Communication protocols specification

Maciej Kozarzewski

Contents

1 Introduction

This document describes the details of how protocols are implemented in AlphaGomoku (AG).

2 Gomocup Protocol

Here are listed all Gomocup protocol [1] commands that are implemented in AG with description of how they work in this particular program.

The move format is assumed to be `[column],[row]`, often abbreviated by [C],[R] or C,R.

Note: In order to use this protocol in AG, field ”protocol” in the configuration file must be set to ”gomocup”.

2.1 START [size]

Upon receiving this command, engine sets the board size.

Note: In AG, [size] can be either 15 or 20. For any other value AG will respond with an `ERROR` message. This is later checked in combination with `INFO rule` value where also an `ERROR` can be sent.

Note: In contrary to the official protocol description, AG does not actually initialize itself here. `START` and `RESTART` commands just clear the board, while the former also sets board size.

Note: If logging to file is enabled, a new file will be created for every `START` command.

2.2 TURN [C],[R]

This command is implemented exactly like in the protocol description. If `TURN [C],[R]` command is sent as the first one after `START`, engine will behave like if it received:

```
BOARD
[C],[R],2
DONE
```

2.3 BEGIN

This command is implemented exactly like in the protocol description.

2.4 BOARD

This command is implemented exactly like in the protocol description.

Note: The third field (move type) value of 3 is not supported.

Note: If the position is invalid (too many black or white stones, etc.), AG will respond with an `ERROR` message.

Note: AG will keep its cache from previous searches, clearing only those entries that are provably useless given the current board state. It means that if the same board is set two times in a row, in the second search AG will use all the information collected during the first search.

2.5 INFO [key] [value]

In theory AG can react to the change of any parameters even during search (except `rule` which must be specified before any search starts), but no guarantees are made if some contradictory values are sent. If anything goes wrong the error description might be found in the logfile.

Note: Official protocol description states that *"The manager does not read messages from the brain when sending INFO command."* [1]. This is why it was suggested that engine should wait with reponse to the `INFO` command until actual match is started. However AG responses immediately if an invalid `INFO` command was sent, assuming that the output is buffered and will eventually reach the manager when it will switch to reading messages.

2.5.1 INFO timeout_turn [x]

Sets the time limit for each move in milliseconds. If $[x] < 0$ the time for move is unlimited.

Note: AG will end the search when it runs out of memory.

Note: AG parses [x] as 64 bit integer.

Note: If $[x] = 0$ is set, it may take some more time than exactly 0 ms for AG to respond.

Note: If remaining timeouts are set to be infinite, AG will search exactly for `timeout_turn` milliseconds.

2.5.2 INFO timeout_match [x]

Sets the time limit of a whole match in milliseconds.

Note: AG parses [x] as 64 bit integer.

Note: If $[x] \leq 0$ the time for match is unlimited.

2.5.3 INFO time_left [x]

Specifies how much time remains for the match in milliseconds.

Note: AG parses [x] as 64 bit integer.

Note: AG will end the search when it runs out of memory.

2.5.4 INFO max_memory [x]

Sets the memory limit in bytes.

Note: If $[x] <= 0$ memory usage is not limited.

Note: AG uses only as much memory as it needs, often lower than [x].

Note: If the memory limit is too low, AG may stop the search when it runs out of memory.

Note: This setting can be increased at any time. But lowering memory limit is only possible when the search is not running and will mean re-creating entire search tree from scratch.

2.5.5 INFO game_type [x]

Note: This information is ignored by the AG.

2.5.6 INFO rule [x]

Specifies the rules used for the game. A bitmask that can have several possible values:

$[x] = 0$	five or more wins - freestyle rule. Supported only for 20x20 and 15x15 board.
$[x] = 1$	exactly five wins - standard rule. Supported only for 15x15 board.
$[x] = 2$	continuous game - not supported.
$[x] = 4$	renju rule. Supported only for 15x15 board.
$[x] = 8$	freestyle caro rule (OXXXXXO is not a win, but OXXXXXXO is). Supported only for 15x15 board.
$[x] = 9$	standard caro rule (OXXXXXO is not a win, OXXXXXXO is not a win too). Supported only for 15x15 board.

Note: This setting cannot be changed during the search.

2.5.7 INFO evaluate [C],[R]

A request to evaluate move with coordinates [C],[R]. AG will answer with a `MESSAGE` with evaluation of the [C],[R] move at root, similar to the one sent after search ends.

2.5.8 INFO folder [x]

Note: AG does not save any temporary data so this info is ignored.

2.5.9 Default values

Some parameters have default values so AG works even if no `INFO` was sent:

- **timeout_turn** = 5000 (5 seconds)
- **timeout_match** = 2^{53} ($\approx$ 285616 years - effectively no limit)
- **max_memory** = 268435456 (256MB)
- **time_left** = 2^{53} ($\approx$ 285616 years - effectively no limit)
- **rule** = 0 (freestyle rule)

2.6 END

This command is implemented exactly like in the protocol description.

Note: AG often exceeds GUI timeout limits because it needs to deallocate all its resources that can reach gigabytes of RAM memory (for long searches using GPUs).

2.7 ABOUT

AG will respond with following informations : "name", "version", "author", "country", "www", "email".

2.8 RECTSTART [width],[height]

Rectangular boards are not supported in AG, but this command is implemented and will return an `ERROR` message.

2.9 RESTART

AG behaves just like for `START` command keeping the previous settings.

2.10 TAKEBACK [C],[R]

This command **might not be** implemented like in the protocol description. This command can be used to undo **only the last move**. It can be used repeatedly to undo several moves, but they must be sent in the opposite order to the one in which they were played. During regular game this condition is obviously fulfilled. Problems may appear when trying to undo moves that were sent in the `BOARD` command as it may send moves in random order. If an incorrect last move was specified, AG will simply send an `ERROR` message with the description which move it considers to be the last one.

2.11 UNKNOWN [error message]

This command is implemented exactly like in the protocol description.

Note: AG will send back the command that was not understood.

2.12 ERROR [error message]

This command is implemented exactly like in the protocol description.

Note: Usually AG will send some information about what went wrong.

2.13 MESSAGE [message]

Currently the only `MESSAGE` AG sends by itself, is the summary of the search sent every time search stops, either naturally or by `STOP` command. It has the following format:

depth [x]-[y] the minimal [x] and maximal [y] depth of the search.

Note: Minimal is always 1 and maximal is taken as the length of principal variation.

ev [x] is the expectation outcome in % (probability of $P_{win} + \frac{1}{2}P_{draw}$) or proven value - (L)oss, (D)raw, (W)in. with number representing number of moves to this outcome (for example W3 is a win in 3 moves)

winrate [x] is the probability of winning the game in %

drawrate [x] is the probability of drawing the game in %

n [x] is the number of evaluated nodes.

n/s [**x**]	is the number of evaluated nodes per second.
tm [**x**]	is the time used for this turn (in milliseconds).
pv [**x1**] [**x2**] **...**	is the principal variation.

Example:
`MESSAGE` depth 1-45 ev 60.1 winrate 56.7 drawrate 6.9 n 288959 n/s 12487 tm 23140 pv Xf4 Of6 Xe6 Of5 Xc5 Oc4 Xh5 Og7 Xh8

Principal variation encoding format:

X or **O**	color - X is black, O is white.
letter	column, from left to right, lowercase.
number	row, from top to bottom, starting from 0.

2.14 DEBUG [message]

AG does not send any `DEBUG` messages.

3 Extended Gomocup Protocol

This protocol implements all commands from the base Gomocup protocol. Additionally it introduces new commands. Sometimes extends how the original ones work.

The move format is assumed to be `[column],[row]`, often abbreviated by [C],[R].

Note: In order to use this protocol in AG, field "protocol" in the configuration file must be set to "extended_gomocup".

3.1 SUGGEST [C],[R]

If `INFO analysis_mode` is set to `1`, engine will answer `BEGIN`, `BOARD` and `TURN` commands with `SUGGEST` [C],[R] instead of [C],[R] and will not change its internal state.

Note: The manager can send any command in response to `SUGGEST` except `TURN` and `TAKEBACK`.

Note: Pondering is never started automatically after `SUGGEST` response.

3.2 PLAY [C],[R]

It is used by the manager only as a respond to `SUGGEST` command. It imposes move [C],[R] to the engine. It must be answered by the engine with the same [C],[R] move as the sent one.

3.3 Additional INFO keys

In addition to the keys described earlier, also the following will be recognized.

3.3.1 INFO evaluate [C1],[R1] [C2],[R2] ...

The original `INFO evaluate` command is extended to handle arbitrary number of moves, so that they describe specific path within the tree. For example

- `INFO evaluate` without any parameters is a request for evaluation info of the current board state.
- `INFO evaluate 3,4` is a request for evaluation info of the board state after making move [3],[4] from the current board state.
- `INFO evaluate 3,4 5,6` is a request for evaluation info of the board state after making moves [3],[4] and [5],[6] from the current board state.

If path specified in the command does not exist in the tree for any reason, the engine should send an empty `MESSAGE`.

Note: Sending excessive amounts of `INFO evaluate` commands during the search is discouraged as it can degrade engine performance.

Note: This commands returns information about moves from the perspective of the player to move. After a move has been returned from the search, AG internally switches to the opponent color so `INFO evaluate` command will not return information about moves in the previous board state.

3.3.2 INFO analysis_mode [x]

Controls how engine responds to the `BEGIN`, `BOARD` and `TURN` commands. The allowed values are:

$[x] = 0$ engine will respond with the best move.

$[x] = 1$ engine will respond with `SUGGEST`.

3.3.3 INFO max_depth [x]

Sets maximum search depth. If $[x] < 0$ the search depth is unlimited.

Note: Since AG evaluates multiple nodes in parallel, it will stop as soon as the limit is reached but may exceed it by small amount.

3.3.4 INFO max_node [x]

Sets maximum number of nodes. If $[x] <= 0$ the number of nodes is unlimited.

Note: Since AG evaluates multiple nodes in parallel, it will stop as soon as the limit is reached but may exceed it by small amount.

3.3.5 INFO time_increment [x]

Sets amount of time that is added after each move (see Fischer time controls [3]). Must not be negative.

3.3.6 INFO auto_pondering [x]

Toggles the automatic pondering after each move. This command overrides the value of "auto_pondering" loaded from the configuration file. Allowed values are:

$[x] = 0$ automatic pondering is turned off

$[x] = 1$ automatic pondering is turned on

Note: This option does not start/stop the pondering by itself, only allows/forbids the engine to automatically start pondering.

Note: Pondering is never started automatically if `INFO analysis_mode` is set to 1.

3.3.7 INFO protocol_lag [x]

Sets the estimated protocol overhead i.e. the time difference (in milliseconds) between sending the message by the engine and receiving it by the GUI (and in the opposite direction). Must not be negative.

3.3.8 INFO thread_num [x]

Sets maximum nuber of threads an engine can use. This command overrides the value of "thread_num" loaded from the configuration file. Must be greater than 0.

Note: If number of threads is changed during the search, AG would not adjust its number of threads until the search ends.

3.3.9 Default values

Some parameters have default values so the engine works even if no `INFO` was sent:

- **analysis_mode** = 0 (disabled)
- **max_depth** = 2147483647 (effectively no limit)
- **max_nodes** = 2147483647 (effectively no limit)
- **time_increment** = 0 ("sudden death" [4] type time controls)
- **auto_pondering** = as loaded from configuration file
- **protocol_lag** ≈ 400 (0.4s - varies from version to version)
- **thread_num** = as loaded from configuration file

3.4 PONDER [value]

This command starts the search in pondering mode for [value] milliseconds. The pondering search mode differs from the regular search mode at one point - it will not output any best move at the end. If [value] is not specified or is negative, engine starts infinite pondering. In any case the pondering can be stopped using `STOP` command. Pondering mode exits automatically if any new position is set, either by `BEGIN`, `BOARD` or `TURN` command.

3.5 STOP

This command stops the current search. If it was regular search, either the best move will be returned or `SUGGEST` message will be sent, depending on the `INFO analysis_mode` settings. If it was pondering search it is simply stopped and no output is produced. Such interrupted pondering can be launched again by another `PONDER` command.

3.6 BALANCE [n]

Note: Currently not implemented in AG.

This command is similar to the `BOARD` command but instead of searching for the best move, it searches for [n] moves that create the most balanced position (balanced position is such where neither player has an advantage). For example

```
BALANCE 2
[C1],[R1],2
[C2],[R2],1
[C3],[R3],2
DONE
```

will order the engine to find two moves that balance this 3-stone position. Value of [n] must be sctictly positive. Only [n]=1 and [n]=2 are required but the engine is encouraged to accept any values of [n]. If the rule is renju, the moves must be sent in the order they were played. For other rules the ordering may be random.

Note: Execution of `BALANCE` command is considered to be a ”turn”, so the engine must obey the time controls (including `timeout_turn`).

3.7 SHOWFORBID

This command is similar to the `BOARD` command but instead of searching for the best move, it returns all forbidden moves according to renju rule. The engine will respond with `FORBID` followed by the list of moves separated by spaces. For example

```
SHOWFORBID
[C1],[R1],[S1]
[C2],[R2],[S2]
...
[Cn],[Rn],[Sn]
DONE
```

and engine responds for example with

```
FORBID 5,4 8,7 10,11
```

If the rule is not renju or there are no forbidden moves on board, engine should respond with a `FORBID` message without any moves.

Note: Execution of `SHOWFORBID` command is **not** considered to be a "turn", so the engine **does not have to** obey the time controls. It is recommended (but not required) for the engine to be able to respond to this command at any time.

3.8 CLEARHASH

Upon receiving this command the engine should clear its internal memory, so that the next search would start "fresh", without any data collected in previous searches. For example in α-β engines, this command probably will mean clearing the hashtable. Sending this command during the search will stop it.

Note: AG does not have such kind of hashtable like in α-β engines, but still there is some cache that will be cleared upon receiving this command.

3.9 PROTOCOLVERSION

The engine should answer to this command with the version number of implemented protocol.

```
PROTOCOLVERSION
[major],[minor]
```

3.10 SWAP1STBOARD

This command is used for swap after first opening rule [6]. It has two cases:

3.10.1 Case 1

The manager asks for the first stone:
```
SWAP1STBOARD
DONE
```
The engine answers with single move
```
C1,R1
```

3.10.2 Case 2

The manager sends the coordinates of the first stone and asks for the choice of options:
```
SWAP1STBOARD
C1,R1
DONE
```
The engine answers with one of the following:

`SWAP`	if the engine decides to swap
`C2,R2`	the coordinates of the 2nd move if the engine decides to stay with white color

After the opening stage, the stones on the board will be treated as an opening for a traditional Gomocup match, and the manager will communicate with the engine using the classical Gomocup protocol in the rest of the match.

Note: Each step of the `SWAP1ST` command is considered to be a "turn", so the engine must obey the time controls (including `timeout_turn`).

3.11 SWAPBOARD

This command is used for swap opening rule [7]. It has two cases:

3.11.1 Case 1

The manager asks for the first three stones:
```
SWAPBOARD
DONE
```
The engine answers with three moves separated by spaces:
```
C1,R1 C2,R2 C3,R3
```

3.11.2 Case 2

The manager sends the coordinates of the first three stones and asks for the choice of options:

```
SWAPBOARD
C1,R1
C2,R2
C3,R3
DONE
```

The engine answers with one of the following:

`SWAP`	if the engine decides to swap
`C4,R4`	the coordinate of the 4th move if the engine decides to stay with white color

After the opening stage, the stones on the board will be treated as an opening for a traditional Gomocup match, and the manager will communicate with the engine using the classical Gomocup protocol in the rest of the match.

Note: Each step of the `SWAPBOARD` command is considered to be a "turn", so the engine must obey the time controls (including `timeout_turn`).

3.12 SWAP2BOARD

This command is used for swap-2 opening rule [6]. It is described on Gomocup webpage and was included here for completeness. It has three cases:

3.12.1 Case 1

The manager asks for the first three stones:

```
SWAP2BOARD
DONE
```

The engine answers with three moves separated by spaces:

```
C1,R1 C2,R2 C3,R3
```

3.12.2 Case 2

The manager sends the coordinates of the first three stones and asks for the choice of options:

```
SWAP2BOARD
C1,R1
C2,R2
C3,R3
```

```
DONE
```

The engine answers with one of the following:

`SWAP`	if the engine decides to swap
`C4,R4`	the coordinates of the 4th move if the engine decides to stay with white color
`C4,R4 C5,R5`	the coordinates of the 4th and 5th stones if the engine decides to put two stones and let the opponent choose the color

3.12.3 Case 3

The manager sends the coordinates of the first five stones and asks for the choice of options:

```
SWAP2BOARD
C1,R1
C2,R2
C3,R3
C4,R4
C5,R5
DONE
```

The engine answers with one of the following:

`SWAP`	if the engine decides to swap
`C6,R6`	the coordinates of the 6th move if the engine decides to stay with white color

After the opening stage, the stones on the board will be treated as an opening for a traditional Gomocup match, and the manager will communicate with the engine using the classical Gomocup protocol in the rest of the match.

Note: Each step of the `SWAP2BOARD` command is considered to be a "turn", so the engine must obey the time controls (including `timeout_turn`).

3.13 SWAP5

This command is used for swap-5 opening rule [6]. It has three steps:

3.13.1 SWAP5STEP1

The manager asks for the first stone:

```
SWAP5STEP1
DONE
```

The engine answers with a single move, for example

```
C1,R1
```

3.13.2 SWAP5STEP2

This case will be repeated several times, each time the command will be sent to the appropriate player. The manager sends the coordinates of already placed stones (from 1 to 5) and asks for the choice of options:

```
SWAP5STEP2
C1,R1
...
CN,RN
DONE
```

The engine answers with one of the following:

`SWAP`	if the engine decides to swap
`C,R`	the coordinates of the next move if the engine decides to stay with the current color

3.13.3 SWAP5STEP3

This case will be repeated several times, each time the command will be sent to the appropriate player. The manager sends the coordinates of already placed stones (from 1 to 5) and the engine must respond with the next move:

```
SWAP5STEP3
C1,R1
...
CN,RN
DONE
```

The engine answers with a single move, for example

```
C,R
```

After the opening stage, the stones on the board will be treated as an opening for a traditional Gomocup match, and the manager will communicate with the engine using the classical Gomocup protocol in the rest of the match.

Note: Each step of the `SWAP5BOARD` command is considered to be a "turn", so the engine must obey the time controls (including `timeout_turn`).

3.14 PROBOARD

This command is used for pro opening rule [7]. It has three cases:

3.14.1 Case 1

The manager asks for the first stone:

```
PROBOARD
DONE
```

The engine answers with a single move at the center of the board i.e. (width/2),(height/2) rounded down to the nearest integer.

```
C1,R1
```

3.14.2 Case 2

This case will be repeated several times, each time the command will be sent to the appropriate player. The manager sends the coordinates of already placed stones (from 1 to 5) and asks for the choice of options:

```
PROBOARD
C1,R1
DONE
```

The engine answers with the coordinates of the second move: `C2,R2`

3.14.3 Case 3

This case will be repeated several times, each time the command will be sent to the appropriate player. The manager sends the coordinates of already placed stones (from 1 to 5) and the engine must respond with the next move:

```
PROBOARD
C1,R1
C2,R2
DONE
```

The engine answers with a single move that must be outside the central 5x5 square (center is defined by the position of the first stone), for example

```
C3,R3
```

After the opening stage, the stones on the board will be treated as an opening for a traditional Gomocup match, and the manager will communicate with the engine using the classical Gomocup protocol in the rest of the match.

Note: Each step of the `PROBOARD` command is considered to be a "turn", so the engine must obey the time controls (including `timeout_turn`).

3.15 LONGPROBOARD

This command is used for long pro opening rule [7]. It is almost exactly like the `PRO` but the third move must be made outside of the central 7x7 square (instead of 5x5).

3.16 RIF

This command is used for RIF opening rule [6]. It has five steps:

3.16.1 RIFSTEP1

The manager asks for the first three stones forming one of the 26 Renju openings (or any three stones for rules other than Renju):

```
RIFSTEP1
DONE
```

The engine answers with three moves separated by spaces, for example

```
C1,R1 C2,R2 C3,R3
```

3.16.2 RIFSTEP2

The manager sends the coordinates of the first three stones and asks for the choice of options:

```
RIFSTEP2
C1,R1
C2,R2
C3,R3
DONE
```

The engine answers with one of the following:

`SWAP`	if the engine decides to swap
`C4,R4`	output the coordinate of the 4th move if the engine decides to stay with the current color

3.16.3 RIFSTEP3

If one of the player chosed to swap in the Case 2, the other player will be sent this variant. The manager sends the coordinates of the first three stones and asks for the 4th move:

```
RIFSTEP3
C1,R1
C2,R2
```

```
C3,R3
DONE
```
The engine answers with the coordinates of the 4th move:
```
C4,R4
```

3.16.4 RIFSTEP4

The manager sends the coordinates of four stones:
```
RIFSTEP4
C1,R1
C2,R2
C3,R3
C4,R4
DONE
```
The engine answers with two candidates for the 5th move separated by spaces (moves must not be symmetrical), for example
```
C5a,R5a C5b,R5b
```

3.16.5 RIFSTEP5

The manager sends the coordinates of four stones along with the two candidates for 5th move:
```
RIFSTEP5
C1,R1
C2,R2
C3,R3
C4,R4
DONE
C5a,R5a C5b,R5b
```
The engine answers with two moves, first must be one of the two offered 5th moves, while the other is the 6th move, for example
```
C5,R5 C6,R6
```
After the opening stage, the stones on the board will be treated as an opening for a traditional Gomocup match, and the manager will communicate with the engine using the classical Gomocup protocol in the rest of the match.

Note: Each step of the `RIF` command is considered to be a "turn", so the engine must obey the time controls (including `timeout_turn`).

3.17 YAMAGUCHI

This command is used for Yamaguchi opening rule [6]. It has five steps:

3.17.1 YAMAGUCHISTEP1

The manager asks for the first three stones forming one of the 26 Renju openings (or any three stones for rules other than Renju):

```
YAMAGUCHISTEP1
DONE
```

The engine answers with three moves separated by spaces and the number of offered 5th moves

```
C1,R1 C2,R2 C3,R3 N
```

3.17.2 YAMAGUCHISTEP2

The manager sends the coordinates of the first three stones along with the number of offered 5th moves and asks for the choice of options:

```
YAMAGUCHISTEP2 N
C1,R1
C2,R2
C3,R3
DONE
```

The engine answers with one of the following:

`SWAP`	if the engine decides to swap
`C4,R4`	output the coordinate of the 4th move if the engine decides to stay with the current color

3.17.3 YAMAGUCHISTEP3

If one of the player chosed to swap in the Case 2, the other player will be sent this variant. The manager sends the coordinates of the first three stones along with the number of offered 5th moves and asks for the 4th move:

```
YAMAGUCHISTEP3 N
C1,R1
C2,R2
C3,R3
DONE
```

The engine answers with the coordinates of the 4th move:

```
C4,R4
```

3.17.4 YAMAGUCHISTEP4

The manager sends the coordinates of four stones along with the number of offered 5th moves:

```
YAMAGUCHISTEP4 N
C1,R1
C2,R2
C3,R3
C4,R4
DONE
```

The engine answers with N candidates for the 5th move separated by spaces (moves must not be symmetrical)

```
C5a,R5a C5b,R5b C5c,R5c ...
```

3.17.5 YAMAGUCHISTEP5

The manager sends the coordinates of four stones along with the candidates for 5th move:

```
YAMAGUCHISTEP5 N
C1,R1
C2,R2
C3,R3
C4,R4
PLAY
C5a,R5a C5b,R5b C5c,R5c ...
```

The engine answers with two moves, first must be one of the offered 5th moves, while the other is the 6th move, for example

```
C5,R5 C6,R6
```

After the opening stage, the stones on the board will be treated as an opening for a traditional Gomocup match, and the manager will communicate with the engine using the classical Gomocup protocol in the rest of the match.

Note: Each step of the `YAMAGUCHI` command is considered to be a ”turn”, so the engine must obey the time controls (including `timeout_turn`).

3.18 SOOSYRV

This command is used for Soosyrv-N opening rule [6]. It has six steps:

3.18.1 SOOSYRVSTEP1

The manager asks for the first three stones forming one of the 26 Renju openings (or any three stones for rules other than Renju):

```
SOOSYRVSTEP1 N
DONE
```

The engine answers with three moves separated by spaces, for example
`C1,R1 C2,R2 C3,R3`

3.18.2 SOOSYRVSTEP2

The manager sends the coordinates of the first three stones along with the maximum number of offered 5th moves and asks for the choice of options:

```
SOOSYRVSTEP2 N
C1,R1
C2,R2
C3,R3
DONE
```

The engine answers with one of the following:

`SWAP`	if the engine decides to swap
`C4,R4 M`	the coordinate of the 4th move and the number of offered 5th moves (from 1 to N) if the engine decides to stay with the current color

3.18.3 SOOSYRVSTEP3

If one of the player chosed to swap in the Case 2, the other player will be sent this variant. The manager sends the coordinates of the first three stones along with the actual number of offered 5th moves and asks for the 4th move:

```
SOOSYRVSTEP3 N
C1,R1
C2,R2
C3,R3
DONE
```

The engine answers with the coordinates of the 4th move and the number of offered 5th moves (from 1 to N):
`C4,R4 M`

3.18.4 SOOSYRVSTEP4

The manager sends the coordinates of four stones along with the actual number of offered 5th moves (specified in the previous cases):

```
SOOSYRVSTEP4 M
C1,R1
C2,R2
C3,R3
```

```
C4,R4
DONE
```
The engine answers with one of the following:

`SWAP`	if the engine decides to swap
C5a,R5a C5b,R5b C5c,R5c ...	M candidates for the 5th move separated by spaces (moves must not be symmetrical)

3.18.5 SOOSYRVSTEP5

If one of the player chosed to swap in the Case 4, the other player will be sent this variant. The manager sends the coordinates of four stones along with the actual number of offered 5th moves (specified in the previous cases):
```
SOOSYRVSTEP5 M
C1,R1
C2,R2
C3,R3
C4,R4
DONE
```
The engine answers with N candidates for the 5th move separated by spaces (moves must not be symmetrical)
```
C5a,R5a C5b,R5b C5c,R5c ...
```

3.18.6 SOOSYRVSTEP6

The manager sends the coordinates of four stones along with the candidates for 5th move:
```
SOOSYRVSTEP6 M
C1,R1
C2,R2
C3,R3
C4,R4
DONE
```
`C5a,R5a C5b,R5b C5c,R5c ...` The engine answers with two moves, first must be one of the offered 5th moves, while the other is the 6th move:
```
C5,R5 C6,R6
```
After the opening stage, the stones on the board will be treated as an opening for a traditional Gomocup match, and the manager will communicate with the engine using the classical Gomocup protocol in the rest of the match.

Note: Each step of the `SOOSYRV` command is considered to be a "turn", so the engine must obey the time controls (including `timeout_turn`).

3.19 TARAGUCHI

This command is used for Taraguchi opening rule [6]. It has two steps:

3.19.1 TARAGUCHISTEP1

The manager sends the coordinates of the first few stones along with the information how many 5th moves can be allowed:

```
TARAGUCHISTEP1 N
C1,R1
...
CN,RN
DONE
```

The engine answers with:

`C,R` the coordinates of the next move, 1st, 2nd, 3rd, 4th and 5th move must be within 1x1 (center), 3x3, 5x5, 7x7 and 9x9 central square, respectively. 6th move can be placed anywhere on the board

3.19.2 TARAGUCHISTEP2

If the opponent swapped colors in step 1 the manager will send this variant to the other player:

```
TARAGUCHISTEP2 N
C1,R1
...
CN,RN
DONE
```

The engine answers with one of the following:

`SWAP` if the engine decides to swap

`C,R` the coordinates of the next move, 2nd, 3rd, 4th and 5th move must be within 3x3, 5x5, 7x7 and 9x9 central square, respectively. 6th move can be placed anywhere on the board

3.19.3 TARAGUCHISTEP3

If one of the player chosed to swap in the Case 4, the other player will be sent this variant. The manager sends the coordinates of four stones along with the actual number of offered 5th moves (specified in the previous cases):

```
TARAGUCHISTEP3 N
```

```
C1,R1
C2,R2
C3,R3
C4,R4
DONE
```

The engine answers with:

`C5a,R5a C5b,R5b ...`	the coordinates of N candidates for the 5th move separated by spaces (moves must not be symmetrical)

3.19.4 TARAGUCHISTEP4

The manager sends the coordinates of four stones along with the candidates for 5th move:

```
TARAGUCHISTEP4 N
C1,R1
C2,R2
C3,R3
C4,R4
DONE
C5a,R5a C5b,R5b ...
```

The engine answers with two moves, first must be one of the offered 5th moves, while the other is the 6th move:

```
C5,R5 C6,R6
```

After the opening stage, the stones on the board will be treated as an opening for a traditional Gomocup match, and the manager will communicate with the engine using the classical Gomocup protocol in the rest of the match.

Note: Each step of the `TARAGUCHI` command is considered to be a "turn", so the engine must obey the time controls (including `timeout_turn`).

3.20 SAKATA

This command is used for SAKATA opening rule [6]. It has three steps:

3.20.1 SAKATASTEP1

The manager asks for the first three stones forming one of the 26 Renju openings (or any three stones for rules other than Renju):

```
SAKATASTEP1
DONE
```

The engine answers with three moves separated by spaces, for example

```
C1,R1 C2,R2 C3,R3
```

3.20.2 SAKATASTEP2

The manager sends the coordinates of the first three stones:
```
SAKATASTEP2
C1,R1
C2,R2
C3,R3
DONE
```
The engine answers with two moves:
`C4,R4 C5,R5` the coordinate of the 4th move (within 7x7 central square) and 5th move (within 9x9 central square)

3.20.3 SAKATASTEP3

The manager sends the coordinates of the first five stones:
```
SAKATASTEP3
C1,R1
C2,R2
C3,R3
C4,R4
C5,R5
DONE
```
The engine answers with one of the following:

`SWAP`	if the engine decides to swap
`C6,R6`	the coordinates of the 6th move if the engine decides to stay with white color

After the opening stage, the stones on the board will be treated as an opening for a traditional Gomocup match, and the manager will communicate with the engine using the classical Gomocup protocol in the rest of the match.

Note: Each step of the `SAKATA` command is considered to be a "turn", so the engine must obey the time controls (including `timeout_turn`).

3.21 TARANNIKOV

This command is used for SAKATA opening rule [6]. It has two steps:

3.21.1 TARANNIKOVSTEP1

The manager sends the coordinates of the first few stones along with the information how many 5th moves can be allowed:

```
TARANNIKOVSTEP1
C1,R1
...
CN,RN
DONE
```

The engine answers with:

`C,R` the coordinates of the next move, 1st, 2nd, 3rd, 4th and 5th move must be within 1x1 (center), 3x3, 5x5, 7x7 and 9x9 central square, respectively. 6th move can be placed anywhere on the board

3.21.2 TARANNIKOVSTEP2

If the opponent swapped colors in step 1 the manager will send this variant to the other player:

```
TARANNIKOVSTEP2
C1,R1
...
CN,RN
DONE
```

The engine answers with one of the following:

`SWAP` if the engine decides to swap

`C,R` the coordinates of the next move, 2nd, 3rd, 4th and 5th move must be within 3x3, 5x5, 7x7 and 9x9 central square, respectively. 6th move can be placed anywhere on the board

After the opening stage, the stones on the board will be treated as an opening for a traditional Gomocup match, and the manager will communicate with the engine using the classical Gomocup protocol in the rest of the match.

Note: Each step of the `TARANNIKOV` command is considered to be a ”turn”, so the engine must obey the time controls (including `timeout_turn`).

3.22 LIANHUAN

This command is used for LianHuan opening rule [6]. It has two steps:

3.22.1 LIANHUANSTEP1

The manager sends the coordinates of the first few stones:

```
LIANHUANSTEP1
```

```
C1,R1
...
CN,RN
DONE
```

The engine answers with one of the following:

`C,R`	the coordinates of the next move
`Ca,Ra Cb,Rb`	the coordinates of two next moves

3.22.2 LIANHUANSTEP2

If the opponent placed two moves in step 1 the player has the ability to swap:

```
LIANHUANSTEP2
C1,R1
C2,R2
C3,R3
DONE
```

The engine answers with one of the following:

`SWAP`	if the engine decides to swap
`C,R`	the coordinates of the next move
`Ca,Ra Cb,Rb`	the coordinates of two next moves

After the opening stage, the stones on the board will be treated as an opening for a traditional Gomocup match, and the manager will communicate with the engine using the classical Gomocup protocol in the rest of the match.

Note: Each step of the `LIANHUAN` command is considered to be a "turn", so the engine must obey the time controls (including `timeout_turn`).

4 YixinBoard Protocol

This protocol implements all commands from the base Gomocup protocol with modifications used by YixinBoard GUI [5].

The move format is assumed to be `[column],[row]`, often abbreviated by [C],[R].

The commands are case-insensitive (usually sent as lowercase).

Note: In order to use this protocol in AG, field "protocol" in the configuration file must be set to "yixinboard".

4.1 start [x] or start [C],[R]

Single command that behaves as `START [x]` or `RECTSTART [C] [R]` in the original Gomocup protocol. It can also be used as replacement to the `RESTART` command (i.e. to clear the board).

4.2 yxboard

This command is implemented exactly like in the specification [5].

4.3 Additional INFO keys

In addition to the keys described in the original Gomocup protocol, also the following will be recognized.

4.3.1 INFO rule [x]

Specifies the rules used for the game. A bitmask that can have several possible values:

$[x] = 0$	five or more wins - freestyle rule. Supported only for 20x20 or 15x15 board.
$[x] = 1$	exactly five wins - standard rule. Supported only for 15x15 board.
$[x] = 2$	renju rule.

Note: This setting cannot be changed during the search.

4.3.2 INFO usedatabase [x]

Controls the usage of move database. The allowed values are:

$[x] = 0$ engine will use database

$[x] = 1$ engine will not use database

4.3.3 INFO max_depth [x]

Sets maximum search depth.

Note: If $[x] < 0$ the search depth is unlimited.

Note: Since AG evaluates multiple nodes in parallel, it will stop as soon as the limit is reached but may exceed it by small amount.

4.3.4 INFO max_node [x]

Sets maximum number of nodes.

Note: If $[x] < 0$ the search depth is unlimited.

Note: Since AG evaluates multiple nodes in parallel, it will stop as soon as the limit is reached but may exceed it by small amount.

4.3.5 INFO time_increment [x]

Sets amount of time that is added after each move (see Fischer time controls [3]).

Note: Must not be negative.

4.3.6 INFO pondering [x]

Toggles the automatic pondering after each move. This command overrides the value of "auto_pondering" loaded from the configuration file. Allowed values are:

$[x] = 0$ automatic pondering is turned off

$[x] = 1$ automatic pondering is turned on

Note: This option does not start/stop the pondering by itself, only allows/forbids the engine to automatically start pondering.

Note: Pondering is never started automatically if `INFO analysis_mode` is set to 1.

4.3.7 INFO nbest_sym [x]

Sets the number of moves to be searched during `yxnbest` search.

4.3.8 INFO checkmate [x]

Controls the search type. Allowed values are:

$[x] = 0$ global search

$[x] = 1$ VCT1 search

$[x] = 2$ VCT2 search

4.3.9 INFO thread_num [x]

Sets maximum nuber of threads an engine can use. This command overrides the value of "thread_num" loaded from the configuration file. Must be greater than 0.

Note: If number of threads is changed during the search, AG would not adjust its number of threads until the search ends.

4.3.10 INFO hash_size [x]

Sets maximum amount of memory an engine can use.

4.3.11 INFO show_detail [x]

Toggles sending real-time information about search progress by the engine. Allowed values are:

$[x] = 0$ real-time info is not sent

$[x] = 1$ real-time info is being sent

4.3.12 Default values

Some parameters have default values so the engine works even if no `INFO` was sent:

- **rule** = 0 (freestyle rule)
- **usedatabase** = 0 (disabled)
- **max_depth** = 2147483647 (effectively no limit)

- **max_nodes** = 2147483647 (effectively no limit)
- **time_increment** = 0 (”sudden death” [4] type time controls)
- **pondering** = as loaded from configuration file
- **nbest_sym** = 0
- **checkmate** = 0 (global search)
- **thread_num** = as loaded from configuration file
- **hash_size** = 256 (256MB)
- **show_detail** = 0 (no real-time info)

4.4 yxhashclear

This command is implemented exactly like in the protocol description.

Note: AG does not have a hashtable like α-β engines. This command clears the search tree instead, which achieves similar result.

4.5 yxhashdump

This command is used to save the hashtable to a file specified by the next line. For example:

```
yxhashdump
somepath/file
```

Note: Currently not implemented in AG as it would require serialization of the entire search tree which is too complex for now.

4.6 yxhashload

This command is used to load the hashtable from a file specified by the next line. For example:

```
yxhashload
somepath/file
```

Note: Currently not implemented in AG as it would require serialization of the entire search tree which is too complex for now.

4.7 yxshowhashusage

The engine should respond with a `MESSAGE` with information about hash usage.

Note: Currently not implemented in AG as it does not have a hashtable like α-β engines.

References

[1] Petr Laštovička, New Gomocup Protocol

[2] Kai Sun, Tianyi Hao, [Updated on 4/10] Experimental Tournament

[3] Fischer time control - Wikipedia

[4] Time control - Wikipedia

[5] YixinBoard protocol

[6] Renju opening rules

[7] Gomoku opening rules