1/4

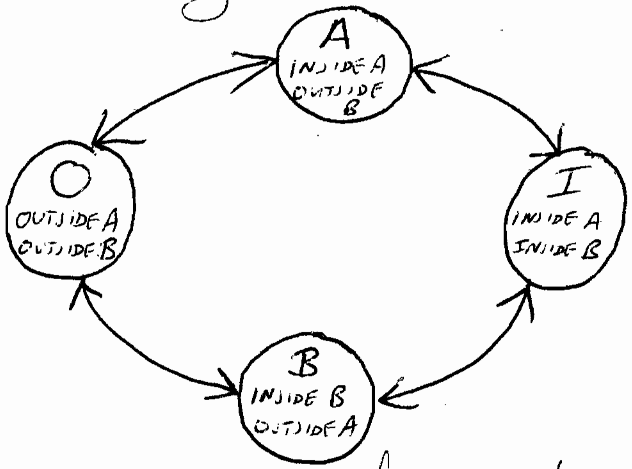
P8Q4
NAD

Solution Notes for Advanced Graphics 2002

(a) An object built using CSG can be represented as a binary tree in which the internal nodes are operators (union, intersection, difference) and the leaf nodes are primitive objects.

Given two ordered lists of intersection ~~nodes~~ points for the two we sort the two lists into a single list, ~~keeping track~~ in order from closest to the eye to farthest from the eye. We remember which object each point comes from and whether it is changing from outside the object to inside or vice-versa.

We then proceed through the list, at each point we change state:



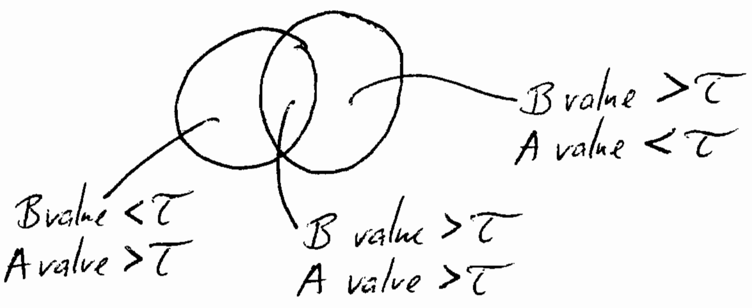
~~The~~ We pass a list of intersection points up to the node's parent. For union this consists of all points in the list which cause a transition into or out of state O. For intersection, state I. For A-B, state A. For B-A, state B.

Do this for every node in the tree. At the root, the first intersection point in the list is the one you want.

(b) To achieve CSG union, use the max operator.
For intersection, the min operator.

These work because implicit surfaces assume that anything above a threshold value, τ, is inside the object and anything below the threshold is outside.

So



$\min(A, B) \geq \tau$ only in the intersection
$\max(A, B) > \tau$ across the whole union

CSG difference, e.g. $A - B$, could be implemented as:

$$\min(A, \tau - B)$$

——————— // ———————

(c) Radiosity is an algorithm which calculates the diffuse illumination of every polygon in the scene. It takes into account the diffuse inter-reflections between surfaces.

① convert your 3D model into a set of polygonal patches. The patches must be sufficiently small that they can capture the detail of the diffuse shading

② for each patch specify its emissivity, E_i, (the amount of light it emits as a light source), and its reflectivity ρ_i.

③ calculate the form factors: $f_{i\to j}$, the proportion of light leaving patch i that hits patch j

④ calculate the radiosities, B_i, by solving the simultaneous equation system:

$$B_i = \left(\rho_i \sum_{j \neq i} B_j f_{i\to j}\right) + E_i$$

⑤ Display the polygon patches using the B_i values.

MORE DETAIL:

④ can be handled by something like Gauss-Seidel iteation

③ is difficult to do & can be tackled by the hemicube method.

Marking scheme & notes for Advanced Graphics

(a) tests "Other ways to create more complex geometry", as does (b) — although (b) stretches the students
(c) tests "Lighting"

Marking scheme – preliminary

(a) correct structure of CSG tree	1
lists consist of ALL intersection points	1
correct way to combine two lists	3
correct identification of "first intersection point"	1
(b) correct identification of <u>min</u> and <u>max</u>	1
explanation of why they work	2
correct mechanism for difference	1
(c) split into patches	1
identify ρ_i and E_i	1
calculate $f_{i \to j}$	1
– how to do this (in outline)	1
calculate B_i	1
– correct equation	1
– how to solve simultaneous eq$\stackrel{\text{n}}{=}$	1
overall understanding including any extra details	2
presentation	1

[illegible]