One special variable x_0 whose value is fixed to 0 is permitted;
this allows to express atoms like $x < 3$ in the form $x - x_0 < 3$.

Solving difference logic:

Let F be a conjunction in DL.
For simplicity: only non-strict inequalities.

Define a weighted graph G:

Vertices V: Variables in F.

Edges E: $x - y \leq c \ \rightsquigarrow\ $ edge (x, y) with weight c.

Theorem: F is unsatisfiable iff G has a negative cycle.

Can be checked in $O(|V| \cdot |E|)$ using the Bellman-Ford algorithm.

1.9 C-Arithmetic

In languages like C: Bounded integer arithmetic (modulo 2^n), in device drivers also combined with bitwise operations.

Bit-Blasting (encode everything as boolean circuits, use CDCL):

Naive encoding: possible, but often too inefficient.

If combined with over-/underapproximation techniques (Bryant, Kroening, et al.): successful.

1.10 Decision Procedures for Data Structures

There are decision procedures for, e. g.,

Arrays (read, write)

Lists (car, cdr, cons)

Sets or multisets with cardinalities

Bitvectors

Note: There are usually restrictions on quantifications. Unrestricted universal quantification can lead to undecidability.

Literature: Further Decision Procedures

Aaron R. Bradley, Zohar Manna: The Calculus of Computation. Springer, 2007.

Aaron R. Bradley, Zohar Manna, Henny B. Sipma: What's decidable about arrays? Verification, Model Checking, and Abstract Interpretation (VMCAI), LNCS 3855, pp. 427-442, Springer, 2006.

Randal E. Bryant, Daniel Kroening, Joël Ouaknine, Sanjit A. Seshia, Ofer Strichman, Bryan Brady: Deciding bit-vector arithmetic with abstraction. 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'07), LNCS 4424, pp. 358–372, Springer, 2007.

George E. Collins: Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition. 2nd. GI Conf. Automata Theory and Formal Languages, LNCS 33, pp. 134–183, Springer, 1975.

D. C. Cooper: Theorem Proving in Arithmetic Without Multiplication. Machine Intelligence, vol. 7, pp. 91–99. American Elsevier, New York, 1972.

George B. Dantzig: Linear Programming and Extensions. Princeton Univ. Press, 1963.

L. V. Kantorovich: Mathematical Methods in the Organization and Planning of Production. Publication House of the Leningrad State University, 1939. Translated in Management Science, 6:366–422, 1960.

Narendra Karmarkar: A New Polynomial Time Algorithm for Linear Programming. Combinatorica, 4(4):373–395, 1984.

Daniel Kroening, Ofer Strichman: Decision Procedures – An Algorithmic Point of View. Springer, 2008.

Mojżesz Presburger: Über der Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. Comptes Rendus Premier Congrès des Mathématiciens des Pays Slaves, Warsaw, pp. 92-101, 1929.

William Pugh: The Omega Test: a fast and practical integer programming algorithm for dependence analysis. Comm. of the ACM, 35(8):102-114, 1992.

Stefan Ratschan: Approximate Quantified Constraint Solving by Cylindrical Box Decomposition. Reliable Computing, 8(1):21–42, 2002.

Alfred Tarski: A Decision Method for Elementary Algebra and Geometry. Univ. of California Press, Berkeley, 1951.

1.11 Combining Decision Procedures

Problem:

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be first-order theories over the signatures Σ_1 and Σ_2.

Assume that we have decision procedures for the satisfiability of existentially quantified formulas (or the validity of universally quantified formulas) w. r. t. $\mathcal{T}_1$ and $\mathcal{T}_2$.

Can we combine them to get a decision procedure for the satisfiability of existentially quantified formulas w. r. t. $\mathcal{T}_1 \cup \mathcal{T}_2$?

General assumption:

Σ_1 and Σ_2 are disjoint.

The only symbol shared by $\mathcal{T}_1$ and $\mathcal{T}_2$ is built-in equality.

We consider only conjunctions of literals.

For general formulas, convert to DNF first and consider each conjunction individually.

Abstraction

To be able to use the individual decision procedures, we have to transform the original formula in such a way that each atom contains only symbols of one of the signatures (plus variables).

This process is known as *variable abstraction* or *purification*.

We apply the following rule as long as possible:

$$\frac{\exists \vec{x}\, (F[t])}{\exists \vec{x}, y\, (F[y] \wedge t \approx y)}$$

if the top symbol of t belongs to Σ_i and t occurs in F directly below a Σ_j-symbol or in a (positive or negative) equation $s \approx t$ where the top symbol of s belongs to Σ_j $(i \neq j)$, and if y is a new variable.

It is easy to see that the original and the purified formula are equivalent.

Stable Infiniteness

Problem:

Even if the Σ_1-formula F_1 and the Σ_2-formula F_2 do not share any symbols (not even variables), and if F_1 is $\mathcal{T}_1$-satisfiable and F_2 is $\mathcal{T}_2$-satisfiable, we cannot conclude that $F_1 \wedge F_2$ is $(\mathcal{T}_1 \cup \mathcal{T}_2)$-satisfiable.

Example:

Consider

$\mathcal{T}_1 = \{\forall x, y, z\,(x \approx y \;\vee\; x \approx z \;\vee\; y \approx z)\}$

and

$\mathcal{T}_2 = \{\exists x, y, z\,(x \not\approx y \;\wedge\; x \not\approx z \;\wedge\; y \not\approx z)\}$.

All $\mathcal{T}_1$-models have at most two elements, and all $\mathcal{T}_2$-models have at least three elements.

Since $\mathcal{T}_1 \cup \mathcal{T}_2$ is contradictory, there are no $(\mathcal{T}_1 \cup \mathcal{T}_2)$-satisfiable formulas.

To ensure that $\mathcal{T}_1$-models and $\mathcal{T}_2$-models can be combined to $(\mathcal{T}_1 \cup \mathcal{T}_2)$-models, we require that both $\mathcal{T}_1$ and $\mathcal{T}_2$ are stably infinite.

A first-order theory $\mathcal{T}$ is called *stably infinite*, if every existentially quantified formula that has a $\mathcal{T}$-model has also a $\mathcal{T}$-model with a (countably) infinite universe.

Note: By the Löwenheim–Skolem theorem, "countable" is redundant here.

Shared Variables

Even if $\exists\vec{x}\, F_1$ is $\mathcal{T}_1$-satisfiable and $\exists\vec{x}\, F_2$ is $\mathcal{T}_2$-satisfiable, it can happen that $\exists\vec{x}\,(F_1 \wedge F_2)$ is not $(\mathcal{T}_1 \cup \mathcal{T}_2)$-satisfiable, for instance because the shared variables x and y must be equal in all $\mathcal{T}_1$-models of $\exists\vec{x}\, F_1$ and different in all $\mathcal{T}_2$-models of $\exists\vec{x}\, F_2$.

Example:

Consider

$F_1 = (x + (-y) \approx 0)$,

and

$F_2 = (f(x) \not\approx f(y))$

where $\mathcal{T}_1$ is linear rational arithmetic and $\mathcal{T}_2$ is EUF.

We must exchange information about shared variables to detect the contradiction.