Chapter 5

Lists

DECISION PROCEDURES
LAST UPDATE OF LECTURE NOTES: TUESDAY, MARCH 7, 2006
LAST UPDATE OF THIS CHAPTER: WEDNESDAY, MARCH 1, 2006.

5.1 Constructors

5.1.1 Definition
The SIGNATURE OF CONSTRUCTORS Σ_{cons} contains the following symbols:

- A sort elem for elements and a sort list for lists of elements.
- the constant symbol nil, of sort list;
- the function symbol cons, of arity $\mathsf{elem} \times \mathsf{list} \to \mathsf{list}$.

5.1.2 Definition
A STANDARD cons-STRUCTURE $\mathcal{A}$ is a Σ_{cons}-structure satisfying the following conditions:

- $A_{\mathsf{list}} = (A_{\mathsf{elem}})^*$;
- $\mathsf{nil}^{\mathcal{A}} = \langle\rangle$;
- $\mathsf{cons}^{\mathcal{A}}(e, \langle e_1, \ldots, e_n \rangle) = \langle e, e_1, \ldots, e_n \rangle$, for each $n \geq 0$ and $e, e_1, \ldots, e_n \in A_{\mathsf{elem}}$.

5.1.3 Definition
The MODELCLASS OF CONSTRUCTORS is the pair $M_{\mathsf{cons}} = (\Sigma_{\mathsf{cons}}, \mathbf{A})$, where $\mathbf{A}$ is the class of all Σ_{cons}-structures that are isomorphic to standard cons-structures.

5.2 Lists

5.2.1 Definition
The SIGNATURE OF LISTS Σ_{list} extends the signature of constructors Σ_{cons} with the function symbols:

- car, of arity $\mathsf{list} \to \mathsf{elem}$;
- cdr, of arity $\mathsf{list} \to \mathsf{list}$.

5.2.2 Definition
A STANDARD list-STRUCTURE $\mathcal{A}$ is a Σ_{list}-structure satisfying the following conditions:

- $\mathcal{A}^{\Sigma_{\mathsf{cons}}}$ is a standard cons-structure;
- $\mathsf{car}^{\mathcal{A}}(\langle e_1, \ldots, e_n \rangle) = e_1$, for each $n > 0$ and $e_1, \ldots, e_n \in A_{\mathsf{elem}}$;
- $\mathsf{cdr}^{\mathcal{A}}(\langle e_1, \ldots, e_n \rangle) = \langle e_2, \ldots, e_n \rangle$, for each $n > 0$ and $e_1, \ldots, e_n \in A_{\mathsf{elem}}$.

5.2.3 Definition
The MODELCLASS OF LISTS is the pair $M_{\mathsf{list}} = (\Sigma_{\mathsf{list}}, \mathbf{A})$, where $\mathbf{A}$ is the class of all Σ_{list}-structures that are isomorphic to standard list-structures.

5.3 From constructors to equality with acyclicity test

5.3.1 Algorithm (IS-SATISFIABLE-CONSTRUCTORS)
Input: A finite set Γ of flat Σ_{cons}-literals
Output: `satisfiable` if Γ is satisfiable; `unsatisfiable` otherwise

function IS-SATISFIABLE-CONSTRUCTORS(Γ)
    $X \leftarrow vars(\Gamma)$
    For each literal of the form $x \approx \mathsf{cons}(e, y)$ in Γ, add to Γ the literals

$$e \approx \mathsf{car}(x)\,, \qquad\qquad y \approx \mathsf{cdr}(x)$$

    $T \leftarrow$ the set of terms occurring in Γ
    $R \leftarrow \{(s, t) \in T \times T \mid \text{the literal } s \approx t \text{ is in } \Gamma\}$
    $C \leftarrow$ the congruence closure of R with respect to T
    Let $\preceq$ be the binary relation of X_{list} defined by letting $x \preceq y$ iff there is a literal $y' \approx \mathsf{cons}(e, x')$ in Γ such that $(x, x') \in C$ and $(y, y') \in C$
    if $x \not\approx y$ is in Γ and $(x, y) \in C$ **then**
        return `unsatisfiable`
    else if $x \approx \mathsf{nil}$ and $y \approx \mathsf{cons}(e, z)$ are in Γ, and $(x, y) \in C$ **then**
        return `unsatisfiable`
    else if $\preceq$ is not acyclic **then**
        return `unsatisfiable`
    else

```
15:         return satisfiable
16:     end if
17: end function
```

5.3.2 Proposition

If Algorithm IS-SATISFIABLE-CONSTRUCTORS *returns* `satisfiable` *then* Γ *is* M_{cons}*-satisfiable.*

PROOF. Assume without loss of generality that $X_{\mathsf{elem}} \neq \varnothing$. Γ is clearly satisfied by the M_{cons}-interpretation $\mathcal{A}$ over X defined as follows. First, we let

$$A_{\mathsf{elem}} = (X_{\mathsf{elem}}/C) \cup \{\nu_0\} \,.$$

where $\nu_0 \notin X_{\mathsf{elem}}/C$. Then, we let

$$e^{\mathcal{A}} = [e]_C \,, \qquad \text{for all } \mathsf{elem}\text{-variables } e \in X_{\mathsf{elem}} \,.$$

We also let

$$x^{\mathcal{A}} = \langle\rangle \,, \qquad \text{if the literal } x \approx \mathsf{nil} \text{ is in } \Gamma \,,$$

and

$$x^{\mathcal{A}} = \langle e^{\mathcal{A}} \rangle \circ y^{\mathcal{A}} \,, \qquad \text{if the literal } x \approx \mathsf{cons}(e, y) \text{ is in } \Gamma \,.$$

For all the other list-variables, we let

$$x^{\mathcal{A}} = \underbrace{\langle \nu_0 \rangle \circ \cdots \circ \langle \nu_0 \rangle}_{h(x) \text{ times}}$$

where $h : X_{\mathsf{list}} \rightarrow \mathbb{N}^+$ is an arbitrarily fixed injective function.

5.3.3 Proposition

If Algorithm IS-SATISFIABLE-CONSTRUCTORS *returns* `unsatisfiable` *then* Γ *is* M_{cons}*-unsatisfiable.*

PROOF. We prove the stronger fact that Γ is M_{list}-unsatisfiable.

By contradiction, assume that Γ is M_{list}-satisfiable. Then there exists an M_{list}-interpretation $\mathcal{A}$ such that $\mathcal{A} \models \Gamma$. Let R' be the binary relation of T defined by letting $(s,t) \in R'$ iff $s^{\mathcal{A}} = t^{\mathcal{A}}$. Then $C \subseteq R'$.

If the algorithm ended at line 9, then $(x, y) \in C$, which implies $x^{\mathcal{A}} = y^{\mathcal{A}}$ contradicting the fact that the literal $x \not\approx y$ is in Γ.

If the algorithm ended at line 11, then $(x, y) \in C$, which implies $\mathsf{nil}^{\mathcal{A}} = [\mathsf{cons}(e, z)]^{\mathcal{A}}$, a contradiction.

If the algorithm ended at line 13, then there is a cycle $x_1 \preceq x_2 \preceq \cdots x_n \preceq x_1$, implying that $x_1^{\mathcal{A}} = [\mathsf{cons}(e_1, x_2)]^{\mathcal{A}}$, $x_2 = [\mathsf{cons}(e_2, x_3)]^{\mathcal{A}}, \ldots, x_n^{\mathcal{A}} = [\mathsf{cons}(e_n, x_1)]^{\mathcal{A}}$, a contradiction.

5.3.4 Proposition

Algorithm IS-SATISFIABLE-CONSTRUCTORS *is correct.*

PROOF. Termination is obvious. Partial correctness follows by Propositions 5.3.2 and 5.3.3.

5.4 From lists to constructors

5.4.1 Algorithm (IS-SATISFIABLE-LISTS)
Input: A finite set Γ of flat Σ_{list}-lists
Output: `satisfiable` if Γ is satisfiable; `unsatisfiable` otherwise

function IS-SATISFIABLE-LISTS(Γ)
  $\Delta \leftarrow$ LISTS-TO-CONSTRUCTORS(Γ)
  return IS-SATISFIABLE-CONSTRUCTORS(Δ)
end function

function LISTS-TO-CONSTRUCTORS(Γ)
  $X \leftarrow vars(\Gamma)$
  Replace each literal of the form $e \approx \mathsf{car}(x)$ in Γ with the formula

$$x \not\approx \mathsf{nil} \;\rightarrow\; x \approx \mathsf{cons}(e, y')\,,$$

  where y' is a fresh free constant symbol of sort list.
  Replace each literal of the form $x \approx \mathsf{cdr}(y)$ in Γ with the formula

$$y \not\approx \mathsf{nil} \;\rightarrow\; y \approx \mathsf{cons}(e', x)\,,$$

  where e' is a fresh free constant symbol of sort elem.
  return Γ
end function

5.4.2 Proposition
In algorithm IS-SATISFIABLE-LISTS, *let* Δ *be the output returned by the call to* LISTS-TO-CONSTRUCTORS(Γ). *Then the following are equivalent:*

1. Γ *is* M_{list}*-satisfiable.*
2. Δ *is* M_{cons}*-satisfiable.*

PROOF. ($1 \implies 2$). Immediate.

($1 \implies 2$). Let $\mathcal{B}$ be an M_{cons}-interpretation over $vars(\Delta)$ satisfying Δ. Then it is easy to check that Γ is satisfied by the M_{list}-interpretation $\mathcal{A}$ over X constructed by letting

$$A_{\mathsf{elem}} = B_{\mathsf{elem}}\,,$$
$$A_{\mathsf{list}} = B_{\mathsf{list}}\,,$$

and

$$e^{\mathcal{A}} = e^{\mathcal{B}}\,, \qquad \text{for each } e \in X_{\mathsf{elem}}\,,$$
$$x^{\mathcal{A}} = x^{\mathcal{B}}\,, \qquad \text{for each } x \in X_{\mathsf{list}}\,.$$

5.4.3 Proposition
Algorithm IS-SATISFIABLE-LISTS *is correct.*

PROOF. Termination is obvious. Partial correctness follows by Proposition 5.4.2.