

ÍNDEX

| | |
|---|-----------|
| 1. DESCRIPCIÓ DEL PROJECTE | 3 |
| 1.1 INTRODUCCIÓ | 3 |
| 1.2 MOTIVACIONS | 7 |
| 1.3 PROPÒSIT DEL PROJECTE | 8 |
| 1.4 OBJECTIUS DEL PROJECTE | 8 |
| 2. ESTUDI DE VIABILITAT | 9 |
| 3. METODOLOGIA | 11 |
| 4. PLANIFICACIÓ | 18 |
| 5. MARC DE TREBALL I CONCEPTES PREVIS | 20 |
| 5.1 FORMALITZACIÓ D'ENUNCIATS LÒGICS | 20 |
| 5.2 CONTRAEXEMPLES PER DEMOSTRACIÓ D'INCORRECTESA | 25 |
| 6. REQUISITS DEL SISTEMA | 27 |
| 6.1 REQUISITS GENERALS DELS EXERCICIS A LA PLATAFORMA ACME | 27 |
| 6.2 REQUISITS DELS EXERCICIS DE FORMALITZACIÓ DE PREDICATS | 28 |
| 6.3 REQUISITS DELS EXERCICIS DE FORMALITZACIÓ DE PROPOSICIONS | 28 |
| 6.4 REQUISITS DELS EXERCICIS DE CONTRAEXEMPLE D'UN RAONAMENT | 28 |
| 7. ESTUDIS I DECISIONS | 29 |
| 7.1 ESTUDI DEL LLENGUATGE DE LES INTERFÍCIES D'USUARI | 29 |
| 7.2 ESTUDI DEL LLENGUATGE DELS NUCLIS CORRECTORS | 31 |
| 7.3 EINES UTILITZADES | 32 |
| 8. ANÀLISI I DISSENY DEL SISTEMA | 33 |
| 8.1 ANÀLISI | 33 |
| 8.2 DISSENY | 38 |
| 9. IMPLEMENTACIÓ I PROVES | 45 |
| 9.1 IMPLEMENTACIÓ DELS FITXERS D'ENUNCIATS | 45 |
| 9.2 IMPLEMENTACIÓ DE LES INTERFÍCIES | 50 |
| 9.3 IMPLEMENTACIÓ DELS NUCLIS CORRECTORS | 52 |
| 9.4 PROVES DE FUNCIONAMENT DE FORMALITZACIÓ DE PREDICATS | 56 |
| 9.5 PROVES DE FUNCIONAMENT DE FORMALITZACIÓ DE PROPOSICIONS | 65 |
| 9.6 PROVES DE FUNCIONAMENT DE CONTRAEXEMPLE DE RAONAMENT | 69 |
| 9.7 PROVES DE FUNCIONAMENT AMB SOLUCIONS BUIDES | 73 |
| 9.8 PROVES DE FUNCIONAMENT COM A PROFESSOR | 74 |
| 10. IMPLANTACIÓ | 76 |
| 11. CONCLUSIONS | 79 |
| 12. TREBALL FUTUR | 81 |
| 13. AGRAÏMENTS | 82 |

| | |
|--|-----------|
| 14. BIBLIOGRAFIA..... | 83 |
| 15. ANNEXOS | 87 |
| 16. MANUALS D'USUARI | 90 |
| 16.1 AJUDA PER AL PROFESSOR DELS EXERCICIS DE FORMALITZACIÓ DE LòGICA DE PREDICATS | 91 |
| 16.2 AJUDA PER A L'ALUMNE DELS EXERCICIS DE FORMALITZACIÓ DE LòGICA DE PREDICATS | 100 |
| 16.3 AJUDA PER AL PROFESSOR DELS EXERCICIS DE FORMALITZACIÓ DE LòGICA DE PROPOSICIONS..... | 101 |
| 16.4 AJUDA PER A L'ALUMNE DELS EXERCICIS DE FORMALITZACIÓ DE LòGICA DE PROPOSICIONS | 110 |
| 16.5 AJUDA PER AL PROFESSOR DELS EXERCICIS DE CONTRAEXEMPLES DE LòGICA DE PREDICATS | 111 |
| 16.6 AJUDA PER A L'ALUMNE DELS EXERCICIS DE CONTRAEXEMPLES DE LòGICA DE PREDICATS..... | 120 |

1. Descripció del projecte

En aquesta secció es desenvoluparà les seccions “Introducció, motivacions, propòsit i objectius del projecte”.

Començarem veient una introducció a la plataforma ACME i les raons d'aquest projecte. A continuació s'exposarà les motivacions que m'han portat a la realització d'aquest projecte en concret, i acabarem veient el propòsit i objectius del projecte.

1.1 Introducció

La Universitat de Girona, i més concretament el Departament d'Informàtica i Matemàtica Aplicada, ja fa uns anys que ha posat en marxa el Projecte Avaluació Continuada i Millora de l'Ensenyament, també anomenat ACME.



Figura 1.1 : Logotip del projecte ACME.

Aquest projecte és una plataforma d'e-learning, és a dir, un sistema d'aprenentatge a través de la xarxa que potencia la col·laboració digital i el bescanvi d'informació entre alumne i professor.

La filosofia pedagògica que hi ha darrera d'aquest tipus de plataformes és la d'afavorir l'aprenentatge des de la pròpia experiència de l'alumne.

D'aquesta manera es complementen les classes presencials on el professor exposa de manera magistral els conceptes i resol alguns exercicis a mode il·lustratiu, amb uns exercicis on l'alumne participa de manera proactiva en el seu aprenentatge.

També és essencial afavorir la planificació correcta, mitjançant l'establiment de terminis per a la realització dels exercicis. D'aquesta manera l'alumne fa l'adquisició de coneixements de les assignatures de manera progressiva i al dia amb el temari impartit a les classes, evitant que l'alumne deixi l'estudi fins a la preparació de l'examen final, amb la dificultat que això li comportaria.

Tenim així un sistema d'avaluació continuada fàcil d'utilitzar tant pels alumnes com pels professors.

L'ACME ha demostrat des dels seus inicis com aquesta manera de treballar ajudava als alumnes a la millor adquisició de coneixements, i la millora dels resultats acadèmics. La clau

del seu èxit rau en l'augment de la motivació de l'alumne en realitzar uns exercicis que abans realitzava de manera opcional i sense seguiment del professor.

En aquesta plataforma, l'alumne sap, i és influenciat, pel fet que el professor estigui fent un seguiment de l'evolució de l'alumne. Però el factor més important és que l'ACME informa a l'alumne si la solució proposada és correcta o incorrecta de manera immediata, i li permet incorporar noves respostes fins a la seva correcta realització.

Un altre dels seus punts forts és la capacitat de crear dossiers personalitzats d'exercicis que s'adaptin a les necessitats de cada alumne de manera molt senzilla per part del professor, permetent adquirir els coneixements previs als estudiants que tenen poca formació bàsica en la temàtica de l'assignatura, i treure el màxim del potencial dels estudiants amb més coneixements i evitant la seva desmotivació amb exercicis adequats al seu nivell.

El professor genera grups d'exercicis de similar temàtica i dificultat que són sortejats entre els alumnes, de manera que es fomenta que cada alumne s'hagi d'enfrontar a la resolució dels exercicis de manera individual i practiqui fins a la seva correcta resolució.

Que la correcció sigui realitzada de manera automàtica pels nuclis correctors de l'ACME deslliura als professors de la correcció d'exercicis, centrant la seva tasca docent en la creació d'exercicis adaptats a les necessitats del seu alumnat i en el seguiment de l'evolució dels alumnes.

Cal esmentar en aquest sentit, que l'ACME disposa de múltiples eines per realitzar el seguiment de la feina realitzada pels alumnes tant de manera individual i com en grup.

| Activitat | N. Exercici | Estat | Result Errors | Syntax Errors | Data limit |
|-----------|-------------|-----------|---------------|---------------|------------|
| 2 | 3 | No Result | 0 | 0 | 22/9/2012 |

Figura 1.2 : Exercici de programació on l'alumne puja un fitxer.

L'ACME permet la autocorrecció de qualsevol problema que tingui un plantejament matemàtic, i ha anat creixent per corregir també programes informàtics en pseudocodi i en la majoria de llenguatges informàtics, grafs d'autòmats, circuits elèctrics, estructura de dades i exercicis de bases de dades que inclouen sentències SQL, diagrames d'Entitat/Relació o esquemes de bases de dades relacionals.

També corregeix exercicis d'assignatures com química inorgànica i orgànica, amb interfície pròpia.

I on té la màxima capacitat d'expansió és amb les eines per corregir problemes estàndard com exercicis de tipus test, omplir els blancs, selecció de la resposta correcta, ordenar frases, relacionar, resposta cert/fals o creació i lliurament de fitxers, eines com portfolis, wikis, glossaris i treball individual i en grup.

D'aquesta manera, l'ACME ha tingut una expansió molt ràpida a dins de la UdG, i actualment és utilitzat per 3.229 alumnes de l'Escola Politècnica Superior, la Facultat de Ciències, Facultat d'Educació, Lletres, la Facultat de Ciències Econòmiques i Empresarials, Facultat de Turisme, Escola Universitària d'Infermeria, en un total de 114 assignatures.

El repositori d'exercicis de l'ACME en aquests moments conté 4.376 exercicis, que gràcies a la seva parametrització poden generar bilions d'enunciats diferents.

Utilització de l'ACME

| | |
|-------------------------|-------|
| Alumnes que l'utilitzen | 3.229 |
| Número d'Assignatures | 114 |
| Número d'exercicis | 4.376 |

Taula 1.1 : Utilització de l'ACME en xifres.

El creixement que ha experimentat la plataforma, ha estat possible gràcies al disseny modular de l'aplicació, on cada nova funcionalitat que estava essent desenvolupada evolucionava de manera independent a la resta de la plataforma i sense comprometre'n el funcionament.

Per afegir un nou mòdul a l'ACME, s'utilitzen els mòduls de base que confereixen l'estructura a la plataforma i les eines per incorporar nous mòduls, amb el que la implantació d'aquests a la plataforma és d'elevada senzillesa.

Actualment, la majoria de les assignatures de primer del Grau d'Informàtica ja disposen d'exercicis adaptats a la plataforma, el que dona una uniformitat en les eines i plataformes que utilitzen els alumnes en la seva formació.

Però no hi són totes les assignatures, falta incloure l'assignatura de Lògica a la plataforma. D'aquesta manera, tant els alumnes com els professors de l'assignatura, podran treure profit dels beneficis esmentats que aporta l'ACME.

Concretament, a l'assignatura de Lògica es disposava d'uns aplicatius a la plataforma Moodle de la Universitat de Girona on es guiava a l'alumne en la resolució dels exercicis. Aquest mètode és molt interessant com a mètode pedagògic per a l'aprenentatge inicial de l'assignatura, però no representa un repte real on els alumnes hagin d'aplicar els seus coneixements per a la seva realització.

Els alumnes també disposaven d'un dossier on se'ls proposava uns exercicis molt complets, i a les pàgines finals d'aquest se'ls donava la solució. Aleshores, de manera voluntària els alumnes realitzaven els exercicis i comprovaven amb la solució si havien après a resoldre'ls.

Per tant, tant els responsables de l'assignatura de Lògica com els responsables de la plataforma ACME van veure que es podrien desenvolupar eines a l'ACME per tal de corregir de forma automàtica els exercicis de l'assignatura. Incorporar l'assignatura de Lògica a l'ACME permet que els alumnes es vegin més involucrats a fer exercicis durant els curs de manera continuada, gràcies als terminis d'entrega, i facilita als professors la seva tasca deslliurant-los de les correccions i oferint-los-hi eines de seguiment.

1.2 Motivacions

Personalment, sempre m'han interessat les maneres i tecnologies que proporcionen diferents metodologies per a l'ensenyament i que cobreixen les diferents necessitats d'aprenentatge en la diversitat de intel·ligències i maneres d'aprendre dels alumnes.

A l'igual que les entitats educatives, valoro la importància de realitzar exercicis pràctics de manera evolutiva enfront de la simple memorització de conceptes nous, i crec fermament que la filosofia que hi ha darrera la plataforma ACME és útil al procés formatiu dels alumnes, com demostren les dades dels seus anys d'existència.

Per tant, he trobat molt atractiva la possibilitat de realitzar un Projecte Final de Carrera que lliga els coneixements de programació assolits al llarg de la carrera amb una aplicació pràctica en el món de l'e-learning.

M'ha agradat el fet que el mòdul a implementar sigui el de Lògica. D'aquesta forma podré contribuir a que els estudiants gaudeixin més aprenent d'una assignatura que aporta uns coneixements bàsics però essencials per a les carreres d'informàtica.

També ha estat valorat el fet de poder treballar en el desenvolupament d'aplicacions web PHP, camp en el que ja tenia una mica d'experiència laboral. Així doncs, podré aprofitar el guiatge d'un tutor que coneix molt aquest medi i continuar recollint experiència en un món que m'interessa com a especialitat professional.

1.3 Propòsit del projecte

Aquest Projecte Final de Carrera pretén incloure un nou mòdul que ha de permetre realitzar els següents tipus d'exercicis de l'assignatura de Lògica dins de la plataforma ACME:

- Formalització d'enunciats de predicats, que poden incloure varies variables i constants.
- Formalització d'enunciats amb proposicions.
- Contraexemple d'un enunciat de predicats.

1.4 Objectius del projecte

Els objectius que s'han de cobrir per la consecució del propòsit del Projecte Final de Carrera són els que aquí s'esmenten:

- Familiaritzar-se amb el funcionament de l'ACME.
- Estudi de les eines i llenguatges més adequats per al desenvolupament del projecte.
- Disseny i especificació del codi en una plantilla d'enunciat per a cadascun dels diferents enunciats, per tal que els professors puguin crear exercicis de manera senzilla.
- Creació de les interfícies que utilitzaran els alumnes per introduir les seves solucions, amb l'objectiu que siguin ben senzilles d'utilitzar. Visualització del resultat de la correcció i del historial de les solucions amb les seves correccions.
- Nucli corrector per a la correcció automàtica dels diferents tipus d'exercici, de manera que l'alumne pugui saber just després d'introduir la seva resposta si la solució proposada és correcta o incorrecta, i que li informi amb un missatge del tipus d'error que ha produït quan s'escaigui.
- Creació d'ajudes per a la utilització del nou mòdul, per a l'alumne i per al professor.
- Integració del nou mòdul a l'ACME.

2. Estudi de viabilitat

Aquest Projecte Final de Carrera ampliarà la varietat d'exercicis d'una plataforma existent, que té implementades una diversitat molt elevada de solucions per resoldre les diferents tipologies d'exercicis. Veurem aquí com podem aprofitar aquesta oferta disponible per realitzar el mòdul de Lògica.

Respecte als llenguatges de programació de les interfícies, l'ACME té exercicis amb JavaScript, PHP, Applets de Java o Flash.

Tenim per tant un ampli ventall a escollir de tecnologies orientades a aplicacions web RIA (Rich Internet Applications) que s'integren a la plataforma sense costos afegits.

Per a la correcció dels exercicis de l'assignatura de Lògica és necessari un llenguatge de programació que permeti la comparació i validació de les expressions de Lògica de proposicions i de predicats.

Amb aquesta finalitat s'ha pensat en el Mathematica, el PHP o el Prolog. Els dos primers ja són utilitzats en altres tipologies d'exercicis a l'ACME, mentre que el Prolog tot i ser compatible amb la plataforma, actualment no es fa servir en cap tipus d'exercici.

D'aquesta manera, independentment de la tria que es faci, la viabilitat del Projecte Final de Carrera queda garantida sense haver d'adquirir noves llicències.

Com que es tenen els coneixements de Lògica necessaris, ja que va ser una assignatura que vaig estudiar el primer curs de carrera, per a la creació del nucli corrector d'aquests exercicis no s'haurà d'invertir temps en conèixer i entendre la temàtica.

La plataforma ACME, a més, disposa d'eines que faciliten la integració de noves tipologies d'exercicis a la plataforma, com diferents plantilles de sintaxis dels fitxers dels exercicis, un programa que valida aquests fitxers o el sorteig d'exercicis entre els alumnes.

Aquestes funcionalitats facilitaran en un elevat grau el desenvolupament i implantació del nou mòdul a l'entorn web existent.

Actualment, l'ACME es troba implantat en un servidor en producció, `acme.udg.edu`, i en un altre servidor de desenvolupament, `acme4.udg.edu`, que durant l'estiu passarà a ser el servidor de producció.

El desenvolupament d'aquest Projecte Final de Carrera es podrà realitzar al servidor de desenvolupament `acme4.udg.edu`, utilitzant un ordinador del departament o un de personal per connectar-nos-hi i per editar els fitxers necessaris.

Necessitarem doncs programes com editors de codi i navegadors web, que són gratuïts, i per desenvolupar el codi del nucli corrector ens connectarem al servidor de desenvolupament.

En temps, el projecte es veia viable en mig any, temps de que dispo, i el fet que sigui un mòdul tancat a implementar, evitarà la dispersió d'objectius i facilitarà la consecució del propòsit del projecte en dit temps.

Els recursos humans necessaris en el projecte són els del tutor del projecte, en Ferran Prados, l'assessorament dels professors de l'assignatura de Lògica, en Josep Humet i en Narcís Coll, i la present estudiant que realitza el Projecte Final de Carrera. No es necessitaran per tant, inversions econòmiques per a la realització del Projecte Final de Carrera i amb la disponibilitat de les persones implicades es va poder començar a treballar de manera immediata.

3. Metodologia

Metodologies àgils

Les metodologies que s'utilitzen en el desenvolupament de projectes informàtics estan en una constant revisió i en el punt de mira dels enginyers de programari.

Hi ha unes metodologies més tradicionals, que es centren en el control del procés, les eines i les notacions, necessaris en projectes on intervenen molts actors amb rols molt diferenciats.

Però moltes empreses i equips de treball, enfront de la quantitat de recursos que consumeix aquesta burocràcia i la poca flexibilitat a que dona marge, deixaven d'aplicar una metodologia, amb el risc per al projecte que això comporta.

Amb la finalitat d'augmentar la productivitat sense perdre en qualitat, els últims anys s'estan promovent unes metodologies més àgils, que es basen en els següents principis:

- Els individus i les interaccions de l'equip són més importants que els processos i les eines
- Donar més importància al programari que funciona que a la documentació exhaustiva
- Col·laborar amb el client enlloc de negociar un contracte
- Ser capaç de donar resposta als canvis enlloc de restringir-se a un pla tancat, gràcies al desenvolupament incremental de programari, amb iteracions curtes

Aquests mètodes àgils s'adapten molt bé a equips de treball petits, com és el cas d'aquest projecte, i quan es disposa de terminis de temps reduïts, requisits canviants o basats en noves tecnologies.

En canvi, no s'adapten bé a grups de treball de més de 20 persones, quan l'entorn no permet la comunicació i col·laboració entre els membres de l'equip, si les tecnologies utilitzades no s'adapten fàcilment als canvis o si el client és reticent a treballar col·laborativament.

Entre les metodologies àgils en destaquen unes quantes amb característiques diferenciades:

- Extreme Programming (XP)
Potencia les relacions interpersonals i promou l'aprenentatge dels desenvolupadors, la simplicitat en les solucions implementades i la valentia per afrontar els canvis.
- SCRUM
Per projectes amb ràpid canvi de requisits, amb iteracions de programari cada 30 dies i reunions diàries de 15 minuts de l'equip de desenvolupament per la coordinació i integració.
- Crystal Methodologies
Centrada en les persones que componen l'equip i la reducció dels blocs produïts
- Dynamic Systems Development Method (DSDM)

Procés iteratiu i incremental on l'equip de desenvolupament i el client treballen junts.

- **Adaptive Software Development (ASD)**
Metodologia iterativa orientada als components de programari i tolerant als canvis on cada iteració es compon de especulació, col·laboració i aprenentatge
- **Feature-Driven-Development (FDD)**
Iteracions curtes, de 2 setmanes com a màxim, centrades en el disseny i implementació del sistema
- **Lean Development (LD)**
Aquesta metodologia veu als canvis com un risc, que analitza amb la finalitat de detectar potencials oportunitats de millora en la productivitat

Entre tota aquesta relació de metodologies àgils s'ha triat la Programació Extrema (XP), que s'adapta perfectament a petits equips de treball, busca la simplicitat en les solucions implementades i la valentia per afrontar els canvis, potencia les relacions interpersonals i promou l'aprenentatge dels desenvolupadors, amb el reconeixement que li donen la gran quantitat d'enginyers de programari que l'utilitzen actualment.

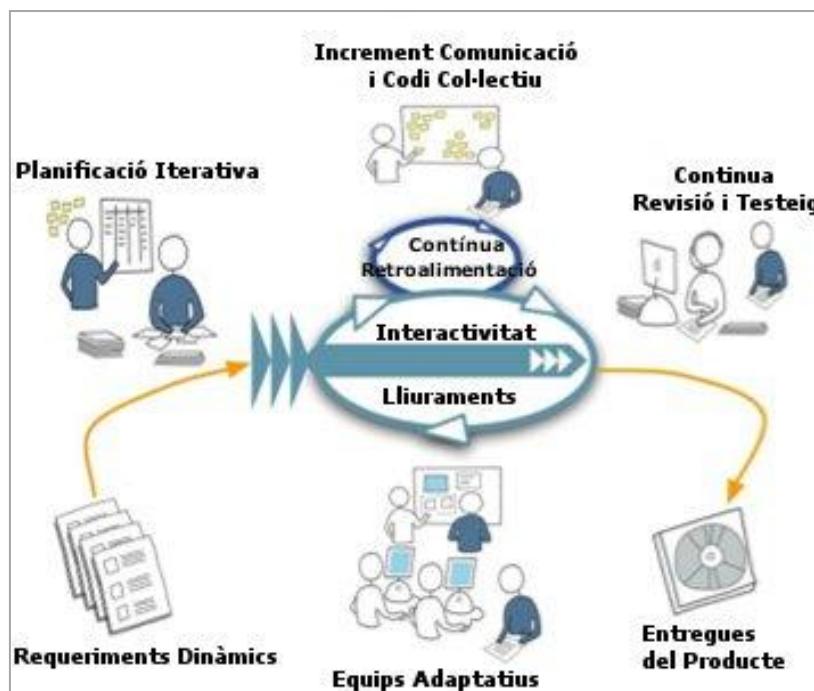


Figura 3.1 : Els pilars de la Programació Extrema.

Si analitzem amb més detall aquesta metodologia, les característiques que l'expliquen són les anomenades "Històries de l'usuari", el seu procés productiu i les pràctiques que segueix.

Les històries de l'usuari són les necessitats que requereix el client, i han de ser prou comprensibles i delimitades per a que necessitin entre una i tres setmanes per programar-se.

El procés productiu es basa en un cicle iteratiu que va aportant noves funcionalitats al projecte. Així podem definir les següents fases:

1. Exploració

El client defineix les històries d'usuari més útils. L'equip de desenvolupament es familiaritza amb les tecnologies a utilitzar.

2. Planificació de la entrega

Es fa un acord entre client i equip de desenvolupament de les històries que s'inclouran en la primera entrega, que no haurà de ser més enllà de tres mesos.

3. Iteracions

S'organitzen les històries en grups, que correspondran a iteracions, que puguin ser realitzats en tres setmanes.

4. Producció

Es realitzen proves i es porta a producció, a l'entorn del client.

5. Manteniment

Es fan les iteracions necessàries per a la segona entrega, tenint en compte que s'ha de mantenir el sistema de producció funcionant.

6. Mort del projecte

Es revisa el rendiment i la robustesa del sistema i es genera la documentació.

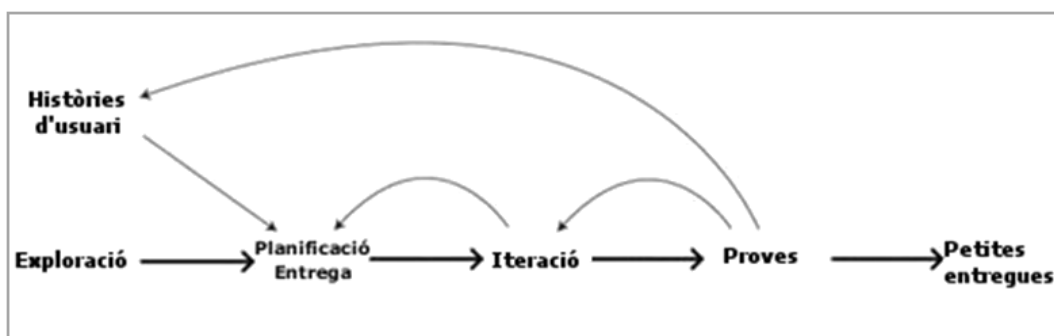


Figura 3.2 : Processos en la Programació Extrema.

Pràctiques de la Programació Extrema

La Programació Extrema no és només un cicle de producte determinat, sinó que es complementa amb un conjunt de bones pràctiques que són la clau de l'èxit com a metodologia àgil.

- ❖ **Procés de planificació**
L'equip de desenvolupament acordarà amb el client les històries d'usuari que han de ser incloses en la primera entrega, que no tindrà un termini superior als tres mesos.
- ❖ **Client in-situ**
El client, o un representant seu, treballarà assessorant l'equip durant la realització de tot el projecte.
- ❖ **Metàfora**
Les metàfores són petites històries de com hauria de funcionar el sistema que creen un vocabulari propi sobre el projecte i ajuden en la nomenclatura de classes i mètodes del sistema.
- ❖ **Disseny simple**
S'ha de dissenyar i codificar la solució més simple que compleixi els requeriments.
- ❖ **Entregues petites**
Cada tres mesos com a molt es fa una entrega operativa.
- ❖ **40 hores per setmana**
Programadors frescs generen codi de major qualitat, pel que es limitarà la jornada laboral a 40 hores setmanals i en cas d'haver de treballar hores extra, no es farà durant un període de temps superior a les dues setmanes seguides.
- ❖ **Estàndards de codificació**
Permeten que el codi sigui més llegible i en facilita el manteniment.
- ❖ **Propietat col·lectiva del codi**
Tots els programadors poden reescriure codi del projecte en qualsevol moment. D'aquesta manera, el codi de propietat compartida és revisat per més programadors.
- ❖ **Integració contínua**
Cada tros de codi s'integra al sistema un cop és enllestit i s'executen les proves.
- ❖ **El principi de proves**
S'han d'establir les proves d'acceptació del programa com si d'una caixa negra es tractés, amb les entrades al sistema i les sortides esperades.
- ❖ **Programació en parelles**
El codi s'escriu entre dos programadors que seuen al mateix ordinador, de manera que el codi es revisa a mida que s'escriu i els problemes es solucionen molt més de pressa.

❖ Refactorització

El codi es revisa i recodifica constantment per tal de millorar el disseny del sistema, flexibilitzar-lo i minimitzar el codi duplicat o ineficient.

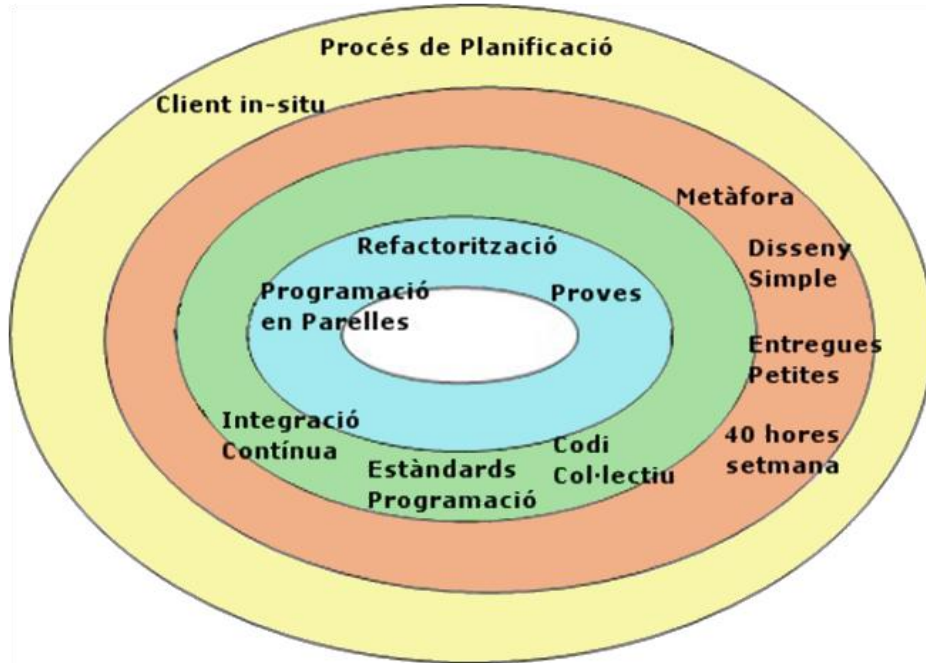


Figura 3.3 : Pràctiques de la Programació Extrema.

Aplicació de la Programació Extrema com a metodologia del projecte

En aquest Projecte Final de Carrera s'ha aplicat una versió una mica reduïda d'aquesta tècnica, ja que no és un projecte amb un únic programador.

Anem a veure punt per punt com s'han aplicat les pràctiques de la Programació Extrema:

❖ Procés de planificació

Els professors de l'assignatura de lògica han definit els tipus de exercicis que tenien necessitat de resoldre i quins eren els més prioritaris.

Els exercicis que han considerat més prioritats han estat els de formalització d'enunciats de lògica proposicional i de predicats, ja que ja disposen a la plataforma Moodle d'uns exercicis guiats per a la resolució de raonaments.

Un cop implementats aquests tipus d'exercicis, van considerar interessant afegir la correcció d'un contraexemple proposat per l'alumne a un raonament de lògica de predicats.

❖ Client in-situ

S'han tingut reunions amb els professors per avaluar la idoneïtat del mètode corrector de cada tipus d'exercici que es volia implementar.

❖ Metàfora

Els professors de l'assignatura de lògica han transmès els requisits del sistema de manera oral.

❖ Disseny simple

S'ha mantingut la simplicitat del codi, trencant el codi en funcionalitats que poguessin ser reutilitzables per altres parts del mòdul de lògica.

❖ Entregues petites

Cada tipus d'exercici s'ha implementat en una entrega

❖ 40 hores per setmana

Els últims mesos la realització del Projecte Final de Carrera s'ha compaginat amb una feina a mitja jornada, pel que la disponibilitat d'hores per a realitzar el projecte ha disminuït.

La suma de les hores dedicades a la feina i al projecte ha arribat en algun cas a les 50 hores setmanals, i per mantenir la motivació i evitar el desgast s'ha realitzat el projecte a la Universitat en un horari establert.

❖ Estàndards de codificació

S'ha mantingut l'estàndard dels mòduls ja implementats a la plataforma ACME.

❖ Propietat col·lectiva del codi

El projecte s'ha realitzat al servidor de desenvolupament que serà el de producció, pel que el codi quedi a disposició de tothom amb permisos per editar els fitxers.

❖ Integració contínua

Les noves funcionalitats s'han anat desenvolupant directament al servidor de proves, que a partir de setembre serà el servidor en producció.

❖ El principi de proves

Totes les noves funcionalitats s'han provat al sistema final de manera manual sobre els enunciats del dossier d'exercicis de l'assignatura proporcionat pels professors.

❖ Programació en parelles

Al només haver-hi una programadora, no ha estat possible provar aquesta pràctica.

❖ Refactorització

A l'anar afegint funcionalitats s'ha anat refactoritzant el codi per tal de flexibilitzar-lo i minimitzar-lo, augmentant-ne la llegibilitat.

Podem dir que s'han seguit les pràctiques de la metodologia de la Programació Extrema de manera bastant exhaustiva, no tant en les pràctiques de treball en equip ja que només hi ha una programadora, però sí en les de planificació de manera col·laborativa amb el client, les contínues revisions i proves i les entregues petites, tot això de manera iterativa per cobrir els diferents tipus d'exercicis a implementar.

4. Planificació

Des del dia de la primera reunió amb el tutor fins a la entrega del Projecte Final de Carrera documentat es disposava de 5 mesos.

En una primera reunió amb els professors de l'assignatura es va acordar que les funcionalitats més interessants eren els exercicis on es corregien formalitzacions d'enunciats de proposicions i predicats.

Per a l'adquisició dels coneixements necessaris per a la creació d'aquests tipus d'exercicis i per a la codificació de les interfícies i nuclis correctors es va planificar una duració de tres mesos, que havien de concloure en una primera entrega de producte testejada i acabada.

Un cop feta la primera entrega del producte, el quart mes s'utilitzaria en la creació d'un nou tipus d'exercici, que es preveia que es podria realitzar en menys temps gràcies als coneixements adquirits en la realització dels dos primers tipus d'exercicis.

En una reunió amb els professors de l'assignatura, aquests van prioritzar la correcció d'exercicis de contraexemple, on l'alumne ha d'introduir un contraexemple que demostrï la invalidesa d'un raonament de lògica de predicats. Aquesta seria la segona entrega de producte al client.

El cinquè més s'invertiria en realitzar la posada en marxa al servidor de producció i a realitzar la documentació del Projecte Final de Carrera, que s'alimentarà d'un quadern de bitàcola on guardarem al llarg de tot el projecte les anotacions que considerem més importants.

Respecte al procés a seguir per a l'obtenció de cadascun dels tres tipus d'exercicis, tots tenien en comú les fases següents:

- Estudi previ
- Creació de la plantilla de fitxer d'enunciat amb solució que haurà d'omplir el professor per crear exercicis amb aquests nuclis correctors
- Creació de la interfície
- Creació del nucli corrector
- Proves

A la pràctica, aquesta planificació inicial es va transformar en el següent calendari:

| | | Novembre 2011 | Desembre 2011 | Gener 2012 | Febrer 2012 | Març 2012 | Abril 2012 |
|-----------------------------------|-----------------|---------------|---------------|-------------|-------------|-------------|-------------|
| Estudi previ de viabilitat | | ■ | | | | | |
| Formació llenguatges i llibreries | | | ■ ■ ■ | ■ ■ | | ■ | |
| Formal. Pred. | Estudi previ | | ■ ■ ■ | | | | |
| | Fitxer tex | | ■ | | | | |
| | Interfície | | | ■ | | | |
| | Nucli corrector | | | ■ ■ ■ ■ ■ ■ | ■ ■ ■ ■ ■ ■ | | |
| | Proves | | | ■ ■ ■ ■ ■ ■ | ■ ■ ■ ■ ■ ■ | | ■ ■ ■ ■ ■ ■ |
| Formal. Prop. | Estudi previ | | | | | | |
| | Fitxer tex | | | | | | |
| | Interfície | | | | | | |
| | Nucli corrector | | | | | | |
| | Proves | | | | | ■ ■ ■ ■ ■ ■ | ■ ■ ■ ■ ■ ■ |
| Contraex. | Estudi previ | | | | | | |
| | Fitxer tex | | | | | | |
| | Interfície | | | | | | |
| | Nucli corrector | | | | | | |
| | Proves | | | | | ■ ■ ■ ■ ■ ■ | ■ ■ ■ ■ ■ ■ |
| Documentació del projecte | | | | | | ■ ■ ■ ■ ■ ■ | ■ ■ ■ ■ ■ ■ |

Taula 4.1 : Planificació del Projecte Final de Carrera.

5. Marc de treball i conceptes previs

El projecte ACME és una plataforma d'e-learning que proporciona una arquitectura on es poden incorporar noves funcionalitats fàcilment.

En aquest Projecte Final de Carrera es vol ampliar les funcionalitats d'aquesta estructura per a que pugui corregir els tipus d'exercicis claus per a l'assignatura de Lògica de primer de les carreres d'informàtica.

Els tipus d'exercicis que s'implementaran, tal com s'ha indicat prèviament són:

- Formalització d'enunciats amb proposicions i amb predicats
- Contraexemple d'un enunciat de predicats

Haurem de disposar d'uns coneixements d'aquesta temàtica per poder implementar correctament aquests exercicis i poder realitzar les proves i tests necessàries. Al ser una assignatura de primer de carrera, ja es disposa dels coneixements necessaris, que s'han hagut de refrescar.

S'ha rebut la col·laboració dels professors de l'assignatura de Lògica Josep Humet i Narcís Coll per a prioritzar els exercicis a implementar i resoldre els dubtes de concepte sobre l'assignatura que han aparegut en la realització del projecte.

El tutor del Projecte Final de Carrera, en Ferran Prados, ha estat vital per la ràpida implantació d'aquests exercicis a la plataforma ACME, ja que té un coneixement profund i exhaustiu d'aquest entorn d'e-learning.

Anem a veure tot seguit els principals conceptes que s'han aplicat.

5.1 Formalització d'enunciats lògics

La formalització d'enunciats lògics consisteix en traduir mitjançant simbologia pròpia de la lògica i les lletres de l'alfabet per representar àtoms, variables i funcions un enunciat en llenguatge natural.

Els connectors lògics i quantificadors que s'utilitzaran són:

- Condicional:
Es tradueix "si ... llavors ...". Per exemple, " $A \rightarrow B$ " es tradueix "si A llavors B"
- ¬ Negació:
Es tradueix per "no", per exemple, " $\neg P$ " es tradueix "no P"

- \wedge Conjunció:
Es tradueix per “i”, per exemple “ $A \wedge B$ ” es tradueix “A i B”
- \vee Disjunció:
Es tradueix per “o”, per exemple “ $A \vee B$ ” es tradueix “A o B”
- \exists Quantificador existencial:
S’acostuma a traduir per “hi ha” o “existeix” i va sempre acompanyat d’una variable.
“ $\exists xP$ ” es llegeix “hi ha x tal que P”
- \forall Quantificador universal:
S’acostuma a traduir “per tot” i va sempre acompanyat d’una variable.
“ $\forall xP$ ” es llegeix “per tot x es compleix P”

La lògica de predicats és la que inclou quantificacions, mentre que la lògica proposicional es construeix únicament amb símbols de proposició i connectors lògics.

Les fórmules lògiques utilitzen els parèntesis com a signes de puntuació, per a resoldre possibles ambigüitats en les expressions, i es poden afegir tants parèntesis addicionals com es vulgui.

Les normes de prioritats són les que indiquen en quin ordre cal restaurar els parèntesis obviats: primer es restauren els parèntesis corresponents als quantificadors i a les negacions, en segon lloc es restauren els de les conjuncions i disjuncions, i en tercer lloc els dels condicionals i bicondicionals.

Per exemple, “ $A \wedge \neg B \wedge C \rightarrow D \vee E$ ” és equivalent a “ $((A \wedge (\neg B) \wedge C) \rightarrow (D \vee E))$ ” i “ $\forall x P(x) \rightarrow Q(x)$ ” és equivalent a “ $(\forall x P(x)) \rightarrow Q(x)$ ”.

Haurem de tenir en compte que no hi ha una única manera d’escriure una fórmula lògica, i les regles de resolució marcaran les relacions d’equivalència.

Les regles són les que aquí s’exposen:

| | |
|-------------------|--|
| \rightarrow | Exemple: $A \rightarrow B \equiv \neg A \vee B$ Regla del condicional. |
| \leftrightarrow | Exemple: $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$ Regla del bicondicional. |
| $\neg\neg$ | Exemple: $\neg\neg A \equiv A$ Simplificació de la doble negació. |
| $\neg\wedge$ | Exemple: $\neg(A \wedge B) \equiv \neg A \vee \neg B$ Negació d’una conjunció. |
| $\neg\vee$ | Exemple: $\neg(A \vee B) \equiv \neg A \wedge \neg B$ Negació d’una disjunció. |

| | |
|---|---|
| Aplana \wedge | Exemple: $(A \wedge B) \wedge (C \wedge D \wedge (E \wedge F)) \Leftrightarrow A \wedge B \wedge C \wedge D \wedge E \wedge F$ Regla d'aplanament de la conjunció. En una conjunció de conjuncions els parèntesis es poden simplificar. |
| Aplana \vee | Exemple: $(A \vee B) \vee (C \vee D \vee (E \vee F)) \Leftrightarrow A \vee B \vee C \vee D \vee E \vee F$ Regla d'aplanament de la disjunció. En una disjunció de disjuncions els parèntesis es poden simplificar. |
| Idempotència \wedge | Exemple: $A \wedge B \wedge C \wedge B \Leftrightarrow A \wedge B \wedge C$ Idempotència de la conjunció. En una conjunció els conjuntants repetits es poden eliminar. |
| Idempotència \vee | Exemple: $A \vee B \vee C \vee B \Leftrightarrow A \vee B \vee C$ Idempotència de la disjunció. En una disjunció els disjuntants repetits es poden eliminar. |
| Distributiva $\vee \wedge$ | Exemple: $(A1 \wedge A2) \vee B \vee (C1 \wedge C2) \Leftrightarrow (A1 \vee B \vee C1) \wedge (A1 \vee B \vee C2) \wedge (A2 \vee B \vee C1) \wedge (A2 \vee B \vee C2)$ Distributiva de la disjunció respecte la conjunció. Permet passar d'una forma disjuntiva a una forma conjuntiva. |
| Distributiva $\wedge \vee$ | Exemple: $(A1 \vee A2) \wedge B \wedge (C1 \vee C2) \Leftrightarrow (A1 \wedge B \wedge C1) \vee (A1 \wedge B \wedge C2) \vee (A2 \wedge B \wedge C1) \vee (A2 \wedge B \wedge C2)$ Distributiva de la conjunció respecte la disjunció. Permet passar d'una forma conjuntiva a una forma disjuntiva. En el mètode de resolució no té cap utilitat. |
| $(A \vee P) \wedge (\neg A \vee Q)$ Resolució PVQ | Regla de resolució. Actua sobre dues clàusules amb literals resolubles: un literal positiu en una clàusula i la corresponent negació en l'altra. La resolvent (conclusió) que s'obté és la clàusula formada per la disjunció de tots els literals restants d'ambdues clàusules. |
| $\neg \forall$ | Exemple: $\neg \forall x P(x) \Leftrightarrow \exists x \neg P(x)$ Negació d'una quantificació universal. |
| $\neg \exists$ | Exemple: $\neg \exists x P(x) \Leftrightarrow \forall x \neg P(x)$ Negació d'una quantificació existencial. |
| \forall finit | Exemple: $\forall x P(x) \Leftrightarrow P(u1) \wedge \dots \wedge P(un)$ Quantificació universal finita. Quan l'univers és $\{u1, \dots, un\}$ (finit) la quantificació universal equival a una conjunció aplicada a tots els elements de l'univers. |
| \exists finit | Exemple: $\exists x P(x) \Leftrightarrow P(u1) \vee \dots \vee P(un)$ Quantificació existencial finita. Quan l'univers és $\{u1, \dots, un\}$ (finit) la quantificació existencial equival a una disjunció aplicada a tots els elements de l'univers. |

Simplifica = Exemple: $a = b \vee P \Leftrightarrow P$
Simplificació de la igualtat de constants (diferents) de l'univers en una disjunció.

Simplifica \neq Exemple: $x \neq x \vee P \Leftrightarrow P$
Simplificació de la no igualtat en una disjunció.

$\forall x P(x)$ Sublimació $P(x)$
Regla de sublimació. Prescindeix de l'escriptura dels quantificadors universals deixant les variables universals (sublimades) lliures per a ser substituïdes per qualsevol terme.

Exemple 1

Anem a veure com a exemple un enunciat de lògica de proposicions:

“Quan menjo arengades tinc set i tinc fred”

A: menjo Arengades

S: tinc Set

F: tinc Fred

formalitzat es transforma a:

$$A \rightarrow S \wedge F$$

o bé, tenint en compte gràcies a la prioritat dels connectors anteriorment esmentats:

$$A \rightarrow (S \wedge F)$$

Exemple 2

I un exemple d'enunciat de lògica de predicats:

“Hi ha aliments que només es poden menjar si són cuits”

A(x) : x és un Aliment

M(x): x es pot Menjar

C(x): x és Cuit

que formalitzat es transforma de la següent manera:

$$\exists x (A(x) \wedge (M(x) \rightarrow C(x)))$$

o bé, aplicant les regles anteriorment vistes, també podríem haver escrit com:

$$\exists x (A(x) \wedge (\neg M(x) \vee C(x)))$$

Com acabem de veure, hi ha varies formalitzacions equivalents per a un mateix enunciat.

Per tant, el corrector haurà de comparar la solució proporcionada pel professor amb la de l'alumne per veure si són equivalents.

5.2 Contraexemples per demostració d'incorrectesa

El tercer tipus d'exercici que resoldrem serà el de cerca d'un contraexemple per a la demostració de la incorrectesa d'un raonament lògic.

Aquesta demostració es fonamenta en que si troben una interpretació que compleixi que les premisses són vertaderes i la conclusió és falsa, el raonament és incorrecte.

Per a aquest tipus d'exercicis, haurem de conèixer les igualtats bàsiques dels connectors lògics:

$$\neg \text{Cert} = \text{Fals}$$

$$\neg \text{Fals} = \text{Cert}$$

$$\text{Cert} \wedge \text{Cert} \wedge \text{Cert} = \text{Cert} \quad (\text{conjunció de Certs és Cert})$$

$$\text{Fals} \wedge \text{Cert} \wedge \text{Cert} = \text{Fals} \quad (\text{conjunció amb algun Fals és Fals})$$

$$\text{Fals} \vee \text{Fals} \vee \text{Fals} = \text{Fals} \quad (\text{disjunció de Fals és Fals})$$

$$\text{Fals} \vee \text{Fals} \vee \text{Cert} = \text{Cert} \quad (\text{disjunció amb algun Fals és Cert})$$

$$\text{Fals} \rightarrow \text{Fals} = \text{Cert}$$

$$\text{Fals} \rightarrow \text{Cert} = \text{Cert}$$

$$\text{Cert} \rightarrow \text{Fals} = \text{Fals}$$

$$\text{Cert} \rightarrow \text{Cert} = \text{Cert}$$

I també la regla de substitució dels quantificadors en un domini d'interpretació $\{c_1, \dots, c_n\}$

$$\forall x F[x] = F[c_1] \wedge \dots \wedge F[c_n]$$

$$\exists x F[x] = F[c_1] \vee \dots \vee F[c_n]$$

Exemple 3

Demostrar que el següent raonament és incorrecte trobant un contraexemple:

$$\forall x (A(x) \rightarrow B(x)) \quad (\text{premissa 1: tots els anarquistes porten barba})$$

$$\forall x (C(x) \rightarrow B(x)) \quad (\text{premissa 2: tots els comunistes porten barba})$$

$$\Rightarrow \forall x (A(x) \rightarrow C(x)) \quad (\text{conclusió: tots els anarquistes son comunistes})$$

per a un domini de dos referencians, $\{a, b\}$, la interpretació

$$A(a) = \text{Cert} \quad A(b) = \text{Fals}$$

$$B(a) = \text{Cert} \quad B(b) = \text{Cert}$$

$$C(a) = \text{Fals} \quad C(b) = \text{Cert}$$

fa que

$$\begin{aligned}\forall x (A(x) \rightarrow B(x)) &= (A(a) \rightarrow B(a)) \wedge (A(b) \rightarrow B(b)) \\ &= (\text{Cert} \rightarrow \text{Cert}) \wedge (\text{Fals} \rightarrow \text{Cert}) \\ &= \text{Cert} \wedge \text{Cert} \\ &= \text{Cert} \qquad \text{La premissa 1 és certa}\end{aligned}$$

$$\begin{aligned}\forall x (C(x) \rightarrow B(x)) &= (C(a) \rightarrow B(a)) \wedge (C(b) \rightarrow B(b)) \\ &= (\text{Fals} \rightarrow \text{Cert}) \wedge (\text{Cert} \rightarrow \text{Cert}) \\ &= \text{Cert} \wedge \text{Cert} \\ &= \text{Cert} \qquad \text{La premissa 2 és certa}\end{aligned}$$

$$\begin{aligned}\forall x (A(x) \rightarrow C(x)) &= (A(a) \rightarrow C(a)) \wedge (A(b) \rightarrow C(b)) \\ &= (\text{Cert} \rightarrow \text{Fals}) \wedge (\text{Fals} \rightarrow \text{Cert}) \\ &= \text{Fals} \wedge \text{Cert} \\ &= \text{Fals} \qquad \text{La conclusió és falsa}\end{aligned}$$

Per tant s'ha trobat un contraexemple, una interpretació que fa que les premisses siguin certes però la conclusió falsa, pel que el raonament proposat queda demostrat que és incorrecte.

Cal esmentar, que pot haver diverses interpretacions que siguin contraexemple d'un raonament, i que el professor no les coneix totes, pel que el corrector automàtic haurà d'avaluar si la interpretació proposada per l'alumne és la correcta.

6. Requisits del sistema

La definició dels requisits del sistema permet concretar d'una manera clara que cal fer perquè el projecte compleixi els seus objectius, tant des de el punt de vista de programari com de maquinari.

Amb aquesta especificació estarem en disposició d'establir un disseny que s'ajusti a aquests requeriments.

Esmentar que en el cas d'aquest projecte, els requeriments són exclusivament de programari, ja que el maquinari ja ens el proporciona la plataforma ACME, amb el servidor acme4.udg.edu, que en aquets moments és un servidor de desenvolupament i que al setembre serà transformat en el servidor de producció pels seus administradors.

Anem a veure doncs quins són els requisits que s'han tingut en compte per tal de portar a terme el Projecte Final de Carrera.

6.1 Requisits generals dels exercicis a la plataforma ACME

- Interfície adaptada als dos tipus d'usuari: alumne i professor.
- Interfície usable, amigable i de fàcil aprenentatge.
- Interfície compatible amb els diferents navegadors web, i a ser possible, sense requeriments de Flash o applets de Java que puguin dificultar la visualització a l'usuari.
- Interfície d'estil coherent amb l'utilitzat dins la plataforma ACME.
- Interfície lleugera i de ràpida descàrrega per a no sobrecarregar el tràfic cap el servidor.
- No s'enviarà a corregir una solució fins que s'hagin omplert tots els camps requerits.
- Robustesa en front entrades de dades invàlides.
- Nucli corrector.
- Consulta de l'historial de solucions enviades per l'alumne amb la seva avaluació corresponent.
- Temps de correcció el més breu possible.
- Crear una ajuda destinada a guiar al professor a la creació de cadascun d'aquests tipus d'exercicis per la plataforma ACME.
- Crear una ajuda per l'alumne per guiar-lo en la utilització de la interfície dels exercicis.

6.2 Requisits dels exercicis de formalització de predicats

- Aquests exercicis constaran d'una única frase de lògica de predicats a formalitzar.
- Poden incloure vàries variables i constants.
- El nucli corrector ha de retornar a la plataforma ACME el resultat de la correcció, podent ser "Correcte", "Incorrecte" o "Error Sintàctic" per a les solucions no acceptades.
- Temps de correcció el més breu possible, especialment amb la dificultat computacional que aporten els quantificadors de lògica de predicats.
- Considerar error sintàctic, enlloc d'error de resultat, les errades amb parèntesis o la utilització de caràcters no associats a l'exercici.

6.3 Requisits dels exercicis de formalització de proposicions

- Aquests exercicis constaran d'un número variable de frases de lògica de proposicions a formalitzar.
- El nucli corrector ha de retornar a la plataforma ACME el resultat de la correcció, podent ser "Correcte", "Incorrecte" o "Error Sintàctic" per a les solucions no acceptades.
- Considerar error sintàctic, enlloc d'error de resultat, les errades amb parèntesis o la utilització de caràcters no associats a l'exercici.

6.4 Requisits dels exercicis de contraexemple d'un raonament

- Comprovar la correctesa d'un contraexemple a un raonament de lògica de predicats.
- El nucli corrector ha de retornar a la plataforma ACME el resultat de la correcció, podent ser "Correcte", "Incorrecte".

7. Estudis i decisions

El mòdul dels exercicis de Lògica que es crearà en aquest Projecte Final de Carrera estarà integrat a la plataforma ACME.

Aquesta plataforma està constituïda per un entorn web en producció que utilitza HTML i CSS per a la presentació de les pantalles, JavaScript per al dinamisme al cantó del client i PHP per la gestió de l'aplicatiu al cantó del servidor, amb una base de dades PostgreSQL.

Els diferents tipus d'exercicis ja existents han estat implementats en diferents llenguatges en funció dels seus requisits específics, tant per a les interfícies com per als nuclis correctors.

Anem a veure doncs quines són els llenguatges més interessants per a la realització del mòdul de Lògica.

7.1 Estudi del llenguatge de les interfícies d'usuari

L'objectiu principal de les interfícies que faran servir els alumnes en la realització dels exercicis és que aquestes siguin molt intuïtives i fàcils d'utilitzar.

Aquest requeriment està guanyant molt força els últims anys, i ha assentat les bases de les "Rich Internet Applications" o RIA, en les que tal com indica el seu nom, s'espera una experiència enriquidora per a l'usuari, amb

- usabilitat similar a la manera natural de les persones de fer les accions
- visualment agradable
- que no necessiti instal·lació ni actualitzacions, ja que la part de client és un navegador web
- disponibles a qualsevol ordinador amb qualsevol sistema operatiu amb Internet
- amb un elevat grau d'interactivitat amb el servidor web
- entorn segur i lliure de virus
- possibilitat de monitoritzar totes les accions

La plataforma ACME és un programari RIA que els alumnes utilitzen a través del seu navegador web, amb una interfície clara, agradable i amb una interactivitat immediata en la seva comunicació amb el servidor web.

L'objectiu d'aquest estudi és seleccionar quina de les tecnologies RIA actual és la més adequada per la implementació de les interfícies dels exercicis del mòdul de lògica.

Anem a veure les principals avantatges i inconvenients d'aquests marcs de treball:

❖ *Llibreries JavaScript*

Col·lecció de funcions JavaScript compatibles amb tots els navegadors amb components RIA com taules, gràfics, utilitats Ajax i interaccions per esdeveniments al

navegador. Entorn de treball gratuït i amb funcionalitats que donaran un acabat professional. Alguns exemples són jQuery, MooTools, YUI o ExtJS.

❖ *Adobe Flash*

El punt fort de Flash, amb la seva extensió Flex per fer components web RIA, és que és compatible amb moltes plataformes i navegadors, visualitzant-se l'aplicatiu de la mateixa manera en tots els sistemes operatius i navegadors. Útil si l'aplicació té gràfics complexes, animacions, té una arquitectura basada en esdeveniments o si els programadors són desenvolupadors Java, ja que el Flash treballa molt bé amb Java com a back-end al cantó de servidor. Com a inconvenient, no tots els usuaris tenen el Adobe Flash Player instal·lat i actualitzat als seus navegadors, tot i que és gratuït i en els dispositius iPad no funciona.

❖ *Silverlight*

La versió de Microsoft de Flash és útil en aplicacions amb gràfics complexes i per proporcionar vídeo streaming, amb arquitectura conduïda per esdeveniments i si l'equip de programadors està format per desenvolupadors .NET. No és compatible amb tots els sistemes operatius i navegadors.

❖ *HTML5*

És el resultat d'unir el millor de HTML4, JavaScript, CSS, llibreries JavaScript i Flash. Però el seu gran inconvenient és que molts navegadors encara no han tret una versió compatible amb aquesta tecnologia.

❖ *Applet de Java*

Útil per crear interaccions visuals, presentar vídeo, objectes tridimensionals i altres visualitzacions complexes. Requereix un esforç de programació a alt nivell per a les comunicacions entre l'applet i el servidor.

Després d'analitzar els diferents pros i contres es decideix utilitzar les llibreries jQuery de JavaScript per la seva compatibilitat amb tots els navegadors i sistemes operatius, sense haver d'instal·lar paquets addicionals.

És cert que fa uns anys hi havia usuaris que navegaven amb el JavaScript deshabilitat, però actualment aquesta opció ve habilitada per defecte a tots els navegadors i són tantes les planes que fan servir aquesta tecnologia que seria inviable per algú navegar amb aquest llenguatge script deshabilitat.

Respecte a la tria de jQuery enfront d'altres llibreries com MooTools, YUI o ExtJS, s'ha triat aquesta per ser la utilitzada en alguns dels mòduls del ACME, el que proporcionarà uniformitat en les eines utilitzades en aquesta plataforma.

També s'utilitza HTML (HyperText Markup Language) per a l'estructuració dels textos i fulls d'estil CSS (Cascading Style Sheet) per definir la presentació.

7.2 Estudi del llenguatge dels nuclis correctors

Llenguatge de programació per la correcció d'exercicis











- *Mathematica*
 - Es disposa de molta informació a Internet de la versió del Mathematica 8, que tot i que no és la versió que hi ha al servidor ACME, que té el Mathematica 5, ens servirà per obtenir molta informació i exemples de les comandes comunes a totes dues versions
 - Disposa de funcions específiques per a la resolució de problemes de lògica
 - Molt fàcil de programar, encara que no es disposi d'experiència prèvia
- *Prolog*
 - Tal com indica el seu nom, està orientat per a la PROgramació LÒGica
 - Dificultós per a programadors amb poca o nul·la experiència en aquest llenguatge de programació
- *PHP*
 - Molt fàcil de programar
 - No disposa de funcions específiques per a la resolució de problemes de lògica, pel que la correcció dels exercicis s'hauria d'implementar completament només disposant de les funcions per treballar amb expressions regulars.

Es decideix utilitzar el Mathematica per a la correcció dels exercicis, amb suport del PHP per al pre-processament de les cadenes de solució del professor i de l'alumne, per les seves avantatges en cadascun dels seus àmbits.

7.3 Eines utilitzades

Per a la realització d'aquest Projecte Final de Carrera s'ha utilitzat un portàtil amb Windows 7 i connexió a Internet.

Les eines que s'ha decidit utilitzar han estat les següents:

- Navegadors web
Chrome, Firefox i Internet Explorer, per comprovar la compatibilitat de les pàgines web creades. 
- Firebug
Plug-in gratuït del Firefox que permet debugar i monitoritzar JavaScript, HTML i CSS, i veure errors i alertes. 
- WAMP Server
Servidor web lliure i gratuït amb Apache, PHP i MySQL per Windows, per tal d'executar el codi PHP en local a les fases inicials del projecte. 
- WinSCP
Client SFTP gràfic, lliure i gratuït per a Windows, utilitzat per la transferència segura de fitxers entre el servidor acme4.udg.edu i l'ordinador local on s'editen. 
- NotePad++
Editor de codi font lliure per a Windows que suporta diversos llenguatges de programació, amb colorejat de sintaxis, plegament i desplegament de blocs i organització de fitxers en pestanyes. S'ha afegit el plug-in "FunctionList". 
- Files Comparer
Programa lliure per comparar fitxers i directoris. Molt útil per comparar diferents versions d'un mateix fitxer textual, destacant-ne visualment les diferències en una presentació costat per costat. 
- Microsoft Office
Paquet ofimàtic no lliure, utilitzat per mantindre el quadern de bitàcola del Projecte Final de Carrera, els calendaris i la redacció de la memòria. 
- Dia
Aplicació informàtica lliure per a la creació de diagrames, utilitzada per als diagrames d'activitat de la documentació. 
- AcmeEditor
Editor de problemes de l'ACME per la creació i validació d'exercicis. 
- Mathematica 8
Programari no lliure per a tot tipus de càlculs. Utilitzat en les proves en l'ordinador local de les instruccions en aquest llenguatge abans d'executar-ho al servidor ACME. 

8. Anàlisi i disseny del sistema

La fase d'anàlisi i disseny ha d'estudiar les necessitats del sistema i requeriments de l'aplicatiu i crear una solució de disseny que posteriorment pugui ser implementada en programari.

8.1 Anàlisi

Anàlisi del sistema

El projecte ACME és una plataforma de e-learning per a l'autocorrecció d'exercicis.

Es troba en producció al servidor acme.udg.edu, i en desenvolupament al servidor acme4.udg.edu, que passarà a ser el servidor de producció aquest estiu.

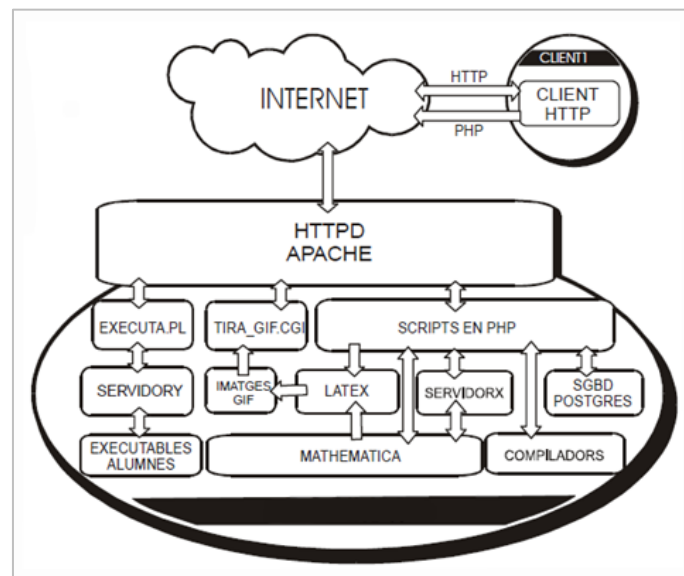


Figura 8.1 : Arquitectura client/servidor de la plataforma ACME.

Té una arquitectura client/servidor, on el servidor és el que genera planes web amb els exercicis, que l'usuari visualitza al seu navegador i que retorna al servidor amb les seves solucions per a ser corregides per aquest.

El flux de treball és el següent:

1. L'alumne es connecta via web a acme.udg.edu i introdueix el seu usuari i contrasenya
2. El servidor l'autentica i li dóna accés a les seves assignatures, generant una plana web
3. L'alumne tria el tema
4. El servidor genera una plana web amb els exercicis de l'assignatura
5. L'alumne tria un exercici
6. El servidor genera la plana web amb la interfície de l'exercici
7. L'alumne introdueix la seva solució a la interfície i envia a corregir

8. El servidor corregeix l'exercici de manera automàtica, guarda a la base de dades la solució i el resultat de la correcció i genera una plana web per a l'alumne amb aquesta informació.

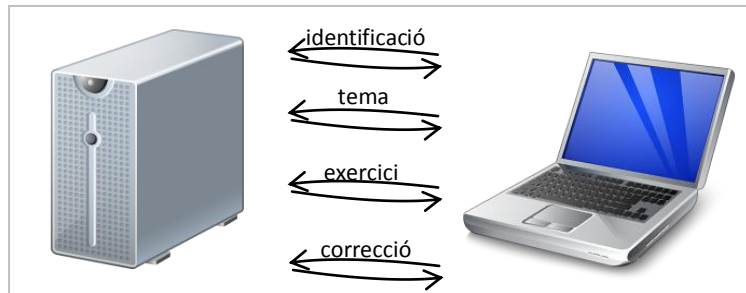


Figura 8.2 : Flux de treball client/servidor.

La programació dels exercicis en el servidor està feta amb orientació a objectes, el que permet generar noves tipologies d'exercicis de manera modular, senzilla i sense comprometre el funcionament de la resta.

Per fer un nou tipus d'exercici s'han de implementar tres blocs:

- Definició del fitxer d'enunciats:
S'haurà de definir la sintaxis que seguiran els fitxers d'enunciats per generar problemes i de la informació necessària per a que el nucli corrector pugui avaluar la solució de l'alumne.
- Creació de la interfície:
Interfície web per a la introducció de la solució de l'exercici per part del alumne. L'ACME ja disposa d'interfícies en llenguatges de programació com JavaScript, PHP, Applets de Java o Flash.
- Corrector automàtic:
Cada tipus d'exercici necessita un nucli corrector específic per a l'avaluació automàtica de la solució proporcionada per l'alumne a la interfície.

Sintaxis del fitxer d'enunciats

Abans de continuar, s'estudiarà més a fons el fitxer d'enunciats, ja que la sintaxis no pot ser del tot lliure, si no que el motor de l'ACME espera unes capçaleres, un bloc d'enunciats i un tercer bloc amb el codi Mathematica per a la resolució de l'exercici, tot plegat amb un format concret.

La capçalera té un format definit per la plataforma ACME on l'única variable és el número identificador associat a aquell tipus de problema, i que s'assigna de manera seqüencial.

```
numTipus
\begin{center}
\large {\bf Problema 1.01---ACME}
```

Per construir el bloc dels enunciats, cal aclarir que un fitxer està format per una sèrie d'exercicis. Cada alumne haurà de resoldre només un dels exercicis entre els que es preparin en aquest fitxer, que serà assignat de manera aleatòria pel sorteig que fa la plataforma ACME.

Per aquesta manera de treballar, l'ACME ens especifica l'embolcall del bloc dels enunciats, ens fixa la posició de l'enunciat genèric que apareixerà a tots els exercicis i permet generar una línia per a cada diferent exercici que es vulgui crear al problema.

Cadascuna d'aquestes línies està composta per un enunciat i la informació necessària per a que el nucli corrector pugui avaluar la solució, separats per uns símbols concrets que s'han de respectar.

Els enunciats i la informació s'hauran d'escriure en LaTeX.

```
\underline{Enunciats:}\par \vspace{0.5cm}
* % <E>
enunciatGenèric
\tt
\hspace{0.5 cm}
% <1>
{\bf P1}
% </E>
\par
\vspace{1cm}
\underline{Par`ametres:}\par \vspace{0.5cm}
% <P>
{\bf P1} \par % <1>
$ enunciat1 $\#$informacio1 $. \par
$ enunciat2 $\#$informacio2 $. \par
$ enunciat3 $\#$informacio3 $. \par
$ enunciat4 $\#$informacio4 $. \par
$ enunciat5 $\#$informacio5 $. \par
% </1>
% </P>
```

I finalment el bloc amb el codi de resolució en el llenguatge Mathematica.

Es recomana fer un 'Clear' de les variables que utilitzi, per evitar problemes de persistència.

Podem fer ús de les variables globals SO i P1, que contenen la solució del professor i la de l'alumne respectivament i que gestiona el nucli corrector de l'exercici.

És important que el codi Mathematica faci un 'Print' de les paraules claus per avaluar el resultat de la correcció, que poden ser 'Correcte', 'Incorrecte' i 'Incomprensible'.

```
\vspace{1cm}
\underline{Codi Matem`atica:}
\begin{verbatim}
SetOptions[ $\$Output$ , PageWidth->Infinity];
CodiResolucioMathematica
If[dacord,Print["Correcte"],Print["Incorrecte"]];
\end{verbatim}
```

Anàlisi dels exercicis

Anem ja a estudiar com ens condicionen els requisits d'aquest Projecte Final de Carrera en la creació de nous tipus d'exercicis.

Respecte al fitxer d'enunciats, en els exercicis de formalització lògica, tant de predicats com de proposicions, junt amb el redactat de dels enunciats hi ha d'haver la solució a l'exercici per a que el corrector pugui comparar la solució de l'alumne i la del professor.

En canvi, en els exercicis de contraexemples a raonaments lògics, s'haurà de proporcionar al corrector el raonament formalitzat per a que amb la introducció del contraexemple el nucli corrector sigui capaç d'avaluar si aquest demostra la incorrectesa del raonament. A més, s'estalvia la tasca de buscar tots els possibles contraexemples vàlids al professor.

En el referent a les interfícies, els exercicis de formalització de lògica de predicats requereixen d'una interfície amb els símbols dels quantificadors i els connectors propis de Lògica i un únic camp per a la introducció de dades. Els exercicis de formalització de lògica de proposicions, tot i que són molt similars, no utilitzen quantificadors, i en canvi, requereixen de varis camps per la introducció de cadascuna de les frases a formalitzar.

En canvi, la interfície dels exercicis de contraexemple és molt diferent, ja que no calen connectors ni quantificadors, si no introduir els valors de *cert* o *fals* per cadascun dels àtoms.

Finalment, analitzant els requeriments dels nuclis correctors dels exercicis de formalització es veu que hauran de ser capaços de comparar la formalització proporcionada pel professor al fitxer d'enunciats amb la solució proporcionada per l'alumne a la interfície.

En el cas dels exercicis de formalització de lògica de predicats es requerirà suport per a quantificadors i constants, mentre que la formalització de lògica de proposicions és molt més senzilla degut a l'absència de quantificadors. Realitzar la correcció per cadascuna de les frases de l'enunciat de proposicions només representaran vàries execucions del nucli corrector per a un mateix exercici.

El corrector per als exercicis de contraexemple torna a diferenciar-se clarament de les altres dues categories d'exercicis, ja que ha de ser capaç d'introduir el contraexemple a una formalització de lògica de predicats i avaluar si realment demostra la incorrectesa d'un raonament amb predicats.

Degut a les grans diferències que aquí s'han vist, s'haurà de crear a la plataforma ACME tres tipus d'exercicis nous i independents i que es corresponen als tres tipus d'exercicis que es demanen en els requeriments del Projecte Final de Carrera.

8.2 Disseny

En la fase de disseny es proposa una solució per al sistema analitzat.

Acabem de veure en la fase d'anàlisi com per complir els requeriments d'aquest Projecte Final de Carrera haurem de crear tres tipus d'exercicis nous i independents a la plataforma ACME.

També hem vist que per crear cada tipus nou d'exercici, s'ha de definir la sintaxis del fitxer d'enunciats, crear una interfície i un nucli de correcció automàtica específics, pel que en total es generaran tres fitxers d'enunciat, tres interfícies i tres correctors.

Anem a veure el disseny de cadascun dels exercicis per separat.

Disseny dels exercicis de formalització de lògica de predicats

Fitxer d'enunciats

El format de cadascun dels exercicis que componen un problema, s'ha vist que és el següent:

```
$ enunciat $# $informacio $. \par
```

Aquesta sintaxi s'ha d'adaptar als requeriments d'aquest tipus d'exercicis.

La part de l'enunciat que ha d'escriure's en LaTeX ja s'adapta a les nostres necessitats. Però hem d'especificar quina és la informació necessària per a que el nucli corrector pugui avaluar l'exercici.

Per aquest tipus, a cada línia d'exercici necessitarem:

- La solució correcta de la formalització, que haurà d'estar codificada per permetre els connectors i quantificadors de lògica en aquest fitxer.
- Els símbols dels àtoms permesos.
- Les variables permeses.
- Les constants, en cas d'haver-n'hi.

Es tria el símbol '\$' per separar cadascuna d'aquestes parts, amb el que el format de les línies d'exercici restarà:

```
$ enunciat $# $ SolucióCodificada %Àtoms %variables %Constants $. \par
```

Interfície

La interfície, que en el estudi dels llenguatges a utilitzar es va decidir que seria jQuery, haurà de tenir les següents parts:

1. Enunciat a formalitzar.
2. Botons amb els símbols lògics i quantificadors.
3. Botons amb les lletres permeses per als àtoms.
4. Botons amb les lletres permeses per a les variables, i si n'hi ha més d'una, el botó de coma.
5. Botons amb les lletres permeses com a constants, només quan l'exercici ho requereix.
6. Camp a on escriure la resposta.
7. Botó per enviar la solució a corregir.
8. Botó d'ajuda.

Corrector

A continuació es mostra el diagrama d'activitats utilitzat per dissenyar aquest tipus d'exercicis:

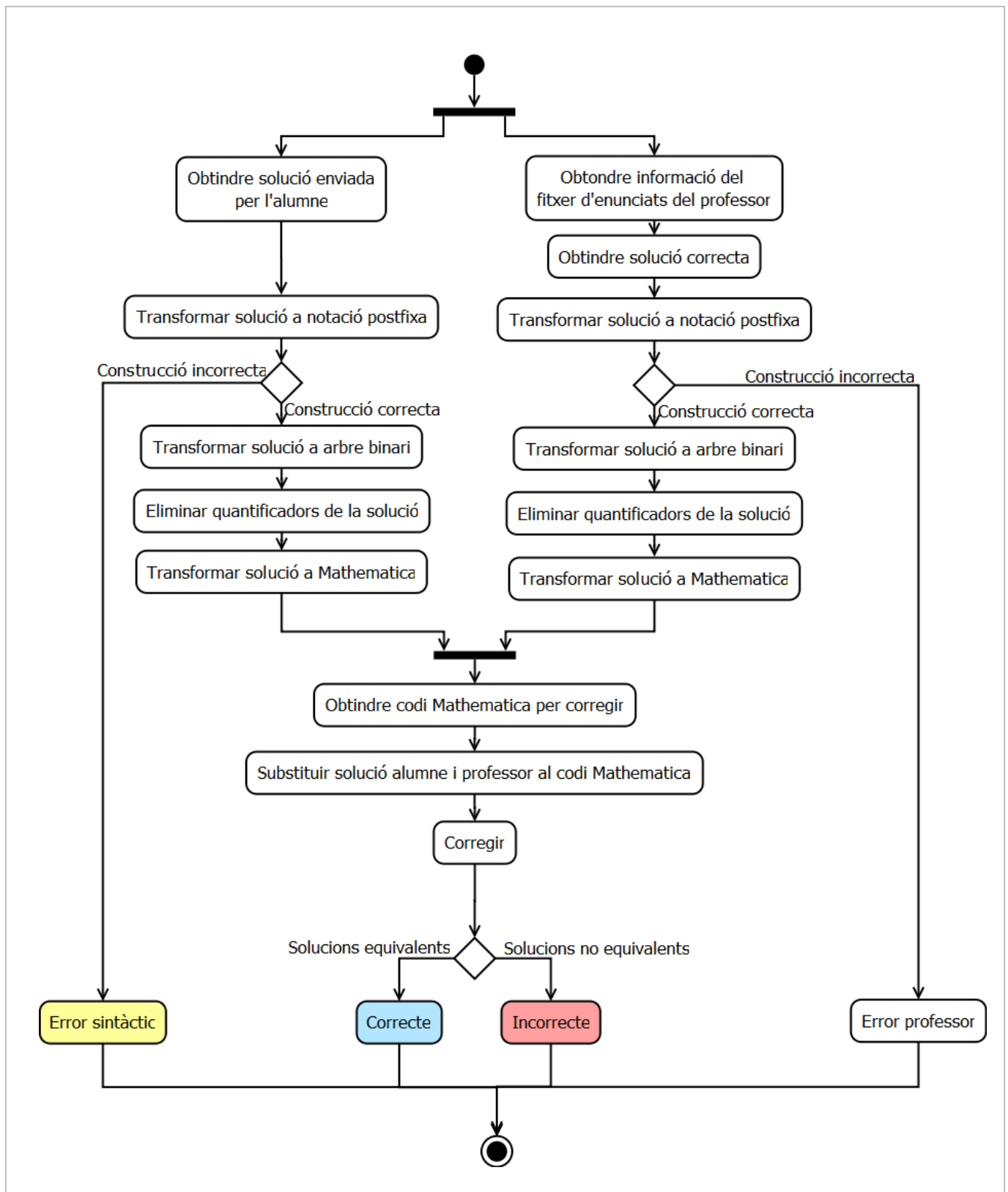


Figura 8.3 : Diagrama d'activitats dels exercicis de formalització de lògica de predicats.

Disseny dels exercicis de formalització de lògica de proposicions

Fitxer d'enunciats

El format genèric de cadascun dels exercicis que componen un problema, és:

```
$ enunciat $# $informacio $. \par
```

Aquesta sintaxi s'ha d'adaptar als requeriments d'aquest tipus d'exercicis.

Altre cop, la part de l'enunciat que ha d'escriure's en LaTeX ja s'adapta a les nostres necessitats i només hem d'especificar quina és la informació necessària per a que el nucli corrector pugui avaluar l'exercici.

Per aquest tipus, a cada línia d'exercici necessitarem:

- Número de frases de l'enunciat.
- La solució correcta de la formalització, que haurà d'estar codificada per permetre els connectors i quantificadors de lògica en aquest fitxer.
Les diferents frases formalitzades es separaran amb una coma.
- Els símbols dels àtoms permesos.

Es tria el símbol '\$' per separar cadascuna d'aquestes parts, amb el que el format de les línies d'exercici restarà:

```
$ enunciat $# $ NumLínies %SolucióCodificada %Àtoms $. \par
```

Interfície

La interfície, que en el estudi dels llenguatges a utilitzar es va decidir que seria jQuery, haurà de tenir les següents parts:

1. Enunciat a formalitzar.
2. Botons amb els símbols lògics.
3. Botons amb les lletres permeses per als àtoms.
4. Camps a on escriure la resposta, un per cada frase i en el mateix ordre en que es redacta l'enunciat.
5. Botó per enviar la solució a corregir.
6. Botó d'ajuda.

Corrector

Anem a veure el diagrama d'activitats utilitzat per dissenyar aquest tipus d'exercicis:

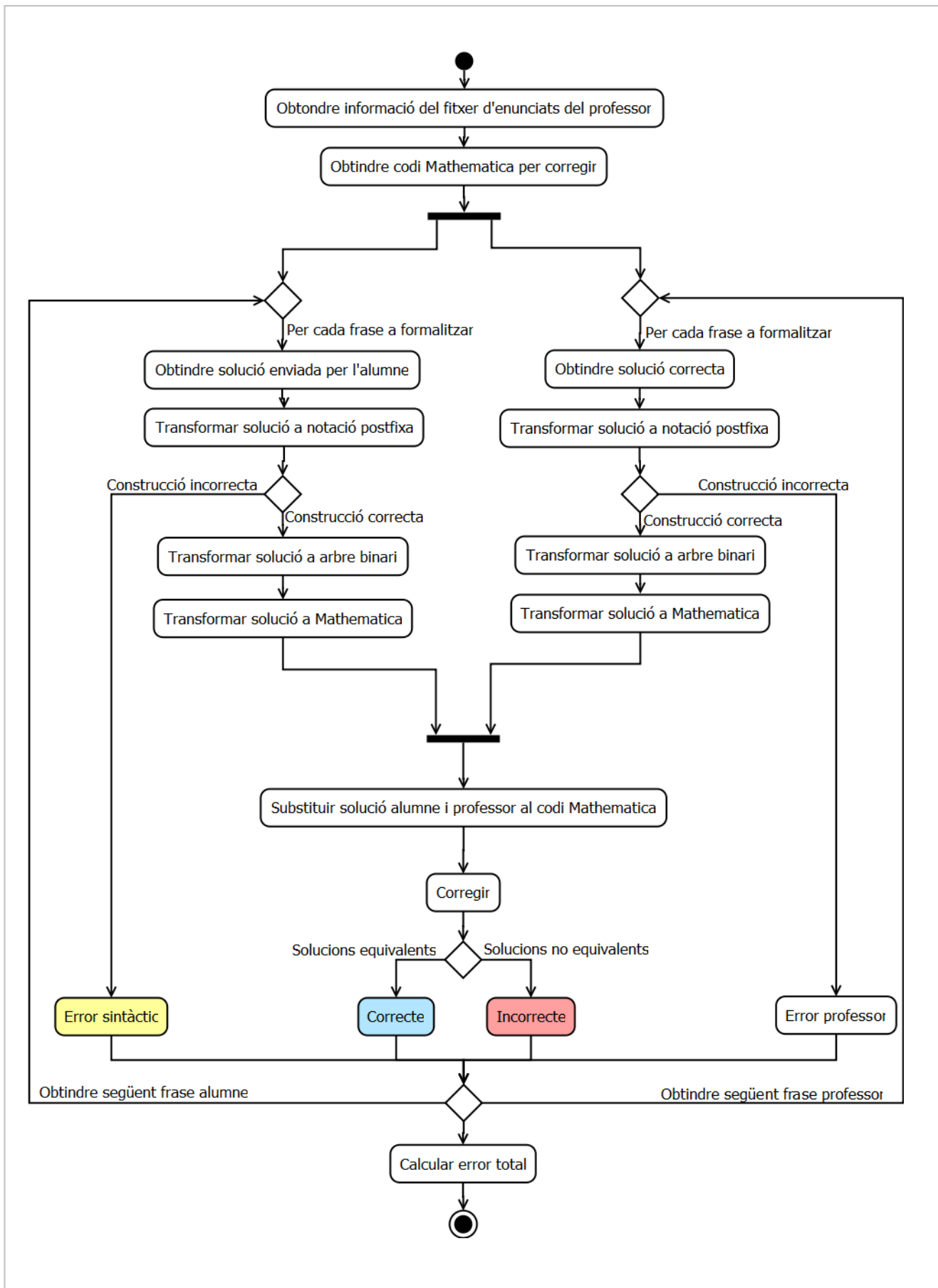


Figura 8.4 : Diagrama d'activitats dels exercicis de formalització de lògica de proposicions.

Disseny dels exercicis de contraexemples a raonaments amb predicats

Fitxer d'enunciats

Adaptarem la sintaxi general de la línia d'exercici a aquest tipus d'exercicis:

```
$ enunciat $# $informació $. \par
```

especificant quina és la informació necessària per a que el nucli corrector pugui avaluar l'exercici.

Per aquest tipus, a cada línia d'exercici necessitarem:

- El número de frases del raonament, és a dir, el número de premisses més 1 per la conclusió.
- La formalització de l'enunciat, en una codificació pròpia per als quantificadors i connectors lògics.
Cada frase es separarà de la següent amb una coma ',' i indistintament si el que ve a continuació és una premissa o ja es tracta de la conclusió.
- Els símbols dels àtoms permesos.
- Les variables permeses.

Es tria el símbol '\$' per separar cadascuna d'aquestes parts, amb el que el format de les línies d'exercici restarà:

```
$ enunciat $# $ NumFrases % EnunciatCodificat % Àtoms % variables $. \par
```

Interfície

La interfície, que en el estudi dels llenguatges a utilitzar es va decidir que seria jQuery, haurà de tenir les següents parts:

1. Enunciat amb el raonament formalitzat incorrecte al qual s'ha de trobar el contraexemple.
2. Desplegable per triar el número de referencians del contraexemple.
3. Desplegables per triar si cadascun dels àtoms amb cada referenciant és *cert* o *fals*.
4. Botó per enviar la solució a corregir.

Corrector

El diagrama d'activitats utilitzat per dissenyar aquest tipus d'exercicis és el següent:

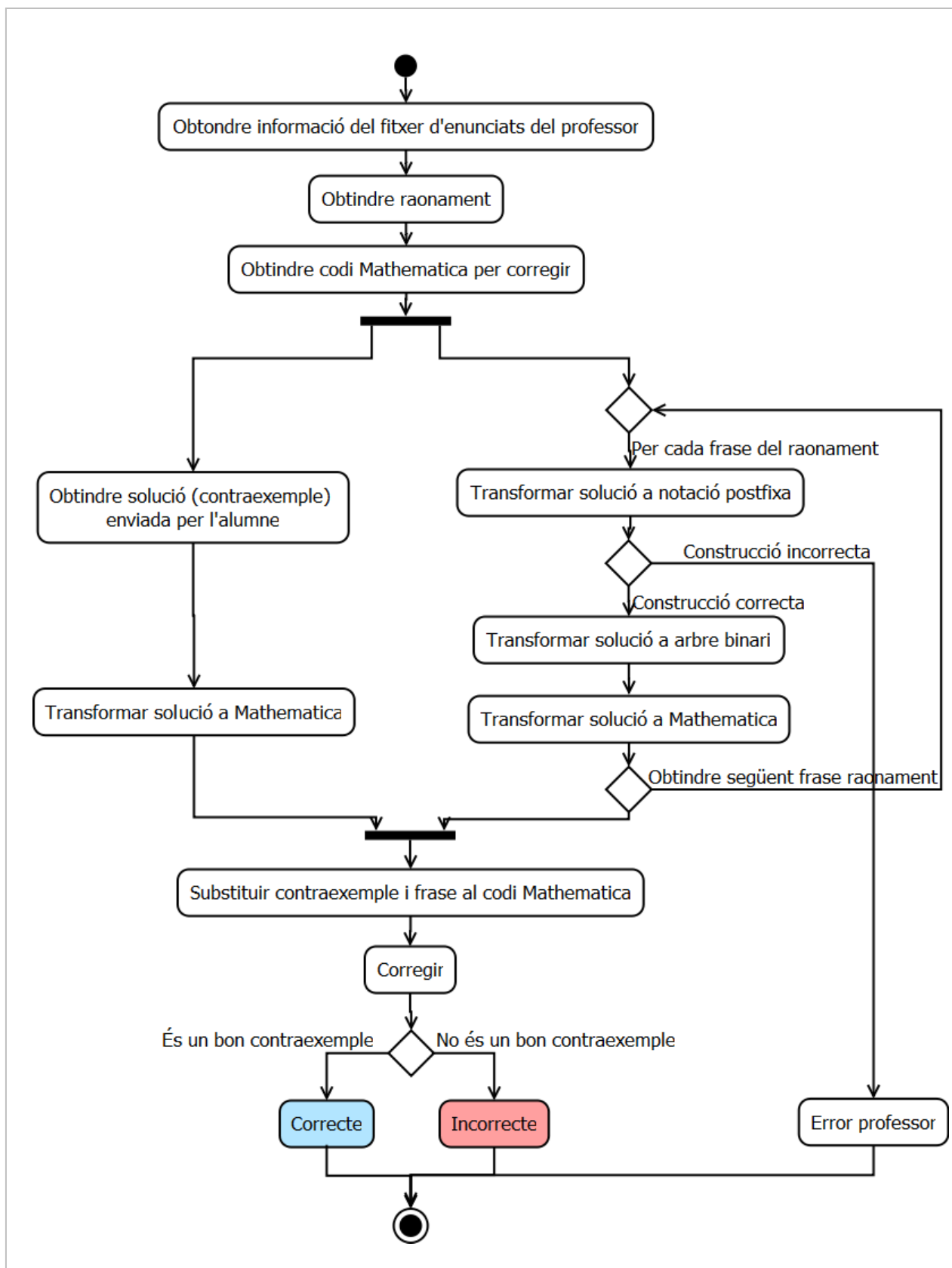


Figura 8.5 : Diagrama d'activitats dels exercicis de contraexemples de lògica de predicats.

9. Implementació i proves

En aquest apartat es donen detalls del procés de implementació, explicant els problemes apareguts i les solucions aportades.

A continuació es mostraran les proves de funcionament dels diferents tipus d'exercicis creats, com a usuari alumne i com a usuari professor, i del comportament enfront a entrades de dades buides o no permeses.

9.1 Implementació dels fitxers d'enunciats

L'estructura i sintaxis dels fitxers d'enunciats ja ha estat analitzada i dissenyada en l'apartat anterior.

La principal dificultat que es troba el professor a l'hora de generar aquests fitxers és en la utilització del llenguatge LaTeX per a la redacció dels enunciats.

Per tal de facilitar-li l'escriptura, s'han creat uns manuals d'ajuda on es detalla punt per punt cada part del fitxer d'enunciats i s'indica la codificació dels caràcters especials necessaris per aquests tipus de problemes en LaTeX, que són els següents:

| Símbol | LaTeX |
|---------------|--------------------------|
| \exists | <code>\exists</code> |
| \forall | <code>\forall</code> |
| \rightarrow | <code>\to</code> |
| \neg | <code>\neg</code> |
| \wedge | <code>\wedge</code> |
| \vee | <code>\vee</code> |
| \Rightarrow | <code>\Rightarrow</code> |

| Símbol | LaTeX |
|---------------|-----------------|
| Salt de línia | <code>\\</code> |

Taula 9.1 : Codificació LaTeX dels caràcters especials i símbols lògics.

La codificació pròpia que s’ha utilitzat per escriure la solució de la formalització dels exercicis de lògica de proposicions i de predicats i el raonament dels exercicis de contraexemples també es pot trobar en les ajudes creades per al professor.

Aquesta codificació ha estat consensuada amb el professor de l’assignatura de Lògica Narcís Coll, que ha optat per utilitzar dues lletres per símbol lògic que fan referència a com s’anomena aquest:

| Símbol lògic | Codificació |
|----------------------------|-------------|
| \forall (qualsevol) | qu |
| \exists (existeix) | ex |
| \rightarrow (implicació) | im |
| \neg (negació) | nn |
| \wedge (i) | ii |
| \vee (o) | oo |

Taula 9.2 : Codificació pròpia per als símbols lògics.

Com que serà el Mathematica l’encarregat de corregir aquestes formalitzacions, aquestes no podran tenir com a símbols d’àtom, variables o constant cap de les constants o funcions d’una lletra que té el Mathematica, i que són:

| | |
|---|---|
| D | Dóna la derivada parcial |
| E | La constant exponencial e |
| I | Representa la unitat imaginària $\sqrt{-1}$ |
| N | Dóna el valor numèric d’una expressió |
| O | Representa un terme d’ordre |

Taula 9.3 : Constants i Funcions de Mathematica d’una lletra.

Aleshores, per exemple si a l’enunciat se’ns demana formalitzar “Per Nadal els estudiants tenen vacances i només aquests”, no es podrà utilitzar ‘E(x)’ per representar ‘els estudiants’ si no que s’haurà de triar lletra de l’alfabet que no figuri a la taula anterior, per exemple S(x).

Abans de crear els fitxers d’enunciats dels diferents tipus d’exercicis només ens resta conèixer els números identificatius de cada tipus d’exercici que ha assignat la plataforma ACME i que són:

- 43 : identificador dels exercicis de formalització de lògica de predicats.
- 44 : identificador dels exercicis de formalització de lògica de proposicions.
- 45 : identificador dels exercicis de contraexemple de raonaments amb predicats.

Anem a veure uns fitxers d'enunciat d'exemple de cada tipus d'exercici, amb el codi Mathematica necessari per la correcció de cada tipus de fitxer ja inserit, i que s'explicarà més endavant.

Exemple de fitxer d'enunciat d'un problema de formalització de lògica de predicats amb dos exercicis:

```

43
\begin{center}
\large {\bf Problema 1.01---ACME}
\end{center}
\underline{Enunciats:}\par \vspace{0.5cm}
* % <E>
Formalitza l'enunciat seg\u{u}ent:
\tt
\hspace{0.5 cm}
% <1>
{\bf P1}
% </E>
\par
\vspace{1cm}
\underline{Par\u`ametes:}\par \vspace{0.5cm}
% <P>
{\bf P1} \par % <1>
$Les pomes i les taronges son gustoses i nutritives. \\\(P(x), T(x), G(x), V(x)) $\#$qu x(P(x) im
G(x) ii V(x) ) ii qu x(T(x) im G(x) ii V(x)) %PTGV %x % $. \par
$Hi ha aliments que només es poden menjar si són cuits. \\\(A(x), M(x), C(x)) $\#$ex x(A(x) ii
(M(x) im C(x))) %AMC %x % $. \par
% </1>
% </P>
\vspace{1cm}
\underline{Codi Matem\`atica:}
\begin{verbatim}
Clear[txtalumne,txtprofessor,dacord];
SetOptions[$Output, PageWidth->Infinity];
txtalumne=SO;
txtprofessor=P1;
dacord=False;
If[(TautologyQ[Equivalent[txtalumne, txtprofessor]]) === True, dacord = True];
If[dacord,Print["Correcte"],Print["Incorrecte"]];
\end{verbatim}

```

Exemple de fitxer d'enunciat d'un problema de formalització de lògica de proposicions amb dos exercicis:

```

44
\begin{center}
\large {\bf Problema 1.01---ACME}
\end{center}
\underline{Enunciats:}\par \vspace{0.5cm}
* % <E>
Formalitzeu les seg\u{u}ents proposicions:
\tt
\hspace{0.5 cm}
% <1>
{\bf P1}
% </E>
\par
\vspace{1cm}
\underline{Par\u`ametres:}\par \vspace{0.5cm}
% <P>
{\bf P1} \par % <1>
\$\$Si el fiscal \u00e9s deshonest i el gerent \u00e9s un estafador, el president \u00e9s un lladre. \\\Si el gerent \u00e9s un estafador per\u00f2 el president no \u00e9s cap lladre, el fiscal no \u00e9s pas deshonest.\\(F, G, P) $\#$2 %
F ii G im P , G ii nn P im nn F % FGP $. \par
\$\$Dormo o navego per internet o no tinc gana. \\\Quan menjo tinc set i no tinc fred. \\\Quan tinc gana no dormo i no navego per internet. \\\Quan no dormo menjo. \\\Quan tinc gana i no navego per internet tinc set. \\\(B, A, G, M, S, F) $\#$5 % B oo A oo nn G , M im S ii nn F , G im nn B ii nn A , nn B im M , G ii nn A im S % BAGMSF $. \par
% </1>
% </P>
\vspace{1cm}
\underline{Codi Matem\u`atica:}
\begin{verbatim}
Clear[txtalumne,txtprofessor,dacord];
SetOptions[$Output, PageWidth->Infinity];
txtalumne=SO;
txtprofessor=P1;
dacord=False;
If[(TautologyQ[Equivalent[txtalumne, txtprofessor]]) === True, dacord = True];
If[dacord,Print["Correcte"],Print["Incorrecte"]];
\end{verbatim}

```


Exemple de fitxer d'enunciat d'un problema de contraexemple de raonament de lògica de predicats amb dos exercicis:

```

45
\begin{center}
\large {\bf Problema 1.01---ACME}
\end{center}
\underline{Enunciats:}\par \vspace{0.5cm}
* % <E>
El següent esquema lògic és incorrecte. Troba un contraexemple:
\tt
\hspace{0.0 cm}
$
% <1>
{\bf P1}
$
% </E>
\par
\vspace{1cm}
\underline{Par`ametes:}\par \vspace{0.5cm}
% <P>
{\bf P1} \par % <1>
$\exists x (P(x) \wedge A(x)) \ \ \ \forall x (P(x) \to G(x)) \ \ \ \exists x (P(x) \wedge \neg A(x)) \ \ \
\Rightarrow \forall x (A(x) \to G(x)) \ $ \# \$ 4 % ex x (P(x) ii A(x)) , qu x (P(x) im G(x)) , ex x (P(x) ii nn
A(x)) , qu x (A(x) im G(x)) % AGP % x % $. \par
$\exists x (P(x) \vee R(x)) \ \ \ \exists x (Q(x) \vee R(x)) \ \ \ \forall x (P(x) \wedge Q(x) \to R(x)) \ \ \
\Rightarrow \exists x R(x) \ $ \# \$ 4 % ex x (P(x) oo R(x)) , ex x (Q(x) oo R(x)) , qu x (P(x) ii Q(x) im
R(x)) , ex x R(x) % PQR % x % $. \par
\vspace{1cm}
\underline{Codi Matem`atica:}
\begin{verbatim}
Clear[Regles,PremissesConclusio,dacord];
SetOptions[Output, PageWidth->Infinity];
Regles=SO;
PremissesConclusio=P1;
dacord=False;
PremissesConclusio;
If[(((Premises /. Regles) == True) && ((Conclusio /. Regles) == False)) === True, dacord = True];
If[dacord,Print["Correcte"],Print["Incorrecte"]];
\end{verbatim}

```

9.2 Implementació de les interfícies

La plataforma ACME està programada amb orientació a objectes, amb uns mètodes que donen accés a parts concretes de la plana web.

Per fer un nou tipus d'exercici s'haurà de crear un objecte nou, amb les mateixes variables i mètodes, pel que la manera més ràpida de generar aquest codi és agafar el fitxer d'un tipus d'exercici similar, duplicar-lo i substituir totes les ocurrences de la paraula 'tipus_xx' per el nou 'tipus_yy' que estem creant.

El tipus d'exercici que s'ha agafat com a model és el tipus d'exercici amb número '12', que identifica els problemes matemàtics que donen missatges d'error. Amb el fitxer 'tipus_12.php' s'han creat els fitxers 'tipus_43.php', 'tipus_44.php' i 'tipus_45.php'.

La interfície d'usuari està formada per dues pantalles, una primera on l'alumne visualitza l'exercici i introdueix la solució, i una segona on se li indica el resultat de la correcció.

Un cop l'exercici ha estat resolt satisfactòriament, a la primera pantalla en lloc de veure's el bloc per introduir la solució es veurà la solució correcta que es va enviar.

A més, es pot accedir a l'historial de solucions enviades amb els resultats de les correccions amb el botó de "Veure solucions enviades", que es mostra en una finestra emergent.

Els mètodes que ens donaran accés a les diferents zones d'aquestes planes web són:

- **Form_Enviar_Solucio**
Genera la interfície per a l'alumne i consulta la base de dades per conèixer si l'exercici ja s'havia intentat resoldre, per inicialitzar-la amb els valors de l'última solució.
- **Form_Enviar_Solucio_Profe**
Genera la interfície si l'usuari té accés de professor.
- **Visualitzar_Solucio**
Mostra les solucions enviades a la finestra emergent d'historial i quan s'ha resolt.
- **Escriure_Resultat**
Mostra el resultat de les solucions enviades a la finestra emergent d'historial.
- **Mostrar_Solucions_Enviades**
Mostra la solució enviada a la plana de resultat de la correcció.
- **Corregir_Problema**
Crida al nucli corrector per corregir la solució proporcionada per l'alumne i guarda el resultat a la base de dades.
- **Corregir_Problema_Visual**
Crida al nucli corrector quan l'usuari té accés de professor, i mostra per pantalla informació del codi Mathematica executat, però no guarda el resultat a la base de dades.

- **Escriure_Resultat_Correccio**
 Genera un bloc amb el missatge resultant de la correcció, de color blau per a resultats correctes, de color vermell per als incorrectes i de color groc per als errors sintàctics.

En la següent figura es pot veure a quina part de les planes web afecten aquests mètodes:

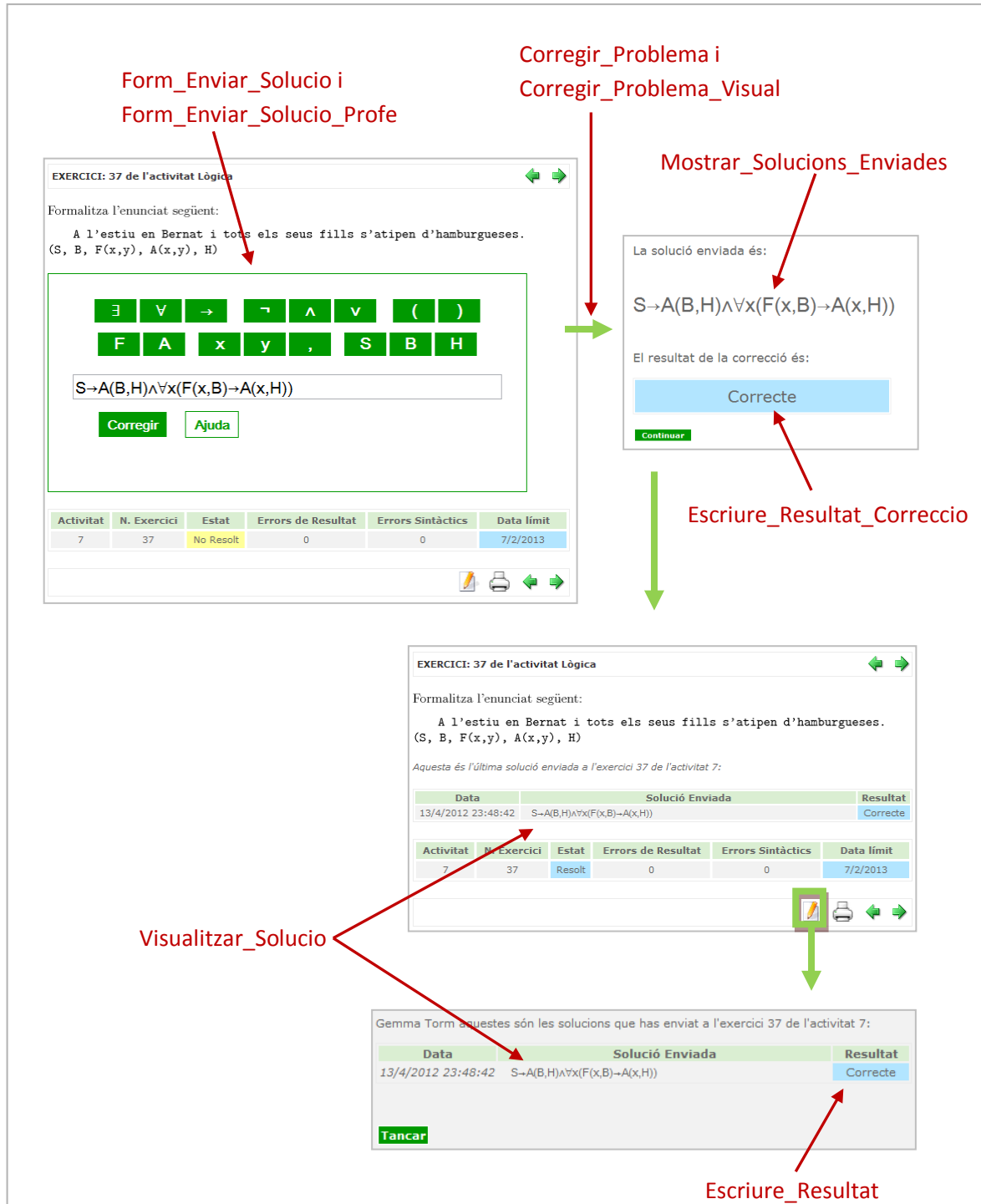


Figura 9.1 : Mètodes relacionats amb la interfície a la plataforma ACME.

9.3 Implementació dels nuclis correctors

En aquesta secció es detallarà els problemes que han aparegut en la implementació dels nuclis correctors, i la solució aportada.

Actualització de la versió del Mathematica al servidor ACME

En la fase d'estudi del llenguatge a utilitzar en els nuclis correctors es va veure que el Mathematica era la opció més adequada per fer la correcció automàtica.

Però per corregir els exercicis de formalització, la versió del Mathematica que tenia originalment el servidor ACME, la versió 5.2, no era capaç d'avaluar l'equivalència entre la solució aportada pel professor al fitxer d'enunciats i la solució introduïda per l'alumne.

Per exemple, en lògica de proposicions, si s'ha aplicat la regla de resolució, el Mathematica 5 no és capaç d'avaluar l'equivalència, i només amb la combinació de les funcions *TautologyQ* i *Equivalent* del Mathematica 8 podem corregir l'exercici.

I com veurem en el punt següent, per resoldre els exercicis de lògica de predicats haurem de fer una transformació a lògica de proposicions limitant el número d'elements de l'univers, pel que aquesta limitació afectarà a tots dos tipus de problemes de formalització.

Com que al Mathematica 5 no es disposa de la funció *Equivalent*, tampoc es poden fer exercicis que utilitzin el connector de doble implicació, l'anomenat 'si i només si': $A \Leftrightarrow B$

Aquestes dificultats, van ser expressades al tutor del Projecte Final de Carrera, que en acord amb els altres administradors de la plataforma ACME van considerar oportú actualitzar el Mathematica al servidor acme4.udg.edu a la versió 8.

Equivalència entre formalitzacions de lògica de predicats

El Mathematica 8 no és capaç de comparar si dues formalitzacions de lògica de predicats són equivalents, ja que hauria de comprovar si per a tots els elements de l'univers infinit i amb totes les combinacions de valors possibles de *cert* i *fals* les dues formalitzacions donen els mateixos resultats.

La solució que s'ha implementat, en consens amb el professor de l'assignatura de Lògica Narcís Coll, ha estat la de limitar el número d'elements de l'univers per tal d'eliminar els quantificadors.

Per exemple, ' $\forall x P(x)$ ' amb un univers de tres elements es transformaria en ' $P(a) \wedge P(b) \wedge P(c)$ ', i ' $\exists x P(x)$ ' en ' $P(a) \vee P(b) \vee P(c)$ '.

Inicialment es va pensar en fixar el número d'elements de l'univers a 25, perquè és clar que com més gran sigui l'univers, més possibilitats que el resultat de l'avaluació de la equivalència sigui correcta, però a un cost de càlcul molt gran.

A l'eliminar el quantificador de 'A(x)' es transformaria en 'A(1), ...,A(25)', es generen 25 àtoms diferents.

Però si es tenen dues variables, com per exemple a A(x,y), a l'eliminar el quantificador es transformaria en 'A(1,1),...,A(1,25),A(2,1),...,A(2,25),...,A(25,1),...A(25,25)', i tindrem 25x25=625 àtoms diferents.

Per conèixer el temps de càlcul necessari segons el número d'elements, s'han realitzat unes proves d'execució a un ordinador personal.

Les característiques d'aquest ordinador són: processador Intel Core i5-2410M, CPU 2,30 GHz, 4 GB de memòria RAM amb un sistema operatiu Windows 7 de 64 bits i el Mathematica amb la versió 8.

En els exercicis de l'assignatura de Lògica no es realitzen exercicis amb més de dues variables, pel que les proves s'han fet amb predicats d'una i dues variables amb un número creixent de elements a l'univers.

Si monitoritzem el temps d'execució d'avaluar la equivalència entre dos raonaments de només una variable

$$\forall x(P(x) \rightarrow G(x) \wedge V(x))$$

$$\forall x(T(x) \rightarrow G(x) \wedge V(x))$$

podem veure que el temps d'execució augmenta exponencialment, i que a partir de només 15 elements el temps d'execució seria inadequat per una aplicació com la plataforma ACME, on prima la velocitat de correcció de l'exercici per a que l'alumne tingui un gran nivell de interactivitat i immediatesa:

| Número d'elements | Segons |
|-------------------|----------|
| 1 | 0. |
| 2 | 0. |
| 10 | 0.016 |
| 12 | 0.016 |
| 15 | 0.312 |
| 18 | 6.224 |
| 19 | 13.618 |
| 20 | 29.516 |
| 21 | 64.787 |
| 22 | 141.524 |
| 25 | Es penja |

Taula 9.4 : Estudi del temps d'execució necessari per l'avaluació de l'equivalència entre dos predicats d'una variable.

Al realitzar l'execució per avaluar la equivalència entre dos raonaments de dues variables:

$$\exists x (F(x) \wedge \forall y (A(x,y) \rightarrow F(y))) \quad \exists x (\neg \exists y A(x,y) \wedge \neg F(y) \wedge F(x))$$

ràpidament es dispara el temps de processament, i amb només 5 elements es torna inviable

| Número d'elements | Segons |
|-------------------|---------|
| 1 | 0.000 |
| 2 | 0.000 |
| 3 | 0.016 |
| 4 | 0.608 |
| 5 | 548.016 |

Taula 9.5 : Estudi del temps d'execució necessari per l'avaluació de l'equivalència entre dos predicats de dos variables.

Val a dir que si els dos raonaments no són equivalents, en el moment que es trobi una combinació que els faci diferents, l'execució es parerà. Per aquest fet s'ha tingut cura de fer l'estudi per al cas més costós, sobre dos enunciats amb dues variables i que són equivalents.

Basant-nos en aquest petit estudi, es va decidir que el número d'elements que s'utilitzarien per eliminar els quantificadors seria de 4, ja que amb 5 elements seria inviable tant per un ordinador personal com per a un servidor.

Aquest número podria semblar una limitació, però per al nivell d'exercicis que es fan en l'assignatura de Lògica es creu que serà suficient.

Preprocessament de les solucions

Quan es va tenir implementat el codi Mathematica per la correcció dels exercicis de Lògica, només restava transformar la solució que introdueix l'alumne a la interfície i la solució que introdueix el professor al fitxer d'enunciats a llenguatge Mathematica per a la correcció dels exercicis de formalització.

Aquest preprocessament de les solucions abans de ser enviades al Mathematica també és necessari en els exercicis de contraexemple, ja que el professor ha d'introduir el raonament de lògica de predicats a la informació dels exercicis al fitxer d'enunciats.

Es tracta doncs de transformar expressions de l'estil de
a llenguatge Mathematica

$$G \wedge \neg P \rightarrow \neg F$$

$$\text{Implies}[(G \ \&\& \ ! P) , \ ! F]$$

I en el cas de formalitzacions de predicats com
en llenguatge Mathematica seria

$$\neg \forall x P(x) \rightarrow \exists x Q(x) \wedge \neg \forall x R(x)$$

$$\text{Implies} [\ !\text{ForAll}[x,P[x]] , (\text{Exists}[x,Q[x]] \ \&\& \ !\text{ForAll}[x,R[x]])]$$

A més, tal com hem vist, haurem d'eliminar els quantificadors substituint-hi un univers de 4 elements abans d'enviar les solucions al codi corrector en Mathematica.

L'exemple que acabem de veure ha de quedar de la forma:

$$\text{Implies [!(P[1]\&\&P[2]\&\&P[3]\&\&P[4]) , (Q[1]||Q[2]||Q[3]||Q[4]) \&\& (R[1]\&\&R[2]\&\&R[3]\&\&R[4])] }$$

La part més complexa d'aquest procediment és canviar l'equació en la forma de la formalització a la forma del Mathematica.

El problema s'ataca en dues fases. Primer convertirem la solució normalitzada de notació infix a postfixa, tenint en compte les prioritats dels operadors i els parèntesis, per a continuació molt més còmodament, traduir la notació postfixa a arbre binari, amb el que serà immediat generar l'equació en llenguatge Mathematica.

L'algoritme "Shunting Yard" ens permet transformar una equació infix a postfixa, i s'anomena així (en anglès "pati de classificació") perquè el seu funcionament s'assembla a un pati de classificació de ferrocarrils, on els combois es poden empilar en una via morta per als seu reordenament.

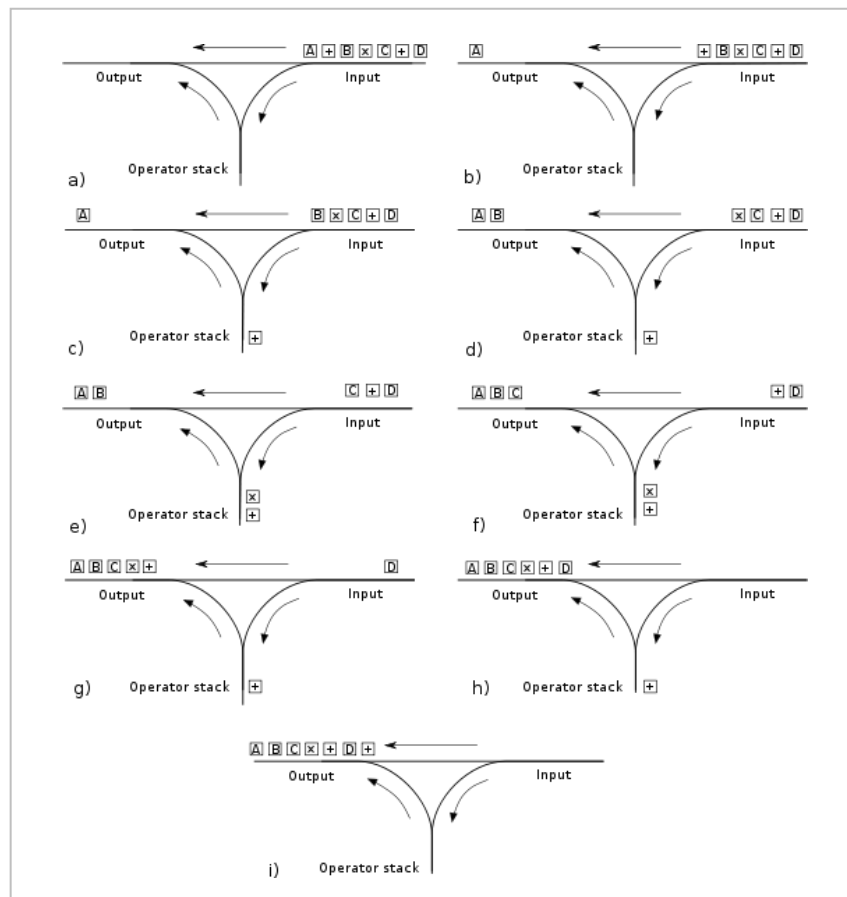


Figura 9.2 : Algoritme Shunting Yard.

9.4 Proves de funcionament de formalització de predicats

Anem a veure com han quedat els diferents exercicis.

Quan l'alumne visualitza per primer cop un exercici, se li mostra l'enunciat a resoldre, i la interfície per donar la seva solució.

Aquesta interfície s'adapta a cada enunciat, ja que proporciona botons per a les lletres dels àtoms i variables que es necessiten, i que el professor ha proporcionat al fitxer de l'enunciat.

A més, a la part inferior se li indica l'estat de l'exercici, en aquest cas, encara no ha estat resolt i no té cap intent de solució amb errors.

EXERCICI: 37 de l'activitat Lògica  

Formalitzza l'enunciat següent:

A l'estiu en Bernat i tots els seus fills s'atipen d'hamburgueses.
(S, B, F(x,y), A(x,y), H)

\exists \forall \rightarrow \neg \wedge \vee ()

F A x y , S B H

| Activitat | N. Exercici | Estat | Error de Resultat | Error Sintàctic | Data límit |
|-----------|-------------|-----------|-------------------|-----------------|------------|
| 7 | 37 | No Resolt | 0 | 0 | 7/2/2013 |

Figura 9.3 : Interfície d'entrada de la solució dels exercicis de formalització de lògica de predicats.

Per facilitar i agilitzar la introducció de la solució per part de l'alumne, també es dona l'oportunitat d'escriure utilitzant únicament el teclat, sense fer clic amb el ratolí als botons proporcionats.

Per a les lletres dels àtoms, les variables, els parèntesis i la coma (en els exercicis que és necessària apareix un nou botó amb aquest símbol), el teclat de l'ordinador ja disposa de les tecles necessàries.

Per als connectors de lògica, s'han creat unes dreceres de teclat associades als números de l'1 al 6, i s'informa a l'ajuda d'aquesta opció, tal i com podem veure a la següent figura:



Figura 9.4 : Interfície amb Ajuda indicant les dreceres de teclat.

Si l'alumne introdueix una solució incorrecta, com per exemple:

EXERCICI: 37 de l'activitat Lògica

Formalitzo l'enunciat següent:

A l'estiu en Bernat i tots els seus fills s'atipen d'hamburgueses.
(S, B, F(x,y), A(x,y), H)

\exists \forall \rightarrow \neg \wedge \vee ()
 F A x y , S B H

$\neg S \rightarrow A(B,H) \wedge \forall x (F(x,B) \rightarrow A(x,H))$

Corregir Ajuda

| Activitat | N. Exercici | Estat | Error de Resultat | Error de Sintàctic | Data límit |
|-----------|-------------|-----------|-------------------|--------------------|------------|
| 7 | 37 | No Resolt | 0 | 0 | 7/2/2013 |

Figura 9.5 : Exemple de formalització de predicats incorrecta.

al prémer el botó de “Corregir”, el corrector mostrarà per pantalla que el resultat proporcionat per l’alumne és Incorrecte:

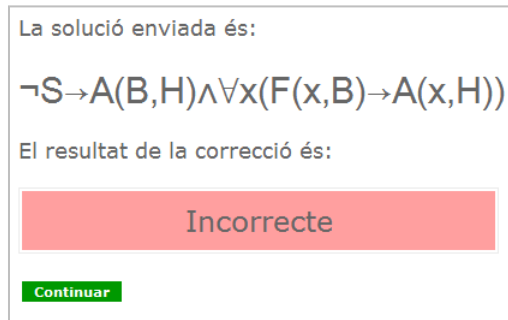


Figura 9.6 : Resultat incorrecte en formalització de predicats.

Quan prem el botó de continuar, es torna a la pantalla de la interfície amb l’última solució proposada per l’alumne com a valor per defecte.

També veiem que el comptador de “Error de Resultat” s’ha incrementat en 1.

Si ara introduïm la solució correcta, que en aquest cas és:

EXERCICI: 37 de l'activitat Lògica

Formalitzza l'enunciat següent:

A l'estiu en Bernat i tots els seus fills s'atipen d'hamburgueses.
 (S, B, F(x,y), A(x,y), H)

S→A(B,H)∧∀x(F(x,B)→A(x,H))

| Activitat | N. Exercici | Estat | Error de Resultat | Error Sintàctic | Data límit |
|-----------|-------------|-----------|-------------------|-----------------|------------|
| 7 | 37 | No Resolt | 1 | 0 | 7/2/2013 |

Figura 9.7 : Exemple de formalització de predicats correcta.

Al prémer el botó de “Corregir”, el corrector mostra per pantalla que el resultat proporcionat és Correcte:

La solució enviada és:

$$S \rightarrow A(B, H) \wedge \forall x (F(x, B) \rightarrow A(x, H))$$

El resultat de la correcció és:

Correcte

[Continuar](#)

Figura 9.8 : Resultat correcte de formalització de predicats.

El fet que el corrector no compari simplement la cadena de la solució introduïda per l'alumne i la proporcionada per el professor al fitxer de l'exercici, sinó que processa la solució de l'alumne traduint-la en una expressió en codi Mathematica per a que el Mathematica miri si són equivalents, permet que el corrector detecti com a correcta la següent expressió equivalent com a correcta:

EXERCICI: 37 de l'activitat Lògica ← →

Formalitza l'enunciat següent:

A l'estiu en Bernat i tots els seus fills s'atipen d'hamburgueses.
(S, B, F(x,y), A(x,y), H)

∃ ∀ → ¬ ∧ ∨ ()

F A x y , S B H

$S \rightarrow A(B, H) \wedge \forall x (\neg F(x, B) \vee A(x, H))$

Corregir
Ajuda

| Activitat | N. Exercici | Estat | Error de Resultat | Error Sintàctic | Data límit |
|-----------|-------------|-----------|-------------------|-----------------|------------|
| 7 | 37 | No Resolt | 1 | 0 | 7/2/2013 |

← →

Figura 9.9 : Exemple de formalització de predicats equivalent.

I efectivament al ser corregit ho dona com a correcte:

La solució enviada és:

$$S \rightarrow A(B, H) \wedge \forall x (\neg F(x, B) \vee A(x, H))$$

El resultat de la correcció és:

Correcte

Continuar

Figura 9.10 : Resultat correcte de formalització de predicats equivalent.

Un cop s'ha introduït la solució correcta, si tornem a la primera plana veurem que enlloc de la interfície apareix el resultat:

EXERCICI: 37 de l'activitat Lògica

Formalitza l'enunciat següent:

A l'estiu en Bernat i tots els seus fills s'atipen d'hamburgueses.
(S, B, F(x,y), A(x,y), H)

Aquesta és l'última solució enviada a l'exercici 37 de l'activitat 7:

| Data | Solució Enviada | Resultat |
|--------------------|--|----------|
| 16/4/2012 19:30:26 | $S \rightarrow A(B, H) \wedge \forall x (\neg F(x, B) \vee A(x, H))$ | Correcte |

| Activitat | N. Exercici | Estat | Errors de Resultat | Errors Sintàctics | Data límit |
|-----------|-------------|--------|--------------------|-------------------|------------|
| 7 | 37 | Resolt | 1 | 0 | 7/2/2013 |

✎ 🖨️ ⬅️ ➡️

Figura 9.11 : Interfície formalització de predicats quan l'exercici ha estat resolt.

A més a més de dir si la solució enviada és correcta o incorrecta, la plataforma disposa de la sortida “Error sintàctic” quan la solució de l’alumne no està ben escrita i per tant no pot ser interpretada pel corrector.

En aquest tipus d’exercici, quan es dóna aquesta situació, a causa de parèntesis mal tancats o per caràcters no permesos, es dóna a l’alumne un missatge informant-lo de la causa de l’error.

Anem a veure un exercici on introduïrem una solució amb un parèntesis de més, error que és molt fàcil de produir-se si l’alumne es descompta.

EXERCICI: 33 de l'activitat Lògica ← →

Formalitzza l'enunciat següent:

Hi ha francesos que són amics de tots els catalans.
 $(F(x), A(x,y), C(x))$

∃
∀
→
¬
∧
∨
(
)

F
A
C
x
y
,

$\exists x(F(x)\wedge\forall y(C(y)\rightarrow A(x,y)))$

Corregir
Ajuda

| Activitat | N. Exercici | Estat | Error de Resultat | Error Sintàctic | Data límit |
|-----------|-------------|-----------|-------------------|-----------------|------------|
| 7 | 33 | No Resolt | 0 | 0 | 7/2/2013 |

← →

Figura 9.12 : Exemple de formalització de predicats amb parèntesis de més.

El corrector ens indica que és un error sintàctic de parèntesis, i no el considera error com a tal, el que es visualitza la plataforma ACME amb un fons de color groc:

La solució enviada és:

$\exists x(F(x)\wedge\forall y(C(y)\rightarrow A(x,y)))$

El resultat de la correcció és:

Error sintàctic de parèntesis

Continuar

Figura 9.13 : Resultat d’error sintàctic de parèntesis en formalització de lògica de predicats.

Si ara premem el botó de continuar, comprovem a la pantalla inicial que no ens ha augmentat el comptador de “Errors de Resultat” sinó el de “Errors Sintàctics”.

Si ara introduïm la solució correcta, i premem el botó de “Corregir”

EXERCICI: 33 de l'activitat Lògica ← →

Formalitzo l'enunciat següent:
 Hi ha francesos que són amics de tots els catalans.
 (F(x), A(x,y), C(x))

∃ ∀ → ¬ ∧ ∨ ()

F A C x y ,

$\exists x(F(x)\wedge\forall y(C(y)\rightarrow A(x,y)))$

Corregir
Ajuda

| Activitat | N. Exercici | Estat | Errors de Resultat | Errors Sintàctics | Data límit |
|-----------|-------------|-----------|--------------------|-------------------|------------|
| 7 | 33 | No Result | 0 | 1 | 7/2/2013 |

← →

Figura 9.14 : Exemple de formalització de predicats amb parèntesis corregits.

Efectivament el corrector ens indica que el resultat és Correcte:

La solució enviada és:

$$\exists x(F(x)\wedge\forall y(C(y)\rightarrow A(x,y)))$$

El resultat de la correcció és:


Correcte

Continuar

Figura 9.15 : Resultat correcte de formalització de predicats corregida.

Si en un exercici utilitzem el teclat per escriure la solució i premem una tecla que no forma part de les lletres i símbols de l'exercici, el corrector ens donarà un missatge de caràcters no permesos.

Per exemple:

EXERCICI: 34 de l'activitat Lògica  

Formalitzo l'enunciat següent:

No pas tothom és ric, culte i educat, ni no tots els rics són educats i cultes.
 $(R(x), C(x), D(x))$

∃
∀
→
¬
∧
∨
(
)

R
C
D
x

A(x)

Corregir
Ajuda

| Activitat | N. Exercici | Estat | Error de Resultat | Error Sintàctic | Data límit |
|-----------|-------------|-----------|-------------------|-----------------|------------|
| 7 | 34 | No Resolt | 0 | 0 | 7/2/2013 |










Figura 9.16 : Exemple de formalització de predicats amb caràcters no permesos.

genera la correcció següent:

La solució enviada és:

A(x)

El resultat de la correcció és:



Hi ha caràcters no permesos a la solució introduïda

Continuar

Figura 9.17 : Resultat de caràcters no permesos en formalització de lògica de predicats.

A la pantalla de la interfície veurem que quan l’alumne ha introduït una solució amb error sintàctic, s’incrementa el comptador de “Errors Sintàctics” enlloc del de “Errors de Resultat”, ja que no es tracta d’un error de concepte, i es porten comptadors separats.

Si per exemple al realitzar un exercici introduïm una solució amb error sintàctic de caràcters no permesos, seguit d’un d’error sintàctic de parèntesis, després una solució incorrecta, i després finalment la correcta, els comptadors ens indiquen que hem fet 2 “Errors Sintàctics”, 1 “Errors de Resultat” i que finalment ha estat Result.

EXERCICI: 34 de l'activitat Lògica  

Formalitzat l'enunciat següent:

No pas tothom és ric, culte i educat, ni no tots els rics són educats i cultes.
 $(R(x), C(x), D(x))$

Aquesta és l'última solució enviada a l'exercici 34 de l'activitat 7:

| Data | Solució Enviada | Resultat |
|--------------------|---|----------|
| 16/4/2012 19:47:33 | $\neg\forall x(R(x)\wedge C(x)\wedge D(x))\wedge\neg\forall x(R(x)\rightarrow D(x)\wedge C(x))$ | Correcte |

| Activitat | N. Exercici | Estat | Errors de Resultat | Errors Sintàctics | Data límit |
|-----------|-------------|--------|--------------------|-------------------|------------|
| 7 | 34 | Result | 1 | 2 | 7/2/2013 |






   

Figura 9.18 : Interfície formalització de predicats quan l’exercici ha estat result després de fer errors sintàctics.

Si anem al llistat de solucions de l’exercici, accessible des de la pantalla d’interfície amb el  botó, podrem veure que s’informa de l’historial de solucions, de les correctes, les incorrectes i dels tipus errors sintàctics produïts.

| Data | Solució Enviada | Resultat |
|-------------------|---|-----------------|
| 4/4/2012 20:45:14 | A(x) [Hi ha caràcters no permesos a la solució introduïda] | Error Sintàctic |
| 4/4/2012 21:17:18 | R(x) [Error sintàctic de parèntesis] | Error Sintàctic |
| 4/4/2012 21:27:02 | $\neg\exists x(R(x)\wedge C(x)\wedge D(x))\wedge\neg\forall x(R(x)\rightarrow D(x)\wedge C(x))$ | Incorrecte |
| 4/4/2012 21:27:18 | $\neg\forall x(R(x)\wedge C(x)\wedge D(x))\wedge\neg\forall x(R(x)\rightarrow D(x)\wedge C(x))$ | Correcte |

Tancar

Figura 9.19 : Historial de solucions i correccions en lògica de predicats.

9.5 Proves de funcionament de formalització de proposicions

Els exercicis de formalització de proposicions són similars als de lògica de predicats, ja que els exercicis que es realitzen a l'assignatura estan formats per varies clàusules.

D'aquesta manera, la interfície ha de mostrar tantes línies a omplir per l'alumne com línies tingui l'enunciat.

Aquest tipus d'exercici també adapta els botons dels àtoms als utilitzats per l'enunciat, i no disposa de botons de quantificadors ni variables ja que en aquest tipus d'exercici no s'utilitzen.

Anem a veure un exemple d'aquests exercicis, on introduïrem una solució incorrecta, ja que utilitzarem la lletra 'V', escrita des de el teclat de l'ordinador, i que no és un dels àtoms permesos:

EXERCICI: 43 de l'activitat Lògica ← →

Formalitzeu les següents proposicions:

Si em gradués en informàtica, aniria a fer un màster a l'estranger.
Si no surto massa per les nits em graduaré en Informàtica.
Si no vull treballar en un banc cal que no surti massa.
Si treballés en un banc acabaria casant-me.
Si no anés a fer un màster a l'estranger acabaria casant-me.
(G, M, N, B, C)

→ ¬ ∧ ∨ ()

G M N B C

| |
|-------|
| G→V |
| ¬N→G |
| ¬B→¬N |
| B→C |
| ¬M→C |

CorregirAjuda

| Activitat | N. Exercici | Estat | Error de Resultat | Error Sintàctic | Data límit |
|-----------|-------------|-----------|-------------------|-----------------|------------|
| 7 | 43 | No Result | 0 | 0 | 7/2/2013 |

Figura 9.20 : Exemple de formalització proposicional amb caràcters no permesos.

I al prémer el botó de corregir ens informa que hi ha caràcters no permesos a la solució introduïda:

La solució enviada és:

$G \rightarrow V$
 $\neg N \rightarrow G$
 $\neg B \rightarrow \neg N$
 $B \rightarrow C$
 $\neg M \rightarrow C$

El resultat de la correcció és:

| | |
|-----------------------------|---|
| G-V | Hi ha caràcters no permesos a la solució introduïda |
| $\neg N \rightarrow G$ | Correcte |
| $\neg B \rightarrow \neg N$ | Correcte |
| B-C | Correcte |
| $\neg M \rightarrow C$ | Correcte |

Continuar

Figura 9.21 : Resultat de caràcters no permesos en formalització de lògica de proposicions.

Al prémer el botó de continuar, tornem a la interfície i comprovem que se'ns ha incrementat el comptador de "Errors Sintàctics", és a dir, hem fet un error de sintaxis.

A més, les línies de la interfície tenen per defecte els valors introduïts en la última solució enviada per l'alumne, per a que aquest en faci les correccions necessàries.

Si ara escrivim la solució correcta, utilitzant els àtoms permesos per l'enunciat:

EXERCICI: 43 de l'activitat Lògica ← →

Formalitzeu les següents proposicions:

Si em gradués en informàtica, aniria a fer un màster a l'estranger.
 Si no surto massa per les nits em graduaré en Informàtica.
 Si no vull treballar en un banc cal que no surti massa.
 Si treballés en un banc acabaria casant-me.
 Si no anés a fer un màster a l'estranger acabaria casant-me.
 (G, M, N, B, C)

→
¬
∧
∨
(
)

G
M
N
B
C

| |
|-----------------------------|
| G→M |
| $\neg N \rightarrow G$ |
| $\neg B \rightarrow \neg N$ |
| B→C |
| $\neg M \rightarrow C$ |

Corregir
Ajuda

| Activitat | N. Exercici | Estat | Errors de Resultat | Errors Sintàctics | Data límit |
|-----------|-------------|-----------|--------------------|-------------------|------------|
| 7 | 43 | No Result | 0 | 1 | 7/2/2013 |

Figura 9.22 : Exemple de formalització de proposicions correcta.

Al corregir ens informa que la solució enviada és correcta:

La solució enviada és:

$G \rightarrow M$
 $\neg N \rightarrow G$
 $\neg B \rightarrow \neg N$
 $B \rightarrow C$
 $\neg M \rightarrow C$

El resultat de la correcció és:

| | | |
|-----------------------------|----------|----------|
| $G \rightarrow M$ | Correcte | Correcte |
| $\neg N \rightarrow G$ | Correcte | |
| $\neg B \rightarrow \neg N$ | Correcte | |
| $B \rightarrow C$ | Correcte | |
| $\neg M \rightarrow C$ | Correcte | |

Continuar

Figura 9.23 : Resultat correcte de formalització de proposicions.

Si tornem a entrar a l'exercici, veurem que se'ns presenta com a resolt, i ens mostra l'última solució enviada:

EXERCICI: 43 de l'activitat Lògica ← →

Formalitzeu les següents proposicions:

Si em gradués en informàtica, aniria a fer un màster a l'estranger.
 Si no surto massa per les nits em graduaré en Informàtica.
 Si no vull treballar en un banc cal que no surti massa.
 Si treballés en un banc acabaria casant-me.
 Si no anés a fer un màster a l'estranger acabaria casant-me.
 (G, M, N, B, C)

Aquesta és l'última solució enviada a l'exercici 43 de l'activitat 7:

| Data | Solució Enviada | Resultat |
|--------------------|-------------------------------|----------|
| 16/4/2012 21:53:36 | 1 $G \rightarrow M$ | Correcte |
| | 2 $\neg N \rightarrow G$ | Correcte |
| | 3 $\neg B \rightarrow \neg N$ | Correcte |
| | 4 $B \rightarrow C$ | Correcte |
| | 5 $\neg M \rightarrow C$ | Correcte |
| | | Correcte |

| Activitat | N. Exercici | Estat | Errors de Resultat | Errors Sintàctics | Data límit |
|-----------|-------------|--------|--------------------|-------------------|------------|
| 7 | 43 | Resolt | 0 | 1 | 7/2/2013 |

← →

Figura 9.24 : Interfície formalització de proposicions quan l'exercici ha estat resolt.

També en aquest tipus d'exercici, quan fem clic al botó de veure les solucions enviades , podem veure l'historial de les solucions enviades, on es desglossa línia per línia si és correcta, incorrecta o té un error sintàctic.



| Data | Solució Enviada | Resultat |
|--------------------|---|-----------------|
| 16/4/2012 21:51:46 | 1 G→V [Hi ha caràcters no permesos a la solució introduïda] | Error Sintàctic |
| | 2 ¬N→G | Correcte |
| | 3 ¬B→¬N | Correcte |
| | 4 B→C | Correcte |
| | 5 ¬M→C | Correcte |
| 16/4/2012 21:53:36 | 1 G→M | Correcte |
| | 2 ¬N→G | Correcte |
| | 3 ¬B→¬N | Correcte |
| | 4 B→C | Correcte |
| | 5 ¬M→C | Correcte |

Tancar

Figura 9.25 : Historial de solucions i correccions en formalització de lògica de proposicions.

9.6 Proves de funcionament de contraexemple de raonament

En aquest tipus d'exercicis, es mostra l'enunciat i l'alumne ha de començar triant quin és el número de referenciats de l'univers en que ha trobat un contraexemple:

EXERCICI: 53 de l'activitat Lògica  

El següent esquema lògic és incorrecte. Troba un contraexemple:

$$\begin{aligned} &\exists x(A(x) \rightarrow \neg B(x)) \\ &\forall x(B(x) \rightarrow \neg C(x)) \\ \Rightarrow &\forall x(C(x) \rightarrow A(x)) \end{aligned}$$

Número de referenciats:

Corregir

| Activitat | N. Exercici | Estat | Error de Resultat | Error Sintàctic | Data límit |
|-----------|-------------|-----------|-------------------|-----------------|------------|
| 7 | 53 | No Resolt | 0 | 0 | 7/2/2013 |







   

Figura 9.26 : Primer pas d'exemple de contraexemple de raonament de predicats.

Si despleguem la llista de referenciats possible, que va de 1 a 5, i per exemple triem 2 referenciats, a la interfície apareixen els àtoms instanciats per als 2 referenciats, anomenats "a" i "b":

EXERCICI: 53 de l'activitat Lògica  

El següent esquema lògic és incorrecte. Troba un contraexemple:

$$\begin{aligned} &\exists x(A(x) \rightarrow \neg B(x)) \\ &\forall x(B(x) \rightarrow \neg C(x)) \\ \Rightarrow &\forall x(C(x) \rightarrow A(x)) \end{aligned}$$

Número de referenciats:

A[a]= A[b]=

B[a]= B[b]=

C[a]= C[b]=

Corregir

| Activitat | N. Exercici | Estat | Error de Resultat | Error Sintàctic | Data límit |
|-----------|-------------|-----------|-------------------|-----------------|------------|
| 7 | 53 | No Resolt | 0 | 0 | 7/2/2013 |





   

Figura 9.27 : Segon pas d'exemple de contraexemple de raonament de predicats.

En aquest moment l'alumne ha d'anar desplegant cada llista i triar si aquell àtom per aquell referenciant és *cert* o *fals*.

Si introduïm una solució incorrecta, per exemple:

EXERCICI: 53 de l'activitat Lògica ← →

El següent esquema lògic és incorrecte. Troba un contraexemple:

$$\begin{aligned} &\exists x(A(x) \rightarrow \neg B(x)) \\ &\forall x(B(x) \rightarrow \neg C(x)) \\ \Rightarrow &\forall x(C(x) \rightarrow A(x)) \end{aligned}$$

Número de referenciats:

A[a]= A[b]=

B[a]= B[b]=

C[a]= C[b]=

Corregir

| Activitat | N. Exercici | Estat | Errors de Resultat | Errors Sintàctics | Data límit |
|-----------|-------------|-----------|--------------------|-------------------|------------|
| 7 | 53 | No Result | 0 | 0 | 7/2/2013 |

← →

Figura 9.28 : Exemple de contraexemple incorrecte de raonament de predicats.

Quan premem el botó de Corregir, efectivament ens donarà un Incorrecte.

La solució enviada és:

A[a]=Cert,
 A[b]=Cert,
 B[a]=Fals,
 B[b]=Cert,
 C[a]=Cert,
 C[b]=Fals

El resultat de la correcció és:

Incorrecte

Continuar

Figura 9.29 : Resultat incorrecte de contraexemple de raonament de predicats.

Si tornem a la finestra de la interfície de l'exercici, amb el botó de "Continuar", comprovem que se'ns ha incrementat el comptador de "Errors de Resultat" i que tenim com a valors per defecte els de l'última solució enviada.

Si corregim aquesta solució i escrivim un contraexemple correcte com és el següent:

EXERCICI: 53 de l'activitat Lògica ← →

El següent esquema lògic és incorrecte. Troba un contraexemple:

$$\begin{aligned} &\exists x(A(x) \rightarrow \neg B(x)) \\ &\forall x(B(x) \rightarrow \neg C(x)) \\ \Rightarrow &\forall x(C(x) \rightarrow A(x)) \end{aligned}$$

Número de referenciats:

A[a]=

B[a]=

C[a]=

Corregir

| Activitat | N. Exercici | Estat | Errors de Resultat | Errors Sintàctics | Data límit |
|-----------|-------------|-----------|--------------------|-------------------|------------|
| 7 | 53 | No Result | 1 | 0 | 7/2/2013 |

← →

Figura 9.30 : Exemple de contraexemple de raonament de predicats correcte.

Al corregir, en aquest cas ens indica que sí que és correcte:

La solució enviada és:

A[a]=Fals,
B[a]=Fals,
C[a]=Cert

El resultat de la correcció és:

Correcte

Continuar

Figura 9.31 : Resultat correcte de contraexemple de raonament de predicats.



Altre cop, en aquest tipus d'exercici podem veure en tot moment l'historial de solucions enviades amb el botó corresponent, que en aquest cas només podran ser correctes o incorrectes:

| Data | Solució Enviada | Resultat |
|--------------------|---|------------|
| 16/4/2012 22:52:57 | A[a]=Cert, A[b]=Cert, B[a]=Fals, B[b]=Cert, C[a]=Cert, C[b]=Fals | Incorrecte |
| 16/4/2012 22:54:53 | A[a]=Fals, B[a]=Fals, C[a]=Cert | Correcte |

Tancar

Figura 9.32 : Historial de solucions i correccions en exercicis de contraexemple.

9.7 Proves de funcionament amb solucions buides

En totes tres tipologies d'exercici, si no s'omplen tots els camps proporcionats, no s'accepta la solució de l'alumne i no es compta com a error sintàctic ni de resultat.

No és necessari mostrar amb cap missatge de text que ha d'omplir la solució, i amb un simple canvi de color de la casella deixada en blanc, l'usuari adverteix que no ha introduït cap solució en aquella casella.

The interface shows a set of logic symbols: \exists , \forall , \rightarrow , \neg , \wedge , \vee , $($, $)$, F , A , S , and x . Below the symbols is a red horizontal bar indicating an empty solution field. At the bottom are two buttons: "Corregir" and "Ajuda".

Figura 9.33 : Exercici de formalització d'enunciats amb predicats amb camp buit.

The interface shows logic symbols: \rightarrow , \neg , \wedge , \vee , $($, $)$, and variables G , M , N , B , C . Below the symbols are four input fields containing the formulas: $G \rightarrow M$, $\neg N \rightarrow G$, $B \rightarrow C$, and $\neg M \rightarrow C$. A red horizontal bar is present between the second and third fields. At the bottom are two buttons: "Corregir" and "Ajuda".

Figura 9.34 : Exercici de formalització de proposicions amb camp buit.

The interface shows a label "Número de referenciats:" followed by a dropdown menu with a red background and three dots. Below is a "Corregir" button.

Figura 9.35 : Exercici de contraexemple amb número de referenciats buit.

The interface shows a label "Número de referenciats:" followed by a dropdown menu with the value "3". Below are two rows of dropdown menus for variable instances: $P[a]$, $P[b]$, $P[c]$ and $Q[a]$, $Q[b]$, $Q[c]$. The values are: $P[a]$ =Cert, $P[b]$ =..., $P[c]$ =Fals, $Q[a]$ =Fals, $Q[b]$ =Cert, $Q[c]$ =Cert. A "Corregir" button is at the bottom.

Figura 9.36 : Exercici de contraexemple amb instàncies buides.

9.8 Proves de funcionament com a professor

Quan el professor entra a l'assignatura de lògica, pot veure l'estat de cadascun dels exercicis per a cada alumne: si ha estat resolt o no, si té errors de resultat o sintàctics, i si ha visualitzat l'exercici.

Aquesta informació la gestiona la plataforma ACME, que guarda a la base de dades les solucions i l'estat dels exercicis dels alumnes.

| | | | | | |
|------------------|----|-----------|---|----|----|
| ☑ Torm, Gemma | 31 | No Resolt | 0 | 0 | 12 |
| | 32 | No Resolt | 0 | 0 | 11 |
| | 33 | Resolt | 0 | 2 | 17 |
| | 34 | No Resolt | 1 | 3 | 20 |
| | 35 | No Resolt | 0 | 1 | 11 |
| | 36 | No Resolt | 0 | 1 | 6 |
| | 37 | Resolt | 2 | 0 | 16 |
| | 41 | No Resolt | 0 | 0 | 12 |
| | 42 | No Resolt | 0 | 18 | 32 |
| | 43 | Resolt | 3 | 30 | 45 |
| | 51 | Resolt | 1 | 0 | 5 |
| | 52 | Resolt | 1 | 0 | 4 |
| | 53 | Resolt | 1 | 0 | 7 |

Figura 9.37 : Professor fent el seguiment dels exercicis realitzats per un alumne.

Fent clic al número d'exercici a consultar per un alumne donat, el professor veurà les diferents solucions enviades per l'alumne. A més a més es carregarà la interfície per enviar solucions que tindrà com a valors per defecte els de la última solució enviada per l'alumne.

Val a dir que si el professor prova una solució en aquesta interfície, i l'envia a corregir, el corrector li dirà si la solució enviada és correcta o incorrecta, o si té algun error sintàctic, si s'escau, però no es guardarà aquesta solució a la base de dades sota el perfil d'alumne ni de professor, i tampoc es mostrarà com a valor per defecte la propera vegada que el professor visualitzi l'exercici.

EXERCICI: 52 de l'activitat Lògica

El següent esquema lògic és incorrecte. Troba un contraexemple:

$$\begin{aligned} & \exists x(P(x) \vee R(x)) \\ & \exists x(Q(x) \vee R(x)) \\ & \forall x(P(x) \wedge Q(x) \rightarrow R(x)) \\ & \Rightarrow \exists xR(x) \end{aligned}$$

| Data | Solució Enviada | Resultat |
|-------------------|---|------------|
| 7/4/2012 13:10:35 | P[a]=Cert, P[b]=Fals, Q[a]=Fals, Q[b]=Cert, R[a]=Cert, R[b]=Cert | Incorrecte |
| 7/4/2012 13:15:44 | P[a]=Cert, P[b]=Fals, Q[a]=Fals, Q[b]=Cert, R[a]=Fals, R[b]=Fals | Correcte |

Número de referenciats: 2 ▾

P[a]= Cert ▾ P[b]= Fals ▾
Q[a]= Fals ▾ Q[b]= Cert ▾
R[a]= Fals ▾ R[b]= Fals ▾

Corregir

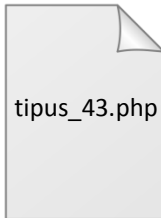
ALUMNE: GEMMA TORM



Figura 9.38 : Interfície per al professor de l'exercici d'un alumne.

10. Implantació

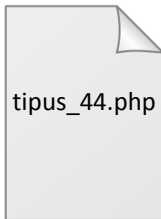
Per implantar un nou tipus d'exercici, hem de crear els següents fitxers a la carpeta `/home/projei/include/models` del servidor



tipus_43.php

Nou tipus d'exercici per resoldre els exercicis de formalització de lògica de predicats.

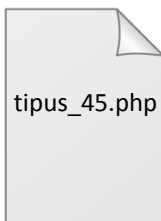
Cadena que l'identifica: LOGICA_FORM



tipus_44.php

Nou tipus d'exercici per resoldre els exercicis de formalització de lògica de proposicions.

Cadena que l'identifica: LOGICA_FORM_PROP



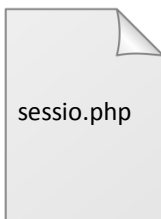
tipus_45.php

Nou tipus d'exercici per resoldre els exercicis de cerca de contraexemple.

Cadena que l'identifica: LOGICA_FORM_CONTRA

Aquestes cadenes identificatives s'han de utilitzar per substituir la constant `MATES_MISSATGE` a les funcions `Sorteig_Enunciats` i `Verificar_Problema`.

A més, s'han de donar d'alta al fitxer de definicions, que es troba a `/home/projei/include/moduls/sessio.php`



sessio.php

```
define("LOGICA_FORM","43");
```

Numero que identifica tipus problema de formalització de predicats

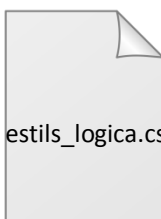
```
define("LOGICA_FORM_PROP","44");
```

Numero que identifica tipus problema de formalització de proposicions

```
define("LOGICA_FORM_CONTRA","45");
```

Numero que identifica tipus problema de contraexemple

També pugem al servidor el full d'estils, a `/home/projei/www/estils/estils_logica.css`

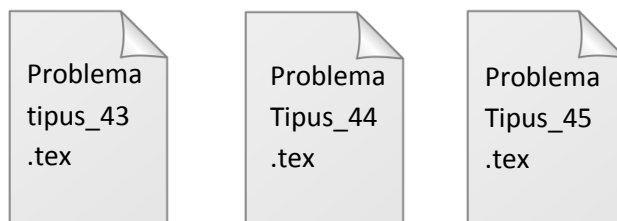


estils_logica.css

Ara aquests tres tipus d'exercicis ja estan implantats al servidor i a punt per a que el professor pugui crear exercicis i sortejar-los entre els alumnes.

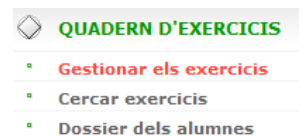
Preparem els fitxers amb els enunciats (i per als de formalització amb les solucions) per carregar-los a la plataforma ACME.

Els exercicis creats per al desenvolupament del projecte els ha fet l'alumna autora d'aquest projecte, i amb les ajudes proporcionades els professors de l'assignatura de lògica podran crear tants exercicis d'aquests tipus com creguin oportuns.



Per donar d'alta un exercici, el professor s'identifica a la plataforma ACME i realitza les següents passes:

Es va a "Gestionar els exercicis" de la secció del "Quadern d'exercicis". Triem "Afegir un exercici" i seleccionem la temàtica "Lògica", i a continuació omplim el formulari amb el títol de l'exercici, una explicació i el fitxer .tex de l'exercici que estem creant, que en el nostre cas pot ser el Problema_tipus_43.tex , Problema_tipus_44.tex o el Problema_tipus_45.tex, i el nivell de l'exercici. L'exercici en aquest moment queda pendent de validar per un administrador de la plataforma ACME.



1. Un administrador de la plataforma ACME valida el fitxer LaTeX del nou tipus d'exercici des de l'apartat de "Validar un exercici" dins de "Gestió exercicis" a la secció "Quadern".
2. Un cop validat per l'administrador, el professor podrà seleccionar l'exercici per sortejar, cercant l'exercici per la temàtica de lògica dins de "Cercar exercicis" a la secció de "Quadern d'exercicis".

Un cop trobat l'exercici, l'afegeix a la cistella amb el botó .

3. El professor ja està en situació de sortejar l'exercici entre els alumnes. Amb tal fi, el professor va a l'apartat de "Dossier dels alumnes" de la secció de "Quadern d'exercicis".

Fent clic a “Assignar un exercici” ja tindrem a punt l’exercici que ha afegit a la cistella, tria el número que li vol donar i inicia el sorteig.

Un mateix exercici es pot sortejar diverses vegades, però ha de tindre en compte que com a molt podrà generar 2/3 exercicis sobre el número d’enunciats diferents a dins del fitxer LaTeX. Aquesta restricció hi és per a que la plataforma ACME pugui assegurar la suficient diversitat d’exercicis i no bloquegi el procés de sorteig.

11. Conclusions

S'han assolit els objectius del Projecte Final de Carrera, integrant els exercicis que hem cregut més interessants de l'assignatura de Lògica a la plataforma ACME.

Els tres tipus d'exercicis implementats han estat:

- Formalització d'enunciats de predicats, que poden incloure varies variables i constants.
- Formalització d'enunciats amb proposicions.
- Contraexemple d'un enunciat de predicats.

El mòdul de Lògica que s'ha realitzat, podrà ser ampliat en el futur per incloure la resta de tipus d'exercicis de l'assignatura de Lògica, com per exemple exercicis d'interpretació d'un raonament formalitzat a llenguatge natural, tot i segurament s'hauran d'emprar eines diferents a les utilitzades en els exercicis aquí implementats.

La realització d'aquest Projecte Final de Carrera m'ha permès fer totes les fases d'un projecte d'enginyeria informàtica, començant pels estudis i planificació, passant per l'anàlisi i el disseny, i acabant amb la implementació, proves i implantació. És en projectes d'aquestes característiques on s'experimenta de primera mà la utilitat de realitzar aquestes etapes que s'havien après als estudis.

També ha estat enriquidor haver de prendre decisions per la bona execució del projecte i per a l'optimització del temps de que disposava.

Treballar en una plataforma d'e-learning com l'ACME, amb una arquitectura modular i robusta, m'ha permès veure de prop com s'ha implementat un sistema de les seves dimensions.

Un dels punts forts d'aquesta plataforma és proporcionar a l'alumne el resultat de la correcció de manera immediata, pel que s'ha donat molta importància a optimitzar els nuclis correctors dels exercicis implementats.

A més, s'han generat unes interfícies usables i amigables per a l'usuari, per a que l'alumne pugui realitzar els exercicis sense familiarització prèvia amb les pantalles.

He pogut aplicar els coneixements adquirits en les assignatures de *Iniciació a la Programació* i *Estructura de Dades i Algorismes* per a la programació dels mètodes associats a la visualització de les dades a la interfície i el processament de les formalitzacions de predicats per traduir-les a codi Mathematica que pugui ser avaluat.

Els coneixements adquirits a l'assignatura d'*Introducció a la Lògica* han estat molt útils per poder realitzar el nucli corrector i les proves d'execució. I l'assignatura d'*Enginyeria del Software* ha estat molt important per a la planificació i gestió del projecte.

He adquirit coneixements de jQuery per a la realització de les interfícies d'entrada de dades i de Mathematica per a la creació dels nuclis correctors, que ha estat la part del Projecte Final de Carrera que he gaudit més.

També he aprofundit en els meus coneixements previs de PHP per a la integració dels diferents elements a la plataforma ACME.

12. Treball futur

Com hem vist en aquest Projecte Final de Carrera s'han realitzat, d'entre la varietat d'exercicis de l'assignatura de Lògica, els que es van considerar més prioritaris.

Ampliacions a aquest treball podrien ser:

- Ampliar els tipus de problema de formalització de raonaments proposicionals amb el bicondicional: $A \Leftrightarrow B$ (A si i només si B).
- Ampliar el tipus de problema de formalització de raonaments amb predicats amb funcions.
- Ampliar el tipus de problema que valida el contraexemple d'un raonament lògic per a que inclogui funcions i múltiples variables.
- Ampliar el número d'elements del domini en l'eliminació dels quantificadors en els exercicis de formalització de lògica de predicats.

També es podria ampliar la varietat d'exercicis de l'assignatura de Lògica implementats a la plataforma ACME:

- Exercicis d'interpretació d'un raonament formalitzat a llenguatge natural.
- Exercicis de demostració de la validesa d'un raonament pas a pas.
- Exercicis relacionats amb conjunts.

13. Agraïments

Per a la consecució d'aquest Projecte Final de Carrera ha estat clau el suport ofert pel tutor del projecte, en Ferran Prados, tant en coneixements tècnics de la plataforma ACME com en la gestió del projecte. Amb gratitud puc dir que he tornat a gaudir aprenent i treballant d'enginyera informàtica.

També agrair a tots els professors que he tingut al llarg de la carrera, i especialment al grup *Broadband Communications and Distributed Systems (BCDS)*, de la Universitat de Girona per l'oportunitat de començar a aplicar els coneixements teòrics al món pràctic durant els meus estudis, fet que em va portar a la meva especialització professional.

A l'*Associació d'Estudiants d'Informàtica de Girona (AEIGi)*, on uns bojos pels ordinadors molt simpàtics i emprenedors em van despertar les ganes d'aprendre.

I als amics, pels records que encara et fan somriure després de tants anys. Gràcies especialment a l'Anna, l'Albert, en Nacho, en Johny i en Xevi per la vostra amistat.

I com no, als pares i la meva germana, que m'ha estat animant per a que acabés els estudis, i a la meva parella, la meva alegria.

Gràcies a tots per acompanyar-me en aquest camí.

14. Bibliografia

Com fer un Projecte Final de Carrera

- [1] Biblioteca UdG.
Com citar documents. Estil APA.
Recuperat 26 març 2012, des de <http://www.udg.edu/LaBibliotecaforma/Comcitardocuments/EstilAPAencatala/tabid/11972/language/ca-ES/Default.aspx>

Projecte ACME

- [1] Projecte ACME. Departament d'Informàtica i Matemàtica Aplicada - Escola Politècnica Superior - Universitat de Girona.
Servidor acme4.udg.edu.
Recuperat 16 abril 2012, des de <http://acme4.udg.edu>
- [2] Poch Garcia, J., Prados Carrasco, F. [et al.] (2008).
L'ACME una eina per a la planificació, seguiment i avaluació dels aprenentatges.
Recuperat 14 novembre 2011, des de <http://dugi-doc.udg.edu/bitstream/10256/1023/1/193.pdf>
- [3] Prados Carrasco, F., Poch Garcia, J. [et al.] (2008).
L'ACME, el dossier d'activitats de l'alumne.
Recuperat 14 novembre 2011, des de <http://acme.udg.edu/articles/cidui2008-2.pdf>

Assignatura de lògica a la UdG

- [1] Humet, J.
Assistent per la validació de raonaments.
Recuperat 15 novembre 2011, des de <http://ima.udg.edu/~humet/logicweb/logicaudg/login.php>
- [2] Humet, J. (2000)
LSD: un sistema de verificació per Lògica Simbòlica Deductiva de predicats.
Recuperat 15 novembre 2011, des de <http://ima.udg.edu/~humet/lzd/>
- [3] Humet, J. (1995)
Apunts i transparències de l'assignatura de Lògica.
Recuperat 16 novembre 2011

Programació Extrema

- [1] Anònim.
Programació extrema.
Recuperat 21 novembre 2011, des de <http://eisc.univalle.edu.co/materias/WWW/material/lecturas/xp.pdf>

- [2] Letelier, P i Penadés, C.
Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).
Recuperat 21 novembre 2011, des de <http://www.willydev.net/descargas/masyxp.pdf>
- [3] Beck, K., Grenning J. [et al.] (2001)
Manifesto for Agile Software Development.
Recuperat 21 novembre 2011, des de <http://www.agilemanifesto.org/>
- [4] Villafuerte, V. (2009)
Extreme Programming.
Recuperat 21 novembre 2011, des de <http://extremeprogramming.host56.com/>
- [5] Dev Zombie Staff.
What is XP: Extreme Programming?
Recuperat 21 novembre 2011, des de <http://www.devzombie.com/pages/xp-extreme-programming>
- [6] Meier, J. D. (2010)
The Four Circles of XP (Extreme Programming)
Recuperat 21 novembre 2011, des de <http://blogs.msdn.com/b/jmeier/archive/2010/04/04/the-four-circles-of-xp-extreme-programming.aspx>

Rich Internet Applications

- [1] Ward, J. (2007)
What is a Rich Internet Application?
Recuperat 5 decembre 2011, des de <http://www.jamesward.com/2007/10/17/what-is-a-rich-internet-application/>
- [2] Haureal Web Studio.
¿Qué son las Rich Internet Applications?
Recuperat 5 decembre 2011, des de <http://haureal.es/innovacion/rich-internet-applications/>
- [3] Rowell, E. (2010)
Building Rich Internet Applications with Frameworks.
Recuperat 5 decembre 2011, des de <http://sixrevisions.com/web-development/building-rich-internet-applications-with-frameworks/>
- [4] Wikipedia. (2011)
Rich Internet application.
Recuperat 5 decembre 2011, des de http://en.wikipedia.org/wiki/Rich_Internet_application
- [5] Aaron Newton. (2009)
jQuery vs MooTools.
Recuperat 5 decembre 2011, des de <http://jqueryvsmootools.com/>

- [6] Miguel Ángel Álvarez. (2002)
Qué son los Applets de Java.
Recuperat 25 novembre 2011, des de <http://www.desarrolloweb.com/articulos/731.php>
- [7] Ethan Heilman [et al.]. (2009)
Java applets vs Silverlight vs Flash (Flex)
Recuperat 25 novembre 2011, des de <http://stackoverflow.com/questions/816428/java-applets-vs-silverlight-vs-flash-flex>

Interfície

- [1] PHP.
Recuperat 25 novembre 2011, des de <http://www.php.net/>
- [2] jQuery
jQuery: The Write Less, Do more.
Recuperat 8 decembre 2011, des de <http://jquery.com/>
- [3] Liñán, J. (2009)
jQuery Hotkeys, controla el teclado con jQuery.
Recuperat 6 febrer 2012, des de <http://www.ribosomatic.com/articulos/jquery-hotkeys-controla-el-teclado-con-jquery/>
- [4] Binny, V. A. (2008)
js-hotkeys. Javascript jQuery for hooking keyboard events.
Recuperat 6 febrer 2012, des de <http://code.google.com/p/js-hotkeys/>

Estudi nucli corrector

- [1] Wolfram.
Wolfram Mathematica 8 – Documentation Center.
Recuperat 23 novembre 2011, des de <http://reference.wolfram.com/mathematica/guide/Mathematica.html>
- [2] Detinko, A.
Prolog for Windows. Manual for MA410 labs.
Recuperat 28 novembre 2011, des de http://www.maths.nuigalway.ie/~detinko/prolog_manual.pdf
- [3] Swi prolog
SWI-Prolog's home.
Recuperat 22 novembre 2011, des de <http://www.swi-prolog.org/>
- [4] Torres de Alba, R. (2006)
Minimanualillo de SWI-Prolog.
Recuperat 28 novembre 2011, des de <http://www.yakiboo.net/apuntes/minimanualillo.pdf>

Algorisme Shunting Yard

- [1] Wikipedia
Algoritmo Shunting yard
Recuperat 25 gener 2012, des de http://es.wikipedia.org/wiki/Algoritmo_shunting_yard

- [2] Vaca, C. (2010)
Traducción de una expresión en un árbol binario.
Recuperat 25 gener 2012, des de <http://www.infor.uva.es/~cvaca/asigs/AlgInfPost.htm>

- [3] Forum Stackoverflow (2010)
How can I modify my Shunting-Yard Algorithm so it accepts unary operators?
Recuperat 25 gener 2012, des de
<http://www.cs.buap.mx/~andrex/AlgoritmoPolacasPosfijo.pdf>

- [4] Robin (2008)
Extension to the 'Shunting Yard' algorithm to allow variable numbers of arguments to functions
Recuperat 25 gener 2012, des de <http://www.kallisti.net.nz/blog/2008/02/extension-to-the-shunting-yard-algorithm-to-allow-variable-numbers-of-arguments-to-functions/>

- [5] Forum (2010)
Intérprete de funciones mediante un árbol binario de expresiones.
Recuperat 25 gener 2012, des de <http://www.esdebian.org/wiki/interpretprete-funciones-mediante-arbol-binario-expresiones>

15. Annexos

Índex de taules

| | | |
|-----------|---|----|
| Taula 1.1 | Utilització de l'ACME en xifres | 5 |
| Taula 4.1 | Planificació del Projecte Final de Carrera | 19 |
| Taula 9.1 | Codificació LaTeX dels caràcters especials i símbols lògics | 45 |
| Taula 9.2 | Codificació pròpia per als símbols lògics..... | 46 |
| Taula 9.3 | Constants i Funcions de Mathematica d'una lletra | 46 |
| Taula 9.4 | Estudi del temps d'execució necessari per l'avaluació de l'equivalència entre dos predicats d'una variable. | 53 |
| Taula 9.5 | Estudi del temps d'execució necessari per l'avaluació de l'equivalència entre dos predicats de dos variables. | 54 |

Índex de figures

| | | |
|------------|---|----|
| Figura 1.1 | Logotip del projecte ACME | 3 |
| Figura 1.2 | Exercici de programació on l'alumne puja un fitxer | 4 |
| Figura 3.1 | Els pilars de la Programació Extrema..... | 12 |
| Figura 3.2 | Processos en la Programació Extrema..... | 13 |
| Figura 3.3 | Pràctiques de la Programació Extrema..... | 15 |
| Figura 8.1 | Arquitectura client/servidor de la plataforma ACME | 33 |
| Figura 8.2 | Flux de treball client/servidor | 34 |
| Figura 8.3 | Diagrama d'activitats dels exercicis de formalització de lògica de predicats..... | 40 |
| Figura 8.4 | Diagrama d'activitats dels exercicis de formalització de lògica de proposicions. | 42 |
| Figura 8.5 | Diagrama d'activitats dels exercicis de contraexemples de lògica de predicats..... | 44 |
| Figura 9.1 | Mètodes relacionats amb la interfície a la plataforma ACME..... | 51 |
| Figura 9.2 | Algoritme Shunting Yard | 55 |
| Figura 9.3 | Interfície d'entrada de la solució dels exercicis de formalització de lògica de predicats..... | 56 |

| | | |
|-------------|--|----|
| Figura 9.4 | Interfície amb Ajuda indicant les dreceres de teclat. | 57 |
| Figura 9.5 | Exemple de formalització de predicats incorrecta. | 57 |
| Figura 9.6 | Resultat incorrecte en formalització de predicats..... | 58 |
| Figura 9.7 | Exemple de formalització de predicats correcta. | 58 |
| Figura 9.8 | Resultat correcte de formalització de predicats. | 59 |
| Figura 9.9 | Exemple de formalització de predicats equivalent..... | 59 |
| Figura 9.10 | Resultat correcte de formalització de predicats equivalent. | 60 |
| Figura 9.11 | Interfície formalització de predicats quan l'exercici ha estat resolt..... | 60 |
| Figura 9.12 | Exemple de formalització de predicats amb parèntesis de més. | 61 |
| Figura 9.13 | Resultat d'error sintàctic de parèntesis en formalització de lògica de predicats..... | 61 |
| Figura 9.14 | Exemple de formalització de predicats amb parèntesis corregits..... | 62 |
| Figura 9.15 | Resultat correcte de formalització de predicats corregida. | 62 |
| Figura 9.16 | Exemple de formalització de predicats amb caràcters no permesos. | 63 |
| Figura 9.17 | Resultat de caràcters no permesos en formalització de lògica de predicats. | 63 |
| Figura 9.18 | Interfície formalització de predicats quan l'exercici ha estat resolt després de fer errors sintàctics. | 64 |
| Figura 9.19 | Historial de solucions i correccions en lògica de predicats. | 64 |
| Figura 9.20 | Exemple de formalització proposicional amb caràcters no permesos. | 65 |
| Figura 9.21 | Resultat de caràcters no permesos en formalització de lògica de proposicions. | 66 |
| Figura 9.22 | Exemple de formalització de proposicions correcta..... | 66 |
| Figura 9.23 | Resultat correcte de formalització de proposicions. | 67 |
| Figura 9.24 | Interfície formalització de proposicions quan l'exercici ha estat resolt. | 67 |
| Figura 9.25 | Historial de solucions i correccions en formalització de lògica de proposicions. | 68 |
| Figura 9.26 | Primer pas d'exemple de contraexemple de raonament de predicats. | 69 |
| Figura 9.27 | Segon pas d'exemple de contraexemple de raonament de predicats. | 69 |
| Figura 9.28 | Exemple de contraexemple incorrecte de raonament de predicats. | 70 |
| Figura 9.29 | Resultat incorrecte de contraexemple de raonament de predicats..... | 70 |
| Figura 9.30 | Exemple de contraexemple de raonament de predicats correcte. | 71 |
| Figura 9.31 | Resultat correcte de contraexemple de raonament de predicats. | 71 |

| | | |
|-------------|---|----|
| Figura 9.32 | Historial de solucions i correccions en exercicis de contraexemple..... | 72 |
| Figura 9.33 | Exercici de formalització d'enunciats amb predicats amb camp buit. | 73 |
| Figura 9.34 | Exercici de formalització de proposicions amb camp buit..... | 73 |
| Figura 9.35 | Exercici de contraexemple amb número de referenciants buit. | 73 |
| Figura 9.36 | Exercici de contraexemple amb instàncies buides. | 73 |
| Figura 9.37 | Professor fent el seguiment dels exercicis realitzats per un alumne. | 74 |
| Figura 9.38 | Interfície per al professor de l'exercici d'un alumne. | 75 |

16. Manuals d'usuari

S'ha realitzat, per a cada tipus d'exercici implementat, un manual d'usuari per al professor, per tal d'ajudar-lo en la creació de fitxers d'enunciats de nous exercicis, i un manual per als alumnes explicant el funcionament de les interfícies d'entrada de dades.

16.1 Ajuda per al professor dels exercicis de Formalització de lògica de predicats

Un **problema** de formalització de lògica de predicats està format per una sèrie d'**exercicis**. Cada alumne haurà de resoldre només un dels exercicis entre els que es preparin en aquest problema, que serà assignat de manera aleatòria pel sorteig que fa la plataforma ACME.

Un exemple de problema amb tres exercicis seria per exemple:

Formalitza l'enunciat següent:

- Les pomes i les taronges són gustoses i nutritives. $(P(x), T(x), G(x), V(x))$.
- Hi ha aliments que només es poden menjar si són cuits. $(A(x), M(x), C(x))$.
- Sense frens no hi ha cap auto segur. $(F(x), A(x), S(x))$.

Mentre que cada alumne només veuria un exercici, per exemple:

Formalitza l'enunciat següent:

Hi ha aliments que només es poden menjar si són cuits. $(A(x), M(x), C(x))$.

Es poden generar varis problemes del mateix tipus, la formalització de lògica de predicats, on s'hagin agrupat els enunciats per nivell de dificultat. D'aquesta manera, cada alumne haurà de fer un enunciat de cada problema que es generi.

Cada problema s'implementarà en un fitxer, utilitzant una sintaxis molt concreta que s'indica a continuació.

Implementació dels problemes d'avaluació continuada

El fitxer que contingui la implementació del problema ha de ser un fitxer en format LaTeX. El fitxer estarà escrit utilitzant la següent sintaxis.

Capçalera del fitxer

Al principi del fitxer hi ha d'haver una petita capçalera que identifica de quin tipus d'exercici es tracta i de quin nivell és el problema que anem a verificar. En aquest cas, el número del tipus d'exercici associat a la formalització de lògica de predicats ha de ser el número **43**. I "Problema 1.01---ACME" indica que es tracta d'un problema de nivell 1, és a dir, d'avaluació continuada.

```
43
\begin{center}
\large {\bf Problema 1.01---ACME}
\end{center}
```

Enunciat del problema

Un problema tindrà un enunciat comú a tots els exercicis del problema. Els enunciats es definiran de la següent manera:

```
\underline{Enunciats:}\par \vspace{0.5cm}
```

```
* % <E>
```

```
Formalitza l'enunciat següent:
```

```
\tt
```

```
\hspace{0.5 cm}
```

```
% <1>
```

```
{\bf P1}
```

```
% </E>
```

```
\par
```

```
\vspace{1cm}
```

L'identificador de l'enunciat és '% <E>' i cal respectar els salts de línia, les majúscules i els espais en blanc (entre '%' i '<E>' o '<1>' hi ha un espai en blanc).

En aquest tipus de problemes, no hi ha paràmetres a l'enunciat, pel que només hem de tenir '% <1>'. És molt important fer un canvi de línia després d'aquest identificador i escriure '{\bf P1}' i tornar a fer un canvi de línia. Aquesta línia no es tindrà en compte a l'hora de generar els enunciats, només és una manera per fer estètic el problema per si es vol visualitzar tot el fitxer LaTeX del problema, però si no hi és, no s'escriu la línia que va a continuació.

Definició dels exercicis com a paràmetre del problema

Els diferents exercicis del problema els escriurem tots com un únic paràmetre del problema.

Per indicar que comencen els paràmetres s'ha d'escriure '% <P>' (igual que amb '% <E>' hi ha un espai entre '%' i '<P>'). Per indicar que els paràmetres han acabat es fa servir l'identificador '% </P>'. A més a més, per indicar que comença el primer paràmetre, que en aquest tipus de problema és l'únic, es fa servir l'identificador '% <1>' seguit d'un salt de línia. I per indicar que acaba el primer paràmetre s'utilitza '% </1>'.

Entre aquests dos identificadors, '% <1>' i '% </1>', s'han d'escriure els diferents exercicis, que en el següent quadre ha estat indicat amb els punt suspensius.

```
\underline{Par`ametes:}\par \vspace{0.5cm}
```

```
% <P>
```

```
{\bf P1} \par % <1>
```

```
...
```

```
% </1>
```

```
% </P>
```

Cada exercici ha d'estar en una línia del fitxer LaTeX.

Les expressions dels exercicis estaran formades per dues parts ben diferenciades:

- L'enunciat d'aquell exercici concret.
Escrit en LaTeX, generarà el enunciat particular que visualitzarà l'alumne
- La solució d'aquell exercici concret.
Escrit en una codificació pròpia, l'utilitzarà el nucli corrector del problema per corregir la solució de l'alumne

Aquestes expressions tenen un '\$' al inici i al final, '\$\#' per diferenciar el enunciat de la solució, i al final de tot de la línia un '. \par' per indicar la finalització de la línia:

```
$ enunciat $\#$solució $. \par
```

La solució està formada per quatre parts:

1. Solució, en un codificació pròpia
2. Els àtoms
3. Les variables, que han de ser minúscules
4. Les constants, que han de ser majúscules

Cadascuna d'aquestes parts es separa amb un ',', pel que la estructura serà la següent:

```
SolucióCodificada %Àtoms %variables %Constants
```

Si per exemple volem fer un problema amb 5 exercicis diferents, la definició dels exercicis tindrà la següent forma:

```
\underline{Par\`ametres:}\par \vspace{0.5cm}  
  
% <P>  
{\bf P1} \par % <1>  
$ enunciat1 $\#$ SolucióCodificada1 %Àtoms1 %variables1 %Constants1 $. \par  
$ enunciat2 $\#$ SolucióCodificada2 %Àtoms2 %variables2 %Constants2 $. \par  
$ enunciat3 $\#$ SolucióCodificada3 %Àtoms3 %variables3 %Constants3 $. \par  
$ enunciat4 $\#$ SolucióCodificada4 %Àtoms4 %variables4 %Constants4 $. \par  
$ enunciat5 $\#$ SolucióCodificada5 %Àtoms5 %variables5 %Constants5 $. \par  
% </1>  
% </P>
```

Definició de cada enunciat específic

Per escriure la part d'enunciat, ho hem de fer en LaTeX. Si volem que l'enunciat específic que veurà l'alumne tingui salts de línia, al fitxer LaTeX ho indicarem amb '\\'.

Per exemple, per a que l'enunciat particular que veu l'alumne sigui:

Per Nadal els estudiants tenen vacances i només aquests.
 $(X, S(x), V(x))$

al fitxer LaTeX s'haurà d'escriure:

Per Nadal els estudiants tenen vacances i només aquests. $\backslash(X, S(x), V(x))$

També és important fixar-se que en aquest exemple s'esperaria que l'àtom 'els estudiants' es representés amb 'E(x)' i que s'ha fet amb 'S(x)'. Això es deu a que el llenguatge que utilitza el nucli corrector, el Mathematica, té unes lletres reservades que no podem utilitzar com a àtoms, i que són les següents:

D E I N O

Tampoc s'ha pogut utilitzar la lletra 'N' per a la constant 'per Nadal' i s'ha optat per la lletra 'X'.

Podrem per tant utilitzar per representar àtoms i constants les lletres de la 'A' a la 'Z' en majúscula i amb excepció de 'D', 'E', 'I', 'N', 'O'.

Definició de cada solució específica

Hem vist que cada solució té la forma:

SolucióCodificada %Àtoms %variables %Constants

Anem a veure com escriurem cadascuna d'aquestes parts:

Els *àtoms* hauran de ser lletres majúscules de la 'A' a la 'Z', excepte 'D', 'E', 'I', 'N', 'O'.

Les *variables* hauran de ser lletres minúscules de la 'a' a la 'z', excepte 'd', 'e', 'i', 'n', 'o'.

Les *constants* hauran de ser lletres majúscules de la 'A' a la 'Z', excepte 'D', 'E', 'I', 'N', 'O'.

A la *SolucióCodificada* s'ha de substituir els connectors lògics i quantificadors propis d'una formalització en lògica de predicats per una codificació pròpia utilitzada en aquest tipus d'exercicis:

| Símbol lògic | Codificació |
|----------------------------|-------------|
| \forall (qualsevol) | qu |
| \exists (existeix) | ex |
| \rightarrow (implicació) | im |
| \neg (negació) | nn |
| \wedge (i) | ii |
| \vee (o) | oo |

Si per exemple volem escriure la solució de

Per Nadal els estudiants tenen vacances i només aquests.
 $(X, S(x), V(x))$

que és

$$N \rightarrow \forall x((S(x) \rightarrow V(x)) \wedge (V(x) \rightarrow S(x)))$$

la Solució Codificada és

$$N \text{ im qu } x((S(x) \text{ im } V(x)) \text{ ii } (V(x) \text{ im } S(x)))$$

els Àtoms són

SV

les variables

x

les Constants

N

Per tant, la solució específica d'aquell enunciat s'escriu:

$$N \text{ im qu } x((S(x) \text{ im } V(x)) \text{ ii } (V(x) \text{ im } S(x))) \%SV \%x \%N$$

Si un exercici no té constants, igualment s'escriurà el separador '%':

$$\text{qu } x(P(x) \text{ im } G(x) \text{ ii } V(x)) \text{ ii qu } x(T(x) \text{ im } G(x) \text{ ii } V(x)) \%PTGV \%x \%$$

Resultat del bloc dels exercicis

Ja tenim totes les peces per construir els enunciats i solucions d'un problema. Si per exemple fem cinc exercicis:

```

\underline{Par`ametes:}\par \vspace{0.5cm}
% <P>
{\bf P1} \par % <1>
1 → $Per Nadal els estudiants tenen vacances i només aquests. \\(X, S(x), V(x) $\\#$N im qu x( (S(x) im V(x) ) ii
(V(x) im S(x)) ) \%SV \%x \%N $. \par
2 → $Les pomes i les taronges son gustoses i nutritives. \\(P(x), T(x), G(x), V(x)) $\\#$qu x(P(x) im G(x) ii V(x) ) ii
qu x(T(x) im G(x) ii V(x)) \%PTGV \%x \% $. \par
3 → $Hi ha aliments que només es poden menjar si són cuits. \\(A(x), M(x), C(x)) $\\#$ex x(A(x) ii (M(x) im C(x)))
%AMC \%x \% $. \par
4 → $Sense frens no hi ha cap auto segur. \\(F(x), A(x), S(x)) $\\#$nn ex x(A(x) ii S(x) ii nn F(x)) %FAS \%x \% $. \par
5 → $Els homes alts són atractius si són rossos i ben plantats. \\(H(x), A(x), T(x), R(x), B(x)) $\\#$qu x (H(x) ii A(x)
im (R(x)ii B(x) im T(x))) %HATRB \%x \% $. \par
% </1>
% </P>

```

Amb constants
Sense constants

Codi Mathematica de resolució

Tots els problemes han de contenir el codi de resolució per comprovar la correctesa de les solucions enviades pels alumnes.

Aquest codi serà sempre el mateix en aquest tipus de problemes.

S'encarrega, primer de tot de reinicialitzar les variables, per a continuació substituir a la variable 'txtalumne' la cadena de la solució que ha escrit l'alumne, a 'txtprofessor' la cadena de la solució que ha escrit el professor a SolucióCodificada al fitxer LaTeX.

A continuació fa l'avaluació de l'equivalència amb la funció 'TautologyQ' del Mathematica 8, que retorna cert en cas que la solució de l'alumne sigui equivalent a la del professor, i fals en cas contrari.

És important que el codi Mathematica contingui les paraules claus per avaluar el resultat de la correcció, que per al nostre cas són 'Correcte' i 'Incorrecte'.

```
\vspace{1cm}

\underline{Codi Matem`atica:}
\begin{verbatim}
Clear[txtalumne,txtprofessor,dacord];

SetOptions[{$Output, PageWidth->Infinity};

txtalumne=SO;
txtprofessor=P1;

dacord=False;

If[(TautologyQ[Equivalent[txtalumne, txtprofessor]]) === True, dacord = True];
If[dacord,Print["Correcte"],Print["Incorrecte"]];
\end{verbatim}
```


Exemple de problema de formalització de lògica de predicats

L'exemple que ve a continuació és un problema que formava part del dossier d'exercicis de l'assignatura de lògica de les enginyeries informàtiques.

El problema demana que es formalitzi un enunciat de lògica de predicats.

```
43
\begin{center}
\large {\bf Problema 1.01---ACME}
\end{center}

\underline{Enunciats:}\par \vspace{0.5cm}

* % <E>
Formalitza l'enunciat seg\u{u}ent:

\tt
\hspace{0.5 cm}
% <1>
{\bf P1}

% </E>
\par

\vspace{1cm}

\underline{Par`ametros:}\par \vspace{0.5cm}

% <P>
{\bf P1} \par % <1>
$Les pomes i les taronges son gustoses i nutritives. \((P(x), T(x), G(x), V(x)) \text{ \$}\#\text{\$qu } x(P(x) \text{ im } G(x) \text{ ii } V(x) ) \text{ ii qu } x(T(x) \text{ im } G(x) \text{ ii } V(x)) \text{ \%PTGV } \%x \% \$ . \par
$Hi ha aliments que només es poden menjar si són cuits. \((A(x), M(x), C(x)) \text{ \$}\#\text{\$ex } x(A(x) \text{ ii } (M(x) \text{ im } C(x))) \text{ \%AMC } \%x \% \$ . \par
$Sense frens no hi ha cap auto segur. \((F(x), A(x), S(x)) \text{ \$}\#\text{\$nn ex } x(A(x) \text{ ii } S(x) \text{ ii nn } F(x)) \text{ \%FAS } \%x \% \$ . \par
$Els homes alts són atractius si són rossos i ben plantats. \((H(x), A(x), T(x), R(x), B(x)) \text{ \$}\#\text{\$qu } x (H(x) \text{ ii } A(x) \text{ im } (R(x) \text{ ii } B(x) \text{ im } T(x))) \text{ \%HATRB } \%x \% \$ . \par
$Els que guanyen en el joc són els que tenen sort i només aquests. \((G(x), S(x)) \text{ \$}\#\text{\$qu } x( (G(x) \text{ im } S(x)) \text{ ii } (S(x) \text{ im } G(x))) \text{ \%GS } \%x \% \$ . \par
$Per Nadal els estudiants tenen vacances i només aquests. \((X, S(x), V(x)) \text{ \$}\#\text{\$N im qu } x( (S(x) \text{ im } V(x) ) \text{ ii } (V(x) \text{ im } S(x)) ) \text{ \%SV } \%x \%N \$ . \par
$No pas tothom és ric, culte i educat, ni no tots els rics són educats i cultes. \((R(x), C(x), D(x)) \text{ \$}\#\text{\$nn qu } x (R(x) \text{ ii } C(x) \text{ ii } D(x)) \text{ ii nn qu } x(R(x) \text{ im } D(x) \text{ ii } C(x)) \text{ \%RCD } \%x \% \$ . \par
$No totes les coses comprades a baix preu són flonges i trencadisses. \((C(x), B(x), F(x), T(x)) \text{ \$}\#\text{\$nn qu } x(C(x) \text{ ii } B(x) \text{ im } F(x) \text{ ii } T(x)) \text{ \%CBFT } \%x \% \$ . \par
$No tot home que deserta és un covard. \((H(x), S(x), C(x)) \text{ \$}\#\text{\$nn qu } x(H(x) \text{ ii } S(x) \text{ im } C(x)) \text{ \%HSC } \%x \% \$ . \par
$Per tenir èxit cal treballar fort si s'està en un negoci, o estudiar de valent si s'entra en una facultat. \((X(x), T(x), G(x), V(x), F(x)) \text{ \$}\#\text{\$qu } x (X(x) \text{ im } (G(x) \text{ im } T(x)) \text{ oo } (F(x) \text{ im } V(x))) \text{ \%XTGVF } \%x \% \$ . \par
```

```

$Hi ha francesos que només són amics dels catalans.  $\forall (F(x), A(x,y), C(x)) \exists x (F(x) \wedge \neg \exists y (A(x,y) \wedge C(y)))$  %FAC %xy % $. \par
$Hi ha francesos que són amics de tots els catalans.  $\forall (F(x), A(x,y), C(x)) \exists x (F(x) \wedge \forall y (C(y) \wedge A(x,y)))$  %FAC %xy % $. \par
$A l'estiu en Bernat i tots els seus fills s'atipen d'hamburgueses.  $\forall (S, B, F(x,y), A(x,y), H) \exists x (S \wedge A(B,H) \wedge \forall y (F(x,y) \wedge A(x,H)))$  %FA %xy %SBH $. \par
$Per passar curs no cal aprovar totes les assignatures.  $\forall (P(x), S(x), A(x)) \exists x (P(x) \wedge \forall y (S(y) \wedge A(x,y)))$  %PSA %xy % $. \par
%</1>
%</P>

```

```
\vspace{1cm}
```

```
\underline{Codi Matem`atica:}
```

```
\begin{verbatim}
```

```
Clear[txtalumne,txtprofessor,dacord];
```

```
SetOptions[Output, PageWidth->Infinity];
```

```
txtalumne=SO;
```

```
txtprofessor=P1;
```

```
dacord=False;
```

```
If[(TautologyQ[Equivalent[txtalumne, txtprofessor]]) === True, dacord = True];
```

```
If[dacord,Print["Correcte"],Print["Incorrecte"]];
```

```
\end{verbatim}
```

Tot seguit podem veure alguns dels enunciats que es poden generar a partir del fitxer d'exemple del quadre anterior:

Formalitza l'enunciat següent:

Les pomes i les taronges son gustoses i nutritives.
 $(P(x), T(x), G(x), V(x))$

Formalitza l'enunciat següent:

Per tenir èxit cal treballar fort si s'està en un negoci, o estudiar de valent si s'entra en una facultat.
 $(X(x), T(x), G(x), V(x), F(x))$

Formalitza l'enunciat següent:

Hi ha francesos que són amics de tots els catalans.
 $(F(x), A(x,y), C(x))$

Formalitza l'enunciat següent:

Per Nadal els estudiants tenen vacances i només aquests.
(X, S(x), V(x))

Formalitza l'enunciat següent:

No pas tothom és ric, culte i educat, ni no tots els rics són educats i cultes.
(R(x), C(x), D(x))

Formalitza l'enunciat següent:

Per passar curs no cal aprovar totes les assignatures.
(P(x), S(x), A(x))

16.2 Ajuda per a l'alumne dels exercicis de Formalització de lògica de predicats

Per dur a terme els exercicis de formalització de lògica de predicats, s'ha de formalitzar l'oració que ens dona l'enunciat.

Mitjançant l'editor (següent imatge) introduïrem la solució:

1 Formalitza l'enunciat següent:

A l'estiu en Bernat i tots els seus fills s'atipen d'hamburgueses.
(S, B, F(x,y), A(x,y), H)

2 \exists \forall \rightarrow \neg \wedge \vee ()

3 F A 4 x y , 5 S B H

6

7 Corregir 8 Ajuda

1. Enunciat a formalitzar.
2. Botons amb els símbols lògics.
3. Botons amb les lletres permeses per als àtoms.
4. Botons amb les lletres permeses per a les variables, i si n'hi ha més d'una, el boto de coma.
5. Botons amb les lletres permeses com a constants, només quan l'exercici ho requereix.
6. Camp a on escriure la resposta.
7. Botó per enviar la solució a corregir.
8. Botó d'ajuda.

La solució també es pot escriure amb el teclat de l'ordinador, sense utilitzar els botons de la interfície.

Per escriure els símbols lògics s'ha creat unes dreceres de teclat que utilitzen els números del 1 al 6 i que es poden consultar a l'ajuda.

Ajuda

Respecta les majúscules i les minúscules. [x]
Dreceres de teclat per als connectors lògics:
1: \exists 2: \forall 3: \rightarrow 4: \neg 5: \wedge 6: \vee

16.3 Ajuda per al professor dels exercicis de Formalització de lògica de proposicions

Un **problema** de formalització de lògica de proposicions està format per una sèrie d'**exercicis**. Cada alumne haurà de resoldre només un dels exercicis entre els que es preparin en aquest problema, que serà assignat de manera aleatòria pel sorteig que fa la plataforma ACME.

Un exemple de problema amb tres exercicis seria per exemple:

Formalitza les següents proposicions:

- *Si per la nit tinc son i no tinc feina dormo bé. Només tinc son per la nit si m'he llevat d'hora al matí i no he fet la migdiada a la tarda. Quan em llevo d'hora al matí no tinc feina per la nit. M'he llevat d'hora i no he fet la migdiada.*
- *Dormo o navego per internet o no tinc gana. Quan menjo tinc set i no tinc fred. Quan tinc gana no dormo i no navego per internet Quan no dormo menjo. Quan tinc gana i no navego per internet tinc set.*
- *O treballa o estudio a la universitat. Si vaig a la universitat hauré de llogar un pis. Calen diners per poder llogar un pis. Només treballant puc tenir diners.*

Mentre que cada alumne només veuria un exercici, per exemple:

Formalitza les següents proposicions:

O treballa o estudio a la universitat. Si vaig a la universitat hauré de llogar un pis. Calen diners per poder llogar un pis. Només treballant puc tenir diners.

Es poden generar varis problemes del mateix tipus, la formalització de lògica de proposicions, on s'hagin agrupat els enunciats per nivell de dificultat. D'aquesta manera, cada alumne haurà de fer un enunciat de cada problema que es generi.

Cada problema s'implementarà en un fitxer, utilitzant una sintaxis molt concreta que s'indica a continuació.

Implementació dels problemes d'avaluació continuada

El fitxer que contingui la implementació del problema ha de ser un fitxer en format LaTeX. El fitxer estarà escrit utilitzant la següent sintaxis.

Capçalera del fitxer

Al principi del fitxer hi ha d'haver una petita capçalera que identifica de quin tipus d'exercici es tracta i de quin nivell és el problema que anem a verificar. En aquest cas, el número del tipus d'exercici associat a la formalització de lògica de proposicions ha de ser el número **44**. I "Problema 1.01---ACME" indica que es tracta d'un problema de nivell 1, és a dir, d'avaluació continuada.

```
44
\begin{center}
\large {\bf Problema 1.01---ACME}
```

Enunciat del problema

Un problema tindrà un enunciat comú a tots els exercicis del problema. Els enunciats es definiran de la següent manera:

```
\underline{Enunciats:}\par \vspace{0.5cm}
```

```
* % <E>
```

Formalitzeu les següents proposicions:

```
\tt
```

```
\hspace{0.5 cm}
```

```
% <1>
```

```
{\bf P1}
```

```
% </E>
```

```
\par
```

```
\vspace{1cm}
```

L'identificador de l'enunciat és '% <E>' i cal respectar els salts de línia, les majúscules i els espais en blanc (entre '%' i '<E>' o '<1>' hi ha un espai en blanc).

En aquest tipus de problemes, no hi ha paràmetres a l'enunciat, pel que només hem de tenir '% <1>'. És molt important fer un canvi de línia després d'aquest identificador i escriure '{\bf P1}' i tornar a fer un canvi de línia. Aquesta línia no es tindrà en compte a l'hora de generar els enunciats, només és una manera per fer estètic el problema per si es vol visualitzar tot el fitxer LaTeX del problema, però si no hi és, no s'escriu la línia que va a continuació.

Definició dels exercicis com a paràmetre del problema

Els diferents exercicis del problema els escriurem tots com un únic paràmetre del problema.

Per indicar que comencen els paràmetres s'ha d'escriure '% <P>' (igual que amb '% <E>' hi ha un espai entre '%' i '<P>'). I per indicar que els paràmetres han acabat es fa servir el identificador '% </P>'. A més a més, per indicar que comença el primer paràmetre, que en aquest tipus de problema és l'únic, es fa servir el identificador '% <1>' seguit d'un salt de línia. I per indicar que acaba el primer paràmetre s'utilitza '% </1>'.

Entre aquests dos identificadors, '% <1>' i '% </1>', s'han d'escriure els diferents exercicis, que en el següent quadre ha estat indicat amb els punt suspensius.

```
\underline{Par\`ametres:}\par \vspace{0.5cm}
```

```
% <P>
```

```
{\bf P1} \par % <1>
```

```
...
```

```
% </1>
```

```
% </P>
```

Cada exercici ha d'estar en una línia del fitxer LaTeX.

Les expressions dels exercicis estaran formades per dues parts ben diferenciades:

- L'enunciat d'aquell exercici concret.
Escrit en LaTeX, generarà el enunciat particular que visualitzarà l'alumne
- La solució d'aquell exercici concret.
Escrit en una codificació pròpia, l'utilitzarà el nucli corrector del problema per corregir la solució de l'alumne

Aquestes expressions tenen un '\$' al inici i al final, '\$\#' per diferenciar el enunciat de la solució, i al final de tot de la línia un '. \par' per indicar el final de línia:

```
$ enunciat $\#$solució $. \par
```

La solució està formada per tres parts:

1. Número de frases de l'enunciat
2. Solució, en un codificació pròpia
3. Els àtoms

Cadascuna d'aquestes parts es separa amb un '%', pel que la estructura serà la següent:

```
NumLínies %SolucióCodificada %Àtoms
```

Si per exemple volem fer un problema amb 5 exercicis diferents, la definició dels exercicis tindrà la següent forma:

```
\underline{Par\`ametres:}\par \vspace{0.5cm}  
  
% <P>  
{\bf P1}\par % <1>  
$ enunciat1 $\#$ NumLínies1 %SolucióCodificada1 %Àtoms1 $. \par  
$ enunciat2 $\#$ NumLínies2 %SolucióCodificada2 %Àtoms2 $. \par  
$ enunciat3 $\#$ NumLínies3 %SolucióCodificada3 %Àtoms3 $. \par  
$ enunciat4 $\#$ NumLínies4 %SolucióCodificada4 %Àtoms4 $. \par  
$ enunciat5 $\#$ NumLínies5 %SolucióCodificada5 %Àtoms5 $. \par  
% </1>  
% </P>
```

Definició de cada enunciat específic

Per escriure la part d'enunciat, ho hem de fer en LaTeX. Si volem que l'enunciat específic que veurà l'alumne tingui salts de línia, al fitxer LaTeX ho indicarem amb '\\'.

Per exemple, per a que l'enunciat particular que veu l'alumne sigui:

Si per la nit tinc son i no tinc feina dormo bé.

Només tinc son per la nit si m'he llevat d'hora al matí i no he fet la migdiada a la tarda.

Quan em llevo d'hora al matí no tinc feina per la nit.

M'he llevat d'hora i no he fet la migdiada.

(S, F, D, M, T)

al fitxer LaTeX s'haurà d'escriure:

```
Si per la nit tinc son i no tinc feina dormo bé. \\Només tinc son per la nit si m'he llevat d'hora al
matí i no he fet la migdiada a la tarda. \\Quan em llevo d'hora al matí no tinc feina per la nit.
\\M'he llevat d'hora i no he fet la migdiada. \\(S, F, B, M, T)
```

També és important fixar-se que en aquest exemple s'esperaria que l'àtom 'dormo bé' es representés amb 'D' i que s'ha fet amb 'B'. Això es deu a que el llenguatge que utilitza el nucli corrector, el Mathematica, té unes lletres reservades que no podem utilitzar com a àtoms, i que són les següents:

D E I N O

Podrem per tant utilitzar per representar àtoms i constants les lletres de la 'A' a la 'Z' en majúscula i amb excepció de 'D', 'E', 'I', 'N', 'O'.

Definició de cada solució específica

Hem vist que cada solució té la forma:

```
NumLínies %SolucióCodificada %Àtoms
```

Anem a veure com escriurem cadascuna d'aquestes parts:

NumLínies és el número que indica de quantes frases està format l'enunciat.

Els *Àtoms* hauran de ser lletres majúscules de la 'A' a la 'Z', excepte 'D', 'E', 'I', 'N', 'O'.

A la *SolucióCodificada* s'ha de substituir els connectors propis d'una formalització en lògica de proposicions per una codificació pròpia utilitzada en aquest tipus d'exercicis.

I cada línia es separarà de la següent amb una coma ','

| Símbol lògic | Codificació |
|----------------|-------------|
| → (implicació) | im |
| ¬ (negació) | nn |
| ∧ (i) | ii |
| ∨ (o) | oo |

Si per exemple volem escriure la solució de

Si per la nit tinc son i no tinc feina dormo bé.

Només tinc son per la nit si m'he llevat d'hora al matí i no he fet la migdiada a la tarda.

Quan em llevo d'hora al matí no tinc feina per la nit.

M'he llevat d'hora i no he fet la migdiada.

(S, F, B, M, T)

que és

$S \wedge \neg F \rightarrow B$

$S \rightarrow M \wedge \neg T$

$M \rightarrow \neg F$

el *NumLínies* és 4

la *SolucióCodificada* és S ii nn F im B , S im M ii nn T , M im nn F , M ii nn T

els *Àtoms* són SFBMT

Per tant, la solució específica d'aquell enunciat s'escriu:

```
4 % S ii nn F im B , S im M ii nn T , M im nn F , M ii nn T % SFBMT
```

Línies separades per comes

Resultat del bloc dels exercicis

Ja tenim totes les peces per construir els enunciats i solucions d'un problema.

Si per exemple fem cinc exercicis:

```
\underline{Par\`ametres:}\par \vspace{0.5cm}

% <P>
{\bf P1} \par % <1>
1 → $Si el fiscal és deshonest i el gerent és un estafador, el president és un lladre. \Si el gerent és un estafador però el president no és cap lladre, el fiscal no és pas deshonest. \((F, G, P) $\#$2 % F ii G im P , G ii nn P im nn F % FGP $. \par
2 → $Dormo o navego per internet o no tinc gana. \Quan menjo tinc set i no tinc fred. \Quan tinc gana no dormo i no navego per internet. \Quan no dormo menjo. \Quan tinc gana i no navego per internet tinc set. \((B, A, G, M, S, F) $\#$5 % B oo A oo nn G , M im S ii nn F , G im nn B ii nn A , nn B im M , G ii nn A im S % BAGMSF $. \par
3 → $Quan em dreprimeixo menjo rovellons i arengades. \Quan menjo arengades tinc set i tinc fred. \Tant si tinc frec com si tinc set, en ambdós casos, menjo galetes. \Quan menjo galetes, si tinc set, no menjo arengades. \Quan menjo arengades no menjo galetes i no em deprimeixo. \((B, R, A, S, F, G) $\#$5 % B im R ii A , A im S ii F , F oo S im G , G im (S im nn A) , A im nn G ii nn B % BRASFG $. \par
4 → $Si em gradués en informàtica, aniria a fer un màster a l'estranger. \Si no surto massa per les nits em graduaré en Informàtica. \Si no vull treballar en un banc cal que no surti massa. \Si treballés en un banc acabaria casant-me. \Si no anés a fer un màster a l'estranger acabaria casant-me. \((G, M, S, B, C) $\#$5 % G im M , nn S im G , nn B im nn S , B im C , nn M im C % GMSBC $. \par
5 → $O treball o estudio a la universitat. \Si vaig a la universitat hauré de llogar un pis. \Calen diners per poder llogar un pis. \Només treballant puc tenir diners. \((T, U, P, B) $\#$4 % T oo U , U im P , P im B , B im T % TUPB $. \par
% </1>
% </P>
```

Codi Mathematica de resolució

Tots els problemes han de contenir el codi de resolució per comprovar la correctesa de les solucions enviades pels alumnes.

Aquest codi serà sempre el mateix en aquest tipus de problemes.

S'encarrega, primer de tot de reinicialitzar les variables, per a continuació substituir a la variable 'txtalumne' la cadena de la solució que ha escrit l'alumne, a 'txtprofessor' la cadena de la solució que ha escrit el professor a SolucióCodificada al fitxer LaTeX. A continuació fa l'avaluació de l'equivalència amb la funció 'TautologyQ' del Mathematica 8, que retorna cert en cas que la solució de l'alumne sigui equivalent a la del professor, i fals en cas contrari. És important que el codi Mathematica contingui les paraules claus per avaluar el resultat de la correcció, que per al nostre cas són 'Correcte' i 'Incorrecte'.

```

\vspace{1cm}

\underline{Codi Matem`atica;}
\begin{verbatim}
Clear[txtalumne,txtprofessor,dacord];
SetOptions[Output, PageWidth->Infinity];
txtalumne=SO;
txtprofessor=P1;
dacord=False;
If[(TautologyQ[Equivalent[txtalumne, txtprofessor]]) == True, dacord = True];
If[dacord,Print["Correcte"],Print["Incorrecte"]];
\end{verbatim}

```

Exemple de problema de formalització de lògica de proposicions

L'exemple que ve a continuació és un problema que formava part del dossier d'exercicis de l'assignatura de lògica de les enginyeries informàtiques.

El problema demana que es formalitzi un enunciat de lògica de proposicions.

```

44
\begin{center}
\large {\bf Problema 1.01---ACME}
\end{center}

\underline{Enunciats:}\par \vspace{0.5cm}

* % <E>
Formalitzeu les següents proposicions:
\tt
\hspace{0.5 cm}
% <1>
{\bf P1}
% </E>
\par

\underline{Par`ametros:}\par \vspace{0.5cm}
% <P>
{\bf P1} \par % <1>
$Si el fiscal és deshonest i el gerent és un estafador, el president és un lladre. \Si el gerent és
un estafador però el president no és cap lladre, el fiscal no és pas deshonest. \((F, G, P) $\#$2
% F ii G im P, G ii nn P im nn F % FGP $. \par
$Dormo o navego per internet o no tinc gana. \Quan menjo tinc set i no tinc fred. \Quan tinc
gana no dormo i no navego per internet. \Quan no dormo menjo. \Quan tinc gana i no
navego per internet tinc set. \((B, A, G, M, S, F) $\#$5 % B oo A oo nn G, M im S ii nn F, G im
nn B ii nn A, nn B im M, G ii nn A im S % BAGMSF $. \par

```

$\text{\$Quan em deprimeixo menjo rovellons i arengades. \text{\$Quan menjo arengades tinc set i tinc fred. \text{\$Tant si tinc frec com si tinc set, en ambdós casos, menjo galetes. \text{\$Quan menjo galetes, si tinc set, no menjo arengades. \text{\$Quan menjo arengades no menjo galetes i no em deprimeixo. \text{\$((B, R, A, S, F, G) \text{\$}\#\text{\$5 \% B im R ii A , A im S ii F , F oo S im G , G im (S im nn A) , A im nn G ii nn B \% BRASFG \$. \text{\par

$\text{\$Si em gradués en informàtica, aniria a fer un màster a l'estranger. \text{\$Si no surto massa per les nits em graduaré en Informàtica. \text{\$Si no vull treballar en un banc cal que no surti massa. \text{\$Si treballés en un banc acabaria casant-me. \text{\$Si no anés a fer un màster a l'estranger acabaria casant-me. \text{\$((G, M, S, B, C) \text{\$}\#\text{\$5 \% G im M , nn S im G , nn B im nn S , B im C , nn M im C \% GMSBC \$. \text{\par

$\text{\$O treball o estudio a la universitat. \text{\$Si vaig a la universitat hauré de llogar un pis. \text{\$Calen diners per poder llogar un pis. \text{\$Només treballant puc tenir dineres. \text{\$((T, U, P, B) \text{\$}\#\text{\$4 \% T oo U , U im P , P im B , B im T \% TUPB \$. \text{\par

$\text{\% </1>$

$\text{\% </P>$

$\text{\vspace{1cm}}$

$\text{\underline{Codi Matem`atica:}}$

$\text{\begin{verbatim}}$

$\text{Clear[txtalumne,txtprofessor,dacord];}$

$\text{SetOptions[\$Output, PageWidth->Infinity];}$

txtalumne=SO;

txtprofessor=P1;

dacord=False;

$\text{If[(TautologyQ[Equivalent[txtalumne, txtprofessor]]) === True, dacord = True];}$

$\text{If[dacord,Print["Correcte"],Print["Incorrecte"]];}$

$\text{\end{verbatim}}$

Tot seguit podem veure alguns dels enunciats que es poden generar a partir del fitxer d'exemple del quadre anterior:

Formalitzeu les següents proposicions:

Si el fiscal és deshonest i el gerent és un estafador, el president és un lladre.

Si el gerent és un estafador però el president no és cap lladre, el fiscal no és pas deshonest.

(F, G, P)

Formalitzeu les següents proposicions:

Dormo o navego per internet o no tinc gana.

Quan menjo tinc set i no tinc fred.

Quan tinc gana no dormo i no navego per internet.

Quan no dormo menjo.

Quan tinc gana i no navego per internet tinc set.

(D, N, G, M, S, F)

Formalitzeu les següents proposicions:

Quan em dreprimeixo menjo rovellons i arengades.
Quan menjo arengades tinc set i tinc fred.
Tant si tinc frec com si tinc set, en ambdós casos, menjo galetes.
Quan menjo galetes, si tinc set, no menjo arengades.
Quan menjo arengades no menjo galetes i no em dreprimeixo.
(D, R, A, S, F, G)

Formalitzeu les següents proposicions:

O treballo o estudio a la universitat.
Si vaig a la universitat hauré de llogar un pis.
Calen diners per poder llogar un pis.
Només treballant puc tenir diners.
(T, U, P, D)

16.4 Ajuda per a l'alumne dels exercicis de Formalització de lògica de proposicions

Per dur a terme els exercicis de formalització de lògica de proposicions, s'ha de formalitzar cadascuna de les frases que ens dóna l'enunciat. Mitjançant l'editor (següent imatge) introduïrem la solució:

Formalitzeu les següents proposicions:

1

Dormo o navego per internet o no tinc gana.
Quan menjo tinc set i no tinc fred.
Quan tinc gana no dormo i no navego per internet.
Quan no dormo menjo.
Quan tinc gana i no navego per internet tinc set.
(D, N, G, M, S, F)

The image shows a web-based editor for formalizing logic propositions. At the top, there is a toolbar with two rows of buttons. The first row (labeled '2') contains logical symbols: \rightarrow , \neg , \wedge , \vee , $($, and $)$. The second row (labeled '3') contains letters: D, N, G, M, S, and F. Below the toolbar is a large text input area (labeled '4') with four horizontal lines. At the bottom of the input area are two buttons: 'Corregir' (labeled '5') and 'Ajuda' (labeled '6').

1. Enunciat a formalitzar.
2. Botons amb els símbols lògics.
3. Botons amb les lletres permeses per als àtoms.
4. Camps a on escriure la resposta, un per cada frase i en el mateix ordre en que es redacta l'enunciat.
5. Botó per enviar la solució a corregir.
6. Botó d'ajuda.

Ajuda

Respecta les majúscules i les minúscules. [x]
Dreceres de teclat per als connectors lògics:
1: \exists 2: \forall 3: \rightarrow 4: \neg 5: \wedge 6: \vee

La solució també es pot escriure amb el teclat de l'ordinador, sense utilitzar els botons de la interfície.

Per escriure els símbols lògics s'ha creat unes dreceres de teclat que utilitzen els números del 1 al 6 i que es poden consultar a l'ajuda.

16.5 Ajuda per al professor dels exercicis de Contraexemples de lògica de predicats

Un **problema** de contraexemples de lògica de predicats està format per una sèrie d'**exercicis**. Cada alumne haurà de resoldre només un dels exercicis entre els que es preparin en aquest problema, que serà assignat de manera aleatòria pel sorteig que fa la plataforma ACME.

Un exemple de problema amb tres exercicis seria per exemple:

El següent esquema lògic és incorrecte. Troba un contraexemple:

$$\begin{aligned} &\forall x(A(x) \rightarrow \neg B(x)) \\ &\exists x(B(x) \wedge C(x)) \\ \Rightarrow &\exists x(A(x) \wedge C(x)) \end{aligned}$$

$$\begin{aligned} &\exists x P(x) \\ &\exists x Q(x) \\ \Rightarrow &\exists x(P(x) \wedge Q(x)) \end{aligned}$$

$$\begin{aligned} &\exists x(P(x) \vee R(x)) \\ &\exists x(Q(x) \vee R(x)) \\ &\exists x(Q(x) \wedge Q(x) \rightarrow R(x)) \\ \Rightarrow &\exists x R(x) \end{aligned}$$

Mentre que cada alumne només veuria un exercici, per exemple:

El següent esquema lògic és incorrecte. Troba un contraexemple:

$$\begin{aligned} &\forall x(A(x) \rightarrow \neg B(x)) \\ &\exists x(B(x) \wedge C(x)) \\ \Rightarrow &\exists x(A(x) \wedge C(x)) \end{aligned}$$

Es poden generar varis problemes del mateix tipus, contraexemples de lògica de predicats, on s'hagin agrupat els enunciats per nivell de dificultat. D'aquesta manera, cada alumne haurà de fer un enunciat de cada problema que es generi.

Cada problema s'implementarà en un fitxer, utilitzant una sintaxis molt concreta que s'indica a continuació.

Implementació dels problemes d'avaluació continuada

El fitxer que contingui la implementació del problema ha de ser un fitxer en format LaTeX. El fitxer estarà escrit utilitzant la següent sintaxis.

Capçalera del fitxer

Al principi del fitxer hi ha d'haver una petita capçalera que identifica de quin tipus d'exercici es tracta i de quin nivell és el problema que anem a verificar.

En aquest cas, el número del tipus d'exercici associat a la formalització de lògica de predicats ha de ser el número **45**. I "Problema 1.01---ACME" indica que es tracta d'un problema de nivell 1, és a dir, d'avaluació continuada.

```
45
\begin{center}
\large {\bf Problema 1.01---ACME}
```

Enunciat del problema

Un problema tindrà un enunciat comú a tots els exercicis del problema. Els enunciats es definiran de la següent manera:

```
\underline{Enunciats:}\par \vspace{0.5cm}

* % <E>
El següent esquema lògic és incorrecte. Troba un contraexemple:

\tt
\hspace{0.5 cm}
% <1>
{\bf P1}

% </E>
\par

\vspace{1cm}
```

L'identificador de l'enunciat és '% <E>' i cal respectar els salts de línia, les majúscules i els espais en blanc (entre '%' i '<E>' o '<1>' hi ha un espai en blanc).

En aquest tipus de problemes, no hi ha paràmetres a l'enunciat, pel que només hem de tenir '% <1>'. És molt important fer un canvi de línia després d'aquest identificador i escriure '{\bf P1}' i tornar a fer un canvi de línia. Aquesta línia no es tindrà en compte a l'hora de generar els enunciats, només és una manera per fer estètic el problema per si es vol visualitzar tot el fitxer LaTeX del problema, però si no hi és, no s'escriu la línia que va a continuació.

Definició dels exercicis com a paràmetre del problema

Els diferents exercicis del problema els escriurem tots com un únic paràmetre del problema.

Per indicar que comencen els paràmetres s'ha d'escriure '% <P>' (igual que amb '% <E>' hi ha un espai entre '%' i '<P>'). I per indicar que els paràmetres han acabat es fa servir el identificador '% </P>'. A més a més, per indicar que comença el primer paràmetre, que en aquest tipus de problema és l'únic, es fa servir el identificador '% <1>' seguit d'un salt de línia.

I per indicar que acaba el primer paràmetre s'utilitza '% </1>'.
Entre aquests dos identificadors, '% <1>' i '% </1>', s'han d'escriure els diferents exercicis, que en el següent quadre ha estat indicat amb els punt suspensius.

```
\underline{Par`ametes:}\par \vspace{0.5cm}
% <P>
{\bf P1} \par % <1>
...
% </1>
%</P>
```

Cada exercici ha d'estar en una línia del fitxer LaTeX.

Les expressions dels exercicis estaran formades per dues parts ben diferenciades:

- L'enunciat d'aquell exercici concret
Escrit en LaTeX, generarà el enunciat particular que visualitzarà l'alumne.
- La informació d'aquell exercici concret.
Escrit en una codificació pròpia, l'utilitzarà el nucli corrector del problema per corregir la solució de l'alumne.

Aquestes expressions tenen un '\$' al inici i al final, '\$\#' per diferenciar el enunciat de la informació, i al final de tot de la línia un '. \par' per indicar el final de línia:

```
$ enunciat $\#$informació $. \par
```

La informació per aquest tipus d'exercicis es està formada per quatre parts:

1. El número de frases del raonament.
2. El enunciat, en un codificació pròpia
3. Els àtoms, que han de ser majúscules
4. Les variables, que han de ser minúscules

Cadascuna d'aquestes parts es separa amb un '%', pel que la estructura serà la següent:

```
NumFrases %EnunciatCodificat %Àtoms %variables %
```

En aquest tipus de problema, el professor no ha de proporcionar cap contraexemple que sigui solució, si no què amb la informació sobre l'exercici que aquí es proporciona, el Mathematica podrà avaluar si la solució del alumne és realment un contraexemple del seu enunciat.

També cal esmentar que després de les variables, hi ha preparat un separador '%' per si algun dia es pot ampliar aquest tipus de problemes per a que resolgui també casos amb informació addicional.

Si per exemple volem fer un problema amb 5 exercicis diferents, la definició dels exercicis tindrà la següent forma:

```

\underline{\Par`ametres:}\par \vspace{0.5cm}

% <P>
{\bf P1} \par % <1>
$ enunciat1 $\#$ NumFrases1 %EnunciatCodificat1 %Àtoms1 %variables1 %$. \par
$ enunciat2 $\#$ NumFrases2 %EnunciatCodificat2 %Àtoms1 %variables2 %$. \par
$ enunciat3 $\#$ NumFrases3 %EnunciatCodificat3 %Àtoms1 %variables3 %$. \par
$ enunciat4 $\#$ NumFrases4 %EnunciatCodificat4 %Àtoms1 %variables4 %$. \par
$ enunciat5 $\#$ NumFrases5 %EnunciatCodificat5 %Àtoms1 %variables5 %$. \par
% </1>
% </P>

```

Definició de cada enunciat específic

Per escriure la part d'enunciat, ho hem de fer en LaTeX. Necessitarem per tant conèixer les codificacions en LaTeX dels connector i quantificador lògics:

| Símbol | LaTeX |
|--------|-------------|
| ∃ | \exists |
| ∀ | \forall |
| → | \to |
| ¬ | \neg |
| ∧ | \wedge |
| ∨ | \vee |
| ⇒ | \Rightarrow |

A més, si volem que l'enunciat específic que veurà l'alumne tingui salts de línia, al fitxer LaTeX ho indicarem amb '\\'.

Per exemple, per a que l'enunciat particular que veu l'alumne sigui:

$$\forall x(A(x) \rightarrow \neg B(x))$$

$$\exists x(B(x) \wedge C(x))$$

$$\Rightarrow \exists x(A(x) \wedge C(x))$$

al fitxer LaTeX s'haurà d'escriure:

```

forall x (A(x) \to \neg B(x)) \\ \exists x (B(x) \wedge C(x)) \\ \Rightarrow \exists x(A(x) \wedge C(x))

```

Hem de tindre en compte que cap lletra que utilitzi l'enunciat com a àtom pot ser una paraula reservada del Mathematica, ja que és en aquest llenguatge que està implementat el nucli corrector dels problemes d'aquest tipus.

El Mathematica, té les següents lletres reservades:

D E I N O

Podrem per tant utilitzar per representar àtoms i variables les lletres de la 'A' a la 'Z' amb excepció de 'D', 'E', 'I', 'N', 'O'.

Definició de la informació específica

Hem vist que cada enunciat té associada una informació específica per a que el Mathematica pugui avaluar si les dades proporcionades per l'alumne són realment un contraexemple.

Aquesta informació necessària té la forma:

```
NumFrases %EnunciatCodificat %Àtoms %variables %
```

Anem a veure com escriurem cadascuna d'aquestes parts:

El *NúmFrases* és el número de frases del raonament, és a dir, el número de premisses més 1 per la conclusió.

Els *àtoms* hauran de ser lletres majúscules de la 'A' a la 'Z', excepte 'D','E','I','N','O'.

Les *variables* hauran de ser lletres minúscules de la 'a' a la 'z', excepte 'd','e','i','n','o'.

L'*EnunciatCodificat* s'ha de substituir els connectors lògics i quantificadors propis d'una formalització en lògica de predicats per una codificació pròpia utilitzada en aquest tipus d'exercicis. I cada frase es separarà de la següent amb una coma ',' i indistintament si el que ve a continuació és una premissa o ja es tracta de la conclusió.

| Símbol lògic | Codificació |
|----------------------------|-------------|
| \forall (qualsevol) | qu |
| \exists (existeix) | ex |
| \rightarrow (implicació) | im |
| \neg (negació) | nn |
| \wedge (i) | ii |
| \vee (o) | oo |

Si per exemple volem escriure la informació de

$$\begin{aligned} &\forall x(A(x) \rightarrow \neg B(x)) \\ &\exists x(B(x) \wedge C(x)) \\ &\Rightarrow \exists x(A(x) \wedge C(x)) \end{aligned}$$

el NumFrases és 2 premisses més 1 conclusió, és a dir:

3

el EnunciatCodificat és

qu x (A(x) im nn B(x)) , ex x (B(x) ii C(x)) , ex x(A(x) ii C(x))

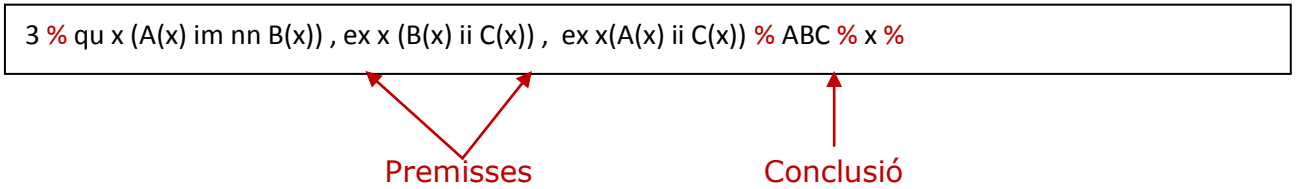
els Àtoms són

ABC

les variables

x

Per tant, la informació específica d'aquell enunciat s'escriu:



Resultat del bloc dels exercicis

Ja tenim totes les peces per construir els enunciats i la seva informació associada d'un problema.

Si per exemple fem cinc exercicis:

```

\underline{Par\`ametros:}\par \vspace{0.5cm}

% <P>
{\bf P1} \par % <1>
1 → \$\exists x (P(x) \wedge A(x)) \\\ \forall x (P(x) \to G(x)) \\\ \exists x (P(x) \wedge \neg A(x)) \\\
\Rightarrow \forall x (A(x) \to G(x)) \$\#\$4 % ex x (P(x) ii A(x)) , qu x (P(x) im G(x)) , ex x (P(x) ii nn
A(x)) , qu x (A(x) im G(x)) \% AGP \% x \% $. \par
2 → \$\exists x (P(x) \vee R(x)) \\\ \exists x (Q(x) \vee R(x)) \\\ \forall x (P(x) \wedge Q(x) \to R(x)) \\\
\Rightarrow \exists x R(x) \$\#\$4 % ex x (P(x) oo R(x)) , ex x (Q(x) oo R(x)) , qu x (P(x) ii Q(x) im R(x))
, ex x R(x) \% PQR \% x \% $. \par
3 → \$\forall x (A(x) \to \neg B(x)) \\\ \exists x (B(x) \wedge C(x)) \\\ \Rightarrow \exists x(A(x) \wedge
C(x)) \$\#\$3 % qu x (A(x) im nn B(x)) , ex x (B(x) ii C(x)) , ex x(A(x) ii C(x)) \% ABC \% x \% $. \par
4 → \$\exists x (A(x) \to \neg B(x)) \\\ \forall x (B(x) \to \neg C(x)) \\\ \Rightarrow \forall x (C(x) \to A(x))
\$\#\$3 % ex x (A(x) im nn B(x)) , qu x (B(x) im nn C(x)) , qu x (C(x) im A(x)) \% ABC \% x \% $. \par
5 → \$\exists x P(x) \\\ \exists x Q(x) \\\ \Rightarrow \exists x (P(x) \wedge Q(x)) \$\#\$3 % ex x P(x) , ex x
Q(x) , ex x (P(x) ii Q(x)) \% PQ \% x \% $. \par
% </1>
%</P>

```

Codi Mathematica de resolució

Tots els problemes han de contenir el codi de resolució per comprovar la correctesa de les solucions enviades pels alumnes.

Aquest codi serà sempre el mateix en aquest tipus de problemes.

S'encarrega, primer de tot de reinicialitzar les variables.

A continuació substitueix a la variable '*regles*' el contraexemple que ha escrit l'alumne.

A *PremissesConclusio* substitueix el raonament que el professor ha escrit a la informació de l'exercici al fitxer LaTeX, i que ja arriba aquí amb dues variables generades pel nucli corrector, *Premisses* i *Conclusio*.

Per tant, quant executem la variable *PremissesConclusio*, tenim per separat les premisses a la variable *Premises* i la conclusió a la variable *Conclusio*.

A continuació substituïm el contraexemple del alumne a les premisses i comprovem que sigui cert, i substituïm el contraexemple del alumne a la conclusió i comprovem que sigui fals.

Altrament, la solució proporcionada per l'alumne no és una solució contraexemple de l'enunciat.

És important que el codi Mathematica contingui les paraules claus per avaluar el resultat de la correcció, que per al nostre cas són 'Correcte' i 'Incorrecte'.

```
\vspace{1cm}

\underline{Codi Matem\`atica;}
\begin{verbatim}
Clear[Regles,PremissesConclusio,dacord];

SetOptions[ $\$Output$ , PageWidth->Infinity];

Regles=SO;
PremissesConclusio=P1;
dacord=False;
PremissesConclusio;
If[(((Premises /. Regles) == True) && ((Conclusio /. Regles) == False)) == True, dacord = True];
If[dacord,Print["Correcte"],Print["Incorrecte"]];
\end{verbatim}
```

Exemple de problema de contraexemple de lògica de predicats

L'exemple que ve a continuació és un problema que formava part del dossier d'exercicis de l'assignatura de lògica de les enginyeries informàtiques.

El problema demana que es formalitzi un enunciat de lògica de predicats.

```
45
\begin{center}
\large {\bf Problema 1.01---ACME}
\end{center}
\underline{Enunciats:}\par \vspace{0.5cm}
* % <E>
El següent esquema lògic és incorrecte. Troba un contraexemple:
\tt
\hspace{0.0 cm}
$
% <1>
{\bf P1}
$
% </E>
\par

\underline{Par\`ametes:}\par \vspace{0.5cm}
% <P>
{\bf P1} \par % <1>
$\exists x (P(x) \wedge A(x)) \implies \forall x (P(x) \to G(x)) \implies \exists x (P(x) \wedge \neg A(x)) \implies \forall x (A(x) \to G(x))$
% ex x (P(x) ii A(x)) , qu x (P(x) im G(x)) , ex x (P(x) ii nn A(x)) , qu x (A(x) im G(x)) % AGP % x % $. \par
$\exists x (P(x) \vee R(x)) \implies \exists x (Q(x) \vee R(x)) \implies \forall x (P(x) \wedge Q(x) \to R(x)) \implies \exists x R(x)$
% ex x (P(x) oo R(x)) , ex x (Q(x) oo R(x)) , qu x (P(x) ii Q(x) im R(x)) , ex x R(x) % PQR % x % $. \par
$\forall x (A(x) \to \neg B(x)) \implies \exists x (B(x) \wedge C(x)) \implies \exists x (A(x) \wedge C(x))$
% qu x (A(x) im nn B(x)) , ex x (B(x) ii C(x)) , ex x(A(x) ii C(x)) % ABC % x % $. \par
$\exists x (A(x) \to \neg B(x)) \implies \forall x (B(x) \to \neg C(x)) \implies \forall x (C(x) \to A(x))$
% ex x (A(x) im nn B(x)) , qu x (B(x) im nn C(x)) , qu x (C(x) im A(x)) % ABC % x % $. \par
$\exists x P(x) \implies \exists x Q(x) \implies \exists x (P(x) \wedge Q(x))$
% ex x P(x) , ex x Q(x) , ex x (P(x) ii Q(x)) % PQ % x % $. \par
% </1>
% </P>

\underline{Codi Matem\`atica:}
\begin{verbatim}
Clear[Regles,PremissesConclusio,dacord];
SetOptions[Output, PageWidth->Infinity];
Regles=SO;
PremissesConclusio=P1;
dacord=False;
PremissesConclusio;
If[(((Premises /. Regles) == True) && ((Conclusio /. Regles) == False)) == True, dacord = True];
If[dacord,Print["Correcte"],Print["Incorrecte"]];
\end{verbatim}
```

Tot seguit podem veure alguns dels enunciats que es poden generar a partir del fitxer d'exemple del quadre anterior:

El següent esquema lògic és incorrecte. Troba un contraexemple:

$$\begin{aligned} & \exists x(P(x) \vee R(x)) \\ & \exists x(Q(x) \vee R(x)) \\ & \forall x(P(x) \wedge Q(x) \rightarrow R(x)) \\ \Rightarrow & \exists xR(x) \end{aligned}$$

El següent esquema lògic és incorrecte. Troba un contraexemple:

$$\begin{aligned} & \exists xP(x) \\ & \exists xQ(x) \\ \Rightarrow & \exists x(P(x) \wedge Q(x)) \end{aligned}$$

El següent esquema lògic és incorrecte. Troba un contraexemple:

$$\begin{aligned} & \forall x(A(x) \rightarrow \neg B(x)) \\ & \exists x(B(x) \wedge C(x)) \\ \Rightarrow & \exists x(A(x) \wedge C(x)) \end{aligned}$$

El següent esquema lògic és incorrecte. Troba un contraexemple:

$$\begin{aligned} & \exists x(P(x) \wedge A(x)) \\ & \forall x(P(x) \rightarrow G(x)) \\ & \exists x(P(x) \wedge \neg A(x)) \\ \Rightarrow & \forall x(A(x) \rightarrow G(x)) \end{aligned}$$

16.6 Ajuda per a l'alumne dels exercicis de Contraexemples de lògica de predicats

Per dur a terme els problemes de contraexemple de lògica de predicats, l'alumne ha d'introduir un contraexemple al raonament de l'enunciat.

L'editor d'aquest tipus de problema ens permetrà en dues passes indicar quin és el contraexemple que es dona com a solució.

En un primer pas, veurem una pantalla com la següent:

1 El següent esquema lògic és incorrecte. Troba un contraexemple:

$$\begin{aligned} & \exists x(P(x) \vee R(x)) \\ & \exists x(Q(x) \vee R(x)) \\ & \forall x(P(x) \wedge Q(x) \rightarrow R(x)) \\ & \Rightarrow \exists xR(x) \end{aligned}$$

2 Número de referenciats: ... ▼

Corregir

1. Enunciat al que cal trobar un contraexemple.
2. Desplegable on triar el número de referenciats que té el contraexemple que es vol introduir, i que pot anar entre 1 i 5.

Si per exemple triem 2 referenciats, a la interfície apareixen els desplegable necessaris per introduir el contraexemple:

El següent esquema lògic és incorrecte. Troba un contraexemple:

$$\begin{aligned} & \exists x(P(x) \vee R(x)) \\ & \exists x(Q(x) \vee R(x)) \\ & \forall x(P(x) \wedge Q(x) \rightarrow R(x)) \\ & \Rightarrow \exists xR(x) \end{aligned}$$

Número de referenciats: 2 ▼

3

P[a]= ... ▼ P[b]= ... ▼
Q[a]= ... ▼ Q[b]= ... ▼
R[a]= ... ▼ R[b]= ... ▼

Corregir 4

3. Desplegables per introduir el contraexemple
4. Botó per enviar la solució a Corregir