Logic, Computability and Incompleteness.
Exercise Set 2: Solutions

1) Consider the Turing machine quadruple $q_2S_1Rq_3$. Adopting the conventions used in the Lecture Slides (and B&J ch 10), including the restriction that the Turing machine in question can read/write only the symbols S_0 and S_1, construct a sentence in first-order logic that formalizes this quadruple.

The general 'axiom' for **move right** with state transition is the universally quantified conditional:
$\forall t\ \forall x\ \forall y[(t\mathbf{Q}_i x \wedge t\mathbf{S}_j x) \rightarrow (t'\mathbf{Q}_m x' \wedge (t\mathbf{S}_0 y \rightarrow t'\mathbf{S}_0 y \wedge \ldots \wedge t\mathbf{S}_r y \rightarrow t'\mathbf{S}_r y))]$

So for $q_2S_1Rq_3$ this yields:
$\forall t\ \forall x\ \forall y[(t\mathbf{Q}_2 x \wedge t\mathbf{S}_1 x) \rightarrow (t'\mathbf{Q}_3 x' \wedge (t\mathbf{S}_0 y \rightarrow t'\mathbf{S}_0 y \wedge t\mathbf{S}_1 y \rightarrow t'\mathbf{S}_1 y))]$

2) Let the configuration of a Turing machine computation at time s be the following:
...00$\underline{1}$011100...
2
where there are only blank squares remaining on the left and right of the tape, and where the number of the currently scanned square is p.

(i) Provide the formal description of time s in FOL.

A *description of time s* is a conjunction of the form:
$\mathbf{o}^{(s)}\mathbf{Q}_i\mathbf{o}^{(p)} \wedge \mathbf{o}^{(s)}\mathbf{S}_{j1}\mathbf{o}^{(p1)} \wedge \ldots \wedge \mathbf{o}^{(s)}\mathbf{S}_j\mathbf{o}^{(p)} \wedge \ldots \wedge \mathbf{o}^{(s)}\mathbf{S}_{jv}\mathbf{o}^{(pr)} \wedge$
$\forall y\ [(y \neq \mathbf{o}^{(p1)} \wedge y \neq \mathbf{o}^{(p)} \wedge \ldots \wedge y \neq \mathbf{o}^{(pr)}) \rightarrow \mathbf{o}^{(s)}\mathbf{S}_0 y]$

In the case of the configuration ...00$\underline{1}$011100...
2 the conjunction will be:

$\mathbf{o}^{(s)}\mathbf{Q}_2\mathbf{o}^{(p)} \wedge \mathbf{o}^{(s)}\mathbf{S}_1\mathbf{o}^{(p)} \wedge \mathbf{o}^{(s)}\mathbf{S}_0\mathbf{o}^{(p+1)} \wedge \mathbf{o}^{(s)}\mathbf{S}_1\mathbf{o}^{(p+2)} \wedge \mathbf{o}^{(s)}\mathbf{S}_1\mathbf{o}^{(p+3)} \wedge \mathbf{o}^{(s)}\mathbf{S}_1\mathbf{o}^{(p+4)} \wedge$
$\forall y\ [(y \neq \mathbf{o}^{(p)} \wedge y \neq \mathbf{o}^{(p+1)} \wedge y \neq \mathbf{o}^{(p+2)} \wedge y \neq \mathbf{o}^{(p+3)} \wedge y \neq \mathbf{o}^{(p+4)}) \rightarrow \mathbf{o}^{(s)}\mathbf{S}_0 y]$

(ii) Provide the formal description of time $s+1$, on the basis of the quadruple instruction in question 1).

The configuration at time s+1 would be:
...001$\underline{0}$11100...
3 yielding the formalization:

$\mathbf{o}^{(s+1)}\mathbf{Q}_3\mathbf{o}^{(p+1)} \wedge \mathbf{o}^{(s+1)}\mathbf{S}_1\mathbf{o}^{(p)} \wedge \mathbf{o}^{(s+1)}\mathbf{S}_0\mathbf{o}^{(p+1)} \wedge \mathbf{o}^{(s+1)}\mathbf{S}_1\mathbf{o}^{(p+2)} \wedge \mathbf{o}^{(s+1)}\mathbf{S}_1\mathbf{o}^{(p+3)} \wedge \mathbf{o}^{(s+1)}\mathbf{S}_1\mathbf{o}^{(p+4)}$
$\wedge\ \forall y\ [(y \neq \mathbf{o}^{(p)} \wedge y \neq \mathbf{o}^{(p+1)} \wedge y \neq \mathbf{o}^{(p+2)} \wedge y \neq \mathbf{o}^{(p+3)} \wedge y \neq \mathbf{o}^{(p+4)}) \rightarrow \mathbf{o}^{(s+1)}\mathbf{S}_0 y]$

3) Given the enumeration of Turing machines developed in the Lecture Slides (and B&J ch 5), determine the 3rd, 4th and 5th Turing machines in the list, give the number of the corresponding 'word', and compute the values of $\boldsymbol{u}(3)$, $\boldsymbol{u}(4)$ and $\boldsymbol{u}(5)$, where $\boldsymbol{u}$ is the antidiagonal function based on the list.

As stated in the slides, the first TM on the list is $q_1S_0Rq_1$, which is word number 1222212221212222 and results from concatenating the 4th 3rd 1st and 4th symbols of the 'alphabet'. The second TM on the list is $q_1S_0Lq_1$ which is word number

12222122212212222 and results from concatenating the 4^{th} 3^{rd} 2^{nd} and 4^{th} symbols of the 'alphabet'.
The third TM on the list is $q_1 S_0 R q_2$ which is word number
122221222121222222 and results from concatenating the 4^{th} 3^{rd} 1^{st} and 6^{th} symbols
The fourth TM on the list is $q_1 S_0 S_0 q_1$ which is word number
122221222122212222 and results from concatenating the 4^{th} 3^{rd} 3^{rd} and 4^{th} symbols of the 'alphabet'.
The fifth TM on the list is $q_1 S_1 R q_1$ which is word number
122221222221212222 and results from concatenating the 4^{th} 5^{th} 1^{st} and 4^{th} symbols of the 'alphabet'.

The machines/numbers/codes can be compared directly as follows

$q_1 S_0 R q_1$	1222212221212222	4 + 3 + 1 + 4 = 12
$q_1 S_0 L q_1$	12222122212212222	4 + 3 + 2 + 4 = 13
$q_1 S_0 R q_2$	122221222121222222	4 + 3 + 1 + 6 = 14
$q_1 S_0 S_0 q_1$	122221222122212222	4 + 3 + 3 + 4 = 14
$q_1 S_1 R q_1$	122221222221212222	4 + 5 + 1 + 4 = 14

The sum of the green enumerating numbers is the number of occurrences of the digit '2' in the corresponding word's number, where the number of digits for single quadruple machines will equal this sum plus 4 (for the four 1's involved). Hence the greater the number of 2's, the greater the corresponding word's number.

The first four of these machines halt when scanning a positive integer in monadic notation as input, and hence compute the identity function $\boldsymbol{i}(x) = x$. The fifth will move right to the end of the input number, and halt in non-standard configuration reading a 0. So it yields an 'undefined'. The antidiagonal function $\boldsymbol{u}$ on positive integers has been defined such that

$$\boldsymbol{u}(n) = \{1, \quad \text{if } f_n(n) \text{ is undefined}$$
$$= \{f_n(n) \;+ 1 \text{ otherwise}$$

Thus the first five values of $\boldsymbol{u}$ are $\boldsymbol{u}(1) = 2$, $\boldsymbol{u}(2) = 3$, $\boldsymbol{u}(3) = 4$, $\boldsymbol{u}(4) = 5$ and $\boldsymbol{u}(5) = 1$.

4) Induction on the complexity of a formula is a form of proof by mathematical induction wherein the basis step for the minimal case is to prove that the statement or property holds for the set of atomic formulas, and the induction step is to prove that, *if* the statement or property holds for all formulas constructed from n or fewer applications of the formation rules for compound formulas, then it holds for $n + 1$ applications. In other words, the induction step requires proving that the result of applying a formation rule will preserve the property, if the inputs to the rule have the property.
Consider the propositional fragment $\mathbf{L}_P$ of FOL defined with sentence letters P_1, P_2,... and the formations rules: all sentence letters of $\mathbf{L}_P$ are atomic formulas of $\mathbf{L}_P$ and

(i) if $\mathbf{A}$ is a formula of $\mathbf{L}_P$, then $\neg\mathbf{A}$ is a formula of $\mathbf{L}_P$
(ii) if $\mathbf{A}$, $\mathbf{B}$ are formulas of $\mathbf{L}_P$, then $(\mathbf{A} \wedge \mathbf{B})$, $(\mathbf{A} \vee \mathbf{B})$, $(\mathbf{A} \rightarrow \mathbf{B})$, and $(\mathbf{A} \leftrightarrow \mathbf{B})$ are formulas of $\mathbf{L}_P$.

Prove by induction on complexity that if two truth value assignments $\mathcal{J}_1$ and $\mathcal{J}_2$ assign the same truth values to the sentence letters in a formula $\boldsymbol{S}$, then $\boldsymbol{S}$ has the same truth value in $\mathcal{J}_1$ and $\mathcal{J}_2$.

Proof by induction on structural complexity.

base step: if $\boldsymbol{S}$ is a statement letter and assignments $\mathcal{J}_1$ and $\mathcal{J}_2$ assign the same truth value to $\boldsymbol{S}$, then trivially $\boldsymbol{S}$ has the same truth value in both assignments, *viz.* $\mathcal{J}_1(\boldsymbol{S}) = \mathcal{J}_2(\boldsymbol{S})$.

induction step (by cases):

(i) suppose $\boldsymbol{S}$ is of the form $\neg\mathbf{A}$. $\mathbf{A}$ is constructed using fewer applications of the formation rules than $\neg\mathbf{A}$, so by induction hypothesis $\mathcal{J}_1(\mathbf{A}) = \mathcal{J}_2(\mathbf{A})$. If that value is false, then the rules of truth for negation require that $\boldsymbol{S}$ is true in both $\mathcal{J}_1$ and $\mathcal{J}_2$. If that value is true, then the rules of truth for negation require that $\boldsymbol{S}$ is false in both $\mathcal{J}_1$ and $\mathcal{J}_2$. In either case, $\mathcal{J}_1(\boldsymbol{S}) = \mathcal{J}_2(\boldsymbol{S})$.

(ii) suppose $\boldsymbol{S}$ is of the form $(\mathbf{A} \wedge \mathbf{B})$. $\mathbf{A}$ and $\mathbf{B}$ are both constructed using fewer applications of the formation rules than $(\mathbf{A} \wedge \mathbf{B})$, so by induction hypothesis both $\mathbf{A}$ and $\mathbf{B}$ individually have the same truth value in $\mathcal{J}_1$ and $\mathcal{J}_2$. If either of $\mathbf{A}$ or $\mathbf{B}$ is false, then by the rules of truth for conjunction, $(\mathbf{A} \wedge \mathbf{B})$ will be false in both $\mathcal{J}_1$ and $\mathcal{J}_2$. And if $\mathbf{A}$ and $\mathbf{B}$ are both true, then by the rules of truth for conjunction, $(\mathbf{A} \wedge \mathbf{B})$ will be true in both $\mathcal{J}_1$ and $\mathcal{J}_2$. So in all cases $\boldsymbol{S}$ has the same truth value in both assignments, and $\mathcal{J}_1(\boldsymbol{S}) = \mathcal{J}_2(\boldsymbol{S})$.

(iii) - (v) analogous reasoning applies to the remaining binary connectives.

5) Using the derivation system developed in the lecture slides (and B&J chapter 9), give proofs of the following two claims. Be sure to specify the relevant set Δ, include annotations, and establish in each case that your derivation is a refutation.

i) $\vdash \neg\exists x Px \rightarrow \forall x(Px \rightarrow Sx)$

The counterexample set = $\{\neg(\neg\exists x Px \rightarrow \forall x(Px \rightarrow Sx))\}$. Need to express in prenex normal form, which first requires quantifier conversion in the antecedent to yield $\{\neg(\forall x\neg Px \rightarrow \forall x(Px \rightarrow Sx))\}$ then relabeling of variables to yield $\{\neg(\forall y\neg Py \rightarrow \forall x(Px \rightarrow Sx))\}$ and finally quantifier extraction to yield:

$\Delta = \{\exists y\forall x\neg(\neg Py \rightarrow (Px \rightarrow Sx))\}$

1. $\exists y\forall x\neg(\neg Py \rightarrow (Px \rightarrow Sx))$ Δ
2. $\forall x\neg(\neg P\mathbf{a} \rightarrow (Px \rightarrow Sx))$ EI, 1.
3. $\neg(\neg P\mathbf{a} \rightarrow (P\mathbf{a} \rightarrow S\mathbf{a}))$ UI, 2.

This derivation is a refutation, since line **3**. constitutes a finite set of quantifier free sentences that is truth-functionally unsatisfiable (you can check this fact with a truth table).

ii) $\exists x\forall y(x = y), Pa \vdash \forall x(x = a)$

The counterexample set = $\{\exists x\forall y(x = y), P\mathrm{a}, \neg\forall x(x = \mathrm{a})\}$

$\Delta = \{\exists x\forall y(x = y), P\mathrm{a}, \exists x\neg(x = \mathrm{a})\}$

1. $\exists x\neg(x = \mathbf{a})$ Δ
2. $\neg(\mathbf{b} = \mathbf{a})$ EI, 1.
3. $\exists x\forall y(x = y)$ Δ
4. $\forall y(\mathbf{c} = y)$ EI, 3.
5. $\mathbf{c} = \mathbf{a}$ UI, 4.
6. $\mathbf{c} = \mathbf{b}$ UI, 4.

This derivation is a refutation, since lines **2**.,**5**., and **6**. constitute a finite set of quantifier free sentences that is unsatisfiable.

6) Show that if a set of sentences Δ has models with arbitrarily large finite domains, then Δ has a model with an infinite domain.

Suppose that the set of sentences Δ has arbitrarily large finite models. For any positive integer n we can write a sentence $\boldsymbol{F}_n$, using n distinct variables, such that $\boldsymbol{F}_n$ is true only in models with at least n element domains. $\boldsymbol{F}_n$ will be of the form $\exists x_1 \exists x_2 \ldots \exists x_n(x_1 \neq x_2 \wedge \ldots \wedge x_1 \neq x_n \wedge \ldots \wedge x_2 \neq x_n \wedge \ldots \wedge x_{n-1} \neq x_n)$, where all pairwise identities are negated. Let Σ be the set of *all* such sentences, for each positive integer n. Every finite subset of $\Delta \cup \Sigma$ has a model, because Δ must have models at least as large as the greatest n for any $\boldsymbol{F}_n \in \Sigma$. The compactness theorem states that a set of sentences Γ is unsatisfiable iff some finite subset $\Gamma_0 \subseteq \Gamma$ is unsatisfiable. As a consequence (and rendering 'satisfiable' in terms of 'having a model'), if every finite subset of Γ has a model, then Γ has a model. So by the compactness theorem $\Delta \cup \Sigma$ has a model. No model of Σ can have a finite domain, so $\Delta \cup \Sigma$ must have a model $\mathcal{J}$ with an infinite domain. And since $\mathcal{J}$ is a model of $\Delta \cup \Sigma$ it must also be a model of Δ alone, and hence Δ has a model with an infinite domain.

7) To each of the subsets of the set of natural numbers there corresponds the distinct truth that 0 is or is not a member of that subset. Establish that there must then exist truths of the full theory of the natural numbers that are not theorems of formal arithmetic (*hint*: use cardinality).

According to Cantor's Theorem, the power set of any set always has greater cardinality than the set itself. The set of natural numbers is denumerable, so by Cantor's Theorem, the set of all sets of natural numbers is *uncountable*. To each such set there corresponds the distinct (and *independent*) truth that 0 is or is not a member of that set. Hence there are uncountably many truths of the full theory of the natural numbers. In contrast, the formulas in the formal theory of arithmetic consist of finite strings of symbols from an enumerable alphabet, and the set of all such finite strings is itself enumerable. So there are only countably many sentences in the language of formal arithmetic, and the theorems are a proper subset of these (assuming that formal arithmetic is consistent!). So the cardinality of the set of theorems of formal arithmetic is strictly less than the cardinality of the set of truths of the full theory of the natural numbers, which means that not all such truths can be expressed as theorems in a formal system.

8) What formula in the language $\boldsymbol{L}$ of Robinson Arithmetic represents the recursive function **sum**(x,y)? Establish that this formula has the required properties.

sum(x,y) is represented by the formula $x + y = z$. To establish this, need to show that if

sum(i, j) = k, then

i) $\vdash_Q$ $\mathbf{i} + \mathbf{j} = \mathbf{k}$ and

ii) $\vdash_Q$ $\forall x\, (\mathbf{i} + \mathbf{j} = x \leftrightarrow x = \mathbf{k})$

Property i) will be proved by induction on j. We will take it as given (by prior construction) that **sum** computes the normal addition function $x + y$.

base step: $j = 0$.

$\vdash_Q$ $\mathbf{i} + \mathbf{o} = \mathbf{i}$ by axiom Q4 [$\forall x(x + \mathbf{o} = x)$]

induction step: $j = m'$

Need to show that $\vdash_Q$ $\mathbf{i} + \mathbf{m}' = \mathbf{k}$. There is some n such that $k = n'$ and

$i + m = n$. So by induction hypothesis $\vdash_Q$ $\mathbf{i + m} = \mathbf{n}$ and therefore $\vdash_Q$ $(\mathbf{i + m})' = \mathbf{n}'$. Since $\vdash_Q$ $(\mathbf{i + m})' = \mathbf{i + m}'$ by axiom Q5 $[\forall x \forall y(x + y') = (x + y)']$ it follows that $\vdash_Q$ $\mathbf{i + m}' = \mathbf{n}'$ and hence $\vdash_Q$ $\mathbf{i + j} = \mathbf{k}$.

Property ii) follows as a direct logical consequence of i). Can check this using your favorite deductive system.
For example, using the method of question 5:
the counterexample set = $\{\mathbf{i + j} = \mathbf{k},\ \neg\forall x\, (\mathbf{i + j} = x \leftrightarrow x = \mathbf{k})\}$ and
$\Delta = \{\ \mathbf{i + j} = \mathbf{k},\ \exists x \neg(\mathbf{i + j} = x \leftrightarrow x = \mathbf{k})\}$

1. $\exists x \neg(\mathbf{i + j} = x \leftrightarrow x = \mathbf{k})$ $\quad\Delta$
2. $\neg(\mathbf{i + j} = \mathbf{a} \leftrightarrow \mathbf{a} = \mathbf{k})$ $\quad$ EI, 1.
3. $\mathbf{i + j} = \mathbf{k}$ $\quad\Delta$

This derivation is a refutation, since lines **2**. and **3**. constitute a finite set of quantifier free sentences that is unsatisfiable.

Express this formula in terms of the base level syntax of the object language (i.e. using the primitive vocabulary items displayed on B&J p. 171 and the official formation rules for FOL).
In the base level syntax of the object language, the expression $x + y = z$ has the form:
$A^2_0\,(f^2_0\,(x_0, x_1), x_2)$

9) Using the Gödel numbering scheme from the lecture slides and B&J, determine gn[$\forall x \exists y A^2_1(x\,,\, f^2_2(x, y))$].

gn[$\forall x \exists y A^2_1(x\,,\, f^2_2(x, y))$] = 495459788915296889915295922

10) Let $\boldsymbol{F}$ be the formula $\forall y A^2_1(x\,,\, y)$. What is the diagonalization of $\boldsymbol{F}$?

The diagonalization of $\forall y A^2_1(x\,,\, y)$ is $\exists x(x = \ulcorner \boldsymbol{F} \urcorner \wedge \forall y A^2_1(x\,,\, y))$.
In somewhat more fine grained detail, gn[$\forall y A^2_1(x\,,\, y)$] = 495978891529592,
so $\ulcorner \boldsymbol{F} \urcorner = \mathbf{o}^{(495978891529592)}$ so
the diagonalization of $\forall y A^2_1(x\,,\, y)$ is $\exists x(x = \mathbf{o}^{(45978891529592)} \wedge \forall y A^2_1(x\,,\, y))$.

Use the derivation system from question 5) to establish that the diagonalization of $\boldsymbol{F}$ is logically equivalent to $\forall y A^2_1(\ulcorner \boldsymbol{F} \urcorner, y)$.

To establish that $\exists x(x = \ulcorner \boldsymbol{F} \urcorner \wedge \forall y A^2_1(x\,,\, y)) \equiv \forall y A^2_1(\ulcorner \boldsymbol{F} \urcorner, y)$, need to show that
(i) $\exists x(x = \ulcorner \boldsymbol{F} \urcorner \wedge \forall y A^2_1(x\,,\, y)) \vDash \forall y A^2_1(\ulcorner \boldsymbol{F} \urcorner, y)$ and
(ii) $\forall y A^2_1(\ulcorner \boldsymbol{F} \urcorner, y) \vDash \exists x(x = \ulcorner \boldsymbol{F} \urcorner \wedge \forall y A^2_1(x\,,\, y))$
For the first derivation, the counterexample set is
$\{\exists x(x = \ulcorner \boldsymbol{F} \urcorner \wedge \forall y A^2_1(x\,,\, y)),\ \neg\forall y A^2_1(\ulcorner \boldsymbol{F} \urcorner, y)\}$ and
$\Delta = \{\exists x \forall y(x = \ulcorner \boldsymbol{F} \urcorner \wedge A^2_1(x\,,\, y)),\ \exists y \neg A^2_1(\ulcorner \boldsymbol{F} \urcorner, y)\}$

1. $\exists x \forall y(x = \ulcorner \boldsymbol{F} \urcorner \wedge A^2_1(x\,,\, y))$ $\quad\Delta$
2. $\forall y(\mathbf{a} = \ulcorner \boldsymbol{F} \urcorner \wedge A^2_1(\mathbf{a}\,,\, y))$ $\quad$ EI, 1.
3. $\exists y \neg A^2_1(\ulcorner \boldsymbol{F} \urcorner, y)$ $\quad\Delta$
4. $\neg A^2_1(\ulcorner \boldsymbol{F} \urcorner, \mathbf{b})$ $\quad$ EI, 3.
5. $(\mathbf{a} = \ulcorner \boldsymbol{F} \urcorner \wedge A^2_1(\mathbf{a}\,,\, \mathbf{b}))$ $\quad$ UI, 2.

The finite set of quantifier-free sentences from lines **4**. and **5**. is unsatisfiable.
For the second derivation, the counterexample set is

$\{\forall y A^2_1(\ulcorner \boldsymbol{F} \urcorner, y), \neg\exists x(x = \ulcorner \boldsymbol{F} \urcorner \wedge \forall y A^2_1(x, y))\}$ and
$\Delta = \{\forall y A^2_1(\ulcorner \boldsymbol{F} \urcorner, y), \forall x \exists y \neg(x = \ulcorner \boldsymbol{F} \urcorner \wedge A^2_1(x, y))\}$

1. $\forall x \exists y \neg(x = \ulcorner \boldsymbol{F} \urcorner \wedge A^2_1(x, y))$ — Δ
2. $\exists y \neg(\ulcorner \boldsymbol{F} \urcorner = \ulcorner \boldsymbol{F} \urcorner \wedge A^2_1(\ulcorner \boldsymbol{F} \urcorner, y))$ — UI, 1.
3. $\neg(\ulcorner \boldsymbol{F} \urcorner = \ulcorner \boldsymbol{F} \urcorner \wedge A^2_1(\ulcorner \boldsymbol{F} \urcorner, \mathbf{a}))$ — EI, 2.
4. $\forall y A^2_1(\ulcorner \boldsymbol{F} \urcorner, y)$ — Δ
5. $A^2_1(\ulcorner \boldsymbol{F} \urcorner, \mathbf{a})$ — UI, 4.

The finite set of quantifier-free sentences from lines **3**. and **5**. is unsatisfiable.

11) Following the proof of the diagonal lemma, construct the sentence $\boldsymbol{G}$ such that

$$\vdash_Q \boldsymbol{G} \leftrightarrow \forall y A^2_1(\ulcorner \boldsymbol{G} \urcorner, y)$$

Given that the formula $\boldsymbol{F}$ is syntactically specified as $\forall y A^2_1(x, y)$, it will be necessary in the construction of $\boldsymbol{G}$ to rename the variables used in the slides.

Let the formula $A^2_d(z, x)$ represent the diagonal function **diag** in $\boldsymbol{Q}$. Let $\boldsymbol{H}$ be defined as the formula $\exists x(A^2_d(z, x) \wedge \forall y A^2_1(x, y))$. $\boldsymbol{H}$ contains just the variable z free.
Define $\boldsymbol{G}$ as the diagonalization of $\boldsymbol{H}$, *viz*.
$\exists z\,(z = \ulcorner \boldsymbol{H} \urcorner \wedge \exists x(A^2_d(z, x) \wedge \forall y A^2_1(x, y))$.
The proof of the diagonal lemma establishes that

$$\vdash_Q \boldsymbol{G} \leftrightarrow \forall y A^2_1(\ulcorner \boldsymbol{G} \urcorner, y)$$

12) Suppose that Robinson Arithmetic is augmented with a 1-place predicate symbol $\boldsymbol{K}$ intended to formalize *knowledge* of arithmetic, perhaps for the purpose of Knowledge Representation in some AI system. Suppose further that this predicate is part of the Gödel numbering scheme, and that the logic of the knowledge predicate includes the principles:
(i) $\vdash \boldsymbol{K}(\ulcorner \Phi \urcorner) \to \Phi$, since knowledge is considered to be factive, and
(ii) if $\vdash \Phi$, then $\vdash \boldsymbol{K}(\ulcorner \Phi \urcorner)$, which is intended to capture the idea that if a statement is proven then it is known. Show that the resulting formal theory of Knowledge Representation is inconsistent.

By the diagonal lemma, there is a sentence $\boldsymbol{G}$ such that:
1) $\vdash \boldsymbol{G} \leftrightarrow \neg \boldsymbol{K}(\ulcorner \boldsymbol{G} \urcorner)$.
2) $\vdash \boldsymbol{K}(\ulcorner \boldsymbol{G} \urcorner) \to \neg \boldsymbol{G}$ from 1).
3) $\vdash \boldsymbol{K}(\ulcorner \boldsymbol{G} \urcorner) \to \boldsymbol{G}$ from (i).
4) $\vdash \neg \boldsymbol{K}(\ulcorner \boldsymbol{G} \urcorner)$ by propositional logic on 2), 3).
5) $\vdash \boldsymbol{G}$ from 1), 4).
6) $\vdash \boldsymbol{K}(\ulcorner \boldsymbol{G} \urcorner)$ from 5), (ii).
7) $\vdash \boldsymbol{K}(\ulcorner \boldsymbol{G} \urcorner) \wedge \neg \boldsymbol{K}(\ulcorner \boldsymbol{G} \urcorner)$ from 4), 6).