Handout #7 – Predicate Logic Trees

Predicate trees: Decomposition Rules

Negated Existential Decomposition (¬∃D)	**Negated Universal Decomposition (¬∀D)**
¬(∃x)**P**✓ (∀x)¬**P**	¬(∀x)**P**✓ (∃x)¬**P**
Existential Decomposition (∃D)	**Universal Decomposition (∀D)**
(∃x)**P**✓ **P**(a/x) where a is an individual constant (name) that does not previously occur in the branch	(∀x)**P** **P**(a/x) where a is any individual constant (name)

Example #1

1	¬(∀x)Px✓	P
2	¬(∃y)Ryy✓	P
3	(∃x)¬Px	1¬∀D
4	(∀y)¬Ryy	2¬∃D

Example #2

1	¬(∀x)Px✓	P
2	¬(∃y)Ryy✓	P
3	(∃x)¬Px✓	1¬∀D
4	(∀y)¬Ryy	2¬∃D
5	¬Pa	3∃D
6	¬Raa	4∀D
7	¬Rbb	4∀D

Example #3

1	(∃x)Px	P
2	(∃x)Qx	P
3	Pa	1∃D
4	Qa	2∃D — **NO!**

Predicate Trees: Strategies & Terminology

In PL, a *completed open branch* is defined as a fully decomposed branch that is not closed. For trees in RL, a *new* definition is required since it branches involving universally quantified propositions cannot be fully decomposed.

Completed Open Branch	A branch is a *completed open branch* if and only if (1) all complex propositions that can be fully decomposed are decomposed, (2) for all universally quantified propositions (∀x)**P** occurring in the branch, there is a substitution instance **P**(a/x) for each constant in that branch, and (3) the branch is not a closed branch.

Example #1

1	Pa	P
2	Rb	P
3	Lcc	P
4	(∀x)Px	P
5	Pa	4∀D
6	Pb	4∀D
7	Pc	4∀D
	0	

At line 4, it is important to note clause (2) of the definition of a *completed open branch*. What this says is that "for all universally quantified propositions (∀x)**P** occurring in the branch, there is a subtitution instance **P**(a/x) for each constant in that branch." Note that there are *three* constants / names in the above branch containing (∀x)Px (a b, and c), thus, (∀x)Px should be decomposed (using ∀D) using each of these as substitution instances.

Example #2

1	(∃x)Px	P
2	(∀x)¬Px	P
3	¬Pa	2∀D
4	Pb	1∃D
	0 — NO!	

In the above example, the universally quantified proposition at line (2) is not decomposed for each constant in the branch.

	Strategic Rules for Decomposing Predicate Truth-Trees
1	Use no more rules than needed.
2	**Decompose negated quantified expressions and existentially quantified expressions first.**
3	Use rules that *close* branches.
4	Use *stacking rules* before *branching rules*.
5	**When decomposing universally quantified propositions, it is a good idea to use constants that *already occur* in the branch.**
6	Decompose more complex propositions before simpler propositions.

Practice

1. $(\forall x)(Px \rightarrow Rx), Pa, \neg Rb$
2. $(\exists x)\neg(Px \lor Rx), (\exists y)\neg(Py \lor Ry), (\forall z)(Pz \lor Rz)$
3. $\neg(\forall x)Px \rightarrow (\exists z)Pz, \neg(\exists x)\neg Px$
4. $(\forall x)\neg Mx, (\forall x)Px, Pa \rightarrow (\exists x)Mx$
5. $Laa, (\exists x)(\exists y)Zxy, (\forall x)(\forall y)\neg Zxy$
6.* $(\exists x)(\exists y)(Pxy \land Lxy), \neg(\exists x)(\exists y)Pxy \lor \neg(\exists x)(\exists y)Lxy$

Predicate Truth Trees: Analysis

Truth trees can be used to determine various semantic properties about propositions, sets of propositions, and arguments. Using truth trees to do this requires that you (i) set up the tree in a specific way to test for a specific property (you can't just stack the propositions in every instance), (ii) know how a closed (or completed open) tree indicates a specific semantic property. Luckily, trees are setup and analyzed exactly how they are in propositional logic, e.g. an argument $\mathbf{P, Q, \ldots, Y \vdash Z}$ is *valid* in $\mathbb{RL}$ if and only if it is impossible for the premises to be true and the conclusion false. A truth tree shows that an argument $\mathbf{P, Q, \ldots, Y \vdash Z}$ is *valid* in $\mathbb{RL}$ if and only if $\mathbf{P, Q, R, \ldots, Y, \neg Z}$ determines a closed tree.

Practice

Contingency, Tautology, Contradiction

1. $(\exists x)(Px \land Rx) \land (\forall x)(\neg Px \land Wx)$
2. $(\exists x)(\exists y)Pxy \land \neg(\exists x)Pxx$
3. $(\forall x)(\forall y)\neg(\exists z)(Sxyz \lor \neg Sxxx)$

Consistency

1. $(\forall x)\neg Paxx, (\exists x)(\forall y)Paxy$
2. $(\exists x)Pxx \lor (\forall x)Pxx, (\exists x)\neg(\exists y)Pyx$
3. $\neg(\forall x)\neg(Px \rightarrow Mx), \neg(\exists x)\neg(Px \land \neg Mx), (\forall y)(Py \rightarrow Zy) \lor (\exists x)(Mx \lor Px)$

Validity

1. $(\exists x)(\forall y)(Mx \land \neg Ry) \vdash (\exists y)(\forall z)(My \land \neg Rz)$
2. $(\exists x)(\exists z)Lxz, (\forall x)(\forall z)Lxz \rightarrow (\exists z)Mxx \vdash (\forall z)\neg Mzz$
3.* $\neg(\exists z)(Lzz \rightarrow Mzz), (\forall z)\neg(Lzz \rightarrow Mzz) \rightarrow (\exists x)Pxx \vdash \neg(\forall w)\neg Pww \lor (\exists z)Mzz$

Answer to #3: Start by Setting up the Tree with a negated conclusion

1	¬(∃z)(Lzz→Mzz)	P
2	(∀z)¬(Lzz→Mzz)→(∃x)Pxx	P
3	¬[¬(∀w)¬Pww∨(∃z)Mzz]	P

Next, decompose line (3) since it stacks!

1	¬(∃z)(Lzz→Mzz),	P
2	(∀z)¬(Lzz→Mzz)→(∃x)Pxx	P
3	¬[¬(∀w)¬Pww∨(∃z)Mzz]✓	P
4	¬¬(∀w)¬Pww	3¬∨D
5	¬(∃z)Mzz	3¬∨D

Next, let's clean up the propositions in the tree using ¬¬D and negated quantifier decomposition rules.

1	¬(∃z)(Lzz→Mzz)✓	P
2	(∀z)¬(Lzz→Mzz)→(∃x)Pxx	P
3	¬[¬(∀w)¬Pww∨(∃z)Mzz]✓	P
4	¬¬(∀w)¬Pww✓	3¬∨D
5	¬(∃z)Mzz✓	3¬∨D
6	(∀z)¬(Lzz→Mzz)	1¬∃D
7	(∀w)¬Pww	4¬¬D
8	(∀z)¬Mzz	5¬∃D

We have a lot of universally quantified propositions (6), (7), and (8). Let's hold off on decomposing these and decompose line (2) first, which is a conditional.

1	¬(∃z)(Lzz→Mzz)✓		P
2	(∀z)¬(Lzz→Mzz)→(∃x)Pxx✓		P
3	¬[¬(∀w)¬Pww∨(∃z)Mzz]✓		P
4	¬¬(∀w)¬Pww✓		3¬∨D
5	¬(∃z)Mzz✓		3¬∨D
6	(∀z)¬(Lzz→Mzz)		1¬∃D
7	(∀w)¬Pww		4¬¬D
8	(∀z)¬Mzz		5¬∃D
9	¬(∀z)¬(Lzz→Mzz)	(∃x)Pxx	2→D

Let's work on the simpler branch first. This is the right branch. We will start with decomposing the existentially quantified proposition at line (9). Remember: do your existentials before you go universal!

1	$\neg(\exists z)(Lzz \to Mzz)$ ✓		P
2	$(\forall z)\neg(Lzz \to Mzz) \to (\exists x)Pxx$ ✓		P
3	$\neg[\neg(\forall w)\neg Pww \vee (\exists z)Mzz]$ ✓		P
4	$\neg\neg(\forall w)\neg Pww$ ✓		$3\neg\vee D$
5	$\neg(\exists z)Mzz$ ✓		$3\neg\vee D$
6	$(\forall z)\neg(Lzz \to Mzz)$		$1\neg\exists D$
7	$(\forall w)\neg Pww$		$4\neg\neg D$
8	$(\forall z)\neg Mzz$		$5\neg\exists D$
9	$\neg(\forall z)\neg(Lzz \to Mzz)$	$(\exists x)Pxx$ ✓	$2\to D$
10		Paa	$9\exists D$

With ‘Paa’ at line (10), we can close the right branch if we decompose line (7) and replace the bound *w*’s with *a’s*.

1	$\neg(\exists z)(Lzz \to Mzz)$ ✓		P
2	$(\forall z)\neg(Lzz \to Mzz) \to (\exists x)Pxx$ ✓		P
3	$\neg[\neg(\forall w)\neg Pww \vee (\exists z)Mzz]$ ✓		P
4	$\neg\neg(\forall w)\neg Pww$ ✓		$3\neg\vee D$
5	$\neg(\exists z)Mzz$ ✓		$3\neg\vee D$
6	$(\forall z)\neg(Lzz \to Mzz)$		$1\neg\exists D$
7	$(\forall w)\neg Pww$		$4\neg\neg D$
8	$(\forall z)\neg Mzz$		$5\neg\exists D$
9	$\neg(\forall z)\neg(Lzz \to Mzz)$	$(\exists x)Pxx$ ✓	$2\to D$
10		Paa	$9\exists D$
11		$\neg Pww$	
		X	

Next, let’s turn to the left branch. Notice, however, that line (9) and line (6) are literal negations of each other. Thus, that branch closes too!

1	¬(∃z)(Lzz→Mzz)✓		P
2	(∀z)¬(Lzz→Mzz)→(∃x)Pxx✓		P
3	¬[¬(∀w)¬Pww∨(∃z)Mzz]✓		P
4	¬¬(∀w)¬Pww✓		3¬∨D
5	¬(∃z)Mzz✓		3¬∨D
6	**(∀z)¬(Lzz→Mzz)**		1¬∃D
7	(∀w)¬Pww		4¬¬D
8	(∀z)¬Mzz		5¬∃D
9	**¬(∀z)¬(Lzz→Mzz)**	(∃x)Pxx✓	2→D
10	**X**	Paa	9∃D
11		¬Pww	
		X	

Viola! We're done. What we have then is a tree where all of the branches are closed, ergo a closed tree. And what does a closed tree tell us about the argument we are teaching for?

It's valid!