Aim to achieve: find the string's status.

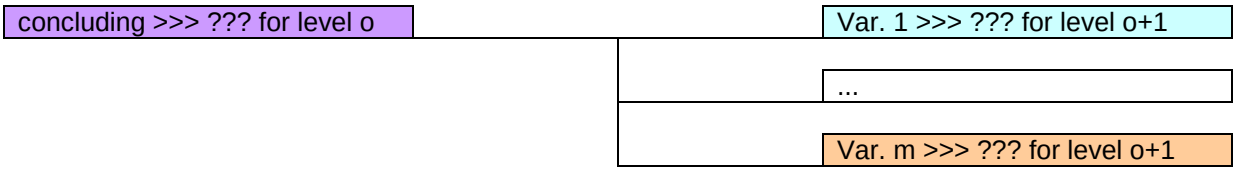
	Opponent cannot force capture of the string (= A)	Opponent can force capture of the string (= B)
Player can force uncapture of the string (= A)	String remains on the board >>> "uncapturable"	--- contradiction ---
Player cannot force uncapture of the string(= B)	--- contradiction ---	String is taken off the board >>> see table 2

	Opponent cannot force no successor on at least one of the string's primary point(s) (=B2)	Opponent can force no successor on each of the string's primary point(s), but cannot force no permanent stone in a special area outside the string's primary point(s) (=B31)	Opponent can force no successor on each of the string's primary point(s), and can force no permanent stone in a special area outside the string's primary point(s) (=B32)
Player can force at least one successor on the string's primary point(s) (= B2)	There is at least one successor on the string's primary point(s) >>> "capturable-1"	--- contradiction --- (player: at least one successor; opponent: no successor)	--- contradiction --- (player: at least one successor; opponent: no successor)
Player cannot force at least one successor on the string's primary points, but can force at least one permanent stone in a special area outside the string's primary points (= B31)	--- contradiction --- (player: no successor; opponent: at least one successor)	There is no stone on the string's primary points, and there is at least one permanent stone in a special area outside the string's primary points >>> "capturable-2\1"	--- contradiction --- (player: at least one permanent stone in ...; opponent: no permanent stone in ...)
Player cannot force at least one successor on the string's primary points, and cannot force at least one permanent stone in a special area outside the string's primary points (= B32)	--- contradiction --- (player: no successor; opponent: at least one successor)	--- contradiction --- (player: no permanent stone in ...; opponent: at least one permanent stone in ...)	There is no stone on the string's primary points, and there is no permanent stone in a special area outside the string's primary points >>> "not alive"

Cassandra's evaluation

Tables for the opponent's choice:

Variation tree



Decision matrix "Opponent" (valid for the concluding result of each pair of variations)

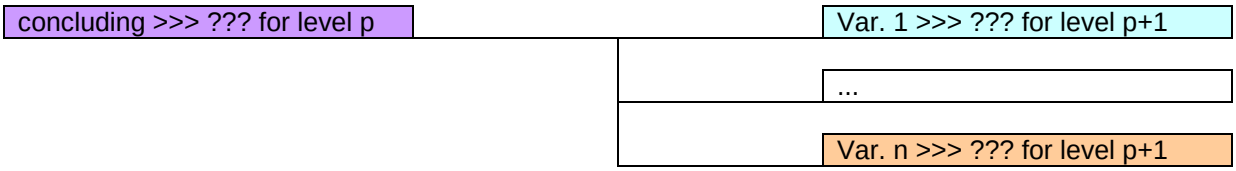
	A	B2	B31	B32
A	A	B2	B31	B32
B2	B2	B2	B31	B32
B31	B31	B31	B31	B32
B32	B32	B32	B32	B32

For more than 2 variations we can shorten the decision procedure with

Concluding result = MINIMUM (var. 1, var. 2, ... , var m) with
"B32" < "B31" < "B2" < "A".

Table for the player's choice:

Variation tree



Decision matrix "Player" (valid for the concluding result of each pair of variations)

	A	B2	B31	B32
A	A	A	A	A
B2	A	B2	B2	B2
B31	A	B2	B31	B31
B32	A	B2	B31	B32

For more than 2 variations we can shorten the decision procedure with

Concluding result = MAXIMUM (var. 1, var. 2, ... , var n) with
"A" > "B2" > "B31" > "B32".

Evaluation:

The variation tree begins with decision matrix "Opponent" (the opponent moves first) at the "root" and continues (in the direction toward "leaves") at any further branching point with decision matrix "Player", if it is the player's turn, or with decision matrix "Opponent", if it is the opponent's turn.

Combining the leaves' results starts near the leaves and ends at the root of the variation tree.

The concluding result at the tree's root is clear cut and depends only on the statuses found for the tree's leaves.

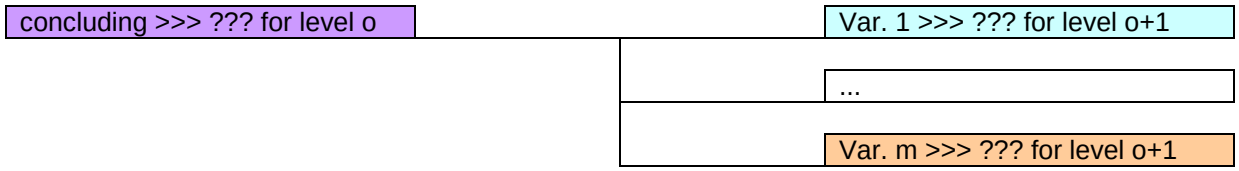
This is true for every single string, independent of its status found.

J2003's evaluation

Step 1: Identify "uncapturable"

Tables for the opponent's choice:

Variation tree



Decision matrix "Opponent" (valid for the concluding result of each pair of variations)

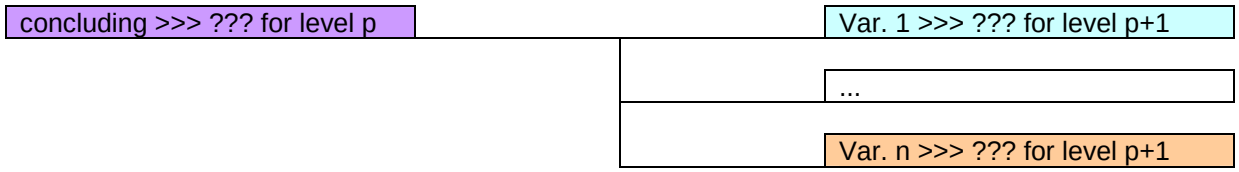
	A	B2	B31	B32
A	A	B2	B31	B32
B2	B2	B2	B31	B32
B31	B31	B31	B31	B32
B32	B32	B32	B32	B32

For more than 2 variations we can shorten the decision procedure with

Concluding result = MINIMUM (var. 1, var. 2, ... , var m) with
"B32" < "B31" < "B2" < "A".

Table for the player's choice:

Variation tree



Decision matrix "Player" (valid for the concluding result of each pair of variations)

	A	B2	B31	B32
A	A	A	A	A
B2	A	B2	B2	B2
B31	A	B2	B31	B31
B32	A	B2	B31	B32

For more than 2 variations we can shorten the decision procedure with

Concluding result = MAXIMUM (var. 1, var. 2, ... , var n) with
"A" > "B2" > "B31" > "B32".

Evaluation:

The variation tree begins with decision matrix "Opponent" (the opponent moves first) and at continues at any further branching point with decision matrix "Player", if it is the player's turn, or with decision matrix "Opponent", if it is the opponent's turn.

As a result of step 1 we have identified all strings, which are "uncapturable".

Step 2: Identify "capturable-1"

To simplify the text, from here on we will do without the graphic of the variation tree and without the table's subscript for more than 2 variations.

We have to take into account that no concluding result "A" is possible any more (we have found all "A" in step 1). Because we do NOT want to cut all the variation tree's "A"-leaves in the very beginning (despite these leaves are not forced) for the sake of further evaluation of the remaining strings, we have to be very careful with the decision matrixes.

Concerning the decision matrixes, we have to pay attention not to hit a cell, which content might be "undefined".

Decision matrix "Opponent"

	A	B2	B31	B32
A	undefined	B2	B31	B32
B2	B2	B2	B31	B32
B31	B31	B31	B31	B32
B32	B32	B32	B32	B32

"A" results in "undefined" now. We have got a leaf with a string that has not become captured, despite it is not uncapturable.
Because this result cannot be "forced" by the opponent, there must be another variation, which shows the "real force" of the opponent.

Decision matrix "Player"

	A	B2	B31	B32
A	undefined	undefined	undefined	undefined
B2	undefined	B2	B2	B2
B31	undefined	B2	B31	B31
B32	undefined	B2	B31	B32

"A" is no "forced" result any more, so the "Player" has to review his decision table, which had been designed for step 1.

A concluding result "A" is not preferable any more for the player, because this would be equivalent to "undefined". So we have to assess that "A" has become "smaller" than "B32" ("B32" > "A").

"A"-"A" is as above for the opponent.

Thus we get new decision tables valid for step 2.

Decision matrix "Opponent"

	A	B2	B31	B32
A	irrelevant	B2	B31	B32
B2	B2	B2	B31	B32
B31	B31	B31	B31	B32
B32	B32	B32	B32	B32

Decision matrix "Player"

	A	B2	B31	B32
A	irrelevant	B2	B31	B32
B2	B2	B2	B2	B2
B31	B31	B2	B31	B31
B32	B32	B2	B31	B32

We can see that no status "A" will survive.

To simplify evaluation we could cut all "A"-leaves now.

Thus we get new (2) decision tables valid for step 2.

Decision matrix "Opponent"

	B2	B31	B32
B2	B2	B31	B32
B31	B31	B31	B32
B32	B32	B32	B32

Decision matrix "Player"

	B2	B31	B32
B2	B2	B2	B2
B31	B2	B31	B31
B32	B2	B31	B32

A comment on the definition of "capturable-1" in J2003:

It would be more clear cut (and without side-effects) to have it

"After having captured the string, the opponent cannot force no permanent stone of the player at local-1 (= the string's primary points)".

J2003's definition

"The opponent cannot force both capture of the string and no permanent stone of the player at local-1 (= the string's primary points)".

has the disadvantage that the opponent is not obliged to do anything he could "force" (here to capture the string). This does not do any harm here, but will become relevant with capturable-2 later.

Here "the opponent allows the string to remain uncaptured" means "there will be at least one permanent stone on the string's primary points", thus the string is capturable-1.

Step 3: Identify "capturable-2"

We have to take into account that no concluding result "B2" is possible any more (we have found all "B2" in step 2). Because we do NOT want to cut all the variation tree's "B2"-leaves in the very beginning (despite these leaves are not forced) for the sake of further evaluation of the remaining strings, we have to be very careful with the decision matrixes.

Concerning the decision matrixes, we have to pay attention not to hit a cell, which content might be "undefined".

Decision matrix "Opponent"

	B2	B31	B32
B2	undefined	B31	B32
B31	B31	B31	B32
B32	B32	B32	B32

"B2" results in "undefined" now. We have got a leaf with a string that will get a successor on at least one of its primary points, despite it is not capturable-1.
Because this result cannot be "forced" by the opponent, there must be another variation, which shows the "real force" of the opponent.

Decision matrix "Player"

	B2	B31	B32
B2	undefined	undefined	undefined
B31	undefined	B31	B31
B32	undefined	B31	B32

"B2" is no "forced" result any more, so the "Player" has to review his decision table, which had been designed for step 2.

A concluding result "B2" is not preferable any more for the player, because this would be equivalent to "undefined". So we have to assess that "B2" has become "smaller" than "B32" ("B32" > "B2").

"B2"-"B2" is as above for the opponent.

Thus we get new decision tables valid for step 3.

Decision matrix "Opponent"

	B2	B31	B32
B2	irrelevant	B31	B32
B31	B31	B31	B32
B32	B32	B32	B32

Decision matrix "Player"

	B2	B31	B32
B2	irrelevant	B31	B32
B31	B31	B31	B31
B32	B32	B31	B32

We can see that no status "B2" will survive.

To simplify evaluation we could cut all "B2"-leaves now.

There is NO leaf left, which has a permanent stone on any of the string's primary points.

Thus we get new (2) decision tables valid for step 3.

Decision matrix "Opponent"

	B31	B32
B31	B31	B32
B32	B32	B32

Decision matrix "Player"

	B31	B32
B31	B31	B31
B32	B31	B32

A comment on the definition of "capturable-2" in J2003:

It would be more clear cut (and without side-effects) to have it

"After having captured the string, the opponent can force no permanent stone of the player at the string's primary points (= local-1)", but cannot force no permanent stone of the player in local-2\1 (=a special area outside the string's primary point(s))."

J2003's definition

"The opponent cannot force both capture of the string and no permanent stone of the player at local-2."

has the disadvantage that the opponent is not obliged to do anything he could "force" (here to capture the string or to have no permanent stone of the player at local-1).

This makes it "attractive" for the opponent to allow a permanent stone on local-1, if he has the chance to do so. This could happen especially after a replacement of "local-2" by "local-2\1".

On the other side, the player is NOT obliged to place a stone at local-1, which is in "danger" to become permanent. He can be satisfied to have got at least one permanent stone on local-2\1. So even the above mentioned replacement would not to do any harm, but could be a source of misunderstanding and confusion.

Including local-1 in the "special area" being relevant for "capturable-2" has the one and only advantage to probably prevent this misunderstanding and confusion (if one wants to keep the original J2003's definitions).

This has NOTHING to do with the fact that there is NO capturable-2 string, which has any permanent stone in local-1 after capture.