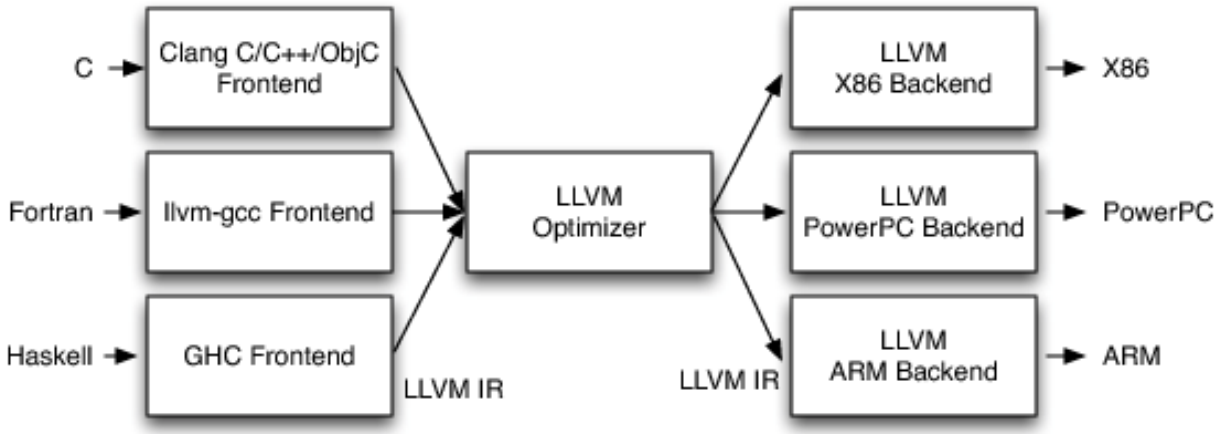
Cíl práce

Redukce stavového prostoru paralelních programů.

- Vstup: program v jazyce LLVM IR + Posix Threads
- Výstup: přechodový systém v jazyce NTS
- Technika: redukce částečným uspořádáním (POR)

LLVM IR

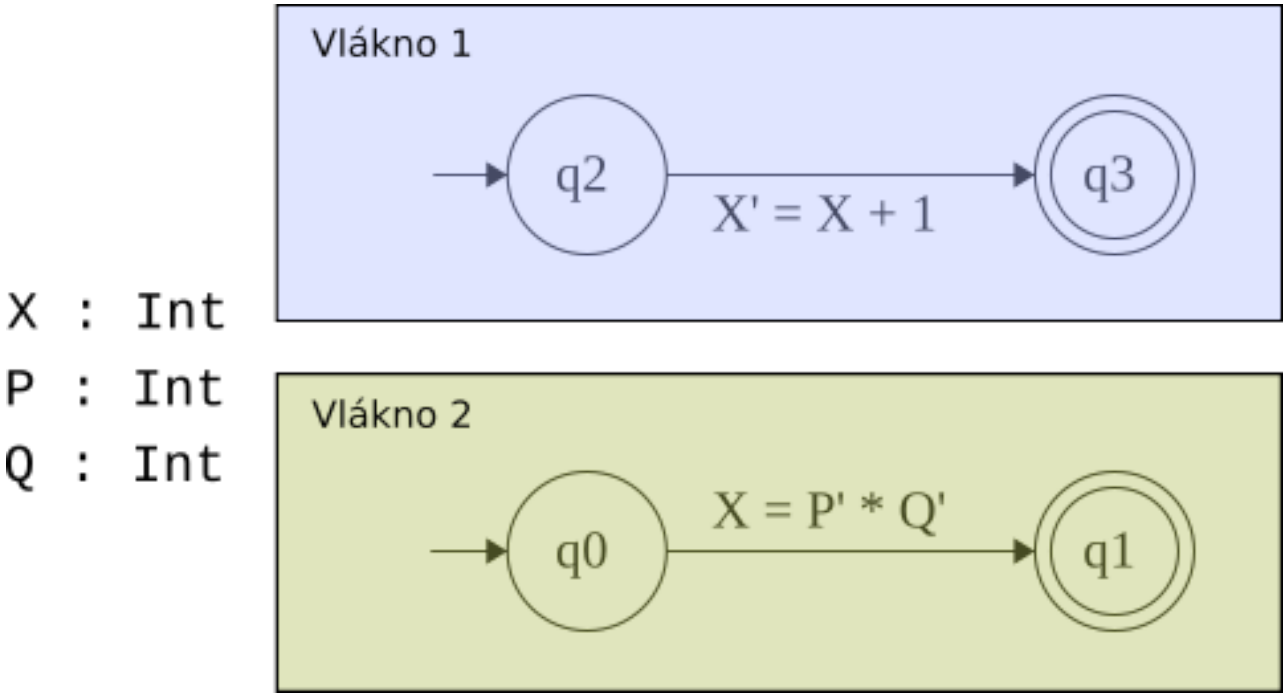


Obrázek: Použití LLVM[1]

- LLVM - framework pro tvorbů překladačů
- LLVM IR - Jazyk pro mezikód ve stylu assembleru
- LLVM IR = assembler (RISC) + typy + funkce

[1] Zdroj obrázku: http://www.aosabook.org/en/llvm.html

Přechodové systémy



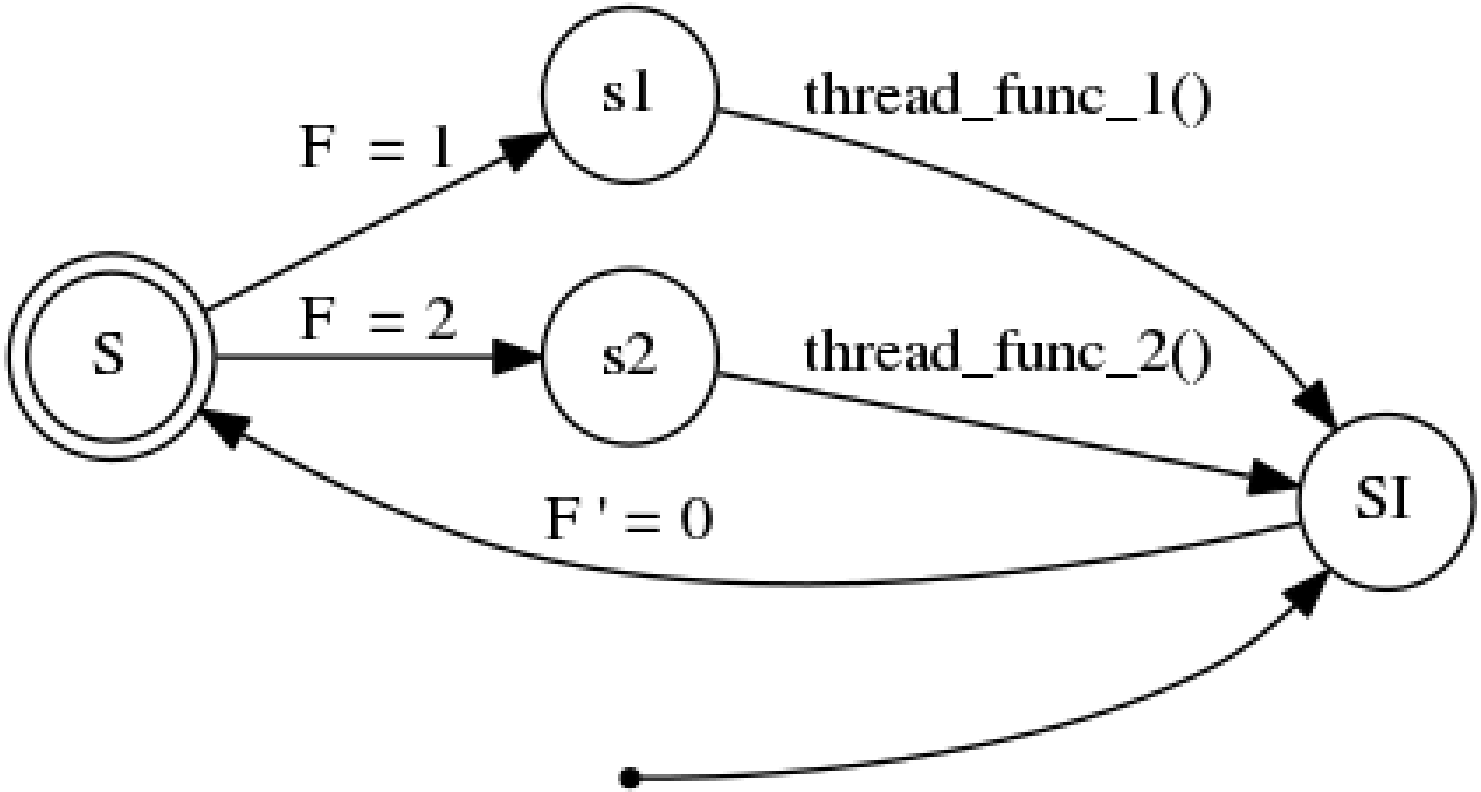
Obrázek: Příklad přechodového systému

- Přechodový systém = konečný automat + proměnné
- NTS - jazyk pro popis přechodových systémů
- Předem daný počet vláken (a jejich kód)

Přístup

- Řešení = překlad + inlining + redukce
- Vlastní reprezentace jazyka NTS (+ rozšíření)
- Žádné předpoklady o vytváření vláken.
- Omezená množina LLVM IR (bez pointerů, signed aritmetiky, . . .)

Paralelismus



Obrázek: Kód pracovního vlákna. Proměnná `F` obsahuje číslo funkce, která má běžet, a je nastavená voláním funkce `pthread_create()`.

Problémy

- Rozdílný model paralelismu => méně efektivní redukce
- Závislost na datech => zkoušení nemožných cest
- Hrubé heuristiky => méně efektivní redukce

Moje práce

- Vybudování reprezentace NTS
- Rozšíření NTS o datový typ BitVector.
- Překlad podmožiny LLVM IR do NTS
- Sekvencializace paralelního NTS (POR).

Výsledky

- Experimentů: málo
- Redukovaný systém je zhruba poloviční (v počtu stavů i hran) oproti neredukovanému.
- Hrubé aproximace, nízká efektivita.

Možná rozšíření

- Lepší heuristiky
- Externí solver
- Další části LLVM a Posix Threads

Dotaz: vylepšení heuristiky pro C1

Otázka

„Je možné vylepšit heuristiku pro volbu A(Q) tak, aby nebylo nutné brát v potaz přechody z neaktivních úloh?"

- Úloha = vlákno v původním programu
- C1: „Zvolíme - li jako A(Q) množinu povolených přechodů z vlákna P, pak nesmí existovat posloupnost přechodů mimo toto vlákno, z nichž některý by byl závislý na některém přechodu z A(Q)."
- Jednoduchá heuristika: „žádné z ostatních vláken neobsahuje přechod závislý na přechodu z A(Q)".
- Problém: každé pracovní vlákno obsahuje přechody všech úloh.

Dotaz: vylepšení heuristiky pro C1

Otázka

„Je možné vylepšit heuristiku pro volbu A(Q) tak, aby nebylo nutné brát v potaz přechody z neaktivních úloh?"

- Požadovaná heuristika: „žádná z ostatních aktivních úloh neobsahuje přechod závislý na některém přechodu z A(Q)".
- Stačí ukázat, že žádný běh mimo zvolené vlákno nemůže aktivovat novou úlohu.
- Třeba vědět, kdy je úloha aktivována.
- Speciální znalost: „Pro aktivaci úlohy je nutné zavolat funkci `thread_create`"
- Znalost dat: „Tato proměnná nikdy nenabude takovou hodnotu, aby byl přechod povolený."