

PROGETTO DI UNA UNITÀ DI RICERCA - MODELLO B
Anno 2005 - prot. 2005015491_002

PARTE I

1.1 Programma di Ricerca afferente a

Area Scientifico Disciplinare 01: Scienze matematiche e informatiche 100%

1.2 Durata del Programma di Ricerca

24 Mesi

1.3 Coordinatore Scientifico del Programma di Ricerca

ROSSI **FRANCESCA** *frossi@math.unipd.it*

INF/01 - Informatica

Università degli Studi di PADOVA

Facoltà di SCIENZE MATEMATICHE FISICHE e NATURALI

Dipartimento di MATEMATICA PURA ED APPLICATA

1.4 Responsabile Scientifico dell'Unità di Ricerca

DELZANNO **GIORGIO**
Ricercatore confermato *06/12/1968* *DLZGRG68T06D969I*

INF/01 - Informatica

Università degli Studi di GENOVA

Facoltà di SCIENZE MATEMATICHE FISICHE e NATURALI

Dipartimento di INFORMATICA E SCIENZE DELL'INFORMAZIONE

010-3536638 *010-3536699* *giorgio@disi.unige.it*
(Prefisso e telefono) *(Numero fax)* *(Indirizzo posta elettronica)*

1.5 Curriculum scientifico del Responsabile Scientifico dell'Unità di Ricerca

Testo italiano

Nel marzo 1997 ha conseguito il dottorato in informatica presso l'Università di Pisa con supervisore Prof. Maurizio Martelli, e titolo della tesi: Logic and Object-oriented Programming in Linear Logic. Nell'ottobre 1997 ha ottenuto una borsa post-doc di 2 anni presso il Max-Planck-Institut fuer Informatik di Saarbruecken. Dal novembre 2000 e' ricercatore presso il Dip. di Informatica e Scienze dell'Informazione dell'Università di Genova. Dal Luglio 2002 e' idoneo per il ruolo di Professore Associato.

I suoi interessi di ricerca riguardano l'uso di programmazione con vincoli per l'analisi di sistemi concorrenti, le tecniche di verifica automatica basata su model checking, la combinazione di linguaggi e risolutori di vincoli e l'uso di logiche non standard per la modellazione di sistemi complessi.

*Nel 1999 ha ottenuto il premio Best Paper Award per l'articolo "Model Checking in CLP" alla conferenza ETAPS.
Nel 2000 ha ricevuto un supporto finanziario per svolgere ricerche sull'analisi di protocolli di coerenza per le cache dal Prof. J. Larus dal centro ricerche Microsoft di Redmond, USA.*

Attualmente ha piu' di 40 pubblicazioni in atti di conferenze e riviste internazionali.

E' stato visiting researcher presso:

*UPENN Philadelphia marzo-maggio 1995
LORIA Nancy ottobre-dicembre 1996
Bell Labs, Murray-Hill, NJ, USA, luglio-agosto 2000.
Microsoft Research, Seattle, USA, agosto-settembre 2000.
Universita della Provenza, Marsiglia, 1 mese 2004.*

Recentemente ha inoltre partecipato ai seguenti progetti:

*MURST 2001: Tecniche di Analisi e Verifica per Protocolli Utilizzati in Sistemi Concorrenti e Distribuiti
Finanziamenti a progetti di singoli e/o giovani ricercatori,
Universita' di Genova, Area 01, Scienze matematiche*

ULB 2001: Financial support from Universite' Libre de Bruxelles for research collaboration on infinite-state model checking, 2001.

*MURST 2002: Verifica di Protocolli di Sicurezza in Ambienti Multi-client
Finanziamenti a progetti di singoli e/o giovani ricercatori -
Universita' i Genova, Area 01, Scienze matematiche*

*Membro dell'unita' di Genova del progetto CoVER
"Verification of reactive systems" finanziato dal MIUR 2002-2004.*

*Membro dell'unita' di Genova del progetto
"Sintesi di procedure di decisione basate sulla deduzione
con applicazioni all'analisi formale e automatica di programmi"
finanziato dal MIUR 2003-2005.*

Membro dell'unita' di Bologna del progetto AIDA su "Abstract interpretation" finanziato dal MIUR 2005-2007.

Testo inglese

Research interests

- *Constraint-based Verification of Concurrent Systems*
- *Tools for the Analysis of Petri Nets*
- *Verification of Security, Consistency, and Communication Protocols*
- *Linear Logic and Logic Programming*
- *Higher-Order and Intuitionistic Logic Programming*

Education

*PhD in Computer Science, University of Pisa, Italy
Advisor: Prof. Maurizio Martelli
Title of the Dissertation: Logic and Object-oriented Programming in Linear Logic, March 1997
Researcher (Assistant Professor) in Computer Science, April 2000
Habilitation for Associate Professor, July 2002.*

Current position:

*Since April 2000, Researcher (Assistant Professor) at the
Dipartimento di Informatica e Scienze dell'Informazione,
Universita' di Genova, Italy.*

Previous Academic Employment

*Postdoctoral Scientist in Computer Science,
Max-Planck-Institut fuer Informatik Saarbruecken, Germany,
October 97-October 99.*

Research Associate, Dipartimento di Informatica e Scienze dell'Informazione,

Università di Genova, Italy, October 99-April 2000.

Visiting Positions

Visiting Fellow at UPENN Philadelphia, USA, Prof. D. Miller, March-May 1995.

Visiting Fellow at Loria Nancy, France, Prof. D. Galmiche, October-December 1996.

Visiting Researcher at University of Munich, Germany, Pr. J. Esparza, one week in June 1999.

Visiting Researcher at LSV-ENS Cachan, Paris, Dr. L. Fribourg, one week in July 1999.

Visiting Researcher at Université Libre de Bruxelles, Belgium, Prof. J.-F. Raskin, one week in Sept. 1999.

Visiting Researcher at UPENN Philadelphia, USA, Prof. R. Alur and Dr. S. La Torre, two weeks in July 2000.

Visiting Researcher at Bell Labs, Murray Hill, USA, Dr. K. Namjoshi, two weeks, August 2000.

Visiting Researcher at Microsoft Research, Redmond, USA, Prof. J. Larus and Dr. S. Rajamani, three weeks between August and September 2000.

Visiting Researcher at Université Libre de Bruxelles, Belgium, Prof. J.-F. Raskin, two weeks, March 2000, and two weeks, May 2001.

Visiting Researcher at the University of Uppsala, Sweden, Prof. P. A. Abdulla, two weeks, April 2001.

Visiting Researcher at the INRIA-LORRAINE, Nancy, France, Prof. M. Rusinowitch, two weeks, May 2002.

Visiting Researcher at the CMI Università de Provence, Marseille, France, Prof. R. Amadio one week novembre 2003.

Main Projects

MURST 2001: Tecniche di Analisi e Verifica per Protocolli Utilizzati in Sistemi Concorrenti e Distribuiti Individual project founded by the Italian Minister for Education, University and Research, 2001.

MSR 2001: Financial support from Microsoft Research, Redmond, Seattle, USA, for research on constraint-based verification, 2001.

ULB 2001: Financial support from Université Libre de Bruxelles for research collaboration on infinite-state model checking, 2001.

MURST 2002: Verifica di Protocolli di Sicurezza in Ambienti Multi-client Individual project founded by the Italian Minister for Education, University and Research, 2002.

Member of the Unit of Genova of the National project CoVER on verification of reactive systems founded by the Italian Minister for Education, University and Research, 2002-2004.

Member of the Unit of Genova of the National project Sintesi di procedure di decisione basate sulla deduzione con applicazioni all'analisi formale e automatica di programmi. founded by the Italian Minister for Education, University and Research, 2003-2005.

Member of the Unit of Bologna of the National project AIDA on abstract interpretation founded by the Italian Minister for Education, University and Research, 2005-2007.

1.6 Pubblicazioni scientifiche più significative del Responsabile Scientifico dell'Unità di Ricerca

1. *BOZZANO M., DELZANNO G., MARTELLI M. (2004). Model Checking Linear Logic Specifications. THEORY AND PRACTICE OF LOGIC PROGRAMMING. vol. 4(5-6) pp. 573-619 ISSN: 1471-0684*
2. *DELZANNO G. (2004). A Symbolic Procedure for Control Reachability in the Asynchronous Pi-calculus. ELECTRONIC NOTES IN THEORETICAL COMPUTER SCIENCE. vol. 98 pp. 21-33 ISSN: 1571-0661*
3. *DELZANNO G., RASKIN J.-F., VAN BEGIN L. (2004). Covering sharing trees: a compact data structure for parameterized verification. INTERNATIONAL JOURNAL ON SOFTWARE TOOLS FOR TECHNOLOGY TRANSFER. vol. 5(2-3) pp. 268-297 ISSN: 1433-2779*
4. *DELZANNO G. (2003). Constraint-Based Verification of Parameterized Cache Coherence Protocols. FORMAL METHODS IN SYSTEM DESIGN. vol. 23(3) pp. 257-301 ISSN: 0925-9856*
5. *DELZANNO G. (2002). An Overview of MSR(C): A CLP-based Framework for the Symbolic Verification of Parameterized*

Concurrent Systems. ELECTRONIC NOTES IN THEORETICAL COMPUTER SCIENCE. vol. 76 ISSN: 1571-0661

6. *DELZANNO G., DELZANNO G., MARTELLI M. (2002). An Effective Fixpoint Semantics for Linear Logic Programs. THEORY AND PRACTICE OF LOGIC PROGRAMMING. vol. 2(1) pp. 85-122 ISSN: 1471-0684*
7. *DELZANNO G. (2001). An Assertional Language for the Verification of Systems Parametric in Several Dimensions. ELECTRONIC NOTES IN THEORETICAL COMPUTER SCIENCE. vol. 50(4) ISSN: 1571-0661*
8. *DELZANNO G., MARTELLI M. (2001). Proofs as Computations in Linear Logic. THEORETICAL COMPUTER SCIENCE. vol. 258(1-2) pp. 269--297 ISSN: 0304-3975*
9. *DELZANNO G., PODELSKI A. (2001). Constraint-based Deductive Model Checking. INTERNATIONAL JOURNAL ON SOFTWARE TOOLS FOR TECHNOLOGY TRANSFER. vol. 3(3) pp. 250-270 ISSN: 1433-2779*
10. *BUGLIESI M., DELZANNO G., LIQUORI L., MARTELLI M. (2000). Object Calculi in Linear Logic. JOURNAL OF LOGIC AND COMPUTATION. vol. 10(1) ISSN: 0955-792X*
11. *DELZANNO G., GALMICHE D., MARTELLI M. (1999). A Specification Logic for Concurrent Object-Oriented Programming. MATHEMATICAL STRUCTURES IN COMPUTER SCIENCE. vol. 9(3) pp. 253-286 ISSN: 0960-1295*

1.7 Risorse umane impegnabili nel Programma dell'Unità di Ricerca

1.7.1 Personale universitario dell'Università sede dell'Unità di Ricerca

Personale docente

n°	Cognome	Nome	Dipartimento	Qualifica	Settore Disc.	Mesi Uomo	
						1° anno	2° anno
1.	DELZANNO	Giorgio	Dip. INFORMATICA E SCIENZE DELL'INFORMAZIONE	Ricercatore Universitario	INF/01	6	6
2.	COSTA	Gerardo	Dip. INFORMATICA E SCIENZE DELL'INFORMAZIONE	Prof. Ordinario	INF/01	4	4
TOTALE						10	10

Altro personale

n°	Cognome	Nome	Dipartimento	Qualifica	Mesi Uomo		
					1° anno	2° anno	
1.	Cordi'	valentina	Dip. INFORMATICA E SCIENZE DELL'INFORMAZIONE	Dottorando senza borsa	4	4	
TOTALE						4	4

1.7.2 Personale universitario di altre Università

Personale docente

Nessuno

Altro personale

Nessuno

1.7.3 Titolari di assegni di ricerca

Nessuno

1.7.4 Titolari di borse

n°	Cognome	Nome	Dipartimento	Anno di inizio borsa	Durata (in anni)	Tipologia	Mesi Uomo	
							1° anno	2° anno
1.	Ganty	Pierre	Dip. INFORMATICA E SCIENZE DELL'INFORMAZIONE	2003	3	Dottorato	4	4
2.	Mantovani	Jacopo	Dip. INFORMATICA, SISTEMISTICA E TELEMATICA	2004	3	Dottorato	4	4
3.	Montagna	Roberto	Dip. INFORMATICA E SCIENZE DELL'INFORMAZIONE	2005	3	Dottorato	4	4
TOTALE							12	12

1.7.5 Personale a contratto da destinare a questo specifico programma

n°	Qualifica	Costo previsto	Mesi Uomo		Note
			1° anno	2° anno	
1.	Assegnista	18.000	11		
TOTALE		18.000	11	0	

1.7.6 Personale extrauniversitario indipendente o dipendente da altri Enti

Nessuno

PARTE II

2.1 Titolo specifico del programma svolto dall'Unità di Ricerca

Testo italiano

Vincoli per Analisi di Sistemi Concorrenti

Testo inglese

Constraints for the Analysis of Concurrent Systems

2.2 Settori scientifico-disciplinari interessati dal Programma di Ricerca

INF/01 - Informatica

2.3 Parole chiave

Testo italiano

VERIFICA DI SISTEMI INFINITE-STATE BASATA SU VINCOLI ; VALIDAZIONE DI SISTEMI CONCORRENTI COME RISOLUZIONE DI VINCOLI ; ANALISI DI PROTOCOLLI BASATA SU VINCOLI DI PROTOCOLLI ; CONSTRAINT-BASED MODEL CHECKING

Testo inglese

CONSTRAINT-BASED VERIFICATION OF INFINITE-STATE SYSTEMS ; VALIDATION OF CONCURRENT SYSTEMS AS CONSTRAINT SOLVING ; CONSTRAINT-BASED ANALYSIS OF PROTOCOLS ; CONSTRAINT-BASED MODEL CHECKING

2.4 Base di partenza scientifica nazionale o internazionale

Testo italiano

Verifica Automatica basata su Model Checking

Logica Temporale e Model Checking

La tecnica di verifica Model Checking è stata introdotta nel 1981 da E. Clarke e E. A. Emerson della Carnegie Mellon University [CE81] come alternativa alle tecniche di simulazione fino ad allora utilizzate per convalidare hardware design. In questo ambito la specifica delle proprietà di un sistema viene data attraverso il linguaggio della logica temporale [Pnu81], mentre il modello della logica sottostante al funzionamento dei circuiti hardware e dei relativi protocolli di comunicazione viene descritto attraverso un sistemi di transizione. Una procedura di decisione efficiente [CE81] viene utilizzata per determinare se il sistema di transizione soddisfa o meno la specifica (descritta in logica temporale) fornita dal designer, fornendo controesempi che possono essere utilizzati per identificare errori nella progettazione. Il vantaggio principale di questa tecnica rispetto a tecniche di simulazione o theorem-proving è il fatto di essere completamente automatica (push-button technology). La procedura di model checking termina con successo se il sistema verifica la specifica, altrimenti restituisce una possibile esecuzione che viola uno dei suoi requisiti. essendo basata sull'esplorazione dello spazio degli stati di un sistema, il maggior problema di questa tecnica è la scalabilità rispetto alla descrizione del sistema da verificare. In particolar modo questo problema, chiamato state explosion, si verifica quando il sistema è formato da più componenti che lavorano in concorrenza. Recentemente, l'applicazione di strutture dati dedicate per la rappresentazione di sistemi di transizione e insiemi di stati (BDD [Bry86]) ha permesso di estendere l'applicazione di tecniche di questo tipo a sistemi di grandi dimensioni [BCL91]. Questa idea ha dato origine alla tecnica cosiddetta di symbolic model checking [BCB+90,McM93]. Altre tecniche quali le riduzioni con ordini parziali, simmetrie o le tecniche di astrazione hanno ulteriormente migliorato i tool basati su model checking [CGL94].

Sistemi con Spazio degli Stati Infinito

Mentre i sistemi originariamente considerati da Clarke erano sistemi hardware e quindi intrinsecamente finiti, gli sforzi maggiori degli ultimi anni sono stati indirizzati verso lo sviluppo di modelli e algoritmi di verifica per sistemi con spazio degli stati potenzialmente infinito (sistemi infinite-state). Alcuni esempi di sistemi per i quali sono state sviluppate tecniche automatiche di verifica sono:

- Sistemi di controllo real-time [AD94,AH92,BLP+99,HNSY94,LPY97,Yov97];*
- Protocolli di sicurezza [CDL+99,MCJ97];*
- Protocolli di comunicazione con canali FIFO [ACJT96,BW98];*
- Sistemi ibridi [ACH+95,Hen96,HHW97,HM00,HHM+00];*
- Sistemi parametrici sul numero di processi[AJ03,ABJN99,DEP99,RKS+00].*

Alcune delle tecniche citate in precedenza (più precisamente le tecniche basate su Automi Temporizzati [AD94]) hanno trovato applicazione per la verifica di protocolli utilizzati a livello industriale quali il Philips audio protocol (es. [LPY9]). Un'altra categoria interessante di sistemi infinite-state e' formata dai sistemi parametrici. Numerosi protocolli di comunicazione e consistenza dei dati utilizzati nei sistemi distribuiti (ad. databases distribuiti, sistemi multiprocessore, Internet) vengono assunti

corretti indipendentemente dal numero di agenti coinvolti nel ciclo di vita del sistema. Un altro esempio interessante di sistemi concorrenti e distribuiti parametrici sono i protocolli utilizzati per fornire servizi in architetture multi-client e multi-server, ad es., Web Services [BXS04,Hull04]. Questo tipo di applicazioni presenta una serie di caratteristiche quali la concorrenza, la mobilità del codice, la presenza di strutture dati complesse (XML) che rendono il problema dell'analisi molto difficile e ricco di spunti per ricerca di base e applicativa. Inoltre è interessante notare che, in generale, i sistemi parametrici richiedono linguaggi e modelli sofisticati già nella fase di modellazione di un sistema.

Verifica Automatica basata su Vincoli

Le idee principali sottostanti le tecniche di model checking possono essere estese direttamente al caso di sistemi infiniti. Ovviamente, mentre nel caso finito il problema della verifica di una proprietà temporale (fissato un modello) è decidibile, nel caso (generico) di sistemi infiniti può diventare indecidibile. Come nel caso del ragionamento automatico per logica del primo ordine, i risultati generali di indecidibilità non hanno scoraggiato lo sviluppo di sistemi di verifica per sistemi infinite-state. Alcuni esempi sono: Kronos [Yov97] e Uppaal [LPY97] per sistemi real-time, o HyTech [HHW97] per sistemi ibridi. Esistono inoltre varie classi di sistemi infiniti per cui è possibile decidere alcune proprietà della logica temporale ed in particolare proprietà esprimibile tramite la raggiungibilità di un certo insieme di stati potenzialmente pericolosi (safety properties), ad esempio, timed e rectangular automata [AD94,HM00] e lossy FIFO systems [ACJT96,ABJN99].

In molte di queste estensioni il concetto di base è la rappresentazione di insiemi di stati tramite linguaggi con vincoli. Quest'idea non è sorprendente in quanto i vincoli non sono altro che formule (ad esempio in logica del primo ordine) interpretate su un dominio prefissato e per le quali esistono algoritmi ed euristiche specializzate per il test di soddisfacibilità, l'eliminazione di variabili (quantificazione esistenziale) e il confronto di formule. La generalizzazione delle estensioni del model checking a linguaggi con vincoli ha dato vita al paradigma chiamato constraint-based model checking. In questo ambito i vincoli vengono utilizzati come linguaggio asserzionale [ACJT96,KMMPS01] per rappresentare collezioni infinite di configurazioni di un sistema concorrente. Le operazioni sui vincoli sono utilizzate quindi all'interno degli algoritmi basati su computazioni di punto fisso sui quali si basa la tecnica di model checking. Per spiegare meglio questa idea, notiamo innanzitutto che il modello utilizzato per rappresentare sistemi finiti può essere esteso in modo naturale a sistemi infiniti generalizzando il tipo di funzioni booleane utilizzate per rappresentare le transizioni. In altre parole in questo ambito si rappresenta la relazione di transizione del sistema tramite una formula nel linguaggio di vincoli considerato. Ad esempio, la formula $T(x,x') = x >= 1 \ \& \ x' = x + 1$ può essere usata per rappresentare l'aggiornamento di una variabile intera tramite un vincolo dell'aritmetica lineare. In modo simile si possono rappresentare insiemi infiniti di possibili stati della variabile x tramite vincoli quali $x >= 2$. Gli operatori di manipolazione dei vincoli possono essere applicati quindi per l'analisi del sistema di partenza, ad esempio, per calcolare, in modo simbolico, l'insieme delle precondizioni che portano il sistema a stati che soddisfano la condizione $x >= 2$. Negli ultimi anni sono state trovate numerose applicazioni per le tecniche di verifica basate su vincoli. In particolare, possiamo dividere i sistemi studiati in due classi principali: sistemi con un numero finito di componenti ma con un'insieme di valori per i dati potenzialmente infinito, e sistemi con un numero finito ma non fissato a priori di componenti (parameterized verification). Alcuni esempi della prima classe di applicazione sono: algoritmi di mutua-esclusione (e.g. Bakery and Ticket) [BGP99,DP99]; il protocollo di Fisher e il protocollo Bounded Retransmission [HHW97,LPY97] e il protocollo Alternating Bit [ACJT96]. Alcuni esempi della seconda classe sono: Unbounded Petri Nets [ACJT96,DRV01], Unbounded Timed Petri Nets con token etichettati con un time-stamp [AJ99], schemi parametrici (sul numero di processi) di programmi concorrenti [BD02], protocolli di consistenza e coerenza per sistemi distribuiti e multiprocessore [Del00,Del00b,DB01], e programmi Java multithreaded [DRV02]. In molte di queste applicazioni la tecnologia basata su vincoli viene combinata con astrazioni definite sia al livello della descrizione del sistema [Del00,DB01,DRV02] che al livello della rappresentazione simbolica delle configurazioni [BD02].

Una delle direzioni più interessanti per la ricerca nel campo dell'applicazione dei vincoli all'analisi di sistemi infiniti riguarda l'analisi di sistemi con tipi di dato eterogenei e di sistemi aperti quali i web services. In questo ambito sembra interessante studiare la combinazione di vincoli con tecniche di composizionalità (per poter scomporre lo studio di un sistema complesso in sottoproblemi) e con teoria dei giochi (per studiare l'interazione tra un agente generico e l'ambiente). Simili tecniche sono state recentemente applicate alla verifica di sistemi finiti [CLM89,AHK02], mentre la loro applicazione a sistemi infiniti è ancora in fase embrionale (vedi ad es. [ABO03,BXS04]).

Testo inglese

Automatic Verification based on Model Checking

Temporal Logic and Model Checking

The automatic technique known as Model checking has been introduced in 1981 by E. Clarke e E. A. Emerson [CE81] as an alternative to simulation techniques for validation of hardware design. In this setting the specification of the system properties is given in temporal logic [Pnu81]. Furthermore, a model of the logic of a circuit and of related communication protocols is described using a transition system. An efficient decision procedure [CE81] is used then to check the model against the temporal properties. The procedure might return counterexamples that could be useful to isolate bugs in the original design. The main advantage of this technique over simulation and theorem proving is that it fully automatic. The model checking algorithm terminates successfully if the model satisfies the temporal property, otherwise it returns a counterexample that violates it. Being based on an exhaustive exploration of the state space, the major problem of model checking however is scalability. This problem, known as state explosion, especially occurs when the system has several components working in parallel. Recently, however, the application of dedicated data structures for representing sets of states (the BDDs of [Bry86]) has enlarged the spectrum of the concurrent systems to which model checking can be applied [BCL91]. Symbolic model checking [BCB+90,McM93] implements this technique. Other techniques like partial order and symmetries reductions, and abstractions has been applied in order to further improve this verification method [CGL94].

Infinite-state Systems

The first applications of model checking have been studied for the logical models of hardware systems, i.e., finite-state systems. In recent years may effort have been spent to extended (symbolic) model checking to system with an infinite-state space. Examples of systems and related research studies are:

-Real time systems [AD94,AH92,BLP+99,HNSY94,LPY97,Yov97];

- Security protocols [CDL+99,MCJ97];
- Communication protocols with FIFO channels [ACJT96,BW98];
- Hybrid systems [ACH+95,Hen96,HHW97,HM00,HMM+00];
- Systems parametric in the number of processes [AJ03,ABJN99,DEP99,RKS+00].

Some of the techniques used to deal with these kind of systems (e.g. those based on Timed Automata) found real applications for the Philips audio protocol (es. [LPY9]). Parametric systems are as interesting as timed systems. Several communication and data consistency protocols used in distributed systems (e.g distributed databases, multiprocessors systems, internet protocols) are developed for open systems in which an potentially unbounded number of agents interacts by means of them. Other interesting examples of parametric concurrent and distributed systems comes from the world of Web Services [BXS04,Hull04]. This applications have features like concurrency, mobility of code, complex data structures (es. XML) that make verification a hard task in practice. Furthermore, the verification of a parametric systems may require sophisticated models and languages already in the specification phase.

Constraint-based Verification

The main ideas underlying model checking can naturally be extended (at least in principle) to systems with an infinite-state space. Clearly, the verification problem of a temporal property might become undecidable for an arbitrary extension of the model. As for theorem proving for first order logic, general undecidability problems have not discouraged the development of specialized verification tools for systems with unbounded state-space. Examples are: Kronos [Yov97] and Uppaal [LPY97] for real-time systems, and HyTech [HHW97] for hybrid systems. Furthermore, there exists several classes of infinite-state systems for which it is possible to decide temporal properties., For instance, timed and rectangular automata [AD94,HM00] and lossy FIFO systems [ACJT96,ABJN99].

In most of the extensions of model checking to infinite-state systems, the key concept is the use of constraints as a symbolic representation of infinite sets of states. This is not surprising since constraints are nothing but formulas (typically of first order logic) interpreted over a fixed domain and equipped with decision procedures and heuristics that implement operations like the satisfiability test and variable elimination (existential quantification). The generalization of the extensions of model checking based on constraints has given origin to the paradigm called constraint-based model checking [ACJT96,KMMP01]. In this setting constraints are used as an assertional language to represent infinite collections of system configurations. Constraint operations are then used to symbolically manipulate sets of states within the fixpoint based algorithms underlying model checking. To illustrate, we can naturally extend the finite-state models used in model checking by a generalization of the Boolean functions used to represent the transition system: In the extended model we can use a constraint to represent a transition relation. As an example, the linear constraint $T(x,x') = x \geq 1 \ \& \ x' = x + 1$ can be used to represent the update of an integer variable. In a similar way we can use a constraint as $x \geq 2$ to represent an infinite set of possible values of variable x . Constraint manipulation operators can be used then to perform an analysis of the original system, e.g., to compute all preconditions for the system to reach a state with $x \geq 2$.

In the last years constraint-based verification has found several practical applications. These applications can be divided in two categories: systems with a bounded number of components but unbounded data, and systems with an unbounded number of components. Examples of applications in the first class are: mutual exclusion protocols (e.g. Bakery and Ticket) [BGP99,DP99]; the Fisher and Bounded Retransmission protocols [HHW97,LPY97], and the Alternating Bit protocol [ACJT96]. Examples in the second class are: Unbounded Petri Nets [ACJT96,DRV01], Unbounded Timed Petri Nets with aged tokens [AJ99], Parametric schemes of protocols [BD02], coherence and consistency protocols [Del00,Del00b,DB01], Java multithreaded programs [DRV02]. In most of these applications constraint-based verification is combined with abstractions defined both at the specification level [Del00,DB01,DRV02] and at the level of symbolic representation of sets of states [BD02].

One of the most interesting research directions for the applications of constraints to verification of infinite-state systems concerns the study of systems with heterogenous data and working in open environments. In this setting it seems important to study the combination of constraints with compositional verification techniques (to compose/decompose the analysis) and with verification techniques based on game theory (to study the interaction of a generic agent with the environment). Although similar techniques have been applied to the finite-state case (see e.g. [CLM89,AHK98]), their application to infinite-state systems is still in a embryonal phase (see e.g. [ABO03,BXS04]).

2.4.a Riferimenti bibliografici

ABO03

P. Abdulla, A. Bouajjani, J. d'Orso
Deciding Monotonic Games.
CSL 2003: 1-14

ABJN99

P. A. Abdulla, A. Bouajjani, B. Jonsson, and M. Nilsson.
Handling Global Conditions in Parameterized System Verification.
CAV 99, 1999.
ACJT96
P. A. Abdulla, K. Cerans, B. Jonsson, and Y.-K. Tsay.
General Decidability Theorems for Infinite-state Systems.
LICS 96, 1996.

AJ03

P. Abdulla, B. Jonsson
Model checking of systems with many identical timed processes.

TCS 290(1): 241-264,2003

ACH+95

R. Alur, C. Courcoubetis, T. A. Henzinger, N. Halbwachs,
P. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine.
The algorithmic analysis of hybrid systems.
TCS 138:3-34, 1995.

AD94

R. Alur and D.L. Dill.
A theory of timed automata.
TCS 126:183-235, 1994.

AH92

R. Alur and T. A. Henzinger.
Logics and models of real time: a survey.
Real Time: Theory in Practice, 1992.

AHK02

R. Alur, T. A. Henzinger, and O. Kupferman
Alternating-time Temporal Logic
J. ACM 49:672-713, 2002.

AJ98

P. A. Abdulla and B. Jonsson.
Verifying Networks of Timed Processes.
TACAS 98, 1998.

BLP+99

G. Behrmann, K. G. Larsen, J. Pearson, C. Weise, and W. Yi.
*Efficient Timed Reachability Analysis Using Clock
Difference Diagrams.*
CAV 99, 1999.

BM99

A. Bouajjani and R. Mayr.
Model Checking Lossy Vector Addition Systems.
STACS 99, 1999.

Bry86

R. E. Bryant.
Graph-based Algorithms for Boolean Function Manipulation.
IEEE Transaction on Computers, C-35(8):667-691, 1986.

BXS04

T. Bultan, X. Fu, J. Su.
Tools for Automated Verification of Web Services.
ATVA 2004: 8-10

BCB+90

J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill,
and J. Hwang.
Symbolic Model Checking: States and Beyond.
LICS 90, 1990.

BCL91

J. R. Burch, E. M. Clarke, and D. E. Long.
Symbolic Model Checking with partitioned transition relations.
VLSI 91, 1991.

BGP99

T. Bultan, R. Gerber, and W. Pugh.
*Model-checking concurrent systems with unbounded
integer variables: Symbolic representations, approximations,
and experimental results.*
ACM TOPLAS, 21(4):747--789, 1999.

CE81

E. M. Clarke and E. A. Emerson.
*Synthesis of synchronization skeletons for branching
time temporal logic.*

Ws.Logic of Programs, 1981.

CES86

*E. M. Clarke and E. A. Emerson and A. P. Sistla.
Automatic verification of finite-state concurrent
systems using temporal logic specifications.
TOPLAS, 8(2):244- 263, 1986.*

CDL+99

*I. Cervesato, N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov.
A Meta-Notation for Protocol Analysis.
CSFW 99, 1999.*

CGL94

*E. Clarke, O. Grumberg and D. Long.
Verification tools for finite-state concurrent systems.
A Decade of concurrency-Reflections and Perspectives.
LNCS 803, 1994.*

CGH+93

*E. M. Clarke, O. Grumberg, H. Hiraishi, S. Jha,
D.E. Long, K.L. McMillan, and L.A. Ness.
Verification of the Futurebus+cache coherence protocol.
CHDLA '93, 1993.*

CLM89

*E.M. Clarke, D. E. Long, K. L. McMillan.
Compositional Model Checking.
LICS 89, 1989*

CR97

*G. Costa, G. Reggio
Specification of Abstract Dynamic-Data Types:
A Temporal Logic Approach.
TCS 173(2): 513-554 (1997)*

CS84

*G. Costa, C. Stirling
A Fair Calculus of Communicating Systems.
Acta Inf. 21: 417-441 (1984)*

DEP99

*G. Delzanno, J. Esparza, and A. Podelski.
Constraint-based Analysis of Broadcast Protocols.
CSL 99, 1999.*

DP99

*G. Delzanno, A. Podelski.
Model Checking in CLP.
TACAS 99, 1999.*

Del00

*G. Delzanno.
Automatic Verification of Parameterized Cache Coherence Protocols.
CAV 00, 2000.*

Del00b

*G. Delzanno.
Verification of Consistency Protocols via Infinite-State
Symbolic Model Checking - A Case Study -
Forte/PSTV 00, 2000.*

DR00

*G. Delzanno and J. F. Raskin.
Symbolic Representation of Upward-Closed Sets.
TACAS 00, 2000.*

Del01

*G. Delzanno.
An assertional language for systems parametric in
several dimensions (Preliminary results).*

VEPAS 01, 2001.

DRV01

G. Delzanno, J.-F. Raskin, and L. Van Begin.
Attacking Symbolic State Explosion.
CAV 01, 2001.

DB01

G. Delzanno and T. Bultan.
Constraint-based Verification of Client-Server Protocols.
CP 01, 2001.

BD02

M. Bozzano, G. Delzanno.
Algorithmic Verification of Invalidation-Based Protocols.
CAV 02: 295-308

Del02

G. Delzanno.
*Constraint-Based Model Checking for
Parameterized Synchronous Systems.*
FroCos 02: 72-86

BD02

M. Bozzano, G. Delzanno.
Beyond Parameterized Verification.
TACAS 02: 221-235

Del02

G. Delzanno: *An Overview of MSR(C): A CLP-based Framework
for the Symbolic Verification of Parameterized Concurrent
Systems.*
ENTCS 76: (2002)

Del03

G. Delzanno
*Constraint-Based Verification of Parameterized
Cache Coherence Protocols.*
FMSD 23(3): 257-301 (2003)

Del04

G. Delzanno: *A Symbolic Procedure for Control Reachability in
the Asynchronous Pi-calculus: Extended Abstract.*
ENTCS 98: 21-33 (2004)

Hen96

T. A. Henzinger.
The theory of hybrid automata.
LICS 96, 1996.

HHM+00

T. A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi.
*Beyond HyTech: Hybrid System Analysis Using Interval
Numerical Methods.*
HSCC 00, 2000.

HHW97

T. A. Henzinger, P.-H. Ho, and H. Wong-Toi.
HYTECH: a Model Checker for Hybrid Systems.
CAV 97, 1997.

HM00

T. A. Henzinger, R. Majumdar.
Symbolic Model Checking for Rectangular Hybrid Systems.
TACAS 2000, 2000.

HNSY94

T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine.
Symbolic model checking for real-time systems.
Information and Computation 111:193-244, 1994.

Hull04

Web Services Composition: A Story of Models, Automata, and Logics, a tutorial for the 2004 EDBT Summer School.
<http://edbtss04.dia.uniroma3.it/Hull.pdf>

KMMPS01

Y. Kesten, O. Maler, M. Marcus, A. Pnueli, E. Shahar
Symbolic model checking with rich assertional languages.
TCS 256(1-2): 93-112 (2001)

LPY95

K. G. Larsen, P. Pettersson and W. Yi.
Diagnostic Model-Checking for Real-Time Systems.
DIMACS Ws VCHS 95, 1995.

LPY97

K. G. Larsen, P. Pettersson, and W. Yi.
UPPAAL in a Nutshell.
STTT, 1(1-2):134-152, 1997.

MCJ97

W. Marrero, E. M. Clarke, and S. Jha.
Model Checking for Security Protocols.
Technical Report CMU-SCS-97-139,
CMU, 1997.

McM93

K. L. McMillan.
Symbolic Model Checking: An Approach to the State Explosion Problem.
Kluwer Academic, 1993.

Pnu81

A. Pnueli.
A temporal logic of concurrent programs.
TCS 13:45-60, 1981.

RKS+00

A. Roychoudhury, K.N. Kumar, C.R. Ramakrishnan,
I.V. Ramakrishnan and S. A. Smolka.
Verification of Parameterized Systems Using
Logic-Program Transformations.
TACAS 2000, 2000.

Yov97

S. Yovine.
Kronos: A verification tool for real-time systems.
STTT, 1(1-2), 1997.

Web

Web Services Activity: <http://www.w3.org/2002/ws/>

2.5 Descrizione del programma e dei compiti dell'Unità di Ricerca

Testo italiano

Descrizione dell'unità'

L'unità' e' formata da specialisti in applicazione di vincoli a problemi di concorrenza (Delzanno) e in teoria della concorrenza e logiche temporali (Costa). Inoltre consta di 4 dottorandi: Montagna e' attualmente impegnato in ricerca su estensioni di logiche temporali basate su giochi, Cordi' su sistemi multi-agente, Ganty e Mantovani su analisi di sistemi basata su risolutori di vincoli proposizionali. La ricerca verra' svolta sia in collaborazione con altre unita' del progetto che con membri del DISI che del DIST (Dip. di Informatica Sistemistica e Telematica dell'Univ. di Genova) che non hanno potuto partecipare formalmente a questo progetto essendo impegnati in progetti PRIN partiti nel 2005. Inoltre si prevede un assegno di un anno per un giovane ricercatore.

Descrizione del programma di ricerca

La ricerca è mirata all'utilizzo di linguaggi e sistemi di programmazione con vincoli per la modellazione e l'analisi di sistemi

concorrenti con spazio degli stati infinito.

La metodologia che verrà seguita si basa su un paradigma nel quale i vincoli sono usati come modello intermedio per rappresentare problemi di analisi di diversi tipi di sistemi concorrenti. Il vantaggio di tale rappresentazione consiste nel poter utilizzare tecniche di risoluzione (eventualmente approssimate) di vincoli per studiare i problemi definiti sui modelli di partenza.

Il paradigma risultante permette quindi di implementare un possibile trasferimento di tecnologia da sistemi con vincoli ad tool per l'analisi di sistemi concorrenti.

In lavoro precedenti sono stati utilizzati con successo diverse istanze di tale paradigma. Ad es. in [Del00,Del03] attraverso la codifica di modelli a macchine a stati in problemi di vincoli su interi è stato possibile utilizzare tecniche di rilassamento dei vincoli lineari e risolutori di vincoli su reali per verificare in modo automatico proprietà di mutua esclusione di protocolli di coerenze delle cache utilizzate in sistemi multiprocessore.

Partendo da questa idea, verranno affrontate le seguenti tematiche di ricerca:

Tematica 1: Combinazione di Teorie

(Partecipanti: Delzanno - Cordi' - Montagna - Mantovani - Ganty)

Questa ricerca è mirata all'estensione del paradigma illustrato precedentemente in modo da combinare tra loro diversi risolutori di vincoli (cioè le relative procedure di decisione) per poter gestire sistemi concorrenti con strutture dati eterogenee. In questo senso sembra di particolare interesse lo studio della combinazione di linguaggi per vincoli (ad es. vincoli lineari) e linguaggi logici basati su multiset rewriting. Le logiche basate su multiset rewriting sono particolarmente adatte infatti per modellare processi concorrenti [Del01,BD02]. Ad es. la regola

$send(m) \mid rec(m) \rightarrow end1 \mid end2$

dove $|$ è un costruttore di multinsiemi modella in modo naturale l'interazione tra un processo che invia un messaggio "m" e un processo pronto a ricevere "m", che porta alla terminazione di entrambi. La combinazione con vincoli permette di esprimere in modo naturale condizioni sui dati utilizzati dai processi.

Ad es. la regola

$send(x) \mid rec(y) \rightarrow end1 \mid end2 : x > y$

introduce condizioni sui dati (in questo caso un vincolo aritmetico) come guardia per la sincronizzazione dei processi. In questa tematica i sotto-obiettivi sono:

1.1) Studiare l'espressività di linguaggi risultanti da diverse combinazioni di multiset rewriting e vincoli.

1.2) Studiare tecniche di analisi basate su risolutori di vincoli per linguaggi che combinano multiset rewriting e vincoli

1.3) Studiare codifiche di diversi modelli e linguaggi per la concorrenza (con particolare interesse per sistemi mobili e multi agente) in multiset rewriting con vincoli.

Tematica 2: Vincoli per Analisi Compositiva

(Delzanno - Ganty - Mantovani)

Un aspetto interessante per quanto riguarda la ricerca nel campo della verifica di sistemi concorrenti riguarda lo sviluppo di tecniche di prova composizionali. In questo ambito l'obiettivo è di raffinare il paradigma illustrato precedentemente in modo da rendere composizionale sia la codifica dei sistemi e problemi iniziali che la fase di analisi. In particolare la ricerca sarà focalizzata sui seguenti punti:

2.1) Codifica composizionale

Partendo da problemi di verifica per modelli definiti tramite operatori di composizione (ad es. process algebra) si vogliono studiare codifiche in problemi di vincoli che siano a loro volta definite in modo composizionale.

2.2) Analisi composizionale

Partendo dai vincoli risultanti da codifiche che soddisfano i requisiti del punto (2.1) si vogliono studiare sistemi di prova (semi) automatici basati sui sottostanti risolutori di vincoli per poter realizzare tool di analisi composizionale.

Tematica 3: Vincoli per Logiche basate sui Giochi

(Delzanno - Montagna - Costa)

La tecnica di model checking e' stata recentemente estesa con alcune nozioni provenienti dalla teoria dei giochi per poter modellare proprieta' di sistemi che interagiscono con l'ambiente, ad es. la logica ATL [AHK02]. In queste logiche si possono esprimere proprieta' legate alla cooperazione di agenti. Sulla base di risultati recenti [ABO03, in questa tematica la ricerca sara' focalizzata sui seguenti punti:

3.1) Studio dell'estensione di logiche come ATL per modellazione di sistemi con spazio degli stati infinito

3.2) Studio di tecniche simboliche basate su vincoli per il model checking di logiche basate sui giochi

Tematica 4: Applicazioni

(Tutta l'unita')

Il target della ricerca proposta nelle 3 tematiche precedenti e' l'applicazione della combinazione di tecniche basate su vincoli e multiset rewriting, su composizionalità e su possibili estensioni del linguaggio di specifica per le proprieta' tramite logiche basate sui giochi all'analisi di sistemi con spazio degli stati infinito.

In particolare utilizzeremo come casi di studio i protocolli utilizzati per i web services [BXS04, Hull04, Web]. Questo tipo di applicazioni presenta caratteristiche quali concorrenza e distribuzione, mobilità del codice, e presenza di sessioni multiple che rendono l'analisi un problema interessante per l'applicazione di nuove tecnologie.

Descrizione dei compiti e collaborazioni con altre unita'

Il primo anno verranno affrontati i punti 1.1-2, 2.1, 3.1.

Il secondo anno verranno affrontati i punti 1.3, 2.2 e 3.2.

La tematica 4 verra' studiata nel corso di tutti e due gli anni (come spunto concreto per studiare modelli e problemi di analisi).

Le tematiche 1 e 2 verranno studiate in collaborazione con l'unita' di Pescara con punti di contatto con l'unita' di Padova per lo studio di risolutori di vincoli. La tematica 3 verra' studiata in collaborazione con l'unita' di Udine.

Testo inglese

Descriptions of Units and Duties

The unit is formed by specialists in constraint-based verification (Delzanno) and concurrency (Costa). Furthermore, 4 Phd students will participate to the research activities: Montagna is currently working on extensions of model checking based on game theory, Cordi on

multi-agent systems, and Ganty and Mantovani are working on analysis techniques based on solvers for propositional logic.

The research will be carried on in collaboration with the other units of the project and with members of our DISI and DIST (Dip. of Informatica, Sistemistica e Telematica of the Univ. of Genova) who could not formally participate to this proposal for constraints due to national projects started in 2005.

Furthermore, we plan to assign a one-year contract to a young researcher.

Research Program Description

Our general goal is the application of constraint programming languages and systems for the modelisation and analysis of infinite-state concurrent systems.

Our methodology is based on a paradigm in which constraints are used as an intermediate language to represent analysis and verification problems for different types of concurrent systems.

Thanks to the constraint-based intermediate representation, the problems formulated on the original models can be attacked by means of (general purpose/specialized/approximated) constraint solving techniques. This link might give rise to unexpected transfers of technology as it happened for the application of constraint solvers for linear arithmetics over the reals to the verification of communication and cache coherence protocols [DP99, DEP99, Del00, Del03].

Starting from this idea, in this project we will investigate the following lines of research.

Theme 1: Combination of Theories

(Participants: Delzanno - Cordi' - Montagna - Mantovani - Ganty)

This research is aimed at an extension of the aforementioned constraint-based verification paradigm in which different constraint solvers cooperate in order to handle concurrent systems with heterogeneous data. In this setting we are particularly interested in the combination of classical constraint languages (e.g. linear arithmetics) with logic languages used in multiset rewriting. Multiset rewriting can naturally be used to specify concurrent systems [Del01, BD02]. As an example, rewriting a rule like

$send(m) \mid rec(m) \rightarrow end1 \mid end2$

where $|$ is a multiset constructor, can be used to model the interaction of two processes, namely a sender and a receiver, which synchronize on message "m".

The combination with constraints allows us to naturally express conditions over the data locally used by single processes. As an example, the rule

$send(x) | rec(y) \dashrightarrow end1 | end2 : x > y$

introduces a condition on the message (a linear constraint) as a guard for process synchronization.

The specific goals in this line of research are:

1.1) To study the expressiveness of the languages resulting from the combination of multiset rewriting and constraints.

1.2) To study techniques of analysis based on constraint solvers for languages that combine multiset rewriting and constraints.

1.3) To study encoding of different models and languages used in concurrency (with particular attention to those used for modeling mobile and multi-agent systems) in multiset rewriting with constraints.

Tematica 2: Constraints for Compositional Analysis

(Delzanno - Ganty - Mantovani)

Compositional proof systems are an important aspect of verification of concurrent systems. Although there exist solid techniques for the finite-state case see e.g. [CLM89]), their extension to infinite-state systems is still under development.

Our goal is to introduce a form of compositional reasoning in the constraint-based verification paradigm.

More specifically, our sub-goals are:

2.1) Compositional Encoding

Starting from verification problems for systems defined via composition operators (e.g. process algebra) we will design constraint languages that could allow us to compositionally build the encoding from models to constraints by using composition operators working at the constraint level that are counterpart of those in the model description language. From preliminary investigations, constraint languages based on the combination of orders (to express the causal relation of events) and arrays or uninterpreted functions (to locally express the update to the global state of the system) seem adequate tool to achieve our purposes.

2.2) Compositional Analysis

Starting from the constraint languages defined in (2.1) we will study compositional proof systems that could be supported by the underlying constraint solvers.

Theme 3: Constraints for Game Logics

(Delzanno - Montagna - Costa)

Model checking has recently been extended with basic notions coming from game theory as in the ATL logic designed for finite-state systems [AHK02]. This extension provides temporal and strategic reasoning

and it can be used to model the cooperation of agents.

On the basis of some recent work [ABO03], our research will focus on the following subgoals:

3.1) To study extensions of logics like ATL for modeling infinite-state systems.

3.2) To study symbolic techniques based on constraints to design verification methods for the logics designed in 3.1.

Tematica 4: Applications

(The whole unit is involved)

The target of the research proposed in Theme 1, 2, and 3 is the application of the techniques based on the combination of constraints and multiset rewriting, on compositional proof systems, and on game theory to the verification of infinite-state systems. More specifically, we will focus our attention on protocols used to design web services [BXS04, Hull04, Web]. This kind of applications presents features like concurrency and distribution, code mobility, and multiple sessions/transactions that make the analysis an challenging problem for new technologies.

Descriptions of Duties and Collaboration with Other Units

In the first year we will investigate 1.1-2, 2.1, 3.1.

In the second year we will attack 1.3, 2.2 e 3.2.

The theme 4 will be investigated during the entire project.

Theme 1 and 2 will be studied in collaboration with the unit of Pescara with other possible links with Padova for the development of constraint solvers. Theme 3 will be studied in collaboration with Udine.

2.6 Descrizione delle attrezzature già disponibili ed utilizzabili per la ricerca proposta con valore patrimoniale superiore a 25.000 Euro

Testo italiano

Nessuna

Testo inglese

Nessuna

2.7 Descrizione delle Grandi attrezzature da acquisire (GA)

Testo italiano

Nessuna

Testo inglese

Nessuna

2.8 Mesi uomo complessivi dedicati al programma

Testo italiano

		Numero	Mesi uomo 1° anno	Mesi uomo 2° anno	Totale mesi uomo
<i>Personale universitario dell'Università sede dell'Unità di Ricerca</i>		3	14	14	28
<i>Personale universitario di altre Università</i>		0	0	0	0
<i>Titolari di assegni di ricerca</i>		0			
<i>Titolari di borse</i>	<i>Dottorato</i>	3	12	12	24
	<i>Post-dottorato</i>	0			
	<i>Scuola di Specializzazione</i>	0			
<i>Personale a contratto</i>	<i>Assegnisti</i>	1	11	0	11
	<i>Borsisti</i>	0			
	<i>Dottorandi</i>	0			
	<i>Altre tipologie</i>	0			
<i>Personale extrauniversitario</i>		0			
TOTALE		7	37	26	63

Testo inglese

		Numero	Mesi uomo 1° anno	Mesi uomo 2° anno	Totale mesi uomo
<i>University Personnel</i>		3	14	14	28
<i>Other University Personnel</i>		0	0	0	0
<i>Work contract (research grants, free lance contracts)</i>		0			
<i>PHD Fellows & PHD Students</i>	<i>PHD Students</i>	3	12	12	24
	<i>Post-Doctoral Fellows</i>	0			
	<i>Specialization School</i>	0			
<i>Personnel to be hired</i>	<i>Work contract (research grants, free lance contracts)</i>	1	11	0	11

	<i>PHD Fellows & PHD Students</i>	0			
	<i>PHD Students</i>	0			
	<i>Other tipologie</i>	0			
	<i>No cost Non University Personnel</i>	0			
TOTALE		7	37	26	63

PARTE III

3.1 Costo complessivo del Programma dell'Unità di Ricerca

Testo italiano

Voce di spesa	Spesa in Euro	Descrizione
Materiale inventariabile	6.000	Acquisto di PC per sviluppo e testing, stampanti ed attrezzature per gestione dati sperimentali
Grandi Attrezzature		Nessuna spesa preventivata
Materiale di consumo e funzionamento	5.000	Acquisto di materiale di cancelleria, e di supporto alla ricerca, spese per stampanti, telefoni e fax.
Spese per calcolo ed elaborazione dati	0	Nessuna spesa preventivata
Personale a contratto	18.000	Assegno di ricerca della durata di 12 mesi per lo sviluppo e il testing di procedure di model-checking basate su constraint solving
Servizi esterni	0	Nessuna spesa preventivata
Missioni	15.000	Partecipazione a meeting del progetto, e visite a laboratori nazionali ed esteri di interesse per l'attività dell'unità (specialmente per i giovani ricercatori)
Pubblicazioni	0	Nessuna spesa preventivata
Partecipazione / Organizzazione convegni	4.000	Per organizzazione di 1 workshop del progetto e partecipazione a conferenze di interesse per l'attività dell'unità
Altro	0	Nessuna spesa preventivata
TOTALE	48.000	

Testo inglese

Voce di spesa	Spesa in Euro	Descrizione
Materiale inventariabile	6.000	PC for development and testing. Printers and other devices.
Grandi Attrezzature		No planned costs
Materiale di consumo e funzionamento	5.000	Consumables, costs for the support of reseach activity, costs associated to the use of printers, telephone and fax.
Spese per calcolo ed elaborazione dati	0	No planned costs
Personale a contratto	18.000	One year contract for a young researcher for the development and testing of model checking procedures based on constraint solving
Servizi esterni	0	No planned costs
Missioni	15.000	Participation to project meetings and visits to national ant foreign research labs (especially planned for our young researchers) working on topics related to the unit activities
Pubblicazioni	0	No planned costs
Partecipazione / Organizzazione convegni	4.000	To organize 1 workshop for the project and for attending conferences on topics related to the unit activities
Altro	0	No planned costs
TOTALE	48.000	

3.2 Costo complessivo del Programma di Ricerca

		Descrizione
Costo complessivo del Programma dell'Unità di Ricerca	48.000	
Fondi disponibili (RD + RA) comprensivi dell'8% max per spese di gestione	14.400	Cofinanziamento di ateneo piu fondi acquisibili
Cofinanziamento di altre amministrazioni		

Cofinanziamento richiesto al MIUR	33.600
--	--------

3.3.1 Certifico la dichiarata disponibilità e l'utilizzabilità dei fondi di Ateneo (RD e RA)

SI

(per la copia da depositare presso l'Ateneo e per l'assenso alla diffusione via Internet delle informazioni riguardanti i programmi finanziati e la loro elaborazione necessaria alle valutazioni; D. Lgs, 196 del 30.6.2003 sulla "Tutela dei dati personali")

Firma _____

Data 05/04/2005 ore 13:24