CS 511, Fall 2020, Lecture Slides 11

Quantified Boolean Formulas (QBF's)

Assaf Kfoury

September 22, 2020

Syntax of QBF's

- BNF definition of QBF's:

$$\varphi ::= \bot \mid \top \mid x \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \to \varphi) \mid (\forall x\, \varphi) \mid (\exists x\, \varphi)$$

where x ranges over *propositional variables*. [1]

[1] We do not say *propositional atoms* in order to emphasize that x can be quantified.

Syntax of QBF's

- BNF definition of QBF's:

$$\varphi ::= \bot \mid \top \mid x \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \to \varphi) \mid (\forall x\, \varphi) \mid (\exists x\, \varphi)$$

  where x ranges over *propositional variables*. [1]

- free and bound variables:
  - a variable x may occur **free** or **bound** in a WFF φ
  - if x is bound in φ, then there are **zero** or more **bound** occurrences of x and **one** or more **binding** occurrences of x in φ
  - a **binding** occurrence of x is of the form "$\forall x$" or "$\exists x$"
  - if a binding occurrence of x occurs as $(\mathbf{Q}\, x\, \varphi)$ where $\mathbf{Q} \in \{\forall, \exists\}$, then φ is the **scope** of the binding occurrence

[1] We do not say *propositional atoms* in order to emphasize that x can be quantified.

Syntax of QBF's

- scopes of two binding occurrences "$\mathbf{Q}\,x$" and "$\mathbf{Q}'\,x'$" may be

  disjoint: $\cdots\ (\mathbf{Q}\,x \underbrace{\cdots\ \cdots})\ \cdots\ (\mathbf{Q}'\,x' \underbrace{\cdots\ \cdots})\ \cdots$

  or **nested**: $\cdots\ (\mathbf{Q}\,x \underbrace{\cdots\ (\mathbf{Q}'\,x' \underbrace{\cdots\ \cdots})\ \cdots})\ \cdots$

  but cannot **overlap**

Syntax of QBF's

- scopes of two binding occurrences "$\mathbf{Q}\,x$" and "$\mathbf{Q}'\,x'$" may be

  disjoint: $\quad \cdots\ (\mathbf{Q}\,x \underbrace{\cdots\ \cdots}) \ \cdots\ (\mathbf{Q}'\,x' \underbrace{\cdots\ \cdots}) \ \cdots$

  or **nested**: $\quad \cdots\ \underbrace{(\mathbf{Q}\,x \cdots\ (\mathbf{Q}'\,x' \underbrace{\cdots\ \cdots})\ \cdots)}\ \cdots$

  but cannot **overlap**

- We define a function FV( ) which collects all the variables occurring **free** in a WFF. Formally:

$$
\mathsf{FV}(\varphi) = \begin{cases} \varnothing & \text{if } \varphi = \bot \text{ or } \top \\ \{x\} & \text{if } \varphi = x \\ \mathsf{FV}(\varphi') & \text{if } \varphi = \neg\varphi' \\ \mathsf{FV}(\varphi_1) \cup \mathsf{FV}(\varphi_2) & \text{if } \varphi = (\varphi_1 \star \varphi_2) \text{ and } \star \in \{\wedge, \vee, \to\} \\ \mathsf{FV}(\varphi') - \{x\} & \text{if } \varphi = (\mathbf{Q}x\ \varphi') \text{ and } \mathbf{Q} \in \{\forall, \exists\} \end{cases}
$$

Note: If x has a bound occurrence in φ, it does not follow that $x \notin \mathsf{FV}(\varphi)$.

Syntax of QBF's

- φ is **closed** iff $\mathsf{FV}(\varphi) = \varnothing$

Syntax of QBF's

- φ is **closed** iff $\mathsf{FV}(\varphi) = \varnothing$
- is the WFF φ of the form:

$$\varphi = \quad \cdots \Big(\mathbf{Q}_1\, x\, (\cdots\ x \cdots)\Big) \cdots \Big(\mathbf{Q}_2\, x\, (\cdots\ x \cdots)\Big) \cdots$$

where $\mathbf{Q}_1, \mathbf{Q}_2 \in \{\forall, \exists\}$, equivalent to:

$$\varphi' = \quad \cdots \Big(\mathbf{Q}_1\, x\, (\cdots\ x \cdots)\Big) \cdots \Big(\mathbf{Q}_2\ \underset{\uparrow}{x'}\ (\cdots\ \underset{\uparrow}{x'}\ \cdots)\Big) \cdots \ \mathbf{??}$$

Syntax of QBF's

- φ is **closed** iff $\mathsf{FV}(\varphi) = \varnothing$
- is the WFF φ of the form:

$$\varphi = \quad \cdots \Big(\mathbf{Q}_1 \, x \, (\cdots \; x \cdots) \Big) \cdots \Big(\mathbf{Q}_2 \, x \, (\cdots \; x \cdots) \Big) \cdots$$

where $\mathbf{Q}_1, \mathbf{Q}_2 \in \{\forall, \exists\}$, equivalent to:

$$\varphi' = \quad \cdots \Big(\mathbf{Q}_1 \, x \, (\cdots \; x \cdots) \Big) \cdots \Big(\mathbf{Q}_2 \; \underset{\uparrow}{x'} \; (\cdots \; \underset{\uparrow}{x'} \; \cdots) \Big) \cdots \; ??$$

- **YES**, φ and φ' are equivalent

  Question: What are the advantages of φ' over φ?

  Question: Can you write a procedure to transform φ into φ'?

Syntax of QBF's

- **Examples of QBF's:**

1. a **closed** QBF (*all* occurrences of prop variables are **bound**):[2]

$$\varphi_1 \triangleq \forall x.\ (x \vee \exists y.\ (y \vee \neg x))$$

2. an **open** QBF (*some* occurrences of propositional variables are **free**):

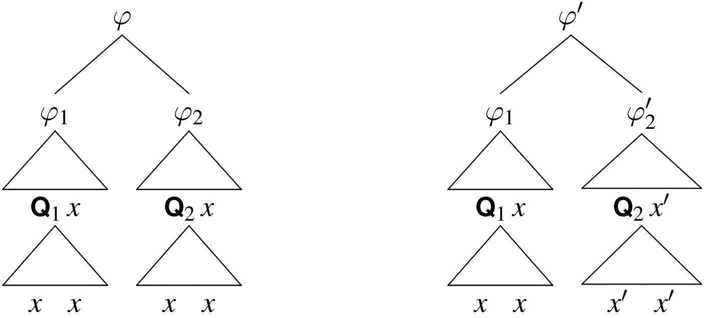
$$\varphi_2 \triangleq (\varphi_1) \wedge (x \to y) = (\varphi_1') \wedge (x \to y)$$

φ_1' is φ_1 after renaming x and y to x' and y'
(what is good about this renaming??)

[2] Note the convention, for better readability, of using "." which is not part of the formal syntax to separate a quantifier from its scope and omit the outer matching parentheses, *i.e.*, we write $\forall x.\ \varphi$ instead $(\forall x\ \varphi)$.

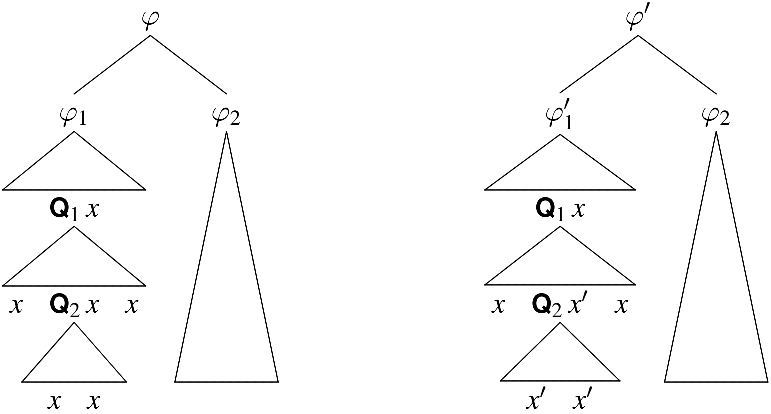
Syntax of QBF's

renaming binding occurrences "$\mathbf{Q}_1\, x$" and "$\mathbf{Q}_2\, x$" in **disjoint** scopes



Syntax of QBF's

renaming binding occurrences "$\mathbf{Q}_1\, x$" and "$\mathbf{Q}_2\, x$" in **nested** scopes

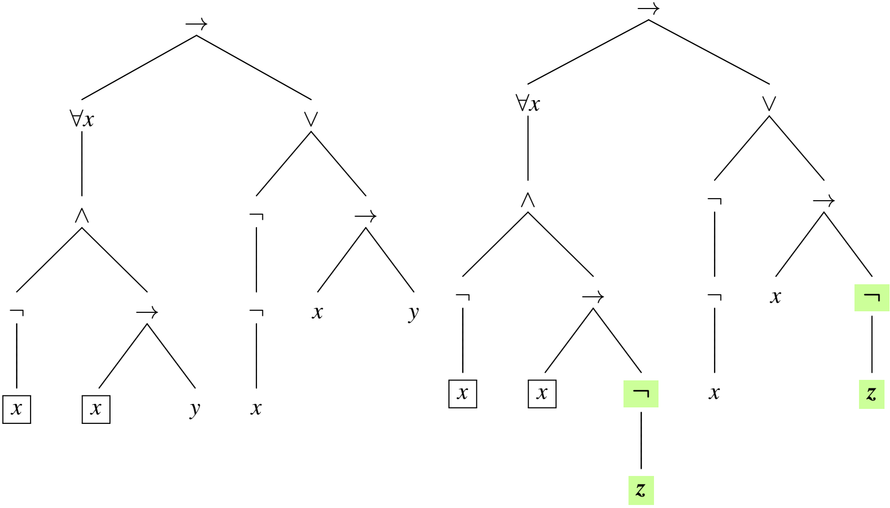


substitution examples in $\varphi = (\forall x\ (\neg x \wedge (x \to y))) \to (\neg\neg x \vee (x \to y))$

substitute $\neg z$ for y in φ: $\varphi[(\neg z)/y]$ or, less ambiguously, $\varphi[y := \neg z]$ or $\varphi[y \leftarrow \neg z]$

substitution examples in $\varphi = (\forall x\ (\neg x \wedge (x \to y))) \to (\neg\neg x \vee (x \to y))$

substitute $\neg z$ for y in φ: $\varphi[(\neg z)/y]$ or, less ambiguously, $\varphi[y := \neg z]$ or $\varphi[y \leftarrow \neg z]$



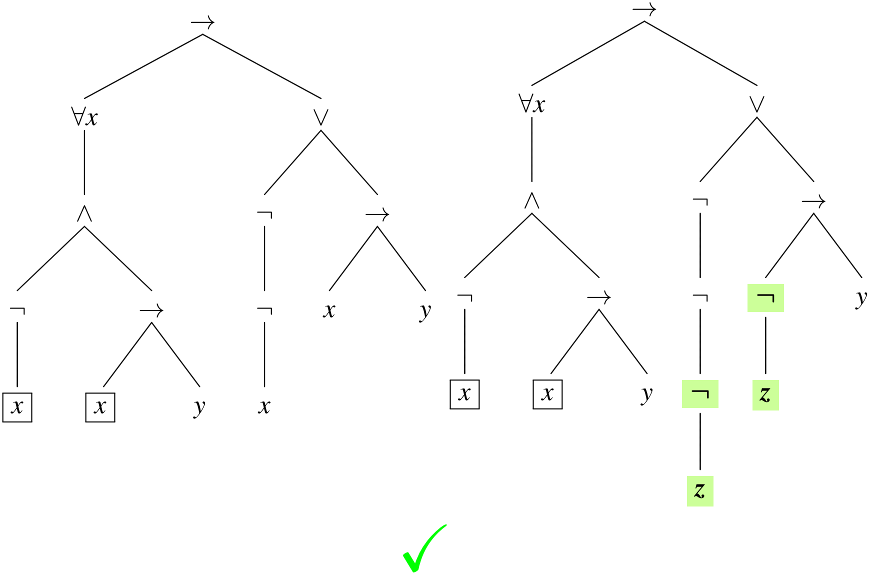
✓

substitution examples in $\varphi = (\forall x\ (\neg x \wedge (x \to y))) \to (\neg\neg x \vee (x \to y))$

substitute $\neg z$ for x in φ: $\varphi[(\neg z)/x]$

substitution examples in $\varphi = (\forall x\,(\neg x \wedge (x \to y))) \to (\neg\neg x \vee (x \to y))$

substitute $\neg z$ for x in φ: $\varphi[(\neg z)/x]$

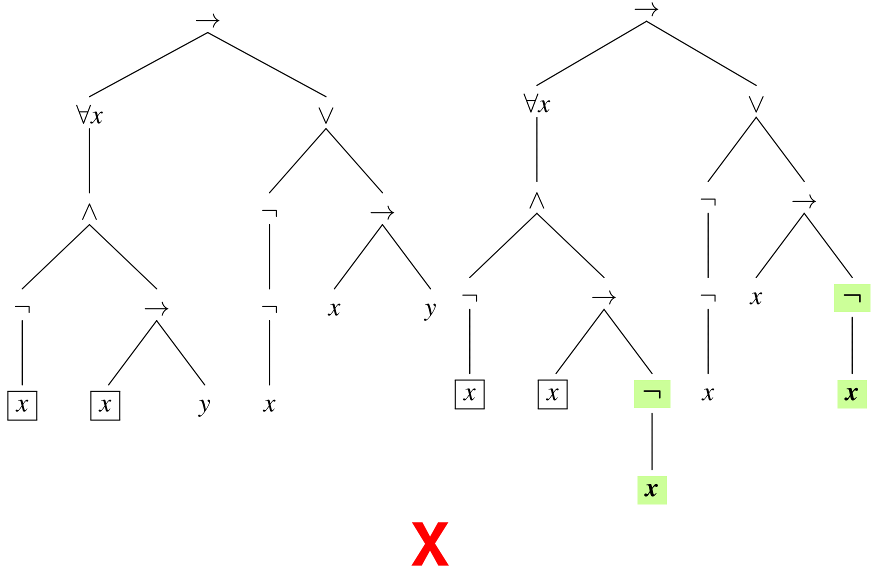


substitution examples in $\varphi = (\forall x\ (\neg x \wedge (x \to y))) \to (\neg\neg x \vee (x \to y))$

substitute $\boldsymbol{\neg x}$ for y in φ: $\varphi[(\neg x)/y]$

substitution examples in $\varphi = (\forall x\ (\neg x \wedge (x \to y))) \to (\neg\neg x \vee (x \to y))$

substitute $\neg x$ for y in φ: $\varphi[(\neg x)/y]$



Syntax of QBF's: substitution in general

- Precise definition of substitution in general for **QBF's** where u here is: $\top$, or $\bot$, or a propositional variable:

$$\varphi[u/x] = \begin{cases} \varphi & \text{if } \varphi = \top \text{ or } \bot \\ \varphi & \text{if } \varphi = y \text{ and } x \neq y \\ u & \text{if } \varphi = y \text{ and } x = y \\ \neg(\varphi'[u/x]) & \text{if } \varphi = \neg\varphi' \\ \varphi_1[u/x] \star \varphi_2[u/x] & \text{if } \varphi = \varphi_1 \star \varphi_2 \text{ and} \\ & \star \in \{\wedge, \vee, \rightarrow\} \\ \mathbf{Q}y\,(\varphi'[u/x]) & \text{if } \varphi = \mathbf{Q}y\,\varphi', \\ & \mathbf{Q} \in \{\forall, \exists\},\, x \neq y, \text{ and} \\ & u \text{ is substitutable for } x \text{ in } \varphi \\ \varphi & \text{if } \varphi = \mathbf{Q}y\,\varphi', \\ & \mathbf{Q} \in \{\forall, \exists\},\, x = y \end{cases}$$

Syntax of QBF's

- **Exercise:** The formal definition of substitution on page 18 can be simplified if every QBF is such that:

  1. there is at most one **binding** occurrence for the same variable,
  2. a variable cannot have both **free** and **bound** occurrences.

  Formalize this idea.

  Hint: You first need to modify the BNF definition on page 2, so that well-formed QBF's are defined simultaneously with $\mathrm{FV}(\ )$.

Why Study QBF's?

1. **theoretical reasons:**
   deciding **validity of QBF's** (sometimes referred to as the *QBF problem* and abbreviated as TQBF for "True QBF") is the archetype PSPACE-complete problem, just as **satisfiability of propositional WFF's** (the SAT problem) is the archetype NP-complete problem.

   (See vast literature relating QBF's to complexity classes.)

2. **practical reasons:**
   QBF's provide an alternative to propositional WFF's which are often cumbersome and space-inefficient in formal modeling of systems.
   trade-off: QBF's are more expressive than propositional WFF's, but harder to decide their validity.

3. **pedagogical reasons:**
   the study of QBF's makes the transition from propositional logic to first-order logic a little easier.

   caution: QBF's are **not** part of first-order logic (why?), **QBF logic** and **first-order logic** extend propositional logic in different ways. Nonetheless:

   Exercise: There is a way of embedding QBF logic into first-order logic, by introducing appropriate binary predicate symbols and . . .

Formal Proof Systems for QBF's

- a **natural deduction** proof system for QBF's is possible and consists of:
  - all the proof rules of natural deduction for propositional logic
  - proof rules for **universal quantification**: "$\forall x$ E" and "$\forall x$ I" (slide 22)
  - proof rules for **existential quantification**: "$\exists x$ E" and "$\exists x$ I" (slide 24)
- **Hilbert-style proof systems** are also possible (with *axioms schemes* and *inference rules*, not discussed here)
- **tableaux**-based proof systems are also possible (with additional *expansion rules* for the quantifiers, not discussed here)
- **resolution**-based proof systems for QBF's are also possible, after transforming QBF's into **conjunctive normal form** (CNF) – *more on QBF's in CNF later*
- **QBF-solvers** are implemented algorithms to decide **validity** of **closed** QBF's (**validity** and **satisfiability** of **closed** QBF's coincide, not **open** QBF's – why?).

  (Development of **QBF-solvers** *is currently far behind that of* **SAT-solvers***.)*

two proof rules for universal quantification

- universal quantifier elimination

$$\frac{\forall x\ \varphi}{\varphi[t/x]}\ \forall x\ \mathsf{E}$$

(where t is $\top$ or $\perp$ or a variable y, provided y is substitutable for x)

two proof rules for universal quantification

- universal quantifier elimination

$$\frac{\forall x\ \varphi}{\varphi[t/x]}\ \forall x\ \mathsf{E}$$

(where t is $\top$ or $\bot$ or a variable y, provided y is substitutable for x)

- universal quantifier introduction

$$\frac{\boxed{\begin{array}{lr} x_0 & \text{fresh} \\ \vdots & \\ \varphi[x_0/x] & \end{array}}}{\forall x\ \varphi}\ \forall x\ \mathsf{I}$$

two proof rules for existential quantification

- existential quantifier introduction

$$\frac{\varphi[t/x]}{\exists x\, \varphi}\ \exists x\ \mathsf{I}$$

(where t is $\top$ or $\perp$ or a variable y, provided y is substitutable for x)

two proof rules for existential quantification

- existential quantifier introduction

$$\frac{\varphi[t/x]}{\exists x\ \varphi}\ \exists x\ \mathrm{I}$$

(where t is $\top$ or $\perp$ or a variable y, provided y is substitutable for x)

- existential quantifier elimination

$$\dfrac{\exists x\ \varphi \qquad \boxed{\begin{array}{lll} x_0 & & \text{fresh} \\ & \varphi[x_0/x] & \text{assumption} \\ & \vdots & \\ & \chi & \end{array}}}{\chi}\ \exists x\ \mathrm{E}$$

(x_0 cannot occur outside its box, in particular, it cannot occur in χ)

- **Note**: Rule $(\exists x\ \mathrm{E})$ introduces both a **fresh** variable and an **assumption**.

Formal Semantics for QBF's

Let $\mathcal{V}$ be a set of propositional variables.

- A valuation (or interpretation or model) of $\mathcal{V}$ is a map $\mathcal{I} : \mathcal{V} \to \{\mathbf{T}, \mathbf{F}\}$.
- Interpretation of wff's is by induction on the (inductive) BNF definition on page 2:
  - $\mathcal{I} \models \top$ and $\mathcal{I} \not\models \bot$
  - $\mathcal{I} \models x$ iff $\mathcal{I}(x) = \mathbf{T}$
  - $\mathcal{I} \not\models x$ iff $\mathcal{I}(x) = \mathbf{F}$
  - $\mathcal{I} \models \neg\varphi$ iff $\mathcal{I} \not\models \varphi$
  - $\mathcal{I} \models \varphi \wedge \psi$ iff $\mathcal{I} \models \varphi$ **and** $\mathcal{I} \models \psi$
  - $\mathcal{I} \models \varphi \vee \psi$ iff $\mathcal{I} \models \varphi$ **or** $\mathcal{I} \models \psi$
  - $\mathcal{I} \models \varphi \to \psi$ iff $\mathcal{I} \models \psi$ **whenever** $\mathcal{I} \models \varphi$
  - $\mathcal{I} \models \forall x\, \varphi$ iff $\mathcal{I} \models \varphi[x := \top]$ **and** $\mathcal{I} \models \varphi[x := \bot]$
  - $\mathcal{I} \models \exists x\, \varphi$ iff $\mathcal{I} \models \varphi[x := \top]$ **or** $\mathcal{I} \models \varphi[x := \bot]$
- For sets Δ, Γ of wff's: $\mathcal{I}$ *is a model of* Δ, written $\mathcal{I} \models \Delta$, iff $\mathcal{I} \models \varphi$ for all $\varphi \in \Delta$.

  Δ *semantically entails* Γ, written $\Delta \models \Gamma$, iff every model $\mathcal{I}$ of Δ is a model of Γ.

Formal Semantics for QBF's (continued)

Useful connections between **closed** QBF's and **open** QBF's
(a special case of **open** QBF's are the propositional WFF's):

Theorem

Let φ be a QBF with free variables FV$(\varphi) = \{x_1, \ldots, x_n\}$. We then have:

- φ *is satisfiable iff the* **closed** *formula* $\exists x_1 \cdots \exists x_n . \varphi$ *is satisfiable.*
- φ *is valid iff the* **closed** *formula* $\forall x_1 \cdots \forall x_n . \varphi$ *is satisfiable.*

Formal Semantics for QBF's (continued)

Theorem

For **closed** *QBF's, the notions of* **truth** *(semantic* **validity***),* **formal deducibility***, and* **satisfiability** *all coincide.*

Specifically, given a **closed** *QBF φ, the following are equivalent statements:*

1. *φ is satisfiable.*
2. *φ is valid.*
3. *$\mathcal{I} \models \varphi$ for some valuation $\mathcal{I} : \mathcal{V} \to \{\mathbf{T}, \mathbf{F}\}$.*
4. *$\mathcal{I} \models \varphi$ for every valuation $\mathcal{I} : \mathcal{V} \to \{\mathbf{T}, \mathbf{F}\}$.*

Because φ is closed and $FV(\varphi) = \varnothing$, the last two statements are equivalent to one:

5. *$\models \varphi$ (there is no mention of a valuation $\mathcal{I}$)*

There is also a **Soundness Theorem**, a **Compactness Theorem**, and a **Completeness Theorem**, all proved as they were for the propositional logic.

Prenex Form of QBF's

1. $(\mathbf{Q}_1 x_1\ \varphi_1)\ \otimes\ (\mathbf{Q}_2 x_2\ \varphi_2)$ transformed to $\mathbf{Q}_1 x_1\ \mathbf{Q}_2 x_2\ (\varphi_1 \otimes \varphi_2)$

   where $\mathbf{Q}_1, \mathbf{Q}_2 \in \{\forall, \exists\}$ and $\otimes \in \{\wedge, \vee\}$, provided

   x_1 is not free in φ_1 and x_2 is not free in φ_2.

$1a.$ special case of case 1 (for better QBF-solver performance):

$(\forall x_1\ \varphi_1)\ \wedge\ (\forall x_2\ \varphi_2)$ transformed to $\forall x_1\ \big(\varphi_1 \wedge \varphi_2[x_2 := x_1]\big)$

$1b.$ special case of case 1 (for better QBF-solver performance):

$(\exists x_1\ \varphi_1)\ \vee\ (\exists x_2\ \varphi_2)$ transformed to $\exists x_1\ \big(\varphi_1 \vee \varphi_2[x_2 := x_1]\big)$

2. $(\forall x\ \varphi)\ \rightarrow\ \psi$ transformed to $\exists x\ (\varphi\ \rightarrow\ \psi)$ provided x not free in ψ.

3. $(\exists x\ \varphi)\ \rightarrow\ \psi$ transformed to $\forall x\ (\varphi\ \rightarrow\ \psi)$ provided x not free in ψ.

4. $\varphi\ \rightarrow\ (\mathbf{Q} x\ \psi)$ transformed to $\mathbf{Q} x\ (\varphi\ \rightarrow\ \psi)$ provided x not free in φ.

5. $\neg(\exists x\ \varphi)$ transformed to $\forall x\ (\neg\varphi)$

6. $\neg(\forall x\ \varphi)$ transformed to $\exists x\ (\neg\varphi)$

Conjunctive Normal Form & Disjunctive Normal Form

- A QBF φ is in

  prenex conjunctive normal form (PCNF) or

  prenex disjunctive normal form (PDNF)

  iff φ is in **prenex form** and its **matrix** is a CNF or a DNF, respectively.

- Generally, validity/satisfiability methods for QBF's

  (tableaux, resolution, QBF solvers, etc.)

  perform best on PCNF (resp. PDNF) if their counterparts for propositional WFF's perform best on CNF (resp. DNF).

- QBF solvers require input WFF φ be transformed into PCNF,

  (the **matrix** of φ is transformed into an **equisatisfiable**, rather than an **equivalent**, propositional WFF to avoid exponential explosion).

- **Warning:** Transformation of a QBF φ into a PCNF ψ (or PDNF ψ) is non-deterministic. Special methods have been developed (and are being developed) for minimizing number of quantifiers and quantifier alternations in the prenex of ψ, for improved performance of QBF-solvers.

transformation of QBF's for better QBF-solver performance

1. introduce abbreviations for subformulas

   - **example** : consider a formula Φ of the form

     $$\Phi = (\varphi \vee \psi_1) \wedge (\varphi \vee \psi_2) \wedge (\varphi \vee \psi_3)$$

   - if we abbreviate (*i.e.*, represent) φ by the fresh variable y, we can write

     $$\Psi = \exists y.\ (y \leftrightarrow \varphi) \wedge (y \vee \psi_1) \wedge (y \vee \psi_2) \wedge (y \vee \psi_3)$$

   - **exercise** : Φ and Ψ are logically equivalent
   - **advantage** of Ψ over Φ:
     subformula φ occurs once (in Ψ) instead of three times (in Φ) for the price of two logical connectives $\{$ "$\wedge$", "$\leftrightarrow$" $\}$ and one propositional variable $\{$ "y" $\}$

transformation of QBF's for better QBF-solver performance

2. unify instances of the same subformula

   - **example** : consider a formula Φ of the form

   $$\Phi = \theta(\varphi_1, \psi_1) \ \wedge\ \theta(\varphi_2, \psi_2) \ \wedge\ \theta(\varphi_3, \psi_3)$$

   - unify the three occurrences of the subformula θ, and introduce fresh variables x and y to represent the φ_i's and the ψ_i's, resp., to obtain:

   $$\Psi = \forall x.\ \forall y.\ \Big(\bigvee_{i=1,2,3} (x \leftrightarrow \varphi_i) \wedge (y \leftrightarrow \psi_i) \Big) \ \rightarrow\ \theta(x, y)$$

   - **exercise** : Φ and Ψ are logically equivalent

3. for many other transformations, for better QBF-solver performance, see:
   U. Bubeck and H. Büning, "Encoding Nested Boolean Functions as QBF's", in
   J. on Satisfiability, Boolean Modeling and Computation, Vol. 8 (2012), pp. 101-116

QBF as a game

A **closed prenex QBF formula** φ can be viewed as a game between an existential player (Player $\exists$) and a universal player (Player $\forall$):

- Existentially quantied variables are owned by Player $\exists$.
- Universally quantied variables are owned by Player $\forall$.
- On each turn of the game, the owner of an outermost unassigned variable assigns it a truth value (**T** or **F**).
- The goal of Player $\exists$ is to make φ be **T**.
- The goal of Player $\forall$ is to make φ be **F**.
- A player owns a literal ℓ if the player owns FV(ℓ).

Player $\exists$ wins if $\models \varphi$,   Player $\forall$ wins if $\models \neg\varphi$.

(THIS PAGE INTENTIONALLY LEFT BLANK)