Constraint Contextual Rewriting

Alessandro Armando

MRG-Lab

DIST, University of Genova

CALCULEMUS Autumn School

Pisa, Sept 30 – Oct 1, 2002

Introduction

- The effective integration of decision procedures in formula simplification is one of the key problems in Automated Reasoning.

- Unfortunately the problem is not easy:
  - it is fairly simple to plug a decision procedure inside a prover,
  - but, to obtain an effective integration can be a challenge.

- Problems occur when the decision procedure is asked to solve goals containing symbols which are **interpreted for the prover** but **uninterpreted for the decision procedure**.

- It is thus often the case that the decision procedure can not solve the problem at hand and therefore it is of no help to the prover.

Boyer & Moore's Augmentation Heuristics

- Boyer & Moore devised and implemented a heuristics, called **augmentation**, which extends the information available to the decision procedure with facts encoding properties of the symbols the decision procedure is not aware of.

- In Boyer & Moore's experience the heuristics is crucial to obtain an effective integration (both in speed and in decreased user interaction).

Example: Augmentation

Let us consider the problem of establishing the unsatisfiability of

$$\{n \geq 4,\ n^2 \geq 2^n,\ n^2 \neq 2^n\} \tag{1}$$

using a decision procedure for total orders

Example: Augmentation

Let us consider the problem of establishing the unsatisfiability of

$$\{n \geq 4,\ n^2 \geq 2^n,\ n^2 \neq 2^n\} \tag{1}$$

using a decision procedure for total orders and the following fact:

$$X \geq 4 \rightarrow X^2 \leq 2^X$$

Example: Augmentation

Let us consider the problem of establishing the unsatisfiability of

$$\{n \geq 4,\ n^2 \geq 2^n,\ n^2 \neq 2^n\} \tag{1}$$

using a decision procedure for total orders and the following fact:

$$X \geq 4 \rightarrow X^2 \leq 2^X$$

We can instatiate X with n and since $n \geq 0$ is in (1) we can extend (1) with $n^2 \leq 2^n$, thereby obtaining

$$\{n \geq 4,\ n^2 \geq 2^n,\ n^2 \neq 2^n, n^2 \leq 2^n\}$$

Example: Augmentation

Let us consider the problem of establishing the unsatisfiability of

$$\{n \geq 4,\ n^2 \geq 2^n,\ n^2 \neq 2^n\} \tag{1}$$

using a decision procedure for total orders and the following fact:

$$X \geq 4 \rightarrow X^2 \leq 2^X$$

We can instatiate X with n and since $n \geq 0$ is in (1) we can extend (1) with $n^2 \leq 2^n$, thereby obtaining

$$\{n \geq 4,\ n^2 \geq 2^n,\ n^2 \neq 2^n, n^2 \leq 2^n\}$$

which is readily found unsatisfiable by the decision procedure.

Boyer & Moore's Augmentation Heuristics (continued)

The problem with the augmentation heuristics is that it *greatly complicates* the integration schema and sophisticated strategies are needed to control the augmentation activity.

- The complexity of the approach makes it **very difficult** any attempt to modify, extend, reuse, and reason about the resulting integration schema.
- Even the proofs of basic and fundamental properties such as soundness and termination can be so difficult to be impractical.
- This situation has probably discouraged many, thereby preventing a wide use of the approach. (NQTHM/ACL2, Tecton, and **RDL** are the only provers based on Boyer & Moore's original ideas.)

Constraint Contextual Rewriting

- Constraint Contextual Rewriting, $\mathrm{CCR}(X)$ for short, is a generalization of (contextual) rewriting that incorporates the functionalities provided by a decision procedure.

- The services of the decision procedure are characterized **abstractly** (i.e. independently from the theory decided by the decision procedure) and the notation $\mathrm{CCR}(X)$ (by analogy with the $\mathrm{CLP}(X)$ notation) is used to stress this fact.

- By using $\mathrm{CCR}(X)$ as a reference model, the problem of the integration of decision procedures in formula simplification is reduced to the implementation of a decision procedure for the fragment of choice.

Constraint Contextual Rewriting: Applications

- A. Armando, L. Compagna, S. Ranise. RDL—Rewrite and Decision procedure Laboratory. In the Proceedings of the International Joint Conference on Automated Reasoning (IJCAR 2001), 2001.

- A. Armando and C. Ballarin. Maple's Evaluation Process as Constraint Contextual Rewriting. In the Proceedings of the Intl. Symposium on Symbolic and Algebraic Computation (ISSAC'2001), 2001.

- A. Armando, M. Rusinowitch e S. Stratulat. Incorporating Decision Procedures in Implicit Induction. In the Journal of Symbolic Computation, 2002.

Roadmap

- ☑ Introduction
- ☐ From Contextual Rewriting to Constraint Contextual Rewriting
- ☐ Constraint Contextual Rewriting
- ☐ Properties of Constraint Contextual Rewriting
- ☐ RDL
- ☐ Maple's evalution process as Constraint Contextual Rewriting

Contextual Rewriting

Extended form of conditional rewriting whereby information contained in the context of the expression being rewritten is used by the rewriting activity.

Let us consider the problem of rewriting the literal p in the clause $\{p\} \cup E$ via a set of conditional rewrite rules, say R.

We call p the focus literal and call the set of the negations of the literals in E the context (denoted by $\overline{E}$).

Key idea: While rewriting the focus literal p it is legal to assume the truth of the literals in the context $\overline{E}$.

Contextual Rewriting: an Example

Let R contain the conditional rule $X \geq 0 \rightarrow \sqrt{X^2} = X$ (1)
and let the clause to be simplified be:

$$\{n \not\geq 0,\ n^2 \neq 2^n,\ \sqrt{2^n} = n\}$$

Contextual Rewriting: an Example

Let R contain the conditional rule $X \geq 0 \rightarrow \sqrt{X^2} = X$ (1)
and let the clause to be simplified be:

$$\{n \not\geq 0,\ n^2 \neq 2^n,\ \sqrt{2^n} = n\}$$

Let $\underbrace{\{n \geq 0,\ n^2 = 2^n\}}_{\text{context}} \qquad \underbrace{\sqrt{2^n} = n}_{\text{focus}}$

Contextual Rewriting: an Example

Let R contain the conditional rule $X \geq 0 \rightarrow \sqrt{X^2} = X$ (1)
and let the clause to be simplified be:

$$\{n \not\geq 0,\ n^2 \neq 2^n,\ \sqrt{2^n} = n\}$$

Let $\underbrace{\{n \geq 0,\ n^2 = 2^n\}}_{\text{context}} \qquad \underbrace{\sqrt{2^n} = n}_{\text{focus}}$

The l.h.s. of (1) does not match with the l.h.s. of the focus.

Contextual Rewriting: an Example

Let R contain the conditional rule $X \geq 0 \rightarrow \sqrt{X^2} = X$ (1)
and let the clause to be simplified be:

$$\{n \ngeq 0,\ n^2 \neq 2^n,\ \sqrt{2^n} = n\}$$

Let $\underbrace{\{n \geq 0,\ n^2 = 2^n\}}_{\text{context}} \qquad \underbrace{\sqrt{2^n} = n}_{\text{focus}}$

The l.h.s. of (1) does not match with the l.h.s. of the focus.

However, $n^2 = 2^n$ allows us to rewrite the focus to $\sqrt{n^2} = n$ and hence enables the application of (1).

Contextual Rewriting: an Example

Let R contain the conditional rule $X \geq 0 \rightarrow \sqrt{X^2} = X$ (1)
and let the clause to be simplified be:

$$\{n \not\geq 0,\ n^2 \neq 2^n,\ \sqrt{2^n} = n\}$$

Let $\underbrace{\{n \geq 0,\ n^2 = 2^n\}}_{\text{context}} \qquad \underbrace{\sqrt{2^n} = n}_{\text{focus}}$

The l.h.s. of (1) does not match with the l.h.s. of the focus.

However, $n^2 = 2^n$ allows us to rewrite the focus to $\sqrt{n^2} = n$ and hence enables the application of (1).

The focus then can be rewritten to an identity provided that condition $n \geq 0$ holds. (This can be easily verified since $n \geq 0$ occurs in the context.)

From Contextual Rewriting. . .

In Contextual Rewriting the context is used both

1. to rewrite the focus literal (aimed at enabling the application of rewrite rules) and

2. to establish the conditions of conditional rewrite rules.

In both cases the kind of reasoning applied amounts to reasoning about the properties of ground equalities.

Contextual rewriting is thus the results of integrating a "decision procedure" for ground equalities with standard conditional rewriting.

. . . to Constraint Contextual Rewriting

- The pattern of interaction between rewriting and the decision procedure does not depend on the theory decided by the decision procedure.

- $\mathrm{CCR}(X)$ is then the result of abstracting contextual rewriting from the theory decided by the decision procedure.

- The traditional notion of contextual rewriting therefore becomes an instance of $\mathrm{CCR}(X)$ whereby X is instantiated by a decision procedure for ground equalities

- New forms of contextual rewriting can be obtained by instantiating X to decision procedures for different decidable theories.

An Example: $\mathbf{CCR}(TO)$

Let R contain the conditional rule $X \geq 0 \rightarrow \sqrt{X^2} = X$ (1)

and let $\overbrace{\{n \geq 4, n^2 \geq 2^n, n^2 \leq 2^n\}}^{\text{context}}$ $\overbrace{\sqrt{2^n} = n}^{\text{focus}}$

As before, the l.h.s. of (1) does not match with the l.h.s. of the focus. However, now the equation $n^2 = 2^n$ is not directly available in the context.

Such an equality is nevertheless entailed by the context and therefore the decision procedure can rewrite the focus to $\sqrt{n^2} = n$.

This step enables the matching of the l.h.s. of (1) and therefore we are left with the problem of establishing $n \geq 0$ which is easily found to be a consequence of the context by the decision procedure.

Remark

(Constraint) Contextual Rewriting differs from conventional rewriting in two essential ways:

1. Proofs do not have a linear structure.
   - This feature (inherited from conditional rewriting) is due to the presence of subsidiary proofs needed to establish the conditions of conditional rewrite rules.
   - This results in a hybrid notion of proof which combines the linear structure of reduction proofs with the tree-structured proofs of sequent calculi.

2. Due to the dependency from the context, rewriting is a ternary relation and not a binary relation as in conventional rewriting.

Contextual Reduction Systems

Let $\mathcal{L}$ be a set of labels. For all $\ell \in \mathcal{L}$:

- let C_ℓ and E_ℓ be sets of expressions and
- let $\mathcal{S}(C_\ell, E_\ell)$ be the set of sequents of the form $c :: e \underset{\ell}{\longrightarrow} e'$ for all $c \in C_\ell$ and $e, e' \in E_\ell$.

A *contextual reduction system* (CRS) is a structure $\langle \{\mathcal{S}(C_\ell, E_\ell)\}_{\ell \in \mathcal{L}}, \mathcal{R} \rangle$, where $\mathcal{R}$ is a set of *inference rules* of the form:

$$(r)\ \frac{c_1 :: e_1 \underset{\ell_1}{\longrightarrow} e'_1 \quad \cdots \quad c_n :: e_n \underset{\ell_n}{\longrightarrow} e'_n}{c :: e \underset{\ell}{\longrightarrow} e'} \text{ if } \mathit{Cond} \qquad (2)$$

Contextual Reduction Systems (continued)

An *ℓ-reduction of e_0 to e_m in context c* is an expression of the form $c :: e_0 \xrightarrow[\ell]{\Theta_1} e_1 \xrightarrow[\ell]{\Theta_2} \cdots e_{m-1} \xrightarrow[\ell]{\Theta_m} e_m$ with $m \geq 1$ s.t.

1. either $m = 1$, $e_1 = e_0$, and $\Theta_1 = [\,]$ (called *trivial reduction*)

2. or $c :: e_{i-1} \xrightarrow[\ell_i]{} e_i$ (for $i = 1, \ldots, m$) is the conclusion of an inference rule with premises $c_{1,j} :: e_{1,j} \xrightarrow[\ell_{1,j}]{} e'_{1,j}, \ldots, c_{n_i,j} :: e_{n_i,j} \xrightarrow[\ell_{n_i,j}]{} e'_{n_i,j}$ and the j-th element of Θ_i is an $\ell_{i,j}$-reduction of $e_{i,j}$ to $e'_{i,j}$ in context $c_{i,j}$.

Roadmap

- ☑ Introduction
- ☑ From Contextual Rewriting to Constraint Contextual Rewriting
- ☐ Constraint Contextual Rewriting
- ☐ Properties of Constraint Contextual Rewriting
- ☐ RDL
- ☐ Maple's evalution process as Constraint Contextual Rewriting

Constraint Contextual Rewriting

- Reasoning Specialists
- Clause Simplification
- Rewriting
- Augmentation

Some Definitions. . .

We consider quantifier-free first-order languages and we assume the usual conceptual machinery (e.g. the notion of substitution).

- By Σ, Π (possibly subscripted) we denote finite sets of function and predicate symbols (with their arity), respectively.
- A *signature* is a pair of the form (Σ, Π).
- V (possibly subscribed) denotes a finite set of variables.
- A (Σ, V)-*term* is a term built out of the symbols in Σ and the variables in V in the usual way.

. . . a few more Definitions

- A (Σ, Π, V)-*atom* is either an expression $q(t_1, \dots, t_n)$ where $q \in \Pi$ and $t_1 \dots t_n$ are (Σ, V)-terms or one of the propositional constants $true$ and $false$.

- (Σ, Π, V)-*formulae* are built in the obvious way using the standard logical connectives (i.e. $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$).

- A (Σ, Π, V)-*literal* p is either a (Σ, Π, V)-atom, $r(t_1, \dots, t_n)$, or a negated (Σ, Π, V)-atom, $\neg r(t_1, \dots, t_n)$.

- A (Σ, Π, V)-*expression* is a (Σ, V)-term or a (Σ, Π, V)-formula.

- A (Σ, Π, V)-*clause* is a disjunction of literals which we indicate as finite set of (Σ, Π, V)-literals.

Abbreviations

- We write (Σ, Π)-atom, (-literal, -expression, -clause) instead of $(\Sigma, \Pi, \emptyset)$-atom (-literal, -expression, -clause).
- If a is an atom, then $\overline{a}$ abbreviates $\neg a$ and $\overline{\neg a}$ stands for a.
- Let Q be a set of literals, then
  - $\overline{Q}$ abbreviates $\{\overline{q} : q \in Q\}$,
  - $Q \rightarrow p$ abbreviates the clause $\overline{Q} \cup \{p\}$, and
  - $\bigwedge Q$ stands for a conjunction of the literals in Q.

Semantics

- If ϕ is a (Σ, Π, V)-formula and Γ is a set of (Σ, Π, V)-formulae, then ϕ is a *logical consequence* of Γ iff $\Gamma \models \phi$, where $\models$ denotes entailment in classical predicate logic with equality.

- A (Σ, Π, V)*-theory* is a set of (Σ, Π, V)-formulae closed under logical consequence.

- If T is a theory, then $\Gamma \models_T \phi$ abbreviates $T \cup \Gamma \models \phi$ and we say that ϕ is T-entailed by Γ.

- A formula ϕ is T*-satisfiable* iff there exists a model of $T \cup \{\phi\}$, and T*-unsatisfiable* otherwise.

- A formula ϕ is T*-valid* iff ϕ is a logical consequence of T or, equivalently, iff $\phi \in T$.

Assumptions

In the following we consider two theories T_c and T_j of signature (Σ_c, Π_c) and (Σ_j, Π_j) respectively s.t.

- $\Sigma_c \subseteq \Sigma_j$,
- $\Pi_c \subseteq \Pi_j$,
- $T_c \subseteq T_j$.

The objective will be to simplify (Σ_j, Π_j)-expressions using:

1. a decision procedure for T_c and
2. a set R of T_j-valid facts.

From Decision Procedures. . .

According to the usual definition, a decision procedure for T_c

- takes a (Σ_c, Π_c)-formula as input and
- returns a 'yes-or-no' answer indicating whether the input formula is T_c-satisfiable or not.

However this definition is seldom adequate in practical applications:

- Efficiency considerations require the procedure to be *incremental*.
- The procedure is often required the ability of "normalizing" any given expression w.r.t. the information available.

. . . to Reasoning Specialists

A *reasoning specialist* is a state-based procedure whose states (called *constraint stores*) are finite sets of (Σ_c, Π_c)-literals represented in some internal form and whose functionalities are:

Functionality		Correctness Condition
$\texttt{cs-init}(C)$	$\Longrightarrow$	$\|C\|$ is T_c-valid

. . . to Reasoning Specialists

A *reasoning specialist* is a state-based procedure whose states (called *constraint stores*) are finite sets of (Σ_c, Π_c)-literals represented in some internal form and whose functionalities are:

Functionality		Correctness Condition
$\texttt{cs-init}(C)$	$\Longrightarrow$	$\|C\|$ is T_c-valid
$\texttt{cs-unsat}(C)$	$\Longrightarrow$	$\|C\|$ is T_c-unsatisfiable

. . . to Reasoning Specialists

A *reasoning specialist* is a state-based procedure whose states (called *constraint stores*) are finite sets of (Σ_c, Π_c)-literals represented in some internal form and whose functionalities are:

Functionality		Correctness Condition
$\texttt{cs-init}(C)$	$\Longrightarrow$	$\|C\|$ is T_c-valid
$\texttt{cs-unsat}(C)$	$\Longrightarrow$	$\|C\|$ is T_c-unsatisfiable
(ax-cs-simp) $\dfrac{}{P :: C \xrightarrow[\text{cs-simp}]{} C'}$	$\Longrightarrow$	$P, C \models_{T_c} \bigwedge C'$

. . . to Reasoning Specialists

A *reasoning specialist* is a state-based procedure whose states (called *constraint stores*) are finite sets of (Σ_c, Π_c)-literals represented in some internal form and whose functionalities are:

Functionality		Correctness Condition
$\texttt{cs-init}(C)$	$\Longrightarrow$	$\lVert C \rVert$ is T_c-valid
$\texttt{cs-unsat}(C)$	$\Longrightarrow$	$\lVert C \rVert$ is T_c-unsatisfiable
(ax-cs-simp) $\dfrac{}{P :: C \xrightarrow[\text{cs}-\text{simp}]{} C'}$	$\Longrightarrow$	$P, C \models_{T_c} \bigwedge C'$
(ax-cs-normal) $\dfrac{}{C :: e \xrightarrow[\text{cs}-\text{normal}]{} e'}$	$\Longrightarrow$	$C \models_{T_c} e \sim e'$

Example: a Reasoning Specialist for Total Orders

- Let $\Sigma_c = \Sigma_j$, $\Pi_c = \{\leq, =\}$, and $\Pi_c = \{\leq, <, \leq, <, =\}$.
- Let T_c is a (Σ_c, Π_c)-theory for total orders.
- Constraint stores are finite sets of (Σ_c, Π_c)-literals of the form $t_1 \leq t_2$ or $t_1 \neq t_2$ closed under:

$$(trans)\ \frac{t_2 \leq t_2 \quad t_2 \leq t_3}{t_1 \leq t_3}$$

For instance,

$\{a \leq b, b \leq c, c \neq a\}$ is a **not** constraint store.

$\{a \leq b, b \leq c, a \leq c, c \neq a\}$ is a constraint store.

Example (continued): a Reasoning Specialist for Total Orders

The functionalities provided by the reasoning specialist are:

- $\texttt{cs-init}(C)$ holds iff $C = \emptyset$.
- $\texttt{cs-unsat}(C)$ holds iff C contains three literals of the form $t_1 \leq t_2$, $t_2 \leq t_1$, and $t_1 \neq t_2$.
- $C :: e \xrightarrow[\text{cs}-\text{normal}]{} e'$ iff $\{s \leq t, t \leq s\} \subseteq C$ and $e' = e[t/s]$.

  For instance:

$$\{\ldots, n^2 \leq 2^n, 2^n \leq n^2, \ldots\} :: \sqrt{2^n} = n \xrightarrow[\text{cs}-\text{normal}]{} \sqrt{n^2} = n$$

Example (continued): a Reasoning Specialist for Total Orders

Let ν be a function associating sets of (Σ_c, Π_c)-literals to (Σ_c, Π_j)-literals defined by:

c	$\nu(c)$	c	$\nu(c)$
$t_1 \leq t_2$	$\{t_1 \leq t_2\}$	$t_1 \neq t_2$	$\{t_1 \neq t_2\}$
$t_1 \not< t_2$	$\{t_2 \leq t_1\}$	$t_1 = t_2$	$\{t_1 \leq t_2, t_2 \leq t_1\}$
$t_1 < t_2$	$\{t_1 \leq t_2, t_1 \neq t_2\}$	$t_1 \not\leq t_2$	$\{t_2 \leq t_1, t_1 \neq t_2\}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

- $P :: C \xrightarrow[\text{cs}-\text{simp}]{} C'$ iff C' is the result of adding $\nu(P)$ to C and closing the result w.r.t. $(trans)$.

  For instance:

$$\{2^n < n^2\} :: \{n^2 \leq 2^n\} \xrightarrow[\text{cs}-\text{extend}]{} \{n^2 \leq 2^n, 2^n \leq n^2, 2^n \neq n^2\}$$

Constraint Store Extension

The extension of the constraint store is modeled by:

$$\textit{(cs-simp)}\ \frac{P :: C \xrightarrow[\mathsf{cs-simp}]{} C'}{P :: C \xrightarrow[\mathsf{cs-extend}]{} C'}$$

where P is a finite set of (Σ_j, Π_j)-literals.

Note: The $\xrightarrow[\mathsf{cs-extend}]{}$ relation is introduced for modularity reasons and will be extended later.

Clause Simplification

$$\textit{(cl-true)}\ \frac{}{E \cup \{true\} \xrightarrow[\text{simp}]{} \{true\}}$$

Clause Simplification

(cl-true) $$\frac{}{E \cup \{true\} \xrightarrow[\text{simp}]{} \{true\}}$$

(cl-false) $$\frac{}{E \cup \{false\} \xrightarrow[\text{simp}]{} E}$$

Clause Simplification

$$\textit{(cl-true)}\ \frac{}{E \cup \{true\} \underset{\mathsf{simp}}{\longrightarrow} \{true\}}$$

$$\textit{(cl-false)}\ \frac{}{E \cup \{false\} \underset{\mathsf{simp}}{\longrightarrow} E}$$

$$\textit{(cl-simp)}\ \frac{\overline{E} :: C_\circ \underset{\mathsf{cs-extend}}{\longrightarrow} C \quad C :: p \underset{\mathsf{ccr}}{\longrightarrow} p'}{E \cup \{p\} \underset{\mathsf{simp}}{\longrightarrow} E \cup \{p'\}} \text{ if } \texttt{cs-init}(C_\circ)$$

Example: Clause Simplification

Let $E = \{n \not\geq 0, n^2 \neq 2^n, \sqrt{2^n} = n\}$.

We show that $E \xrightarrow[\text{simp}]{}^{*} \{true\}$.

Example: Clause Simplification

Let $E = \{n \not\geq 0, n^2 \neq 2^n, \sqrt{2^n} = n\}$.

We show that $E \xrightarrow[\text{simp}]{}^* \{true\}$.

We apply the inference rule *(cl-simp)*

Example: Clause Simplification

Let $E = \{n \not\geq 0, n^2 \neq 2^n, \sqrt{2^n} = n\}$.

We show that $E \xrightarrow[\text{simp}]{}^* \{true\}$.

We apply the inference rule *(cl-simp)*

$$E \xrightarrow[\text{simp}]{\left[\{n\geq 0, n^2=2^n\}::\emptyset \xrightarrow[\text{cs-extend}]{[\cdots]} C \quad C::\sqrt{2^n}=n \xrightarrow[\text{ccr}]{\Pi} true\right]} \{n \not\geq 0, n^2 \neq 2^n, true\} \cdot$$

where $C = \{0 \leq n, n^2 \leq 2^n, 2^n \leq n^2\}$.

Example: Clause Simplification

Let $E = \{n \not\geq 0, n^2 \neq 2^n, \sqrt{2^n} = n\}$.

We show that $E \xrightarrow[\text{simp}]{}^* \{true\}$.

We apply the inference rule *(cl-simp)* and then *(cl-true)*

$$E \xrightarrow[\text{simp}]{\left[\{n \geq 0, n^2 = 2^n\} :: \emptyset \xrightarrow[\text{cs-extend}]{[\cdots]} C \quad C :: \sqrt{2^n} = n \xrightarrow[\text{ccr}]{\Pi} true\right]} \{n \not\geq 0, n^2 \neq 2^n, true\} \cdots$$

$$\cdots \{n \not\geq 0, n^2 \neq 2^n, true\} \xrightarrow[\text{simp}]{} \{true\}$$

where $C = \{0 \leq n, n^2 \leq 2^n, 2^n \leq n^2\}$.

Rewriting

(cxt-entails) $$\frac{\{\overline{p}\} :: C \xrightarrow[\text{cs}-\text{extend}]{} C'}{C :: p \xrightarrow[\text{ccr}]{} true}$$ if p is a (Σ_j, Π_c)-literal and $\texttt{cs-unsat}(C')$

Rewriting

$$\textit{(cxt-entails)}\ \frac{\{\overline{p}\} :: C \xrightarrow[\text{cs}-\text{extend}]{} C'}{C :: p \xrightarrow[\text{ccr}]{} true}\ \text{if}\ \begin{array}{l} p \text{ is a } (\Sigma_j, \Pi_c)\text{-literal and} \\ \texttt{cs-unsat}(C') \end{array}$$

$$\textit{(normal)}\ \frac{C :: e \xrightarrow[\text{cs}-\text{normal}]{} e'}{C :: e \xrightarrow[\text{ccr}]{} e'}$$

Rewriting

$$\textit{(cxt-entails)}\ \frac{\{\overline{p}\} :: C \xrightarrow[\text{cs-extend}]{} C'}{C :: p \xrightarrow[\text{ccr}]{} true} \text{ if } \begin{array}{l} p \text{ is a } (\Sigma_j, \Pi_c)\text{-literal and} \\ \texttt{cs-unsat}(C') \end{array}$$

$$\textit{(normal)}\ \frac{C :: e \xrightarrow[\text{cs-normal}]{} e'}{C :: e \xrightarrow[\text{ccr}]{} e'}$$

$$\textit{(crew)}\ \frac{C :: Q\sigma \xrightarrow[\text{ccr}]{} \emptyset}{C :: s[l\sigma] \xrightarrow[\text{ccr}]{} s[r\sigma]} \text{ if } \begin{array}{l} (Q \rightarrow l = r) \in R \text{ and} \\ \sigma \text{ is a ground substitution s.t.} \end{array}$$

Example: ccr-reduction

The ccr-reduction occurring in the example before is:

$$C :: \sqrt{2^n} = n \xrightarrow[\text{ccr}]{\left[C::\sqrt{2^n}=n \xrightarrow[\text{cs}-\text{normal}]{} \sqrt{n^2}=n \right]} \sqrt{n^2} = n \cdots$$

Example: ccr-reduction

The ccr-reduction occurring in the example before is:

$$C :: \sqrt{2^n} = n \xrightarrow[\text{ccr}]{\left[C::\sqrt{2^n}=n \xrightarrow[\text{cs}-\text{normal}]{} \sqrt{n^2}=n \right]} \sqrt{n^2} = n \cdots$$

$$\cdots \sqrt{n^2} = n \xrightarrow[\text{ccr}]{\left[C::n\geq 0 \xrightarrow[\text{ccr}]{\left[\{n \ngeq 0\}::C \xrightarrow[\text{cs}-\text{extend}]{} C' \right]} true \right]} n = n$$

Example: ccr-reduction

The ccr-reduction occurring in the example before is:

$$C :: \sqrt{2^n} = n \xrightarrow[\text{ccr}]{\left[C::\sqrt{2^n}=n \xrightarrow[\text{cs-normal}]{} \sqrt{n^2}=n \right]} \sqrt{n^2} = n \cdots$$

$$\cdots \sqrt{n^2} = n \xrightarrow[\text{ccr}]{\left[C::n\geq 0 \xrightarrow[\text{ccr}]{\left[\{n \not\geq 0\}::C \xrightarrow[\text{cs-extend}]{} C' \right]} true \right]} n = n$$

$$\cdots n = n \xrightarrow[\text{ccr}]{[\{n \neq n\}::C \xrightarrow[\text{cs-extend}]{} C'']} true$$

where C' and C'' are readily found T_c-unsatisfiable by `cs-unsat`.

Augmenting the Constraint Store

- When the context is T_j-unsat but not T_c-unsat, then the T_j-unsatisfiability of the context can not possibly be detected by the reasoning specialist.

- The occurrence in the context of (function) symbols interpreted in T_j but not in T_c is the main cause of the problem.

- *Augmentation* extends the context with T_j-valid facts, thereby providing the reasoning specialist with properties of function symbols it is otherwise not aware of.

Augmenting the Constraint Store (continued)

- By adding T_j-valid facts to the context the heuristics aims at generating a T_j-equivalent but T_c-unsat context whose T_j-unsatisfiability can therefore be detected by the reasoning specialist.

- The selection of suitable T_j-valid facts is done by looking through the available lemmas.

Example: Augmentation

Let T_j be a theory of integer numbers with the usual interpretation of symbols. Let $\Sigma_c = \Sigma_j$ and let Π_c, T_c, and $\xrightarrow[\text{cs}-\text{simp}]{}$ be as before. Let the following two formulae be in R (and hence in T_j):

$$X \geq 0 \rightarrow \sqrt{X^2} = X \tag{3}$$

$$X \geq 4 \rightarrow X^2 \leq 2^X \tag{4}$$

Notice that (3) and (4) are in T_j, but not in T_c.

Let $\overbrace{\{n \geq 4, n^2 \geq 2^n\}}^{\text{context}} \; \overbrace{\sqrt{2^n} = n}^{\text{focus}}$

Unlike the previous examples, $n^2 = 2^n$ is not T_c-**entailed** by the context. However it is T_j-**entailed** by the context.

Example (continued): Augmentation

- Now, if we extend the context with $n^2 \leq 2^n$ we get a new context that T_c-**entails** $n^2 = 2^n$.

- This enables the decision procedure to rewrite the focus literal to $\sqrt{n^2} = n$ and the reasoning continues as the previous example.

- Notice that the task of establishing that $n^2 = 2^n$ is T_j-**entailed** by the initial context falls largely beyond the scope of a decision procedure for total orders.

- The problem is nevertheless solved thanks to the use of the augmentation heuristics.

Constraint Store Extension with Augmentation

We recall that the activity of extending the constraint store is modeled by:

$$\textit{(cs-simp)}\ \frac{P :: C \xrightarrow[\textsf{cs}-\textsf{simp}]{} C'}{P :: C \xrightarrow[\textsf{cs}-\textsf{extend}]{} C'}$$

Constraint Store Extension with Augmentation

We recall that the activity of extending the constraint store is modeled by:

$$\textit{(cs-simp)}\ \frac{P :: C \xrightarrow[\text{cs}-\text{simp}]{} C'}{P :: C \xrightarrow[\text{cs}-\text{extend}]{} C'}$$

This is now supplemented with the following rule for augmentation:

$$\textit{(aug)}\ \frac{C :: q_1\sigma \xrightarrow[\text{ccr}]{} true \quad \cdots \quad C :: q_n\sigma \xrightarrow[\text{ccr}]{} true \quad \{c\sigma\} :: C \xrightarrow[\text{cs}-\text{simp}]{} C'}{P :: C \xrightarrow[\text{cs}-\text{extend}]{} C'}$$

if $(\{q_1, \ldots, q_n\} \to c) \in R$ and
σ is a ground substitution s.t. $c\sigma$ is a (Σ_j, Π_c)-literal

Roadmap

- ☑ Introduction
- ☑ From Contextual Rewriting to Constraint Contextual Rewriting
- ☑ Constraint Contextual Rewriting
- ☐ Properties of Constraint Contextual Rewriting
- ☐ RDL
- ☐ Maple's evalution process as Constraint Contextual Rewriting

Properties of $\mathbf{CCR}(X)$

Theorem 1. [Soundness] *CCR(X) is sound, i.e.*

1. *if* $C :: e \xrightarrow[\text{ccr}]{}^{*} e'$, *then* $\|C\| \models_{T_j} e \sim e'$;

2. *if* $P :: C \xrightarrow[\text{cs}-\text{extend}]{}^{*} C'$, *then* $P, \|C\| \models_{T_j} \bigwedge \|C'\|$.

Theorem 2. [Termination] *(The version on the paper of)* *CCR(X) is terminating, i.e. all the reductions of the form:*

- $E_0 \xrightarrow[\text{simp}]{} E_1 \xrightarrow[\text{simp}]{} \cdots$

- $C :: e_0 \xrightarrow[\text{ccr}]{} e_1 \xrightarrow[\text{ccr}]{} \cdots$

have finite size.

Reference paper on CCR

- A. Armando and S. Ranise. Constraint Contextual Rewriting. To appear on the Journal of Symbolic Computation, special issue on First Order Theorem Proving. Peter Baumgartner and Hantao Zhang, Editors.

Conclusions

- $\mathrm{CCR}(X)$ is a generalized form of contextual rewriting which incorporates the functionalities provided by a decision procedure.
- $\mathrm{CCR}(X)$ is sound and terminating.
- By using $\mathrm{CCR}(X)$ as a reference model, the problem of the integration of decision procedures in formula simplification is reduced to the implementation of a decision procedure for the fragment of choice.