Giorgi Japaridze

Theory of Computability

PSPACE-Completeness

Section 8.3

PSPACE-completeness defined

Definition 8.8 A language **B** is ***PSPACE-complete*** iff it satisfies two conditions:

1. **B** is in PSPACE, and
2. every **A** in PSPACE is polynomial time reducible to **B**.

If **B** merely satisfies condition 2, we say that it is ***PSPACE-hard***.

Why do we still appeal to polynomial time reducibility and not, say, polynomial space reducibility, philosophically speaking?

A reduction must be *easy* relative to the class (of difficult problems) that we are defining. Only then it is the case that if we find an easy way to solve a (PSPACE-, NP- or whatever-) complete problem, easy solutions to other (reducible to it) problems would also be found. If the reduction itself is hard, it does not at all offer an easy way to solve problems.

The TQBF problem

Universal quantifier $\forall$: $\forall x P(x)$ means "for any $x \in \{0,1\}$, $P(x)$ is true"

Existential quantifier $\exists$: $\exists x P(x)$ means "for some $x \in \{0,1\}$, $P(x)$ is true"

We consider ***fully quantified Boolean formulas*** (in the *prenex* form). These are Boolean formulas prefixed with either $\forall x$ or $\exists x$ for each variable x.

Examples (true or false?):

$\forall x(x \vee \underline{x})$

$\exists x(x \vee \underline{x})$

$\exists x(x \wedge \underline{x})$

$\exists x \exists y(x \wedge y)$

$\forall x \forall y\ (x \vee y)$

$\forall x \exists y\ ((x \wedge y) \vee (\underline{x} \wedge \underline{y}))$

$\exists x \forall y\ ((x \wedge y) \vee (\underline{x} \wedge \underline{y}))$

$\exists z \forall x \exists y\ ((x \wedge y \wedge z) \vee (\underline{x} \wedge \underline{y} \wedge z))$

TQBF = $\{\langle \phi \rangle \mid \phi$ is a true fully quantified Boolean formula$\}$

(**T**rue **Q**uantified **B**oolean **F**ormulas)