Decidable Logics

This chapter is about decidable logics. There are many decidable fragments of first-order logic, some of them are discussed in Chapter 3 and Chapter 5.
Here I discuss logics that are typically not representable in first-order logic, e.g., linear integer arithmetic, Section 6.2, or logics where specialized decision procedures exist, beyond the general procedures discussed in previous chapters, e.g., equational reasoning on ground terms by congruence closure, Section 6.1

Congruence Closure

An equational clause

$$\forall \vec{x}\,(t_1 \approx s_1 \vee \ldots \vee t_n \approx s_n \vee l_1 \not\approx r_1 \vee \ldots \vee l_k \not\approx r_k)$$

is valid iff

$$\exists \vec{x}\,(t_1 \not\approx s_1 \wedge \ldots \wedge t_n \not\approx s_n \wedge l_1 \approx r_1 \wedge \ldots \wedge l_k \approx r_k)$$

is unsatisfiable iff the Skolemized (ground!) formula

$$(t_1 \not\approx s_1 \wedge \ldots \wedge t_n \not\approx s_n \wedge l_1 \approx r_1 \wedge \ldots \wedge l_k \approx r_k)\{\vec{x} \mapsto \vec{c}\}$$

is unsatisfiable iff the formula

$$(t_1 \approx s_1 \vee \ldots \vee t_n \approx s_n \vee l_1 \not\approx r_1 \vee \ldots \vee l_k \not\approx r_k)\{\vec{x} \mapsto \vec{c}\}$$

is valid.

Flattening

$E = l_1 \approx r_1 \wedge \ldots \wedge l_n \approx r_n$

Flattening $E[f(t_1, \ldots, t_n)]_{p_1,\ldots,p_k} \Rightarrow_{\text{CCF}}$
$E[c/p_1, \ldots, p_k] \wedge f(t_1, \ldots, t_n) \approx c$
provided all t_i are constants, the p_j are all positions in E of $f(t_1, \ldots, t_n)$, $|p_k| > 2$ for some k, or, $p_k = n.2$ and $E|_{m.1}$ is not a constant for some n, and c is fresh

As a result: only two kinds of equations left.
Term equations: $f(c_{i_1}, \ldots, c_{i_n}) \approx c_{i_0}$
Constant equations: $c_i \approx c_j$.

Congruence Closure

The congruence closure algorithm is presented as a set of abstract rewrite rules operating on a pair of equations E and a set of rules R, $(E; R)$, similar to Knuth-Bendix completion, Section 4.4.

$(E_0; R_0) \Rightarrow_{\text{CC}} (E_1; R_1) \Rightarrow_{\text{CC}} (E_2; R_2) \Rightarrow_{\text{CC}} \cdots$

At the beginning, $E = E_0$ is a set of constant equations and $R = R_0$ is the set of term equations oriented from left-to-right. At termination, E is empty and R contains the result.

Simplify $(E \uplus \{c \mathrel{\dot{\approx}} c'\}; R \uplus \{c \to c''\}) \Rightarrow_{\mathrm{CC}}$
$(E \cup \{c'' \mathrel{\dot{\approx}} c'\}; R \cup \{c \to c''\})$

Delete $(E \uplus \{c \approx c\}; R) \Rightarrow_{\mathrm{CC}} (E; R)$

Orient $(E \uplus \{c \mathrel{\dot{\approx}} c'\}; R) \Rightarrow_{\mathrm{CC}} (E; R \cup \{c \to c'\})$
if $c \succ c'$

Deduce $(E; R \uplus \{t \to c,\ t \to c'\}) \Rightarrow_{\text{CC}}$
$(E \cup \{c \approx c'\}; R \cup \{t \to c\})$

Collapse $(E; R \uplus \{t[c]_p \to c',\ c \to c''\}) \Rightarrow_{\text{CC}}$
$(E; R \cup \{t[c'']_p \to c',\ c \to c''\})$
$p \neq \epsilon$

For rule Deduce, t is either a term of the form $f(c_1, \ldots, c_n)$ or a constant c_i.
For rule Collapse, t is always of the form $f(c_1, \ldots, c_n)$