Constraints in Line Drawings

Here's the basic idea of what's going on: two dimensional drawings of blocks world (like this one) are made up of lines. These lines might represent convex, concave, or boundary edges in the real, three-dimensional world --- and we humans can effortlessly figure out what type each line is. Part of the problem of getting a computer to see the way a human does is to correctly interpret the lines in a drawing as the appropriate kind of edges in the real world. It would also be really cool if a computer could see why images like this one are ambiguous/confusing for humans to interpret.

How do you do it? Lines in a drawing can meet up in a few different ways; places where lines meet up are called junctions. Not all junctions with edges labelled as convex/concave/boundary are physically possible, however. For example, I believe you can't make a fork junction with two concave edges and one convex edge (try it).

So what you do is you turn the problem into a search: iterate over each junction in the drawing and tentatively interpret it as one of the physically allowed junctions. This makes each of the lines in the junction a convex, concave, or boundary line. And when lines are assigned convex/concave/boundary interpretations, they affect the possible values of their neighbors. Here's an example:

Originally the line drawing might look like this:
http://aurellem.org/dl/draw_000.png
(The four junctions are drawn in black)

If you tentatively pick this interpretation of the junction, it affects how you can interpret the other junctions
http://aurellem.org/dl/draw_001.png

This partial arrangement is allowed, because the plus signs match up.
http://aurellem.org/dl/draw_002.png

In contrast, this arrangement is not allowed --- the arrow conflicts with the plus sign.
http://aurellem.org/dl/draw_003.png

Finally, the big-picture result is that because you've found the right way to describe the problem (junctions, concave/convex/boundary lines, and a dictionary of physically possible junctions), your search is highly constrained: you don't have to brute-force search; you can just backtrack immediately if you create an interpretation that's inconsistent.