \neg(p \wedge q \leftrightarrow r) \wedge (s \rightarrow (p \wedge t))$

Convert into NNF and interpret as CNF:

$$\big((p \wedge q \wedge \neg r) \vee (r \wedge (\neg p \vee \neg q))\big) \wedge (\neg s \vee (p \wedge t))$$

Which results in the following definitions:

- $d_0 \leftrightarrow p \wedge q$
- $d_1 \leftrightarrow d_0 \wedge \neg r$
- $d_2 \leftrightarrow \neg p \vee \neg q$
- $d_3 \leftrightarrow r \wedge d_2$
- $d_4 \leftrightarrow p \wedge t$

Final result: $(d_1 \vee d_3) \wedge (\neg s \vee d_4)$ plus definition clauses

Tseitin: Plaisted-Greenbaum Encoding

In most cases only one direction of the definition is required.

Example

Recall the formula $\Gamma = p \vee (q \wedge r)$

The Tseitin transformation resulted in the CNF:

$$(p \vee d) \wedge (\neg d \vee q) \wedge (\neg d \vee r) \wedge (d \vee \neg q \vee \neg r)$$

Which clause is redundant (not required for equisatisfiability)?

Removing $(d \vee \neg q \vee \neg r)$ reduces $d \leftrightarrow q \wedge r$ to $d \rightarrow q \wedge r$

When starting with NNF, we only need $d \rightarrow \mathrm{DEF}$

Tseitin: Bringing it all Together

Consider the formula $\Gamma = \neg(p \wedge q \leftrightarrow r) \wedge (s \rightarrow (p \wedge t))$

Convert into NNF and interpret as CNF:

$$((p \wedge q \wedge \neg r) \vee (r \wedge (\neg p \vee \neg q))) \wedge (\neg s \vee (p \wedge t))$$

The Tseitin transformation results in the following clauses:

$$(d_3 \vee d_1) \wedge (d_4 \vee \neg s) \wedge (\neg d_0 \vee p) \wedge (\neg d_0 \vee q) \wedge (\neg p \vee \neg q \vee d_0) \wedge$$

$$(\neg d_1 \vee d_0) \wedge (\neg d_1 \vee \neg r) \wedge (\neg d_0 \vee r \vee d_1) \wedge (\neg d_2 \vee \neg p \vee \neg q) \wedge$$

$$(p \vee d_2) \wedge (q \vee d_2) \wedge (\neg d_3 \vee r) \wedge (\neg d_3 \vee d_2) \wedge$$

$$(\neg r \vee \neg d_2 \vee d_3) \wedge (\neg d_4 \vee p) \wedge (\neg d_4 \vee t) \wedge (\neg p \vee \neg t \vee d_4)$$

Plaisted-Greenbaum removed the colored ones ($d_i \leftarrow \mathrm{DEF}$).

Tseitin Transformation

Unit Propagation and Resolution

Pure Literals and Autarkies

Unit Propagation: Introduction

Unit propagation (UP) is the most important SAT solving simplification technique:

- A clause is unit if it has only one literal
- The only way to satisfy it is assigning the literal to $\top$
- Removing falsified literals can produce unit clauses
- Satisfying unit clauses until fixpoint can be expensive

Unit Propagation: Partial Assignments

Evaluation of clauses and formulas can be generalized to partial assignments:

- Only some variables are assigned to $\top$, $\bot$
- For a clause C, $[\![C]\!]_\tau$ removes literals falsified by τ from C
  - $[\![C]\!]_\tau = \top$ if τ satisfies a literal in C
- For a formula Γ, $[\![\Gamma]\!]_\tau$ replaces all clauses $C \in \Gamma$ by $[\![C]\!]_\tau$
  - Clauses satisfied by τ are removed from $[\![\Gamma]\!]_\tau$

Partial assignments are very important in SAT solving

Unit Propagation: Extending the Assignment

Unit propagation makes unit clauses true until fixpoint

Given an assignment τ and a formula Γ, unit propagation extends τ by assigning all unit clauses in $[\![\Gamma]\!]_\tau$ to $\top$.

Two possible fixpoints (termination)

1. $[\![\Gamma]\!]_\tau$ contains a falsified clause ($\bot$)
2. $[\![\Gamma]\!]_\tau$ contains no more unit clauses

Unit propagation can consume 90% of solver runtime

- Data-structures are optimized for unit propagation
- Unit propagation is hard to parallelize

Unit Propagation: Example

$$\Gamma_{\text{unit}} := (\neg p_1 \vee \neg p_3 \vee p_4) \wedge (\neg p_1 \vee \neg p_2 \vee p_3) \wedge$$
$$(\neg p_1 \vee p_2) \wedge (p_1 \vee p_3 \vee p_6) \wedge (\neg p_1 \vee p_4 \vee \neg p_5) \wedge$$
$$(p_1 \vee \neg p_6) \wedge (p_4 \vee p_5 \vee p_6) \wedge (p_5 \vee \neg p_6)$$

$$\tau = \{p_1 = \top,\ p_2 = \top,\ p_3 = \top,\ p_4 = \top\}$$

Unit Propagation: Proposition

Proposition

Unit propagation does not change the number of satisfying assignments

True or false?

Proof.

True. Let formula Γ have a unit clause p. All satisfying assignments of Γ must assign p to $\top$. Hence there cannot be a satisfying assignment with p assigned to $\bot$. □

Unit Propagation: Resolution

The resolution rule allows for a formula containing the clauses $C \vee p$ and $\neg p \vee D$ to be extended by the clause $C \vee D$

$$\frac{C \vee p \qquad \neg p \vee D}{C \vee D}$$

Resolution proofs:

- A *resolution proof* is a sequence $C_1, \ldots, C_m$ of clauses.
- Every clause is either contained in the formula or derived from two earlier clauses via the *resolution rule*.
- C_m is the *empty clause* (containing no literals): $\bot$.
- There exists a resolution proof for every unsatisfiable formula.

Unit Propagation: Resolution Proofs

Example

$\Gamma := (\neg p \lor \neg q \lor r) \land (\neg r) \land (p \lor \neg q) \land (\neg s \lor q) \land (s)$

Resolution proof: $(\neg p \lor \neg q \lor r)$, $(\neg r)$, $(\neg p \lor \neg q)$, $(p \lor \neg q)$, $(\neg q)$, $(\neg s \lor q)$, $(\neg s)$, (s), $\bot$

$$\dfrac{\dfrac{\neg s \lor q \qquad \dfrac{\dfrac{\neg p \lor \neg q \lor r \qquad \neg r}{\neg p \lor \neg q} \qquad p \lor \neg q}{\neg q}}{\neg s} \qquad s}{\bot}$$

Unit Propagation: Relation to Resolution

Let Γ be a formula. A clause C is implied by Γ via unit propagation (UP) if UP on $\Gamma \wedge \neg C$ results in a conflict.

Example

$$\Gamma := (p \vee q \vee \neg r) \wedge (\neg p \vee \neg q \vee r) \wedge (q \vee r \vee \neg s) \wedge (\neg q \vee \neg r \vee s) \wedge (p \vee r \vee s) \wedge (\neg p \vee \neg r \vee \neg s)$$

clause	$(p \vee q)$	$(p \vee q \vee \neg r)$	$(q \vee r \vee \neg s)$	$(p \vee r \vee s)$
units	$\neg p \wedge \neg q$	$\neg r$	$\neg s$	$\bot$

$$\cfrac{\cfrac{(p \vee r \vee s) \qquad (q \vee r \vee \neg s)}{(p \vee q \vee r)} \qquad (p \vee q \vee \neg r)}{(p \vee q)}$$

Tseitin Transformation

Unit Propagation and Resolution

Pure Literals and Autarkies

Autarkies: Pure Literal Rule

A literal l is pure in a CNF formula Γ if the literal $\neg l$ does not occur in Γ.

The pure literal rule simplifies a formula by making pure literals true.

Example

Consider the formula $\Gamma = (p \vee \neg q) \wedge (q \vee \neg r) \wedge (\neg q \vee r)$.
The literal p is pure in Γ.
Let $\tau(p) = \top$. The pure literal rule will reduce Γ to $[\![\Gamma]\!]_\tau$.
In other words, it will remove the first clause.

Autarkies: Proposition

Proposition

Assigning a pure literal to $\top$ does not change the number of satisfying assignments

True or false?

Proof.

False. A counterexample:
$\Gamma = (p \vee \neg q) \wedge (q \vee \neg r) \wedge (\neg q \vee r)$ has three satisfying assignments, while $[\![\Gamma]\!]_\tau$ with $\tau(p) = \top$ has only two. □

Autarkies: Definition

An autarky is a partial assignment that satisfies all clauses that are “touched” by the assignment:

- a pure literal is an autarky
- a satisfying assignment is an autarky
- “interesting” autarkies are between pure literals and satisfying assignments
- removing clauses that are satisfied by an autarky results in an equisatisfiable formula
- observe that for an autarky τ it holds that $[\![\Gamma]\!]_\tau \subseteq \Gamma$

Autarkies: Example

$$\Gamma_{\text{unit}} := (\neg p_1 \lor \neg p_3 \lor p_4) \land (\neg p_1 \lor \neg p_2 \lor p_3) \land$$
$$(\neg p_1 \lor p_2) \land (p_1 \lor p_3 \lor p_6) \land (\neg p_1 \lor p_4 \lor \neg p_5) \land$$
$$(p_1 \lor \neg p_6) \land (p_4 \lor p_5 \lor p_6) \land (p_5 \lor \neg p_6)$$

$\tau = \{p_1 = \top,\ p_2 = \top,\ p_3 = \top,\ p_4 = \top\}$

The extended τ is an autarky for Γ_{unit}

Autarkies: Theorem

Theorem (Monien and Speckenmeyer, 1985)

Let τ *be an autarky for formula* Γ*. Then* Γ *and* $[\![\Gamma]\!]_\tau$ *are equisatisfiable.*

Proof.

If Γ is satisfiable, then since $[\![\Gamma]\!]_\tau \subseteq \Gamma$, we know that $[\![\Gamma]\!]_\tau$ is satisfiable as well.

Conversely, suppose $[\![\Gamma]\!]_\tau$ is satisfiable and let τ_1 be an assignment that satisfies $[\![\Gamma]\!]_\tau$. We can assume that τ_1 only assigns values to the variables of $[\![\Gamma]\!]_\tau$, which are distinct from the variables of τ. Then the assignment τ_2 which is the union of τ and τ_1 satisfies Γ. □