The GNU Go Engine

- Overview
- Analyzing Moves
- Move Generation
- Worms and Dragons
- Eyes and Half Eyes
- The Pattern Code
- Tactical Reading
- Pattern Based Reading
- Influence Function

Overview – Directory Structure

`engine/`	Main GNU Go engine
`pattern/`	Pattern matching
`interface/`	User interfaces (Terminal, GTP, GMP, Emacs)
`sgf/`	SGF file reading / writing
`regression/`	GTP regression tests
`utils/`	Code not related to Go
`doc/`	Documentation (Texinfo)

Overview – Gathering Information

`examine_position`

1. Find blocks – `make_worms`
2. Tactical reading
3. Call influence (1)
4. Find groups – `make_dragons`
5. L&D / Semeai analysis for weak groups (*owl code*)
6. Call influence (2) $\rightarrow$ Territory

Overview – Move Generators

Block capture / defend	`worm_reasons`
Group kill / defend	`owl_reasons`
Semeai	`semeai_move_reasons`
Break into opponent's territory	`break_in_move_reasons`
Fuseki	`fuseki`
Patterns	`shapes`
Combination threats / atari sequences	`combinations`
Ensure opponent's dead groups stay dead	`revise_thrashing_dragon`
Endgame	`endgame_shapes`
Ensure semeai win stays	`revise_semeai`
Fill liberties	`fill_liberty`

Overview – Move Valuation

`review_move_reasons`

1. Territorial effect

2. Other effects: combination attacks, strategical effects, connections
   (large set of heuristics necessary to avoid duplication)

Overview – Move Generation

`genmove`

`estimate_score`

`choose_strategy`

`collect_move_reasons`

`fuseki`

`break_mirror_go`

`shapes`

`combinations`

`review_move_reasons`

If ahead and there is a thrashing dragon, reconsider the position

`endgame_shapes`

`endgame`

If no move found yet, revisit any semeai

If no move found yet, run `fill_liberty`

Analyzing Moves

- Verbose debug output

  `gnugo --trace [--allpats]`

- GTP commands

  e.g. `dragon_data P12`

- SGF Traces

  `start_sgftrace`, `finish_sgftrace`

- Debugger

  `.gdbinit` defines new commands, e.g.

  `(gdb) dragon P12`

Move Generation – Move Reasons

Attacking and Defending	Worm attack and defense
Threats to Attack or Defend	Worm threats, sente, ko threats
Multiple Attack or Defense	Combined attacks and defenses
Cutting and Connecting	Primariliy pattern based
Semeai Winning Moves	
Making or Destroying Eyes	Vital eye moves
Antisuji Moves	Never played, pattern based
Territorial Moves	Block or expand territory
Attacking and Defending Dragons	Owl attack and defense, L&D
Combination Attacks	Coordinated threats such as double ataris

Move Generation – Valuation

Value is measured in actual points

- Territoral Value

  Estimate change in territory `estimate_territorial_value`

  Considers L&D and influence `influence_delta_territory`

- Strategical Value

  Strengthen or weaken dragons

- Shape Factor

  Pattern attribute, multiplies score, complicate scheme to avoid overvaluation

- Secondary Value

  E.g. capture also connects, low weight

Worms

Worms – `struct worm_data`

- Basic properties of blocks (color, origin, ...)
- Size, higher order liberties
- Cutting stones
- Tactical attack/defense moves

Dragons

Amalgamation – `make_dragons`

- Eye based `dragon_eye`
- Shared/protected liberties
- `patterns/conn.db`

Dragons – `struct dragon_data, struct dragon_data2`

- Adjacent dragons
- Moyo size
- Weakness, genus, escape
- Safety
- Semeai/owl attack/defense points

Eyes and Half Eyes

- Eye spaces

  surrounded by same dragon and marginal points

- `patterns/eyes.db`

- Reduction to smaller eyespaces by search

- Isomorphic eyeshapes

- Eye local game values

  attacker moves twice, attacker moves first,

  defender moves first, defender moves twice

- Half eyes and false eyes

- Eyes with ko

The Pattern Code

```
?X?.?        jump under
O.*oo
O....
o....
-----


:8,ed,NULL
```

```
O   friendly stone
X   enemy stone
.   empty vertex
*   O's next move
o   O or empty
x   X or empty
?   Don't care
-   Edge
```

Pattern information

`:trfno, classification, [values], helper_function`

Pattern attributes

- Constraints e.g. `o` – every `O` is `DEAD` or `UNKNOWN`
- Actions e.g. `a` – move attackes all `X` dragons
- Optional values e.g. `shape(x)` adds `x` to move's shape value

Tactical Reading

- Return codes 0, KO_B, KO_A or WIN
- Block is safe with ≥ 5 liberties (≥ 3 if `stackp>depth`)
- Moves generated depend on number of liberties and search depth
- Zobrist hashing
- Persistent reading cache (cleared after each permanent move)
- Connection reading

Pattern Based Reading

Owl code

- `owl_attackpats.db` – attack moves
- `owl_defendpats.db` – defend moves
- `owl_vital_apats.db` – override eyeshape graph analysis

Combination reading

- Find double threats
- `aa_attackpats.db`

Influence Function

Computation

- Influence from alive stones decreases exponentially
- Alive stones and connections block influence
- Pattern based corrections `influence.db`, `barriers.db`
- Reading based corrections (Break ins)
- Classification Territory, Moyo, Area

Usage

- Static safety of dragons (moyo)
- Territorial value of moves
- Escape potential of dragons