DOCUMENT RESUME

ED 382 668 TM 023 106

AUTHOR Bennett, Randy Elliot
TITLE Toward Intelligent Assessment: An Integration of Constructed Response Testing, Artificial Intelligence, and Model-Based Measurement.
INSTITUTION Educational Testing Service, Princeton, N.J.
REPORT NO ETS-RR-90-5
PUB DATE May 90
NOTE 45p.
PUB TYPE Reports - Evaluative/Feasibility (142)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Artificial Intelligence; *Constructed Response; *Educational Assessment; *Measurement Techniques; Models; Scoring; *Standardized Tests; Testing; Test Items; Test Use; Test Validity
IDENTIFIERS Experts; Partial Credit Model

ABSTRACT

A new assessment conception is described that integrates constructed-response testing, artificial intelligence, and model-based measurement. The conception incorporates complex constructed-response items for their potential to increase the validity, instructional utility, and credibility of standardized tests. Artificial intelligence methods are invoked to produce item-level partial-credit scores and diagnostic analyses similar to those of a human expert. Finally, cognitively-grounded measurement models provide diagnostic statements based on commonalities in performance across items. Progress toward achieving this conception is examined, with emphasis on two automated scoring programs. Potential applications of intelligent assessment are discussed. (Contains 45 references and 6 figures.) (Author)

RR-90-5

RESEARCH REPORT

TOWARD INTELLIGENT ASSESSMENT: AN INTEGRATION OF CONSTRUCTED RESPONSE TESTING, ARTIFICIAL INTELLIGENCE, AND MODEL-BASED MEASUREMENT

Randy Elliot Bennett

Educational Testing Service
Princeton, New Jersey
May 1990

Toward Intelligent Assessment: An Integration of Constructed Response Testing, Artificial Intelligence, and Model-Based Measurement

Randy Elliot Bennett

Educational Testing Service

To appear in N. Frederiksen, R. J. Mislevy, and I. Bejar (Eds), Test theory for a new generation of tests, Hillsdale, NJ: Lawrence Erlbaum Associates.

Abstract

A new assessment conception is described that integrates constructed-response testing, artificial intelligence, and model-based measurement. The conception incorporates complex constructed-response items for their potential to increase the validity, instructional utility, and credibility of standardized tests. Artificial intelligence methods are invoked to produce item-level partial-credit scores and diagnostic analyses similar to those of a human expert. Finally, cognitively-grounded measurement models provide diagnostic statements based on commonalities in performance across items. Progress toward achieving this conception is examined, with emphasis on two automated scoring programs. Potential applications of intelligent assessment are discussed.

Toward Intelligent Assessment: An Integration of Constructed Response Testing, Artificial Intelligence, and Model-Based Measurement

This paper describes a new testing conception called intelligent assessment. The paper outlines a formulation and rationale for this conception, describes progress toward realizing it, and offers some ideas on practical applications including ones related to large-scale standardized testing programs.

Intelligent Assessment

Intelligent assessment is conceived of as an integration of three research lines, each dealing with cognitive performance from a different perspective: constructed-response testing, artificial intelligence, and model-based measurement. This integration is envisioned as producing assessment methods consisting of tasks closer to the complex problems typically encountered in academic and work settings. These tasks will be scored by automated routines that emulate the behavior of an expert, providing a rating on a partial-credit scale for summative purposes as well as a qualitative description designed to impart instructionally useful information. The driving mechanisms underlying these tasks and their scoring are cognitively-grounded measurement models that may dictate what the characteristics of items should be, which items from a large pool should be administered, how item responses should be combined to make more general inferences, and how uncertainty should be handled.[1]

It is important to stress that the emphasis is on assessment that facilitates instruction rather than on instruction that embeds assessment, as some intelligent tutoring systems at least implicitly do (J. R. Frederiksen & White, 1988). This emphasis was chosen to encourage developments that might result in near-term, incremental improvements to major standardized testing

programs (such as achievement and admissions programs), in more sophisticated environments for preparing for such tests, and, in the longer term, in more effective assessment modules for intelligent tutoring systems (Wenger, 1987).

Complex Constructed-Response Tasks

A constructed response task can be thought of as any task for which the space of examinee responses is not limited to a small set of presented options. As such, the examinee is forced to formulate, rather than recognize, an answer. This definition implies a substantial range in the complexity of the responses, from an item that calls for rearranging the sentences in a paragraph to one that requires performing a musical piece (Bennett, Ward, Rock, & LaHart, in press). In this paper, interest is focused toward responses of greater complexity. Hence, a complex constructed-response item is one for which scoring decisions cannot typically be made immediately and unambiguously, using mechanical application of a limited set of explicit criteria, but rather require some degree of expert judgment. Figure 1 presents examples of such items in the algebra and computer programming domains.

Insert Figure 1 about here

The use of such items may engender many difficulties. Surely, more multiple-choice questions can be completed per unit time than complex constructed responses. Tests composed of the latter item type will often have reduced content coverage and, potentially, lower reliability and validity (as seen from the traditional test theoretic perspective). Scoring constructed responses for any large-scale testing program is a major undertaking: detailed, defensible scoring keys must be produced; human judges trained,

housed, and fed while scoring is underway; and rater performance must be constantly monitored to maintain inter-judge agreement and scale stability.

Given these difficulties, why be interested in such items at all? For one, these items are likely to measure skills different from those tapped by multiple-choice tests (Ackerman & Smith, 1988; Ward, N. Frederiksen, & Carlson, 1980). Because constructed-responses may more closely represent real-world tasks, they should more readily engage many of the higher-order cognitive processes required in academic and work settings. As a result, important constructs that are presumably not measured by multiple-choice tests are likely to be assessed. Such increases in construct validity may, in turn, lead to enhanced predictive value (N. Frederiksen & Ward, 1978), especially when constructed responses are combined with multiple-choice items (Breland, Camp, Jones, Morris, & Rock, 1987).

A second argument for using complex constructed-response items is that the responses provide a window onto the strategies examinees use in arriving at a solution. This window facilitates the gathering of diagnostic information not easily attainable in the multiple-choice format (Birenbaum & Tatsuoka, 1987).

Third, it has been argued that constructed-response questions play an important role in making the outcomes of instruction more clearly visible (Breland, Camp, Jones, Morris, & Rock, 1987; J. R. Frederiksen & Collins, 1989). As implied, multiple-choice tests in many instances attempt to measure skill using a format different from those commonly encountered in real-world tasks. Even when substantially similar cognitive operations underlie the two formats, instruction would seem less efficient as teachers (and students) use one format to achieve the objectives of the curriculum and the other to prepare for the test. The real danger comes in those situations where the

multiple-choice format requires a substantially different (and less valued) set of operations than the criterion performance, and test preparation is emphasized over curricular achievement (J. R. Frederiksen & Collins, 1989; N. Frederiksen, 1984)--as, in high-stakes testing situations, it inevitably will be. The result might be a class of students good at recognizing isolated facts but poor at integrating the knowledge, skill, and strategies needed for more involved tasks. Complex constructed-response items should eliminate this conflict by focusing both instruction and assessment on the same criterion tasks.

Finally, because it looks different from--and more mechanical than--real-world tasks, the multiple-choice format is easily characterized as irrelevant and trivial (e.g., Fiske, 1990). The persistency and frequency with which these characterizations have been made would seem to reduce the credibility of conventional testing programs. Such characterizations would be much less convincing if the test and the criterion tasks were more similar.

The research literature underlying these purported advantages of complex constructed-response is quite limited, leading to few definitive conclusions (Traub & MacRury, in press). Though there is, for example, considerable evidence that different abilities are demanded by multiple-choice and free-response items, the nature of these differences is not clearly understood. Even so, the arguments for constructed response appear logically and often theoretically well-grounded, justifying continued exploration of this format as, at the least, a potential supplement to multiple-choice items.

Scoring

The second feature in the proposed conception of intelligent assessment is evaluative feedback approximating the analyses of an expert. For summative purposes, an expert would be expected to rate a complex constructed response

on a partial-credit scale. To help the learner, a qualitative analysis of the response should also be provided.

Why score responses on a partial-credit scale? Though partial-credit models for scoring multiple-choice items have been proposed (Millman & Greene, 1989), standardized tests have traditionally treated responses to these items as right or wrong. With a complex response, dichotomous scoring throws a rich data base away thereby potentially reducing test reliability and validity.

Figure 2 presents several incorrect answers generated by GRE General Test examinees to an algebra word problem (Sebrechts, Bennett, & Rock, in press). One means of decomposing the problem is in terms of the following goals: (1) find the time for the first part of the trip, (2) find the missing distance for the second part, (3) find the time for the second part, (4) add the two times, and (5) add the total trip time to the starting time. The first response recapitulates this decomposition exactly. The answer is wrong only because a minor computational error is made in finding the missing distance for the second part (line 4).

Insert Figure 2 about here

In the second response, three of the five goals are correctly structured. The missing distance is generated (as implied from its use in line 7), the two travel times (though incorrectly derived) are summed (line 13), and the total is added (implicitly) to the start time to produce a finish time. Structural flaws are present, however, in finding the travel times (lines 1-5 and 7-11).[2]

The last response is the most seriously wrong. This response achieves only one of the five goals, find the missing distance, which is indicated on line 4.

These three responses differ considerably in the extent to which they approach a correct problem solution, as well as in the understanding of the problem they imply. Yet each result would have been treated equivalently under a dichotomous scoring scheme.

In addition to partial credit, an expert would be expected to generate a qualitative analysis to give individual examinees an indication of where they went wrong in solving the problem. In applications of intelligent assessment within major standardized testing programs, this feedback might simply serve as a more complete explanation of test performance than scores can provide. In a test preparation system, such feedback would hopefully have more tangible value, giving examinees the information needed to modify their solution strategies.

Figure 3 gives examples of item responses that, while very close to a correct solution, contain qualitatively different errors. Response #1 contains only a simple computational error (line 4). In Response #2, the error is in considering 6.3 to be the same as 6 and 1/3 (lines 7 and 9). Response #3's error is in lines 10-11 where 6.30 hours is mistakenly transformed to 6 hours 30 minutes.

Insert Figure 3 about here

Cognitively Motivated Measurement Models

Though statements about what an examinee did incorrectly on a given item might be helpful, it would seem that far greater instructional value would

accrue from more general diagnostic statements based on commonalities in performance across items. Such commonalities are more likely to be indicative of stable errors associated with particular skill deficiencies. To guide aggregations across items, a model, or general set of rules applicable to a class of assessment purposes, can be used. Measurement models guided primarily by the semantics of the domain should offer more efficient and psychologically meaningful statements than ad hoc approaches to aggregation (Masters & Mislevy, in press). In addition, they might suggest, among other things, the cognitive and psychometric characteristics of items, which of a large pool of items should be administered, the order in which they should be given, and how noise in the data should be dealt with.

Progress Toward Intelligent Assessment

Scoring[3]

Perhaps the greatest problem in operationalizing intelligent assessment lies in scoring. Complex constructed-response items are used routinely in some large-scale testing programs (e.g., the College Board's Advanced Placement Program). In these programs, responses are scored by human experts. This method is extremely expensive because judges must be trained, fed, housed, and paid. Because of the tremendous volume (responses in the hundreds of thousands for the Advanced Placement Program), scoring is time consuming, taking hundreds of experts a week or more to complete. Qualitative analyses are not included and, given the additional time this analysis would take, it is unlikely that they ever could be. Finally, an appreciable degree of error is introduced in part because even if a reasonably objective scoring rubric can be established, judges' accuracy in consistently applying this rubric often changes during an operational grading (Braun, 1988). These difficulties argue for attempts to create machine-based methods for scoring complex tasks.

Machine-scorable tasks would reduce the operational expense associated with human grading, speed up the scoring process, permit the introduction of qualitative analyses, and eliminate within-judge inconsistency (because the machine would apply its rules the same way every time).

Our work toward developing machine-scorable, complex constructed-response tasks is proceeding in two domains, computer programming and algebra, using two major standardized testing programs, the College Board's Advanced Placement Computer Science (APCS) Examination and the Graduate Record Examinations (GRE) General Test, as settings. In both instances, the underlying scoring mechanism is an expert system--a computer program that emulates one or more aspects of the behavior of a master judge. Both systems were originally built as research tools within the field of intelligent tutoring (Sleeman & Brown, 1982; Wenger, 1987). Because it is centrally concerned with issues related to individualizing instruction, this field has initiated some extremely provocative approaches to instructional diagnosis, offering useful bases for building intelligent assessment.

Computer programming. The expert system being used in our work in this domain is MicroPROUST, a derivative of PROUST (Johnson, 1986; Johnson & Soloway, 1985).[4] PROUST was developed to study the conceptual errors made by students in learning to program in Pascal. MicroPROUST was built as a portable demonstration of the concepts embodied in PROUST and, consequently, is less powerful in its analytical techniques.

MicroPROUST attempts to find non-syntactic bugs in Pascal programs. The system has knowledge to reason about selected programming problems within a framework called intention-based diagnosis (Johnson, 1986; Johnson & Soloway, 1985). Intention-based diagnosis is derived from an extended research program on the development of programming expertise (e.g., Soloway & Ehrlich, 1984;

Soloway & Iyengar, 1986). This research suggests that in debugging programs, experts first attempt to map the program into a deep-structure goal-and-plan representation. Goals are the objectives to be achieved in a program, whereas plans are stereotypical means (i.e., step-by-step procedures) for achieving those goals. Following the lead of experts, MicroPROUST attempts to "understand" student solutions by identifying the goals and plans that the student intended to realize in the program, and the bugs produced. The term "bug" takes its traditional meaning of "programming error" and is conceptualized as an unsuccessful or incorrectly realized plan for satisfying a goal. Bugs located are verbally described to the student in hope of preventing similar errors and of stimulating thinking about how to approach the problem correctly.

To analyze a problem, MicroPROUST must have what is in essence a reasonably deep understanding of the problem domain. Developing this understanding involves a labor-intensive cognitive analysis. For each problem, student responses are analyzed to derive one or more goal decompositions, correct plans for achieving the goals, and bug rules representing faulty plan implementations.

In evaluating a student solution, MicroPROUST first reads the problem specification contained in its knowledge base. This specification identifies the goals the student should be attempting to achieve in solving a particular problem. The system uses this goal specification, its plan and bug rule knowledge bases, and the student's code to construct the solution intended by the student. For example, the specification for the "rotate array" problem (see Figure 1b) includes the goal, "shift each array element to the right." The system would use this goal to locate in its knowledge base a set of plans to achieve the desired result. As of this writing, MicroPROUST's knowledge

base had 24 such plans which varied in their loop control structures, how they accomplished the shift (using a single array, a pair of arrays, or a pair of temporary variables), and so on.

Next, the system would attempt to match one of these plans to a portion of the student's code. If a match is found, inferences about the student's intentions with respect to this code segment can be made, for instance, what meaning to attribute to particular variables. On the basis of these inferences, the system can predict how these variables will be used in achieving the next goal needed to satisfy the problem specification and, in addition, where in the program relative to the current segment that next goal should reside. If an appropriate code segment cannot be found for achieving that next goal, an attempt is made to match the segment using buggy-plan rules. This goal-plan matching strategy provides considerable leverage; correct plans and bug rules can be juxtaposed in different combinations to handle the variety of responses generated by novice programmers.

Our first study with MicroPROUST examined the extent of agreement between the program and human readers in diagnostically and numerically scoring solutions to each of two APCS programming problems (Bennett, Gong, Kershaw, Rock, Soloway, & Macalalad, in press). Each problem asked the student to write a short program or procedure to satisfy a given specification (see the problem statement in Figure 1a). The rules employed by the high school and college teachers who operationally grade the APCS exam were used to develop a rubric for MicroPROUST that deducted points depending upon the particular bug detected. Solutions were graded by MicroPROUST and by a sample of teachers drawn from the grading pool. MicroPROUST was able to produce an analysis for approximately 70% of the solutions (it offered no analysis on the remaining papers), and for those programs it could analyze, its performance

was indistinguishable from human readers for one problem and not dramatically different from them for the second (the correlations between the machine and mean rater scores for the two problems were .96 and .75). In a cross-validation sample, however, the percentage of papers the program was able to analyze was considerably lower at 42%, though, again, its scores were very similar to a human judge's.

To improve MicroPROUST's performance, especially with respect to the percentage of solutions analyzed, several approaches might be tried. Our second study looked at one such approach, constraining the constructed-response task (Braun, Bennett, Frye, & Soloway, in press). The subjects for this study were two samples of examinees taking the APCS examination. Each sample was given the same problem specification as in the previous study, but also an incorrect solution to the problem. The task was to correct the faulty program instead of writing it ab initio. The corrected programs were scored by MicroPROUST (without any change to its knowledge base to adapt it to this new problem type). Results showed, first, a substantial increase in the percentage of solutions analyzed--from 42% to 83%. Second, those solutions that MicroPROUST could not analyze were almost always incorrect: 93% had one or more bugs. Third, reasonable agreement was found between MicroPROUST's scores and a human rater--the product-moment correlation was .86. Finally, whereas agreement on scores was good, agreement with the rater on bug diagnosis was more moderate: the two agreed on the exact nature and location of individual bugs in 56% of cases. Further work to identify the causes of this disagreement needs to be undertaken.

Our third study focused on the construct validity of MicroPROUST's scores for the constrained free-response item, specifically on whether this item functioned more like multiple-choice or free-response questions (Bennett,

Rock, Braun, Frye, Spohrer, & Soloway, in press). This issue is important because our goal was to produce an item type that, while machine-scorable, retained the cognitive demand characteristics of free-response. To address this issue, data from the two samples of APCS examinees used in the previous study were analyzed. (In that study, the two samples were distinguished by having taken constrained free-response items that differed in the number of seeded bugs: 1 vs. 3.) Confirmatory factor analysis (Joreskog & Sorbom, 1988) was applied to the data to estimate the relationships among the three items types (multiple-choice, free-response, constrained free-response). Results suggested that the proficiency information distilled from the three item types was essentially the same, implying that the constrained free-response might constitute a reasonable supplement to the two existing APCS item formats.

Our ongoing work in computer programming is directed at several goals. The first is to understand better the reasons for the functional similarity of the three item types by undertaking additional studies of their structural relations and cognitive demand characteristics. Second, a successor to MicroPROUST is being developed that will contain more items and item formats, and larger knowledge bases. Rather than operating in "batch" mode as MicroPROUST does, the successor will be linked to a standard programming environment, thereby forming an interactive system that presents programming problems, accepts responses, and provides immediate feedback. Finally, a program is being built to make constructing plan and bug knowledge bases easier, so that greater coverage of student solutions--and higher analysis rates--can be efficiently achieved.

Algebra. Our work in this domain is centered upon building constructed-response formats for algebra word problems adapted from the GRE General Test.

Student solutions to these problems differ fundamentally from programming solutions (Sebrechts & Schooler, 1987). First, steps are frequently left out, oftentimes because they can be mentally computed. Second, syntax is considerably looser: students use assignment to values, include free (unbound) expressions, and occasionally use multiple symbols to represent the same variable or the same symbol to represent different variables. Finally, the algebraic expressions that compose a solution typically culminate in a single, easily verifiable result. The nearest analogue in programming is an output, which varies as a function of the input and which cannot be generated without compiling the solution. These characteristics of students' algebra problem solutions make the task of scoring and diagnosis significantly different from that in programming.

To score solutions to algebra word problems, GIDE, also a derivative of PROUST, is being used (Sebrechts, LaClaire, Schooler, & Soloway 1986). In keeping with the nature of students' solutions, GIDE was constructed to accept productions in relatively unconstrained forms.[5] Solutions must be written linearly (though not in a strict order), any names can be used to identify variables or constants, and there is no restriction on the degree to which examinees are allowed to deviate from a correct solution path. Though this lack of constraint makes solutions substantially more difficult to interpret, it appears to be more consistent with the ways in which problems are solved in real-world settings.

Like MicroPROUST, GIDE can only analyze responses to problems about which it is knowledgeable. GIDE's algebra word problem knowledge presently is enough to handle responses to several variants of three basic problems (see Figure 1b for an example). The knowledge base for these problems was developed by asking ETS mathematics test developers to specify correct and

incorrect problem solutions, and by analyzing the written solutions and think-aloud protocols of university undergraduates.

GIDE's evaluation of responses is guided by several strategies. As with MicroPROUST, these strategies help it to build an understanding of the response in terms of a goal-plan structure. For example, consider the "rate x time" problem in Figure 1b, which can be decomposed into the following goals:

(1) find the time for the first part of the trip,

(2) find the missing distance for the second part,

(3) find the time for the second part,

(4) add the times for the two parts to get a total time, and

(5) add the total trip time to the starting time.

In determining if a goal is satisfied, GIDE will attempt to match one of the several plans it has for that goal to a portion of the examinee's solution. GIDE does this by matching plans for the form and numerical value of equations (e.g., for goal #2, part 2 distance = 600 miles - 285 miles, 315 = 600 - 285), for free-standing expressions (e.g., 315 appearing in isolation as the result of a mental computation), and for groups (e.g., when a goal consists of a list of elements that can take on any order, such as group of numbers to be summed). As a result of this matching, values as well as names used by the examinee to represent variables or constants, are bound to GIDE's internal representations. These bindings are available for use in analyzing subsequent goals.

As part of GIDE's analyses, it attempts to separate conceptual from computational errors. It does this by noting instances in which erroneous values are associated with correct symbolic forms and then carrying these computational errors through to subsequent goals. So, for example, in Response #1 in Figure 3, the examinee incorrectly computed the distance for

part 2. GIDE would assign the result of this incorrect computation (i.e., 415) to its internal representation of the part 2 distance for all remaining computations. In this way, GIDE is able to determine if the rest of the solution is conceptually correct and if a wrong answer was produced only because of a low-level mistake.

The above strategies are successful as long as plans representing conceptually correct solutions to the active goal can be matched to portions of the examinee's solution. If such a plan cannot be matched to the examinee's solution, GIDE attempts to match to the solution plans that incorporate conceptual errors commonly made in achieving that goal or bug rules that, in GIDE, represent more general errors.

In some cases, however, none of these plans or rules match an examinee solution. When this occurs, GIDE searches the proposed solution for a name it has associated with the current active goal to find a clue as to what the examinee was doing. For goal #2 of the above problem (the distance for part 2), such names might include "part 2 distance," "distance 2," "missing distance," and "dist 2." If such a name is found, GIDE checks the associated expression and result to see if they can reasonably be considered deviations from a correct plan for that goal.

In those cases where GIDE is not able to account for how the examinee has attempted to satisfy the goal through either correct or incorrect plans, GIDE waits to see if satisfaction of subsequent goals will fulfill the currently active goal implicitly. Implicit matching is triggered when explicit matching has failed and a dependency link is active; that is, a goal presumes the satisfaction of one or more prior goals. So, for example, in the above problem, if plans for goals #1-3 (find the missing distance and calculate the times for the two parts) are not matched--perhaps because the

examinee did the computations mentally--but a plan satisfying goal #4 is matched (add the two times together), goals #1-3 would be matched implicitly.

Finally, when no plan, buggy or correct, can be matched to a portion of the solution, the goal is considered missing. GIDE, in essence, "understands" this solution component to be physically absent, a presumption which will likely be correct if the domain analysis has been well done.

After completing its analysis, GIDE issues a brief bug report and a partial-credit score. The bug report identifies the errors detected. Because of its experimental nature, GIDE's algebra bug reports are relatively unrefined, giving only enough detail to permit verification of an error's existence by an independent source. In any operational implementation, these descriptions would need to be carefully crafted to communicate clearly the nature of the error and perhaps a method for resolving it.

GIDE's partial-credit scores are derived from goal-plan analysis. This linkage is meant to give the scores a principled, cognitive basis. The rubric awards full credit if all goals are achieved, suggesting the student was able to decompose the problem, correctly structure each goal, and compute its solution. Credit is deducted differentially depending on the errors detected for each goal. The largest deduction is made for missing goals because these absences suggest the student was unaware that addressing the goal was necessary to achieving a correct result. Less credit is deducted for conceptual bugs because such bugs suggest both recognition of the goal's importance and a coherent, though incorrect, attempt to solve the goal. The smallest deduction is for computational errors which imply only trivial procedural slips.

Because students may approach a problem using an alternative decomposition (see Figure 4), GIDE has the capability to process solutions

against such alternatives. The mechanisms for handling alternatives are, however, largely ad hoc and represent one area for the program's further development.

Insert Figure 4 about here

GIDE's performance in analyzing responses to algebra word problems has not yet been evaluated. Its success in diagnosing errors in statistical problem-solving, though, has been reported (Sebrechts & Schooler, 1987). In a sample of 60 responses, GIDE was able to account for approximately 82% of the lines and 95% of the goals, an imperfect but promising performance.

Cognitively Motivated Measurement Models

In their current experimental states, GIDE and MicroPROUST produce analyses only for item-level responses. That is, scores and diagnostic comments are restricted to performance on a single item. These scores and comments have potential value for describing how an examinee did on that item and perhaps for helping him or her avoid those same mistakes next time. But, as noted, more dependable statements about an examinee's skills might be derived from model-based aggregations of performance made across constructed-response tasks.

Several approaches can be taken to response modelling. In psychometrics, methods like item response theory (Lord, 1980) have been built on purely statistical foundations. As Mislevy (in press) notes, these approaches work well for some assessment purposes (e.g., selection) and far less well for others (e.g., instructional diagnosis).

Intelligent tutoring, in contrast, has focussed on developing models incorporating an understanding of the domain in which responses are to be

aggregated (Wenger, 1987). As a result, these models promise interpretations of performance that are more clearly tied to instructional decisions. At the same time, however, these deterministic formulations generally do not deal well with the inconsistency that often characterizes human performance (Wenger, 1987).[6]

Given this situation, it would seem sensible to work toward some combination of probabilistic methods and the cognitively based diagnosis exemplified by intelligent tutoring. As suggested by Masters and Mislevy (in press), the probabilistic methods should be subservient to cognitive considerations: domain semantics should shape the model's application in any given case.

Several recent measurement models attempt to fill this requirement, including the Hierarchically-Ordered Skills Test (HOST) model (Rock & Pollack, 1987), the Hybrid model (Yamamoto, 1987), and Masters' Partial Credit Model (Masters & Mislevy, in press).[7] A brief summary of the first two models is given here as an introduction to how they might be applied in intelligent assessment (for more complete descriptions see Gitomer & Rock, in press, and Yamamoto, in press).

In the HOST model, groups of items are written to represent levels of proficiency, with each succeeding level requiring one or more new cognitive operations in addition to those of the preceding level. If the model fits, standing on the scale denotes what operations the examinee is and is not able to perform. Because individuals often come to proficiency in an area by different paths, the model provides a measure of fit for each examinee. When the model does not fit an examinee's performance, that performance can usually be placed on a more general ability scale. In addition to measures of individual fit, the model provides estimates of the probabilities associated

with being at particular skill levels. These probabilities have proven particularly useful for measuring individual change because the probabilities seem less sensitive than other metrics to the ceiling and floor effects that have perennially hampered attempts to measure individual growth (Rock & Pollack, 1987).

Rock has studied the fit of the HOST model to mathematics achievement data from the 1980 sophomore High School and Beyond (HS&B) cohort and from the population taking the SAT (Rock & Pollack, 1987; Gitomer & Rock, in press). In these studies, the overwhelming majority of examinees fit the model: 90% for the HS&B sample and 96%-98% for the SAT sample. Further, the model fit equally well for males and females, and for majority and minority examinees.[8]

The second approach, Yamamoto's (1987) Hybrid model, combines latent class models with item response theory (IRT). Latent class models are built on the idea of a categorical latent variable (Lazarsfeld, 1960). Because a hierarchy of classes is not required, information can be provided about unordered qualitative states that characterize examinees (e.g., a tendency toward a specific error type). In addition, the probability that an examinee's response pattern belongs to a given class is provided.

In practice, not all examinee response patterns can be captured by a limited set of classes. More classes may exist than are represented in the model, or individuals may respond in an extremely inconsistent fashion. Performance that does not fit one of the hypothesized latent classes may be modeled by a continuous model that makes no assumptions about examinees' qualitative understandings. The Hybrid model accounts for this eventuality by scaling these examinees along a general dimension underlying a problem set, while simultaneously providing diagnostic information for those individuals who fit a latent class.

The performance of the Hybrid model has been assessed using data on electronic technicians' ability to interpret logic gate symbols (Gitomer & Yamamoto, 1988). Five latent classes were represented based on specific errors commonly made by technicians. The model's latent class portion was able to capture 36% of the response patterns, a very respectable performance given the specificity of the error classes. In addition, for individuals picked up by the latent classes, the distinction among error classes given particular response patterns was quite sharp, making class assignments very clear. Finally, the probability of belonging to any latent class was unrelated to overall ability estimates, supporting the model's capacity to represent qualitative states.

Depending upon the domain and the assessment purpose, either the HOST or the Hybrid models might be used. Alternatively, they might be employed together to provide complementary aggregations of item information. Figure 5 shows four algebra item formats hypothesized to form a hierarchical ordering (Sebrechts, Bennett, & Rock, in press). The formats are open ended (only the problem stem is presented), goal specification (the problem stem, a list of givens, and a list of unknowns, or goals, is presented), equation setup (the problem stem and the equations, or plans, needed to derive the unknowns are given), and faulty solution (the stem and an incorrect plan are presented for the examinee to correct). The problems presented in each format are isomorphs (i.e., the same solution process can be applied to all four problems). A theoretical justification of the hierarchy is presented in Figure 6 as a list of cognitive operations suggested to underlie each proficiency level. If the HOST model fit a complete test built around this illustration, examinees who successfully completed items in the open-ended format would generally succeed with the other formats (though the reverse would not necessarily be true).

The operations underlying the levels would form the basis for diagnostic statements that might be made about individuals whose performance fit the model.

Insert Figure 5 about here

Insert Figure 6 about here

The Hybrid model might be used on this same test to give information about the error class to which an examinee's performance belongs. As with HOST, this classification is semantically driven: error classes derive from the domain and the nature of examinee performance, not directly from the measurement model. In Figure 3, three qualitatively different errors were illustrated in computation, transforming decimals to fractions, and converting units. These errors could be represented by eight classes: (1) computation errors only, (2) transformation errors only, (3) unit conversion errors only, (4) computation and transformation only, (5) computation and unit conversion only, (6) transformation and unit conversion only, (7) all three error types, and (8) none of the types. Examinees whose response patterns place them into one of the seven error classes can be identified, depending on the class, as needing a specific type of attention if success in the domain is to be achieved.

Potential Applications

The three components of intelligent assessment--complex constructed response, intelligent scoring, and cognitively driven measurement models--are in different states of readiness for operational use. Complex constructed

response has, of course been employed for quite some time in large scale testing programs such as the College Board's Advanced Placement Program. As a result, much practical experience has accumulated about the item type's development, administration, and scoring using human judges. As noted, however, the item format's measurement characteristics have not been fully explored. Though these items are unarguably more "direct" measures of the constructs schools aim to teach (J. R. Frederiksen & Collins, 1989), whether they are in reality more valid measures of these constructs remains an open question (Traub & MacRury, in press).

Methods for automatically scoring complex constructed responses are generally not ready for operational use. For example, neither GIDE nor MicroPROUST can accurately score all the responses encountered. Other scoring systems are in a similar state (e.g., Bejar, 1988; Freedle, 1988). Due to the diversity of human performance, perfect accuracy may be far in the future.

Finally, the cognitively motivated measurement models required to support this notion of assessment are only beginning to emerge (Mislevy, in press). A considerable period of research will likely be required before these models begin to see widespread use.

Even though the foundations for intelligent assessment are not yet firmly established, enough progress has been made to justify building some initial applications, the study of which should begin to provide the knowledge to support operational realizations. Three ideas are discussed, ranging from a heavily constrained implementation that could be quickly built to a fully featured intelligent assessment system that may take many years to construct. Each idea is structured so as to explore some of the central issues in intelligent assessment.

The least ambitious idea requires as context a computer-delivered testing program. Several such programs exist (e.g., the College Board's Computerized Placement Tests), and more are under development ("ETS research plan," 1989). Many of these programs will be computerized adaptive tests (Wainer, Dorans, Green, Flaugher, Mislevy, Steinberg, & Thissen, 1990). Computerized adaptive tests dynamically home in on the estimated skill level of the examinee, presenting fewer but more informative items than conventional tests. Consequently, they take less time to administer while maintaining the content coverage and reliability of paper-and-pencil analogues.

One profitable way to use some of this saved time might be to supplement the multiple-choice item pool with a small number of intelligently-scored complex constructed-response items, such that each examinee encounters one or two of them. In the event that the constructed responses were found to measure the same trait as the rest of the test, a plausible occurrence in some instances (Bennett, Rock, Braun, Frye, Spohrer, & Soloway, in press; Traub & Fisher, 1977; Ward, 1982), all items might be placed on the same IRT scale using, for example, the Partial Credit Model (Masters & Mislevy, in press). Item parameters would be used not only to select constructed responses appropriate to the examinee's skill level, but also might be employed in scoring the test. Though chosen adaptively, the constructed-response items would be presented last to avoid the cognitive disruption that might occur from mixing item formats. If after analyzing the solution the expert system was able to account for each goal, a report of the examinee's constructed-response performance would be displayed. This application leaves content coverage intact and provides the examinee with information beyond the total test score. The effects of any potential scoring inaccuracy are mitigated because item-level feedback is provided (and factored into overall test score)

only if all parts of the examinee's production are explained. Finally, the application gives some visibility (though far less than deserved) to the behaviors that should be the focus of instruction, a feature particularly important for achievement, college admissions, and other large-scale programs that can influence school curricula.

A second potential application is a self-assessment intended to help develop skills and prepare students for a particular standardized test. The self-assessment should make visible the standards for domain performance so that the student can internalize and use them for judging his or her own productions (J. R. Frederiksen & Collins, 1989). In the APCS program, such an assessment might be built around a pool of free-response and constrained free-response programming problems that students could access on demand. Responses to the problems would be analyzed by an expert system, such as MicroPROUST or its planned successor. For incorrect solutions, the system would print not only a diagnostic analysis and a partial-credit score, but a goal-plan decomposition and the rules for judging the item using that decomposition. The student might then be given two tasks: (1) to verify that the system's analysis was correct and (2) to revise the solution utilizing the system's comments and the problem's goal-plan decomposition. For instances in which the system was not able to produce an analysis, the goal-plan decomposition would be printed with a direction to consult the classroom teacher (or perhaps a more skilled peer). Student and teacher might then collaboratively analyze the solution to see how it diverged from the goal-plan decomposition, bringing to light other legitimate approaches to the problem or rare errors beyond the system's understanding. In this instantiation, the system's inability to flawlessly analyze all responses is a virtue: it forces the student to seek others' counsel, hopefully encouraging both collaborative problem solving and

the internalization by student and teacher of goal-plan analysis as one approach to problem solution.

The last potential application is a model-governed intelligent assessment system for instructional diagnosis. Such a system might complement the College Board's Computerized Placement Tests (CPTs), which are used to select students needing remedial instruction from the freshmen class entering an institution.

At the front end of this system would be an adaptive, multiple-choice assessment module. This module's purpose would be to estimate efficiently the examinee's general skill level in the domain so that appropriate constructed-response tasks could be presented. Accurate assignment is critical not only because responses to overly difficult items will almost invariably be wrong, but because those responses will usually be severely flawed and, as a result, indecipherable by an expert scoring system. Responses to items that are too easy will likely be correct, and though decipherable, will contribute no useful information. This module would not need to be used if a skill estimate was already available from a companion test, such as the CPTs.

The second component, the constructed-response module, would need to be built from a deep understanding of the domain. It would be composed of problems requiring the application and integration of key knowledge and skill. On top of this domain structure, would rest a measurement model, like HOST or Hybrid, able to generate diagnostically useful information. On the basis of constructed-response performance, the model would generate hypotheses about the examinee's proficiency. Both because of inconsistencies in examinee performance and because any expert scoring system will sometimes fail to understand a production, these hypotheses would in many cases be based on incomplete or contradictory information. To reduce uncertainty, the

constructed-response module would pass its list of competing, plausible hypotheses to a verification module.

The verification module would be composed of two parts, called upon as necessary. One part would consist of a series of testlets, homogeneous multiple-choice item clusters focused on a specific skill (Wainer & Kiely, 1987). The contents of these testlets would derive from the same comprehensive analysis of the domain that formed the basis for the constructed-response module. Only those testlets that might serve to confirm or disconfirm an active diagnostic hypothesis would be administered. A second module component would ask the examinee for an estimate of his or her understanding of the skills in question and use this estimate in the verification process, a strategy used in rudimentary ways in several intelligent tutors (Wenger, 1987).

In addition to indicating whether a student is in fact behaving consistently, this verification process might also confirm whether consistently manifested errors (e.g., converting 10.63 hours to 11 hours 3 minutes) represent slips or real misunderstandings (Matz, 1982). In one case, simply pointing out the error to the examinee might resolve it; in the other, a more extended explanation would be required. An assessment system with such verification and feedback capabilities begins to take on J. R. Frederiksen and Collins' (1989) notion of "systemic validity" by helping students improve the skills it is attempting to test.

Conclusion

This paper has presented a conceptualization of intelligent assessment as an integration of constructed-response testing, scoring methods based on artificial intelligence, and cognitively motivated measurement models. To illustrate progress toward this conception, two intelligent scoring systems--

MicroPROUST and GIDE--and two measurement models--HOST and Hybrid--were described. It is worth emphasizing that these approaches take particular perspectives, especially the scoring systems, which derive from the same theoretical base. Other approaches to both scoring and response modelling exist and it is likely to be some time before any individual method becomes generally accepted.

Second, it should be evident that many unresolved issues are associated with intelligent assessment. The development of even the least ambitious realization implies a considerable effort--in domain understanding and knowledge base development, item writing, scoring rules, feedback contents and processes, programming, pilot testing, and validation research, among other things--with no certainty that the result will prove substantially better than current testing approaches. But the purpose of this paper, the work it describes, and research generally is to develop and test new ideas, to discover their effects and the conditions under which they manifest. Only through this inquiry process will we be able to build the innovative assessment systems needed to help shape an educational system that meets the demands of an increasingly complex world.

References

Ackerman, T. A., & Smith, P. L. (1988). A comparison of the information provided by essay, multiple-choice, and free-response writing tests. Applied Psychological Measurement, 12, 117-128.

Bejar, I. I. (1988). A sentence-based automatic approach to assessment of writing: A feasibility study. Machine-Mediated Learning, 2, 321-332.

Bennett, R. E., Gong, B., Kershaw, R. C., Rock, D. A., Soloway, E., & Macalalad, A. (In press). Assessment of an expert system's ability to automatically grade and diagnose students' constructed-responses to computer science problems. In R. O. Freedle (Ed), Artificial intelligence and the future of testing, Hillsdale, NJ: Lawrence Erlbaum Associates.

Bennett, R. E., Rock, D. A., Braun, H. I., Frye, D., Spohrer, J. C. & Soloway, E. (In press). The relationship of constrained free-response to multiple-choice and open-ended items. Applied Psychological Measurement.

Bennett, R. E., Sebrechts, M. M., & Rock, D. A. (In press). The validity of machine-scorable, constructed-response GRE General Test Quantitative items. Princeton, NJ: Educational Testing Service.

Bennett, R. E., Ward, W. C., Rock, D. A., & LaHart, C. (In press). Toward a framework for constructed-response items. Princeton, NJ: Educational Testing Service.

Birenbaum, M., & Tatsuoka, K. K. (1987). Open-ended versus multiple-choice response formats--It does make a difference for diagnostic purposes. Applied Psychological Measurement, 11, 385-395.

Braun, H. I. (1988). Understanding scoring reliability: Experiments in calibrating essay readers. Journal of Educational Statistics, 13, 1-18.

Braun, H. I., Bennett, R. E., Frye, D, & Soloway, E. (In press). Scoring constructed responses using expert systems. Journal of Educational Measurement.

Breland, H. M., Camp, R., Jones, R. J., Morris, M. M., & Rock, D. A. (1987). Assessing writing skill. New York: College Entrance Examination Board.

Bunderson, C. V., Inouye, D. K., & Olsen, J. O. (1989). The four generations of computerized educational measurement. In R. L. Linn (Ed), Educational measurement (Third Edition). New York: American Council on Education/MacMillan.

ETS research plan designed to create a new generation of Graduate Record Examinations. (1989, February 23). Examiner, 18(26).

Fiske, E. B. (1990, January 31). But is the child learning? Schools trying new tests. The New York Times, pp. 1, B6.

Frederiksen, J. R., & Collins, A. (1989). A systems approach to educational testing. Educational Researcher, 18(9), 27-32.

Frederiksen, J. R., & White, B. Y. (1988). Implicit testing within an intelligent tutoring system. Machine-Mediated Learning, 2, 351-372.

Frederiksen, N. (1984). The real test bias: Influences of testing on teaching and learning. American Psychologist, 39, 193-202.

Frederiksen, N., & Ward, W. C. (1978). Measures for the study of creativity in scientific problem solving. Applied Psychological Measurement, 2, 1-24.

Freedle, R. (1988). A semi-automatic procedure for scoring protocols resulting from a free-response sentence-combining writing task. Machine-Mediated Learning, 2, 309-319.

Gitomer, D. H., & Rock, D. A. (In press). Addressing process variables in test analysis. In N. Frederiksen, R. J. Mislevy, and I. Bejar (Eds), Test theory for a new generation of tests. Hillsdale, NJ: Lawrence Erlbaum Associates.

Gitomer, D. H., & Yamamoto, K. (1988, March). Using embedded cognitive task analysis in assessment. Paper presented at the annual meeting of the American Educational Research Association, San Francisco.

Johnson, W. L. (1986). Intention-based diagnosis of novice programming errors. Los Altos, CA: Morgan Kaufmann.

Johnson, W. L., & Soloway, E. (1985). PROUST: An automatic debugger for Pascal programs. Byte, 10(4), 179-190.

Joreskog, K., & Sorbom, D. (1988). LISREL 7: A guide to the program and applications. Chicago, IL: SPSS Inc.

Lazarsfeld, P. F. (1960). Latent structure analysis and test theory. In H. Gulliksen and S. Messick (Eds), Psychological scaling: Theory and applications. New York: Wiley, 83-86.

Lord, F. M. Applications of item response theory to practical testing problems. Hillsdale, NJ: Erlbaum.

Masters, G. N., & Mislevy, R. J. (In press). New views of student learning: Implications for educational measurement. In N. Frederiksen, R. J. Mislevy, and I. Bejar (Eds), Test theory for a new generation of tests. Hillsdale, NJ: Lawrence Erlbaum Associates.

Matz, M. (1982). Towards a process model for high school algebra. In D. H. Sleeman and J. S. Brown (Eds), Intelligent tutoring systems. London: Academic Press, Inc.

Millman, J., & Greene, J. (1989). The specification and development of tests of achievement and ability. In R. L. Linn (Ed), Educational measurement (Third Edition). New York: American Council on Education/MacMillan.

Mislevy, R. J. (In press). Foundations of a new test theory. In N. Frederiksen, R. J. Mislevy, and I. Bejar (Eds), Test theory for a new generation of tests. Hillsdale, NJ: Lawrence Erlbaum Associates.

Rock, D. A., & Pollack, J. (1987). Measuring gains--A new look at an old problem. Paper presented at the ETS/DoD Conference, San Diego, CA.

Sebrechts, M. M., Bennett, R. E., & Rock, D. A. (In press). Developing and evaluating machine-scorable, constructed-response item types for the GRE General Test Quantitative scale. Princeton, NJ: Educational Testing Service.

Sebrechts, M. M., LaClaire, L., Schooler, L. J., & Soloway, E. (1986). Towards generalized intention-based diagnosis: GIDE. In R. C. Ryan, (Ed), Proceedings of the 7th National Educational Computing Conference. Eugene, OR: International Council on Computers in Education.

Sebrechts, M. M., & Schooler, L. J. (1987, July). Diagnosing errors in statistical problem-solving: Associative problem recognition and plan-based error detection. Proceedings of the Ninth Annual Cognitive Science Society Conference.

Sleeman, D. H., & Brown, J. S. (Eds). (1982). Intelligent tutoring systems. London: Academic Press, Inc.

Soloway, E., & Ehrlich, K. (1984). Empirical investigations of programming knowledge. IEEE Transactions on Software Engineering, 10, 595-609.

Soloway, E., & Iyengar, S. (Eds). (1986). Empirical studies of programmers. Norwood, NJ: Ablex Publishing.

Traub, R. E., Fisher, C. W. (1977). On the equivalence of constructed-response and multiple-choice tests. Applied Psychological Measurement, 1, 355-369.

Traub, R. E., & MacRury, K. (In press). Multiple-choice vs. free-response in the testing of scholastic achievement. In K. Ingenkamp (Ed), Yearbook on educational measurement. Weinheim: Beltz Publishing Company.

Wainer, H., Dorans, N. J., Green, B. F., Flaugher, R., Mislevy, R. J., Steinberg, L., & Thissen, D. (1990). Computerized adaptive testing: A primer. Hillsdale, NJ: Lawrence Erlbaum Associates.

Wainer, H., & Kiely, G. L. (1987). Item clusters and computerized adaptive testing: A case for testlets. Journal of Educational Measurement, 24, 185-201.

Ward, W. C. (1982). A comparison of free-response and multiple-choice forms of verbal aptitude tests. Applied Psychological Measurement, 6, 1-11.

Ward, W. C., Frederiksen, N., & Carlson, S. B. (1980). Construct validity of free-response and machine-scorable forms of a test. Journal of Educational Measurement, 17, 11-29.

Wenger, E. (1987). Artificial intelligence and tutoring systems. Los Altos, CA: Morgan Kaufmann.

Yamamoto, K. (1987). A model that combines IRT and latent class models. Unpublished doctoral dissertation, University of Illinois, Champaign-Urbana.

Yamamoto, K. (In press). Application of a Hybrid model to a test of cognitive skill representation. In N. Frederiksen, R. J. Mislevy, and I. Bejar (Eds), Test theory for a new generation of tests. Hillsdale, NJ: Lawrence Erlbaum Associates.

Footnotes

1. Bunderson, Inouye, and Olsen (1989) offer a different view of intelligent assessment as the application of artificial intelligence to any of the subprocesses of educational measurement: test development, test administration, and test analysis and research.

2. The examinee's error in computing the travel times was in dividing when a multiplication was called for (lines 3 and 9) and multiplying when division was required (lines 4 and 10).

3. The work described in this section builds upon many years of effort by our principal collaborators, Elliot Soloway and Marc M. Sebrechts, and several years of our own work, with central contributions by Henry I. Braun and Donald A. Rock.

4. The system descriptions in this and the next section are condensed and consequently simplified. Further, the systems are still evolving, so that these descriptions may not accurately reflect the programs' current operation.

5. GIDE was originally designed to diagnose student errors in statistics and automotive mechanics problems.

6. Unfortunately, there have been relatively few attempts within intelligent tutoring to address this issue and, as a consequence, no generally applicable models capable of efficiently handling uncertainty have emerged.

7. Masters and Mislevy (in press) offer examples of other appropriate models.

8. In both the HS&B and SAT studies, the HOST model was fit to a specially chosen subset of items rather than to the complete mathematical scale.

Figure Captions

1. Complex constructed-response items in algebra and programming (with example correct responses).

2. Wrong answers to a complex constructed-response item.

3. Responses of similar correctness but containing qualitatively different errors.

4. Alternative goal decompositions for an algebra word problem.

5. Four item formats hypothesized to form a hierarchical ordering: (a) open-ended, (b) goal specification, (c) equation setup, (d) buggy solution. Note. Print size is reduced, space for writing solutions shortened, and page arrangement modified for publication purposes.

6. Operations suggested to underlie a proposed hierarchical arrangement of item formats.

Figure 1

a. On a 600-mile motor trip, Bill averaged 45 miles per hour for the first 285 miles and 50 miles per hour for the remainder of the trip. If he started at 7:00 a.m., at what time did he finish the trip (to the nearest minute)?

Example Correct Response:

Time 1 = 285 miles / 45 miles per hour
Time 1 = 6.33 hours
Distance 2 = 600 miles - 285 miles
Distance 2 = 315 miles
Time 2 = 315 miles / 50 mile per hour
Time 2 = 6.3 hours
Total time = 6.33 hours + 6.3 hours
Total time = 6 hours 20 min + 6 hours 18 min
Total time = 12 hours 38 min
End time = 7:00 am + 12 hours 38 min
End time = 7:38 pm

Answer: 7:38 pm

b. Write a procedure that rotates the elements of an array s with n elements so that when the rotation is completed, the old value of s[1] will be in s[2], the old value of s[2] will be in s[3],..., the old value of s[n-1] will be in s[n], and the old value of s[n] will be in s[1]. Your procedure should have s and n as parameters. You may assume that the type Item has been declared and s is of type List which has been declared as List = array[1..Max] of Item.

Example Correct Response:

```
program foo (input,output);                    {initialize program}
const
  max = 100;
type
  item = integer;
  list = array[1..max] of item;
var
  PassedAsS : list;
  PassedAsN : integer;
Procedure RotateArray(var s:list; var n:integer);
  var                                          {initialize local variables}
    temp : integer;
    i : 1..max;
  begin
    temp := s[n];                              {move last element to temporary storage}
    for i := n downto 2 do                     {move each element to the right}
      begin
        s[i] := s[i-1];
      end;
    s[1] := temp;                              {move last element from temporary storage to the first position}
  end;
begin
  RotateArray(PassedAsS,PassedAsN);
end.
```

Figure 2

On a 600-mile motor trip, Bill averaged 45 miles per hour for the first 285 miles and 50 miles per hour for the remainder of the trip. If he started at 7:00 a.m., at what time did he finish the trip (to the nearest minute)?

Response #1:

1. 285 miles/45 miles per hr = 6.33 hrs
2. 6.33 hrs = 6 hrs 20 min
3.
4. 600 - 285 = 415 miles
5.
6. 415 miles/50 miles per hr = 8.3 hrs
7. 8.3 hrs = 8 hrs 18 min
8.
9. 6 hrs 20 min + 8 hrs 18 min = 14 hrs 38 min
10.
11. 7:00 + 14 hrs 38 min = 9:38 pm

Answer: 9:38 pm

Response #2

1. 285/x = 45/60
2. 285/x = 3/4
3. 285/4 = 71.25
4. 71.25 * 3 = 213.75
5. 213.75/60 = 3.56
6.
7. 315/x = 50/60
8. 315/x = 5/6
9. 315/6 = 52.5
10. 52.5 * 5 = 262.5
11. 262.5/60 = 4.375
12.
13. 3.56 hours + 4.38 hours = 7.94

Answer: approximately 3 pm

Response #3

1. On a 600 mile motor trip
2. 285 miles = 45 miles/hr
3.
4. 285 + 315 = 600
5.
6. 315 = 50 miles/hr
7.
8. 45 * 50 = 22.50 = 22:50 minutes

Answer: 22:50 minutes

Figure 3

On a 600-mile motor trip, Bill averaged 45 miles per hour for the first 285 miles and 50 miles per hour for the remainder of the trip. If he started at 7:00 a.m., at what time did he finish the trip (to the nearest minute)?

Response #1:

1. 285 miles/45 miles per hr = 6.33 hrs
2. 6.33 hrs = 6 hrs 20 min
3.
4. 600 - 285 = 415 miles
5.
6. 415 miles/50 miles per hr = 8.3 hrs
7. 8.3 hrs = 8 hrs 18 min
8.
9. 6 hrs 20 min + 8 hrs 18 min = 14 hrs 38 min
10.
11. 7:00 + 14 hrs 38 min = 9:38 pm

Answer: 9:38 pm

Response #2

1. 600 mile - 285 = miles at 50 mi/hr
2.
3. time at 45 mi/hr = 285 mi * 1 hr/45 mi
4. time at 45 mi/hr = 6 and 1/3 hrs
5.
6. time at 50 mi/hr = 315 * 1 hr/ 50 mi
7. time at 50 mi/hr = 6.3
8.
9. total time = 12 + 2/3 hrs
10. 2/3 hr = 2/3 hr * 60 min/hr
11. total time = 12 hrs 40 min

Answer: 7:40 pm

Response #3

1. 600 - 285 = 315
2.
3. 45 : 285
4. 50 : 315
5.
6. 285 mi /45 mi per hr = 6.33
7. 315 /50 = 6.30
8.
9. 6.33 hrs : 285 miles
10. 6.30 hrs : 315 miles
11. 6 hrs 20 min + 6 hrs 30 min = 12 hrs 50 min
12.
13. 7 am + 12 hrs 50 min = 7:50 pm

Answer: 7:50 pm

Figure 4

The active ingredient is 0.25 percent of a 3-ounce dose of a certain cold remedy. What is the number of doses a patient must take before receiving the full 3 ounces of the active ingredient?

Correct Answer = 400 doses

a. 1. 0.25% = .0025
   2. Active Ingredient per dose = .0025 * 3 oz
      Active Ingredient per dose = .0075 oz
   3. Number of doses required = 3 oz/.0075 ,oz per dose
      Number of doses required = 400 doses

b. 1. .25%x dose = 100% dose
      x dose = 100% dose/.25% dose
      x = 400 doses

Figure 5

a. On a 600-mile motor trip, Bill averaged 45 miles per hour for the first 285 miles and 50 miles per hour for the remainder of the trip. If he started at 7:00 a.m., at what time did he finish the trip (to the nearest minute)?

_____
_____
_____

ANSWER:_____

b. 800 gallons of a 2,400 gallon tank flow in at the rate of 75 gallons per hour through a clogged hose. After the hose is unclogged, the rest of the tank is filled at the rate of 250 gallons per hour. At what time to the nearest minute will the filling of the tank be finished if it starts at 5:30 a.m.?

Givens

Tank Capacity = _____
Filling Rate 1 = _____
Filling Amount 1 = _____
Filling Rate 2 = _____
Start Time for Filling = _____

Unknown

Filing Time 1 = _____
Filling Amount 2 = _____
Filling Time 2 = _____
Total Filling Time = _____
Ending Time for Filling = _____

ANSWER:_____

c. Of the 720 pages of printed output of a certain program, 305 pages are printed on a printer that prints 15 pages per minute and the rest are printed on a printer that prints at 50 pages per minute. If the printers run one after the other and printing starts at 10 minutes and 15 seconds after the hour, at what time to the nearest second after the hour will the printing be finished?

Equations that Will Provide a Solution:

Time for Printing on Printer 1 = Number of Pages on Printer 1 / Printing Rate of Printer 1
Number of Pages on Printer 2 = Total Number of Pages - Number of Pages on Printer 1
Time for Printing on Printer 2 = Number of Pages on Printer 2 / Printing Rate of Printer 2
Total Printing Time = Time for Printing on Printer 1 + Time for Printing on Printer 2
Time Print Job Finished = Starting Print TIME + Total Printing Time

Your Solution:

_____
_____
_____
_____

ANSWER:_____

d. A Department of Transportation road crew paves 15 mile city portion of a 37.4 mile route at the rate of 1.8 miles per day and paves the rest of the route, which is outside the city, at a rate of 2.1 miles per day. If the Department of Transportation starts the project on day 11 of its work calendar, on what day of its work calendar will the project be completed?

Time for Portion 1 = 15 miles/1.8 miles per day
Time for Portion 1 = 8 and 1/3 days
Time for Portion 2 = 37.4 miles/2.1 miles per day
Time for Portion 2 = 17.81 days
Total Time = 8.30 days + 17.81 days
Total Time = 26.11 days
Completion Day = 27

Your Corrected Solution:

_____
_____
_____
_____

ANSWER:_____

Figure 6

Level	Format	Operations
4	Open ended	Identify givens and unknowns. Create representation for problem based on knowns and unknowns. Map equations onto problem statement. Solve equations. Check solution, detect error(s), and recover.
3	Goal specification	Create representation for problem based on knowns and unknowns. Map equations onto problem statement. Solve equations. Check solution, detect error(s), and recover.
2	Equation setup	Map equations onto problem statement. Solve equations. Check solution, detect error(s), and recover.
1	Buggy solution	Check solution against problem statement, detect error(s), and recover.