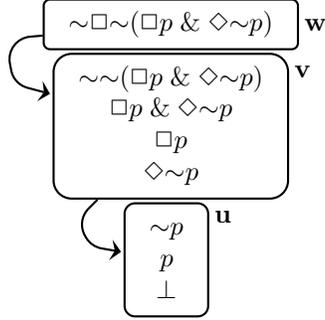


In a K-tree it is important that a world-subproof be placed *within* the subproof corresponding to the world that “spawned” it, i.e., the world to which it’s accessible. This may mean placing a world subproof within a world subproof. Consider:



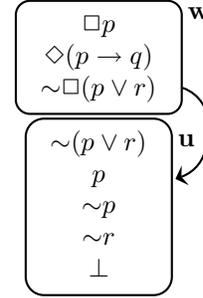
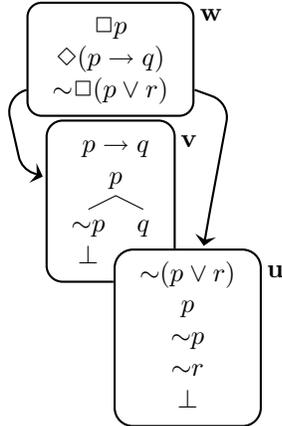
Here we seem to face a quandary. The world subproof for **u** cannot be placed inside the world subproof for **v**, since this would make it impossible to apply (\Box Out) to get “ p ” into world **u**, as “ $\Box p$ ” is only found in world **w**. However, we cannot finish the proof by cases within world **v** prior to the introduction of world **u**, and hence we cannot finish its world subproof prior to starting that one.

The solution to this quandary is to note that since “siblings” in a K-tree (two worlds spawned by the same world on the same branch) are never accessible to one another, the contradiction in the final sibling never utilizes results of earlier siblings. The older siblings can simply be *removed* from the tree and the tree will close anyway.

This converts to:

- | | | | |
|-----|----------|--|----------------------------|
| 1. | w | $\sim\Box\sim(\Box p \ \& \ \Diamond\sim p)$ | |
| 2. | | $\Diamond\sim\sim(\Box p \ \& \ \Diamond\sim p)$ | 1 ($\sim\Box$) |
| 3. | v | $\Box, \sim\sim(\Box p \ \& \ \Diamond\sim p)$ | |
| 4. | | $\Box p \ \& \ \Diamond\sim p$ | 3 (DN) |
| 5. | | $\Box p$ | 4 ($\&$ Out) |
| 6. | | $\Diamond\sim p$ | 4 ($\&$ Out) |
| 7. | u | $\Box, \sim p$ | |
| 8. | | p | 5 (\Box Out) |
| 9. | | \perp | 7,8 (\perp In) |
| 10. | | \perp | 6,7-9 ($\Diamond\perp$) |
| 11. | | \perp | 2,3-10 ($\Diamond\perp$) |
| 12. | | $\Box\sim(\Box p \ \& \ \Diamond\sim p)$ | 1-11 (IP) |

Care must be taken, however, if a world is introduced in a tree that is spawned by a world not immediately above it on the branch, such as in this example:

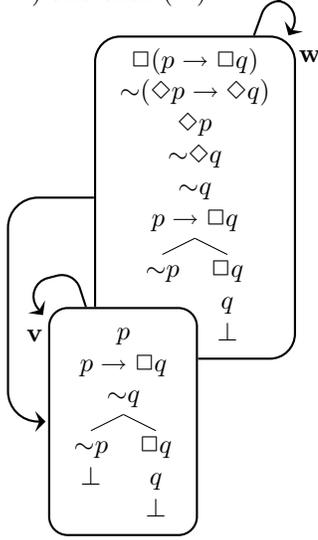


Garson calls such “removable” worlds “failed siblings”. Conversion:

- | | | | |
|-----|----------|-----------------------------|----------------------------|
| 1. | w | $\Box p$ | |
| 2. | | $\Diamond(p \rightarrow q)$ | |
| 3. | | $\sim\Box(p \vee r)$ | |
| 4. | | $\Diamond\sim(p \vee r)$ | 3 ($\sim\Box$) |
| 5. | u | $\Box, \sim(p \vee r)$ | |
| 6. | | p | 1 (\Box Out) |
| 7. | | $\sim p \ \& \ \sim r$ | 5 (DM) |
| 8. | | $\sim p$ | 7 ($\&$ Out) |
| 9. | | $\sim r$ | 7 ($\&$ Out) |
| 10. | | \perp | 6,8 (\perp In) |
| 11. | | \perp | 4,5-10 ($\Diamond\perp$) |
| 12. | | $\Box(p \vee r)$ | 3-11 (IP) |

Converting M-trees is only marginally more difficult than converting K-trees. Wherever the tree contains an inference from $\Box A$ to A within the same world (due to reflexivity), we merely insert the corresponding instance of axiom (M), $\Box A \rightarrow A$ into the proof and then perform the (MP) (or cite the derived

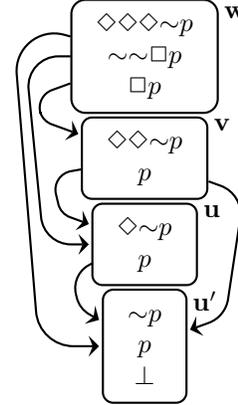
rule, (M), which is shorthand for this (MP)). Where we move from $\sim\Diamond A$ to $\sim A$ within the same world, we cite $(\sim\Diamond)$ and then (M).



Yields:

1.	w	$\Box(p \rightarrow \Box q)$	
2.		$\sim(\Diamond p \rightarrow \Diamond q)$	
3.		$\Diamond p$	2 ($\rightarrow F$)
4.		$\sim\Diamond q$	2 ($\rightarrow F$)
5.		$\Box\sim q$	5 ($\sim\Diamond$)
6.		$\sim q$	5 (M)
7.		$p \rightarrow \Box q$	1 (M)
8.		$\sim p \vee \Box q$	7 (Impl)
9.		$\sim p$	
10.	v	\Box, p	
11.		$p \rightarrow \Box q$	1 ($\Box Out$)
12.		$\sim q$	5 ($\Box Out$)
13.		$\sim p \vee \Box q$	11 (Impl)
14.		$\sim p$	
15.		\perp	10,14 ($\perp In$)
16.		$\Box q$	
17.		q	16 (M)
18.		\perp	12,17 ($\perp In$)
19.		\perp	13,14–15,16–18 ($\vee Out$)
20.		\perp	3,10–19 ($\Diamond \perp$)
21.		$\Box q$	
22.		q	21 (M)
23.		\perp	6,22 ($\perp In$)
24.		\perp	8,9–20,21–23 ($\vee Out$)
25.		$\Diamond p \rightarrow \Diamond q$	2–24 (IP)

Conversions of 4-trees is not much more complicated. In order to apply the rule to a statement of the form $\Box A$ in order to bring A into another world that is accessible to the world of $\Box A$ because of transitivity, one needs only use (4) to add one (—or more, in the case of multiple “jumps”—) \Box 's, so that one gets $\Box A$ in the intermediate world subproofs, so that ($\Box Out$) yields A in the final subproof. Consider:



Becomes:

1.	w	$\Diamond\Diamond\Diamond\sim p$	
2.		$\sim\sim\Box p$	
3.		$\Box p$	2 (DN)
4.		$\Box\Box p$	3 (4)
5.		$\Box\Box\Box p$	4 (4)
6.	v	$\Box, \Diamond\Diamond\sim p$	
7.		p	3 ($\Box Out$)
8.		$\Box p$	4 ($\Box Out$)
9.		$\Box\Box p$	5 ($\Box Out$)
10.	u	$\Box, \Diamond\sim p$	
11.		p	8 ($\Box Out$)
12.		$\Box p$	9 ($\Box Out$)
13.	u'	$\Box, \sim p$	
14.		p	12 ($\Box Out$)
15.		\perp	13,14 ($\perp In$)
16.		\perp	10,13–15 ($\Diamond \perp$)
17.		\perp	6,10–16 ($\Diamond \perp$)
18.		\perp	1,6–17 ($\Diamond \perp$)
19.		$\sim\Box p$	2–18 (IP)

To convert an S4 tree, simply combine the methods for converting (K)4-trees/ with those for converting M-trees.

For converting (K)B-trees and 5-trees, note that, given the definitions of ‘ \diamond ’ and ‘ \vee ’, theorems (B’) and (5’) are merely notational variants of these:

$$\begin{aligned} (\vee B) \quad & \Box \sim \Box A \vee A \\ (\vee 5) \quad & \Box \sim \Box A \vee \Box A \end{aligned}$$

The process for converting these trees is more complicated than those we’ve seen so far.

Because tree rules for (K)B-trees can sometimes be applied *earlier* on the same branch, we cannot simply convert the statements on the tree into steps of the proof in order, as it is impossible in a proof to justify an earlier step by means of a later one.

To overcome this, it is necessary to alter the tree before converting it. In particular, we change the tree so that it becomes a K-tree, only with instances of axiom schema ($\vee B$) added as additional lines of the tree.

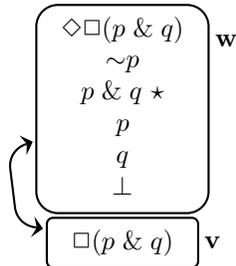
In particular, at those places where in the original tree we wrote A into a given world box \mathbf{w} because of the presence of $\Box A$ in some world box \mathbf{v} below it, we put the instance of ($\vee B$) of the form $\Box \sim \Box A \vee A$ in its place in the modified tree.

Applying the tree rules to it, it will branch, and you’ll get the following results:

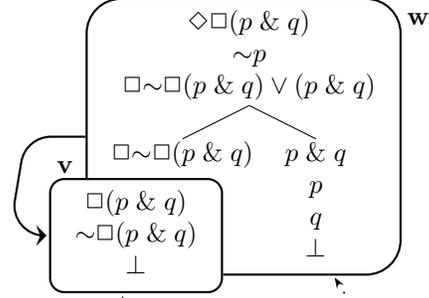
On *the right branch* we’ll have A in world \mathbf{w} , and hence have precisely the same statements as \mathbf{w} did on the original tree, and that branch will close in the same way the original tree did.

On *the left branch*, we’ll have $\Box \sim \Box A$ in \mathbf{w} . When we apply the rule that spawns world \mathbf{v} on this branch underneath it, we’ll infer $\sim \Box A$ according to the normal K-tree for necessity statements. Since world \mathbf{v} also has $\Box A$ in it (from which we applied the symmetry rule in the original tree), we’ll get a contradiction there, and this branch will close as well.

This is best seen by example. Consider this KB-tree:

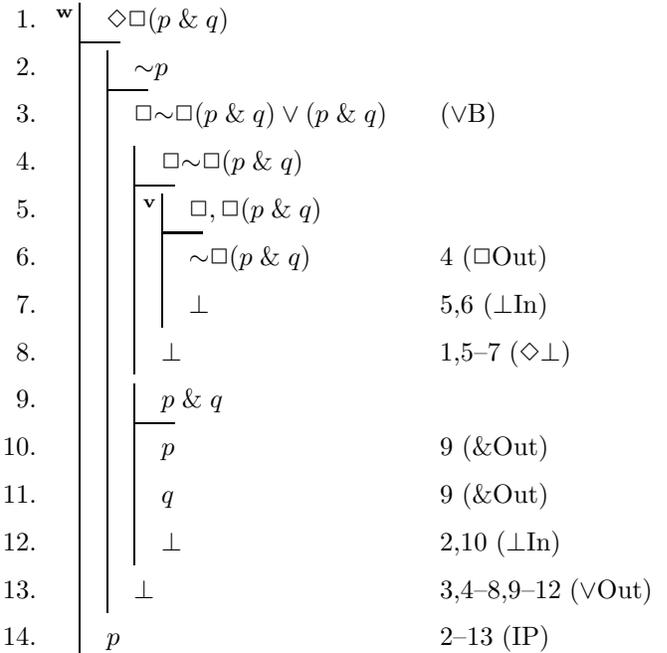


The line here marked by ‘ \star ’ comes from world \mathbf{v} below. In our modified tree we replace this with an instance of ($\vee B$), and branch off of it.



The left disjunct of ($\vee B$) puts the contradictory of the necessity statement into world \mathbf{v} . This branch closes just like the original world \mathbf{w} .

The tree can now be converted just like a K-tree, except appealing to ($\vee B$) where it was inserted:



A B-tree can be converted by simply combining the methods for KB-trees with those for M-trees.

The process for converting 5-trees is similar. We modify the original 5-tree into a K-tree in which instances of ($\vee 5$) are inserted.

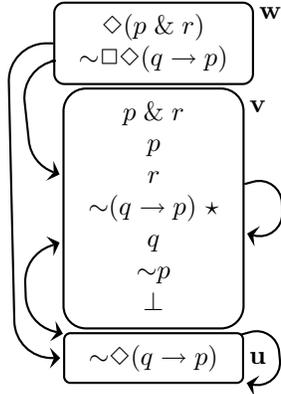
Suppose that in a 5-tree, $\mathbf{wR}\mathbf{u}$ and $\mathbf{wR}\mathbf{v}$, and in virtue of this fact we concluded that $\mathbf{uR}\mathbf{v}$ and hence moved from a statement of the form $\Box A$ in \mathbf{u} to A in \mathbf{v} . Here we could call \mathbf{w} the “parent” world, \mathbf{u} the “FROM-child” and \mathbf{v} the “TO-child”.

We then modify the tree by inserting the appropriate instance of ($\vee 5$) into the parent world, and branch off of it.

On the *left branch*, we spawn the FROM-child, and the left disjunct of (∇5) will insert into it the contradictory of the necessity statement there, closing the branch.

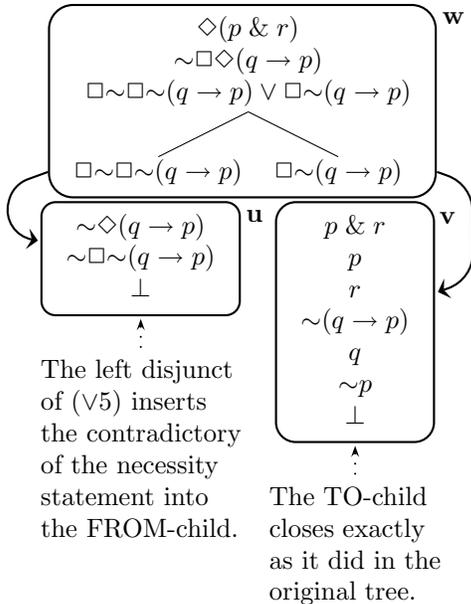
On the *right branch*, we spawn the TO-child, and the right disjunct of (∇5) will insert A into it, and the branch will close as it did in the original tree.

Again, an example helps. Consider this K5-tree:

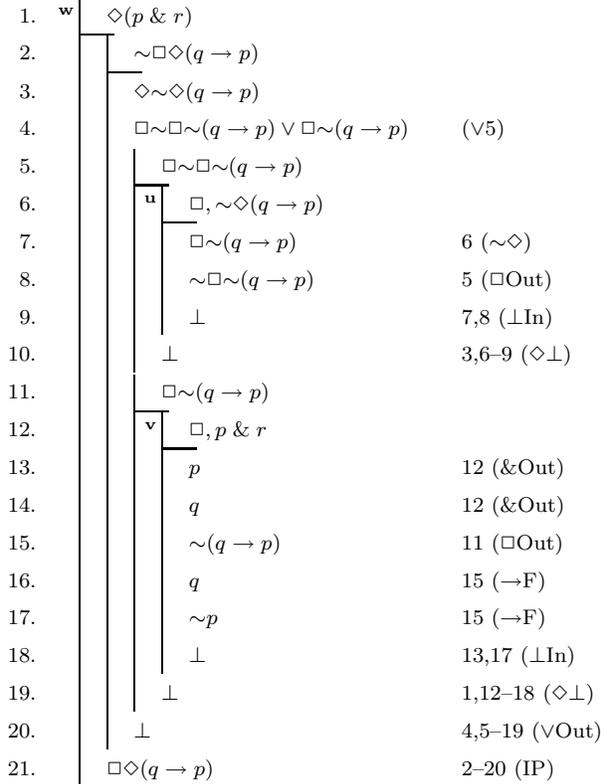


Here, the starred step in v comes from world u below. Here, w is the parent, u is the FROM-child and v is the TO-child.

In the converted tree, the instance of (∇5) for the starred line is placed in the parent world and then branched. The FROM-world is put on the left branch and the TO-world on the right.



We can now convert the tree.



Converting B-trees or 5-trees in which the symmetric or Euclidian nature of the accessibility relation is made use of more than once can be quite difficult. It usually helps to make several modified trees, each eliminating one such use at a time, before converting. More details and suggestions can be found in the book.

S5-trees, as you've no doubt guessed, can be converted by combining this method with that for M-trees. Those relying on the symmetry or transitivity of the accessibility relation in S5-models may be shortened by using the methods for 4-trees and B-trees, since (4) and (B) are theorem schemata of S5 as well. (See the handouts, page 10.)

We've now seen that:

- If the K-tree for $A_1, A_2, \dots, A_n, \sim B$ closes then $A_1, A_2, \dots, A_n \vdash_K B$.
- If the M-tree for $A_1, A_2, \dots, A_n, \sim B$ closes then $A_1, A_2, \dots, A_n \vdash_T B$.
- And so on for systems K4, S4, KB, B, K5 and S5.

More mathematically rigorous "proofs" of these metatheoretic results are given in the book.