The paper “Deep Blue” describes the techniques used by IBM for building a world-class chess engine, culminating in Deep Blue being the first computer program to beat reigning world champion Gary Kasparov in 1997, on a score of 3.5 to 2.5.

In this document, the authors explain how Deep Blue II differs from previous versions as Deep Thought I and II and Deep Blue I. From early on, the system was using multiprocessing and a powerful search software. Deep Blue II was a big improvement over Deep Blue I, which failed to beat Kasparov the very previous year in 1996. IBM’s team worked hard on adapting their approach and especially the program’s evaluation function and the machine’s hardware.

The machine’s power relied on a hybrid approach to search, where hardware search provided a huge computational power, and the software search provided the flexibility of searching deeper into the directions of the most interesting moves.

Deep Blue’s system was based on a master-slaves architecture of nodes utilizing specialized chess chips, where one master node did the initial search iterations, and distributed the workload to the slave chips as it got overloaded. Having such a huge search capacity for a single task was a bit novel at that time, so Deep Blue’s team had to do clever design decisions to be utilizing this power to its best.

The chips themselves had three main parts: a move generator, an evaluation function, and a search control. The move generator was a pretty simple finite state machine, but a clever trick allowed it to generate all moves simultaneously, which helped reduce latency.
The search control was done using null-window alpha-beta search. A disadvantage of the hardware search was that it could not benefit from transposition tables.
The evaluation function was where Deep Blue II really improved a lot over its predecessors. It used a fast evaluation to compute easy things like piece placement; and slow evaluation for more complex and deep concepts such as square control, king safety or pins.

IBM’s team designed the search software to use standard techniques in game theory AI like quiescence search, iterative deepening and transposition tables.
The big novelty of this paper is that it introduced the concept of “dual credit with delayed extensions”. The basic idea is that it is best to explore first the moves that are the most promising or are the most forced. And Deep Blue’s team did this so that the software would only extend a sequence of moves if it has determined that it has enough importance (credit). Credit of a move sequence is influenced by singularity, the existence of threats, influence of preceding moves and other domain dependent mechanisms as check evasion or pushing passed pawns.

Deep Blue also introduced a pruning technique the team called “No progress”: if a move is good for a given side, it is best to play it earlier rather than later. This goes in the same direction as the credit system as it encourages the program to explore the most promising moves first.

There is no big discussion about results in the paper as it is quite difficult to quantify the efficiency of a engine for a game of which we do not have perfect understanding yet. The paper mentions very briefly that Deep Blue had an efficiency of about 8% in tactical positions, and of about 12% in quiet positions, but there is no mention of what this efficiency actually means.
The fact that the machine beat Kasparov in a match is somehow a result, but there were long debates in the chess community that IBM probably cheated in this match as they were so eager to get the publicity of winning against a world champion. One sign this might be the truth is that the machine was dismantled immediately after this match, when many thought that there might have been human help for the machine.