1 Validity in PC

Let β be a term that is free for term α in PC - wff ϕ.

1.1

We need to prove by induction that for any wff ϕ and model $\mathscr{M}$, if variable assignments g and h agree on all variables with free occurrences in ϕ, then $V_{\mathscr{M},g}(\phi) = V_{\mathscr{M},h}(\phi)$.

Proof. Base Case. We consider wffs that do not involve any connectives, of the form $\Gamma(\alpha_1, \ldots, \alpha_n)$, where Γ is an n - place predicate and $\alpha_1, \ldots, \alpha_n$ are terms. Let our variable assignments g and h agree for all variables free in ϕ. For the case that:

$$V_{\mathscr{M},g}\Gamma(\alpha_1, \ldots, \alpha_n) \neq V_{\mathscr{M},h}(\Gamma(\alpha_1, \ldots \alpha_n))$$

Then our variable assignments, for some α_i would denote different objects. Specifically, there must be an instance where $[\alpha_i]_{\mathscr{M},g} \neq [\alpha_i]_{\mathscr{M},h}$. But since α_i is a constant, then the denotation is assigned by $\mathscr{I}$ in $\mathscr{M}$, then the denotations must be identitcal.

$$[\alpha_i]_{\mathscr{M},g} = [\alpha_i]_{\mathscr{M},h}$$

Since g and h agree on all variablee assignments, then they will agree on all free variables in $\Gamma(\alpha_1, \ldots, \alpha_n)$. Therefore $[\alpha_i]_{\mathscr{M},g} = [\alpha_i]_{\mathscr{M},h}$.
Inductive Step. We now want to show that more complex sentences built from atomic wffs like ϕ will also maintain this property. First, for $(\neg)$:

$$V_{\mathscr{M},g}(\neg\phi) = 1 \iff V_{\mathscr{M},g}(\phi) = 0$$

By hypothesis we know that $V_{\mathscr{M},g}(\phi) = V_{\mathscr{M},h}(\phi)$. So it follows immediately that $V_{\mathscr{M},g}(\neg\phi) = V_{\mathscr{M},h}(\neg\phi)$. The connective $(\neg)$ will preserve denoted equivalence. Next, for the connective $(\rightarrow)$.

$$V_{\mathscr{M},g}(\phi_i \rightarrow \phi_j) = 1 \iff V_{\mathscr{M},g}(\phi_i) = 0 \text{ or } V_{\mathscr{M},g}(\phi_j) = 1$$

Since $V_{\mathscr{M},g}(\phi) = V_{\mathscr{M},h}(\phi)$, then we confidently say that any instance of $V_{\mathscr{M},g}(\phi_i) = V_{\mathscr{M},h}(\phi_i)$ and $V_{\mathscr{M},g}(\phi_j) = V_{\mathscr{M},h}(\phi_j)$ from our base case. Therefore the connective $(\rightarrow)$ will also preserve denoted equivalence. Lastly, $(\forall)$.

$$V_{\mathscr{M},g}(\forall\alpha\phi) = 1 \iff V_{\mathscr{M},g}(g[\alpha \dashrightarrow u])(\phi) = 1 \text{ s.t. } u \in \mathscr{D}$$

In other words, $(\forall)$ evaluates to true for a sentence ϕ in a model, when each object u that is referenced by g makes ϕ true. We know that $V_{\mathscr{M},g}(\phi) = V_{\mathscr{M},h}(\phi)$ by hypothesis and that h and g agree on all free variables. Therefore $(\forall)$ will also preserve denoted equivalence. Therefore, for complex sentences including $(\neg, \rightarrow, \forall)$ will also maintain $V_g(\phi) = V_h(\phi)$ for a model $\mathscr{M}$. □

1.2

We now endeavor to show that if $[\alpha]_g = [\beta]_g$, then $V_g(\phi) = V_g(\phi(\beta/\alpha))$.

Proof. Base Case. In a similar strategy as before, we consider wffs that do not involve any connectives, of the form $\Gamma(\alpha_1, \ldots, \alpha_n)$, where Γ is an n - place predicate and $\alpha_1, \ldots, \alpha_n$ are terms. Let our variable assignments g and h agree for all variables free in ϕ. Knowing that $[\alpha]_g = [\beta]_g$, then:

$$V_{\phi,g}(\Gamma(\alpha_1,\ldots,\alpha_n)) = V_{\phi,g}(\Gamma(\beta_1,\ldots,\beta_n))$$

This because $\mathscr{I}(\alpha) = \mathscr{I}(\beta)$. The interpretation function for the variable assignment function g will map to the same object when computing $[\alpha]$ and $[\beta]$. Therefore, $[\alpha]_g = [\beta]_g \to V_{\mathscr{M},g}(\phi(\beta/\alpha))$.
Inductive Step. We add connectives $(\neg, \to, \forall)$, and proceed in very similar fashion as the previous proof. Firstly, $(\neg)$.

$$V_{\mathscr{M},g}(\neg\phi) = 1 \iff V_{\mathscr{M},g}(\phi) = 0$$

By hypothesis (and from our base case) we know that $V_{\mathscr{M},g}(\phi(\alpha)) = V_{\mathscr{M},g}(\phi(\beta))$. So, we immediately get $V_{\mathscr{M},g}(\neg\phi(\alpha)) = V_{\mathscr{M},g}(\neg\phi(\beta))$. The connective $(\neg)$ will preserve denoted equivalence. Next, for the connective $(\to)$.

$$V_{\mathscr{M},g}(\phi_i \to \phi_j) = 1 \iff V_{\mathscr{M},g}(\phi_i) = 0 \text{ or } V_{\mathscr{M},g}(\phi_j) = 1$$

Since $V_{\mathscr{M},g}(\phi(\alpha)) = V_{\mathscr{M},g}(\phi(\beta))$, then we know that any instance of $V_{\mathscr{M},g}(\phi_i(\alpha)) = V_{\mathscr{M},g}(\phi_i(\beta))$ and $V_{\mathscr{M},g}(\phi_j(\alpha)) = V_{\mathscr{M},g}(\phi_j(\beta))$ from our base case. Therefore the connective $(\to)$ will also preserve denoted equivalence. Lastly, $(\forall)$.

$$V_{\mathscr{M},g}(\forall\alpha\phi) = 1 \iff V_{\mathscr{M},g}(g[\alpha \dashrightarrow u])(\phi) = 1 \text{ s.t. } u \in \mathscr{D}$$

In other words, $(\forall)$ evaluates to true for a sentence ϕ in a model, when each object u that is referenced by g makes ϕ true. We know that $V_{\mathscr{M},g}(\phi(\alpha)) = V_{\mathscr{M},h}(\phi(\beta))$ by hypothesis. Nothing has changed with the $\mathscr{I}$ interpretation function, and since it maps β and α to the same object u, $V_{\mathscr{M},g}(\forall\phi\alpha) = V_{\mathscr{M},g}(\forall\phi\beta)$. Therefore $(\forall)$ will also preserve denoted equivalence for β and α.
To conclude, complex sentences including $(\neg, \to, \forall)$ will also maintain $V_g(\phi(\alpha)) = V_g(\phi(\beta))$ for a model $\mathscr{M}$. □

1.3

Now the task is to show that $\vDash_{PC} \forall\alpha\phi \to \phi(\beta/\alpha)$

Proof. Assume for reductio that $V_{\mathscr{M},g}(\forall\alpha\phi) = 1$ while $V_{\mathscr{M},g}(\phi(\beta/\alpha)) = 0$.

1. This would entail that for each object $u \in \mathscr{D}$, $V_{\mathscr{M},g}[\alpha \dashrightarrow u](\phi) = 1$.
2. That is in (1), $\mathscr{I}$ and the variable assignment g map α to some object u, s.t. $\phi(\alpha)$ obtains.
3. We know that $[\alpha]_g = [\beta]_g$ in $\mathscr{M}\,\langle\mathscr{I}\rangle$.
4. $\mathscr{I}$ and variable assingmnt g must map to the same object u for α, so we cannot obtain $V_{\mathscr{M},g}[\alpha \dashrightarrow u](\phi) = 1$ while $V_{\mathscr{M},g}[\beta \dashrightarrow u](\phi) = 0$
5. $\bot(1),(4)$.
6. $\therefore \vDash_{PC} \forall\alpha \to \phi(\beta/\alpha)$

□

1.4

Next, that $\vDash_{PC} \forall\alpha(\phi \to \psi) \to (\phi \to \forall\alpha\psi)$, whenever α does not occur free in ϕ.

1. Let $V_{\mathscr{M},g}(\forall\alpha(\phi \to \psi)) = 1$, that for every object u s.t. $g[\alpha \dashrightarrow u]$ by $\mathscr{I}$, the statement $\phi \to \psi$ is true.
2. Assume now that $V_{\mathscr{M},g}(\phi \to \forall\alpha\psi) = 0$ for reductio.
3. From (2), it must be that $V_{\mathscr{M},g}(\phi) = 1$ while $V_{\mathscr{M},g}(\forall\alpha\psi) = 0$.
4. From (3), given that $V_{\mathscr{M},g}(\phi) = 1$, the consequent in (1) cannot be false.
5. More specifically, α does not occur in ϕ. Thus the truth value of ϕ is independent of α, and we have (semantically) $V_{\mathscr{M},g}(\forall\alpha((1) \to \psi)) = 1$.
6. ψ in (1, 5) is still dependent on α. For the relation in (1, 5) to hold, $V_{\mathscr{M},g}(\forall\alpha(\psi)) \neq 0$.
7. $\bot(3, 6)$.
8. $\therefore\ V_{\mathscr{M},g}(\phi \to \forall\alpha\psi) = 1$.

We discharge the entire proof with:

$$\vDash_{PC} \forall\alpha(\phi \to \psi) \to (\phi \to \forall\alpha\psi)$$

1.5

Lastly, that if $\vDash_{PC} \phi$, then $\vDash_{PC} \forall\alpha\phi$.

1. Assume that in $\mathscr{M}\ \langle\mathscr{D}, \mathscr{I}\rangle$ with a variable assignment function g, $V(\phi) = 1$.
2. This entails that $V_{\mathscr{M},g}(\phi) = 1$.
3. ϕ is independent of any term.
4. Thus, $\forall\alpha$ s.t. $g[\alpha \dashrightarrow u]$ where $u \in \mathscr{D}$, $V_{\mathscr{M},g}(\phi) = 1$.
5. $\therefore\ V_{\mathscr{M},g}(\forall\alpha\phi) = 1$ (Since $\mathscr{I}$ and g will not map to any object where that makes ϕ false, given the independence of ϕ).

We discharge with $\vDash_{PC} \phi \to \forall\alpha\phi$.

2

We are given $\mathscr{L}_=$, $\mathscr{L}_\iota$, and $\mathscr{L}_{\iota,\lambda}$. We want to symbolize with the dictionary:

(*) The King of France is not bald.
K : ... is king of France.
F : ... is bald.

2.1

Firstly, we endeavor to obtain two semantically non-equivalent symbolizations of (*) in $\mathscr{L}_=$.

$$\exists x((Kx \land \neg Fx) \land \forall y(Ky \to y = x))$$
$$\neg \exists x((Kx \land Fx) \land \forall y(Ky \to y = x))$$

2.2

We now symbolize (*) using ι.

$$\neg F(\iota x Kx)$$

Parentheses are added for clarity. This articulation is equivalent to the first statement in the above section. This is because it asserts the existence of a king of France. For the second statement using the negated existential qualifier, we say there exists no king of France that is bald. This statement is satisfied *even if* there exists no king of France at all.

Therefore, this is equivalent to the more-assertive first statement where we say that *there exists* a king of France–exactly one–who is not bald.

2.3

Using $\mathscr{L}_{\iota,\lambda}$, we can acquire:

$$\neg \iota y(\lambda x(Kx \land Fx)(y))$$

To show equivalence, we would need identical truth conditions. In the $\mathscr{L}_{\iota,\lambda}$ statement, we have a λ statement that obtains when the x is the king of France and bald. But the ι operator asserts that there is no such $y \in \mathscr{D}$ that satisfies that statement. The statement would be true if there were no king of France, if there were no object $u \in \mathscr{D}$ that satisfied the left conjuct of $\lambda x(Kx \land Fx)$, it would never obtain. And thus the ι operator would also obtain, given that no y exists to satisfy the λ statement. Therefore this is logically equivalent to $\neg \exists x((Kx \land Fx) \land \forall y(Ky \to y = x))$, since each permits the non-existence of a king, while asserting that if there is some u s.t. Ku obtains, it will obtain Fu simultaneously. The key is that for both of these statements, we say "there does not exist".

2.4

We now compare and contrast the statements using existential quantifiers and $\mathscr{L}_{\iota,\lambda}$. It is obvious that we employ different notation and operators in each case. Besides this, the most glaring difference is that there is some nuance where the symbolizations produce different results. The two symbolizations both produce the desired truth values for the case where a king does exist in the domain. That is at least satisfactory. But in terms of stronger logical implication, it seems more desirable to assert a king exists. In these senses, the identity operator is redundant if we are employing the ι and even λ operators.

3

Binary generalized quantifiers.

3.1

Firstly, $V_{\mathscr{M},g}((\text{No } \alpha : \phi)\psi) = 1$ iff $|\phi^{\mathscr{M},g} \cap \psi^{\mathscr{M},g,\alpha}| = 0$. This sentence can be symbolized using (=).

$$\forall x(\phi(x) \to \neg\psi(x))$$

Next, $V_{\mathscr{M},g}((\text{At least two } \alpha : \phi)\psi) = 1$ iff $|\phi^{\mathscr{M},g} \cap \psi^{\mathscr{M},g,\alpha}| \geq 2$. This sentence can also be symbolized using $\mathscr{L}_=$.

$$\exists x \exists y((\phi(x) \land \phi(y) \land (\psi(x) \land \psi(y))) \land \neg(x = y))$$

For $V_{\mathscr{M},g}((\text{Finitely many } \alpha : \phi)\psi) = 1$ iff $|\phi^{\mathscr{M},g} \cap \psi^{\mathscr{M},g,\alpha}|$ is finte. We would have no way of expressing this sentence in $\mathscr{L}_=$. This is because there is no way to describe infinity using $\mathscr{L}_=$. If we can't say infinite, then we unable to say not infinite using $\mathscr{L}_=$. By extension, we don't have way to say finite. See LfP 120 - 121.

For $V_{\mathscr{M},g}((\text{Most } \alpha : \phi)\psi) = 1$ iff $|\phi^{\mathscr{M},g} \cap \psi^{\mathscr{M},g,\alpha}| > |\phi^{\mathscr{M},g,\alpha} - \psi^{\mathscr{M},g,\alpha}$. This would be inexpressible using $\mathscr{L}_=$. This is because we have no way of comparing set *sizes* with the notation. See LfP 120 - 122. Even if you tried to say something like: "There are at least x number of things," you could not say that they are *more than* the number of other things.

3.2

While Sider introduces $\exists_\infty$ for monadic expressions, second order logic quantifies over predicates. This means that we can express a finite number and also an amount (so we can also say 'finitely many' and 'most'), as well as 'no' and 'at least'.

We would do this by quantifying over sets. If there were some second order variable X that accrued only ceratain x that satisfied a sentence. I found on a guide that a standard expression of finiteness is available via $f : A \to A$. In this definition of finiteness, every injective is surjective (learned about this online).

In SOL, size comparisons are available to use due to the same new technicalities we are afforded. Again, we can quantify over functions. A function that we quantify over in SOL is an injection if different inputs always yield different outputs. So if you inject the results from one function A into another, this means that A will be less than the second function! Bijection (where every element in a set is obtained by some element in another set), then you can compare sizes– if there is a bijection between them, they are the same size.