Report Number: WUCS-90-03

1990-04-01

The equivalence of connectionist energy minimization and propositional calculus satisfiability

Authors: Gadi Pinkas

Quadratic energy minimization is the essence of certain connectionist models. We define high order connectionist models to support the minimization of high order energy functions and we prove that high order energy functions are equivalent to quadratic ones. We show that the standard quadratic models can minimize high order functions using additional hidden units and we demonstrate trade-offs of size (number of hidden units), order of the model, and fan-out.
We prove an equivalence between the problem of satisfiability in propositional calculus and the problem of minimization of energy functions. An energy function describes a Well Formed Formula (WFF) if the set of solutions to the minimization of the function is equal to the set of models (truth assignments) that satisfy the WFF. We show that every satisfiable WFF is described by some energy function and that every energy function describes some WFF. Algorithms are given to transform any propositional WFF into an energy function that describes it and vice versa.
A connectionist propositional inference engine that features incremental updating of the knowledge can...
Read complete abstract on page 2.

The equivalence of connectionist energy minimization and propositional calculus satisfiability

Complete Abstract:

Quadratic energy minimization is the essence of certain connectionist models. We define high order connectionist models to support the minimization of high order energy functions and we prove that high order energy functions are equivalent to quadratic ones. We show that the standard quadratic models can minimize high order functions using additional hidden units and we demonstrate trade-offs of size (number of hidden units), order of the model, and fan-out.

We prove an equivalence between the problem of satisfiability in propositional calculus and the problem of minimization of energy functions. An energy function describes a Well Formed Formula (WFF) if the set of solutions to the minimization of the function is equal to the set of models (truth assignments) that satisfy the WFF. We show that every satisfiable WFF is described by some energy function and that every energy function describes some WFF. Algorithms are given to transform any propositional WFF into an energy function that describes it and vice versa.

A connectionist propositional inference engine that features incremental updating of the knowledge can be implemented using these algorithms. The results have applications in reasoning and AI, and also give a better understanding of the limitations and the capabilities of connectionist energy minimization models.

THE EQUIVALENCE OF CONNECTIONIST ENERGY MINIMIZATION AND PROPOSITIONAL CALCULUS SATISFIABILITY

Gadi Pinkas

WUCS-90-03

April 1990

Center for Intelligent Computing Systems
Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
Saint Louis, MO 63130-4899

Abstract

Quadratic energy minimization is the essence of certain connectionist models. We define high order connectionist models to support the minimization of high order energy functions and we prove that high order energy functions are equivalent to quadratic ones. We show that the standard quadratic models can minimize high order functions using additional hidden units and we demonstrate trade-offs of size (number of hidden units), order of the model, and fan-out.

We prove an equivalence between the problem of satisfiability in propositional calculus and the problem of minimization of energy functions. An energy function describes a Well Formed Formula (WFF) if the set of solutions to the minimization of the function is equal to the set of models (truth assignments) that satisfy the WFF. We show that every satisfiable WFF is described by some energy function and that every energy function describes some WFF. Algorithms are given to transform any propositional WFF into an energy function that describes it and vice versa.

A connectionist propositional inference engine that features incremental updating of the knowledge can be implemented using these algorithms. The results have applications in reasoning and AI, and also give a better understanding of the limitations and the capabilities of connectionist energy minimization models.

The sponsors of the center are McDonnell Douglas Corporation, Southwestern Bell Corporation and Mitsubishi Electronics America, Inc.

Contents

1 Introduction 1

2 Logic and energy functions 4

2.1 Propositional calculus 4

2.2 High order energy functions and high order connectionist models 8

2.3 Energy functions describe WFFs 10

3 The equivalence between high and low order energy functions 13

3.1 Converting high order energy function into a lower order function 14

3.2 Eliminating hidden variables 15

3.3 The tradeoff between the number of hidden variables, the number of connections and the order of the energy function 17

4 The mapping between satisfiability and energy minimization 19

4.1 An economical way to convert a WFF into quadratic energy function 19

4.1.1 Equivalence between WFFs 19

4.1.2 Converting A WFF into an equivalent Conjunction of Triples Form WFF 20

4.1.3 An algorithm to transform a WFF into a second order energy function 21

4.1.4 Complexity analysis 22

4.2 Every energy function describes some satisfiable WFF. 23

4.3 Equivalence classes 25

5 Summary and conclusions 26

6 Acknowledgments 27

6 Acknowledgments 27

A Proofs 28

A.1 Proof of lemma: converting high order term with negative coefficient to low order terms. 28

A.2 Proof of lemma: converting high order term with positive coefficient to low order terms. 30

A.3 Proof of theorem: eliminating hidden variable by converting terms to possibly higher order terms. 32

A.4 Proof of theorem: the attribute grammar generates an equivalent WFF in a conjunction of triples form. 35

A.5 Proof of lemma: H_E is the characteristic function of φ. 37

1. Introduction

Certain connectionist models used for parallel constraint satisfaction are based on the minimization of quadratic energy functions [1] [1],[2], [3], [4]. Energy minimization connectionist models are massively parallel architectures that are composed of a network of simple processing units, each connected to a subset of the others. Typically each unit asynchronously computes the gradient of the energy function and adjusts its activation value, so that energy decreases monotonically. The network eventually reaches either a local or a global minimum and settles in an equilibrium. (Some models use stochastic techniques to escape from local minima [3], [5]). It has been first demonstrated by Hopfield and Tank [6], that certain complex problems can be approximated by this kind of networks. Since then, energy minimization models were used by several researchers in the area of connectionist reasoning and knowledge representation. Foe examples see: a restricted form of unit resolution [7], a production system [8], a semantic network [9].

The problem of satisfiability in propositional calculus is to decide whether there exists a truth assignment (a model) for the variables of a given propositional well formed formula (WFF), such that the formula is evaluated to be true. In many cases it is not enough just to decide whether a WFF is satisfiable or not. A truth assignment that satisfies is also desired. Many hard problems may be stated as satisfiability problems of appropriate WFFs. It is well known that any of the $\mathcal{NP}$ problems can be mapped to the problem of finding what truth assignments satisfy a certain WFF. In the area of AI for example, logic is used as a compact way to represent knowledge, and inference mechanisms are used to draw conclusions from this knowledge. Inferring what *must* be the truth values of the atomic propositions for a knowledge base to be consistent let one further decide whether novel, compound WFFs logically follow from the knowledge base or contradict it.

This report shows that the satisfiability problem in propositional calculus is equivalent to the problem of finding global minima for a "quadratic binary" function. The equivalence between these two problems means that in order to decide whether a WFF is satisfiable and to find a truth assignment that satisfies it, we can find global minima to some "quadratic binary " function such that the values of the variables of this function when the minimum is reached can be translated to truth values that satisfy the original

[1] "Quadratic energy functions" are quadratic functions that admit binary-valued arguments and return a real number.

that the values of the variables of this function when the minimum is reached can be translated to truth values that satisfy the original WFF. Also, any quadratic binary function minimization problem may be described as an equivalent satisfiable WFF that is satisfied for the same truth assignments that causes the function to reach this minima.

There is a direct translation from quadratic energy functions into connectionist energy minimization networks and vice versa. Variables map into units, coefficients of quadratic terms into weights and coefficients of one-variable terms into thresholds. Thus, the equivalence shown in this report is important both to the application of logic reasoning on massively parallel architectures and to the understanding of the limitations and capabilities of these networks.

We will conclude that

1. Propositional logic can be represented efficiently by energy minimization connectionist networks and thus may give us a fast parallel implementation of a propositional inference engine. Small incremental updates to the knowledge base is done by small changes to the connectionist network and with out the need for re-calculating the network .

2. High order connectionist models may be defined to minimize high order functions. There is a tradeoff between the size of the network and the order of the model we implement. Any WFF or any boolean function can be implemented in $n-$order model without any hidden units, but extra units are needed to implement the same WFF (or function) using a quadratic model.

3. All that can be expressed in propositional logic and *nothing more* can also be expressed in energy minimization networks.

In the following sections an algorithm is described for converting a propositional WFF into a possibly high order energy function. Then a constructive proof is given to show that high order energy functions are equivalent to quadratic energy functions with hidden variables, and that any quadratic energy function with hidden variables can be transformed into a higher order energy function with no hidden variables. This higher order energy function is then shown to be equivalent to some satisfiable WFF. An algorithm is given for converting a propositional WFF into a quadratic

energy function with number of hidden variables ("hidden units" in connectionist terminology) linear in the length of the WFF. Complexity issues are discussed throughout the report.

2. Logic and energy functions

2.1. Propositional calculus

DEFINITION 2.1 PROPOSITIONAL WELL FORMED FORMULA (WFF)

A propositional WFF over a set V of variable symbols (Atomic propositions) is defined recursively as follows:

Let φ a string of symbols, then φ is WFF if φ is either:

- $\varphi = x_i$ and x_i is a variable symbol in V
- $\varphi = (\varphi_1 \vee \varphi_2)$ and φ_1 and φ_2 are WFFs
- $\varphi = (\varphi_1 \wedge \varphi_2)$ and φ_1 and φ_2 are WFFs
- $\varphi = (\neg\varphi_1)$ and φ_1 is a WFF
- $\varphi = (\varphi_1 \rightarrow \varphi_2)$ and φ_1 and φ_2 are WFFs

Nothing else is a WFF

DEFINITION 2.2 TRUTH ASSIGNMENT

A *truth assignment* over a set V of variable symbols is defined to be a function $S : V \rightarrow \{0, 1\}$.

We define the *instantiation* $\bar{x}$ of a vector $\hat{X} = (x_1, \ldots, x_n)$ under an assignment S, as the vector $\bar{x} = (S(x_1), \cdots, S(x_n))$. $((x_1, \ldots, x_n)$ is a vector of variable symbols).

The truth assignment function is the interpretation assigned to the atomic propositions (the variables). "1" means "true" and "0" means "false". We will sometimes use the notation $\bar{x}$ to denote an instantiation (or model) of the variables by some truth assignment.

DEFINITION 2.3 CHARACTERISTIC FUNCTION

The characteristic function H_φ of a WFF φ is defined to be a boolean function: $H_\varphi : 0,1^n \rightarrow \{0,1\}$ such that:

- $H_{x_i}(x_1,\ldots,x_n) = x_i$
- $H_{(\neg\varphi)}(x_1,\ldots,x_n) = 1 - H_\varphi(x_1,\ldots,x_n)$
- $H_{(\varphi_1 \vee \varphi_2)}(x_1,\ldots,x_n) = H_{\varphi_1}(x_1,\ldots,x_n) + H_{\varphi_2}(x_1,\ldots,x_n) - H_{\varphi_1}(x_1,\ldots,x_n) \times H_{\varphi_2}(x_1,\ldots,x_n)$
- $H_{(\varphi_1 \wedge \varphi_2)}(x_1,\ldots,x_n) = H_{\varphi_1}(x_1,\ldots,x_n) \times H_{\varphi_2}(x_1,\ldots,x_n)$
- $H_{(\varphi_1 \rightarrow \varphi_2)}(x_1,\ldots,x_n) = H_{(\neg\varphi_1 \vee \varphi_2)}(x_1,\ldots,x_n)$

The characteristic function H evaluates the truth-value of the WFF given a specific instantiation. H_φ is a boolean function on $\{0,1\}^n$ into $\{0,1\}$.

EXAMPLE 2.1

$$\begin{aligned} H_{((A \vee (\neg B)) \wedge C)} &= (A + (1-B) - A(1-B))C \\ &= AC + C - BC - AC + ABC \\ &= ABC - BC + C \end{aligned}$$

DEFINITION 2.4 *Satisfiability of a WFF*

Let $\bar{x}$ be an instantiation of $(x_1,\ldots,x_n)$ by assignment S.

Then the assignment S *satisfies* φ iff $H_\varphi(\bar{x}) = 1$

If φ is satisfied by $\bar{x}$ we write: $\varphi(\bar{x}) = 1$

EXAMPLE 2.2 $\bar{x} = (0,0,1)$ is an instantiation of (A,B,C)

$$H_{((A \vee (\neg B)) \wedge C)}(0,0,1) = (ABC - BC + C)(0,0,1) = 0 - 0 + 1 = 1$$

The following penalty function gives a penalty to every subexpression of the WFF that is not satisfied. It looks at the conjunctive terms in the upper level of the WFFs structure. For every term φ_i in $\varphi_1 \wedge \varphi_2 \wedge \ldots \varphi_l$, it computes the characteristic of the negation of φ_i.

DEFINITION 2.5 THE PENALTY FUNCTION

The penalty P_φ of a WFF φ is a function $P_\varphi : V^n \rightarrow \mathcal{N}$, such that:

- $P_{x_i}(x_1, \ldots, x_n) = 1 - H_{x_i}(x_1, \ldots, x_n)$
- $P_{(\neg\varphi_1)}(x_1, \ldots, x_n) = H_{\varphi_1}(x_1, \ldots, x_n)$
- $P_{(\varphi_1 \vee \varphi_2)}(x_1, \ldots, x_n) = H_{((\neg\varphi_1)\wedge(\neg\varphi_2))}(x_1, \ldots, x_n)$
- $P_{(\varphi_1 \wedge \varphi_2)}(x_1, \ldots, x_n) = P_{\varphi_1}(x_1, \ldots, x_n) + P_{\varphi_2}(x_1, \ldots, x_n)$
- $P_{(\varphi_1 \rightarrow \varphi_2)}(x_1, \ldots, x_n) = P_{((\neg\varphi_1)\vee\varphi_2)}(x_1, \ldots, x_n)$

A more intuitive way to look at the penalty function is to observe that if $\varphi = \bigwedge_{i=1}^{m} \varphi_i$ then

$$P_\varphi = \sum_{i=1}^{m}(1 - H_{\varphi_i}) = \sum_{i=1}^{m} H_{(\neg\varphi_i)}$$

Using this definition, a penalty of one is computed for any conjunctive term that is not satisfied. If all terms are satisfied, P_φ gets the value zero. Otherwise, the function computes the number of unsatisfied terms.

EXAMPLE 2.3

$$\begin{aligned} P_{((A\vee(\neg B))\wedge C)} &= P_{(A\vee(\neg B))} + P_C \\ &= H_{((\neg A)\wedge B)} + 1 - H_C \\ &= (1 - A)B + 1 - C \\ &= -AB + B - C + 1 \end{aligned}$$

$P_{((A\vee(\neg B))\wedge C)}(0, 1, 0) = 2$ since both C and $(A \vee (\neg B))$ are evaluated to false.

DEFINITION 2.6 ENERGY OF A WFF

The energy function E_φ equals to the penalty function P_φ but is expressed in a sum of products form (normal form).

The process of generating a penalty function from the original WFF using the previous recursive definition, generates expressions that are nested (like the original WFF). Conversion of this nested form into sum of products is done by simplifying the expression. (using distributive ,associative and commutative laws plus the boolean idempotent law: $X \cdot X = X$). We insist on the sum of products form since it has a direct translation into a connectionist network topology.

EXAMPLE 2.4

$$\begin{aligned} P_{(\neg((A\vee B)\wedge(A\vee C)))} &= H_{((A\vee B)\wedge(A\vee C))} \\ &= (A+B-AB)(A+C-AC) \\ &= AA+AC-AAC+BA+BC-ABC-AAB-ABC+ABAC \\ &= A+AC-AC+BA+BC-ABC-AB-ABC+ABC \\ &= A+BC-ABC = E_{(\neg((A\vee B)\wedge(A\vee C)))} \end{aligned}$$

LEMMA 2.1 φ IS SATISFIED BY S IFF E_φ IS MINIMIZED BY S AND THE GLOBAL MINIMA IS ZERO.

Proof: By induction on the level (k) of nesting of φ:

If $k = 0$

then $P_{x_i} = 1 - H_{x_i} = 1 - x_i = 0$ iff $x_i = 1$

iff $\varphi = x_i$ is satisfied by S.

Step:

$\neg\varphi$ is satisfied by S, iff $H_\varphi = 0$ iff $P_{\neg\varphi}$ is minimized to zero.

$\varphi_1 \vee \varphi_2$ is satisfied by S, iff $H_{\varphi_1\vee\varphi_2} = 1$ iff $H_{\neg(\varphi_1\vee\varphi_2)} = 0$ iff $H_{\neg\varphi_1\wedge\neg\varphi_2} = 0$ iff $P_{\varphi_1\vee\varphi_2}$ is minimized to zero by S.

$\varphi_1 \wedge \varphi_2$ is satisfied by S, iff $H_{\varphi_1\wedge\varphi_2} = 1$ iff $H_{\neg(\varphi_1\wedge\varphi_2)} = 0$ iff $H_{\neg\varphi_1\vee\neg\varphi_2} = 0$ iff $H_{\neg\varphi_1} + H_{\neg\varphi_2} = 0$ iff $P_{\varphi_1\wedge\varphi_2}$ is minimized to zero by S.

$\varphi_1 \rightarrow \varphi_2$ is satisfied iff $\neg\varphi_1 \vee \varphi_2$ is satisfied, iff $\neg\varphi_1 \vee \varphi_2$ is satisfied iff $P_{\neg\varphi_1\vee\varphi_2}$ is minimized to zero .

□

We saw that every WFF φ has a function E_φ that is minimized to zero on instantiation $\bar{x}$ iff $\bar{x}$ satisfies φ(when φ is a contradiction, $E_\varphi > 0$).

Both $(-H_\varphi)$ and E_φ have this property. However, we prefer E_φ since it has a heuristic knowledge about how "far" the current instantiation is from the global minima. E_φ counts the number of conjunctive sub-expressions that have not been satisfied yet. This property makes E_φ attractive when considering a hill climbing technique for the minimization (like the one connectionist models use). A direct use of $-H_\varphi$ for hill climbing energy minimization will increase the chances to fall into a local minimum because the characteristic function does not give any indication to the direction we have to step when some constraints are not met. E_φ on the other hand suggests the direction as to satisfy as many subexpressions as possible. (Note that the heuristic knowledge about the satisfiability of φ is maximized if φ is in conjunctive normal form).

2.2. High order energy functions and high order connectionist models

So far, the energy function defined can not be minimized by the "classical" connectionist model because connectionist models can minimize only quadratic energy functions. We will see now that all energy functions are equivalent to quadratic ones.

DEFINITION 2.7 K-ORDER ENERGY FUNCTION

A K-order energy function is a function $E : \{0,1\}^n \rightarrow \mathcal{R}$ that can be expressed in a sum of products form, and when expressed so, has terms with a product of up to k variables.
We will denote the sum of product form of a k-order energy function by:
$E^k(x_1, \ldots, x_n) =$

$$\sum_{1 \le i_1 < i_2 < \cdots < i_k \le n} w^k_{i_1,\ldots,i_k} x_{i_1} \cdots x_{i_k} + \sum_{1 \le i_1 < \cdots < i_{k-1} \le n} w^{k-1}_{i_1,\ldots,i_{k-1}} x_{i_1} \cdots x_{i_{k-1}} + \cdots + \sum_{1 \le i \le n} w_i x_i + w^0$$

To denote the coefficients (or weights in connectionist terminology) we will use a k-dimensional triangular matrix W of $(n+1)^k$ elements and with an index of $0, 1, 2, \ldots, n$ in each dimension, such that

$$w^j_{i_1,\ldots,i_j} = W[0, 0, \ldots, 0, i_1, \ldots, i_j]$$

where $0 \le i_1 < i_2 < \cdots < i_j \le n$ and $0 \le j \le k$.

Quadratic energy functions (or second order energy functions) are special cases of energy functions in the form :

$$\sum_{1\leq i<j\leq n} w_{ij}^2 x_i x_j + \sum_{i<n} w_i^1 x_i + w^0$$

We can extend the quadratic models ([1],[2] ,[3]) to minimize also high order energy functions by mapping these functions into hyper-graphs. Variables map into processing units and terms map into hyper arcs. A coefficient of a k-variable term becomes the weight (with opposite sign) of a hyper arc that connects these k variables. Each unit computes

$$net_i = \frac{dE}{dX_i} = \sum_{i_1<i_2<\cdots<i_{k-1}} w_{i,i_1,..,i_{k-1}} x_{i_1} x_{i_2} \cdots x_{i_{k-1}}$$

and adjust its activation value according to the specific model that we extend. When a symmetric k-dimensional matrix of weights is used energy decreases monotonically and eventually an equilibrium is reached when the network finds a local or a global minimum.

DEFINITION 2.8 VISIBLE VARIABLES AND HIDDEN VARIABLES

We can arbitrarily divide the variables of an energy function [2] into two sets:

1. Visible variables are usually of interest to an observer. Their final values may be the output of a system that is used to solve a problem stated as an energy minimization problem. An instantiation to these output variable is considered to be an answer to the problem.

2. Hidden variables are usually not of interest to an external observer. Hidden variables correspond to hidden units in connectionist models terminology.

We will denote sometimes a function with hidden variables as a function $E(\bar{x}, \bar{t})$ of two vectors, where $\bar{x}$ is the vector of visible variables and $\bar{t}$ is the vector of hidden variables.

An energy function may have n visible variables and j hidden variables. It is of interest to compare what can be computed by functions of n visible variables but with different number of hidden variables. In chapter 3 we will see that we can get a quadratic energy function from a high

[2] Later we'll see that the same distinction between hidden and visible variables can also be applied to the variables of a WFF.

order one by adding new hidden variables, and we can get a high order energy function from a lower order one by eliminating some or more of the hidden variables.

DEFINITION 2.9 THE MINIMIZING SET OF AN ENERGY FUNCTION

The *minimizing set* of an energy function E of n visible variables is the set of all instantiations $\bar{x}$ such that $E(\bar{x})$ is minimized. These instantiations are considered to be *"solutions"* to the minimization problem. The set of minimizing solutions projected on the visible variables is called "the set of visible solutions".

Formally: The set of visible solutions is:

$$\{\bar{x} \mid (\exists \bar{t}) E(\bar{x}, \bar{t}) = min_{\bar{y},\bar{t}}\{E(\bar{x}, \bar{t})\}\}$$

EXAMPLE 2.5 The set of visible solutions of $-X + a$ is $\{1\}$, for any real a.

EXAMPLE 2.6 The set of visible solutions of

$$E = XY - XT - YT + ZT + 2.9T$$

is:

$\{(000), (001), (010), (0110), (100), (101)\}$. When X, Y, Z are visible and T is hidden, since $E(X, Y, Z, T)$ is evaluated to be:

$E(0000) = E(0010) = E(0100) = E(0110) = E(1000) = E(1010) = 0;$

$E(1100) = E(1110) = 1; E(0001) = E(0111) = E(1011) = E(1111) = 2.9; E(0011) = 3.9;$

$E(1101) = E(0101) = E(1001) = 1.9;$

2.3. Energy functions describe WFFs

DEFINITION 2.10 DESCRIBING A WFF BY AN ENERGY FUNCTION

An energy function E *describes* a WFF φ if the set of models that satisfy φ is equal to the set of visible solutions of E.

Formally: E describes φ if $(\forall \bar{x})(\varphi(\bar{x}) = 1 \Longleftrightarrow ((\exists \bar{t})E(\bar{x}, \bar{t})) = MIN_{\bar{y}}\{E(\bar{y})\})$.

THEOREM 2.1 IF φ IS SATISFIABLE THEN E_φ DESCRIBES φ

Proof: From Definition 2.10 and Lemma 2.1

□

COROLLARY 2.1 φ IS A TAUTOLOGY IFF $E_\varphi = 0$.

Proof: If $E_\varphi = 0$, then for every instantiation $\bar{x}$, $E_\varphi(\bar{x}) = 0$ and 0 is the global minimum.

Since E_φ describes φ, $\bar{x}$ satisfies φ for all $\bar{x}$, and therefore φ is a tautology.

If φ is a tautology then for any $\bar{x}$, $\varphi(\bar{x}) = 1$. But, if $\varphi(\bar{x}) = 1$ then $E_\varphi(\bar{x}) = 0$. Therefore, for all $\bar{x}$ $E_\varphi(\bar{x}) = 0$.

We now prove by induction that all the coefficients of the sum of products are 0.

By taking $\bar{0}$ (the zero instantiation) all variables becomes zero and we can conclude that the constant (w^0) in the sum of product is zero.

By induction, assume that $w^j_{i_1,\ldots,i_j} = 0$ for $j < k$, we can select an instantiation $\bar{x}$ that instantiates to zero all variables except $x_{i_1}, \ldots, x_{i_k}$. Since $E(\bar{x}) = 0$ we can conclude that $w^k_{i_1,\ldots,x_{i_k}} = 0$

□

EXAMPLE 2.7

$$\begin{aligned} E_{((A \vee (\neg A))} &= H_{((\neg A) \wedge A)} \\ &= (1 - A)A = A - AA = 0 \end{aligned}$$

COROLLARY 2.2 A CONTRADICTION CAN NOT BE DESCRIBED BY ANY ENERGY FUNCTION

Proof: For every energy function there exists an instantiation that minimizes the function. If E describes φ then φ is satisfied by this instantiation. Therefore, φ is satisfiable.

□

COROLLARY 2.3 IF $\varphi(x_1, \ldots, x_n)$ IS A WFF OF N VARIABLES THEN φ IS DESCRIBED BY n-ORDER ENERGY FUNCTION WITH NO HIDDEN VARIABLES.

Proof: To show that $E\varphi$ is in the order of n: Assume that the sum of products form of E_φ has a term of $j > n$ variables that can not further be simplified , then at least one of the variables appears more then once. But since $X_i X_i = X_i$ the term can be simplified to have less then j variables. Contradiction.

□

EXAMPLE 2.8

$$\begin{aligned} E_{(A \vee \neg B \vee \neg C)} &= H_{(\neg A \wedge B \wedge C)} \\ &= (1 - A)BC = BC - ABC \end{aligned}$$

COROLLARY 2.4 IF φ IS A SATISFIABLE CONJUNCTION OF WFFS EACH OF MAXIMUM K VARIABLES, THEN φ IS DESCRIBED BY A k-ORDER ENERGY FUNCTION WITH NO HIDDEN VARIABLES.

Proof: From definition 2.5, the penalty of a conjunction of sub-formulas is equal to the sum of the penalties of each sub-formulas. Each sub-formula has maximum k variables, so the penalty of it is of order k

□

EXAMPLE 2.9

$$\begin{aligned} E_{A \wedge (B \vee (\neg C)))} &= E_A + E_{(B \vee (\neg C))} \\ &= (1 - A) + H_{((\neg B) \wedge C)} \\ &= (1 - A) + (1 - B)C = 1 - A + C - BC \end{aligned}$$

3. The equivalence between high and low order energy functions

Infinitely many energy functions seems to solve only a single minimization problem. For example there is only one minimizing set to all the functions of the form $E(x_1,\ldots,x_n)+\alpha$, for all Real α. We call energy functions that have the same set of visible solutions *equivalent*. We will show now that any high order energy function is equivalent to a low order one with additional hidden variables. An efficient algorithm is given for the conversion of high order energy into low order one. Also, another algorithm is given for transforming a low order energy function into (possibly) higher one by elimination of some or all of the hidden variables.

DEFINITION 3.1 EQUIVALENCE BETWEEN ENERGY FUNCTIONS

Two energy functions E_1 and E_2 with the same $\hat{X}=(x_1,\ldots,x_n)$ visible variables and arbitrary number of hidden variables $\hat{T}_1$ and $\hat{T}_2$ respectively are *equivalent* if the sets of visible solutions of E_1 and E_2 are equal. We denote the equivalence by $E_1 \approx E_2$.
Formally:

$$E_1 \approx E_2 \text{ iff } \{\bar{x} \mid (\exists \bar{t}_1)E_1(\bar{x},\bar{t}_1) = min_{\bar{y}\bar{t}}\{E_1(\bar{y},\bar{t})\}\} = \{\bar{x} \mid (\exists \bar{t}_2)E_2(\bar{x},\bar{t}_2) = min_{\bar{y}\bar{t}}\{E_2(\bar{y},\bar{t})\}\}$$

EXAMPLE 3.1 $aXY+b \approx aXY+c$ for any a,b or c.

EXAMPLE 3.2 $E_1 = 5XY-3YZ-XYZ \approx 5XY-3YZ-2XT-2YT-2ZT+5T = E_2$.
The following table shows the values of E_1 and E_2 for all possible instantiations of the variables X,Y,Z,T:

XYZ	E_1	$XYZT$	E_2
000	0	0000	0
		0001	5
001	0	0010	0
		0011	3
010	0	0100	0
		0101	3
011	-3	0110	-3
		0111	-2
100	0	1000	0
		1001	3
101	0	1010	0
		1011	1
110	5	1100	5
		1101	6
111	1	1110	2
		1111	1

The set of minimal solutions of E_1 is $\{(011)\}$.

The set of minimal solutions of E_2 is $\{(0110)\}$.

The set of visible solutions of E_2 is $\{(011)\}$ and is equal to the set of visible solutions of E_1.

It is easy to see that the relation is reflexive, symmetric and transitive, therefore it is an equivalence relation.

3.1. Converting high order energy function into a lower order function

High order energy functions can be converted into equivalent lower energy functions using the following constructive theorem:

THEOREM 3.1 EVERY k ORDER ENERGY FUNCTION E CAN BE TRANSFORMED INTO AN EQUIVALENT $(k-1)$ ORDER ENERGY FUNCTION BY ADDING EXTRA HIDDEN VARIABLES. TRANSFORMATION IS DONE BY REPLACING EACH OF THE K-ORDER TERMS IN E BY A (K-1) ORDER EXPRESSION, WHICH IS DETERMINED BY THE FOLLOWING LEMMA 3.1 AND LEMMA 3.2.

Proof: Using the following lemmas a constructive proof can easily be shown.

□

LEMMA 3.1 ANY K-ORDER TERM $(\alpha \prod_{i=1}^{k} x_i)$, WITH A NEGATIVE COEFFICIENT α, CAN BE REPLACED BY A SUM OF QUADRATIC TERMS OF THE FORM : $\sum_{i=1}^{k} 2\alpha X_i T - (2k-1)\alpha T$ GENERATING AN EQUIVALENT ENERGY FUNCTION WITH ONE ADDITIONAL HIDDEN VARIABLE.

Proof: In appendix A.1

□

EXAMPLE 3.3

$$XY - 3XYZU \approx XY - 6XT - 6YT - 6ZT - 6UT + 21T$$

LEMMA 3.2 ANY K-ORDER TERM $(\alpha \prod_{i=1}^{k} x_i)$, WITH A POSITIVE COEFFICIENT α, CAN BE REPLACED BY A SUM OF TERMS (OF ORDER $(k-1)$) OF THE FORM : $\alpha \prod_{i=1}^{k-1} x_i - \sum_{i=1}^{k-1} 2\alpha X_i T + 2\alpha X_k T + (2k-3)\alpha T$, GENERATING AN EQUIVALENT ENERGY FUNCTION WITH ONE ADDITIONAL HIDDEN VARIABLE.

Proof: In appendix A.1

□

EXAMPLE 3.4

$$\begin{aligned} -XY + XYZU &\approx -XY + XYZ - 2XT - 2YT - 2ZT + 2UT + 5T \\ &\approx -XY + XY - 2XT' - 2YT' + 2ZT' + 3T' - 2XT - 2YT - 2ZT + 2UT + 5T \\ &= -2XT' - 2YT' + 2ZT' + 3T' - 2XT - 2YT - 2ZT + 2UT + 5T \end{aligned}$$

3.2. Eliminating hidden variables

The symmetric transformation, from low order into high order energy by eliminating hidden variables, is also possible

THEOREM 3.2 EVERY k- ORDER ENERGY FUNCTION WITH AT LEAST ONE HIDDEN VARIABLE T, CAN BE TRANSFORMED INTO AN EQUIVALENT HIGHER ORDER ENERGY FUNCTION THAT DOES NOT INCLUDE T, USING THE FOLLOWING METHOD.

Method:Assume T is a hidden variable to be eliminated. We replace: $(\sum_{j=1}^{l} \alpha_j X_{i_j})T$ with a new sum of terms that is generated using the following procedure:

Consider all instantiations $\bar{x} = (x_{i_1}, \ldots, x_{i_l})$, of the variables $X_{i_1}, \ldots X_{i_l} = \hat{X}$ such that

$$\beta_S = \sum_{j=1}^{l} \alpha_j x_{i_j} < 0$$

where S is an assignment for just the l variables.

For each such instantiation $(x_{i_1}, \ldots, x_{i_l})$ obtained by assignment S, let the function L_S^j be:

$$L_S^j(\hat{X}) = \begin{cases} X_{i_j} & \text{if } S(X_{i_j}) = 1 \\ 1 - X_{i_j} & \text{if } S(X_{i_j}) = 0 \end{cases}$$

Then, generate the term:

$$newterm = \sum_{\text{S such that } \beta_S < 0} \beta_S \prod_{j=1}^{l} L_S^j(\hat{X})$$

Note that there are maximum 2^l different assignments S for those l variables.

Replace the old term: $(\sum_{j=1}^{l} \alpha_j X_{i_j})T$ with "newterm" which does not include T.

Proof: See Appendix A.3

□

EXAMPLE 3.5 Let T be the hidden variable to be eliminated, then:

$$AB + TAC - TA + 2TB - T = AB + T(AC - A + 2B - 1)$$

The following assignments for (A, B, C) cause β to be less then zero:

$\beta_{(0,0,0)} = -1$

$\beta_{(0,0,1)} = -1$

$\beta_{(1,0,0)} = -2$

$\beta_{(1,0,1)} = -1$

The new term equals:

$$-(1-A)(1-B)(1-C)-(1-A)(1-B)C-2A(1-B)(1-C)-A(1-B)C = -ABC+AB+AC-A+B-1$$

Therefore:

$$AB + TAC - TA + 2TB - T \approx -ABC + 2AB + AC - A + B$$

Note that $\{(1,0,0)\}$ is the set of visible solutions for both functions.

3.3. The tradeoff between the number of hidden variables, the number of connections and the order of the energy function

With respect to energy minimization connectionist models, we will analyze the complexity of the energy function by counting the number of hidden variables (hidden units) and the fan-out of the variables. The fan-out is counted by the number of different other variables that a variable shares a term with, and it is equivalent to the number of connections to other units in connectionist models. The energy function can be viewed as a hyper graph such that its nodes are variables, and its hyper arcs are terms connecting several variables. The weight of such arc is the coefficient of the term. The fan-out is the number of different nodes connected to the variable.

We saw that one k-order term can be converted into $O(k)$ terms of lower order with a single additional hidden variable that has a fan-out of k. In the worst case we need $\binom{n}{k}$ new variables to transform a k-order function into order of $k-1$. To convert a k-order function into quadratic one we need: $\sum_{i=3}^{k}\binom{n}{i}$ new variables; therefore the worst case to convert n-order function into a quadratic one uses $O(2^n)$ new variables. There is an obvious tradeoff between the order of the function and the number of hidden variables. We can reduce the order by adding more variables, and we can eliminate hidden variables by adding to the order of the function. There is no need for hidden variables at all if we allow the order of the function to be n.

A tradeoff also exists between the order of the function and the fan-out of the variables. Eliminating a variable with fan-out of k may results in the creation of k-order terms, while reducing the order of a k-order term results in adding new variables with fan out of k. Later we will see that in order to implement any boolean mapping or any propositional satisfiability problem, we can find

an n-order function with a fan-out of $O(n)$, or a quadratic one with hidden variables with fan-out that is bounded by a constant. Note that the order of the energy function (k) and the fan-out (l) determine the maximum number of terms (weights) which is another complexity measure. The maximum number of terms shared by a variable is therefore $O(\sum_{i=1}^{k-1}\binom{l}{i})$.

We saw in section 2 that every WFF φ is described by some energy function E_φ. In addition we know that every boolean function h characterizes some WFF φ (the boolean implementation of h with AND , OR and NOT gates, for example). We can therefore implement any boolean function using an energy minimization model by implementing $E_{out \leftrightarrow \varphi}$ where "out" is the unit where we expect to find the result of the function. High order models of the type described in section 2.2 can be used to implement any WFF or any boolean function with no additional hidden variables and with maximum $n(n-1)/2$ connections. In this sense high order models can be used as universal networks that can implement any function just by changing weights (although an exponential number of weights may be needed to implement some functions). Quadratic models using the algorithms described here will need exponential number of units in order to be able to implement any function.

4. The mapping between satisfiability and energy minimization

In previous sections we have shown that for every WFF there exist an energy function (E_φ) that describes it. Further, we saw that it is possible to convert any high order energy function into a quadratic one. However, too many hidden variables are needed ($O(2^n)$) when applying the simple algorithm described in section 3.1. In this section we show that it is possible to convert a satisfiable WFF φ into a quadratic energy function that describes φ by adding only $O(length(\varphi))$ hidden variables. We also show that for any energy function there exists a WFF that is described by the function. The constructions we build are used to show a one to one mapping between classes of equivalent energy functions and classes of equivalent satisfiable WFFs.

4.1. An economical way to convert a WFF into quadratic energy function

First, we convert the WFF into an equivalent WFF that is composed out of conjunction of triples. A triple is a proposition expression that involves up to three variables. Each triple can be described by a 3^{rd} order (cubic) energy function; therefore, the penalty function of the conjunction of triples is a cubic energy function. We then transform the cubic energy function into a quadratic one. This transformation produces hidden variables in the order of the length of the original WFF.

4.1.1. Equivalence between WFFs .

DEFINITION 4.1 EQUIVALENCE BETWEEN WFFS.

We call the set of all instantiations that satisfy the WFF, projected on the visible variables: the *set of visible satisfying models of the WFF.*

φ_1 is equivalent to φ_2 if the set of visible satisfying models of φ_1 is equal to the set of visible satisfying models of φ_2.

Formally: $\varphi_1 \approx \varphi_2$ iff $(\forall \bar{x})((\exists \bar{t})\varphi_1(\bar{x},\bar{t}) = 1 \Longleftrightarrow (\exists \bar{t}')\varphi_2(\bar{x},\bar{t}') = 1)$.

It means that any instantiation $\bar{x}$ of the visible variables that can cause φ_1 (by some assignment to the hidden variables of φ_1) to be true, can also cause φ_2 to be true (by some assignment to the

hidden variables of φ_2), and vice versa. It is easy to see that $\approx$ is reflexive transitive and symmetric therefore it is an equivalence relation.

Next we will see that every WFF φ is equivalent to a WFF ψ in a conjunction of triples form (with additional O(length(φ)) hidden variables).

4.1.2. Converting A WFF into an equivalent Conjunction of Triples Form WFF. A WFF φ is in Conjunction of Triples Form (CTF) if $\varphi = \bigwedge_{i=1}^{m} \varphi_i$ where φ_i is a sub-formula of maximum three variables.

Every WFF can be translated into an equivalent WFF in CTF in the following intuitive way: For every variable or logical connective (Eg: $\wedge, \vee, \neg$) we generate a new hidden variable. We "name" the variable or the connective using the logical "if and only if" connective ($\leftrightarrow$). We do this operation bottom up on the parse tree of the original WFF except from the top most connective. Each time we generate such new sub-formula, we use in it the previously allocated hidden variables. Thus, each sub-formula has maximum three variables, and the conjunction of these sub-formulas is in CTF.

For example: To convert $(A \vee (\neg B)) \rightarrow (C \vee D)$ we name each of the variables by allocating new variables T_1, T_2, T_3, T_4 and generate $((A \leftrightarrow T_1), ((\neg B) \leftrightarrow T_2), (C \leftrightarrow T_3), (D \leftrightarrow T_4))$.
We name each of binary operations bottom up. (except the top most binary operation:
$((T_1 \vee T_2) \leftrightarrow T_5)$ for $(A \vee (\neg B))$
$((T_3 \vee T_4) \leftrightarrow T_6$ for $(C \vee D)$
and finally the top most connective (which we do not "name") is:
$(T_5 \rightarrow T_6)$
The conjunction of these expressions is in CTF and it is equivalent to the original WFF. For a similar transformation of an expression in conjunctive normal form see 3-sat problem in [10].

To formally prove this algorithm we use the following attribute grammar:
Syntax directed parsing is done to any input WFF, and an equivalent WFF in CTF is generated in the "t" attribute of the nonterminal S_0 . We add a new logic connective $(\varphi_1 \leftrightarrow \varphi_2)$ which is equivalent to $((\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1))$. The WFFs that this parser accept, do not contain explicitly the connective: $\rightarrow$. We assume that any subexpressions in the original WFF of the form $(\varphi_1 \rightarrow \varphi_2)$

were converted into: $((\neg\varphi_1) \vee \varphi_2)$ prior to parsing. The attribute S.val represents the contribution of the current production, while the attribute S.t represents the accumulated conjunction of triples, not including the current contribution.

$$
\begin{array}{lll}
S0 \longrightarrow S & & S_0.t := (s.t \wedge s.val) \\
S \longrightarrow L & & S.val := L.val \\
 & & S.t := \phi \\
 \longrightarrow (S \vee S) & & T_1 := allocatenew() \\
 & & T_2 := allocatenew() \\
 & & S.val := (T_1 \vee T_2) \\
 & & S.t := (((S^1.t \wedge S^2.t) \wedge (S^1.val \leftrightarrow T_1)) \wedge (S^2.val \leftrightarrow T_2)) \\
 \longrightarrow (S \wedge S) & & T_1 := allocatenew() \\
 & & T_2 := allocatenew() \\
 & & S.val := (T_1 \wedge T_2) \\
 & & S.t := (((S^1.t \wedge S^2.t) \wedge (S^1.val \leftrightarrow T_1)) \wedge (S^2.val \leftrightarrow T_2)) \\
 \longrightarrow (\neg S) & & T := allocatenew() \\
 & & S.val := (\neg T) \\
 & & S.t := (S.t \wedge (S.val \leftrightarrow T)) \\
L \longrightarrow X_i & & L.val := X_i \\
 \longrightarrow (\neg X_i) & & L.val := (\neg X_i)
\end{array}
$$

THEOREM 4.1 THE GRAMMAR [3] GENERATES A WFF ψ FROM φ SUCH THAT φ IS EQUIVALENT TO ψ AND ψ IS IN CTF AND CONTAINS $O(length(\varphi))$ NEW HIDDEN VARIABLES.

Proof: See appendix A.4.

□

EXAMPLE 4.1 Converting $((A \wedge B) \vee (\neg C))$ into conjunction of triples generates:

$$(((((A \leftrightarrow T_1) \wedge (B \leftrightarrow T_2)) \wedge ((T_1 \wedge T_2) \leftrightarrow T_3)) \wedge ((\neg C) \leftrightarrow T_4)) \wedge (T_3 \vee T_4))$$

4.1.3. An algorithm to transform a WFF into a second order energy function .

- Convert into conjunction of triples. (Base on theorem 4.1)
- Convert conjunction of triples into a cubic energy function that describes it. (Using Definition 2.5 , simplify it to a sum of products form and use corollary 2.4)

[3] Other variations of the grammar may generate less hidden variables, but still $O(length(\varphi))$ (for example: hidden variables may be generated just for binary connectives).

- Convert third order terms into second order terms. (Using Theorem 3.1)

EXAMPLE 4.2 Converting $((A \wedge B) \vee (\neg C))$ into conjunction of triples generates:

$$(((((T_1 \leftrightarrow A) \wedge (T_2 \leftrightarrow B)) \wedge (T_3 \leftrightarrow (T_1 \wedge T_2))) \wedge (T_4 \leftrightarrow (\neg C))) \wedge (T_3 \vee T_4))$$

Eliminating $\leftrightarrow$:

$$((\neg T_1) \vee A) \wedge (T_1 \vee (\neg A)) \wedge ((\neg T_2) \vee B) \wedge (T_2 \vee (\neg B)) \wedge ((\neg T_3 \vee T_1) \wedge ((\neg T_3) \vee T_2) \wedge ((T_3 \vee (\neg T_1) \vee (\neg T_2)) \wedge ((\neg T_4) \vee (\neg C)) \wedge (T_4 \vee C) \wedge (T_3 \vee T_4)$$

Generating the 3^{rd} order energy function:

$$\begin{gathered} T_1(1-A) + (1-T_1)A + T_2(1-B) + (1-T_2)B + T_3(1-T_1) + T_3(1-T_2) \\ +(1-T_3)T_1T_2 + CT_4 + (1-T_4)(1-C) + (1-T_3)(1-T_4) \\ = \\ T_1 - 2AT_1 + A + T_2 - 2BT_2 + B + T_3 - T_1T_3 - T_2T_3 + T_1T_2 - T_1T_2T_3 + 2CT_4 + 2 - C - 2T_4 + T_3T_4 \end{gathered}$$

Converting into quadratic function:

$$\begin{gathered} T_1 - 2AT_1 + A + T_2 - 2BT_2 + B + T_3 - T_1T_3 - T_2T_3 + T_1T_2 \\ -2T_1T_5 - 2T_2T_5 - 2T_3T_5 + 5T_5 + 2CT_4 + 2 - C - 2T_4 + T_3T_4 \end{gathered}$$

4.1.4. Complexity analysis. The algorithm described above transforms a WFF into a quadratic energy function in time linear in the length of the WFF:
The conversion into conjunction of triples and the conversion into cubic energy function are operations that parse the nested structure of the WFF in linear time.
Simplifying a 3-variable subexpression takes a constant time, and conversion of all the cubic terms into quadratic terms is linear in their number (the number of terms is in the order of the number of binary connectives).

The number of hidden units that are generated is linear in the length of the original WFF:
Conversion into triples generates new variables as the number of connectives in the WFF.
Conversion into quadratic function generates hidden variables in the order of the number of binary connectives. (Only triples of three variables generate a cubic term).

The fan-out of all these hidden variables is maximum six for those generated by the attribute grammar and three for those generated by the conversion into quadratic function. The visible variables may have a fan-out of $O(n)$.

COROLLARY 4.1 ANY WFF φ CAN BE CONVERTED INTO A QUADRATIC ENERGY FUNCTION IN TIME $O(length(\varphi))$, AND BY ADDING $O(length(\varphi))$ HIDDEN VARIABLES WITH FAN-OUT BOUNDED BY CONSTANT.

COROLLARY 4.2 ANY BOOLEAN FUNCTION h CAN BE IMPLEMENTED IN A QUADRATIC ENERGY MINIMIZATION NETWORK OF SIZE THAT IS PROPORTIONAL TO THE LENGTH OF THE BOOLEAN EXPRESSION φ THAT IS CHARACTERIZED BY h.

Proof: Converting the WFF: $(out \leftrightarrow \varphi)$.

□

4.2. Every energy function describes some satisfiable WFF.

To complete the proof that the satisfiability problem is equivalent to the energy minimization problem, we need to show that for any energy function E with n visible variables and j hidden variables there exists a satisfiable WFF φ, such that E describes φ. We have proved in section 3.2 that any energy function φ with hidden variables is equivalent to another energy function (that may be of higher order) with no hidden variables at all. All we need to complete the proof is the following theorem:

THEOREM 4.2 FOR ANY K-ORDER ENERGY FUNCTION E WITH NO HIDDEN VARIABLES THERE EXIST A SATISFIABLE WFF φ SUCH THAT E DESCRIBES φ. A METHOD FOR CONSTRUCTING φ IS GIVEN BELLOW:

Method: Given E and n variables $\hat{X} = (x_1, \ldots, x_n)$, there are $\mid \{0,1\}^n \mid = 2^n$ possible instantiations. Compute $E(\bar{x})$ for all instantiations $S(\hat{X}) = \bar{x} \epsilon \{0,1\}^n$ and find $min_{\bar{x}}\{E(\bar{x})\} = min_E$.

Let H_E be a boolean function which will be the characteristic function of φ:

$$H_E(\hat{X}) = \begin{cases} 1 & \text{if } E(\hat{X}) = min_E \\ 0 & \text{otherwise} \end{cases}$$

Build a WFF:

$$\varphi = \bigvee_{H_E(S(\hat{X}))=1} (\bigwedge_{i=1}^{n} L_S^i)$$

Where

$$L_S^i = \begin{cases} X_i & \text{if } S(X_i) = 1 \\ (\neg X_i) & \text{if } S(X_i) = 0 \end{cases}$$

φ is a disjunction of terms. Each term is of the form: $t_S = (\bigwedge_{i=1}^{n} L_S^i)$

Proof: We want to show that E is minimized by instantiation $\bar{x}$ iff φ is satisfied by $\bar{x}$.

But by construction of H_E, E is minimized by $\bar{x}$ iff $E(\bar{x}) = min_E$ iff $H_E(\bar{x}) = 1$.

By lemma 4.1, $H(\bar{x}) = 1$ iff $\varphi(\bar{x}) = 1$. Therefore, E is minimized by instantiation $\bar{x}$ iff $\varphi(\bar{x}) = 1$.

□

LEMMA 4.1 H_E IS THE CHARACTERISTIC FUNCTION OF φ

Proof: See appendix A.5.

□

THEOREM 4.3 EVERY ENERGY FUNCTION DESCRIBES SOME SATISFIABLE WFF

Proof: Using theorem 3.2 and theorem 4.2.

□

EXAMPLE 4.3

$$E(X,Y) = -XY + 1.5X$$

Trying all instantiations:

$$E(0,0) = 0$$
$$E(0,1) = 0$$
$$E(1,0) = 1.5$$
$$E(1,1) = 0.5$$

The characteristic function is:

$$H(0,0) = 1$$
$$H(0,1) = 1$$
$$H(1,0) = 0$$
$$H(1,1) = 0$$

The WFF that is described by E is therefore:

$$(((\neg X) \wedge (\neg Y)) \vee ((\neg X) \wedge Y))$$

4.3. Equivalence classes

The relation $\approx$ is an equivalence relation for both WFFs and energy functions of n visible variables. There is a one to one mapping (bijection) between the classes of satisfiable WFFs and the classes of energy functions. One class of WFFs, the one with empty set of visible satisfying models ("contradictions") does not map to any of the energy functions classes. There is also a bijection between the set of non-zero boolean functions and the classes of energy function (and satisfiable WFFs). Any non-zero boolean function H characterizes some satisfiable WFF φ which determines an energy function E_φ: $H_\varphi \overset{1-1}{\longleftrightarrow} [\varphi]_\approx \overset{1-1}{\longleftrightarrow} [E_\varphi]_\approx$. The cardinality of the set of classes of energy functions (of n visible variables) is therefore $2^{2^n}-1$. The class of tautologies for example map to the class of constant energy functions and to the boolean function 1 ($[E(x_1,\ldots,x_n)=0]_\approx \leftrightarrow [True]_\approx \leftrightarrow f(x_1,\ldots,x_n)=1$).

5. Summary and conclusions

We have shown an equivalence between the problem of satisfiability of propositional calculus and the problem of minimizing energy functions.

Any propositional WFF can be described by an n order energy function with no hidden variables, or by a quadratic energy function with additional hidden variables. Any quadratic (or higher order) energy function with some hidden variables is equivalent to a higher order energy function with no hidden variables, and any energy function describes some propositional WFF. The algorithm to convert a WFF into quadratic energy function efficiently generates linearly bounded number of hidden variables with constant bounded fan-out.

We have extended current quadratic energy minimization models to support high order functions and have identified procedures to convert high order energy functions into quadratic energy functions and vice versa. (By adding or eliminating hidden variables). As a result we can implement any WFF or any boolean function in energy minimization connectionist networks of any order (quadratic or higher), and we can build a universal n-order network with no hidden units that can implement any WFF (or boolean function).

As a consequence of the equivalence relations defined on both energy functions and WFFs we can show that there is a one-to-one mapping between the set of non zero boolean functions and the set of equivalence classes of energy functions. A one to one mapping also exists between classes of satisfiable WFFs and classes of energy functions. There are only $2^{(2^n)} - 1$ different classes of energy functions (of n visible variables) independently of the order and the number of hidden variables.

A connectionist energy minimization network can be built directly from the energy function, therefore connectionist networks can be used as inference engine for propositional calculus. (Some extensions are needed. For example: The use of three value logic to deduce "unknown" when a variable is instantiated to both "true" and "false" in two satisfying models). A system like this can deduce which truth assignment the atomic propositions *must* have for the WFF to be consistent. Contradiction may be sensed by energy level greater then zero,[4] and novel WFFs may be checked

[4] The problem of local minima causes the connectionist system to be uncertain whether a greater than zero energy is caused by a contradiction or just by a local minimum.

to see if they follow a certain set of WFFs, contradict it or are consistent with it. This form of knowledge representation is capable of being incrementally updated. When we add a new fact or rule to our knowledge base we do not have to re-compute all of the weights. We can find the energy function that describes the new fact and then take advantage of the fact that the penalty of a conjunction is the sum of the penalties. All we have to do is add the new function to the old one. This way only the weights that are affected by the new fact will be updated. (Deleting a fact is done by subtracting the function). We should note here that in traditional theorem proving (resolution for example) symbolic algorithms are used to deduce $S \vdash \varphi$ using sound syntactic rules, while the connectionist system described here deduces $S \models \varphi$ in the model space.

We may also conclude an important limitation to the kind of problems energy minimization connectionist networks *can* solve. The expressive power of systems based on energy minimization is identical to that of propositional calculus. Only those problems that can be stated as satisfiability problems of propositional calculus can be stated as energy minimization problems. We can conclude therefore that certain semi-decidable problems, like the satisfiability of predicate calculus or even decidable problems like Quantified Boolean Formulas [10], (unless P-space = NP) , *cannot* be precisely and efficiently represented in energy minimization networks.

6. Acknowledgments

I thank William Ball, Dan Kimura, Arun Kumar, Stan Kwasny, Andy Laine and Ron Loui for helpful discussions and suggestions.

A. Proofs

A.1. Proof of lemma: converting high order term with negative coefficient to low order terms.

To show that when $\alpha > 0$ then:

$$E_1(\hat{X}) = f(\hat{X}) - \alpha x_{i_1}, \ldots, x_{i_k} \approx f(\hat{X}) - 2\alpha \sum_{j=1}^{k} T x_{i_j} + (2k-1)\alpha T = E_2(\hat{X}, T)$$

Proof:

Let $\bar{x}$ be an instantiation by assignment S.

We denote: $S(x_{i_1}) = S(x_{i_2}) = \cdots = S(x_{i_k}) = 1$ by $\bar{x} \Rightarrow x_{i_1}, \ldots, x_{i_k}$.

If not all the variables $x_{i_1}, \ldots, x_{i_k}$ are instantiated to be "1" by S then there exists a j such that $S(x_{i_j}) = 0$. We denote this fact by $\bar{x} \not\Rightarrow x_{i_1}, \ldots, x_{i_k}$.

We prove the lemma by assuming that $\bar{x}$ minimizes one of the functions and showing that $\bar{x}$ minimizes the other function. We first prove i) that if $\bar{x}$ minimizes E_1 there exists an instantiation for T such that $E(\bar{x}, T)$ is minimized. Then we prove ii) that if $\bar{x}, T$ minimize E_2, then $\bar{x}$ also minimizes E_1.

In each prove we examine two basic cases: In case 1, we assume that $\bar{x}$ is instantiated to all ones, and in case 2 we assume that $\bar{x}$ is not all ones.

In each such case we prove that $\bar{x}$ minimizes the other function by showing that for every $\bar{y}$ the function gets a value that is greater or equal to the value it gets on $\bar{x}$. Thus we examine two sub-cases for each basic case:

In sub-case 1, we assume $\bar{y}$ is all ones, and in sub-case 2, we assume that $\bar{y}$ is not all ones.

i) Assume $E_1(\bar{x}) = min_{E_1}$ we want to show that $(\exists t')$ such that $E_2(\bar{x}, t') = min_{E_2}$.

case 1: Assume $\bar{x} \Rightarrow x_{i_1}, \ldots, x_{i_k}$, we want to show that $E_2(\bar{x}, 1) \leq E_2(\bar{y}, \bar{t})$ for all instantiations $\bar{y}$ and $\bar{t}$.

sub-case 1.1: Assume $\bar{y} \Rightarrow x_{i_1}, \ldots, x_{i_k}$ then:

$$E_2(\bar{x}, 1) = f(\bar{x}) - \alpha = E_1(\bar{x}) \leq E_1(\bar{y}) = f(\bar{y}) - \alpha = E_2(\bar{y}, 1) \leq f(\bar{y}) = E_2(\bar{y}, 0)$$

sub-case 1.2: Assume $\bar{y} \not\Rightarrow x_{i_1}, \ldots, x_{i_k}$ then:

$$E_2(\bar{x},1) = f(\bar{x}) - \alpha = E_1(\bar{x}) \leq E_1(\bar{y}) = f(\bar{y}) = E_2(\bar{y},0) \leq E_2(\bar{y},1)$$

(Since $E_2(\bar{y},1) = f(\bar{y}) - 2l\alpha + (2k-1)\alpha = f(\bar{y}) + r\alpha$) where $r > 0$ and $l < k$).

Therefore $E_2(\bar{x},1) \leq E_2(\bar{y},\bar{t})$ for any $\bar{y}$ and $\bar{t}$.

case 2: Assume $\bar{x} \not\Rightarrow x_{i_1}, \ldots, x_{i_k}$, we want to show that $E_2(\bar{x},0) \leq E_2(\bar{y},\bar{t})$ for all instantiations $\bar{y}$ and $\bar{t}$.

sub-case 2.1: Assume $\bar{y} \Rightarrow x_{i_1}, \ldots, x_{i_k}$ then:

$$E_2(\bar{x},0) = f(\bar{x}) = E_1(\bar{x}) \leq E_1(\bar{y}) = f(\bar{y}) - \alpha = E_2(\bar{y},1) \leq f(\bar{y}) = E_2(\bar{y},0)$$

sub-case 2.2: Assume $\bar{y} \not\Rightarrow x_{i_1}, \ldots, x_{i_k}$ then:

$$E_2(\bar{x},0) = f(\bar{x}) = E_1(\bar{x}) \leq E_1(\bar{y}) = f(\bar{y}) = E_2(\bar{y},0) \leq E_2(\bar{y},1)$$

(Since $E_2(\bar{y},1) = f(\bar{y}) - 2l\alpha + (2k-1)\alpha = f(\bar{y}) + r\alpha$) where $r > 0$).

Therefore $E_2(\bar{x},0) \leq E_2(\bar{y},\bar{t})$ for any $\bar{y}$ and $\bar{t}$.

Therefore there exists $\bar{t}'$ such that $E_2(\bar{x},\bar{t}') = min_{E_2}$

ii) Assume $E_2(\bar{x},\bar{t}) = min_{E_2}$ we want to show that $E_1(\bar{x}) = min_{E_1}$.

case 1: Assume $\bar{x} \Rightarrow x_{i_1}, \ldots, x_{i_k}$, we want to show that $E_1(\bar{x}) \leq E_2(\bar{y})$ for all instantiations $\bar{y}$.

$E_2(\bar{x},1) < E_2(\bar{x},0)$ therefore E_1 can not be a minimum

and therefore $E_2(\bar{x},1) = min_{E_2}$.

Subcase 1.1: Assume $\bar{y} \Rightarrow x_{i_1}, \ldots, x_{i_k}$

$$E_1(\bar{x}) = f(\bar{x}) - \alpha = E_2(\bar{x},1) \leq E_2(\bar{y},1) = f(\bar{y}) - \alpha = E_1(\bar{y})$$

Subcase 1.2: Assume $\bar{y} \not\Rightarrow x_{i_1}, \ldots, x_{i_k}$ then:

$$E_1(\bar{x}) = f(\bar{x}) - \alpha = E_2(\bar{x},1) \leq E_2(\bar{y},0) = f(\bar{y}) = E_1(\bar{y})$$

Therefore $E_1(\bar{x}) \leq E_1(\bar{y})$ for any $\bar{y}$.

case 2: Assume $\bar{x} \not\Rightarrow x_{i_1}, \ldots, x_{i_k}$, we want to show that $E_1(\bar{x}) \leq E_2(\bar{y})$ for all instantiations $\bar{y}$.

$E_2(\bar{x},0) = f(\bar{x}) < f(\bar{x}) - 2j\alpha + (2k-1)\alpha = E_2(\bar{x},1)$ therefore $E_2(\bar{x},0) = min_{E_2}$.

Subcase 2.1: Assume $\bar{y} \Rightarrow x_{i_1}, \ldots, x_{i_k}$

$$E_1(\bar{x}) = f(\bar{x}) = E_2(\bar{x},0) \leq E_2(\bar{y},1) = f(\bar{y}) - \alpha = E_1(\bar{y})$$

Subcase 2.2: Assume $\bar{y} \not\Rightarrow x_{i_1}, \ldots, x_{i_k}$ then:

$$E_1(\bar{x}) = f(\bar{x}) = E_2(\bar{x}, 0) \leq E_2(\bar{y}, 0) = f(\bar{y}) = E_1(\bar{y})$$

Therefore $E_1(\bar{x}) \leq E_1(\bar{y})$ for any $\bar{y}$.

Therefore $E_1(\bar{x}) = min_{E_1}$.

Therefore $E_1(\bar{x}) = min_{E_1}$ iff $(\exists \bar{t}')E_2(\bar{x}, \bar{t}') = min_{E_2}$.

Therefore $E_1 \approx E_2$.

□

A.2. Proof of lemma: converting high order term with positive coefficient to low order terms.

To show that when $\alpha > 0$ then:

$$E_1(\hat{X}) = f(\hat{X}) + \alpha x_{i_1}, \ldots, x_{i_k} \approx f(\hat{X}) + \alpha x_{i_1} \cdots x_{i_{k-1}} - 2\alpha \sum_{j=1}^{k-1} T X_{i_j} + 2\alpha x_k T + (2k-3)\alpha T = E_2(\hat{X}, T)$$

Proof:

Using a proof technique that is similar to the previous proof we show two directions:

i) If $\bar{x}$ minimizes E_1 then it also minimizes E_2.

ii) If $\bar{x}$ minimizes E_2 then it also minimizes E_1.

In each direction we consider three cases:

1) $\bar{x}$ is instantiated to all ones.

2) $x_{i_1} \ldots x_{i_{k-1}}$ is instantiated to all ones but x_{i_k} is instantiated to zero.

3) $x_{i_1} \ldots x_{i_{k-1}}$ is not instantiated to all ones.

In each case we consider three sub-cases:

1) $\bar{y}$ is instantiated to all ones.

2) $y_{i_1} \ldots y_{i_{k-1}}$ is instantiated to all ones but y_{i_k} is instantiated to zero.

3) $y_{i_1} \ldots y_{i_{k-1}}$ is not instantiated to all ones.

Let $\bar{x}$ be an instantiation by assignment S.

i) Assume $E_1(\bar{x}) = min_{E_1}$ we want to show that $(\exists t')$ such that $E_2(\bar{x}, t') = min_{E_2}$.

case 1: Assume $\bar{x} \Rightarrow x_{i_1}, \ldots, x_{i_k}$, we want to show that $E_2(\bar{x}, 0) \leq E_2(\bar{y}, \bar{t})$ for all

instantiations $\bar{y}$ and $\bar{t}$ $E_1(\bar{x}) = f(\bar{x}) + \alpha = E_2(\bar{x}, 0)$.

sub-case 1.1: Assume $\bar{y} \Rightarrow x_{i_1}, \ldots, x_{i_k}$ then:

$$E_2(\bar{x},0) = E_1(\bar{x}) \leq E_1(\bar{y}) = f(\bar{y}) + \alpha = E_2(\bar{y},0) \leq E_2(\bar{y},1)$$

(since $f(\bar{y}) + \alpha - 2(k-1)\alpha + 2\alpha + (2k-3)\alpha = f(\bar{y}) + 2\alpha = E_2(\bar{y},1)$).

sub-case 1.2: Assume $\bar{y} \Rightarrow x_{i_1}, \ldots, x_{i_{k-1}}$ and $\bar{y} \not\Rightarrow x_{i_k}$.

$$E_2(\bar{x},0) = E_1(\bar{x}) \leq E_1(\bar{y}) = f(\bar{y}) = E_2(\bar{y},1)$$

$$= f(\bar{y}) + \alpha - 2(k-1)\alpha + (2k-3)\alpha < E_2(\bar{y},0)$$

(since $E_2(\bar{y},0) = f(\bar{y}) + \alpha$).

sub-case 1.3: Assume $\bar{y} \not\Rightarrow x_{i_1}, \ldots, x_{i_{k-1}}$

$$E_2(\bar{x},0) = E_1(\bar{x}) \leq E_1(\bar{y}) = f(\bar{y}) = E_2(\bar{y},0) < f(\bar{y}) + \alpha \leq E_2(\bar{y},1).$$

Therefore $E_2(\bar{x},0) \leq E_2(\bar{y},t')$ for any $\bar{y}$ and t'.

case 2: Assume $\bar{x} \Rightarrow x_{i_1}, \ldots, x_{i_{k-1}}$ and $\bar{x} \not\Rightarrow x_{i_k}$.

(Using similar proof)

Therefore $E_2(\bar{x},1) \leq E_2(\bar{y},t')$ for any $\bar{y}$ and t'.

case 3: Assume $\bar{x} \not\Rightarrow x_{i_1}, \ldots, x_{i_{k-1}}$.

(Using similar proof)

$E_2(\bar{x},0) \leq E_2(\bar{y},t')$ for any $\bar{y}$ and t'.

Therefore if $E_1(\bar{x}) = min_{E_1}$ then $(\exists \bar{t}) E_2(\bar{x},\bar{t}) = min_{E_2}$.

ii) Assume $E_2(\bar{x},\bar{t}) = min_{E_2}$ we want to show that $E_1(\bar{x}) = min_{E_1}$.

case 1: Assume $\bar{x} \Rightarrow x_{i_1}, \ldots, x_{i_k}$

$$E_2(\bar{x},0) = f(\bar{x}) + \alpha < f(\bar{x}) + 2\alpha = E_2(\bar{x},1).$$

Therefore $E_2(\bar{x},0) = min_{E_2}$.

sub-case 1.1: Assume $\bar{y} \Rightarrow x_{i_1}, \ldots, x_{i_k}$ then:

$$E_1(\bar{x}) = E_2(\bar{x},0) \leq E_2(\bar{y},0) = f(\bar{y}) + \alpha = E_1(\bar{y}).$$

sub-case 1.2: Assume $\bar{y} \Rightarrow x_{i_1}, \ldots, x_{i_{k-1}}$ and $\bar{y} \not\Rightarrow x_{i_k}$.

$$E_1(\bar{x}) = E_2(\bar{x},0) \leq E_2(\bar{y},1) = f(\bar{y}) - E_1(\bar{y}).$$

sub-case 1.3: Assume $\bar{y} \not\Rightarrow x_{i_1}, \ldots, x_{i_{k-1}}$

$$E_1(\bar{x}) = f(\bar{x}) = E_2(\bar{x},0) \leq E_2(\bar{y},0) = f(\bar{y}) = E_1(\bar{y}).$$

Therefore $E_1(\bar{x}) < E_1(\bar{y})$ for any $\bar{y}$.

case 2: Assume $\bar{x} \Rightarrow x_{i_1}, \ldots, x_{i_{k-1}}$ and $\bar{x} \not\Rightarrow x_{i_k}$.

$$E_1(\bar{x}) = f(\bar{x}) = E_2(\bar{x}, 1) < f(\bar{x}) + \alpha = E_2(\bar{x}, 0).$$

Therefore $E_2(\bar{x}, 1) = min_{E_2}$.

(Using similar proof)

therefore $E_1(\bar{x}) \leq E_1(\bar{y})$ for any $\bar{y}$.

case 3: Assume $\bar{x} \not\Rightarrow x_{i_1}, \ldots, x_{i_{k-1}}$

$$E_1(\bar{x}) = f(\bar{x}) = E_2(\bar{x}, 0) < f(\bar{x}) + \alpha \leq E_2(\bar{x}, 1).$$

Therefore $E_2(\bar{x}, 0) = min_{E_2}$.

(Using similar proof)

$E_1(\bar{x}) \leq E_1(\bar{y})$ for any $\bar{y}$.

Therefore $E_1(\bar{x}) = min_{E_1}$ iff $(\exists \bar{t}) E_2(\bar{x}, \bar{t}) = min_{E_2}$.

Therefore $E_1 \approx E_2$.

□

A.3. Proof of theorem: eliminating hidden variable by converting terms to possibly higher order terms.

To show:

$$E_1(x_1, \ldots, x_n, T) = f(x) + \sum_{j=1}^{k} (\alpha_i X_j) T \approx f(\bar{x}) + \sum_{\beta_S < 0} \beta_S \prod_{j=1}^{k} L_S^j(X_j) = E_2(x_1, \ldots, x_n)$$

where:

$$\beta_S = \sum_{j=1}^{l} \alpha_j x_{i_j} < 0$$

and

$$L_S^j(X) = \begin{cases} X_{i_j} & \text{if } S(X_{i_j}) = 1 \\ 1 - X_{i_j} & \text{if } S(X_{i_j}) = 0 \end{cases}$$

Proof: We have generated the expression: $\prod L_S^j$ so that:

$$\prod_{j=1}^{k} L_S^j(S'(\hat{X})) = \begin{cases} 1 & \text{if } S' = S \\ 0 & \text{if } S' \neq S \end{cases}$$

Therefore, for every assignment S (and instantiation $\bar{x}$):

If $\beta_S \leq 0$ then $E_2(\bar{x}) = f(\bar{x}) + \beta_S$

If $\beta_S \geq 0$ then $E_2(\bar{x}) = f(\bar{x})$.

Like we did in the previous proofs of this appendix, we show now two directions:

i) If $\bar{x}$ can minimize E_1 then it minimizes also E_2.

ii) If $\bar{x}$ minimize E_2 then it can minimize also E_1.

In each direction we examine two cases:

case 1: We assume the $\bar{x}$ causes $\beta_S < 0$.

case 2: We assume the $\bar{x}$ causes $\beta_S \geq 0$.

For each of the cases we prove that $\bar{x}$ minimizes also the other function by showing that for every $\bar{y}$ the function gets a value that is greater or equal then the value it get for $\bar{x}$.

We examine two sub-cases:

sub-case 1: We assume the $\bar{y}$ causes $\beta_S < 0$.

sub-case 2: We assume the $\bar{y}$ causes $\beta_S \geq 0$.

i) Assume $\bar{x}$ is an instantiation of an assignment S that minimizes E_1,

case 1: If $\beta_S \leq 0$ then $E_1(\bar{x}, 1)$ is the minimum,

since $E_1(\bar{x}, 1) = f(\bar{x}) + \beta_S = E_2(\bar{x}) \leq f(\bar{x}) = E_1(\bar{x}, 0)$.

But then for all instantiations $\bar{y}$ of assignments S':

sub-case 1.1 : if $\beta_{S'} \leq 0$ then

$$E_1(\bar{x}, 0) = f(\bar{x}) + \beta_S \leq E_1(\bar{y}, 1) = f(\bar{y}, 1) + \beta_{S'} = E_2(\bar{y}), \text{ and}$$

sub-case 1.2 : if $\beta_{S'} > 0$ then

$$E_1(\bar{x}, 1) = f(\bar{y}) + \beta_S \leq E_1(\bar{y}, 0) = f(\bar{y}) = E_2(\bar{y})$$

Therefore $E_1(\bar{x}, 1) = E_2(\bar{x}) \leq E_2(\bar{y})$ for all instantiations $\bar{y}$.

case 2: If $\beta_S > 0$ then $E_1(\bar{x}, 0)$ is the minimum

since $E_1(\bar{x}, 0) = f(\bar{x}) = E_2(\bar{x}) < f(\bar{x}) + \beta_S = E_1(\bar{x}, 1)$.

But then for all instantiations $\bar{y}$ of assignment S' :

sub-case 2.1: if $\beta_{S'} \leq 0$ then

$$E_1(\bar{x}, 0) = f(\bar{x}) \leq E_1(\bar{y}, 1) = f(\bar{y}) + \beta_{S'} = E_2(\bar{y}) \text{ and}$$

sub-case 2.2: if $\beta_{S'} > 0$ then

$$E_1(\bar{x},0) = f(\bar{x}) \leq E_1(\bar{y},0) = f(\bar{y}) = E_2(\bar{y})$$

Therefore $E_1(\bar{x},1) = E_2(\bar{x}) \leq E_2(\bar{y})$.

Therefore, $\bar{x}$ also minimizes E_2.

ii) Assume $\bar{x}$ is an instantiation of S that minimize E_2.

case 1: If $\beta_S \leq 0$ then $E_2(\bar{x}) = f(\bar{x}) + \beta_S = E_1(\bar{x},1)$ is a minimum.

Then, for all instantiations $\bar{y}$ of S':

sub-case 1.1: If $\beta_{S'} \leq 0$ then

$$E_2(\bar{x}) = f(\bar{x}) + \beta_S \leq E_2(\bar{y}) = f(\bar{y}) + \beta_{S'} = E_1(\bar{y},1) \leq f(\bar{y}) = E_1(\bar{y},0)$$

sub-case 1.2: If $\beta_{S'} > 0$ then

$$E_2(\bar{x}) = f(\bar{x}) + \beta_S \leq E_2(\bar{y}) = f(\bar{y}) = E_1(\bar{y},0) \leq f(\bar{x}) + \beta_S = E_1(\bar{y},1)$$

Therefore $E_1(\bar{x},1) \leq E_1(\bar{y},T')$ for all $\bar{y}$ and T'

case 2: If $\beta_S > 0$ then $E_2(\bar{x}) = f(\bar{x}) = E_1(\bar{x},0)$ is a minimum.

Then, for all instantiations $\bar{y}$ of S':

sub-case 2.1: If $\beta_{S'} \leq 0$ then

$$E_2(\bar{x}) = f(\bar{x}) \leq E_2(\bar{y}) = f(\bar{y}) + \beta_{S'} = E_1(\bar{y},1) \leq f(\bar{y}) = E_1(\bar{y},0)$$

sub-case 2.2: If $\beta_{S'} > 0$ then

$$E_2(\bar{x}) = f(\bar{x}) \leq E_2(\bar{y}) = f(\bar{y}) = E_1(\bar{y},0) \leq f(\bar{x}) + \beta_S = E_1(\bar{y},1)$$

Therefore $E_1(\bar{x},0) \leq E_1(\bar{y},T')$ for all $\bar{y}$ and T'

Therefore $\bar{x}$ also minimizes E_1.

Therefore $E_2(\bar{x})$ is minimized iff there exists an instantiation to T such that $E_1(\bar{x},T)$ is minimized.

Therefore the minimizing sets of E_1 and E_2 (when T is a hidden variable in E_1) are equal. and $E_1 \approx E_2$.

□

A.4. Proof of theorem: the attribute grammar generates an equivalent WFF in a conjunction of triples form.

To show: The attribute grammar of section 4.1 transforms a WFF φ into an equivalent WFF ψ in a conjunction of triples form. (With the variables of φ as visible variables).

Proof:

By induction on k the depth of the parse tree that is generated by the grammar from φ, we show that:

If $S \stackrel{*}{\Rightarrow} \varphi$ then:

1. $S.val$ contains either: $var, (\neg var), (var1 \vee var2)$ or $(var1 \wedge var2)$ such that $var1$ and $var2$ are visible or hidden variables. We call such an expression a "simple term".

2. $S.t$ is either empty or contains a conjunction of expressions of the form: $(var \leftrightarrow T), ((\neg var) \leftrightarrow T), ((var_1 \vee var_2) \leftrightarrow T)$ or $((T_1 \wedge T_2) \leftrightarrow T_3)$ such that the T's on the right are new variables, not introduced before. (This is their first appearance from the left). Eg: $((((A \leftrightarrow T_1) \wedge (B \leftrightarrow T_2)) \wedge ((T_1 \vee T_2) \leftrightarrow T_3))$.

3. φ is equivalent to $(S.t \wedge S.val)$

Base: $k = 2$, when $S \rightarrow L \rightarrow X_i$ or $S \rightarrow L \rightarrow (\neg X_i)$ then

1. $S.val = X_i$ or $S.val = (\neg X_i)$

2. $S.t$ is empty

3. $(S.t \wedge S.val) = \varphi$

Step: Assume the Induction Hypothesis (I.H.) is true for $k < n$, to show that it is also true for $k = n$.

1. $S \rightarrow (S^1 \vee S^2) \stackrel{*}{\Rightarrow} (\varphi_1 \vee \varphi_2)$, by induction hypothesis: $S^1.val, S^2.val$ are simple terms, $S^1.t, S^2.t$ are conjunctions of triples.

By construction $S.t$ is also conjunction of triples and $S.val$ is a simple term.

We need now to show that φ is equivalent to $(S.t \wedge S.val)$:

TRUE always equivalent to $S.t$ because the new variables in the right of each triple *can* get a truth value equal to the truth value of the simple term on the left of it. Following this method, hidden variables are instantiated so that the whole conjunction is evaluated to be TRUE.

Assume: $\varphi = (\varphi_1 \vee \varphi_2)$ is instantiated to TRUE, then either φ_1 or φ_2 is instantiated to TRUE.

Assume φ_1 is TRUE, then by I.H $(S^1.t \wedge S^1.var)$ can be instantiated to be TRUE.

Therefore $(S^1.t \wedge (S^1.val \leftrightarrow T_1))$ can be made TRUE by making T_1 TRUE.

Therefore $(T_1 \vee T_2)$ then becomes TRUE and therefore $(S.t \wedge (T_1 \vee T_2))$ is TRUE.

Therefore: If φ is TRUE then $(S.t \wedge S.val)$ can be made TRUE by some instantiation to the hidden variables.

Assume $\psi = ((S^1.t \wedge S^2.t) \wedge (S^1.val \leftrightarrow T_1) \wedge (S^2.val \leftrightarrow T_2) / wedge(T_1 \vee T_2))$ is instantiated to be TRUE.

Then either T_1 or T_2 must be instantiated to TRUE.

Assume T_1 is true, then $S^1.val$ is TRUE.

Therefore $(S^1.t \wedge S^1.val)$ is TRUE . By I.H: $\varphi_1 \approx (S^1.t \wedge S^1.val)$.

Therefore φ_1 is TRUE. Therefore $(\varphi_1 \vee \varphi_2) = \varphi$ is TRUE.

Therefore if ψ is TRUE then φ is TRUE.

Therefore $\varphi \approx \psi$.

2. $S \rightarrow (S^1 \wedge S^2) \stackrel{*}{\Rightarrow} (\varphi_1 \wedge \varphi_2)$

   (Similar proof).

3. $S \rightarrow (\neg S^1) \stackrel{*}{\Rightarrow} \varphi$

   By construction, $S.val = (\neg T)$ is a simple term and $S.t$ is a conjunction of triples.

   We need now to show that $\varphi \approx (S.t \wedge S.val)$:

   Assume $(\neg\varphi)$ is TRUE by instantiation $\bar{x}$.

   Then φ_1 is instantiated to FALSE by $\bar{x}$ (of the visible variables).

   Therefore $(S^1.t \wedge S^1.val)$ can not be made TRUE by any instantiation to the new variables.

$S^1.t$ can always be made TRUE, but for those instantiation that make $S^1.t$ TRUE,

$S^1.val$ is always FALSE.

Therefore $((S^1.t \wedge (S^1.val \leftrightarrow T)) \wedge (\neg T))$ can be made TRUE

by making $S^1.t$ TRUE and T FALSE.

Therefore, for any instantiation $\bar{x}$ that instantiates φ to be TRUE,

$(S.t \wedge S.val)$ can be instantiated also to be TRUE.

Assume $\psi = ((S^1.t \wedge (S.val \leftrightarrow T)) \wedge (\neg T)$ is instantiated to TRUE by $\bar{x}$ and T,

then T must be FALSE and $S.val$ is also FALSE.

Therefore, $(S^1.t \wedge S^1.val)$ is FALSE.

$\varphi_1 \approx (S^1.t \wedge S^1.val)$ so φ_1 is FALSE.

Therefore φ_1 is FALSE for any instantiation that makes ψ TRUE.

Therefore, $\varphi = (\neg\varphi_1)$ is TRUE for any instantiation that makes ψ TRUE.

Therefore $\psi \approx \varphi$.

□

A.5. Proof of lemma: H_E is the characteristic function of φ.

To show that $H_E(\bar{x}) = 1$ iff $\varphi(\bar{x}) = 1$

Where

$$H_E(\bar{x}) = \begin{cases} 1 & \text{if } E(\bar{x}) = min_E \\ 0 & \text{Otherwise} \end{cases}$$

and

$$\varphi = \bigvee_{H_E(S(\hat{X}))=1} \left(\bigwedge_{j=1}^{n} L_S^j\right)$$

where

$$L_S^j = \begin{cases} X_i & \text{if } S(X_i) = 1 \\ (\neg X_i) & \text{if } S(X_i) = 0 \end{cases}$$

Proof: We have constructed the term

$$t_S = \bigwedge_{j=1}^{n} L_S^j$$

so that t_S has the property:

$$t_S(S'(\hat{X})) = \begin{cases} 1 & \text{if } S = S' \\ 0 & \text{if } S \neq S' \end{cases}$$

(i.e: t_S is evaluated to be TRUE only by assignment S).

Let S be the assignment for instantiation $\bar{x}$. If $H_E(\bar{x}) = 1$ then $H_E(S(\hat{X})) = 1$.

Therefore t_S is included in φ.

$t_S(S(\hat{X})) = 1$ (property of t_S).

Therefore $\varphi(S(\hat{X})) = \bigvee'_{H_E(S'(\hat{X}))=1} t_S(S'(\hat{X})) = 1$

Therefore $H_E(S(\hat{X})) \Rightarrow \varphi(S(\hat{X})) = 1$.

If $\varphi(\bar{x}) = 1$ then at least one of the terms $t_{S'}$ is instantiated to TRUE: $t_{S'}(S(\hat{X})) = 1$

Therefore $S = S'$ (Property of t_S).

Therefore $H_E(S(\hat{X})) = 1$ (since t_S is included in φ, by the construction of φ).

Therefore $\varphi(S(\hat{X})) = 1 \Rightarrow H_E(S(\hat{X})) = 1$.

Therefore $\varphi(\bar{x}) = 1$ iff $H_E(\bar{x}) = 1$.

Therefore H_E is the characteristic function of φ.

□

References

[1] J. J. Hopfield "Neural networks and physical system with emergent collective computational abilities," *Proceedings of the National Academy of Sciences USA, 1982,79,2554-2558.*

[2] J.J Hopfield "Neurons with graded response have collective computational properties *Proceedings of the National Academy of Science USA* 81, 3088-3092 (1984)

[3] G.E Hinton , T.J Sejnowski " learning and re-learning in Boltzman Machines" in J. L. McClelland, D. E. Rumelhart "Parallel Distributed Processing:Explorations in the Microstructure of Cognition" Vol I pp. 282 - 317 *MIT Press* 1986

[4] P. Smolensky "Information Processing in Dynamical Systems: Foundations of Harmony Theory", in J. L. McClelland, D. E. Rumelhart "Parallel Distributed Processing:Explorations in the Microstructure of Cognition" Vol I, pp 194-281. MIT Press 1986

[5] S. Kirkpatrick, C.D. Gelatt, M. P. Vecchi, "Optimization by simulated annealing," *Science*, 1983, 220, 671-680

[6] J.J. Hopfield, D.W. Tank, "Neural Computation of decisions in optimizations problems" *Biological Cybernetics, Vol 52, pp. 144-152.*

[7] D. H. Ballard "Parallel Logical Inference and Energy Minimization" *Proceedings of the 5th National conference on Artificial Intelligence* Philadelphia, Pa., August 1986, pp. 203-208

[8] D.S. Touretzky, G.E. Hinton, "A distributed connectionist production system" *Cognitive Science* 12(3):423-466. 1988.

[9] M. Derthick "A Connectionist Architecture for representing and reasoning about structured knowledge", *Proceedings of the Ninth Cognitive Science Society*, 1987, Seattle, WA.

[10] M.R Garey, D.S Johnson "Computers and Intractability - A Guide to the theory of NP- Completeness" *(W.H. Freeman and Company San Francisco)* 1979