Properties of the Fourier-Motzkin Procedure

- Any ground set N of linear arithmetic atoms can be easily decided.
- $\mathrm{FM}(N)$ terminates on any N as in recursive calls N has strictly less variables.
- The set $N' \cup N^*$ is worst case of size $O(|N|^2)$.
- $\mathrm{FM}(N)$=true iff N is satisfiable in $\mathcal{A}_{\mathrm{LA}}$.
- The procedure was invented by Fourier (1826), forgotten, and then rediscovered by Dines (1919) and Motzkin (1936).
- There are more efficient methods known, e.g., the simplex algorithm.

2.3 The DPLL(T) Procedure

Goal:
Given a propositional formula in CNF (or alternatively, a finite set N of clauses), where the atoms represent ground formulas over some theory T, check whether it is satisfiable in T. (and optionally: output *one* solution, if it is satisfiable).

Assumption:
Again, clauses contain neither duplicated literals nor complementary literals.

Remark:
We will use LA as an ongoing example for T and consider DPLL(LA).

On LA as a Theory

We consider a specific formula language together with a satisfiability check for conjunctions of atoms (literals) as a theory T. Note that a valuation M is interpreted as the conjunction of its literals.

Later on we will introduce theory notions based on sets of formulas or models.

For LA we consider the language defined before and Fourier-Motzkin as the satisfiability check for conjunctions of atoms. Variables in formulas without quantification can actually be considered as constants.

Notions with Respect to the Theory T

If a partial valuation M is T-consistent (satisfiable) and F a formula such that $M \models_T F$, then we say that *M is a T-model of F*.

If F and G are formulas then F *entails* G *in* T, written $F \models_T G$ if $F \wedge \neg G$ is T-inconsistent.

Example: $x > 1 \not\models x > 0$ but $x > 1 \models_{\mathrm{LA}} x > 0$

Remark

M stands again for a list of propositional literals. As every propositional literal stands for a ground literal from T, there are actually two interpretations of M. We write $M \models F$ if F is entailed by M propositionally. We write $M \models_T F$ if the T ground formulas represented by M entail F.

DPLL(T) Rules from DPLL

Unit Propagate:

$$M \parallel N \cup \{C \vee L\} \;\Rightarrow_{\mathrm{DPLL(T)}}\; M\, L \parallel N \cup \{C \vee L\}$$

if C is false under M and L is undefined under M.

Decide:

$$M \parallel N \;\Rightarrow_{\mathrm{DPLL(T)}}\; M\, L^{\mathrm{d}} \parallel N$$

if L is undefined under M.

Fail:

$$M \parallel N \cup \{C\} \;\Rightarrow_{\mathrm{DPLL(T)}}\; \mathit{fail}$$

if C is false under M and M contains no decision literals.

Specific DPLL(T) Rules

T-Backjump:

$$M\, L^{\mathrm{d}}\, M' \parallel N \cup \{C\} \;\Rightarrow_{\mathrm{DPLL(T)}}\; M\, L' \parallel N \cup \{C\}$$

if $M\, L^{\mathrm{d}}\, M' \models \neg C$
there is some "backjump clause" $C' \vee L'$ such that
$N \cup \{C\} \models_T C' \vee L'$ and $M \models \neg C'$
L' is undefined under M, and
L' or $\overline{L'}$ occurs in N or in $M\, L^{\mathrm{d}}\, M'$.

T-Learn:

$$M \parallel N \;\Rightarrow_{\mathrm{DPLL(T)}}\; M \parallel N \cup \{C\}$$

if $N \models_T C$ and each atom of C occurs in N or M.

T-Forget:

$$M \parallel N \cup \{C\} \;\Rightarrow_{\mathrm{DPLL(T)}}\; M \parallel N$$

if $N \models_T C$.

T-Propagate:

$M \parallel N \ \Rightarrow_{\mathrm{DPLL(T)}} \ M\,L \parallel N$

if $M \models_T L$ where L is undefined in M and
L or $\overline{L}$ occurs in N.

DPLL(T) Properties

The DPLL modulo theories system DPLL(T) consists of the rules Decide, Fail, Unit-Propagate, T-Propagate, T-Backjump, T-Learn and T-Forget.

The Lemma 1.14 and the Lemma 1.15 from DPLL hold accordingly for DPLL(T). Again we will reconsider termination when the needed notions on orderings are established.

Lemma 2.2 *If $\emptyset \parallel N \ \Rightarrow^*_{\mathrm{DPLL(T)}} \ M \parallel N'$ and there is some conflicting clause in $M \parallel N'$, that is, $M \models \neg C$ for some clause C in N, then either Fail or T-Backjump applies to $M \parallel N'$.*

Proof. As in Lemma 1.16. □

Lemma 2.3 *If $\emptyset \parallel N \ \Rightarrow^*_{\mathrm{DPLL(T)}} \ M \parallel N'$ and M is T-inconsistent, then either there is a conflicting clause in $M \parallel N'$, or else T-Learn applies to $M \parallel N'$, generating a conflicting clause. (Proof follows)*

Proof. If M is T-inconsistent, then there exists a subsequence $(L_1, \dots, L_n)$ of M such that $\emptyset \models_T \overline{L_1} \vee \ldots \vee \overline{L_n}$. Hence the conflicting clause $\overline{L_1} \vee \ldots \vee \overline{L_n}$ is either in $M \parallel N'$, or else it can be learned by one T-Learn step. □

Theorem 2.4 *Consider a derivation $\emptyset \parallel N \ \Rightarrow^*_{\mathrm{DPLL(T)}} \ S$, where no more rule of the DPLL(T) procedure is applicable to S except T-Learn or T-forget, and if S has the form $M \parallel N'$ then M is T-consistent. Then*

(1) S is fail iff N is T-unsatisfiable.

(2) If S has the form $M \parallel N'$, then M is a T-model of N.