1 DPL Representations

Consider the sentence (1) with its DPL representation (2).

(1) If Pedro owns a donkey, he beats it.

(2) $(\exists x.\mathsf{donkey}(x) \wedge \mathsf{own}(p^*, x)) \rightarrow \mathsf{beat}(p^*, x)$

(a) Determine the denotation of (2) using the definitions from the lecture. Simplify your result as much as possible.

(b) Consider the following alternative DPL representations of (2).

(3) $(\neg\exists x.(\mathsf{donkey}(x) \wedge \mathsf{own}(p^*, x))) \vee \mathsf{beat}(p^*, x)$

(4) $\forall x.(\mathsf{donkey}(x) \wedge \mathsf{own}(p^*, x) \rightarrow \mathsf{beat}(p^*, x))$

Determine which of (2), (3), and (4) are equivalent or statically equivalent to each other. Explain why you believe in each equivalence or non-equivalence. You can justify your answers either by computing the denotations, or by general considerations.

2 Equivalence

We showed in the lecture that the connectives $\vee$, $\rightarrow$, and $\forall$ can be defined in DPL using the connectives $\neg$, $\wedge$, and $\exists$. The converse is false; in particular, none of the following claims of (static) equivalence are true for all formulas φ, ψ.

(a) $\varphi \wedge \psi \;\Leftrightarrow\; \neg(\varphi \rightarrow \neg\psi)$

(b) $\varphi \wedge \psi \;\Leftrightarrow_S\; \neg(\neg\varphi \vee \neg\psi)$

(c) $\varphi \rightarrow \psi \;\Leftrightarrow_S\; \neg\varphi \vee \psi$

(d) $\exists x.\varphi \;\Leftrightarrow\; \neg\forall x\neg\varphi$

Explain why each claim is false, and give formulas for φ and ψ that illustrate this. Feel free to compute denotations where this is useful for you, but you can also argue more generally.

3 Entailment

Determine whether $\varphi \models_S \psi$, $\varphi \leq \psi$, or $\varphi \models \psi$ generally hold if

(a) φ is of the form $\neg A$ and ψ is of the form $\neg\neg\neg A$;

(b) φ is of the form $A \wedge B$ and ψ is of the form A;

(c) φ is of the form $(A \rightarrow B) \wedge A$ and ψ is of the form B.

Justify your claims, and give examples for each negative claim. Feel free to compute denotations where this is useful for you, but you can also argue more generally.

4 DRT to PL via DPL

In the lectures about DRT, we showed how each DRS and each condition can be translated into a formula of standard predicate logic that is satisfied by the same models using a translation T.

Now put DPL into this picture by doing the following:

(a) Give a translation T_1 from DRT to DPL with the following properties:

- T_1 translates a DRS K into a formula $T_1(K)$ such that K and $T_1(K)$ are true in the same models.
- T_1 translates a condition C into an externally static formula $T_1(C)$ such that C and $T_1(C)$ are true in the same models.
- T_1 is as simple as possible.

(b) Give a translation T_2 from DPL to standard predicate logic such that T_2 translates a DPL formula φ into a formula $T_2(\varphi)$ that is true in the same models. It is sufficient to define T_2 on the results of the translation T_1.

(c) Argue briefly that for any DRS K, $T(K) = T_2(T_1(K))$.

(d) * Extend T_2 to a translation T_2' that works for *all* DPL formulas.

5 * "Static" formulas

The satisfaction set $\backslash\varphi\backslash_M$ of a DPL formula and the set $S_M(\varphi) = \{g \mid [\![\varphi]\!]^{M,g} = 1\}$ of all satisfying variable assignments of a formula of standard predicate logic are closely related notions. They are identical for many formulas φ – but not for all. For example, the satisfaction set of $\exists x((\exists x.P(x)) \wedge \neg P(x))$ is always empty because this formula is unsatisfiable in DPL. But in some models the set of satisfying variable assignments is non-empty, because the variables in $P(x)$ and $\neg P(x)$ are bound by different quantifiers in standard predicate logic.

Characterise the set of all formulas φ for which $\backslash\varphi\backslash_M = S_M(\varphi)$ for all M.

To be turned in by 16/06/2005, 11:15 am