Computational Logic Lab II

Camilo Thorne Sergio Tessaris

4/3/2011 (due by 11/3/2011)

Normal Forms. Let $SF(\varphi)$ denote the set of all the subformulas of a propositional formula φ. The *definitorial form* $defi(\varphi)$ of a propositional formula φ can be computed with the following algorithm:

Algorithm 1 Computing the definitorial form of φ

```
1: procedure DEFI(φ)
2:     for ψ ∈ SF(φ) \ Atom(φ) do
3:         χ ← p_ψ ⇔ ψ;
4:         φ ← φ[p_ψ/ψ];
5:         ξ ← φ ∧ χ;
6:     end for
7:     return ξ;
8: end procedure
```

Notice that p_ψ, for $\psi \in SF(\varphi)$, is a "fresh" atom (i.e., $p_\psi \notin Atom(\varphi)$). The definitorial form transformation is satisfiability-preserving, but not equivalence-preserving.

1. Consider the negation normal form $nnf(\cdot)$ or the disjunctive $dnf(\cdot)$ or conjunctive $cnf(\cdot)$ normal form transformations.
   i. Prove, by induction on φ, that φ is equivalent to
   $$nnf(\varphi), \quad cnf(\varphi) \quad \text{and} \quad dnf(\varphi).$$
   ii. Prove that φ is equisatisfiable to $defi(\varphi)$.
   iii. Show by a counterexample that they are not equivalent. Does any direction of the equivalence actually hold? If so, prove it.
   iv. Show that $cnf(\varphi)$ is exponential in $|\varphi|$, but $cnf(defi(\varphi))$ is polynomial in $|\varphi|$.

2. Define two OCaml functions `equisat` and `equiv` that check whether two propositional formulas are

   i. equisatisfiable or
   ii. equivalent

   and use them to check over a convenient sample formula φ that *defi*(φ) is equisatifiable but not equivalent to φ.

3. Define an OCaml function `opt-definitorial` that computes the optimized definitorial form (see the course slides!).

Propositional Logic Syntax. Extend Harrison's OCaml propositional formula parser to cover formulas built out of English expressions, viz., like

$$(\text{“Mary is happy”} \vee \neg \text{“Mary is happy”}) \wedge \text{“hell broke loose”}.$$

(Optional) SAT Solving and Model Building. Consider the (undirected) graph G_s defined by

- Vertexes: South American countries.
- Edges: Any two bordering countries.

We recall that a graph (V, E) is *k-colorable* if there exists a function $c\colon V \to \{1, \ldots, k\}$ s.t. if $(v, v') \in E$, $c(v) \neq c(v')$. With the aid of a SAT solver (or a propositional model builder) automatically find, or otherwise verify, a 3-coloring for G_s.