Abstract (in Italian)

Il lavoro di questa tesi è incentrato sul concetto di sintesi. Nella verifica formale di sistemi il termine sintesi è legato sia al problema della sintesi dei controllori che alla sintesi di sistemi. Questo lavoro combina entrambe queste aree di ricerca.
Come primo aspetto, la tesi si concentra sul problema di sintetizzare controllori modulari, ossia consideriamo giochi su grafi ricorsivi (recursive game graph) con il requisito che la strategia per il protagonista sia modulare. Un recursive game graph è composto da un insieme di moduli, i cui vertici possono essere standard o corrispondere ad invocazioni di altri moduli, e le posizioni sono divise in due insiemi controllati rispettivamente da due giocatori diversi. Una strategia è modulare se per il protagonista la mossa successiva può essere scelta solo sulla base della storia locale del modulo, quindi non dipende dal contesto in cui il modulo è invocato. In questa tesi viene considerato per la prima volta il problema della sintesi di controllori modulari rispetto a condizioni di vittoria espresse attraverso linguaggi su automi pushdown. Dimostriamo che questi giochi, detti pushdown modular game, sono indecidibili in generale, ma diventano decidibili per specifiche espresse da automi visibly pushdown. Inoltre caratterizzeremo precisamente la complessità computazionale del problema di decisione considerato. In particolare, mostriamo che i giochi modulari con specifiche espresse da automi visibly pushdown universali di Buchi or co-Buchi appartengono alla classe dei problemi EXPTIME-complete e, quando la condizione di vittoria è espressa come una formula delle logiche temporali CaRet oppure Nwtl, il problema è 2EXPTIME-complete. Il problema resta 2EXPTIME-complete anche per frammenti molto semplici di tali logiche. Come ulteriore contributo, presentiamo un ulteriore algoritmo che, in caso in cui consideriamo formule molto lunghe e moduli con numerose uscite, presenta una complessità computazionale migliore.
Nella seconda parte della tesi, introduciamo e risolviamo un nuovo problema di sintesi di sistemi basato su componenti, che generalizza sia il problema della sintesi da librerie introdotto da Lustig e Vardi che il problema della sintesi di controllori modulari introdotto da Alur e altri. Le componenti delle nostre librerie sono moduli di gioco in cui i punti di chiamata non sono associati ad altre componenti, e consideriamo inizialmente un problema semplice, ossia le specifiche di correttezza richiedono di raggiungere almeno un vertice contenuto in un dato insieme. Per risolvere questo problema presentiamo un algoritmo di punto fisso, il cui tempo di esecuzione è esponenziale nella taglia delle specifiche: l'algoritmo calcola annotazioni dei vertici, esplorandoli a ritroso a partire dai nodi dell'insieme target. Inoltre, dimostriamo il limite inferiore della complessità riducendo il problema delle macchine alternanti di Turing a spazio lineare al nostro problema e questo dimostra l'EXPTIME-completeness. Proponiamo in aggiunta un secondo algoritmo che risolve questo problema eseguendo l'annotazione dei vertici attraverso una tavola e risolvendo i problemi di raggiungibilità localmente per ogni componente. Questo algoritmo risulta essere esponenziale solo nel numero di uscite delle componenti, e dimostra che il problema è fixed-parameter tractable.
Infine, introduciamo un problema di sintesi basato su componenti per sistemi pushdown che è ancora più generale del precedente. Il problema prende il nome di sintesi modulare da libreria di componenti (in breve LMS). Come nel precedente modello, le componenti delle nostre librerie sono moduli di gioco in cui i punti di chiamata rappresentano invocazioni ad altri moduli, ma adesso la composizione può essere guidata per alcuni punti, prevedendo una mappatura di tali punti ad altre componenti. Una istanza di una componente è una copia di essa con una strategia che risolve il non-determinismo del protagonista. Quello che vogliamo ottenere è un sistema pushdown sintetizzato dalla libreria, ossia un insieme di istanze con una funzione che associa ogni punto di chiamata ad una istanza del sistema e che sia consistente con la mappatura della libreria. Presentiamo una soluzione al problema LMS con condizioni di vincita espresse come vertici da raggiungere, automi a stati finiti deterministici e automi visibly pushdown. Dimostriamo che per tutte le specifiche considerate il problema è EXPTIME-complete.

Abstract (in English)

This thesis is focused on synthesis. In formal verification synthesis can be referred to the controller synthesis and the system synthesis. This work combines both this area of research.
First we focus on synthesizing modular controllers considering game on recursive game graph with the requirement that the strategy for the protagonist must be modular. A recursive game graph is composed of a set of modules, whose vertices can be standard vertices or can correspond to invocations of other modules and the standard and the set of vertices is split into two sets each controlled by one of the players. A strategy is modular if it is local to a module and is oblivious to previous module invocations, and thus does not depend on the context of invocation. We study for the first time modular strategies with respect to winning conditions that can be expressed languages of pushdown automata. We show that pushdown modular games are undecidable in general, and become decidable for visibly pushdown automata specifications. We carefully characterize the computational complexity of the considered decision problem. In particular, we show that modular games with a universal Buchi or co-Buchi visibly pushdown winning condition are EXPTIME-complete, and when the winning condition is given as a CaRet or Nwtl temporal logic formula the problem is 2EXPTIME-complete, and it remains 2Exptime-hard even for simple fragments of these logics. As a further contribution, we present a different synthesis algorithm that runs faster than known solutions for large specifications and many exits.
In the second part of this thesis, we introduce and solve a new component-based synthesis problem that subsumes the synthesis from libraries of recursive components introduced by Lustig and Vardi with the modular synthesis introduced by Alur et al. for recursive game graphs. We model the components of our libraries as game modules of a recursive game graph with unmapped boxes, and consider as correctness specification a target set of vertices. To solve this problem, we give an exponential-time fixed-point algorithm that computes annotations for the vertices of the library components by exploring them backwards. We show a matching lower-bound via a direct reduction from linear-space alternating Turing machines, thus proving EXPTIME-completeness. We also give a second algorithm that solves this problem by annotating in a table the result of many local reachability game queries on each game component. This algorithm is exponential only in the number of the exits of the game components, and thus shows that
the problem is fixed-parameter tractable.
Finally, we study a more general synthesis problem for component-based pushdown systems, the modular synthesis from a library of components (LMS). We model each component as a game graph with boxes as placeholders for calls to components, as in the previous model, but now the library is equipped also with a box-to-component map that is a partial function from boxes to components. An instance of a component C is essentially a copy of C along with a local strategy that resolves the nondeterminism of the protagonist. An RSM S synthesized from a library is a set of instances along with a total function that maps each box in S to an instance of S and is consistent with the box-to-component map of the library. We give a solution to the LMS problem with winning conditions given as internal reachability objectives, or as external deterministic finite automata (FA) and deterministic visibly pushdown automata (VPA) (6). We show that the LMS problem is EXPTIME-complete for any of the considered specifications.