More Games of No Chance
MSRI Publications
Volume **42**, 2002

# The Game of Hex: The Hierarchical Approach

Abstract. Hex is a beautiful and mind-challenging game with simple rules and a strategic complexity comparable to that of Chess and Go. Hex positions do not tend to decompose into sums of independent positions. Nevertheless, we demonstrate how to reduce evaluation of Hex positions to an analysis of a hierarchy of simpler positions. We explain how this approach is implemented in Hexy, the strongest Hex-playing computer program, and Gold Medalist of the 5th Computer Olympiad in London, August 2000.

## 1. Introduction

The rules of Hex are extremely simple. Nevertheless, Hex requires both deep strategic understanding and sharp tactical skills. The massive game-tree search techniques developed over the last 30–40 years mostly for Chess (Adelson-Velsky, Arlazarov, and Donskoy 1988; Marsland 1986), and successfully used for Checkers (Schaeffer et al. 1996), and a number of other games, become less useful for games with large branching factors like Hex and Go. For a classic $11 \times 11$ Hex board the average number of legal moves is about 100 (compare with 40 for Chess and 8 for Checkers).

Combinatorial (additive) Game Theory provides very powerful tools for analysis of sums of large numbers of relatively simple games (Conway 1976; Berlekamp, Conway, and Guy 1982; Nowakowski 1996), and can be also very useful in situations, when complex positions can be decomposed into sums of simpler ones. This method is particularly useful for an analysis of Go endgames (Berlekamp and Wolfe 1994; Müller 1999).

Hex positions do not tend to decompose into these types of sums. Nevertheless, many Hex positions can be considered as combinations of simpler subgames. We concentrate on the hierarchy of these subgames and define a set of deduction rules, which allow to calculate values of complex subgames recursively, starting from the simplest ones. Integrating the information about subgames of this hierarchy, we build a far-sighted evaluation function, foreseeing the potential of Hex positions many moves ahead.

In Section 2 we introduce the game of Hex and its history. In Section 3 we discuss the concept of virtual connections. In Section 4 we introduce the AND

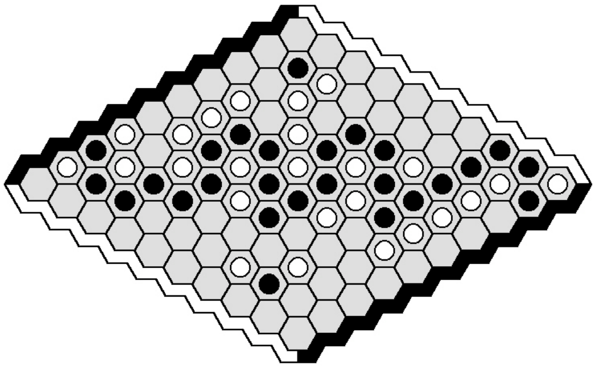

**Figure 1.** The chain of black pieces connects black boundaries. Black has won the game.

and OR deduction rules. In Section 5 we show how to recursively calculate the hierarchy of virtual connections. In Section 6 we present an electrical resistor circuits model, which allows us to combine information about the hierarchy of virtual connections into a global evaluation function. In Section 7 we explain how this approach is implemented in Hexy, the strongest Hex-playing computer program, and the Gold Medallist of the 5th Computer Olympiad in London, August 2000. A Windows version of the program is publicly available at http://home.earthlink.net/~vanshel.

The major ideas of this work were presented on the MSRI Combinatorial Game Theory Workshop in Berkeley, July 2000 and on the 17th National Conference on Artificial Intelligence in Austin, July-August 2000 (Anshelevich 2000a).

## 2. Hex and Its History

The game of Hex was introduced to the general public in Scientific American by Martin Gardner (Gardner 1959). Hex is a two-player game played on a rhombic board with hexagonal cells (see Figure 1). The classic board is $11 \times 11$, but it can be any size. The $10 \times 10$, $14 \times 14$ and even $19 \times 19$ board sizes are also popular. The players, Black and White, take turns placing pieces of their color on empty cells of the board. Black's objective is to connect the two opposite black sides of the board with a chain of black pieces. White's objective is to connect the two opposite white sides of the board with a chain of white pieces (see Figure 1). The player moving first has a big advantage in Hex. In order to equalize chances, players often employ a "swap" rule, where the second player has the option of taking the first player's opening move.

Despite the simplicity of the rules, the game's strategic and tactical ideas are rich and subtle. An introduction to Hex strategy and tactics can be found in the book written by Cameron Browne (Browne 2000).

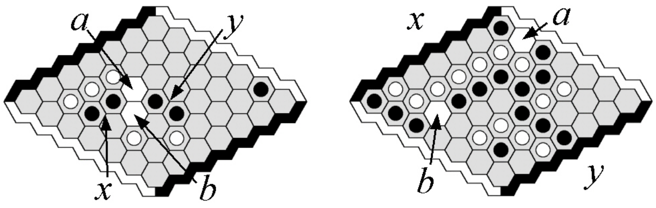


**Figure 2.** Groups of black pieces, $x$ and $y$, form two-bridges. In the position on the right, those groups are connected to the black boundaries.

Hex was invented by a Danish poet and mathematician Piet Hein in 1942 at the Niels Bohr Institute for Theoretical Physics, and became popular under the name of Polygon. It was rediscovered in 1948 by John Nash, when he was a graduate student at Princeton (Gardner 1959). Parker Brothers marketed a version of the game in 1952 under the name Hex.

The game of Hex can never end in a draw. This follows from the fact that if all cells of the board are occupied then a winning chain for Black or White must necessarily exist. While this two-dimensional topological fact may seem obvious, it is not at all trivial. In fact, David Gale demonstrated that this result is equivalent to the Brouwer fixed-point theorem for 2-dimensional squares (Gale 1979). It follows that there exists a winning strategy either for the first or second player. Using a "strategy stealing" argument (Berlekamp, Conway, and Guy 1982), John Nash showed that a winning strategy exists for the first player. However, this is only a proof of existence, and the winning strategy is not known for boards larger than $7 \times 7$.

S. Even and R. E. Tarjan (Even and Tarjan 1976) showed that the problem of determining which player has a winning strategy in a generalization of Hex, called the Shannon switching game on vertices, is PSPACE complete. A couple of years later S. Reisch (Reisch 1981) proved this for Hex itself.

A Hex-playing machine was built by Claude Shannon and E. F. Moore (Shannon 1953). Shannon associated a two-dimensional electrical charge distribution with any given Hex position. His machine made decisions based on properties of the corresponding potential field. We gratefully acknowledge that our work is greatly inspired by the beauty of the Shannon's original idea.

## 3. Virtual Connections and Semi-Connections

In this and the two following sections we characterize Hex positions from Black's point of view. White's point of view can be considered in a similar way.

Consider the four polygonal boundary bands as additional cells (see Figure 1). We assume that black boundary cells are permanently occupied by black pieces, and white boundary cells are permanently occupied by white pieces.

Consider the two positions in Figure 2. In both positions White cannot prevent Black from connecting the two groups of connected black pieces, $x$ and $y$,

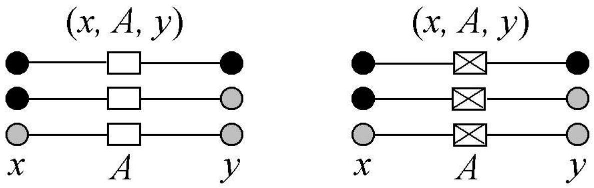


**Figure 3.** Diagrams of virtual connections (on the left) and virtual semi-connections (on the right): black-black, black-empty, and empty-empty.

even if White moves first, because there are two empty cells $a$ and $b$ adjacent to both $x$ and $y$. If White occupies one of those empty cells, then Black can move to the other. Note that the black connection between groups x and y is secured as long as two cells $a$ and $b$ stay empty. Black can postpone moving to either $a$ or $b$ and can use his precious moves for other purposes. In this type of situation we say that the groups of black pieces $x$ and $y$ form a *two-bridge*. In a battle, where Black tries to connect groups $x$ and $y$, and White tries to prevent it, the result of this battle is predictable two moves ahead. This provides an important advantage to Black. In the position on the left this advantage is local. In the position on the right this advantage is decisive, and White should resign immediately.

The following definitions generalize the two-bridge concept. First we need to clarify some terms. We say that a cell is *black* if and only if it is occupied by a black piece, and we refer to a group of connected black cells as a single black cell.

**Definition.** Let $x$ and $y$ be two different cells, and $A$ be a set of empty cells of a position. We assume that $x \notin A$ and $y \notin A$. The triplet $(x, A, y)$ defines a *subgame*, where Black tries to connect cells $x$ and $y$ with a chain of black pieces, White tries to prevent it, and both players can put their pieces only on cells in $A$. We say that $x$ and $y$ are *ends* of the subgame, and $A$ is its *carrier*.

**Definitions.** A subgame is a *virtual connection* if and only if Black has a winning strategy even if White moves first.

A subgame is a *virtual semi-connection* if and only if Black has a winning strategy if he moves first, and does not have one if he moves second.

We represent virtual connections and semi-connections with diagrams as in Figure 3.

In practice, it is more convenient to use the following recursive definitions.

**Definitions.** A subgame is a virtual connection if and only if for every White's move there exists a Black's move such that the resulting subgame is a virtual connection.

A subgame is a virtual semi-connection if and only if it is not a virtual connection, and there exists a Black's move such that the resulting subgame is a virtual connection.

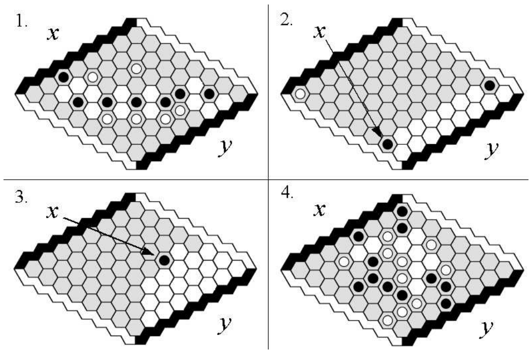


**Figure 4.** Black cells $x$ and $y$ form virtual connections. In each diagram the cell y is formed by the black pieces connected to the bottom right black boundary. The cells of their carriers are marked white. 1: A chain of two-bridges; $depth = 12$. 2: A ladder; $depth = 14$. 3: An edge connection from the fourth row; $depth = 10$. 4: this virtual connection will be analyzed in the next section; $depth = 6$.

Assume that in a given position with a virtual connection, White moves first. The number of moves, which must be made in order for Black to win this subgame, under the condition that Black does his best to minimize this number, and White does his best to maximize it, characterizes the *depth of the virtual connection.* In other words, the depth of virtual connection is a depth of a game-tree search required to discover this virtual connection. Thus, virtual connections with the depth $d$ contain information about development of Hex position $d$ moves ahead.

We make several remarks:

- Pairs of neighboring cells form virtual connections with empty carriers. The depths of these virtual connections are equal to zero.
- Two-bridges form virtual connections with a depth of two.
- The ends $x$ and $y$ can form virtual connections with several different carriers. The virtual connection $(x, A, y)$ is *minimal* if and only if there does not exist a virtual connection $(x, B, y)$ such that $B \subset A$ and $B \neq A$. We will be primarily interested in minimal virtual connections.
- A special role is played by a *winning virtual connection* formed by the additional boundary cells. If it exists, then there exists a global winning strategy for Black, even if White moves first.

In Figures 4 and 5 you can see samples of virtual connections and virtual semi-connections.

## 4. Deduction Rules

In this section we define two binary operations, conjunction ($\wedge$) and disjunction ($\vee$), on the set of subgames belonging to the same position. These

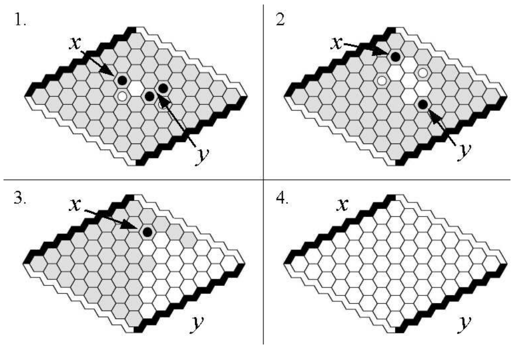


**Figure 5.** Black cells $x$ and $y$ form virtual semi-connections. The cells of their carriers are marked white. Diagram 4 shows the initial position. According to the Nash theorem mentioned in Section 2, the initial position is a virtual semi-connection.

operations will allow us to build complex virtual connections starting from the simplest ones.

**Definition.** Let two subgames $G = (x, A, u)$ and $H = (u, B, y)$ with common end $u$ and different ends $x \neq y$ belong to the same position, and $x \notin B$, $y \notin A$.

If common end $u$ is black, then conjunction of these subgames is the subgame $G \wedge H = (x, A \cup B, y)$.

If common end $u$ is empty, then conjunction of these subgames is the subgame $G \wedge H = (x, A \cup u \cup B, y)$.

**Definition.** Let two subgames $G = (x, A, y)$ and $H = (x, B, y)$ with common ends $x$ and $y$ belong to the same position. Then disjunction of these subgames is the subgame $G \vee H = (x, A \cup B, y)$.

**Theorem 1** (The AND deduction rule). *Let two subgames $G = (x, A, u)$ and $H = (u, B, y)$ with common end $u$ and different ends $x \neq y$ belong to the same position, and $x \notin B$, $y \notin A$. If both subgames $G$ and $H$ are virtual connections and $A \cap B = \varnothing$, then*

(a) *$G \cap H$ is a virtual connection if $u$ is black, and*

(b) *$G \cap H$ is a virtual semi-connection if $u$ is empty.*

*Proof.* If cell $u$ is empty, then Black can occupy this cell, and this reverts to case (a). Since $A \cap B = \varnothing$, White cannot attack both virtual connections simultaneously. Suppose that White occupies a cell $a \in A$. Since the subgame $G = (x, A, u)$ is a virtual connection, then there exists a cell $b \in A$ where Black can play to create a new virtual connection $(x, A', u)$. The new carrier $A'$ is obtained from $A$ by removing two cells $a$ and $b$. (The new virtual connection belongs to a position different than the original one). In short, if White occupies a cell from $A$, then Black can restore the first virtual connection by moving to

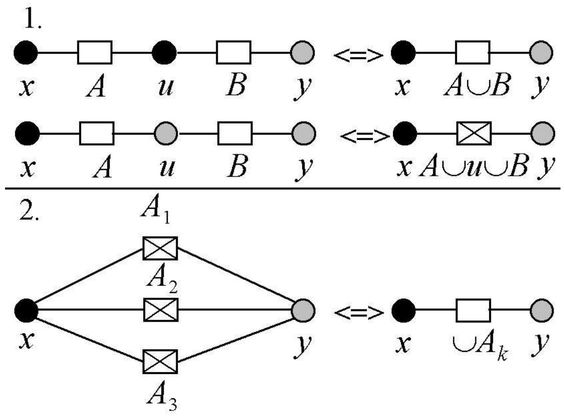


**Figure 6.** 1: The AND deduction rule. 2: The OR deduction rule.

an appropriate cell of $A$. The same is true for $B$, and thus the result follows by induction. □

Diagram 1 in Figure 6 shows a graphical representation of this deduction rule.

**Theorem 2** (The OR deduction rule). *Let subgames $G_k = (x, A_k, y)$ $(k = 1, 2, \ldots, n$, for $n > 1)$ with common ends $x$ and $y$ belong to the same position. If all games $G_k$ are virtual semi-connections and*

$$\bigcap_{k=1}^{n} A_k = \varnothing,$$

*then $G = \bigvee G_k$ is a virtual connection.*

*Proof.* If White occupies a cell $a \in A_i$, there exists a different carrier $A_j$ such that $a \notin A_j$. Therefore, Black can move to $A_j$ and convert the virtual semi-connection $G_j$ to a virtual connection. □

Diagram 2 in Figure 6 graphically represents this deduction rule (for $n = 3$).

**Theorem 3** (The OR decomposition). *Let a subgame $G = (x, A, y)$ be a minimal virtual connection, with $A \neq \varnothing$. There exist virtual semi-connections $G_k = (x, A_k, y)$ $(k = 1, 2, \ldots, n$, for $n > 1)$ such that*

$$\bigcap_{k=1}^{n} A_k = \varnothing$$

*and $G = \bigvee G_k$.*

*Proof.* Since $G$ is a minimal virtual connection, then for every White's move $a \in A$, the game $G_a = (x, A - a, y)$ is a virtual semi-connection. Besides, $G = \bigvee G_a$ and

$$\bigcap_{a} A - a = \varnothing. \qquad \square$$

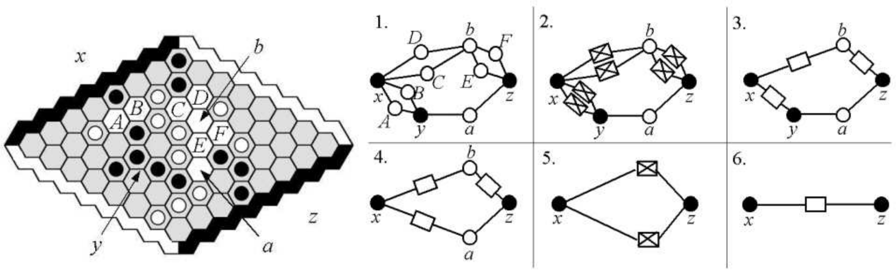


**Figure 7.** Diagram 1 represents the subgame on the board. Diagram 3 is obtained from Diagram 1 by applying the AND deduction rule six times and then the OR deduction rule three times. Diagram 4 results from the AND deduction rule. The winning virtual connection in Diagram 6 follows from application of the AND deduction rule 2 times and final application of the OR deduction rule.

The last theorem means that the OR deduction rule provides a universal way of building virtual connections from virtual semi-connections. On the other hand, there exist virtual semi-connections, which cannot be obtained from virtual connections by applying the AND deduction rule. An example will be given in the next section.

## 5. Hierarchy of Virtual Connections

Figure 7 demonstrates how the AND and OR deduction rules can be used for proving virtual connections. Diagram 1 in Figure 7 represents the position on the board. The sequence of transformations in diagrams 2 through 6 graphically demonstrates the application of the AND and OR deduction rules, and proves that Black has a winning position, even if White moves first.

**The H-process.** Consider the simplest virtual connections, namely pairs of neighboring cells, as the first generation of virtual connections. Applying the AND deduction rule to the appropriate groups of the first generation of virtual connections we build the second generation of virtual connections and semi-connections. Then we apply the AND and OR deduction rules to both the first and the second generations of virtual connections and semi-connections to build the third generation of virtual connections and semi-connections, etc. This process stops when no new virtual connections are produced.

In general, this process can start from any initial set of virtual connections and semi-connections.

This iterative process can build all of the virtual connections shown in Figures 2 and 4. A formal proof for the subgame on Diagram 2 in Figure 4 is provided in Appendix as an example.

Is the set of the AND and OR deduction rules complete, i.e. can this process build all virtual connections? The answer is negative. The diagram in Figure 8

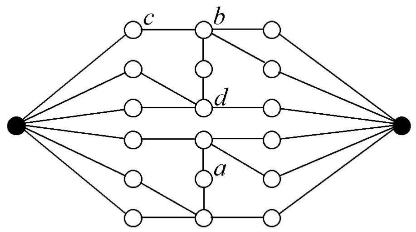


**Figure 8.** The two black cells form a virtual connection, which cannot be built using the AND and OR deduction rules.

represents a counter-example of a virtual connection that cannot be built by the H-process.

It is easy to check that this subgame is a virtual connection. Indeed, if White plays at $a$, Black can reply with $b$, forcing White to occupy $c$. Then Black plays d securing the win. This virtual connection is a disjunction of two equivalent virtual semi-connections with disjoint carriers. A computer program was used to verify that no combination of the AND and OR deduction rules can establish neither these virtual semi-connections nor the overall virtual connection.

## 6. Electrical Resistor Circuits

The H-process introduced in the previous section is useful in two ways. First, in some positions it can reach the ultimate objective by building a winning virtual connection between either black or white boundaries. Second, even if it is impossible due to incompleteness of the AND and OR deduction rules and the limited computing resources, the information about connectivity of subgames is useful for the evaluation of the entire position.

In this section we introduce a family of evaluation functions based on an *electrical resistor circuit* representation of Hex positions. One can think of an electrical circuit as a graph. Edges of the graph play a role of electrical links (resistors). The resistance of each electrical link is equal to the length of the corresponding edge of the graph. Yet, we consider that the "electrical circuit" language better suits our needs.

With every Hex position we associate two electrical circuits. The first one characterizes the position from Black's point of view (Black's circuit), and the second one from White's point of view (White's circuit). To every cell $c$ of the board we assign a resistance $r$ in the following way:

$$r_B(c) = \begin{cases} 1, & \text{if } c \text{ is empty,} \\ 0, & \text{if } c \text{ is occupied by a black piece,} \\ +\infty, & \text{if } c \text{ is occupied by a white piece,} \end{cases}$$

for Black's circuit, and

$$r_W(c) = \begin{cases} 1, & \text{if } c \text{ is empty,} \\ 0, & \text{if } c \text{ is occupied by a white piece,} \\ +\infty, & \text{if } c \text{ is occupied by a black piece,} \end{cases}$$

for White's circuit. For each pair of neighboring cells, $(c_1, c_2)$, we associate an electrical link with resistance

$$r_B(c_1, c_2) = r_B(c_1) + r_B(c_2), \quad \text{for Black's circuit,}$$
$$r_W(c_1, c_2) = r_W(c_1) + r_W(c_2), \quad \text{for White's circuit.}$$

These circuits take into account only virtual connections between neighboring cells, and describe the microstructure of Hex position.

We are now going to enhance these circuits by including information about more complex known virtual connections. We focus on Black's circuits only. White's circuits can be dealt with in a similar way.

A seemingly natural way of doing this is to add an additional electrical link between cells $x$ and $y$ to Black's circuit if $x$ and $y$ form a virtual connection. Then all virtual connections would be treated as neighboring cells. However, virtual connections between nearest neighbors are stronger than other virtual connections, so our circuit should reflect this. Instead of connecting black cells $x$ and $y$ with a shortcut, we add other links to Black's circuit in the following way. If an empty cell $c$ is a neighbor of one of the ends of this virtual connection, say $x$, then we also treat this cell $c$ as a neighbor of the other end $y$. This means that we connect cells $c$ and $y$ with an additional electrical link in the same way as actual neighbors.

Let $R_B$ and $R_W$ be distances between black boundaries in Black's circuit and between white boundaries in White's circuit, correspondingly. We define an evaluation function:

$$E = \log(R_B/R_W).$$

One of the reasonable distance metrics is the length of the shortest path on the graph, connecting boundaries. We can also measure distances in a different way. Apply an electrical voltage to the opposite boundaries of the board and measure the total resistance between them, $R_B$ for Black's circuit, and $R_W$ for White's circuit (see Figure 9).

We prefer this way for measuring distances, because according to the Kirchhoff electrical current laws, the total resistance takes into account not only the length of the shortest path, but also all other paths connecting the boundaries, their lengths, and their intersections.

Virtual connections with the depth $d$ contain information about development of Hex position $d$ moves ahead. Thus, we can expect that by including electrical links, which correspond to virtual connections with depth less or equal than $d$, we obtain an evaluation function with foreseeing abilities up to $d$ moves ahead.

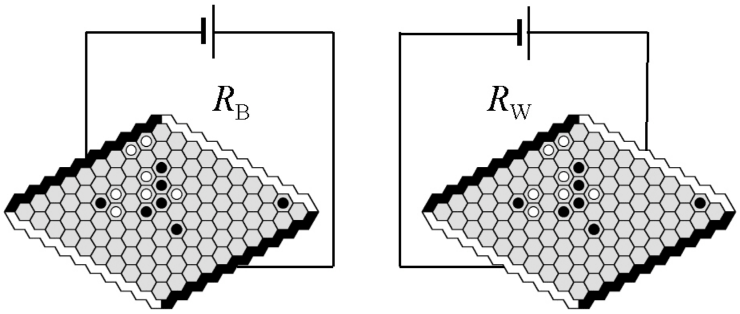


**Figure 9.** Black's and White's circuits.

## 7. Hexy Plays Hex

Hexy is a Hex-playing computer program, which utilizes the ideas presented in this paper. It runs on a standard PC with Windows, and can be downloaded from the website http://home.earthlink.net/~vanshel.

Hexy uses a selective alpha-beta search algorithm, with the evaluation functions described in the previous section. For every node to be evaluated, Hexy calculates the hierarchy of virtual connections for both Black's and White's circuits using the H-process described in Section 5. Then Hexy calculates distances $R_B$ and $R_W$ between Black and White boundaries, correspondingly. For calculation of the shortest path, a version of Dijkstra algorithm is applied. For calculation of the total electrical resistance between boundaries, Hexy solves the Kirchhoff system of linear equations using a method of iterations (see (Strang 1976), for example).

In practice, Hexy does not start the H-process from pairs of neighboring cells, but looks for changes in the hierarchy of virtual connections, caused by an additional piece, placed on the board. Besides, the program keeps track of only minimal available virtual connections and semi-connections.

The program has two important thresholds, $D$ and $N$. The parameter $D$ is the depth of the game-tree search. The second parameter, $N$, sets the limit to the number of different minimal virtual connections with the same ends, built by the program. This threshold indirectly controls the total number of calculated minimal virtual connections. The larger $N$, the more minimal virtual connections the H-process builds for every node of the game-tree. However, we do not put any limits on the number of iterations of the algorithm, or the total number of virtual connections, or their depths. The process stops when the next iteration of the algorithm does not produce new virtual connections.

There is an obvious trade-off between parameters $D$ and $N$, and finding an optimum is an important task. Since our major objective has been a creation of Hex-playing program, which can provide fun for Hex fans, we confine ourselves by a condition, that Hexy should be able to complete a game on the $10 \times 10$ board for less then 8 minutes on a standard PC with 300 MHz processor and

32 MB RAM. Thus, we try to find optimal (in terms of playing strength) values of these thresholds, satisfying the above condition. Experiments show that the dependence of Hexy's strength on the parameter N, which controls the number and the depth of virtual connections, is much more dramatic than its dependence on the depth D of game-tree search. The best results determined experimentally, are obtained with values of $D = 3$ and $N = 20$ (for a $10 \times 10$ board). This version of Hexy (called Advanced level) performs a very shallow game-tree search (200–500 nodes per move), but routinely detects virtual connections with depth 20 or more. It means that this version of Hexy routinely foresees some lines of play 20 or more moves ahead.

Hexy demonstrates a clear superiority over all known Hex-playing computer programs. This program won a Hex tournament of the 5th Computer Olympiad in London on August 2000, with the perfect score (Anshelevich 2000b).

Hexy was also tested against human players on the popular game website Playsite (http://www.playsite.com/games/board/hex). Hexy cannot compete with the best human players. Nevertheless, after more than 100 games, the program achieved a rating within the highest Playsite red rating range.

## 8. Conclusion

In this paper we have described a hierarchical approach to the game of Hex, and explained how this approach is implemented in Hexy - a Hex-playing computer program. Hexy does not perform massive game-tree search. Instead, this program spends most computational resources on deep analysis of a relatively small number of Hex positions.

We have concentrated on a hierarchy of positive subgames of Hex positions, called virtual connections, and have defined the AND and OR deduction rules, which allow to build complex virtual connections recursively, starting from the simplest ones. Integrating the information about virtual connections of this hierarchy, we have built a far-sighted evaluation function, foreseeing the potential of Hex positions many moves ahead.

The process of building virtual connections, the H-process, has its own cost. Nevertheless, the resulting foreseeing abilities of the evaluation function greatly outweigh its computational cost. This approach is much more efficient than brute-force search, and can be considered as both alternative and complimentary to the alpha-beta game-tree search.

## Acknowledgements

I would like to express my gratitude to the organizers and the participants of the Combinatorial Game Theory Workshop in MSRI, Berkeley, July 2000, for the exciting program and fruitful discussions.

## Appendix

In this Appendix we show how to prove that the ladder in Figure 10 is a virtual connection using the AND and OR deduction rules. We use abbreviation VC for virtual connections, VSC for virtual semi-connections, brackets [ ] for carriers, and parentheses () for subgame triplets.

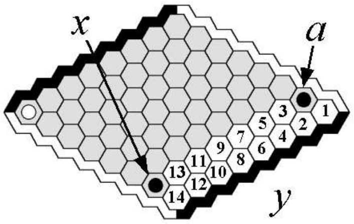


**Figure 10.** Black cells $x$ and $y$ form a ladder. The cells of the carrier are enumerated.

Examples:
$[a, b]$ is a carrier consisting of two cells $a$ and $b$.
$[\,]$ is an empty carrier.
$(x, [a, b, c, d], y)$ is a subgame with ends x and y and carrier $[a, b, c, d]$.
$(x, [\,], y)$ is a subgame with ends x and y and an empty carrier.

The following sequence of deductions proves that the subgame in Figure 10 is a virtual connection.

$(3, [\,], a)$ is VC, $(a, [\,], 1)$ is VC. Apply AND: $(3, [\,], 1)$ is VC.

$(3, [\,], 1)$ is VC, $(1, [\,], y)$ is VC. Apply AND: $(3, [1], y)$ is VSC.
$(3, [\,], 2)$ is VC, $(2, [\,], y)$ is VC. Apply AND: $(3, [2], y)$ is VSC.
$(3, [1], y)$ is VSC, $(3, [2], y)$ is VSC. Apply OR: $(3, [1, 2], y)$ is VC.

$(5, [\,], 3)$ is VC, $(3, [1, 2], y)$ is VC. Apply AND: $(5, [1, 2, 3], y)$ is VSC.
$(5, [\,], 4)$ is VC, $(4, [\,], y)$ is VC. Apply AND: $(5, [4], y)$ is VSC.
$(5, [1, 2, 3], y)$ is VSC, $(5, [4], y)$ is VSC. Apply OR: $(5, [1, 2, 3, 4], y)$ is VC.

$(7, [\,], 5)$ is VC, $(5, [1, 2, 3, 4, ], y)$ is VC. Apply AND: $(7, [1, 2, 3, 4, 5], y)$ is VSC.
$(7, [\,], 6)$ is VC, $(6, [\,], y)$ is VC. Apply AND: $(7, [6], y)$ is VSC.
$(7, [1, 2, 3, 4, 5], y)$ is VSC, $(7, [6], y)$ is VSC. Apply OR: $(7, [1, 2, 3, 4, 5, 6], y)$ is VC.

$(9, [\,], 7)$ is VC, $(7, [1, 2, 3, 4, 5, 6], y)$ is VC. Apply AND: $(9, [1, 2, 3, 4, 5, 6, 7], y)$ is VSC.
$(9, [\,], 8)$ is VC, $(8, [\,], y)$ is VC. Apply AND: $(9, [8], y)$ is VSC.
$(9, [1, 2, 3, 4, 5, 6, 7], y)$ is VSC, $(9, [8], y)$ is VSC. Apply OR: $(9, [1, 2, 3, 4, 5, 6, 7, 8], y)$ is VC.

$(11, [\,], 9)$ is VC, $(9, [1,2,3,4,5,6,7,8], y)$ is VC. Apply AND: $(11, [1,2,3,4,5,6,7,8,9], y)$ is VSC.
$(11, [\,], 10)$ is VC, $(10, [\,], y)$ is VC. Apply AND: $(11, [10], y)$ is VSC.
$(11, [1,2,3,4,5,6,7,8,9], y)$ is VSC, $(11, [10], y)$ is VSC. Apply OR: $(11, [1,2,3,4,5,6,7,8,9,10], y)$ is VC.

$(13, [\,], 11)$ is VC, $(11, [1,2,3,4,5,6,7,8,9,10], y)$ is VC. Apply AND: $(13, [1,2,3,4,5,6,7,8,9,10,11], y)$ is VSC.
$(13, [\,], 12)$ is VC, $(12, [\,], y)$ is VC. Apply AND: $(13, [12], y)$ is VSC.
$(13, [1,2,3,4,5,6,7,8,9,10,11], y)$ is VSC, $(13, [12], y)$ is VSC. Apply OR: $(13, [1,2,3,4,5,6,7,8,9,10,11,12], y)$ is VC.

$(x, [\,], 13)$ is VC, $(13, [1,2,3,4,5,6,7,8,9,10,11,12], y)$ is VC. Apply AND: $(x, [1,2,3,4,5,6,7,8,9,10,11,12,13], y)$ is VSC.
$(x, [\,], 14)$ is VC, $(14, [\,], y)$ is VC. Apply AND: $(x, [14], y)$ is VSC.
$(x, [1,2,3,4,5,6,7,8,9,10,11,11,12,13], y)$ is VSC, $(x, [14], y)$ is VSC. Apply OR: $(x, [1,2,3,4,5,6,7,8,9,10,11,12,13,14], y)$ is VC.

## References


Adelson-Velsky, G.; Arlazarov, V.; and Donskoy, M. 1988. Algorithms for Games. Springer-Verlag.

Anshelevich, V. V. 2000a. The Game of Hex: An Automatic Theorem Proving Approach to Game Programming. Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000), 189–194, AAAI Press, Menlo Park, CA.

Anshelevich, V. V. 2000b. Hexy Wins Hex Tournament. The ICGA Journal, 23(3):181–184.

Berlekamp, E. R.; Conway, J. H.; and Guy, R. K. 1982. Winning Ways for Your Mathematical Plays. New York: Academic Press.

Berlekamp, E.R.; and Wolfe, D. 1994. Mathematical Go: Chilling Gets the Last Point. A. K. Peters, Wellesley.

Browne, C. 2000. Hex Strategy: Making the Right Connections. A. K. Peters, Natick, MA.

Conway, J. H. 1976. On Numbers and Games. London. Academic Press.

Even, S.; and Tarjan, R. E. 1976. A Combinatorial Problem Which Is Complete in Polynomial Space. Journal of the Association for Computing Machinery 23(4): 710–719.

Gale, D. 1979. The Game of Hex and the Brouwer Fixed-Point Theorem. American Mathematical Monthly 86: 818–827.

Gardner, M. 1959. The Scientific American Book of Mathematical Puzzles and Diversions. New York: Simon and Schuster.

Marsland, T. A. 1986. A Review of Game-Tree Pruning. Journal of the International Computer Chess Association 9(1): 3–19.

Müller, M. 1999. Decomposition search: A combinatorial games approach to game tree search, with applications to solving Go endgames. In IJCAI-99, 1: 578–583.

Nowakowski, R. (editor) 1996. Games of No Chance. MSRI Publications 29. New York. Cambridge University Press.

Reisch, S. 1981. Hex ist PSPACE-vollständig. Acta Informatica 15: 167–191.

Schaeffer, J.; Lake, R.; Lu, P.; and Bryant M. 1996. Chinook: The World man-Machine Checkers Champion. AI Magazine 17(1): 21–29.

Shannon, C. E. 1953. Computers and Automata. Proceedings of Institute of Radio Engineers 41: 1234–1241.

Strang, G. 1976. Linear Algebra and Its Applications. Academic Press.

Vadim V. Anshelevich
vanshel@earthlink.net