A's for All (as Time and Interest Allow)

Dan Garcia
UC Berkeley
Berkeley, CA, USA
ddgarcia@berkeley.edu

Armando Fox
UC Berkeley
Berkeley, CA, USA
fox@berkeley.edu

Solomon Russell
El Camino College
Torrance, CA, USA
srussell@elcamino.edu

Edwin Ambrosio
El Camino College
Torrance, CA, USA
eambrosio@elcamino.edu

Neal Terrell
CSU Long Beach
Long Beach, CA, USA
neal.terrell@csulb.edu

Mariana Silva
University of Illinois,
Urbana-Champaign
Urbana, IL, USA
mfsilva@illinois.edu

Matthew West
University of Illinois,
Urbana-Champaign
Urbana, IL, USA
mwest@illinois.edu

Craig Zilles
University of Illinois,
Urbana-Champaign
Urbana, IL, USA
zilles@illinois.edu

Fuzail Shakir
UC Berkeley
Berkeley, CA, USA
fuzailshakir@berkeley.edu

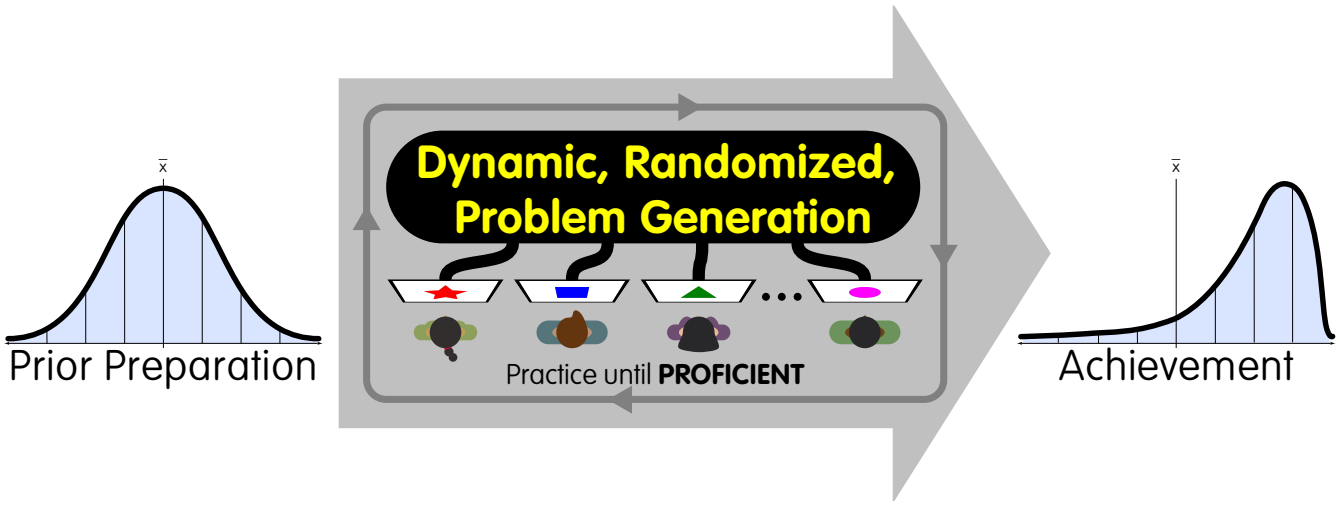

Figure 1: Our proficiency learning model, a key component necessary for "A's for All (as time and interest allow)".

ABSTRACT

"A's for All (as time and interest allow)" is a position that says it is increasingly possible to aim for a world in which students can achieve any grade (level of mastery) that they are willing to work for, even if some students take longer than others or require more practice to get there. Achieving this goal would have profound effects on fairness, equity, and participation in computing, to say nothing of student learning outcomes. We describe what this goal would entail, why it is worth pursuing, what the mechanism and policy requirements are for making progress, and why now is a good time to do it. We give specific and actionable recommendations, many based on our own experience so far, that our colleagues who are excited about the approach can put into immediate practice, and address a number of concerns and objections that our proposal may raise. Importantly, our proposed approach is not all-or-nothing, but all-or-something: there are many things instructors can do within existing policy frameworks and course constraints to move their course experience in this direction.

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada

ACM ISBN 978-1-4503-9431-4/23/03.
https://doi.org/10.1145/3545945.3569847

CCS CONCEPTS

• **Social and professional topics** → **Student assessment**.

KEYWORDS

assessment, pedagogy, communities of practice, scalable

ACM Reference Format:
Dan Garcia, Armando Fox, Solomon Russell, Edwin Ambrosio, Neal Terrell, Mariana Silva, Matthew West, Craig Zilles, and Fuzail Shakir. 2023. A's for All (as Time and Interest Allow). In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569847

1 INTRODUCTION

In early 2022, the community of international computing educators on the `SIGCSE-MEMBERS` mailing list of the ACM Special Interest Group in Computer Science Education [17] was discussing a cluster of connected topics: *growth mindset* [10, 13], *proficiency (aka mastery) learning* [5], *grading for equity* [3, 11], and *specifications grading* [16]. Each of these is an important step toward the ultimate goal of *A's for All [A4A] (as time and interest allow)*: the "A" line doesn't move, but every student gets an opportunity to achieve proficiency and earn whatever grade they are willing to put in the time and effort for [15]. The focus shifts from *"fixed time, variable learning"* to *"fixed learning, variable time"* [22]. Notwithstanding the name, it also supports students who only want a "B"—the idea is that students are given full agency and are allowed the time and support to achieve whatever proficiency (grade) they want out of the class. In other words, rather than "fixed time, variable learning," the student is offered the opportunity to experience "fixed learning over variable time". We believe this has significant potential positive effects on equity as well as on student learning outcomes. The core theoretical ideas motivating "A's for all" are not new [4, 6, 15]. What *is* new is that we now have the technology to implement these ideas in practice. The goal of this position paper is to provide educators and administrators with tools to do just that. We will cover the *what*, *why*, and the *how*, from both a mechanism and policy perspective, and answer some frequently asked *"but how do you..."* and *"but what happens when..."* questions.

As we will see, from a practical standpoint, A4A requires writing autograders for all course projects, adopting technology that affords dynamic, randomized problem generation (for practice and multiple-chance exams), and changing policies and messaging so that students know they will be fully supported, even if it takes them a little more time to learn the material and do the work.

2 BACKGROUND: WHAT ARE GRADES FOR?

Two scholars who vocally advocate for "A's for All" are David MacKay and Alfie Kohn. Kohn counters the preconception that grades should represent a "relative degree of success" to have any "coherent meaning" [14]. The emphasis is ours:

> The goal, in other words, isn't to do well but to defeat other people who are also trying to do well. Grades in this view should be used to announce who's beating whom. And if the students in question have already been sorted by the admissions process, well, they ought to be sorted again. A school's ultimate mission, apparently, is not to help everyone learn but to rig the game so that there will always be losers. *This makes no sense in any context.*

MacKay's persuasive treatment [15] argues that students often fail to achieve an "A" not because they aren't smart enough, but because of differential natural paces of learning, prior preparation, or workload. The more a student is left behind, he argues, the slower they often learn, because we are "piling on new material before they have assimilated the old." He suggests how courses might support these students with "consolidation periods" to allow students to re-sync before beginning a followup unit.

We find these lines of argument highly inspirational, but to act on them, we must first identify how and where our current approaches to grading fall short. Most grading systems attempt to present and assess students in a manner that is appropriate to their level of understanding of course content to date. Concepts introduced later in the course usually dependent on those introduced earlier. As such, the failure of a student to grasp an early concept is often *magnified* in later assessments.

Mastery learning remedies this, as students must first master earlier concepts before moving on to later ones [5]. Mastery learning has had a long history of success [8]; a five-year study with mastery-based learning conducted in the University of Otago showed high levels of engagement, and resulted in the lowest failure rate they had seen [9]. Today, the introduction of computer-based assessment has transformed the process through tracking at the individual student level, both by through bookkeeping and techniques like Bayesian Knowledge Tracing [1]. With individual tracking, each student can move at their own pace: consistent with a growth mindset [10], a student either knows something, or just doesn't know it *yet*.

While the focus on individual students provides a pedagogical advantage, it does require a shift from the traditional grading system and mindset — rather than the entire class focusing on a single concept on a given day, with the new approach, students will be learning different topics during the same time period, perhaps making it slightly more challenging for the instructor to manage [19]. This calls for the application of a different grading mechanism such as *specifications grading* [16], in which assessments and assignments are directly linked to learning outcomes. As we will see, this core idea serves the goals of "A's for All" extremely well.

3 *WHAT* IS "A'S FOR ALL"?

Every student obtaining mastery in a discipline is an aspirational goal that requires a progressive view of learning and the willingness to reexamine entrenched educational policies. For instructors, "A's for All" might be a mindset shift in how to view instruction. We need not teach the same way we were taught. Our classes needn't be like a bus that enforces one speed and one route for all passengers. If students get to our mastery destination, what does it matter if they took the local or the express?

"A's for All" doesn't water down the proficiency needed to achieve an "A"; instead it provides multiple opportunities to show proficiency, so students can revisit, correct, and continue to learn from their submitted work instead of viewing it as a score fixed in time. Students are given the agency to spend as much time and effort as they choose to achieve any desired level of mastery, whether that is an "A" or a lower grade. Doing this requires identifying and lowering institutional barriers that prevent students from using as

much time as needed to gain mastery. We discuss some of these barriers and our approaches to surmounting them in Section 6.

4 *WHY* SHOULD WE ADOPT "A'S FOR ALL"?

The system that most of us operate in now is *Time-Based Learning* (TBL) – "*variable* learning in a fixed time", typically a term [6]. Among other problems, this approach favors students with prior preparation and privilege, raising serious equity concerns [3, 11].

In contrast, many high-stakes professional licenses, such as a commercial airline pilot license, are based on *Proficiency-Based learning* (PBL) — continuous practice until one demonstrates a certain threshold of knowledge and skills before advancing. The credit and recognition achieved are based only on that demonstration of proficiency, not on how long it took to achieve it. Presumably most of us would feel less safe if all we knew about the pilot's training is that they had a certain nominal number of hours of practice and achieved (say) at least a B+ on their assessments.

PBL, or "mastery learning", has roots in the 1950s–1960s when John Carroll stated that "aptitude is the amount of time required by the learner to attain mastery of a learning task" [5]. Time-based teaching systems are also historically imprecise at indicating what skills and knowledge a student knows. If our hypothetical pilot above received a "B" in a course, does that mean they mastered some concepts at an "A" level and others only at a "C" level? Or did they perform at a "B" level consistently across all concepts? To stakeholders such as employers who are trying to value the credential (or passengers on the airplane), the difference matters.

Time-based learning also disproportionately disadvantages students entering computing with less preparation than their peers [3, 11]—the very ones whose success we want to equalize with their peers when we try to broaden participation in CS. An underprepared student receiving a grade other than an A (who could have received an "A" in our proposal) may simply say, "Well, I guess I'm not *meant* to be a CS major." This outcome works against the sense of belonging that is crucial to retention in the computing field for underrepresented students. Any zero-sum grading system that effectively pits students against each other works against the building of a positive classroom climate and student-to-student interaction that builds a supportive academic community [2].

5 *MECHANISM* TO ACHIEVE "A'S FOR ALL"

Implementing PBL with respect to a body of material (e.g. a course) requires a few technical ingredients:

(1) The opportunity for students to get *as much practice as desired, with immediate and detailed feedback,* until they master a skill or concept. We propose to do this with parameterized question generators (PQGs), described below, including autograding.
(2) A way for instructors and students to track what the student has and has not yet mastered in the course material. Our proposed mechanism for doing this is a software-backed *concept map* that graphically represents the knowledge and skills in a course and the dependencies among them.
(3) Business process automation so that instructors can rapidly generate targeted assessments that focus on what a given student doesn't know, and more importantly, help manage the administrative details of resolving Incomplete grades and other processes that may rub up against institutional policies (which we discuss in the next section).

We describe our proposal and efforts so far with respect to each of these items.

5.1 Many Opportunities for Practice

PBL implies that students work *as long as it takes* on different problems until they achieve proficiency on a particular skill or concept. A human instructor creating problem after problem on the fly is not an approach that will scale to large numbers of students [7]. Our proposal is to help automate this by moving away from thinking about *questions* and moving towards thinking about *question generators,* which provide a parameterized "framework" for generating a number of variants of the same underlying question, in particular when some aspects of the question can be randomized. This is shown in Figure 1.

While this idea exists in primitive form in most commodity Learning Management Systems—for example, numerical problems can randomize the parameter values—we go much further by building on PrairieLearn (PL), an open-source platform developed at the University of Illinois at Urbana-Champaign (UIUC) for mastery-based online exercises in STEM courses [20, 21]. PL essentially provides a flexible framework within which arbitrarily rich questions can be created that take advantage of randomization, with arbitrarily sophisticated logic for both question generation and autograding. To give just a few examples, a question about combinational logic can randomize the gate types and input values while still displaying the combinational circuit in a Web browser and allowing the student to interactively enter (for example) intermediate values present at each output. A question about database joins can create multiple example tables populated with fake-but-realistic data, randomize the elements and parameters used in the `SELECT` or `JOIN` clauses, and so on. One can readily imagine similar examples for regular expressions, construction of web URLs, Boolean truth tables, and more. Indeed, we have built all of these and others in support of our current efforts.

PL has a long history at UIUC and has been used to support both formative and summative assessments [23] in over 170 courses across a broad range of STEM disciplines (CS, Engineering, Math, Statistics, Chemistry, Nutrition), and has seen significant recent uptake by other prominent large CS departments.

Creating a new question *type* in PL, with its corresponding autograder, has a nontrivial learning curve, though many graduate and advanced undergraduate students at our institutions have done so. Once a question type and its associated autograder have been created, the learning curve for creating new *question instances* of that type is far lower, and dozens of undergraduates have managed to get started in question authoring with barely a few hours of preparation. Content creation or migration is arguably the largest barrier to course-level adoption of a computer based assessment tool like PrairieLearn, but it scales relatively well with TA support, especially if cross-institutional sharing of content can be coordinated.

In all, a plethora of online assessment tools exists, but we have found PrairieLearn to be the best platform for achieving *"A's for All"*,

given its rich affordances, extensibility, and ability to interoperate with other tools.

5.2 Concept Map

A *Concept Map* graphically represents all the knowledge and skills and conceptual dependencies in a course, and is typically created by the course staff. During the course, it can be shared with students, and even programmatically color-coded to show the student which concepts (nodes in the map) have been mastered. For example, mastered concepts might appear in green, suggestions for what to work on next (e.g. neighboring concepts) in yellow, and concepts whose prerequisites have not yet been met in gray.

5.3 Business Process Automation Support

Instructors need a way to rapidly generate a targeted assessment that lets a student demonstrate proficiency on concepts where they were previously weak. The demonstrated proficiency must then be noted as part of the student's progress in the course (for example, updating a persistent copy of that student's concept map), and potentially affect a student's grade even retroactively, as in the case of "second chance exams" or "clobber policies", in which demonstrating mastery on an assessment item "cancels out" a prior poor grade on a summative assessment [12]. In extreme cases, an instructor may even allow a student to adjust a course-level grade after satisfactorily demonstrating some required level of mastery—perhaps just changing from No Credit to Pass, or perhaps changing from one letter grade to another higher one!

Managing this information and process manually is a major administrative burden for instructors. But if the assessment tools talked to the concept map tools, the gradebook, and potentially even the campus LMS, much of this could be automated. We are working on taking advantage of PrairieLearn's openness to build exactly these integrations.

5.4 Challenges Related to Mechanism

The two main mechanism-related challenges are *onboarding* and *institutional support*. Students in courses must become accustomed to use new PBL tooling, such as PrairieLearn, for taking exams, working on homework, and just plain repeated practice. Instructors and TAs must learn to "think PBL" when creating content—thinking about question "stems" that can be used as the core of a question generator, rather than writing out specific exam or lab questions. TAs and student assistants must learn how to implement those concepts in the assessment tool, in our case PrairieLearn. We are developing robust onboarding materials combined with hands-on training in both theory and practice for PBL.

Institutional support speeds integration of new learning tools, especially those with the broad potential to modernize teaching and learning at scale. Department-or-higher level support will make the difference between a pilot program in a few courses in one department and an *enterprise service* that becomes available with front-line support to all interested students, instructors, and their TAs. Fortunately, choosing open-source tools that are easy to deploy (as we have done) lowers the barrier to entry and adoption and makes it possible to demonstrate positive results at limited scale as a way of achieving higher-level buy-in and argue for the centralization of common resources at scale. For example, UIUC's *Computer-Based Testing Facility (CBTF)* [23], which delivers over 70,000 midterm and final exams for more than 8,000 unique students enrolled in more than 40 classes each semester, employs its own proctors and physical plant security rather than relying on each course's staff for exam integrity.

If the course is the first through the gate to adopt this model, there may be socialization among colleagues that is needed. For example, there may be post-course audits of overall grade distributions to make sure courses aren't suffering from grade inflation. Switching a course to "A's for All" would certainly trigger alarm bells. The burden would be on us, as instructors, to demonstrate that we have not "lowered the bar" for earning a particular grade, but rather provided multiple pathways for students to do so. Indeed, today a panicked Department Chair might react initially by asking, "Why did you give everyone an A?" Our aspirational goal is that over time, a more likely email from the Chair might ask, "Why *didn't* everyone get an A?"

6 *POLICIES* TO ACHIEVE "A'S FOR ALL"

In order for "A's for All" to become a reality, institutional and course policies need to adapt as well. The following sections cover two situations—must all the work for the class be finished *before* the term ends (we call "term constraints"), or are students allowed to continue to work *after* the term is over? In the latter case, there are three different models to allow all students to earn an A (or whatever grade they wish), as time and interest allow. The important idea behind all of them is that our approach is not "all or nothing," but rather "all or something": we believe (and have demonstrated) that important improvements can be achieved within the constraints of existing policy frameworks.

6.1 "A's for Some" with unit synchronization

In this scenario, imagine a four-unit course broken up into four 1-unit pieces. The instructor will allow the students to work at their natural pace *within* a unit, with soft assignment deadlines, second- and third-chance exams, and so on, but everyone must synchronize by the *end* of that unit before the next one begins. Whatever grades are earned at that time become permanent in the gradebook. MacKay calls these "consolidation periods", and it's a first step toward balancing an instructor's synchronization needs with student learning pace preferences.

6.2 "A's for Some" due to term constraints

Here, the instructor has completely bought into the principle of allowing the students the time it take to learn the material at an "A" level, but is thwarted by department, college, or institutional policies that prevent any work completed after the end of the term from being recognized for grading purposes. We call these two policies *"A's for some"*, since not all students might be able to reach the "A" level by the end. That said, there are various things instructors can do to make their classes more equitable. Here are a few:

- Transition from grading on a curve to an absolute scale.
- Provide autograders with no hidden test cases for *all* projects, allowing students to know what it takes to receive full credit, and to choose to persevere until they've earned it.

- Soften assignment deadlines. If students are always working with an autograder, then in theory they could continue to work up until the end of the term on that particular project. On the other hand, telling students on day one that there are *no* deadlines will mean that some students will never start [18], so as with other aspects of our proposals, instructors must seek a balance.
- Reorganize the Concept Map to make the prerequisite chain more shallow. This can allow a student who hasn't achieved proficiency on a particular assignment or topic to stay with the regular pace of the class and not get stalled.
- Make all exams cumulative, that is, every exam covers the totality of what has been covered so far rather than being restricted to a subset of topics; then allow for a higher score for a question on a later exam to *clobber* a lower score for a question on the same topic from an earlier exam.
- Move the syllabus language from fungible *points* to reaching proficiency on *topics* [11].

6.3 "A's for All" via "Incomplete" Grades

In this scenario, we have reached the end of the term, not all students have achieved the level of proficiency they wanted, and they have time *and* interest to continue working and learning. The instructor would enter an **Incomplete (I)** in the grade book. At one of our institutions, this is the mechanism used if a student gets ill or otherwise has to unexpectedly stop attending a course partway through; the student would then continue the work after the term has officially ended, and once they turn in the work, the I is resolved to a letter grade. Depending on the student's timing, this approach has the double advantage that their pace of graduation progress wouldn't stall, and they wouldn't have to worry about resolving their incomplete *on top of* taking a full course load the following term.

If students *don't* finish before the start of the subsequent term, there are trade-offs. If the class in question is a strict prerequisite for a followup class, they will be denied enrollment until they resolve the Incomplete. This may actually be appropriate if the student hasn't demonstrated a proficiency level sufficient to succeed in that next class. The important point is that the decision to take the incomplete is *completely within the purview of the student*: instead of taking an I, they can choose to take the (non-A) grade they had at the end of the term and be done with the class.

In a perfect world, supporting that student's learning as they progressed to proficiency would not be a drain on staff resources. Autograders and automated formative assessment practice activities certainly help, but what if the student just can't get over a conceptual hurdle? One idea is that these students could support *each other*, much as they do in our current class question boards. Student *Academic Interns* could also receive credit to support these students, with appropriate pedagogy training to put them on a "career path" to TA positions.

A suggestion from the UC Berkeley Registrar was that rather than overloading the *Incomplete* designation (technically for handling situations "out of a student's control"), we could introduce a new grading category "Not Yet Proficient (NYP)" or "Not Yet Done (NYD)," depending on whether we mean "Not yet A-level" or "Not yet passing." Finally, we are in the early stages of working with the UC Berkeley Registrar's IT team to create an API that would allow automating the updating of NYD grades when students finish.

6.4 "A's for All" via *Variable Units*

Another idea to support students who don't finish with the grade they wanted by the end of the term is to change the course from fixed to variable credit hours or units. For example, the instructor would clearly designate what each of four units entails, and the student would then move through the course as if they were taking four separate one-unit courses consecutively in one term. A student who did "A" work in the first unit but couldn't get past the first assignment in the next unit by term's end could simply have their transcript reflect an "A" for that first unit. If they wished to return in a subsequent semester to finish more units, they could. Again, this requires some policy flexibility, as a university might not allow certain courses to be repeated for credit.

Assuming the policy issues can be navigated, what are some of the trade-offs of this approach? It fundamentally changes the course from a quiet, synchronous, four-unit course with one recommended "pace", to a chaotic mix of students entering with 0- 1-, 2-, or 3-units already completed, and working at a hundred different paces. This might be possible at a large University where an army of TAs could handle all the different situations, but would certainly be a challenge for a smaller class.

But since each unit has a grade attached, there's more transparency into student performance in the course concepts, at the unit level. A student whose grades in the four units are B, B, B, B would be easy to distinguish from one whose grades are A, A, A, D, even though the overall average (final grade) would be comparable between the two. A student who prefers to hide the "D" benefits from the current one-grade-for-the-whole-course system; an employer who needs to know what specific skills the student mastered would likely prefer the grade-per-unit approach, essentially an incremental step towards competency-based learning.

In this scenario, the institution needs to be very explicit about what the contents of the four units are, for downstream courses and employers who would need to discern exactly what a student knows and can do. Let's say a following course doesn't *need* the student to complete the last unit of the earlier course (which might be, say, an open-ended final project). Then a student who only had the first three units in the books (and hopes to work on the last unit later) wouldn't be prevented from registering for the next course – but would have been in the previous model.

6.5 "A's for All" via *Grade Clobber*

This is most controversial policy and may require approval at levels higher than the instructor or even the Department. Here, at the end of the term, the student is given whatever grade they've earned at the final checkpoint, just like "A's for Some" due to term constraints. However, students are allowed to continue to work on autograded projects and retake randomized summative assessments on their own, or with staff support, and when the automated system determines that a new level of proficiency has been reached, the earlier lower grade is overwritten ("clobbered") by the newer grade. A senior who finally understands recursion thanks to (say)

an internship could have their freshman-year CS1 grade bump up from a "B" to an "A".

Imagine a CS1 class that does not support exam grade clobbering. In that world, a student's class grade reflects what they knew and could do *at all exam checkpoints*. If it did support an exam clobber policy, then a student's class grade reflects what they knew and could do *by the end of class*, which many would argue is all that should really matter. This policy says that a student's class grade reflects what they know and can do *now*.

7 FREQUENTLY ASKED QUESTIONS

Q: If under-resourced students need more time with a course, won't they be overwhelmed to finish that course's Incomplete (for no credit) while taking a full load the next term? A: First, they might be able to finish the Incomplete during the break between terms. Also, institutional support for this initiative is key, with students allowed to take a lower minimum course-load if they're finishing Incompletes.

Q: How to staff the next semester if a ton of students don't finish? Is that fair to the followup instructor? It might even be an unfunded mandate. A: MOOC-level automation can help a lot, along with for-credit Academic Interns who can support them. Many students may only need to finish one or two elements, and can complete them over the break (or during the summer). At the very least, we recommend connecting with downstream instructor(s) to align to the same vision.

Q: Is it really equitable to force under-resourced students to take an extra year to graduate if that's what results from a slower pace through the program? A: We should consider whether a student who graduates in 5 years with all A's has been better served than if they had graduated in 4 years with a mix of B's and C's. Additionally, any interventions already in place at an institution for helping under-resourced students should remain equally effective in an "A's for All" classroom.

Q: How do you incent students to do their work on time when there are no deadlines? A: Our preferred message is "there is a standard-pace deadline, but it's *always* OK if you need more time". One best practice we have found is to require students to fill in a form whenever they need any assignment extension, where they specify how much time they need. Any extension request of fewer than seven days is granted, no questions asked. Any extension request of greater than seven days triggers a 1-on-1 appointment with a TA who checks in with the student, makes sure they have access to physical- or mental-health services, and listens to their situation. At the end of the conversation, the student is assured the extension has been granted.

Q: My exam questions don't easily lend themselves to question generators; they are carefully scripted and authored, without room for meaningful randomization. How can I give students extra chances on exams without them gaming the questions? A: With a large enough question pool, computer-based testing software can build near-infinite variations of an exam. The questions themselves don't need to have many variants if an exam is configured to use question pools. It is a lot of effort, but transferring years' worth of your best questions into testing software will give you the variance you need to allow retries in your class.

Q: Haven't instructors advocated that there should be some penalty for not getting things on the first exam (because otherwise they treat the first exam as a "practice" and never start, thereby "burning" an exam that could have been used?) A: With enough retake exams (with enough randomness) in the back pocket, and infinite retries, that doesn't become an issue.

Q: How does a student learn time-management skills in a class with no deadlines? A: Needing extra time for an assignment does not necessarily result from poor time management skills. We would rather see a student take extra time to complete their work at an "A" level than submit incomplete or rushed "B/C"-level work to meet a deadline.

Q: With the emphasis on tracking proficiency in individual skills and concepts, how do you assess a student's high-level cumulative learning? (There's more to playing basketball than knowing how to pass, dribble, and shoot – you have to put them together in the right way.) A: Like any methodology, PBL shouldn't be used in isolation; group activities, project-based learning, etc. remain an integral part of a well-designed "A's for All" course. Concept maps should include high-level cumulative learning objectives that are assessed with appropriate activities; computer-based testing is not the only source of proficiency evidence.

8 CONCLUSIONS

"A's for All" is about fixed learning in variable time rather than the current practice of variable learning in fixed time, based on long-standing ideas from education research and with the potential to substantially improve fairness and equity in computing education. It is an aspirational vision, but not one that is entirely out of reach. The technical tools to support the *mechanisms* of A's for All are now widely available. The adjustments to policy sometimes lie within the boundaries of what instructors can do now, and sometimes require creative pushing of those boundaries after having socialized and gained support among their colleagues for these ideas. And most importantly, it is not an "all or nothing" vision, but an "all or something" collection of practices that all of us can find places for in our educational efforts starting immediately. We hope others will be as enthusiastic as we are to make progress on this agenda.

9 ACKNOWLEDGEMENTS

This material is based upon work supported by the California Education Learning Lab grant OPR19186. Opinions, findings, and conclusions or recommendations expressed here are those of the author(s) and do not necessarily reflect those of the funders.

REFERENCES

[1] John R. Anderson Albert T. Corbett. 1994. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. (March 1994).

[2] Lecia J Barker, Charlie McDowell, and Kimberly Kalahar. 2009. Exploring factors that influence computer science introductory course students to persist in the major. *ACM Sigcse Bulletin* 41, 1 (2009), 153–157.

[3] Andrew Berns, J. Philip East, and J. Ben Schafer. 2021. *Grading for Equity: A Curriculum Development and Grading Process to Enhance Instruction.* Association for Computing Machinery, New York, NY, USA, 1351. https://doi.org/10.1145/3408877.3432503

[4] Donald L. Bitzer and Peter G. Braunfeld. 1962. Computer Teaching Machine Project: PLATO on ILLIAC. *Computers and Automation* 11, 2 (1962).

[5] Benjamin S. Bloom. 1968. Learning for Mastery. Instruction and Curriculum. Regional Education Laboratory for the Carolina and Virginia, Topical Papers and Reprints, Number 1. *Evaluation comment* 1, 2 (1968), n2.

[6] Benjamin S Bloom. 1974. Time and learning. *American psychologist* 29, 9 (1974), 682.

[7] Benjamin S. Bloom. 1984. The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher* 13, 6 (1984), 4–16. https://doi.org/10.3102/0013189X013006004 arXiv:https://doi.org/10.3102/0013189X013006004

[8] Robert L. Bangert-Drowns Chen-Lin C. Kulik, James A. Kulik. 1990. Effectiveness of Mastery Learning Programs: A Meta-Analysis. (June 1990).

[9] Nick Meek Claudia Ott, Brendan McCane. 2018. Five years of Mastery Learning – What did we learn? (November 2018).

[10] C.S. Dweck. 2006. *Mindset: The New Psychology of Success.* Random House Publishing Group. https://books.google.com/books?id=fdjqz0TPL2wC

[11] J. Feldman. 2018. *Grading for Equity: What It Is, Why It Matters, and How It Can Transform Schools and Classrooms.* SAGE Publications. https://books.google.com/books?id=ORhoDwAAQBAJ

[12] Geoffrey L. Herman, Zhouxiang Cai, Timothy Bretl, Craig Zilles, and Matthew West. 2020. Comparison of Grade Replacement and Weighted Averages for Second-Chance Exams. In *ICER '20: Proceedings of the 2020 ACM Conference on International Computing Education Research.* https://doi.org/10.1145/3372782.3406260

[13] Sarah Hug. 2021. *How Do Faculty Convey Growth Mindset in Computer Science Teaching? A Preliminary Qualitative Study.* Association for Computing Machinery, New York, NY, USA, 1285. https://doi.org/10.1145/3408877.3439663

[14] Alfie Kohn. 2019. Opinion | Why Can't Everyone Get A's? *New York Times* (June 2019).

[15] David MacKay. 2005. *Everyone Should Get an A.* https://www.inference.org.uk/mackay/exams.pdf

[16] L.B. Nilson and C.J. Stanny. 2015. *Specifications Grading: Restoring Rigor, Motivating Students, and Saving Faculty Time.* Stylus Publishing. https://books.google.com/books?id=UrCpCwAAQBAJ

[17] ACM SIGCSE. 2022. *Mailing Lists.* https://sigcse.org/membership/mailing-lists.html

[18] Kristin Stephens-Martinez. 2022. Soft, hard, and late deadlines (or another failed experiment). https://ksm-cs.blogspot.com/2022/02/soft-hard-and-late-deadlines-or-another.html

[19] Stephen Fancsali Susan Berman Steve Ritter, Michael Yudelson. 2016. How Mastery Learning Works. (April 2016).

[20] Matthew West, Geoffrey L. Herman, and Craig Zilles. 2015. PrairieLearn: Mastery-based Online Problem Solving with Adaptive Scoring and Recommendations Driven by Machine Learning. In *2015 ASEE Annual Conference & Exposition.* ASEE Conferences, Seattle, Washington.

[21] Matthew West, Nathan Walters, Mariana Silva, Timothy Bretl, and Craig Zilles. 2021. Integrating Diverse Learning Tools using the PrairieLearn Platform. In *Proceedings of SPLICE 2021 workshop CS Education Infrastructure for All III: From Ideas to Practice.*

[22] Ursula Wolz, Gail Carmichael, Dan Garcia, Bonnie MacKellar, and Nanette Veilleux. 2020. To Grade or Not To Grade. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '20).* Association for Computing Machinery, New York, NY, USA, 479–480. https://doi.org/10.1145/3328778.3366978

[23] Craig Zilles, Matthew West, Geoffrey Herman, and Timothy Bretl. 2019. Every University Should Have a Computer-Based Testing Facility. In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU).*