CONNECTING SCIENCES

Jaap van den Herik
Aske Plaat
Tilburg University

Jos Vermaseren
Jan Kuipers
Nikhef, Amsterdam

Keynote Lecture ICAART 2013

Barcelona, February 15, 2013

CONTENTS

ACKNOWLEDGEMENT

We would like to thank the organisors of the ICAART 2013 for the invitation to show interdisciplinary research.

Moreover, we would like to recognize the support by Ivo van Vulpen and Stan Bentvelsen, and the Atlas research group who were amongst others instrumental in discovering the Higgs particle. They provided us with some insightful sheets.

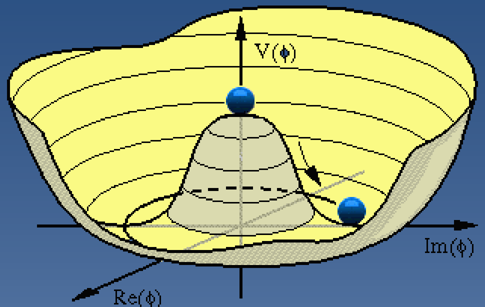
The discovery of the Higgs boson

Ivo van Vulpen (Uva/Nikhef)

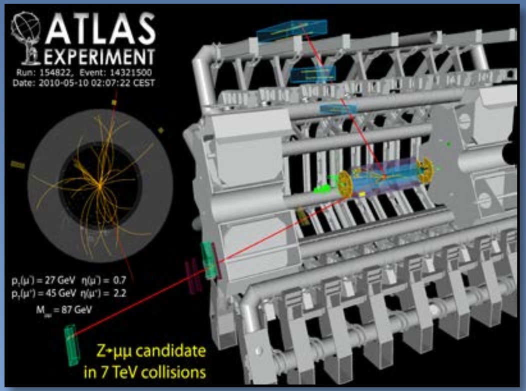
CERN in Geneva, Switzerland

Things to remember

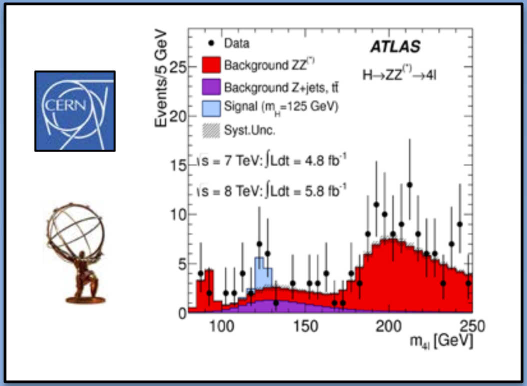
1) Higgs mechanism is at the heart of the Standard Model

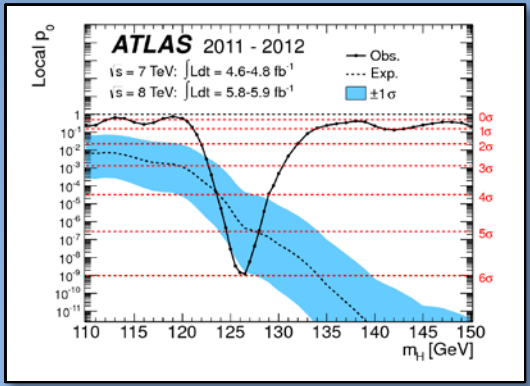


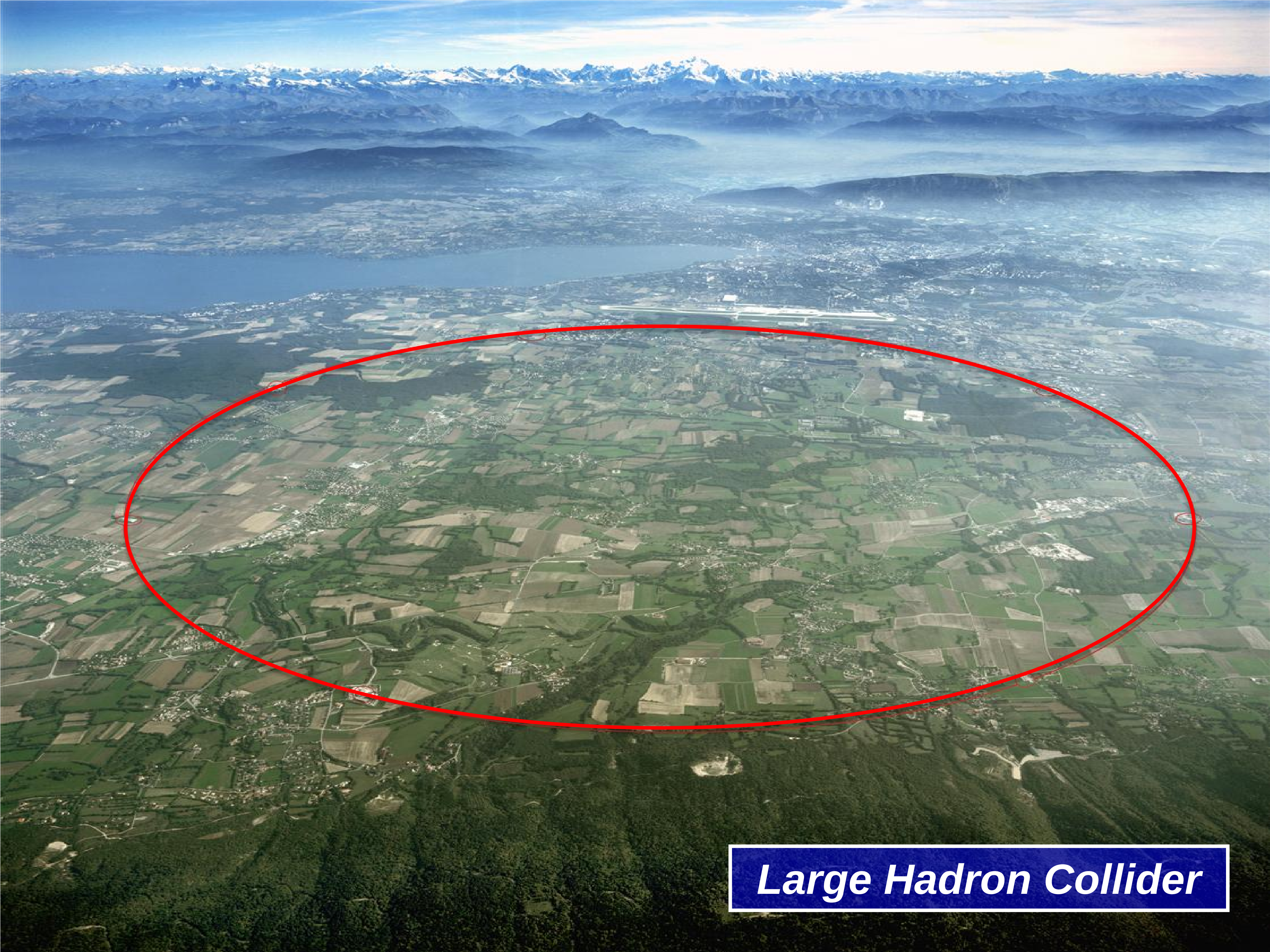
2) LHC and ATLAS detector operating fine!



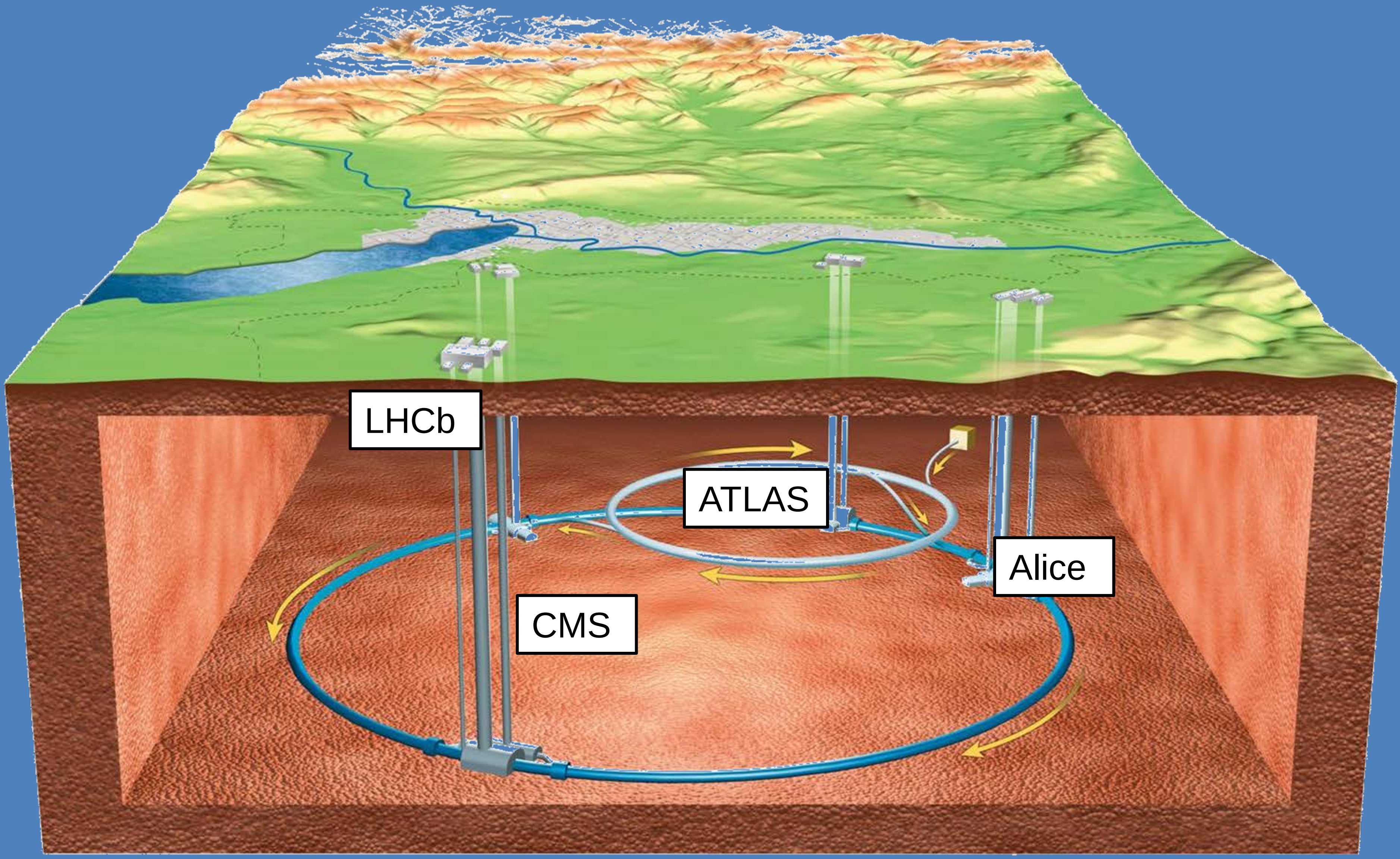
3) Discovery of the Higgs boson (interpretation)





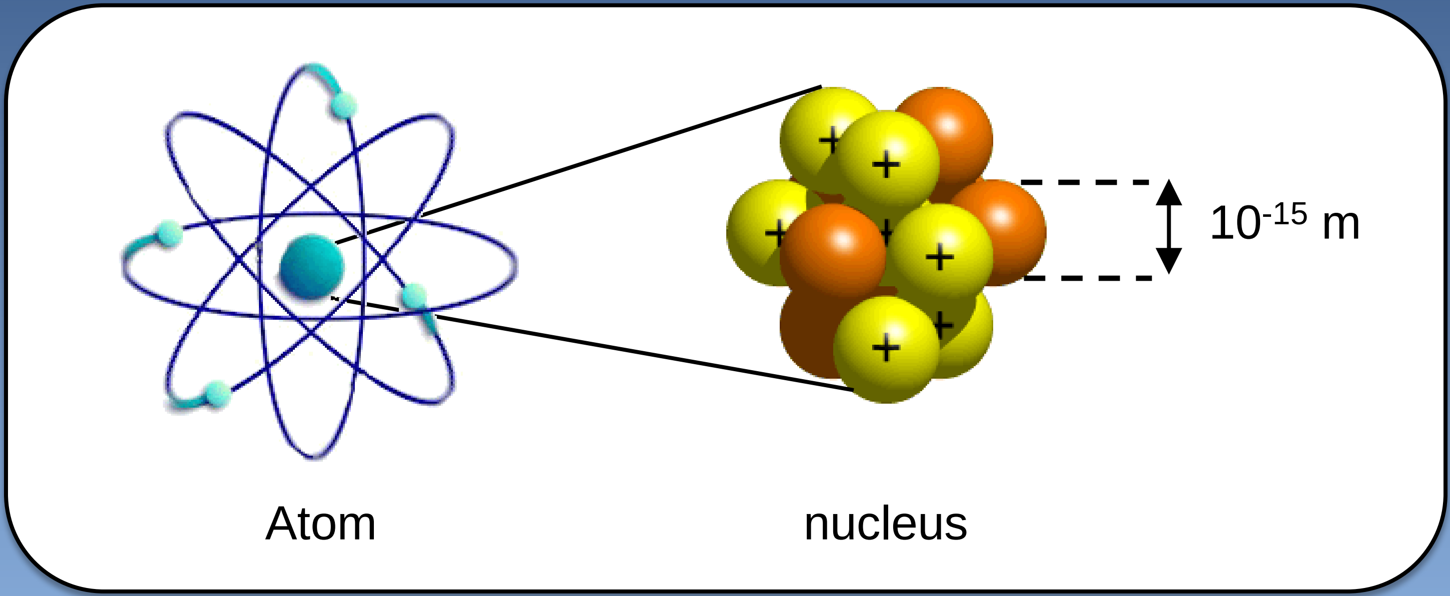

Large Hadron Collider

The Large hadron collider



Particle Physics

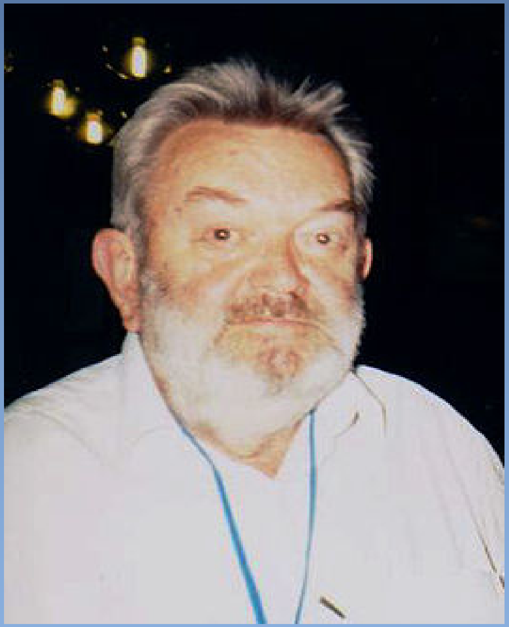
Studies nature at distance scales < 10^{-15} m



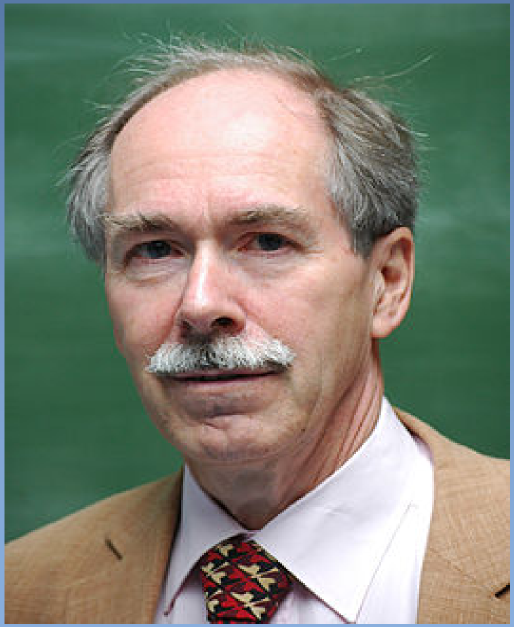
Standard Model:
Quantumfield theory that describes phenomena down to 10^{-18} m

2. Some History of Particle Physics

Schoonschip (1963) Nobelprijs voor Physics in 1999



Martin Veltman



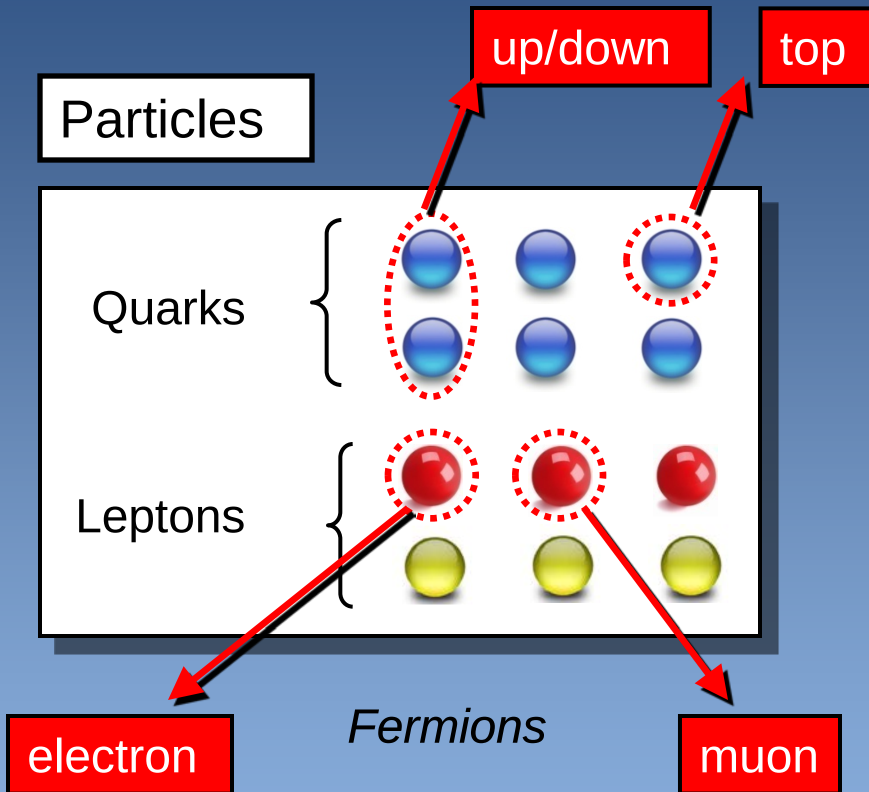
Gerard ‘t Hooft

For elucidating the quantum structure of electroweak interactions in Physics

The original building blocks of Particle Physics

- Quarks
- Leptons
- Force mediators

The Standard Model



Fermions

Forces

1) Electromagnetism (photon)

2) Weak force (W^+, W^-, Z)

3) Strong force (8 gluons)

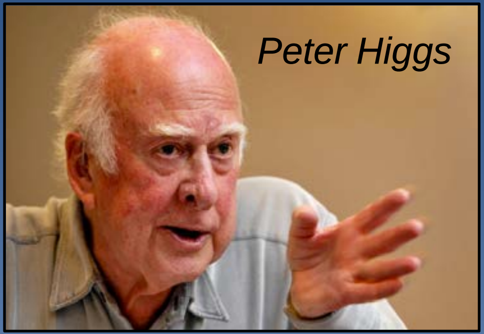
Bosons

The Standard Model

$$SU(2)_L \otimes U(1)_Y \otimes SU(3)_C$$

Weak iso-spin, hypercharge, colour

The Higgs mechanism



Massive gauge bosons in a local gauge invariant theory

$$SU(2)_L \otimes U(1)_Y \otimes SU(3)_C$$

There has to be a **Higgs boson**


BROKEN SYMMETRIES, MASSLESS PARTICLES AND GAUGE FIELDS

P. W. HIGGS

Received 27 July 1964

- September 1964 -

Breakthrough 2012

Champagne at Nikhef

The current building blocks of Particle Physics

- Quarks
- Leptons
- Force mediators
- Higgs particle

Explanation

Particle Physics

Particle physics is a part of physics that investigates the fundamental building blocks of matter and the forces between them. These building blocks are:

quarks There are 6 types of quarks: u,d,s,c,b,t.

leptons Charged leptons are the electron, the muon and the tau. The chargeless leptons are called neutrino's and they come also in three varieties.

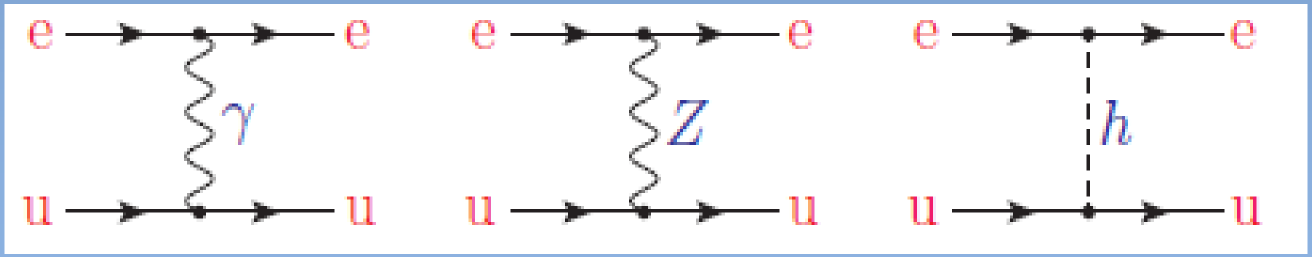
force mediators The photon (electromagnetism), the $W^{\pm}$ and Z (weak forces), the gluon (strong interactions) and the graviton (gravity).

the Higgs particle A necessary ingredient to keep the best model we have (excluding gravity) physical (ie finite).

The quarks and the leptons come in two varieties: particles and anti-particles. The force mediators and the Higgs particle are their own anti-particles. (With the W we have a W^+ and a W^- which are each others anti-particles).

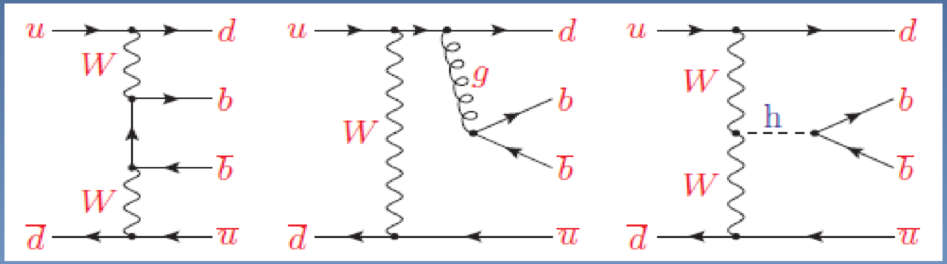
Interactions

Currently there is no good theory that incorporates all interactions (forces), but we have a very good theory that includes everything except for gravity. This is called the ”standard model”. It predicted the Higgs particle although it could not predict its mass. When particles interact with each other, we call that a reaction. These interactions can be seen as particles exchanging force mediators (or Higgs particles) as in:



Feynman Diagrams in action

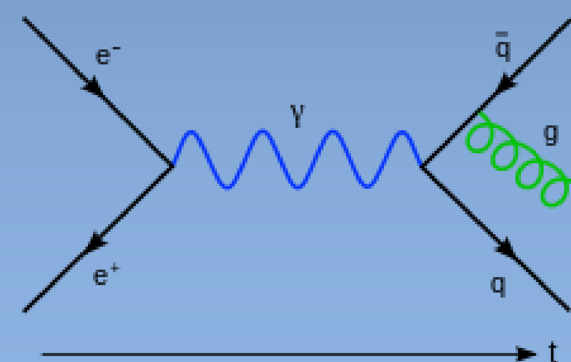
Also a particle and an anti-particle can annihilate and form one or more force mediators, or force mediators can produce a particle anti-particle pair.



The above pictures are called Feynman diagrams. In a proper theory, each element in a diagram (lines and vertices) represents an element of a formula and when you want to calculate a reaction, you have to write down all diagrams that can contribute to it, write for each diagram its complete formula, square the sum of the diagrams, and work out the formulas. This does involve quite some mathematics.

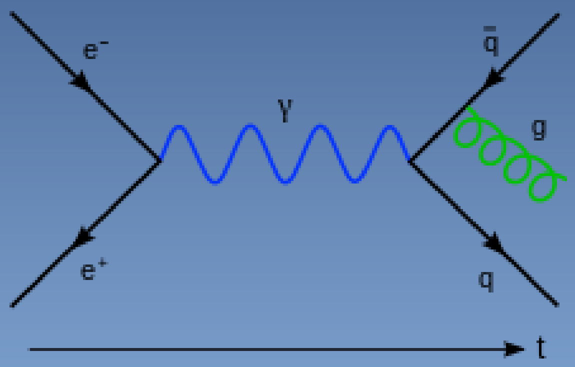
Feynman Diagrams

Graph Element	Mathematical Equivalent	Physical Interpretation
p, s	$V^{-1/2}\left(\frac{m_0}{\nu_p}\right)^{1/2}\bar{u}(\mathbf{p};s)$	N emitted
p, s	$V^{-1/2}\left(\frac{m_0}{\nu_p}\right)^{1/2}v(\mathbf{p};s)$	$\bar{N}$ emitted
p, s	$V^{-1/2}\left(\frac{m_0}{\nu_p}\right)^{1/2}u(\mathbf{p};s)$	N absorbed
p, s	$V^{-1/2}\left(\frac{m_0}{\nu_p}\right)^{1/2}\bar{v}(\mathbf{p};s)$	$\bar{N}$ absorbed
k	$V^{-1/2}\frac{1}{\sqrt{2\omega_k}}$	π emitted
k	$V^{-1/2}\frac{1}{\sqrt{2\omega_k}}$	π absorbed
p p	$\frac{i}{(2\pi)^4}S_F(p) \equiv \frac{i}{(2\pi)^4}\frac{\gamma p + m_0}{m_0^2 - p^2 - i\epsilon}$ and $\int dp \cdots$	Virtual N Virtual $\bar{N}$
k	$\frac{i}{(2\pi)^4}\Delta_F(p) \equiv \frac{i}{(2\pi)^4}\frac{1}{m_0^2 - k^2 - i\epsilon}$ and $\int dk \cdots$	Virtual π
q^3, q^1, q^2	$-g_0\gamma_5(2\pi)^4\,\delta(q^1 - q^2 - q^3)$	Interaction
	$-\mathrm{Tr}$	—



Feynman formulas

- Path integral formulation:



$$\psi(y;t+\epsilon)=\int_{-\infty}^{\infty}dx\quad\psi(x;t)\int_{x(t)=x}^{x(t+\epsilon)=y}e^{i\int_t^{t+\epsilon}(\dot{x}^2-V(x))dt}\,Dx(t)\qquad(1)$$

Big Formulas

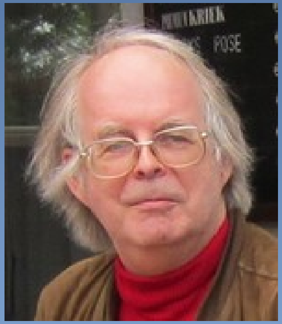
These formulas can become rather big in two ways:

- It can happen that one starts with one (complicated) diagram, both the input and the output fit on a few lines, but at intermediate stages one could have many Gbytes of formula.

- One may end up with a formula that takes millions of terms and this formula needs to be integrated over by numerical means. In both cases it should be clear that this is way beyond manual processing.

FORM

- FORM, an Open source formula manipulation system designed for handling large equations
- history: SCHOONSCHIP 1963
- Martinus Veltman, Gerard ‘t Hooft
- Form is written by Jos Vermaseren



- 2006 Von Humboldt research award: outstanding long term contributions to precision calculations in Quantum Chromodynamics (QCD), notably on the scaling violations of the nucleon structure functions. The calculations allow to determine the strong coupling constant αs at higher precision from the HERA data. These results could only be achieved applying effective computer algebra systems such as FORM, developed by him, which finds a widespread use in present day high energy physics equations

Equation solving

- Computer Algebra Systems have been around since the dawn of computing
- They evolved from two fields: the requirements of theoretical physicists, and research into artificial intelligence
- Mathematica, Maple, Matlab target ease of use and graphing functions
- FORM targets speed, the capability to solve large formulas, and programmability

Solving an equation

- 200 (x - 33) + 3233 = 566
- 200 x - 6600 + 3233 = 566 (distributive prop)
- 200 x - 3367 = 566 (combine like terms)
- 200 x = 3933 (add 3367 to both sides)
- x = 19.665

(a+b) ^5

```
    Symbols a,b;
    Local F = (a+b)^5;
    Print;
    .end

Time =        0.00 sec    Generated terms =          6
                F         Terms in output =          6
                          Bytes used      =        204

   F =
      b^5 + 5*a*b^4 + 10*a^2*b^3 + 10*a^3*b^2 + 5*a^4*b + a^5;
```

(a+b+c+d+e+f+g) ^30

```
    Symbols a,b,c,d,e,f,g;
    Local F = (a+b+c+d+e+f+g)^30;
    .end

Time =          6.51 sec    Generated terms =     1947792
                  F         Terms in output =     1947792
                            Bytes used      =    99425612
```

F (21)

```
Symbols n;
CFunction f;
Local Fibonacci = f(21);
Repeat;
  id f(n?{>2}) = f(n-1)+f(n-2);
EndRepeat;
id f(1) = 1;
id f(2) = 1;
Print;
.end

Time =        0.05 sec    Generated terms =      10946
       Fibonacci          Terms in output =          1
                          Bytes used      =         20

   Fibonacci =
      10946;
```

Champions league

The next example is more cryptic. It computes all possible drawings for the eighth finals of the champions league in December 2012. It vetoes teams of the same nationality playing against each other. There were two groups of eight teams and teams from the first group had to play against teams of the second group. Each group had two Spanish, one French and one Italian team. Other teams could not cause conflicts.

```
    Tensor f;
    Index  i1,...,i8;
    Local  F = f(i1,...,i8)*e_(i1,...,i8)*e_(1,...,8);
    Contract;                        * Generates the 8! permutations
    id f(i1?{1,2},?a) = 0;           * First Spanish team in group 2
    id f(i1?,i2?{1,2},?a) = 0;       * Other Spanish team in group 2
    id f(i1?,i2?,3,?a) = 0;          * Italian teams
    id f(i1?,i2?,i3?,4,?a) = 0;      * French teams
    .end

Time =       0.05 sec    Generated terms =      17088
               F         Terms in output =      17088
                         Bytes used      =     527676
```

Chance on same result

The program first generates all $8! = 40320$ possibilities and then eliminates the 'forbidden' combinations. There are 17088 possibilities left. Hence the chance that the trial run and the real drawing would give the same result is $1/17088$. Note how short such a program can be if you know what you are doing.

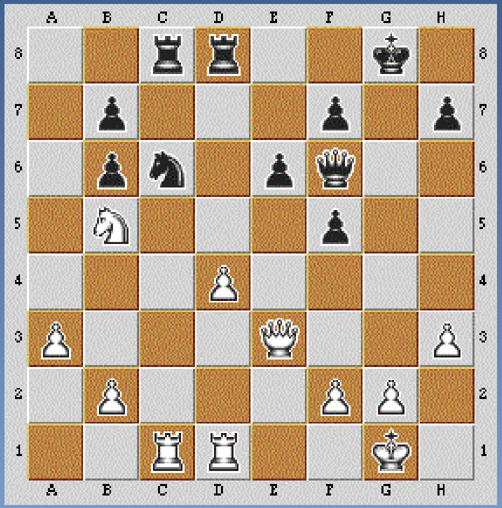
Solving Larger Formulas

- FORM needs to solve larger formulas
- Guiding the solving process by hand to where the fruitful areas are, using knowledge of the theoretical physicist
- We need a "meta-solver" to search through the possible solving strategies

Minimax

- [John Von Neumann, Zur Theorie der Gesellschaftsspiele 1928]
- Adversarial games
- your win is my loss
- I maximize my outcome, you minimize it
- For example: Chess

Games, Minimax, MCTS
Chess

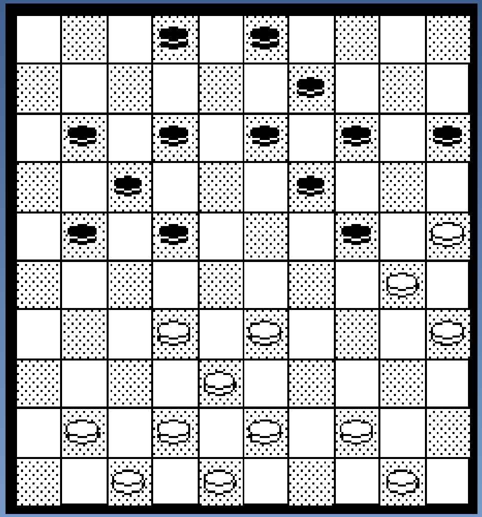


- Much research has been performed in computer chess
- DEEP BLUE (IBM) defeated the world champion Kasparov in 1997
- FRITZ defeated Kramnik (December 2006)
- Techniques Minimax enhancements

Solving Checkers

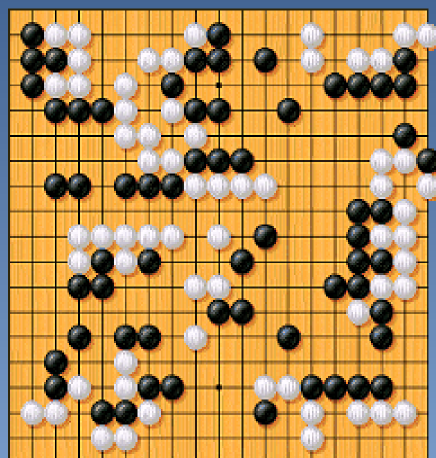
- Schaeffer, Björnsson, Burch, Kishimoto, Müller, Lake, Lu, and Sutphen
- Spring 2007
- Checkers is Solved
  Science, Vol. 317, No. 5844, pp. 1518-1522

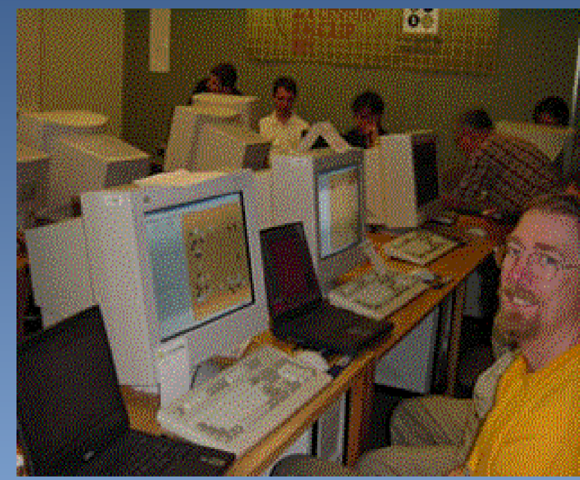
International Draughts



- MAXIMUS and KINGSROW INTERNATIONAL are the best draughts programs
- Human better than computer, but the margin is small
- Challenge: More knowledge in program
- MAXIMUS (Jan-Jaap van Horssen) vs. Alexander Schwarzman

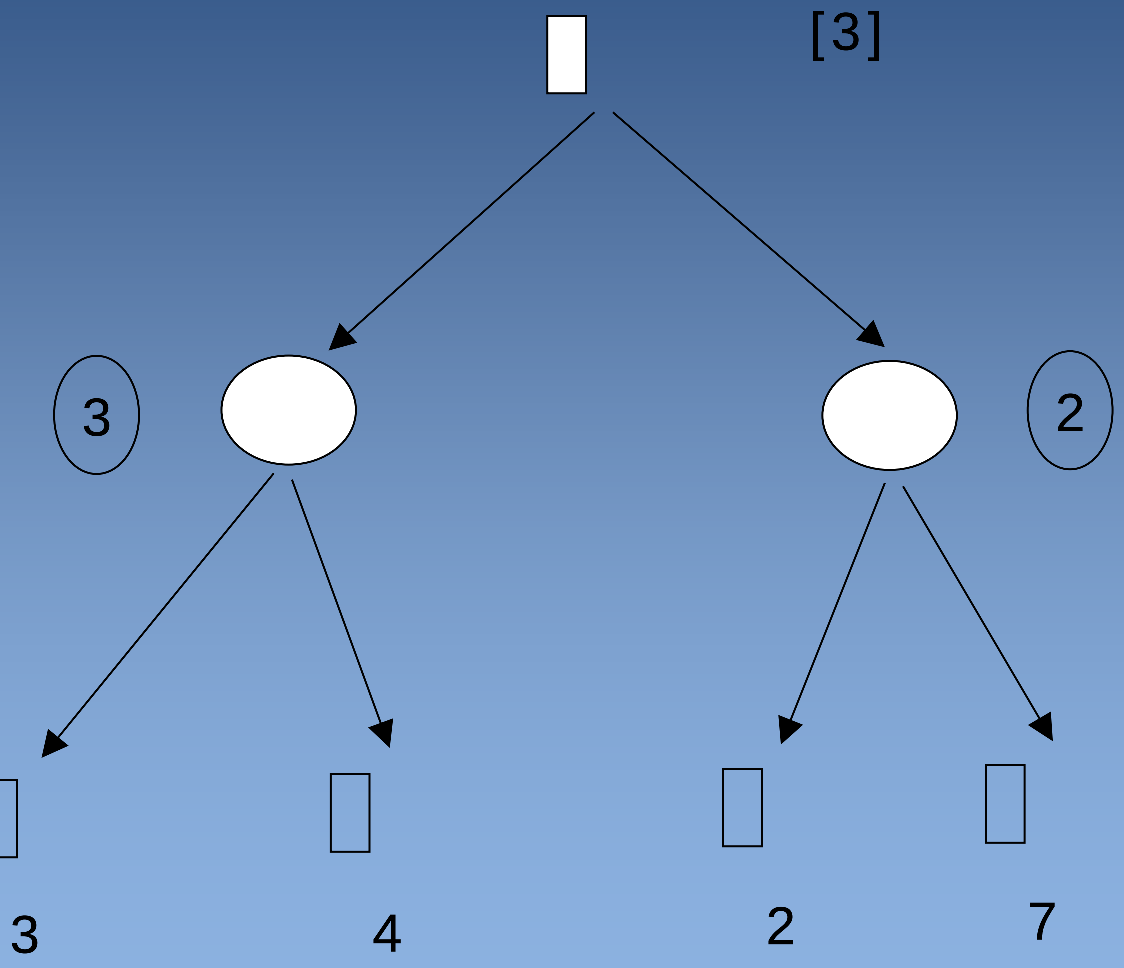
Go



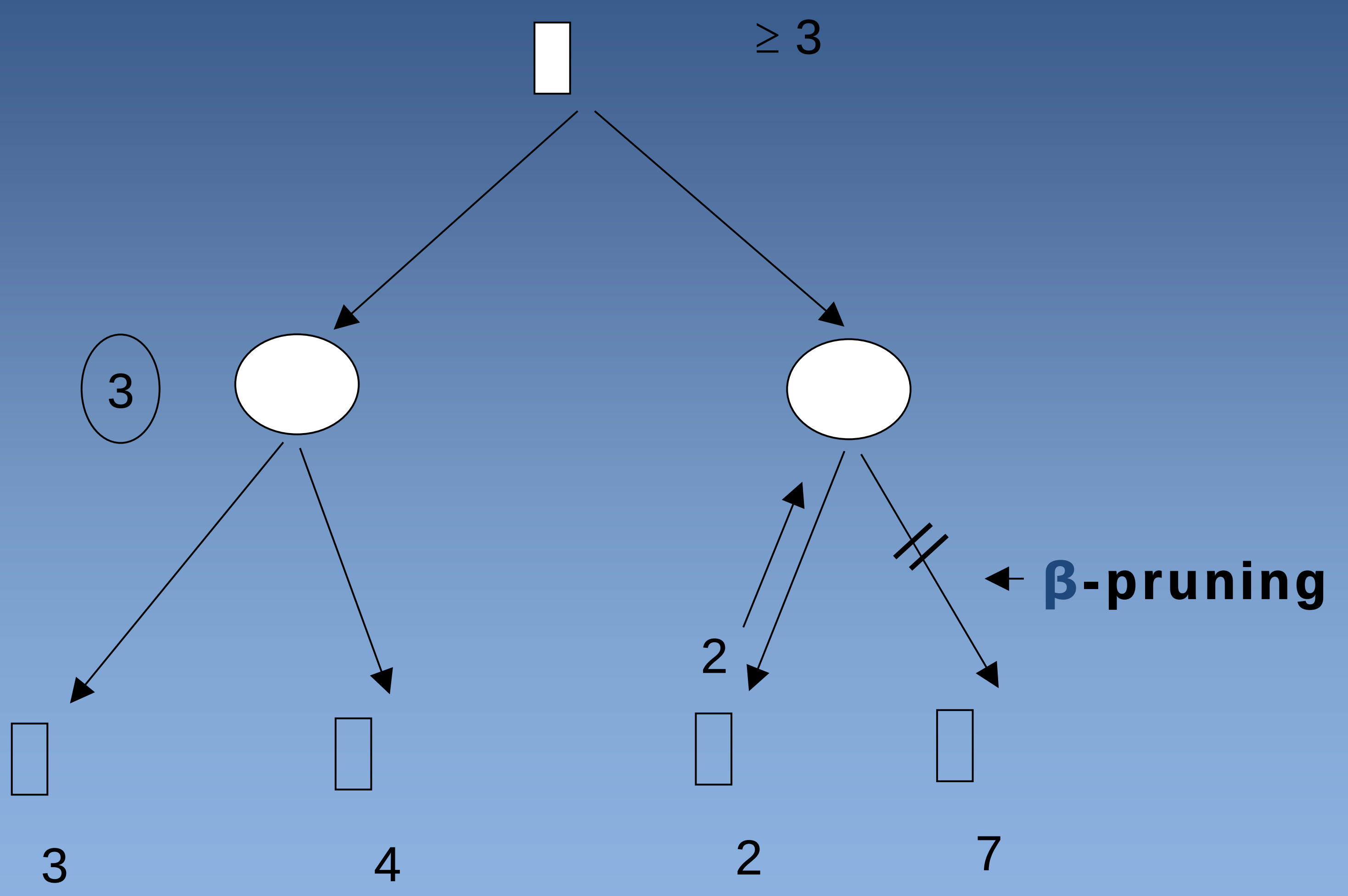


- Computer Go programs have an advanced level
- Top Go programs: ZEN, FUEGO, MOGO, PACHI, ERICA
- Problem: recognition of patterns
- Solution: MCTS

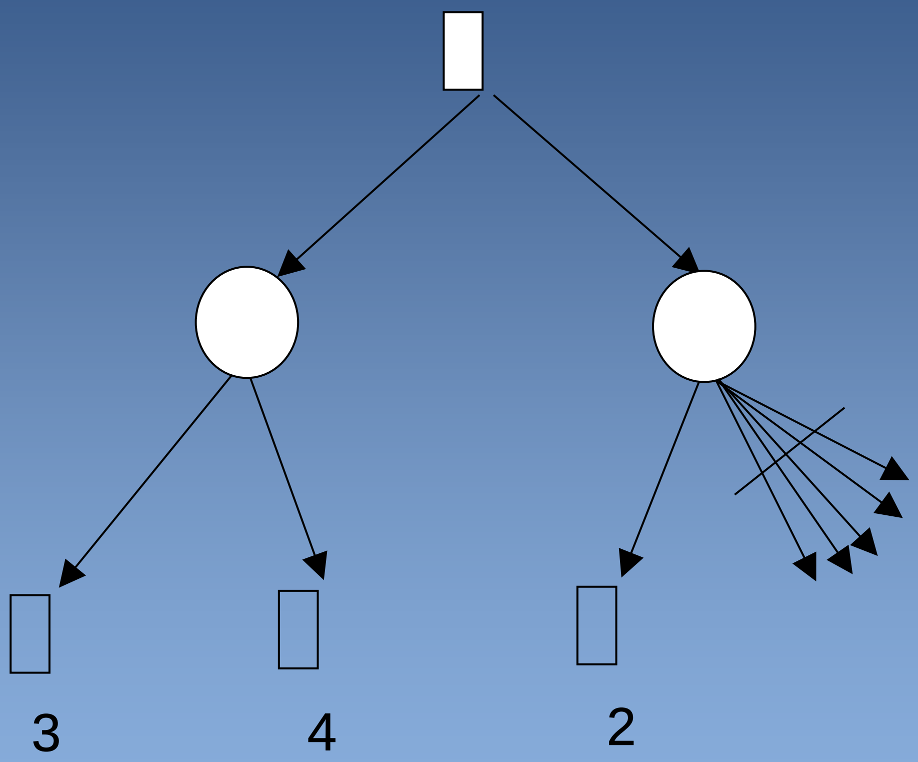
Minimax



α-β Algorithm

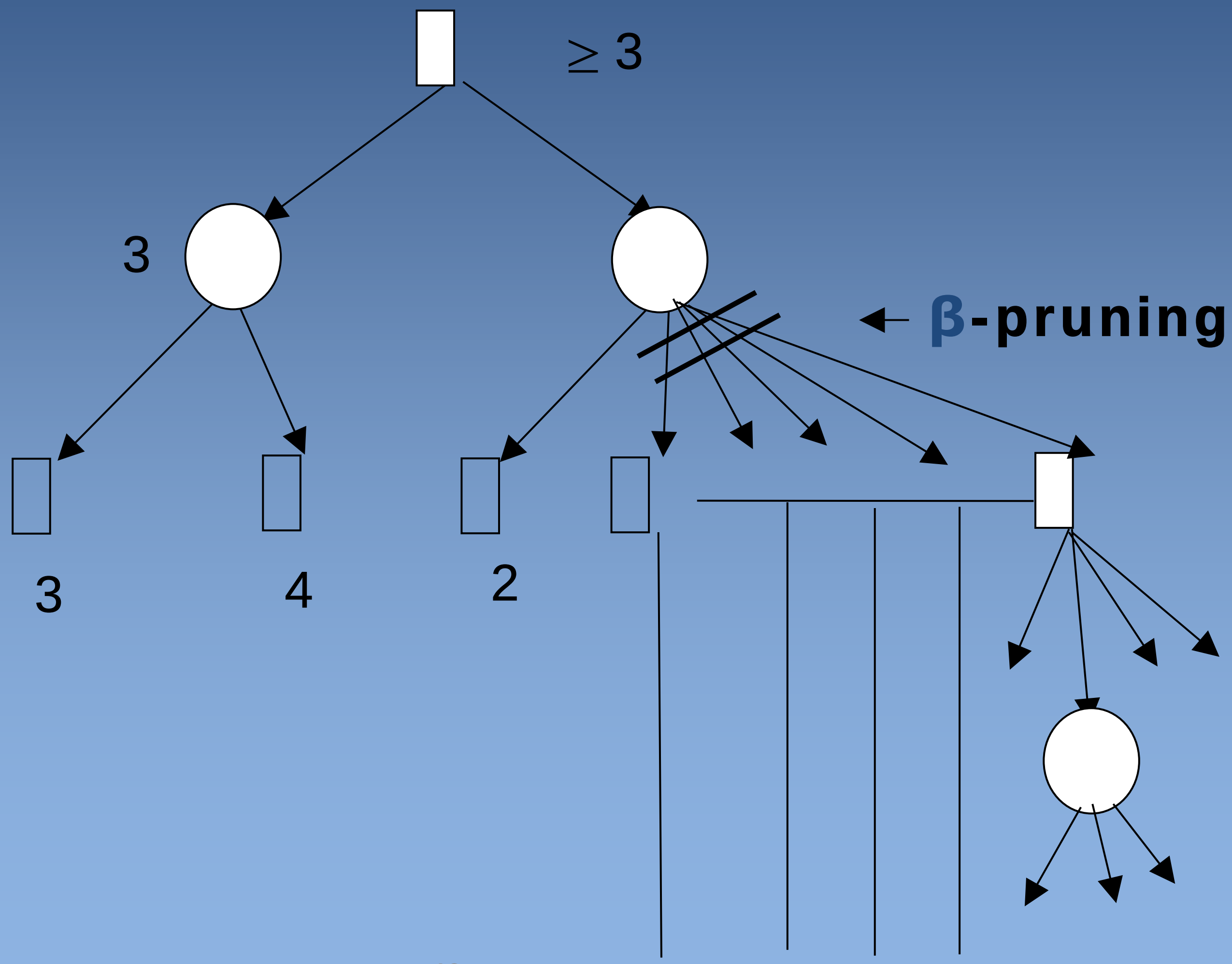


The Strength of α-β



More than thousand prunings

The Importance of α-β Algorithm



The Possibilities of Chess

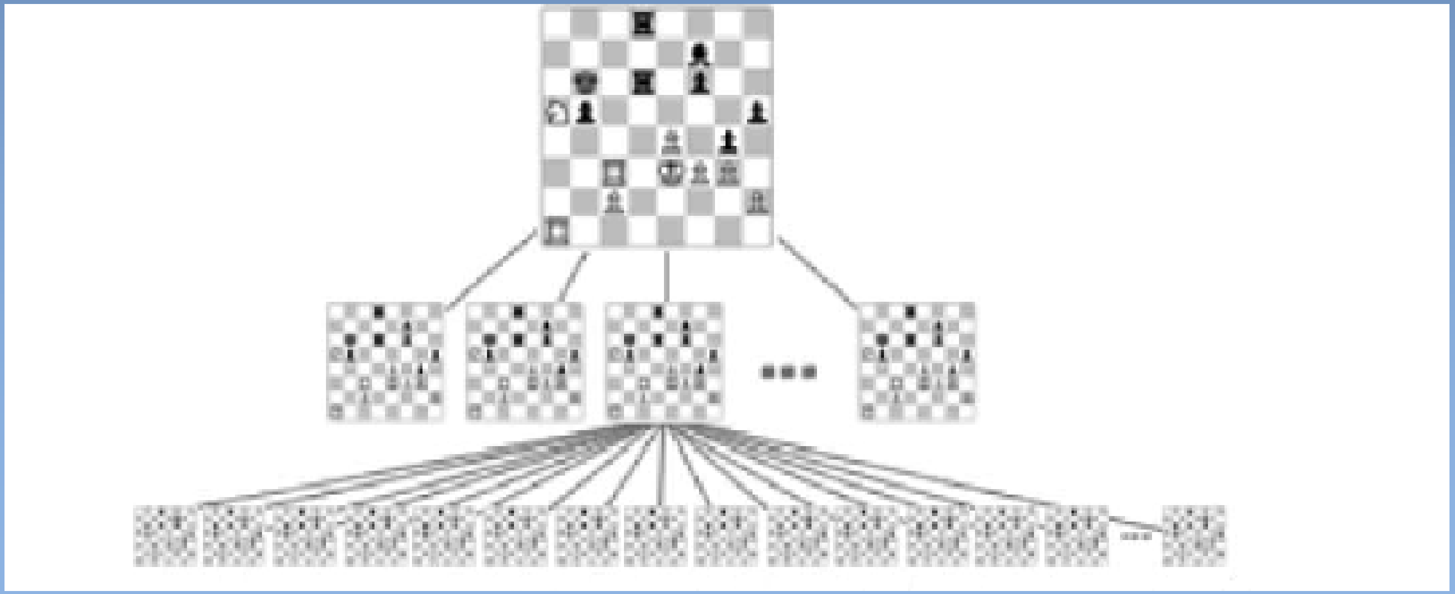
THE NUMBER OF DIFFERENT, REACHABLE

POSITIONS IN CHESS IS

(CHINCHALKAR): 10^{46}

Chess minimax tree

- Size: $O(10^{46})$
- 1. Search
- 2. (since tree is too large)
  Heuristic Evaluation



Minimax Improvements

- Alpha Beta
  [McCarthy 1956] [Brudno 1963]
  - complexity $O(w^d)$ -> $O(sqrt(w^d))$
- Proof Number Search
  [Allis, Van der Meulen, Van den Herik, 1994]
  - most efficient for solving games
- MTD(f)
  [Plaat, Schaeffer, Pijls & De Bruin 1994]
  - Pure null-window search, 1994 ICGA award, current most efficient minimax algorithm

A Clever Algorithm (α-β)

Saves the square root of the number of possibilities, √n, this is more than

99.999999999999999999999%

[1% of $10^{46} = 10^{44}$

$\sqrt{10^{46}} = 10^{23}$

44 – 23 = 21 (9's behind the decimal point)]

A Calculation (1^{st} set)

NUMBER OF POSSIBILITIES:	10^{46}
SAVINGS BY α-B ALGORITHM:	10^{23}
1000 PARALLEL PROCESSORS:	10^{3}
POSITIONS PER SECOND:	10^{9}
LEADS TO: 10^{23-12} =	10^{11} SECONDS
A CENTURY IS	10^{9} SECONDS
SOLVING CHESS:	10^{2} CENTURIES

SO 100 CENTURIES OR 10,000 YEAR

WE RETURN TO THIS NUMBER.

Moore's Law

The computer capacity

is doubled

every 18 months

A New Calculation (2^{nd} set)

NUMBER OF POSSIBILITIES:	10^{46}
SAVINGS BY α-B ALGORITHM:	10^{23}
1000 PARALLEL PROCESSORS:	10^{3}
POSITIONS PER SECOND:	10^{14} (9+6=15; 15-1=14)
LEADS TO: 10^{23-17} =	10^{6} SECONDS
A CENTURY IS	10^{9} SECONDS
SOLVING CHESS:	10^{-3} CENTURIES

So roughly 37 days in 2035.

This is for Chess.

Quantum Computer

Leo Kouwenhoven (Delft Univ. of Technology)

Carlo Beenhakker (Leiden University)

Received 17 M euro for building

a Quantum Computer

The computer capacity is estimated 10^24

of the current computer.

Chess will be solved in less than one day.

Contributions from Science

- Computers play stronger than humans.
- Computers can solve chess.
- Computers enable an alternative form of game experience.

Provisional Conclusions on Chess

1. Checkers is a frontrunner among the games
2. Chess is a direct follower
3. Kasparov's defeat has become a victory for brute force in combination with knowledge and opponent modelling

Go: new Drosophila Melanogaster

The Difference between Chess and Go

- Chess: Search

  Tactics play an important role

- Go: Pattern Recognition

  Strategy is much more important

MCTS

- The Problem with Go:
  No good evaluation function
- simulation
  - 2006 MCTS: use average of simulated playouts as evaluation function
    [Chaslot, Saito, Bouzy, Uiterwijk, Van den Herik, 2006]
- Search Balancing
  - Exploration/Exploitation balancing
    Sampling actions selectively: MCTS uses UCT (Upper Confidence Bounds applied to Trees)
    [Kocsis, Szepesvári, 2006]

Two breakthroughs that enabled Go to play at "acceptable level"

1. Monte Carlo Search
(Brügmann and Bouzy)

2. UCT – algorithm (Kocsis, Szepesvari)
(Chaslot, Coulom),

UCT stands for
Upper Confidence bounds applied to Trees

UCT Formula

Assume that we compensate for the lack of an evaluation function by playing the game out randomly and then counting the score.

This sounds completely crazy!

But we add something to it: We will try this many times (this is the essence of a Monte Carlo method) and we will let the part of the tree that we will try depend on the results of previous attempts.

The selection of the next attempt will be according to the UCT (Upper Confidence level for Trees), introduced by Kocsis and Szepesvári in 2006:

$$UCT_i = \langle x_i \rangle + 2C_p \sqrt{\frac{2 \log n}{n_i}}$$

At any point in the tree the child with the highest UCT value is selected. Here

$\langle x_i \rangle$ is the average score of child i over the previous traversals

n_i is the number of times child i has been visited before

n is the number of times the node itself has been visited

C_p is a problem-dependent constant. Should be determined empirically.

The value of C_p

$$UCT_i = \langle x_i \rangle + 2C_p\sqrt{\frac{2\log n}{n_i}}$$

The first term in the equation favours trying previously successful branches in the tree. This is called exploitation. The second term favours branches that have not been visited much before (if never, the term is even infinite). This is called exploration. The value of C_p determines the balance between the two.

This approach can be successful if positive outcomes are clustered in the tree.

In games this often works because a good move will usually leave many more favourable endpositions than a bad move.

When the value of C_p is too small, we will only sample one seemingly good branch in the tree and eventually end up in a local maximum.

When the value of C_p is too big, we will basically be sampling randomly and forget to pursue branches that seem promising.

Let us have a look at this in an example based on what we will discuss in the next section. The aim is to have as small an outcome as possible.

Exploration-exploitation

-

⇒Exploration-exploitation dilemma:

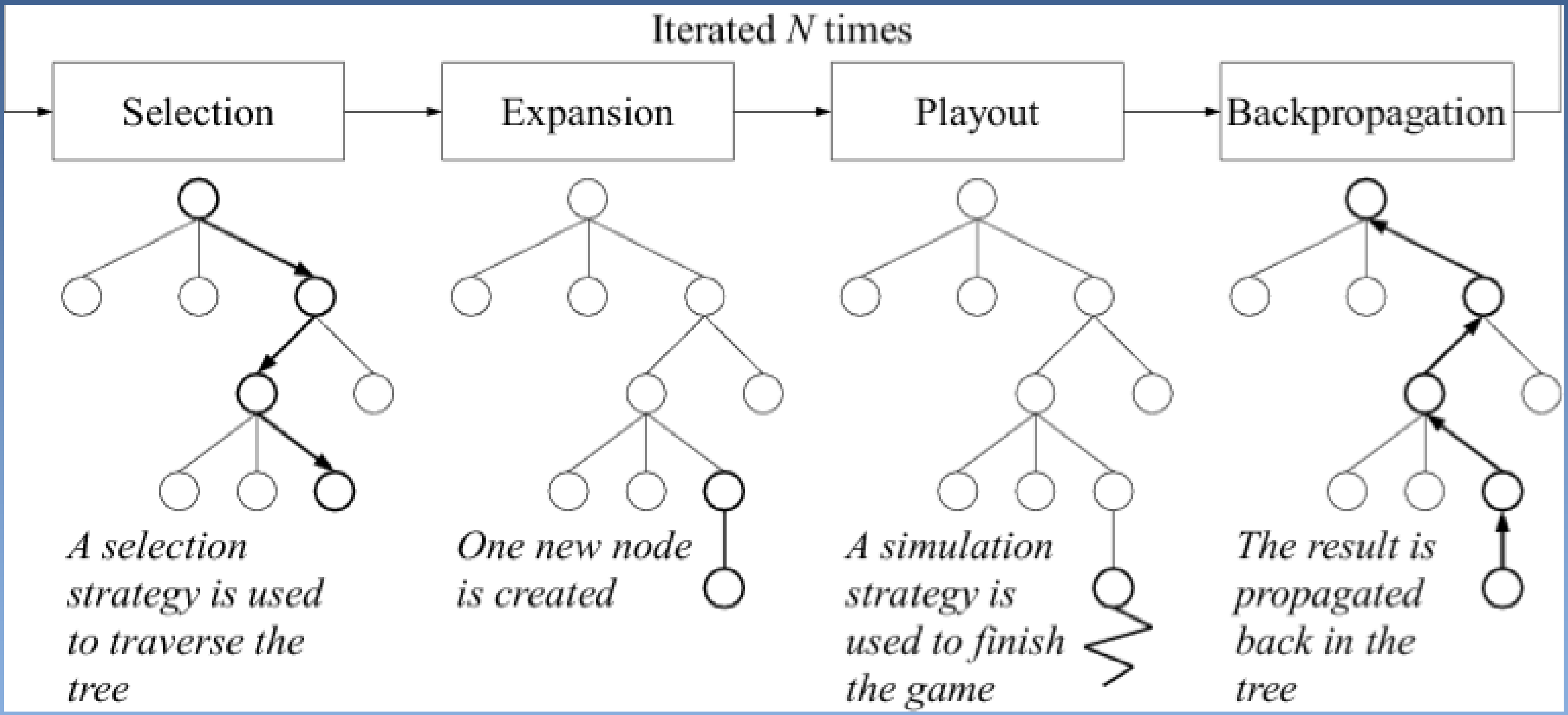
If only the best moves are explored (too few explorations), the algorithm is focusing on a few moves, and moves that did not seem promising are forgotten.

-

If too many moves are explored, the branching factor is too high and the search is not deep enough

Alternative solutions have to be found (Progressive strategies, RAVE, etc…)

MCTS tree



Iterated N times
Selection
Expansion
Playout
Backpropagation
A selection strategy is used to traverse the tree
One new node is created
A simulation strategy is used to finish the game
The result is propagated back in the tree

Development of MoGo

- Started in 2006 by Sylvain Gelly and Yizao Wang at University of Paris-Sud
- August 2006: Takes the highest rank program on the 9x9 Computer Go Server. It still holds this rank for 2 years long.
- June 2007: wins the 12th Computer Olympiads in Amsterdam, and first program ever to defeat a professional on 9x9 in a blitz game.
- April 2008: wins the first non-blitz game against a professional.
- May 2008: involvement of the project GoForGo leading to MoGo-Titan.
- August 2008: wins the first match ever against a professional on 19x19 with 9 stones handicap (running on Huygens). This result is acknowledged as a milestone for AI.

Development other programs

Sept. 2008: CRAZY STONE wins 8-stone and later 7-stone handicap (19x19)

May 2009: Pamplona: 1. ZEN 2. FUEGO 3. MOGO (19x19)

Aug. 2009: MOGO wins 7-stone handicap (against 9P) (19x19)
wins 6-stone handicap against 1P (19x19)

Oct. 2009: MOGO TW wins first 9x9 game against top professional

June 2011: ZEN defeats 4 professionals on 9x9 (one 9P)
ZEN wins two 6-stone handicap against 9P (19x19)

Nov. 2012: ZEN wins 4-stone handicap against 5P (19x19)

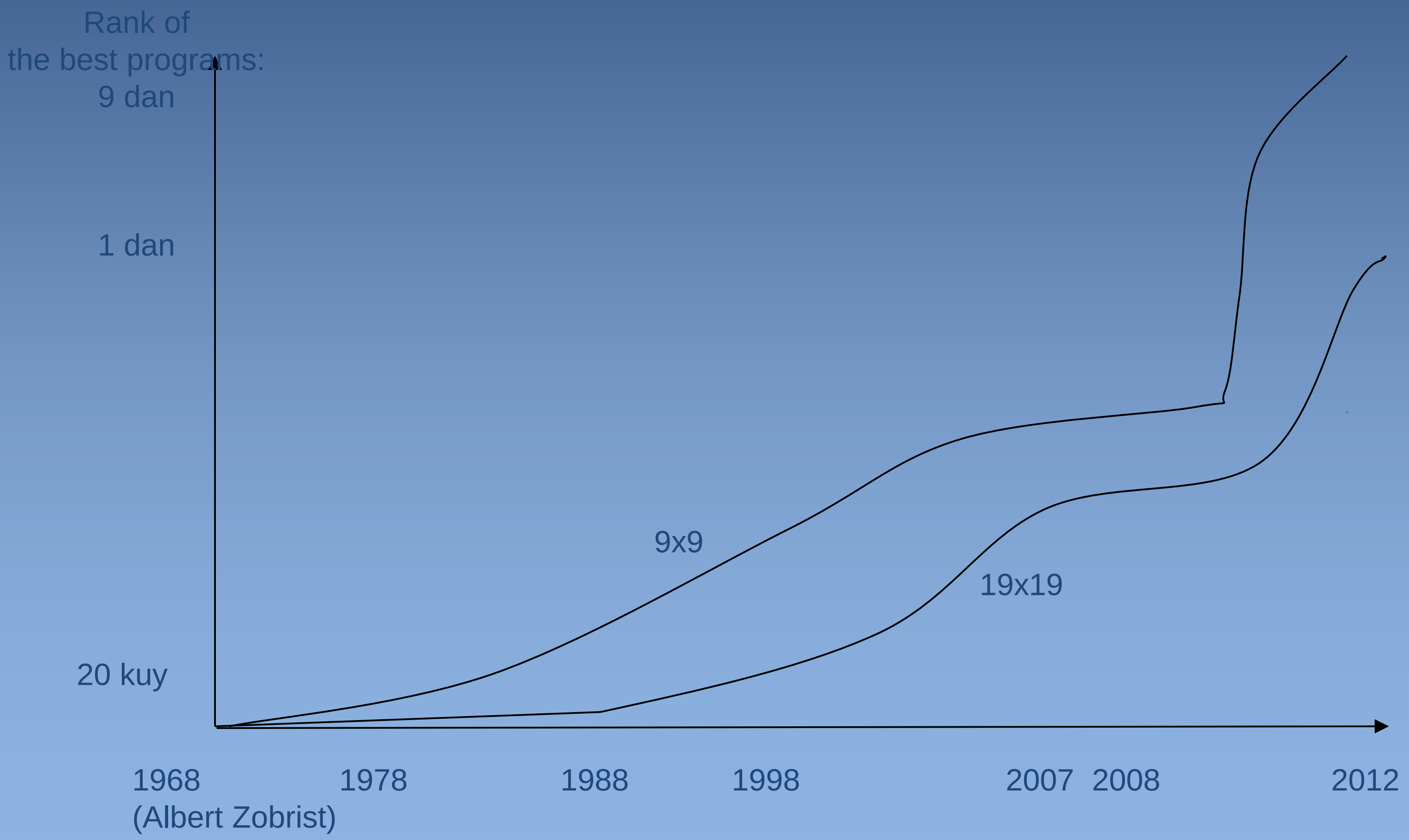
Human-Computer Matches in Go

- For a long time, a prize of 40,000,000 NTD (**1,400,000 $**) for the first computer Go-playing program that would succeed in beating a Taipei Go Professional without handicap. The prize was donated by Ing Chang-Ki and was valid until 2000, due to the death of Ing Chang-Ki.

- 400,000 NTD (14,000 $) were offered to a program that would beat a professional at 9 stones. Numerous attempts were made but no program ever won.

- More information on the numerous attempts are listed here: http://senseis.xmp.net/?IngPrize

Evolution of the level of programs



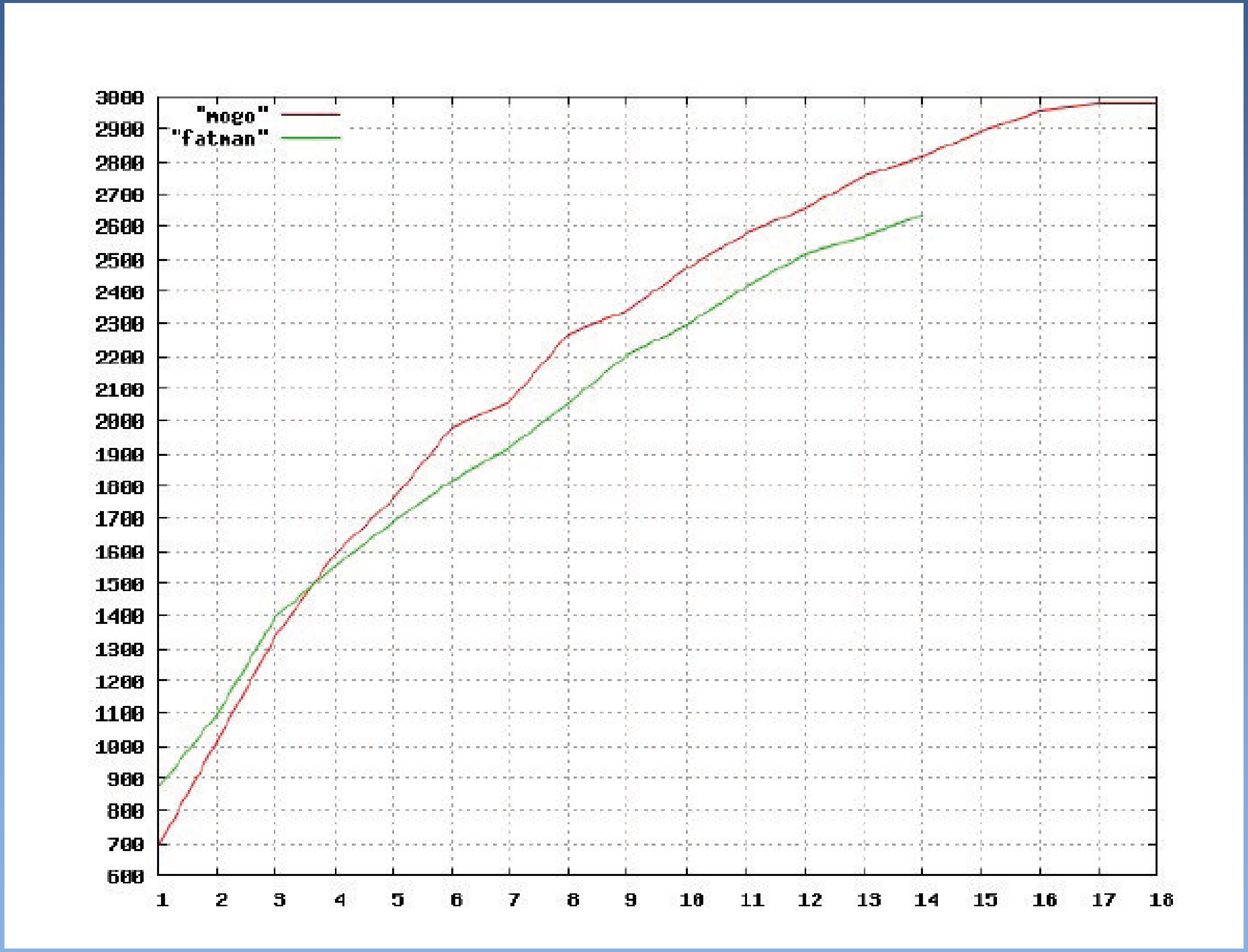
Parallelisation

One of the nice things about MCTS is that is can make use of many processors simultaneously. The major problem is the updating of the tree information. There are of course also other problems to be considered, but that would take us too far. The scalability of such Monte Carlo algorithms was investigated in 2008 by Don Dailey. He tested two programs on a 9x9 board (and had them play many games against GnuGo of which the strength was known). The number of tree evaluations was:

Mogo	$64 \times 2^{(N-1)}$ simulations
FatMan	$1024 \times 2^{(N-1)}$ simulations

Both programs used MCTS. Mogo was at the time the strongest program which can be seen from the fact that it needed fewer tree evaluations for comparable strength. The result is in the figure.

A Comparison



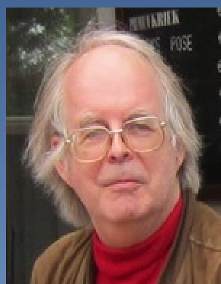
CONCLUSIONS ON GAMES

1. Computers will solve a range of games.
2. New games will emerge.
3. Humans will continuously learn from computers.
4. The Games Research will envisage new games and even more new computer techniques.
5. Game techniques will enter the world of particle physics

Back to HEP

- Could this work for guiding the search for solving HEP formulas
- Replacing the tedious manual input to guide the strategy to solving the huge formulas?

HEPGAME
ERC Advanced Grant



About the ERC Advanced Grants:
The ERC Advanced Grant is given to exceptional individual researchers to pursue cutting-edge ground-breaking projects that open new directions in their respective research fields or other domains. Every year a few thousand applications are received by the European Research Council, of which only a few hundred are honoured.

About the proposal:
The calculations proposed have been intractable thus far due to their enormous demand of man and computerpower. The team will make use of the Monte Carlo Tree Search technique from the fields of Artificial Intelligence and gaming to resolve this issue and to automatise the derivation of formulas and the construction of computer programs. This will be done in the framework of the (open source) computer algebra system FORM developed by Jos Vermaseren. The new technology will be made available for other researchers, enabling a wide range of calculations at a new level of precision.

Challenge: improving Horner

- After we submitted the proposal to ERC in February 2012
- We decided to play around with MCTS and FORM

Main Programmer

Jan Kuipers

Horner's method

- One of the things FORM needed was a solver for multivariate polynomials
- The basic approach is to apply William Horner's rule for single variable polynomials [1819]*, and extend it to most-occurring variable first

  *) generally assumed to be due to Liu hui, a third century Chinese Mathematician
- We* decided to see if MCTS could find better evaluation orders

  *) Jan Kuipers, Jos Vermaseren, Aske Plaat, Jaap van den Herik

Polynomial Evaluation

- $x^2 + 2xy + y^2 = 0$
  - Has 2 "+" operations and 4 "*" operations
- $(x + y)^2 = 0$
  - has 1 "+" operation and 1 "*" operation

Evaluation Order

- Original
  - $a = y - 6x + 8xz + 2x^2yz - 6x^2y^2z + 8x^2y^2z^2$
    - 5 +, 18 *
- Horner evaluation order: $x < y < z$
  - $a = y + x(-6 + 8z + x(y(2z + y(z(-6 + 8z)))))$
    - 5 +, 8 *
- Common Subexpression Elimination
  - $T = -6 + 8z$
    $a = y + x(T + x(y(2z + y(zT))))$
    - 4 +, 7 *

Open problem

- Finding the optimal order of variables for the Horner scheme is an open problem for all but the smallest polynomials
- With appropriately chosen search parameters, MCTS finds better variable orders

Results

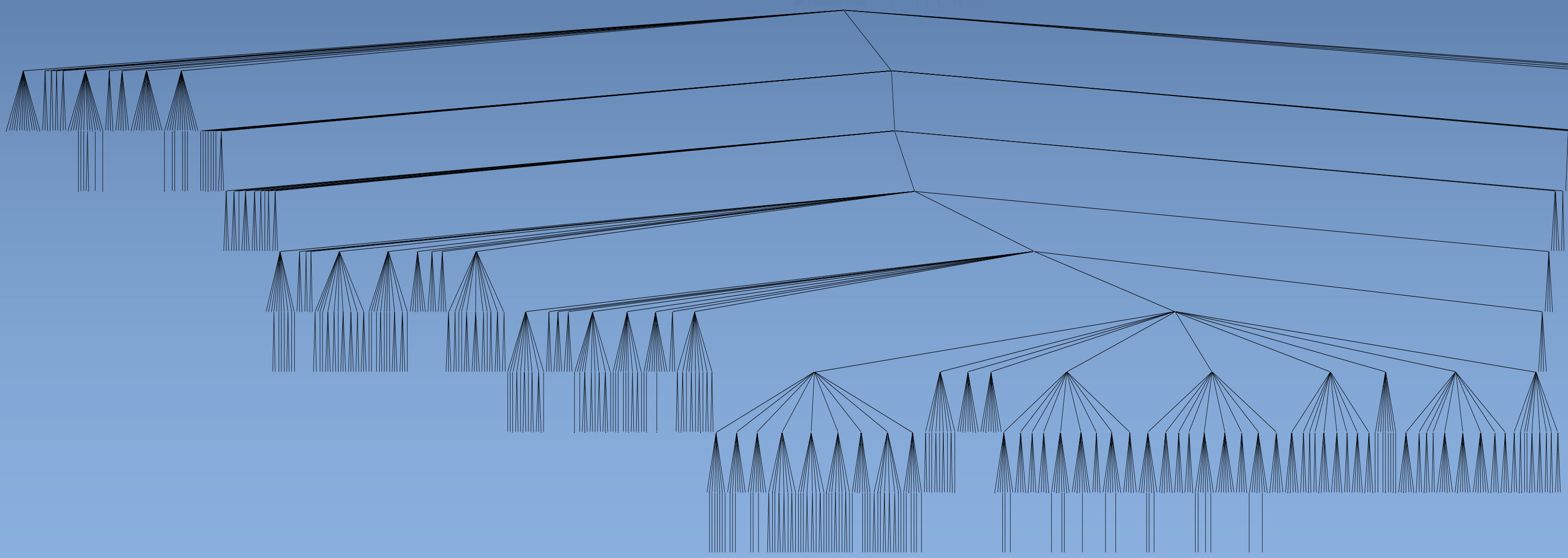
- As an example, For one of our HEP polynomials, HEP(sigma)
- No optimization:
  47424 operations (+ and *)
- Horner Occurrence order + CSE:
  6744 operations
- MCTS:
  3401 operations
  (for this polynomial a 98% improvement over Horner)

MCTS Horner

- The root of the search tree represents that no variables are chosen yet
- This root node has n children, each representing a choice for variables in the trailing part of the order
- A node has n children: the remaining unchosen variables
- In the simulation step the incomplete order is completed with the remaining variables added randomly
- This complete order is then used for Horner's method followed by CSE. The number of operators in this optimized expression is counted.
- The selection step uses the UCT criterion with as score the number of operators in the original expression divided by the number of operators in the optimized one. This number increases with better orders.

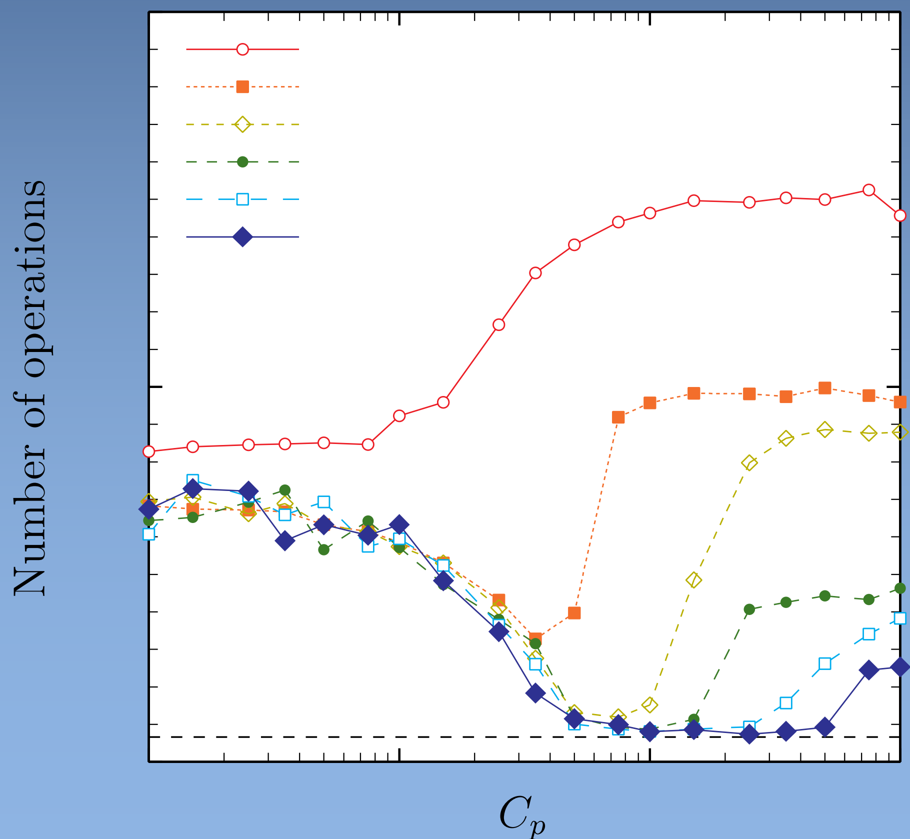
MCTS Horner Tree

- In MCTS the search tree is built in an incremental and asymmetric way



Search parameters

- N, the Number of tree expansions
- C_p, the Exploration/Exploitation parameter



Conclusions

- Without any domain knowledge, MCTS can find significantly better variable orderings for polynomial evaluation

- MCTS holds promise for improving FORM's ability to solve larger equations

Future Work

- Explore MCTS search parameters in the domain of evaluation of polynomials
- Explore MCTS search parameters in the domain of equation solving
- Explore sensitivity of MCTS to different polynomials
- Explore application areas in FORM