Master Sciences Informatiques

Solvers Principles and Architectures (SPA)

Answers Final Exam, Fall 2017

Khalil Ghorbal

1 SAT/SMT Solvers

A. CNF vs DNF

We have seen that converting any boolean (well formed) formula to an equivalent conjunctive normal form (CNF) increases linearly the number of logical connectives using Tseytin transformations.

1. Is this possible for disjunctive normal forms (DNF)?
2. Explain the reason for this asymmetry.

Answer. (Easy)

For tautological equivalence, we have seen that the number of logical connectives grows in general exponentially when transforming a formula into both a CNF or a DNF. The canonical examples being the "product of sums" for DNF and the "sum of products" for CNF (cf. lecture 1, slide 17 for CNF). For instance, the following wff in CNF

$$\psi = (x_1 \vee y_1) \wedge \cdots \wedge (x_n \vee y_n),$$

has $2n$ variables, n $(\vee)$ connectives, and $n-1$ $(\wedge)$. Its DNF has 2^n clauses, each of which with n literals. Observe here the parallel with distributing the polynomial $\Pi_{i=1}^{n}(x_i + y_i)$. The result, seen as a "sum of products" has 2^n monomials each of which has degree n.

So as far as logical equivalence is concerned, the size of both normal formalea can grow exponentially. However, we observed that we can weaken the notion of tautological equivalence and consider instead *equisatisfiability* for which any wff can be put in CNF with a linear growth in size using Tseytin transformation. The main trick for those transformations is the use of extra variables to label disjunctions and then append the definition of those extra variables by means of equivalences (which are themselves CNF) thereby controlling the size of the equi-sat formula. Converting a wff in an equi-sat DNF using the same tricks suffers the same exponential growth as one ends up appending a formula like ψ above to the original wff.

B. Resolution Rule

The resolution rule allows to eliminate via equivalent satisfiability a variable that appears both positively and negatively in different clauses. Assuming infinite memory, if one is to apply the original Davis/Putnam (DP) method to a given CNF formula till saturation (that is till reaching a fixed point).

1. What are the possible results of the algorithm and why?
2. If the formula is UNSAT, how can we extract a certificate of unsatisfiability?
3. How does such certificate relate to Craig interpolants?

Answer. (Difficult)

1. A fixed point is reached whenever no variable appears both negatively and positively in two distinct clauses. This in particular means that a fixed point has two forms. The first is a unique clause (a disjunction of literals), encoding all possible solutions of the original problem. The

second is the empty set, proving that the formula is unsatisfiable.

2. Although possibly suboptimal (in the sense of too detailed/refined), the tree of the successive resolutions (together with the unit and forward propagations) is an unsat certificate as it gives an UNSAT equi-sat formula of the original problem.

3. Firstly, Craig's interpolants serve the same purpose, that is they provide an UNSAT certificate. Secondly, they can be extracted from the certificate one extracts by solely relying on the resolution rule by clustering together several small steps (which is a form of abstraction). For this reason, such interpolants are not canonical nor unique in general. Lastly, although CDCL re-introduces the use of the resolution rule via the learning clauses, one almost never have access to the refined certificate obtained by the sole use of the resolution rule: Craig's interpolants give almost always a sound abstraction of such certificates.

C. CDCL

Conflict-Driven Clause Learning (CDCL) allows to prune the search tree built by adding new clauses (tautologically) implied by the original formula. Consider the following CNF formula

$$\begin{aligned}\phi &= c_1 \wedge c_2 \wedge c_3 \wedge c_4 \wedge c_5 \wedge c_6 \\ &= (x_5 \vee x_6) \wedge (x_1 \vee x_8 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (x_9 \vee \neg x_4 \vee \neg x_6)\end{aligned}$$

Assume the following decision assignments have been already made $x_9 = 0@2$ and $x_8 = 0@3$ and that the current decision assignment is $x_1 = 0@5$ (where the notation $x = b@n$ means that the variable x is assigned the value b at depth n).

1. Build the resulting implication graph.
2. Suggest a clause to learn from the observed conflict.
3. Prove that augmenting ϕ with such a formula is SAT equivalent to the original problem ϕ.
4. Based on the learned clause, at what depth would you backtrack?

Answer. (Easy)

We can learn several clauses that lead to the conflict on x_5. Graphically, if we cut the graph in a way that isolates the sources (the previous decisions) from the conflict, then the negation of the conjunctions of the assigned variables that lead to the conflict is the learned clause.

If one is to cut right before the conflict, the learned clause would be c_1, which is trivially implied by ϕ. On the other hand, if we cut right after the sources, the learned clause would involve all decision variables, namely $cl_1 := x_1 \vee x_8 \vee x_9$. Indeed, the implication graph tells us that $\phi \wedge \neg cl_1$ is UNSAT because if cl_1 satisfied then ϕ reduces to false by unit propagation. Thus $\phi \models cl_1$ (tautological implications can be formulated as (un)satisfiability problems). So cl_1, as a logical implication of ϕ, can be safely added to ϕ and the SAT-equivalence of ϕ and $\phi \wedge cl_1$ becomes trivial. There are several other clauses one could also learn. For instance $cl_2 := \neg x_4 \vee x_6$, since appending its negation to ϕ leads to a conflict (using the clauses c_1 and c_5). Other clauses that could be learned from the same implication graph are $cl_3 := \neg x_4 \vee x_9$ and $cl_4 := x_2 \vee x_3 \vee x_9$.

The interest of making explicit the implied clauses stems from their ability to avoid reaching the same conflict (on x_5 in our case). For instance, by adding cl_3, we prevent setting $x_4 = 1$ given the previous decision $x_9 = 0$ because we know that would lead to the same conflict on x_5. So by adding such clause to the original problem, with respect to the same decision for x_9, x_4 will be forced to 0.

There is non canonical choice for the backtracking depth. It is a heuristic of the solvers. Ideally one would want to backtrack to a decision that immediately exploits the learned clause to immediately fix an additional variable. In our example, by learning cl_3 and backtracking to depth 2, x_4 will be set to 0 by unit propagation. In fact, the choice of cl_3 and depth 2 correspond to the *first Unit Implication Point* strategy used in Chaff for instance.

D. SAT Reduction

The multiprocessing scheduling problem asks the following question. Given a finite set A of tasks, a measure (or time length) $\ell(a) : A \mapsto \mathbb{N}$ for each task $a \in A$, a number m of processors and a deadline

$D \in \mathbb{N}$, is there a partition $A = A_1 \cup A_2 \cup \cdots \cup A_m$ of A into m disjoint sets such that

$$\max_{1\le i\le m} \left\{ \sum_{a\in A_i} \ell(a) \right\} \le D \quad ?$$

Prove that the multiprocessing scheduling problem is NP-complete.

Answer. (Easy/Moderate)

The problem is in NP since, given a partition, one can check the inequality by computing the max over i. To prove NP-completeness, we have to reduce a known NP-complete problem to the multiprocessing scheduling problem, that is by encoding an instance of the known NP-complete problem as an instance of this problem in polynomial time.

One way to go is to encode the partition problem, which is NP-complete, into the multiprocessing scheduling problem. Given a finite set A and a nonnegative measure s on A (that is to say a function $s : A \mapsto \mathbb{N}$), the partition problem asks whether there exists a subset A' of A such that

$$\sum_{a\in A'} s(a) = \sum_{a\in A\setminus A'} s(a).$$

The partition problem is a particular instance of the multiprocessing scheduling problem with:

$$D = \frac{1}{2}\sum_{a\in A} s(a), \qquad m = 2, \qquad s = \ell.$$

If there exists a partition of A into A_1 and A_2 such that the inequality holds and suppose without loss of generality that the total weight over A_2 is greater than or equal the total weight over A_1, that is

$$\sum_{a\in A_1} s(a) \le \sum_{a\in A_2} s(a),$$

then

$$\max_{1\le i\le 2} \left\{ \sum_{a\in A_i} s(a) \right\} = \sum_{a\in A_2} s(a) \le D = \frac{1}{2}\sum_{a\in A} s(a) = \frac{1}{2}\sum_{a\in A_1} s(a) + \frac{1}{2}\sum_{a\in A_2} s(a)$$

which implies

$$\frac{1}{2}\sum_{a\in A_2} s(a) \le \frac{1}{2}\sum_{a\in A_1} s(a)$$

Therefore

$$\sum_{a\in A_1} s(a) = \sum_{a\in A_2} s(a) = D.$$

2 Convex Optimization

A. Linear Programming

The diet problem can be stated as follows: choose quantities $x_1, \ldots, x_n$ of n foods to find the cheapest healthy diet such that ($\imath$) one unit of food j costs c_j and contains amount a_{ij} of nutrient i, and ($\imath\imath$) a healthy diet requires nutrient i in quantity at least b_i.

1. Formulate the problem as a Linear Program (LP).
2. What is the interpretation (meaning) of the dual variables in this case? (write down the dual problem and comment.).

Answer. (Moderate)

the cost of the diet is straightforward: $\sum_j c_j x_j$, or using the scalar product notation $c \cdot x$. The nutrient i is provided by the different x_j with a_{ij}, giving a total quantity of $\sum_{j=1}^{n} a_{ij} x_j$. For the diet to be healthy, such sum has to be at least equal to b_i. Denoting by A the matrix having a_{ij} as its components, one obtains the feasible space $Ax \geq b$, leading to the following LP:

$$\begin{aligned} \min \quad & c \cdot x && (1) \\ s.t. \quad & Ax \geq b \quad (\text{where } x \geq 0) && (2) \end{aligned}$$

The Lagrangian $L(x, \lambda) = c \cdot x - \lambda \cdot (Ax - b)$, where the vector λ is nonnegative. Thus

$$\sup_{\lambda} L(x, \lambda) = \begin{cases} c \cdot x & \text{if } Ax \geq b \\ +\infty & \text{otherwise} \end{cases}$$

Therefore the objective value for our problem is reached by minimizing the above supremum. Recall that the weak duality gives:

$$\inf_{x} \sup_{\lambda} L(x, \lambda) \leq \sup_{\lambda} \inf_{x} L(x, \lambda)$$

and that we can rewrite the Lagrangian as follows $L(x, \lambda) = b \cdot \lambda + x \cdot (c - A^t \lambda)$, which then lead to

$$\inf_{x} L(x, \lambda) = \begin{cases} b \cdot \lambda & \text{if } A^t \lambda \leq c \\ -\infty & \text{otherwise} \end{cases}$$

and the following dual optimization problem

$$\begin{aligned} \max \quad & b \cdot \lambda && (3) \\ s.t. \quad & A^t \lambda \leq c \quad (\text{where } \lambda \geq 0) && (4) \end{aligned}$$

The primal formalizes the problem by attempting to minimize the *cost* of the diet while maintaining a minimal nutritive supply for it to be healthy. The dual problem formalizes it by maximizing the *profit*, or in our context, the benefits of the different nutriments of the diet, subject to a fixed unit cost.

Said differently, for a given task that you want to perform (for various reasons: benefits, profits, etc.) but that costs you (money, time, energy, etc.), either you minimize your total cost while ensuring that your task is actually still performed, or you maximize your profit/benefit from it while accounting for the unavoidable unit costs for the job to be done. That's in words the essence of duality.

B. Simplex vs interior point methods

We want to solve the following problem, where f is twice continuously differentiable and assuming the primal objective value is finite and attained:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & Ax = b, \quad (\operatorname{rank} A = p) \end{aligned}$$

1. Using the KKT conditions, deduce the optimality conditions on $x^\star$ and $\nu^\star$ (the Lagrange multiplier vector).
2. Suppose $\hat{x}$ is a solution for $Ax = b$, eliminate the equality constraint from the above problem.
3. Let $z^\star$ denote the optimal vector of this new unconstrained problem, deduce how $x^\star$ and $\nu^\star$ from $z^\star$.
4. Suppose f is linear, explain briefly the descent method used by the simplex algorithm with respect to the unconstrained problem and contrast it with the descent methods used in the interior point methods.

Answer. (Moderate)

1. In this case, the complementarity conditions do not play any role. Recall that the Lagrangian for this problem is $L(x, \nu) = f(x) + \nu \cdot (Ax - b)$, where in this case ν is not constrained to the nonnegative orthant. The KKT conditions translate to the following conditions on $x^\star$ and $\nu^\star$:

$$Ax^\star - b = 0 \quad \text{and} \quad (\nabla_x L)(x^\star, \nu^\star) = (\nabla f)(x^\star) + A^t \nu^\star = 0$$

2. By selecting a basis for the matrix A and decomposing A and x accordingly, the condition $Ax = b$ translates to $A_B x_B + A_N x_N = b$ or equivalently $x_B = A_B^{-1}(b - A_N x_N)$. Thus, the objective

function f can be rewritten as

$$f(x) = f((x_B, x_N)) = f((A_B^{-1}(b - A_N x_N), x_N))$$

where only the (sub)vector x_N acts as a variable. The condition $Ax = b$ is enforced by linking x_B and x_N, so that the optimization problem becomes *unconstrained* (for our choice of the basis B):

$$\min_{x_N} f((A_B^{-1}(b - A_N x_N), x_N)), \quad (x_N \in R^{|N|}).$$

3. If $z^\star$ is the optimal value for x_N, then $x_B^\star = A_B^{-1}(b - A_N z^\star)$, and thus $x^\star = (x_B\star, z^\star)$. By (1.), we deduce $\nu^\star$ from $x^\star$.

4. For a given basis B, the Simplex method computes $z^\star$, then deduces $x^\star$ as explained above. When f is linear the KKT condition for $\nu^\star$ (see 1.) gives $c + A^t \nu^\star = 0$, which is potentially an overdetermined system since $p \leq n$ and one has n equations for p unknowns ($|\nu| = p$). This leads to

$$\begin{pmatrix} A_B^t \\ A_N^t \end{pmatrix} \nu^\star = \begin{pmatrix} -c_B \\ -c_N \end{pmatrix} \implies A_N^t((A_B^t)^{-1} c_B) = c_N.$$

So the Simplex method will update iteratively over the vertices of the polyhedron (or likewise bases of the matrix A) till finding a basis that satisfies the above condition. It will do so by minimizing the reduced cost.

In contrast, the interior point method will go attempt to reach the optimum basis by traveling through the polyhedron (by traversing its interior). The different updates will be guided by the gradient of the objective function.

C. Duality for non convex problems

The two-way partitioning problem is stated as follows ($x \cdot y$ denotes the usual scalar product, W is a square matrix):

$$\begin{array}{ll} \text{min} & x \cdot Wx \\ \text{s.t.} & x_i^2 = 1, \quad i = 1, \ldots, n \end{array}$$

1. Is this a convex problem (explain)?
2. Compute its Lagrangian and state its dual problem while classifying it (QP, LP, SDP, etc.).
3. Deduce a lower bound for the primal optimal value $p^\star$.

Answer. (Difficult)

In this problem, the components of the vector x are not even continuous since each x_i can be either 1 or -1. The feasible set is therefore not a convex set, it is not even continuous.

We can still compute the Lagrangian as usual $L(x, \lambda) = x \cdot Wx + \sum \lambda_i (x_i^2 - 1)$, where λ_i are unconstrained. To obtain the dual problem, it suffices to compute $\inf_x L(x, \lambda)$. First, observe that we rewrite the Lagrangian as $L(x, \lambda) = x \cdot (W + \Lambda)x - \sum_i \lambda_i$, where Λ is the diagonal matrix obtained by stacking the λ_i on its diagonal. Thus

$$\inf_x L(x, \lambda) = \begin{cases} -\sum_i \lambda_i & \text{if } W + \Lambda \succeq 0 \\ -\infty & \text{otherwise} \end{cases}$$

The dual problem is therefore an SDP problem:

$$\max \quad -\sum_i \lambda_i \tag{5}$$

$$s.t. \quad W + \Lambda \succeq 0 \tag{6}$$

A necessary and sufficient condition for the matrix $W + \Lambda$ to be positive semidefinite is to have all its eigenvalues to be nonnegative which can be achieved by fixing all λ_i to be equal to $-\nu_{min}$, where ν_{min} denotes the minimal eigenvalue of W (assuming all are reals). Thus, for any ν_i eigenvalue of

W, $\nu_i - \nu_{min}$ is an eigenvalue of $W + \Lambda = W - \nu_{min} I$ since

$$\det(W + \Lambda - (\nu_i - \nu_{min})I) = \det(W - \nu_{min} I - (\nu_i - \nu_{min})I) = \det(W - \nu_i I) = 0.$$

Therefore, $\nu_i - \nu_{min} \geq 0$ for all i, and $W + \Lambda$ is positive semidefinite. Thus

$$n\nu_{min} \leq d^\star \leq p^\star,$$

where $d^\star$ and $p^\star$ denote respectively the dual and primal objective values. The lower bound $n\nu_{min}$ is non trivial lower bound for the primal non convex problem obtained by solving an SDP problem.

D. "Visualizing" symmetric positive semidefinite matrices

Recall that a symmetric matrix S is positive semidefinite if for all vectors z, the scalar product of z and Sz is nonnegative.

1. Define the set on which vary the components of the matrix S as a quantifier elimination problem.
2. Solve the problem for $n = 1$, n being the dimension of z. (the case $n = 2$ is depicted below)

Answer. (Easy - Bonus)

The positive semi definiteness of the matrix S can be expressed as

$$\forall z_1, \ldots, z_n. \quad z \cdot Sz \geq 0$$

By removing the quantifiers (that is by projecting on the space defined by the components of the matrix S), one obtains a condition on the components of S for the matrix to be positive semidefinite.

When $n = 1$, the matrix S is a scalar, s say., we therefore get

$$\forall z. \quad sz^2 \geq 0$$

which reduces to $s \geq 0$.