Encoding Nested Boolean Functions as Quantified Boolean Formulas

Uwe Bubeck and Hans Kleine Büning

University of Paderborn

26.9.2010

Outline

- Introduction
- Nested Boolean Functions
- Transformation from NBF To QBF
- Transformation from QBF To NBF
- Conclusion

Encoding Nested Functions

Introduction

Compact Encodings with QBF 1/3

QBF extends propositional logic with quantifiers:

$\forall x\ \phi(x)$ is true if and only if

$\phi[x/0]$ is true and $\phi[x/1]$ is true

$\exists x\ \phi(x)$ is true if and only if

$\phi[x/0]$ is true or $\phi[x/1]$ is true

$\rightarrow$ Distinction between quantified and free variables:
only free variables are directly taken into account for equivalence.

Compact Encodings with QBF 2/3

Given: propositional formula $\phi(z_1,...,z_r)$.

Wanted:

shorter QBF $\Phi(z_1,...,z_r)$ with free variables $z_1,...,z_r$

such that $\mathfrak{I}_\Phi(z_1,...,z_r) = \mathfrak{I}_\phi(z_1,...,z_r)$ („equivalence“).

QBF often allows much more compact encodings.

Well known: quantified CNF (w/ free vars) is exponentially more concise than propositional CNF.

Compact Encodings with QBF 3/3

2 Main QBF Encoding Patterns:

- Abbreviations for repeating subformulas

$$(A \vee \neg B \vee C \vee D) \wedge (A \vee \neg B \vee C \vee \neg E) \wedge (A \vee \neg B \vee C \vee F)$$

$$\approx \exists y\ (y \rightarrow (A \vee \neg B \vee C)) \wedge (y \vee D) \wedge (y \vee \neg E) \wedge (y \vee F)$$

- Combining multiple instantiations of a subformula with different arguments

$$\phi(a_{1,1}, ..., a_{1,l}) \wedge \phi(a_{2,1}, ..., a_{2,l}) \wedge ... \wedge \phi(a_{k,1}, ..., a_{k,l})$$

$$\approx \forall x_1 ... \forall x_l \left(\bigvee_{i=1}^{k} \bigwedge_{j=1}^{l} (x_j \leftrightarrow a_{i,j}) \right) \rightarrow \phi(x_1, ..., x_l)$$

Encoding Nested Functions

Nested Boolean Functions

Nested Boolean Functions 1/4

Instantiations might also be nested:

$((a_1 \wedge a_2) \vee (\neg a_1 \wedge \neg a_2)) \rightarrow ((b_1 \wedge b_2) \vee (\neg b_1 \wedge \neg b_2))$

is of the form

$\psi(\phi(a_1,a_2), \phi(b_1,b_2))$.

$\rightarrow$ Can we combine both instantiations of ϕ?

Unfortunately, the encoding pattern from the previous slide is not sufficient!

Nested Boolean Functions 2/4

Nested Boolean functions are a powerful concept.

Example: parity of n Boolean variables

$$f_0(p_1,p_2) := (\neg p_1 \wedge p_2) \vee (p_1 \wedge \neg p_2)$$
$$f_1(p_1,p_2,p_3,p_4) := f_0(f_0(p_1,p_2),f_0(p_3,p_4))$$
$$f_2(p_1,\ldots,p_{16}) := f_1(f_1(p_1,\ldots,p_4),\ldots, f_1(p_{13},\ldots,p_{16}))$$
$$\ldots$$

$\rightarrow \log_2 \log_2 n + 1$ definitions of size $O(n)$.

Propositional case (only $\wedge,\vee,\neg$): quadratic lower bound

Nested Boolean Functions 3/4

Definition

A Nested Boolean Function (NBF) is a sequence of functions $F = (f_0,...,f_k)$ with

- initial functions $f_0,...,f_t$
  with $f_i(x^i) := \alpha_i(x^i)$ for a propositional formula α_i
  over $x^i := (x^{i,1},..., x^{i,n_i})$.
- compound functions $f_{t+1},...,f_k$
  with $f_i(x^i) := f_{j_0}(f_{j_1}(x^i{}_1),...,f_{j_r}(x^i{}_r))$
  for previously defined functions $f_{j_0},...,f_{j_r} \in \{f_1,...,f_{i\text{-}1}\}$
  and $x^i{}_1,...,x^i{}_r$ being subsequences of x^i.

Nested Boolean Functions 4/4

The definition of NBF closely matches the definition of Boolean Programs [Cook/Soltys 2003].

Given a NBF $F = (f_0,...,f_k)$, the problem of deciding whether $f_k(a_1,...,a_{n_k}) = 1$ for given arguments $a_1,...,a_{n_k}$ is PSPACE-complete [Cook/Soltys 2003].

Deciding whether there exist satisfying arguments is essentially the same problem (as with QBF).

PSPACE-completeness of both NBF and QBF suggests efficient transformations in both directions.

Encoding Nested Functions

From NBF To QBF

NBF To QBF: Overview

Given NBF $F = (f_0,...,f_k)$, find poly-size QBF $\Phi(x^k) \approx f_k(x^k)$:

- indirectly via a polynomial-space Turing machine and its polynomial-size QBF encoding
- quadratic encoding by adapting a transformation from BPLK to G [Skelley 2004].
  But: Encoding of f_i contains negated encoding of $f_{i\text{-}1}$, encoding scheme needs partial unfolding of definitions.
- now: linear encoding

NBF To QBF: Idea 1/2

Idea

Combination of basic encoding patterns:

- universal quantifiers represent different attributes
- existential quantifiers store results of instantiations

Example: $\psi(\phi(a_1), \phi(a_2))$

1. Separation of specific arguments from instantiation

$$\forall x$$

$$((x \leftrightarrow a_1) \rightarrow \phi(x)) \wedge ((x \leftrightarrow a_2) \rightarrow \phi(x))$$

NBF To QBF: Idea 1/2

Idea

Combination of basic encoding patterns:

- universal quantifiers represent different attributes
- existential quantifiers store results of instantiations

Example: $\psi(\phi(a_1), \phi(a_2))$

2. Abbreviation of $\phi(x)$

$$\forall x \; \exists y \; (\phi(x) \leftrightarrow y) \wedge$$

$$((x \leftrightarrow a_1) \rightarrow \quad y) \; \wedge ((x \leftrightarrow a_2) \rightarrow \quad y)$$

NBF To QBF: Idea 1/2

Idea

Combination of basic encoding patterns:

- universal quantifiers represent different attributes
- existential quantifiers store results of instantiations

Example: $\psi(\phi(a_1), \phi(a_2))$

3. Store intermediate results

$$\exists b_1\ \exists b_2\ \forall x\ \exists y\ (\phi(x) \leftrightarrow y) \wedge$$

$$((x \leftrightarrow a_1) \rightarrow (b_1 \leftrightarrow y)) \wedge ((x \leftrightarrow a_2) \rightarrow (b_2 \leftrightarrow y))$$

NBF To QBF: Idea 1/2

Idea

Combination of basic encoding patterns:

- universal quantifiers represent different attributes
- existential quantifiers store results of instantiations

Example: $\psi(\phi(a_1), \phi(a_2))$

3. Invoke ψ with stored results

$$\exists b_1\, \exists b_2\, \forall x\, \exists y\, (\phi(x) \leftrightarrow y) \wedge$$

$$((x \leftrightarrow a_1) \rightarrow (b_1 \leftrightarrow y)) \wedge ((x \leftrightarrow a_2) \rightarrow (b_2 \leftrightarrow y)) \wedge$$

$$\psi(b_1,b_2)$$

NBF To QBF: Idea 2/2

$$\exists b_1\ \exists b_2\ \forall x\ \exists y\ (\phi(x) \leftrightarrow y) \wedge$$
$$((x \leftrightarrow a_1) \rightarrow (b_1 \leftrightarrow y)) \wedge ((x \leftrightarrow a_2) \rightarrow (b_2 \leftrightarrow y)) \wedge$$
$$\psi(b_1,b_2)$$

Why must $\exists b_1$ and $\exists b_2$ precede $\forall x$?

$$\forall x\ \exists y\ \exists b_1\ \exists b_2\ (\phi(x) \leftrightarrow y) \wedge$$
$$((x \leftrightarrow a_1) \rightarrow (b_1 \leftrightarrow y)) \wedge ((x \leftrightarrow a_2) \rightarrow (b_2 \leftrightarrow y)) \wedge (b_1 \vee b_2)$$

$\rightarrow$ tautological regardless of ϕ, because we could choose favorable, but incorrect, values for b_i in the cases that $x \neq a_i$.

NBF To QBF: Encoding

Complete Encoding:

1. Initial functions $f_0,...,f_t$:

$$\Phi_t := \forall x^t ... \forall x^0 \; \exists y^t ... \exists y^0 \; (f_0(x^0) = y^0) \wedge ... \wedge (f_t(x^t) = y^t)$$

2. Compound function $f_i(x^i) := f_{j_0}(f_{j_1}(x^i{}_1),...,f_{j_r}(x^i{}_r))$:

$$\Phi_i := \; \forall x^i \exists y^i \exists b^{i,1} ... \exists b^{i,r} Q_{i\text{-}1}$$

$$\phi_{i\text{-}1} \wedge$$

$$(x^i{}_1 = x^{j1} \rightarrow b^{i,1} = y^{j1}) \wedge ... \wedge (x^i{}_r = x^{jr} \rightarrow b^{i,r} = y^{jr}) \wedge$$

$$((b^{i,1},...,b^{i,r}) = x^{j0} \rightarrow y^i = y^{j0})$$

3. Drop leading universal quantifiers $\forall x^k$ and add unit y^k.

Encoding Nested Functions

From QBF To NBF

QBF To NBF: Idea 1/2

Given a QBF $\Phi(z_1,...,z_r) = Q_n v_n \ ... \ Q_1 v_1 \ \phi(v_1,...,v_n,z_1,...,z_r)$,

find a polynomial-size NBF $F = (f_1,...,f_k)$, such that

$\Phi(z_1,...,z_r) \approx f_k(z_1,...,z_r)$.

Idea: Expansion of quantifiers

$\forall x \ \Phi(x,z_1,...,z_r) \approx \Phi(0,z_1,...,z_r) \wedge \Phi(1,z_1,...,z_r)$

$\exists x \ \Phi(x,z_1,...,z_r) \approx \Phi(0,z_1,...,z_r) \vee \Phi(1,z_1,...,z_r)$

$\rightarrow$ Let $g_\forall(a,b) := a \wedge b$ and $g_\exists(a,b) := a \vee b$.

QBF To NBF: Idea 2/2

Repeated expansion:

$$\forall x_2 \exists x_1 \; \phi(x_1,x_2,z) \approx g_{\forall}(g_{\exists}(\phi(0,0,z),\phi(1,0,z)),g_{\exists}(\phi(0,1,z),\phi(1,1,z))$$

$\rightarrow$ we need to avoid exponential growth!

QBF To NBF: Idea 2/2

Repeated expansion:

$$\forall x_2 \exists x_1 \; \phi(x_1,x_2,z) \approx g_\forall(g_\exists(\phi(0,0,z),\phi(1,0,z)),g_\exists(\phi(0,1,z),\phi(1,1,z))$$

$\rightarrow$ we need to avoid exponential growth!

Solution:

Define a new function after each single expansion step

$$f_1(x_2,z) := g_\exists(\phi(0,x_2,z),\phi(1,x_2,z))$$
$$f_2(z) := g_\forall(f_1(0,z),\; f_1(1,z))$$

QBF To NBF: Transformation

Given a QBF $\Phi(z) = Q_n v_n \ldots Q_1 v_1 \; \phi(v_1,\ldots,v_n,z)$, we define:

$f_0(v_1,\ldots,v_n,z) := \phi(v_1,\ldots,v_n,z)$

$g_\forall(a,b) := a \wedge b$

$g_\exists(a,b) := a \vee b$

and for $i = 1,\ldots,n$:

$f_i(v_{i+1},\ldots,v_n,z) := g_{Qi}(f_{i-1}(0,v_{i+1},\ldots,v_n,z),\ f_{i-1}(1,v_{i+1},\ldots,v_n,z))$

$\rightarrow f_n(z) \approx \Phi(z)$

and $F = (f_0,g_\forall,g_\exists,f_1,\ldots,f_n)$ has length $O(|\Phi(z)|^2)$.

Quantified NBF 1/2

The ability to simulate quantifier expansion justifies an extension of NBF:

Quantified Nested Boolean Functions (QNBF)

with definitions of the form

$$f_i(x^i) := Qv^i \; \alpha_i(v^i, x^i)$$

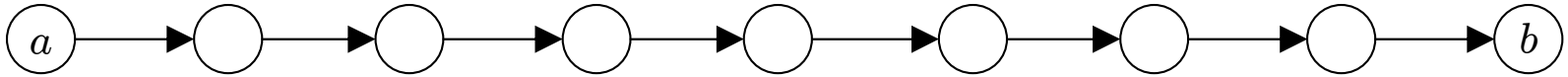
or

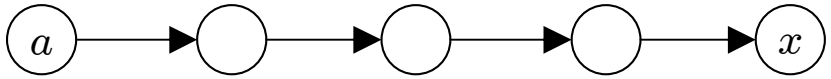
$$f_i(x^i) := Qv^i \; f_{j_0}(f_{j_1}(v^i{}_1, x^i{}_1), \ldots, f_{j_r}(v^i{}_r x^i{}_r))$$

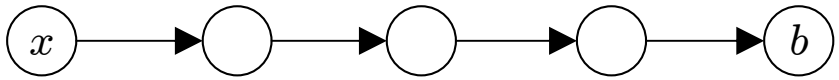
Quantified NBF 2/2

Example: iterative squaring for bounded reachability

Does a given directed graph contain a path of length at most 2^k from a to b?







$$f_0(a,b) := \delta(a,b) \vee (a{=}b)$$

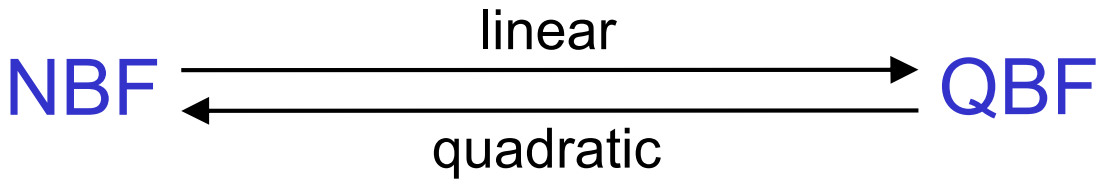
$$f_i(a,b) := \exists x\ f_{i\text{-}1}(a,x) \wedge f_{i\text{-}1}(x,b) \quad \text{for } i{=}1\ldots k$$

Encoding Nested Functions

Conclusion

Conclusion

We have presented 2 transformations:



Open Questions:

- Is there a linear-size transformation from QBF to NBF?
- What about subclasses of NBF?