

The Reduction Approach to Decision Procedures

Calogero G. Zarba

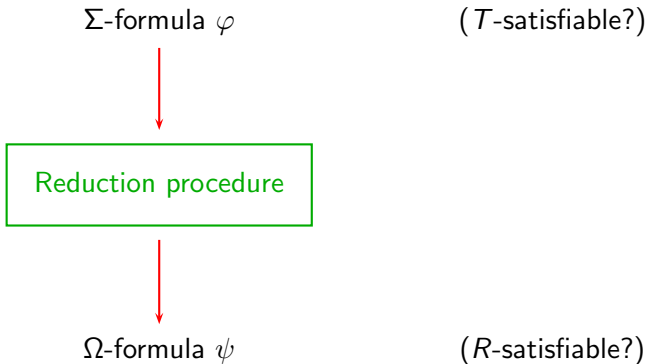
Universität des Saarlandes

Dagstuhl, October 2007

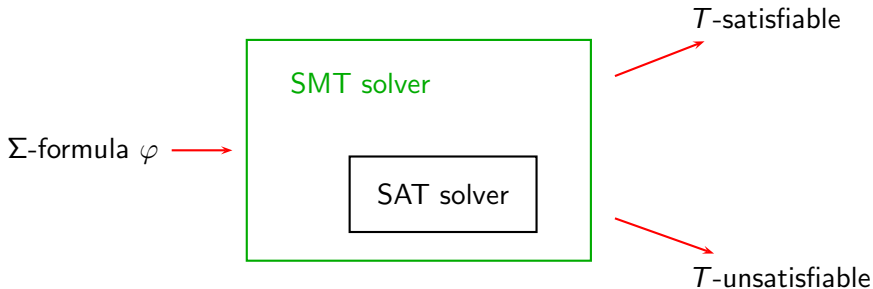
Outline

- ① Reduction and SMT solvers
- ② Reduction from sets to equality
- ③ Combining reduction procedures

Reduction



SMT solvers



Where T is the combination of the following theories:

T_{\approx} equality

T_{bv} fixed-size bit-vectors

T_{int} integer linear arithmetic

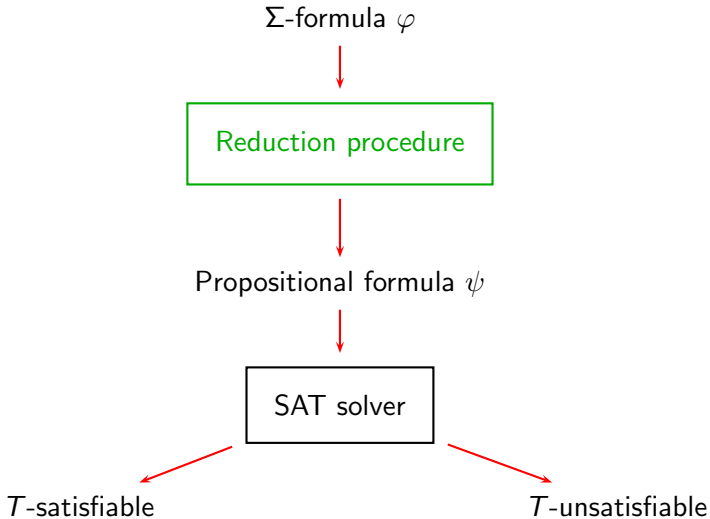
T_{array} arrays

T_{set} sets

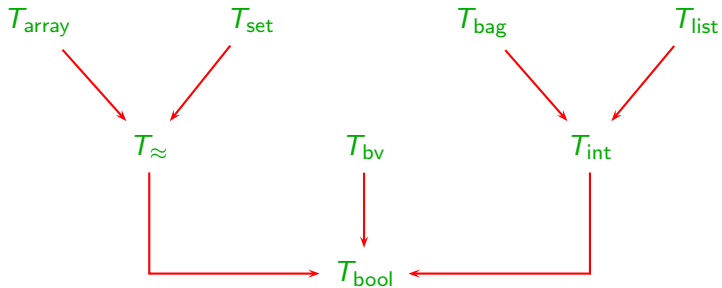
T_{bag} multisets

T_{list} lists

Reduction and SMT solvers



Reduction and SMT solvers



- Each arrow represents a reduction procedure
- These reduction procedures are **combined** in order to obtain a global reduction procedure:

$$T_{\approx} \oplus T_{\text{bv}} \oplus T_{\text{int}} \oplus T_{\text{array}} \oplus T_{\text{set}} \oplus T_{\text{bag}} \oplus T_{\text{list}} \longrightarrow T_{\text{bool}}$$

Outline

① Reduction and SMT solvers

② Reduction from sets to equality

③ Combining reduction procedures

Reduction from sets to equality

$$T_{\text{set}[\alpha]} \longrightarrow T_{\approx}^{\alpha}$$

Step 1: Normalize

Step 2: Introduce new variables

Step 3: Replace

Step 1: Normalize

$\Sigma_{\text{set}[\alpha]}$ -formula φ



Normalization



$$\left\{ \begin{array}{lll} \psi_{\text{prop}} & u \leftrightarrow x \approx_{\text{set}[\alpha]} y & u \leftrightarrow a \approx_{\alpha} b \\ u \leftrightarrow a \in x & x \approx \emptyset & x \approx \{a\} \\ x \approx y \cup z & x \approx y \cap z & x \approx y \setminus z \end{array} \right\}$$

Step 2: Introduce new variables

$$u \leftrightarrow x \approx_{\text{set}[\alpha]} y$$



$$w_{xy} : \alpha$$

Step 3: Replace

$$u \leftrightarrow x \approx_{\text{set}[\alpha]} y \quad \Rightarrow \quad u \leftrightarrow \bigwedge_c (c \in x \leftrightarrow c \in y)$$

$$u \leftrightarrow a \approx_\alpha b$$

$$u \leftrightarrow a \in x$$

$$x \approx \emptyset \quad \Rightarrow \quad \bigwedge_c (c \notin x)$$

$$x \approx \{a\} \quad \Rightarrow \quad \bigwedge_c (c \in x \leftrightarrow c \approx a)$$

$$x \approx y \cup z \quad \Rightarrow \quad \bigwedge_c (c \in x \leftrightarrow c \in y \vee c \in z)$$

$$x \approx y \cap z \quad \Rightarrow \quad \bigwedge_c (c \in x \leftrightarrow c \in y \wedge c \in z)$$

$$x \approx y \setminus z \quad \Rightarrow \quad \bigwedge_c (c \in x \leftrightarrow c \in y \wedge c \notin z)$$

Outline

- ① Reduction and SMT solvers
- ② Reduction from sets to equality
- ③ Combining reduction procedures

Combining reduction procedures

Given

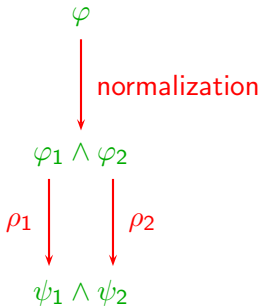
$$\Sigma_1\text{-theory } T_1 \xrightarrow{\rho_1} \Omega_1\text{-theory } R_1$$

$$\Sigma_2\text{-theory } T_2 \xrightarrow{\rho_2} \Omega_2\text{-theory } R_2$$

we want to obtain

$$T_1 \oplus T_2 \xrightarrow{\rho} R_1 \oplus R_2$$

Combining reduction procedures



- Clearly

$$T_1 \oplus T_2 \models \varphi \iff T_1 \oplus T_2 \models \varphi_1 \wedge \varphi_2$$

- Under what conditions we also have the following?

$$T_1 \oplus T_2 \models \varphi_1 \wedge \varphi_2 \iff R_1 \oplus R_2 \models \psi_1 \wedge \psi_2$$

Reduction procedures

Let:

$$\Sigma\text{-theory } T \xrightarrow{\rho} \Omega\text{-theory } R$$

$$\psi = \rho(\varphi)$$

$$X = \text{vars}(\varphi)$$

$$Y = \text{vars}(\psi)$$

Then:

$$\textcircled{1} \mathcal{A} \models_T \varphi \implies \exists \mathcal{B} \models_R \psi \text{ such that } \mathcal{A}^{\Sigma \cap \Omega, X \cap Y} \cong \mathcal{B}^{\Sigma \cap \Omega, X \cap Y}$$

$$\textcircled{2} \mathcal{B} \models_R \psi \implies \exists \mathcal{A} \models_T \varphi \text{ such that } \mathcal{A}^{\Sigma \cap \Omega, X \cap Y} \cong \mathcal{B}^{\Sigma \cap \Omega, X \cap Y}$$

Reduction theorem

Given

$$\Sigma_1\text{-theory } T_1 \xrightarrow{\rho_1} \Omega_1\text{-theory } R_1$$

$$\Sigma_2\text{-theory } T_2 \xrightarrow{\rho_2} \Omega_2\text{-theory } R_2$$

Let $\psi_i = \rho_i(\varphi_i)$, $X_i = \text{vars}(\varphi_i)$, $Y_i = \text{vars}(\psi_i)$

Then

$$T_1 \oplus T_2 \models \varphi_1 \wedge \varphi_2 \iff R_1 \oplus R_2 \models \psi_1 \wedge \psi_2$$

provided that

- 1 $\Sigma_1 \cap \Sigma_2 = \Omega_1 \cap \Omega_2$ (signature condition)
- 2 $X_1 \cap X_2 = Y_1 \cap Y_2$ (variable condition)

Variable condition

Let $\psi_i = \rho_i(\varphi_i)$, $X_i = \text{vars}(\varphi_i)$, $Y_i = \text{vars}(\psi_i)$

W.l.o.g.

- 1 $Y_i \supseteq X_i$
- 2 $Y_i \setminus X_i$ is a set of fresh variables

Therefore

$$(Y_1 \setminus X_1) \cap (Y_2 \setminus X_2) = \emptyset$$

which implies

$$X_1 \cap X_2 = Y_1 \cap Y_2$$

Signature condition

Assume:

- 1 Sort α in T_0
- 2 All sorts in T_0 are simpler than $\text{set}[\alpha]$

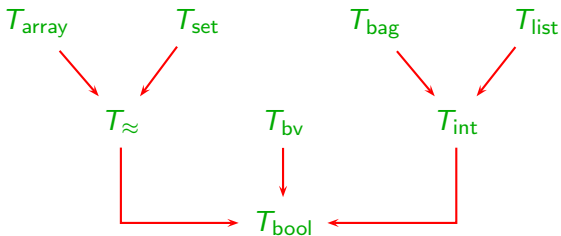
$$\begin{array}{c} T_{\text{set}[\alpha]} \oplus T_0 \\ \downarrow \quad \downarrow \\ \underbrace{T_{\approx}^{\alpha} \oplus T_0}_{T_0} \end{array}$$

Shared signature = $\langle \{\alpha\}, \emptyset, \emptyset, \emptyset \rangle$

Shared signature = $\langle \{\alpha\}, \emptyset, \emptyset, \emptyset \rangle$

Signature condition is satisfied when we apply the reduction procedures in order, from the most complex data type to the simplest.

Global reduction algorithm



- 1 Apply reductions for T_{array} , T_{set} , T_{bag} , and T_{list} in order, from the most complex data type to the simplest
- 2 Apply reduction for T_{\approx} , T_{bv} , and T_{int}
- 3 Convert to CNF (DIMACS format) and send result to a SAT solver

Implementation

- SMT solver **CAISSA**
 - Support for the following theories:

T_{\approx} equality

T_{bv} fixed-size bit-vectors

T_{int} integer linear arithmetic

$T_{array[\alpha,\beta]}$ arrays

$T_{set[\alpha]}$ sets

$T_{bag[\alpha]}$ multisets

$T_{list[\alpha]}$ lists

- <http://react.cs.uni-sb.de/caissa/>
- Next developments:
 - Interpolation à la [KMZ06]
 - Incremental reduction à la DPLL(T)