CS 511, Fall 2024, Lecture Slides 27

Gilmore's Algorithm

Assaf Kfoury

16 October 2024

review and reminders *(run simultaneously with an example on the board)*

From lecture notes and Lecture Slides 26: $\boxed{\text{sko,pre}}(\varphi) \stackrel{\text{def}}{=} \boxed{\text{skolem}}(\boxed{\text{prenex}}(\varphi))$.

In $4, 5, \ldots, 12$ below, assume φ does not mention equality symbol '$\approx$' for simplicity :

1. If φ is a first-order sentence, then $\boxed{\text{sko,pre}}(\varphi)$ is its Skolem form.
2. In particular, $\boxed{\text{sko,pre}}(\varphi)$ is a universal first-order sentence, *i.e.*, it is in prenex normal form and all the quantifiers in its prenex are universal.
3. φ and $\boxed{\text{sko,pre}}(\varphi)$ are equisatisfiable

review and reminders *(run simultaneously with an example on the board)*

From lecture notes and Lecture Slides 26: $\boxed{\text{sko,pre}}(\varphi) \stackrel{\text{def}}{=} \boxed{\text{skolem}}(\boxed{\text{prenex}}(\varphi))$.

In $4, 5, \ldots, 12$ below, assume φ does not mention equality symbol '$\approx$' for simplicity :

1. If φ is a first-order sentence, then $\boxed{\text{sko,pre}}(\varphi)$ is its Skolem form.
2. In particular, $\boxed{\text{sko,pre}}(\varphi)$ is a universal first-order sentence, *i.e.*, it is in prenex normal form and all the quantifiers in its prenex are universal.
3. φ and $\boxed{\text{sko,pre}}(\varphi)$ are equisatisfiable
4. $\text{H\_Expansion}\big(\boxed{\text{sko,pre}}(\varphi)\big) \stackrel{\text{def}}{=}$ "*delete the prefix of* $\boxed{\text{sko,pre}}(\varphi)$ *and substitute ground terms for variables in the matrix of* $\boxed{\text{sko,pre}}(\varphi)$ *in all possible ways*."
5. φ and $\text{H\_Expansion}\big(\boxed{\text{sko,pre}}(\varphi)\big)$ are equisatisfiable

review and reminders *(run simultaneously with an example on the board)*

From lecture notes and Lecture Slides 26: $\boxed{\text{sko,pre}}(\varphi) \stackrel{\text{def}}{=} \boxed{\text{skolem}}(\boxed{\text{prenex}}(\varphi))$.

In $4, 5, \ldots, 12$ below, assume φ does not mention equality symbol '$\approx$' for simplicity :

1. If φ is a first-order sentence, then $\boxed{\text{sko,pre}}(\varphi)$ is its Skolem form.
2. In particular, $\boxed{\text{sko,pre}}(\varphi)$ is a universal first-order sentence, *i.e.*, it is in prenex normal form and all the quantifiers in its prenex are universal.
3. φ and $\boxed{\text{sko,pre}}(\varphi)$ are equisatisfiable
4. $\text{H\_Expansion}\big(\boxed{\text{sko,pre}}(\varphi)\big) \stackrel{\text{def}}{=}$ "*delete the prefix of* $\boxed{\text{sko,pre}}(\varphi)$ *and substitute ground terms for variables in the matrix of* $\boxed{\text{sko,pre}}(\varphi)$ *in all possible ways*."
5. φ and $\text{H\_Expansion}\big(\boxed{\text{sko,pre}}(\varphi)\big)$ are equisatisfiable
6. $\boxed{\text{FOL} \mapsto \text{PL}}\Big(\text{H\_Expansion}\big(\boxed{\text{sko,pre}}(\varphi)\big)\Big) \stackrel{\text{def}}{=}$ "*replace every ground atom* α *in* $\text{H\_Expansion}\big(\boxed{\text{sko,pre}}(\varphi)\big)$ *by a propositional variable* X_α."
7. φ is satisfiable (in FOL) iff
   $\boxed{\text{FOL} \mapsto \text{PL}}\Big(\text{H\_Expansion}\big(\boxed{\text{sko,pre}}(\varphi)\big)\Big)$ is satisfiable (in PL).

review and reminders *(run simultaneously with an example on the board)*

8. φ is satisfiable (in FOL) iff
$\boxed{\text{FOL} \mapsto \text{PL}}\big(\text{H\_Expansion}\big(\boxed{\text{sko,pre}}(\varphi)\big)\big)$ is **finitely** satisfiable (in PL).

review and reminders *(run simultaneously with an example on the board)*

8. φ is satisfiable (in FOL) iff
$\boxed{\text{FOL} \mapsto \text{PL}}\Big(\text{H\_Expansion}\Big(\boxed{\text{sko,pre}}(\varphi)\Big)\Big)$ is **finitely** satisfiable (in PL).

9. Contrapositively:
φ is **not** satisfiable (in FOL) iff
there is a **finite** subset of $\boxed{\text{FOL} \mapsto \text{PL}}\Big(\text{H\_Expansion}\Big(\boxed{\text{sko,pre}}(\varphi)\Big)\Big)$
which is **not** satisfiable (in PL).

review and reminders *(run simultaneously with an example on the board)*

8. φ is satisfiable (in FOL) iff $\boxed{\text{FOL} \mapsto \text{PL}}\big(\text{H\_Expansion}\big(\boxed{\text{sko,pre}}(\varphi)\big)\big)$ is **finitely** satisfiable (in PL).

9. Contrapositively:
   φ is **not** satisfiable (in FOL) iff there is a **finite** subset of $\boxed{\text{FOL} \mapsto \text{PL}}\big(\text{H\_Expansion}\big(\boxed{\text{sko,pre}}(\varphi)\big)\big)$ which is **not** satisfiable (in PL).

10. Recall that a first-order sentence ψ is **valid** iff $\neg\psi$ is **not** satisfiable.

11. Suppose we want to test whether a first-order sentence ψ is valid. Let

$$\boxed{\text{FOL} \mapsto \text{PL}}\big(\text{H\_Expansion}\big(\boxed{\text{sko,pre}}(\neg\psi)\big)\big) = \{\theta_1,\ \theta_2,\ \theta_3, \ldots\}$$

Note the inserted logical negation "$\neg$". All the θ_i's are propositional WFF's.

review and reminders *(run simultaneously with an example on the board)*

8. φ is satisfiable (in FOL) iff
$\boxed{\text{FOL} \mapsto \text{PL}}\Big(\text{H\_Expansion}\Big(\boxed{\text{sko,pre}}(\varphi)\Big)\Big)$ is **finitely** satisfiable (in PL).

9. Contrapositively:
φ is **not** satisfiable (in FOL) iff
there is a **finite** subset of $\boxed{\text{FOL} \mapsto \text{PL}}\Big(\text{H\_Expansion}\Big(\boxed{\text{sko,pre}}(\varphi)\Big)\Big)$
which is **not** satisfiable (in PL).

10. Recall that a first-order sentence ψ is **valid** iff $\neg\psi$ is **not** satisfiable .

11. Suppose we want to test whether a first-order sentence ψ is valid. Let

$$\boxed{\text{FOL} \mapsto \text{PL}}\Big(\text{H\_Expansion}\Big(\boxed{\text{sko,pre}}(\neg\,\psi)\Big)\Big) = \{\theta_1,\ \theta_2,\ \theta_3, \ldots\}$$

Note the inserted logical negation "$\neg$". All the θ_i's are propositional WFF's.

12. ψ is **valid** (in FOL) iff
there is a **finite** subset of $\{\theta_1, \theta_2, \theta_3, \ldots\}$ which is **not** satisfiable (in PL).

Gilmore's algorithm

Assume equality symbol '≈' does not occur in ψ for simplicity .
Details for how to proceed when '≈' occurs are in lecture notes.

1. **input**: first-order sentence ψ to be tested for validity ;
2. $k := 0$;
3. **repeat** $k := k + 1$
   generate first k wff's $\{\theta_1, \ldots, \theta_k\}$ in:

   $$\boxed{\text{FOL} \mapsto \text{PL}}\Big(\text{H\_Expansion}\Big(\boxed{\text{sko,pre}}\big(\neg\,\psi\big)\Big)\Big)$$

   until $\bigwedge_{1 \leqslant i \leqslant k} \theta_i$ is unsatisfiable; // (as a wff of PL)
4. **output**: ψ is valid; // (as a wff of FOL)

Gilmore's algorithm

Assume equality symbol '$\approx$' does not occur in ψ for simplicity.
Details for how to proceed when '$\approx$' occurs are in lecture notes.

1. **input**: first-order sentence ψ to be tested for validity;
2. $k := 0$;
3. **repeat** $k := k + 1$
   generate first k wff's $\{\theta_1, \ldots, \theta_k\}$ in:

   $$\boxed{\text{FOL} \mapsto \text{PL}}\Big(\text{H\_Expansion}\Big(\boxed{\text{sko,pre}}\big(\neg\,\psi\big)\Big)\Big)$$

   until $\bigwedge_{1 \leqslant i \leqslant k} \theta_i$ is unsatisfiable; // (as a wff of PL)
4. **output**: ψ is valid; // (as a wff of FOL)

- **Fact**: Gilmore's algorithm terminates iff the input sentence ψ is valid.
- **Major Drawback**: Gilmore's algorithm is highly inefficient; in particular, its performance depends on the order in which the θ_i's are generated.

Gilmore's algorithm

- **Exercise**: Let $\varphi_1, \ldots, \varphi_n$ and ψ be first-order sentences.
  Define an algorithm based on Gilmore's algorithm which terminates iff the semantic entailment $\varphi_1, \ldots, \varphi_n \models \psi$ holds.
- **Problem**: Can you define an algorithm $\mathcal{A}$ which, given a first-order sentence ψ, always terminates and decides whether ψ is valid or not valid? *Hint*: No.

Gilmore's algorithm

- **Exercise**: Let $\varphi_1, \ldots, \varphi_n$ and ψ be first-order sentences.
  Define an algorithm based on Gilmore's algorithm which terminates iff the semantic entailment $\varphi_1, \ldots, \varphi_n \models \psi$ holds.
- **Problem**: Can you define an algorithm $\mathcal{A}$ which, given a first-order sentence ψ, always terminates and decides whether ψ is valid or not valid? *Hint*: No.

- Gilmore's algorithm is said to be a semi-decision procedure, because it terminates only if the input ψ is valid.
- Gilmore's algorithm was invented in the late 1950's and it was the best semi-decision procedure for first-order validity until the mid-1960's, when more efficient early versions of the tableaux and resolution methods were first introduced.

(THIS PAGE INTENTIONALLY LEFT BLANK)