

# „Poker“

Vortrag von Daniel Schreiber

Seminar „Knowledge Engineering und Lernen in Spielen“,  
SS04, Prof. Fürnkranz

„How long does it take to learn poker, Dad?“

„All your life, son.“

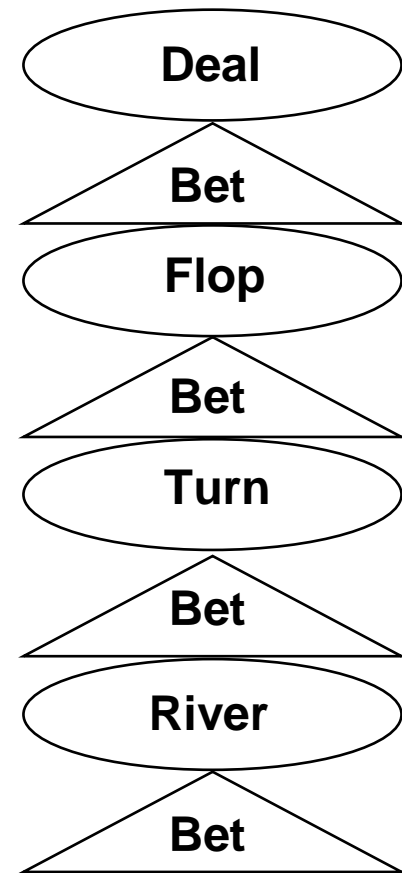
David Spanier, "Total Poker" (1977)

# Übersicht

- Texas Hold'em Regeln
- Spielbaum
- Pokerprogramme
  - Selby
  - Poki
  - PsOpti

# Regeln Texas Hold'em

- Zwei Karten verdeckt („Hole Cards“)
- 1. Wettrunde
- Drei Karten offen in die Mitte („Flop“)
- 2. Wettrunde
- Eine Karte offen in die Mitte („Turn“)
- 3. Wettrunde
- Eine Karte offen in die Mitte („River“)
- 4. Wettrunde
- Showdown



# Wettrunde I

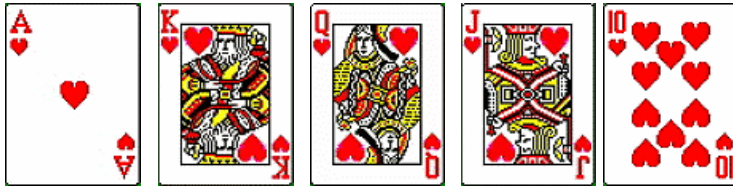
- passen („fold“), mitgehen („call“ / „check“), erhöhen („bet“ / „raise“)
- fold: Keine Zahlung, Spieler scheidet aus
- call: Spieler zahlt den geforderten Einsatz
- raise: Spieler zahlt den geforderten Einsatz und erhöht. Die anderen Spieler sind wieder gefordert.

# Wettrunde II

- Eigene Wette darf nicht erhöht werden
- Maximal 3 x „raise“ pro Runde
- Einsatzhöhe festgelegt  
z.B.: 10\$ Runde 1 - 2, \$20 Runde 3 - 4
- Am Ende einer Runde haben alle Spieler den gleichen Betrag im „Pot“ oder gepasst

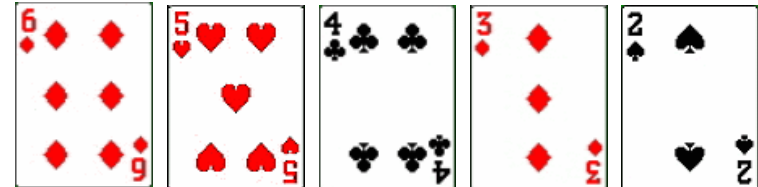
# Rangfolge

1.



Straight Flush

5.



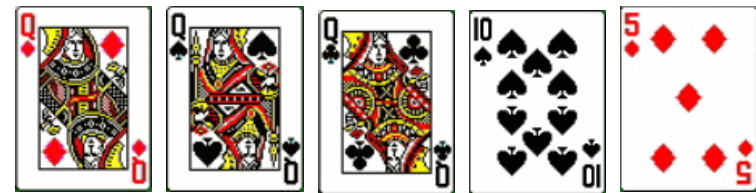
Straight

2.



Four of a kind

6.



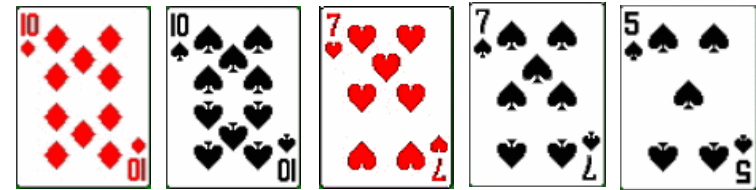
Three of a kind

3.



Full house

7.



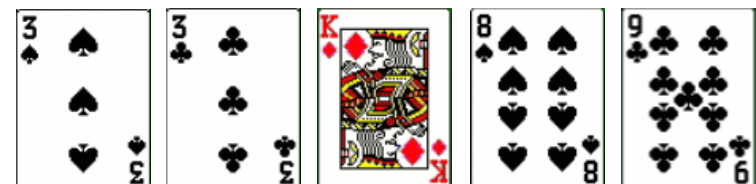
Two Pairs

4.



Flush

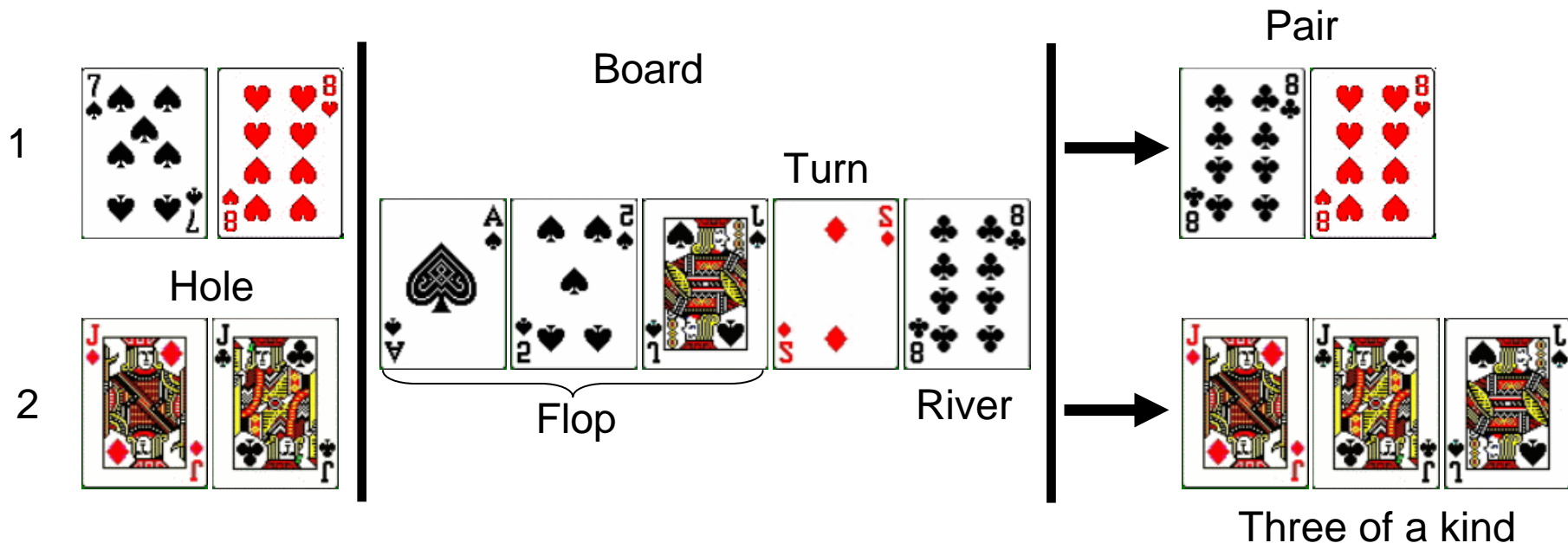
8.



Pair

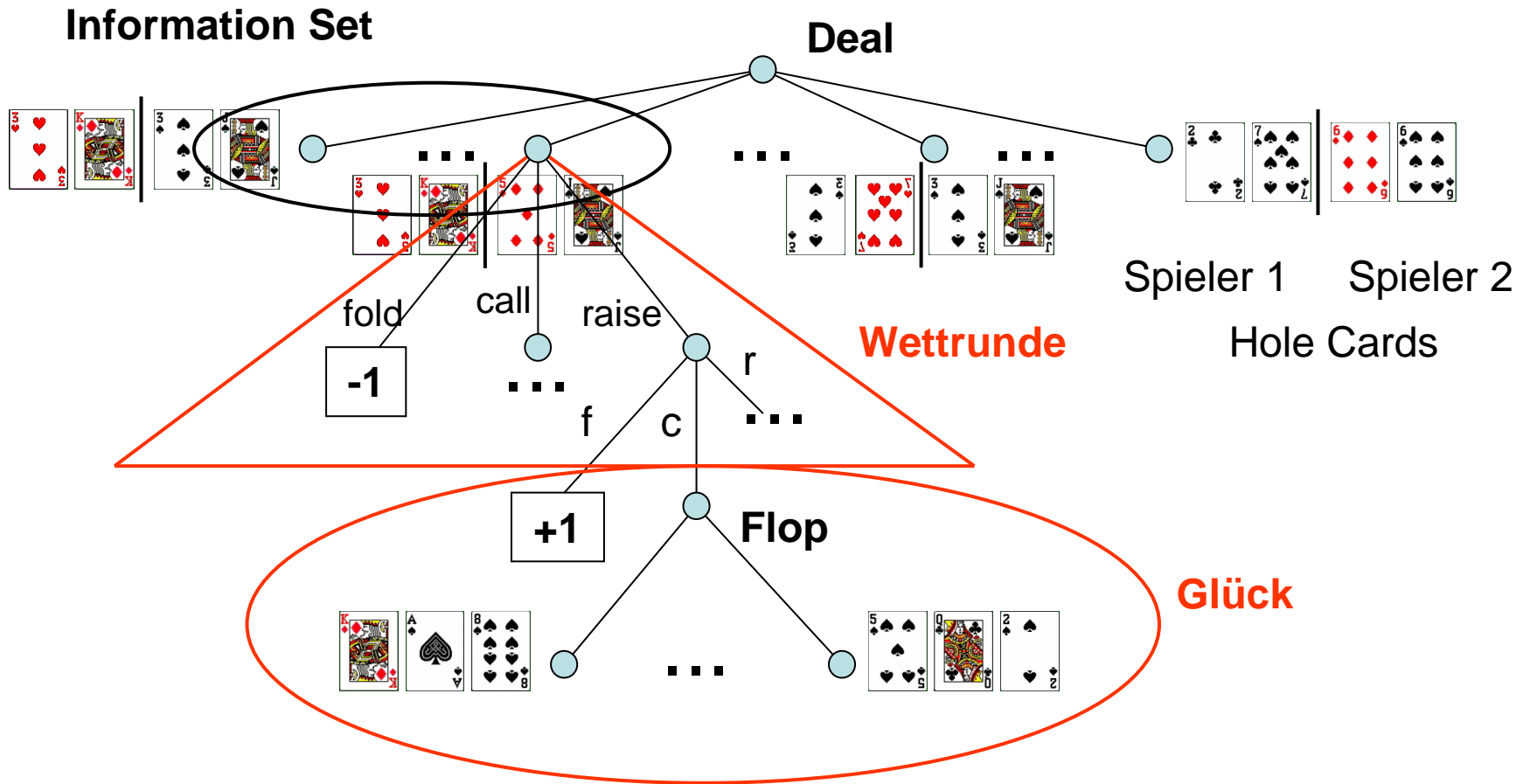
# Showdown

- Haben alle Spieler außer einem gepasst, kassiert der den Pot
- Beste „Hand“ aus den „Hole Cards“ und den „Board Cards“ gewinnt den Pot





# Spielbaum



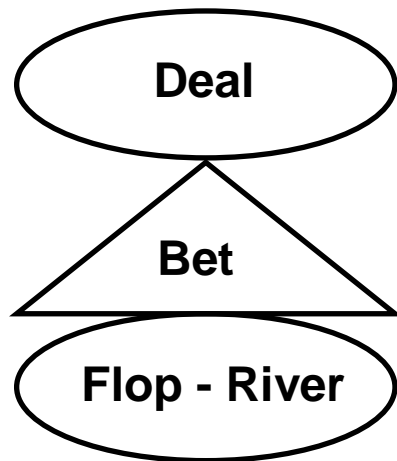
**insgesamt  $10^{18}$  Knoten**

# Computer Poker

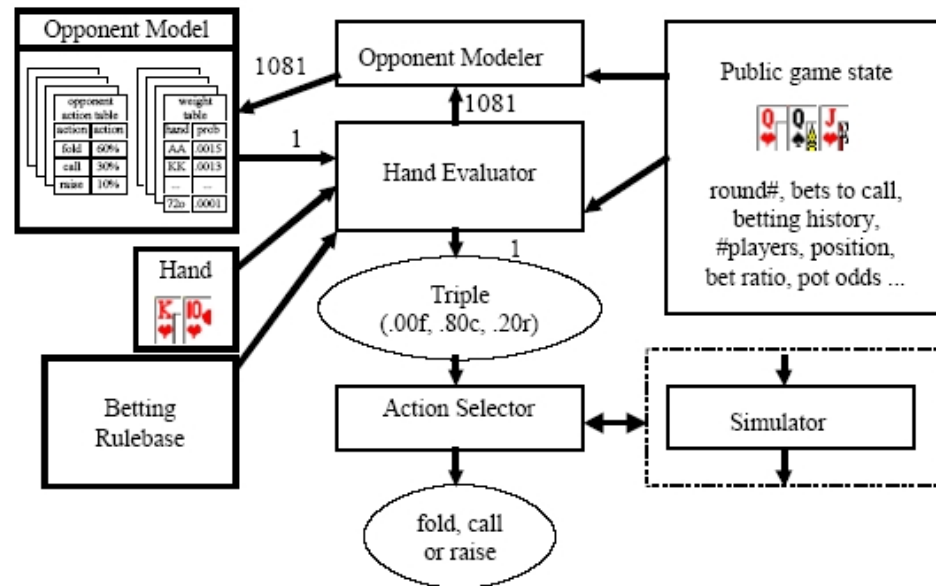
- Spielbaum zu groß für vollständige Suche
- Vereinfachung nötig
- pseudo-Optimale Strategie gesucht
- 3 Beispiele:
  - Pre-Flop Texas Hold'em
  - Poki
  - PsOpti

# Pre Flop Hold'em

- Keine Wetten nach dem Flop
- $EV = Pot * P(\text{strongest hand})$
- Maximierung mit Simplex Algorithmus

[illegible]

# Poki



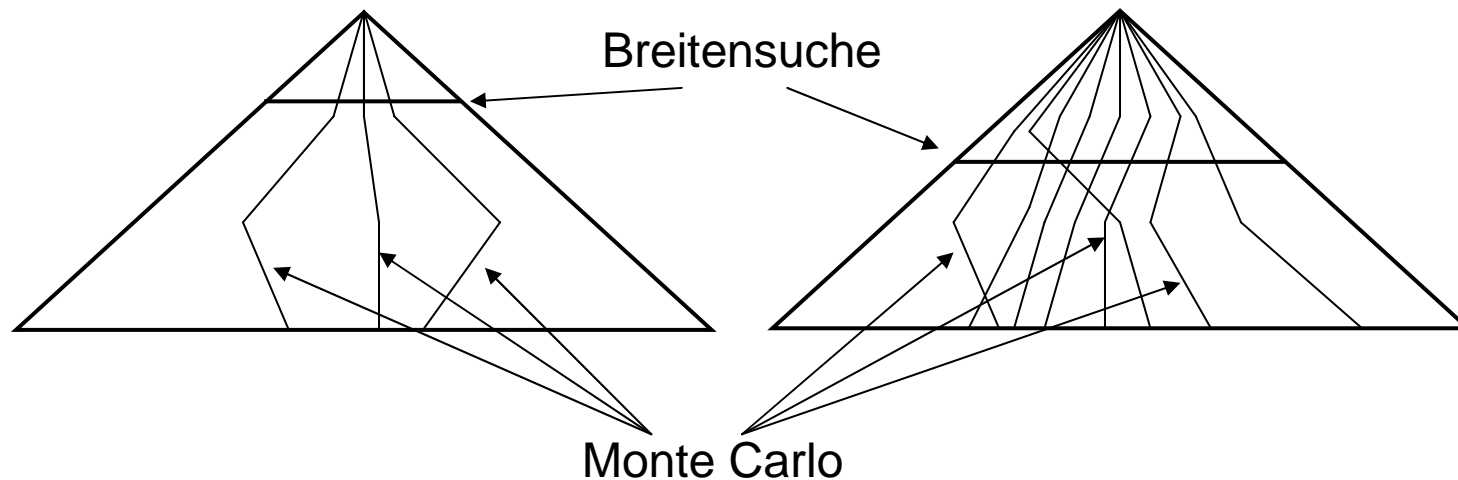
- Action selector: Pokerwissen
- Opponent Modeler: Neuronales Netz, Bayessches Lernen

# Action Selector

- Eingabe: „Effective hand strenght“  
     $\approx P(\text{ich habe die besten Karten})$
- Sitzposition, gemachte Wetten ...
- Verarbeitung mit Experten Regeln
- Ausgabe: (  $P(\text{„fold“})$ ,  $P(\text{„call“})$ ,  $P(\text{„raise“})$  )
- unflexibel, „unschön“ ABER spielt gut

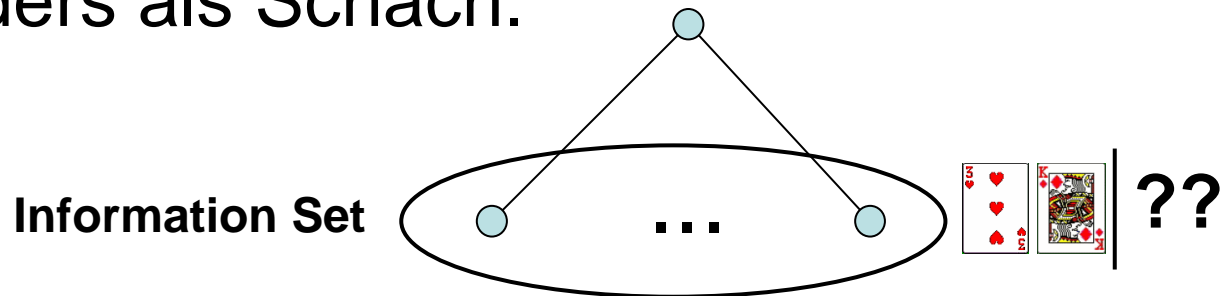
# Simulator

- Kein Expertenwissen  $\Rightarrow$  Spielbaum-Suche
- Keine Breitensuche  $\Rightarrow$  Monte-Carlo Simulation
- „schöner“ ABER spielt schlechter



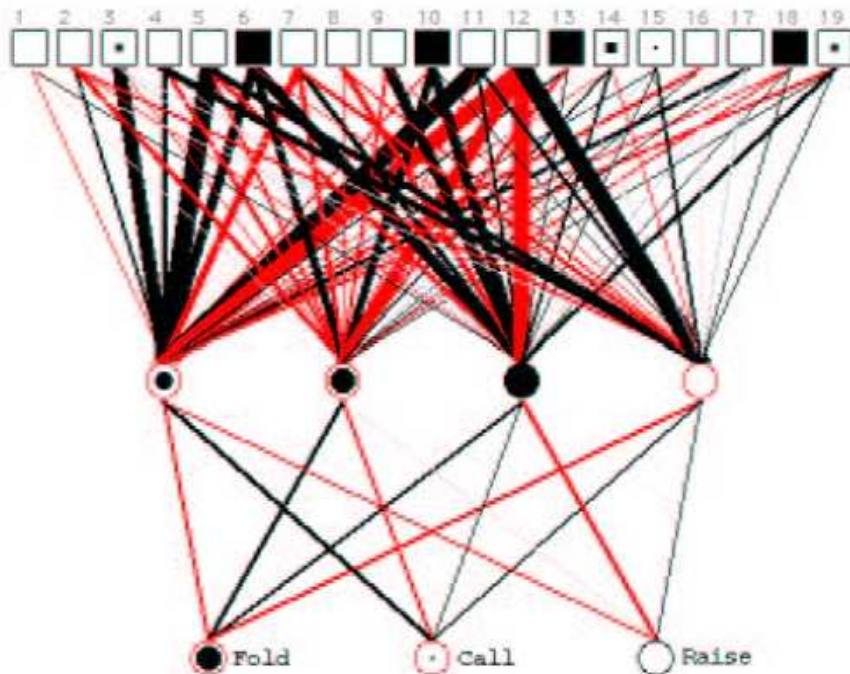
# Opponent Model

- Was macht der Gegner?
  - Wie Schach, ABER: Spieltheoretisch optimaler Zug nicht immer profitabelster Zug
- In welcher Position ist das Spiel?
  - Information Set hat mehr als ein Element!  
Anders als Schach.



# Was macht der Gegner?

- 3 Schichten mit 19, 4 und 3 Neuronen



#	type	description
1	Real	Immediate Pot Odds
2	Real	Bet Ratio: bets/(bets+calls)
3	Real	Pot Ratio: amount_in / pot_size
4	Boolean	Committed in this Round
5	Boolean	Bets-To-Call == 0
6	Boolean	Bets-To-Call == 1
7	Boolean	Bets-To-Call >= 2
8	Boolean	Stage == FLOP
9	Boolean	Stage == TURN
10	Boolean	Stage == RIVER
11	Boolean	Last-Bets-To-Call > 0
12	Boolean	Last-Action == BET/RAISE
13	Real	(#players Dealt-In) / 10
14	Real	(# Active Players) / 10
15	Real	(# Unacted Players) / 10
16	Boolean	Flush Possible
17	Boolean	Ace on Board
18	Boolean	King on Board
19	Real	(#AKQ on Board) / (# Board Cards)

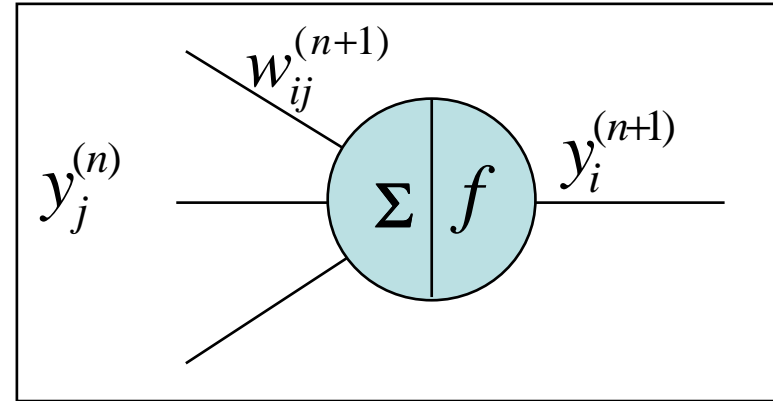


# Backpropagation

$$\text{Rein: } e_i^{(n+1)} = \sum_j y_j^{(n)} w_{ij}^{(n+1)}$$

$$\text{Raus: } y_i^{(n+1)} = f(e_i^{(n+1)})$$

Feed Forward

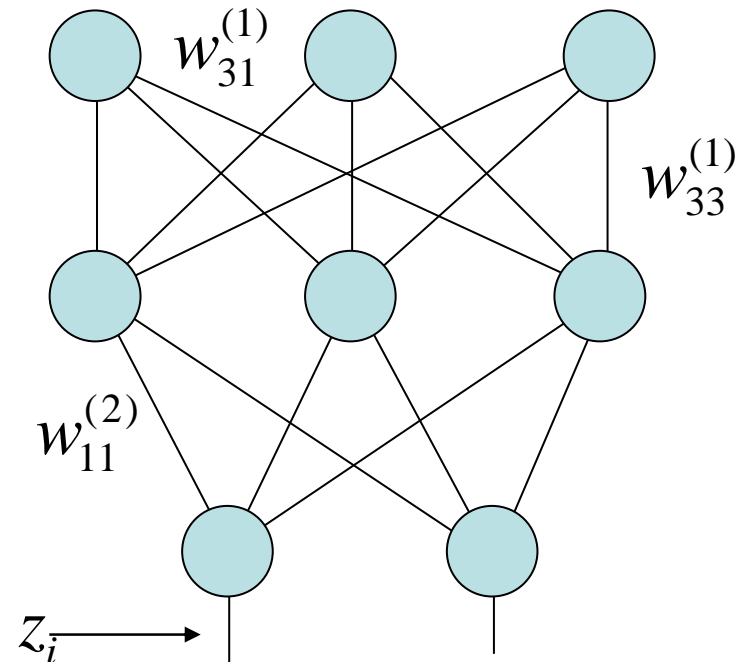


$$\delta_i^{(n_{\max})} = f'(e_i^{(n_{\max})})(y_i^{(n_{\max})} - z_i)$$

$$\delta_j^{(n)} = f'(e_j^{(n)}) \sum_i \delta_i^{(n+1)} w_{ij}^{(n+1)}$$

$$w_{ij}^{(n)'} = w_{ij}^{(n)} + \alpha \delta_i^{(n)} y_j^{(n-1)}$$

Backpropagation



# Resultate

Predictor	Bots	Humans	$\sigma$
Statistics I	60.94%	66.76%	0.023
Statistics II	63.94%	69.04%	0.115
Expert Formula	67.87%	55.98%	0.061
Neural Network I	64.67%	79.34%	0.086
Neural Network II	73.49%	79.91%	0.054
Meta-Predictor	75.11%	80.49%	0.023

# Wo ist das Spiel?

- Wahrscheinlichkeit für Gegnerkarten
- Aktionen  $\Rightarrow$  Änderung der W'keiten

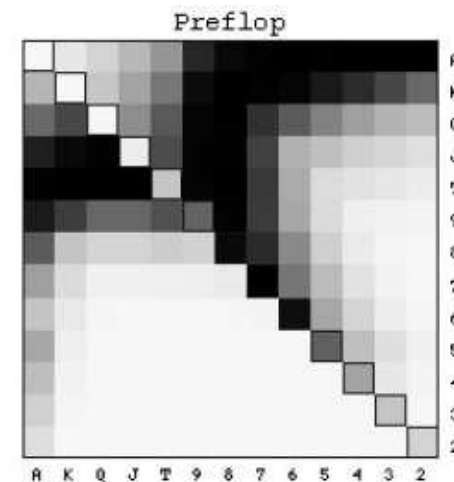
Vorher:  $A\clubsuit K\heartsuit = 0.4$

$P(\{\text{fold}, \text{call}, \text{raise}\}) = \{0, 0.3, 0.7\}$

Beobachtet: „call“

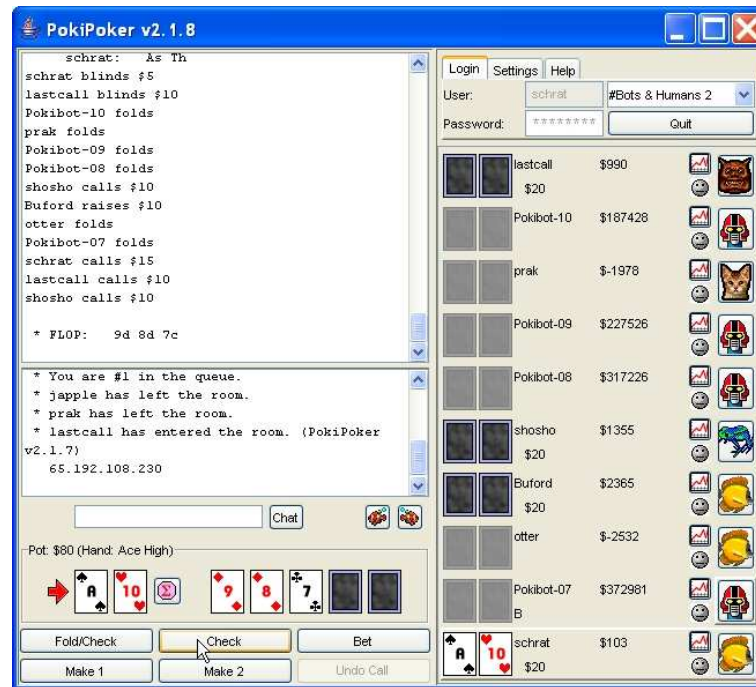
Nachher:  $A\clubsuit K\heartsuit = 0.4 * 0.3 = 0.12$

Opponent Model



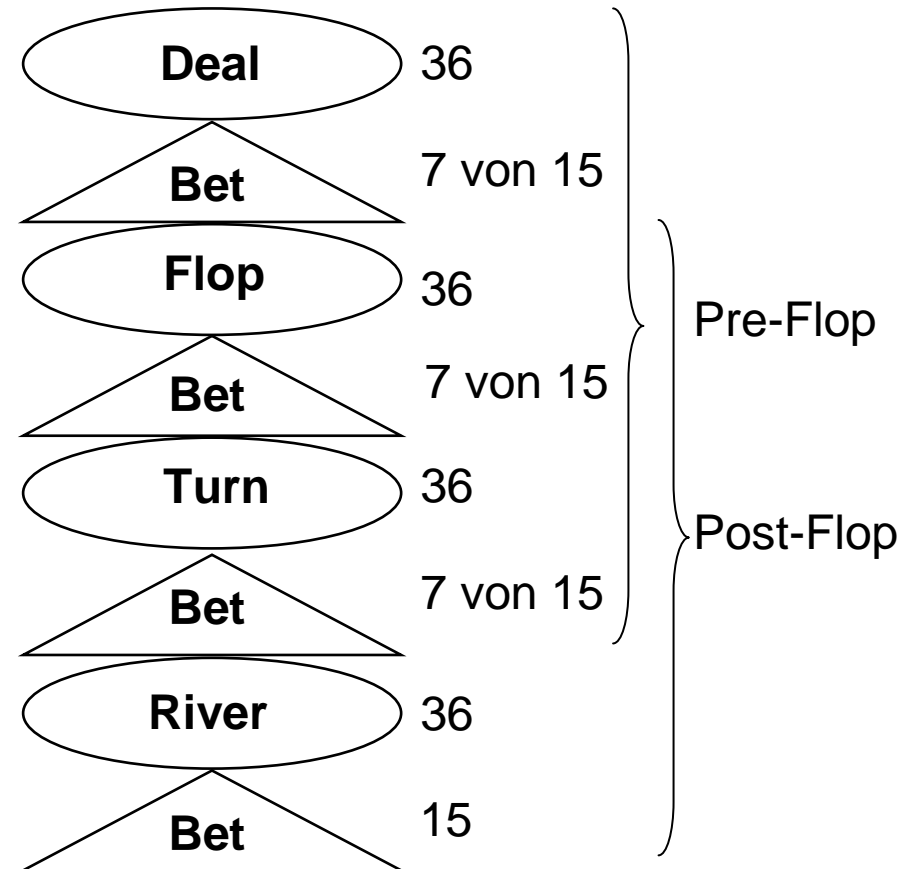
# Resultate

- Mittelmäßig bei 10 Spieler Hold'em
- Schwach bei 2 Spieler Poker
- Ausprobieren:



# PsOpti

- 2 Spieler Poker
- Spielbaumreduktion  
 $10^{18} \Rightarrow 2 \times 10^7$
- Techniken
  - Wettrunden kürzen
  - Wettrunden streichen
  - „bucketing“



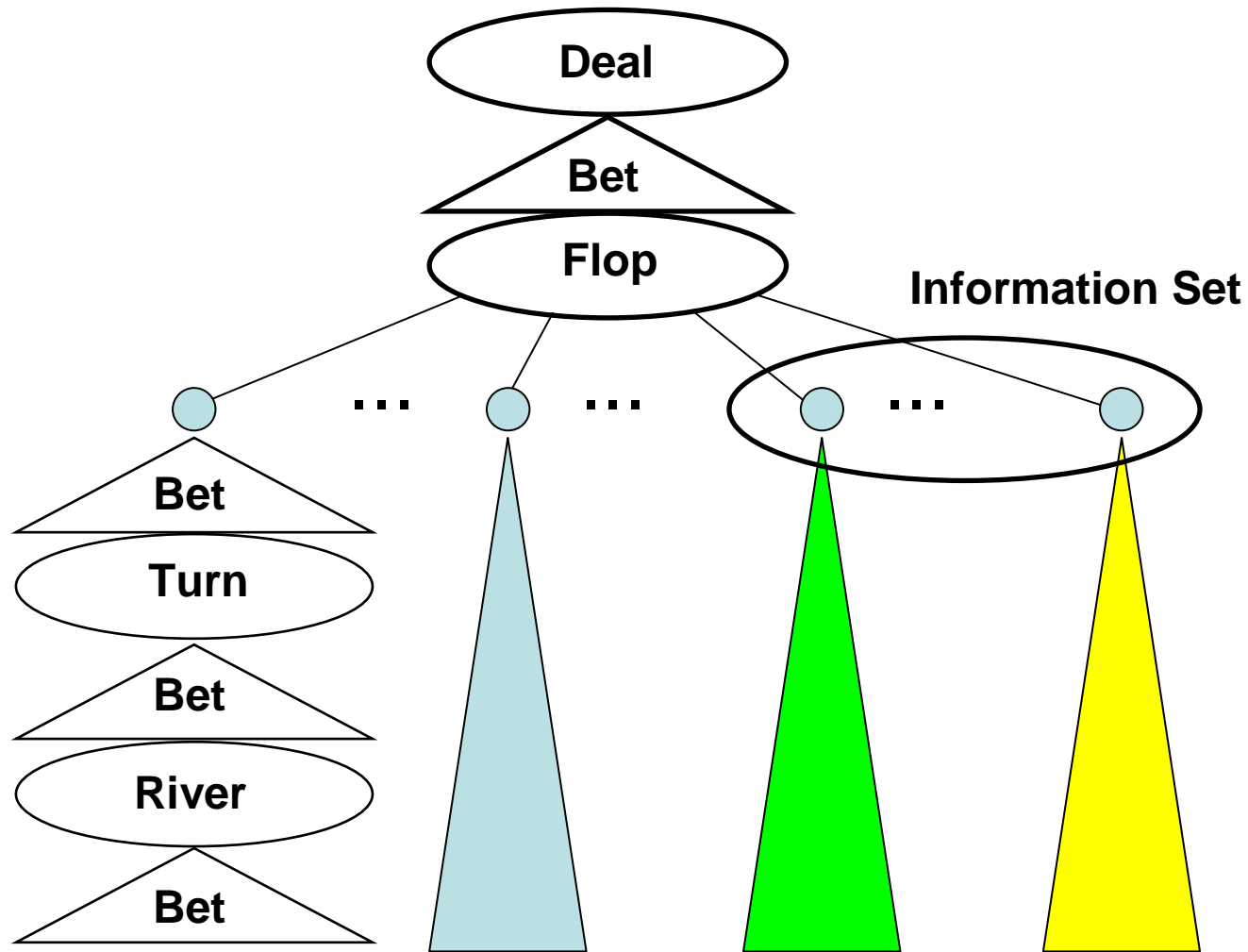
# Wettrunden kürzen

- Normal: 19 mögl. Sequenzen pro Runde
- bet – call, bet – raise – fold ...
- 9 davon gehen in die nächste Runde
- Reduziert: 7 von 15 in die nächste Runde
- Sequenzen mit 3 raise werden gestrichen
- Rechtfertigung: Experimente

# Wettrunden streichen

- Letzte Wettrunde streichen
- Kaum Auswirkung auf pre-Flop Spiel
- vgl. Selby
- Erste Wettrunde streichen
- Informationen in post-Flop Modell hinübernehmen ABER nicht alle
- Wettsequenz JA, Kartenverteilung nur als Wahrscheinlichkeit

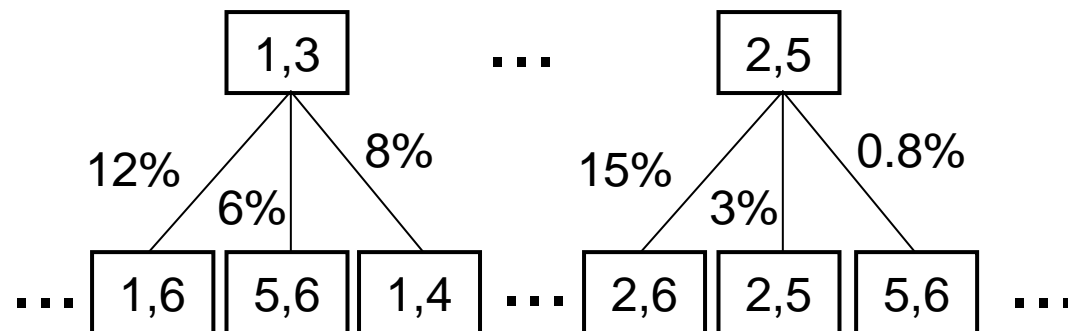
# Post-Flop Modelle





# Bucketing

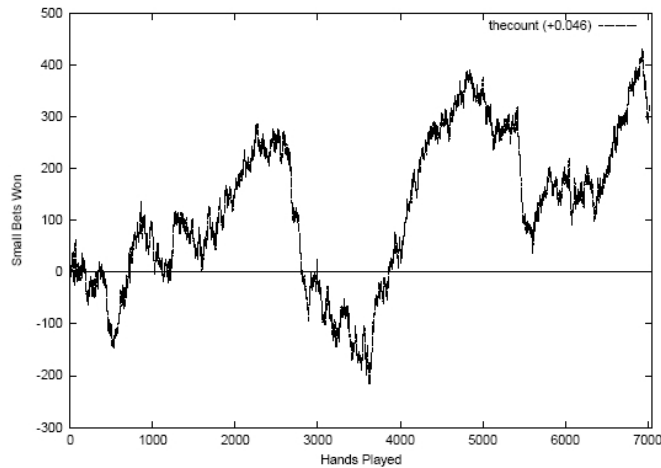
- Idee:  $K♥A♥ \sim K♣A♣$  - Äquivalenzklassen
- Mit 3 Wettrunden nur 6 „Buckets“
- Glücksknoten überführen von einem in den anderen Bucket



- Lösung mit Linearem Programm

# Resultate

- Gegen Profi nicht „untergegangen“



- Fehlt: Opponent Modelling

# Links / Literatur

- Selby:  
[www.archduke.demon.co.uk/simplex/art](http://www.archduke.demon.co.uk/simplex/art)
- PsOpti / Poki:  
[www.cs.ualberta.ca/~games/poker](http://www.cs.ualberta.ca/~games/poker)  
[www.cs.ualberta.ca/~darse/Papers/AIJ02.pdf](http://www.cs.ualberta.ca/~darse/Papers/AIJ02.pdf)  
[www.cs.ualberta.ca/~darse/Papers/IJCAI03.pdf](http://www.cs.ualberta.ca/~darse/Papers/IJCAI03.pdf)  
[www.cs.ualberta.ca/~darse/Papers/ICAI00.pdf](http://www.cs.ualberta.ca/~darse/Papers/ICAI00.pdf)
- Online Poker:  
[games.cs.ualberta.ca/webgames/poker/downloads/PokiPoker.zip](http://games.cs.ualberta.ca/webgames/poker/downloads/PokiPoker.zip)