Questions about Turing's Undecidability and Gödel's Incompleteness and Completeness Theorems

Ever since I read the book *Gödel, Escher, Bach* in my youth, I thought I had a good understanding of Gödel's Incompleteness Theorem and the area around it. Recently, I found myself bewildered by some basic lack of understanding. A and B are some preparatory questions, the hard conundrums are C and D.

A) For which system did Gödel prove completeness (Gödel's Vollständigkeitssatz, 1930)? ("Die Vollständigkeit der Axiome des logischen Funktionenkalküls")

According to Wikipedia: rather *semantic* completeness: Theorem 1. Every valid logical expression is provable. Equivalently, every logical expression is either satisfiable or refutable.

B) Is there a difference between the answer to (A) and *engerer Funktionenkalkül* (*restricted functional calculus*, *engerer Prädikatenkalkül*, *first-order predicate calculus* (modern terms: *first-order logic*, *Prädikatenlogik*, *Prädikatenlogik erster Stufe*))? In particular, the system that Turing used?

Turing, footnote † (p. 252/221[1]): The expression "the functional calculus" is used throughout to mean the *restricted* Hilbert functional calculus.

[Remark: *erweiterter Prädikatenkalkül* = *higher-order logic.*]

C) Why did Turing not mention that Gödel's Incompleteness Theorem is a direct corollary of his undecidability statement, nor did his contemporary followers and reviewers?

Turing (p. 259/261): "what I shall prove is quite different from the well-known results of Gödel".

Turing (p. 259/262): "If the negation of what Gödel has shown had been proved, ..., then we should have an immediate solution to the Entscheidungsproblem. ... Sooner or later $\mathcal{K}$ will reach either $\mathfrak{A}$ or $-\mathfrak{A}$."

Is there a reason for the formulation "If the negation ... had been proved", as opposed to the formulation "If the negation were true" or "If ... were false"?

In the introduction (p. 230/267): "..., conclusions are reached which are superficially similar to those of Gödel." Are these expressions just polite circumlocutions because he was modest (as Petzold surmises) or shy? Was it simply lack of knowledge and experience? (He was a 24-year-old master student.) Why did nobody among his contemporaries say clearly (or notice?) that this is an alternative proof of Gödel's Incompleteness Theorem? We know for example that Paul Bernays must have read it thoroughly, because Turing thanks him for pointing out errors, in the follow-up Erratum publication.

D) How can Gödel's Completeness Theorem be reconciled with some undecidability statements that Turing proved? (This is connected with questions (A) and (B).)

Turing models the decision problem for TM (Does a given TM ever print a zero, i.e., the symbol S_1?) in predicate calculus. He includes some of the Peano axioms among the clauses of the statements whose provability is to be checked, because predicate calculus does not contain numbers. (E.g., $F(x, y)$ is the successor predicate, p. 260/264; in the Erratum, this formulation is expanded and corrected.) Turing (p. 259/262): "Owing to the abscence of integers in **K** the proofs appear somewhat lengthy." He goes on to define a certain formula "Un($\mathcal{M}$)" in terms of the description of a Turing machine $\mathcal{M}$.

Lemma 1. (p. 261/269) *If S_1 appears on the tape in some complete configuration of $\mathcal{M}$ then* Un($\mathcal{M}$) *is provable.*

Actually, Turing states "$\Longleftrightarrow$" (p. 260/269), but the other direction is easy (Lemma 2, p. 262/276).

By Gödel's Completeness Theorem, either Un($\mathcal{M}$) or its negation must be provable. How does this go together with undecidability?

[1]Secondary page numbers after the slash refer to "The Annotated Turing" by Charles Petzold, Wiley 2008.

Some further unconnected remarks. The reduction "from Turing to Gödel" that is occasionally found in textbooks uses integers as given. [S. Arora and B. Barak: Computational Complexity — A Modern Approach, p. 24].

[Sipser, Introduction to the Theory of Computation, Section 6.2.] Introduction of "formal proof" on pp. 209–210 takes some leap of faith, without specifying precisely a formal system, only stating "reasonable properties" of proof: (1) Proof can be checked by a machine; (2) soundness.

Thm 6.14 and 6.15. Some true statement in $\mathrm{Th}(\mathcal{N}, +, \times)$ is not provable, in particular the statement $\psi_{\text{unprovable}}$.

$\mathrm{Th}(\mathcal{N}, +, \times)$ is defined semantically, as the set of all true statements.

It is *not* necessary that the proof system is sufficiently powerful to express, e.g., natural numbers. (If it is not powerful enough, it is a fortiori incomplete.)

p. 206 bottom: "Alonzo Church, *building on the work of Kurt Gödel* (?), showed that no algorithm can decide in general whether statements in number theory are true or false."

Hey! Theorem 6.6 requires that the output of f be interpretable as the description of a TM.